Journal of Computing in Higher Education
Fall 2003, Vol.15(1), 65-88.

Agent Toolkit Satisfaction and Use in Higher Education

Alexander Serenko
Brian Detlor
Michael G. DeGroote School of Business
McMaster University

ABSTRACT

THIS PAPER presents results from a recent research investigation on the satisfaction and use of intelligent agent toolkits by instructors in higher education. An adaptation of the Technology Acceptance Model (TAM) linking agent toolkit satisfaction and usage to key characteristics of user considerations, performance, and functionality serves as the study's theoretical framework.

Data collection consists of the completion of an online questionnaire by 87 international instructors of agent-related courses. Results indicate that no single uniform toolkit satisfies the needs of instructors. Moreover, findings suggest that satisfaction levels are influenced primarily by user interactions with the toolkit, followed to a lesser extent by toolkit performance and functionality. This has a bearing on the utility of agent toolkits in the classroom as results point to a strong relationship between instructor satisfaction and the continuation of use of agent toolkits in future agent-related courses. Characteristics of an ideal agent toolkit for the classroom are also identified.
(Keywords: agent toolkits, intelligent agents, higher education, satisfaction, technology acceptance model)

INTRODUCTION

IN TODAY'S INTERNET WORLD, a newer form of software, called *intelligent agents,* offers people the potential to navigate and utilize Web-based information resources more effectively and efficiently than ever before. Intelligent agents are software programs that act on behalf of users to find and filter information, negotiate for services, automate complex tasks, and collaborate with other agents to solve complex problems (AgentBuilder, 2000). Agents perform these tasks continuously and autonomously in particular environments often inhabited by other agents and processes (Shoham, 1997, pp. 271-72).

The use of agents has been well-documented in the electronic commerce domain (Maes, 1999; 1994; Maes, Guttman, & Moukas, 1999; Rahman & Bignall, 2001), especially in terms of industrial, commercial, medical, and entertainment applications (Jennings & Wooldridge, 1998). With the advent of the Semantic Web proposed by Tim Berners-Lee, agents are envisioned to play a more significant role in the near future (Port, 2002; Berners-Lee et al., 2001). Given the importance and rise of this newer form of software, agent toolkits are becoming more of a necessity to help build, reuse, and deploy intelligent agents. As such, agent toolkits are being introduced and incorporated in the curriculums of postsecondary education courses geared to train the next generation of electronic commerce managers and programmers.

To gain insight on the use of agent toolkits in higher education, a project was conducted exploring the satisfaction of instructors utilizing agent toolkits in the classroom. Data collection and analysis involved the deployment of a Web-based questionnaire to 87 international postsecondary course instructors. This paper reports on the project's findings.

BACKGROUND

AGENT TOOLKITS

THERE IS NO UNIVERSAL DEFINITION OF AGENT TOOLKITS. Each vendor uses its explanation of the term. For example, authors of the Java Agent Development Environment (JADE) define their toolkit as "a software framework to make easy the development of agent applications . . . for interoperable multi-agent systems" (Bellifemine et al., 2000). Reticular Systems states that its AgentBuilder toolkit application "is an integrated tool suite for constructing intelligent software agents" (AgentBuilder, 2000).

For this paper, an agent toolkit is defined to be any software package, application, or development environment that provides agent builders with a sufficient level of abstraction to allow them to implement intelligent agents with desired attributes, features, and rules. Some toolkits may offer only a simple environment for creating basic agent systems, whereas others may provide a complicated platform for agent development with features for visual programming.

Although agent toolkits have had a relatively short history on the software market so far, there are many toolkits now available. They differ in terms of the functionality they offer, their ease of use, area of application, and underlying technology. Overall, four major categories of agent toolkits are identified: mobile agent toolkits, multi-agent toolkits, general-purpose toolkits, and Internet agent toolkits (Serenko & Detlor, 2002).

The need for agent toolkits has been well documented in the agent literature (Eiter & Mascardi, 2001; Jennings et al., 1998; Sloman, 1998; Wooldridge & Ciancarini, 2001). First, toolkits provide a certain level of abstraction and encapsulation in which programmers can develop their objects. Second, toolkits incorporate some features of visual programming, which save time and make development easier, more attractive, and enjoyable. This is critical given the constantly increasing complexity of software and the skyrocketing costs of development and deployment of agent software systems. Third, they offer

run-time simulation, monitoring, analyzing, testing, and debugging environments. Luck et al. (1997) and Wooldridge and Jennings (1998) have highlighted the importance of such environments. Last, agent toolkits allow programmers to reuse classes of previous designs so that new developers do not have to start from scratch every time. This is essential given that creating intelligent agent software currently requires significant training and skills (Winikoff et al., 2001).

Many of the reasons why agent developers use agent toolkits are similar to the reasons why software developers who deal with object-oriented programming (OOP) prefer to use special development environments like Java VisualAge or MS Visual Basic. However, such OOP development platforms and compilers do not support all facets of agent development such as agent interaction rules, communication languages, and common knowledge bases. This is why agent toolkits have emerged on the software market in the last few years: to provide a development environment that fully supports agent creation, analysis, testing, debugging, and reuse.

AGENT TOOLKIT SATISFACTION

Currently, more postsecondary schools are offering graduate and undergraduate courses in intelligent agents where students study agent technologies by experimenting with existing agents or building new ones. Preliminary research has shown that students benefit from utilizing agent-based multimedia environments in introductory artificial intelligence classes. These benefits include increases in subject interest, confidence in course material, and the ability to understand and utilize presented techniques (Holder & Cook, 2001). However, despite these findings, more research is needed that examines agent toolkit utilization in higher education and the success instructors have had with them.

In order to gain insight on the major factors affecting the level of instructor satisfaction with agent toolkits, a comprehensive review of the agent construction tools literature was conducted. Three major determinants were identified: functionality—whether the toolkit matches an instructor's needs; performance—whether the toolkit is

capable of implementing all advertised functions; and user considerations—whether the toolkit offers a user-friendly and positive experience.

In terms of agent toolkit **functionality**, several variables were discovered. First, the toolkit should be capable of creating agents that inherit basic *agent features and functions*. Second, both a *simulation environment and infrastructure* are needed to run and test developed agents (Luck et al., 1997; Sloman, 1998; Sun, 2000). Such underlying infrastructures are needed to reduce the time required to design and implement agents (Wooldridge & Jennings, 1998). Third, *debugging facilities* are needed to eliminate errors in the difficult and complex task of agent building (Jennings et al., 1998; Winikoff et al., 2001). Fourth, the availability of *agent skeletons and templates* are required to help agent developers build applications faster and more easily (Eiter & Mascardi, 2001). Fifth, *sample demo agents and prototypes* are necessary to demonstrate the capabilities that an agent toolkit can provide. Last, agent toolkits should inherit all the major techniques of object-oriented programming, including *code reusability* (Sun, 2000; Wooldridge & Jennings, 1998).

With respect to toolkit **performance**, *reliability*, *robustness*, *efficiency* and *stability* of agent toolkit were identified as being significant (Eiter & Mascardi, 2001; Howden et al., 2001).

In terms of **user considerations**, five variables were identified. These were the *graphical user interface*, *user training*, *vendor support, toolkit documentation*, and a *favorable learning curve* (Eiter & Mascardi, 2001; Sloman, 1998; Winikoff et al., 2001). Failing to address such user considerations can become an obstacle to the successful deployment of agent systems (Schoepke, 1999).

THEORETICAL FRAMEWORK

TO INVESTIGATE THE SATISFACTION AND USE of agent toolkits in higher education, a theoretical framework was developed based on the Technology Acceptance Model (TAM)

introduced by Davis (1989). Davis' model provides an explanation of the determinants of computer acceptance in general. Specifically, TAM identifies a causal relationship among several system design features: perceived usefulness, perceived ease of use, attitude toward use, and actual usage behavior (Davis, 1993). Figure 1 illustrates the workings of TAM.

According to the model, a user's attitude towards using a particular system is the major factor why he or she actually uses it. Two interrelated determinants: perceived usefulness and perceived ease of use affect the attitude towards utilizing a system. Davis (1989) defines perceived usefulness of the system as "the degree to which a person believes that using a particular system would enhance his or her job performance" and perceived ease as "the degree to which a person believes that using a particular system would be free of physical and mental effort."

The viability of TAM has been effectively tested in various technology acceptance studies. For example, Chau and Hu (2002) utilized the model to investigate telemedicine technology acceptance by physicians. Chen et al. (2002) successfully applied TAM to explain consumers' use of a virtual store. Horton et al. (2001) found that TAM may be a valuable tool for analyzing and understanding intranet usage. Al-Gahtani & King (1999) tested and extended the model by introducing several new variables, including compatibility, user characteristics, system rating and the end-user computing satisfaction (EUCS) construct. These studies support the viability of applying or extending TAM to test the user acceptance of newer technologies such as agent toolkits.

In order to adapt TAM to measure the level of satisfaction with agent toolkits, the model was modified and a number of new variables were introduced. Figure 2 presents an extended TAM for agent toolkits. The extensions were based on our review of the agent construction tools literature previously described.

According to this model, the level of instructor satisfaction with a toolkit is the most important factor that influences the decision to utilize an agent toolkit in subsequent courses. The level of satisfac-

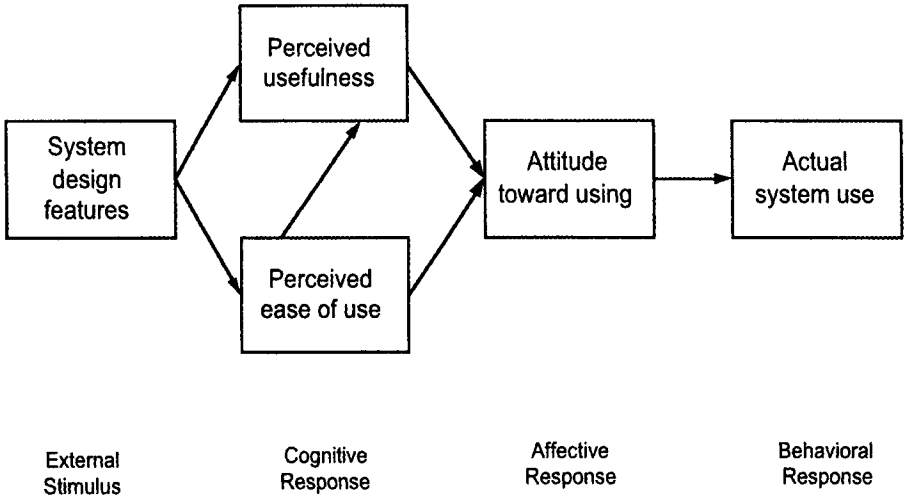

Figure 1. Davis' TAM (1993, p. 476). Reprinted with permission from Elsevier Science

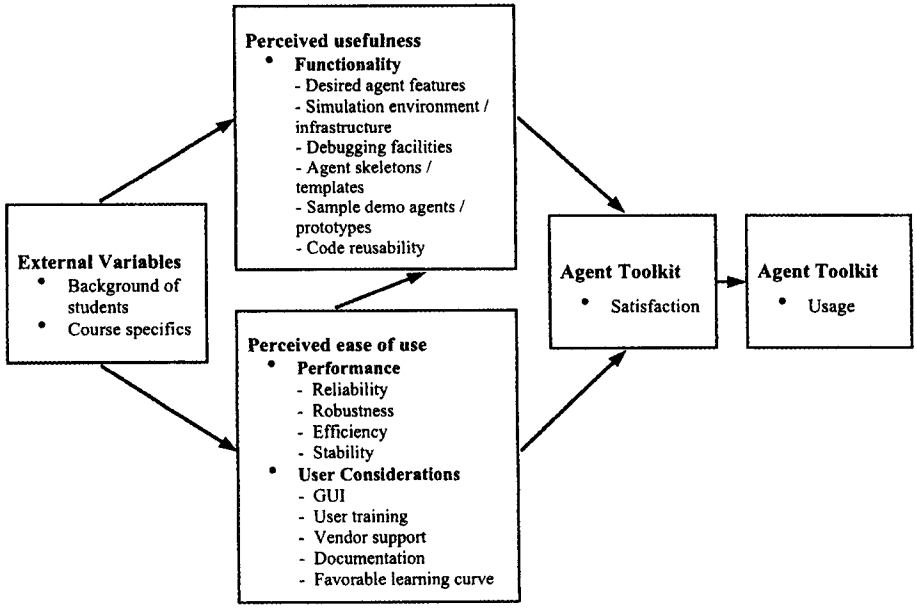

Figure 2. TAM for Agent Toolkits

tion, in turn, is jointly determined by perceived usefulness and ease of use of the system. In terms of perceived usefulness, as discussed earlier, a toolkit's functionality is the major determinant. It is represented by a number of variables such as the toolkit's capability of implementing all required agent features, and its provision of an adequate simulation environment, debugging facilities, agent skeletons, sample demo agents, and code reusability functions. With respect to perceived ease of use, there are two determinants: performance and user considerations. Performance is represented by reliability, robustness, efficiency, and stability. User-friendly GUI, user training, vendor support, the comprehensiveness of the toolkit's documentation, and a favorable learning curve represent user considerations. The model presumes that comprehensive functionality, strong performance, and well-addressed user needs would lead to higher satisfaction of agent developers. Last, the model incorporates several external variables such as the background of students and course specifics that, in turn, affect perceived usefulness and ease of use with the toolkit.

RESEARCH QUESTIONS

Considering the purpose of the study and utilizing Figure 2 as a theoretical guide, the project addresses the following research questions:

1. In general, how are agent toolkits utilized in postsecondary courses today?
2. What is the relationship between user satisfaction and agent toolkit functionality, user satisfaction and toolkit performance, and between user satisfaction and user considerations?
3. To what extent does the level of satisfaction with an agent toolkit affect its planned future usage by instructors teaching agent-related courses?
4. What are the characteristics of an ideal agent toolkit for higher education?

METHODOLOGY

IN ORDER TO OBTAIN SUFFICIENT INFORMATION to answer these research questions, a Web-based questionnaire was devised. The questionnaire was tailored to instructors of agent-related postsecondary courses who currently utilized agent toolkits in the classroom, as well as those who might in the future.

Instructors who did not utilize agent toolkits were asked to provide insights on the reasons why they did not include an agent toolkit in their course curriculums and to give their personal opinions regarding possible benefits of agent toolkits as teaching tools.

Instructors who did utilize agent toolkits were asked a different set of questions pertaining to the specific toolkit used and the course in which it was employed. These instructors were explicitly asked to rate their satisfaction with the toolkit and to highlight the underlying reasons for their rating. Further, they were polled to identify those features that an "ideal" agent toolkit should possess. Finally, to project a trend on the future use of agent toolkits in academic courses and to identify a possible relationship between satisfaction and future toolkit usage, the instructors were asked about their future plans to use the toolkit in subsequent courses.

Participants were recruited in two ways: (1) an intensive Web search and (2) a general call for participants through the ISWorld listserv (see http://www.isworld.org). The Web search for college and university instructors who taught agent-related courses involved visiting a number of discussion forums and Web portals pertaining to agents, as well as the use of search engines and directories. This strategy identified 256 potential participants. Each was contacted through a personalized e-mail message that explained the purpose of the project and that asked them to take a few minutes to complete the online questionnaire. In the end, 77 participants were recruited in this manner.

With respect to the listserv, ISWorld is an international Web portal dedicated to servicing the needs of information systems academics.

It was thought that course instructors having an interest in agent toolkit technologies would be members of this community. The call for participants reached 2,927 recipients and yielded an additional 10 responses.

In total, 87 questionnaires were completed. Though not an overly large number, we believe that a sizeable and random representation of academics with interests in utilizing agent toolkits in their curriculums was achieved. This statement is based on our extensive search that found a limited number of course syllabi pertaining to agent technologies in general.

To analyze the collected data, descriptive statistics and t-tests were conducted on the quantitative portions of the questionnaire. Content analysis was performed on segments of the questionnaire where respondents replied with free-form text.

FINDINGS

OF THE 87 RESPONDENTS who filled out questionnaires, 25 utilized agent toolkits and 62 did not. The questionnaire results for each of these two sets of respondents are discussed separately in the following two subsections.

INSTRUCTORS UTILIZING AGENT TOOLKITS

The study identified 23 different toolkits used by the 25 respondents who utilized agent toolkits in their courses. Most instructors stated that they used only one toolkit in their courses; a few identified a couple of toolkits. JADE was the most popular toolkit utilized in that it was identified by five instructors. Overall, these findings suggest that instructors utilized a wide variety of toolkits, and that, with the exception of JADE, there was no definitive toolkit utilized by a sizeable percentage of instructors. It appears that instructors may acquire toolkits that best match an instructor's specific course requirements and that no single "uniform toolkit" exists that may meet the specific needs of all agent-related courses.

Table 1.

Course Name	Agent Toolkits Utilized
Intelligent agents	AgentBuilder, Agora, IBM Aglets, Grasshopper, JADE, Pathwalker
Multi-agent and multi-robot systems	ABLE, DECAF, FIPA-OS, Jade, Jack, MACE3J, MADKit, Pathwalker, RePast, TeamBots
Agent technologies	JADE, JESS, ZEUS
Artificial intelligence (Distributed AI)	Agora, JACK, MICE, SimAgent, Wumpus World Simulator
Knowledge-based systems	Soar
Machine learning	GA Playground
Information gathering	DECAF

Table 1 lists the agent toolkits according to the name of the agent course in which they were used. The categories of agent course names were derived from a content analysis of agent course titles identified in the participant questionnaires. Figure 3 summarizes the results of Table 1 by showing the percentage breakdown of agent toolkit usage by course name category. Note some toolkits appear across more than

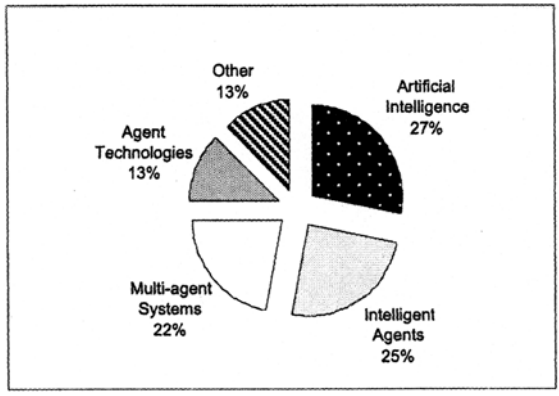

Figure 3. Types of courses in which the agent toolkits were used.

one course name categories (i.e., they were not utilized in just one type of course). The degree to which the course focused on artificial intelligence (AI) was a major distinction between courses. Dividing the courses in this way, it was found that AI focused courses constituted 36% of all courses in which agent toolkits were utilized and that non-AI focused courses comprised 64%.

Recall that one of the questions on the questionnaire asked instructors to rate their level of satisfaction with the agent toolkits. Specifically, two satisfaction-related questions were asked: one concerning the instructors' level of satisfaction and another about their perception of students' satisfaction with the toolkits. Scores for these questions utilized a Likert scale of 1 to 4 (1 being "not satisfied" and 4 being "very satisfied"). Though there were some deviations in the levels of satisfaction rated by instructors for these two questions, in general the scores given for these two questions were identical. The response scores from these two questions were averaged into a single index of toolkit satisfaction. Across all instructors, the average satisfaction index score was 2.59 (somewhat satisfied/satisfied).

In terms of AI course orientation, it was found that the average satisfaction level of AI focused courses was 2.89 and that of non-AI courses was 2.47. A t-test confirmed the average satisfaction level of AI focused courses was higher than non-AI focused courses ($t = 2.06$, $p < .05$).

In terms of toolkit manufacturer, the study showed that 52% of utilized toolkits were developed by academic institutions, 39% by commercial, and 9% by nonprofit manufacturers. It was found that the average satisfaction level for academically developed toolkits was 2.83 and for commercially developed toolkits 2.36. A t-test confirmed that the average satisfaction level of academically developed toolkits was higher than that of commercially developed ones ($t = 2.48$, $p < .05$).

Recall the three determinants identified in the Technology Acceptance Model for agent toolkits: functionality, performance and user considerations. Utilizing these three determinants as a usability guide, a content analysis was performed on the free-form text responses made by participants in the questionnaire for the two items which asked instructors to explicate the reasons behind their indicated levels of satisfaction with the toolkits.

Figure 4 displays the overall breakdown of this content analysis for those comments elicited by participants, which implied satisfaction with the toolkit. Roughly half of the responses pertained to user consideration variables as discussed in the framework section (e.g.,

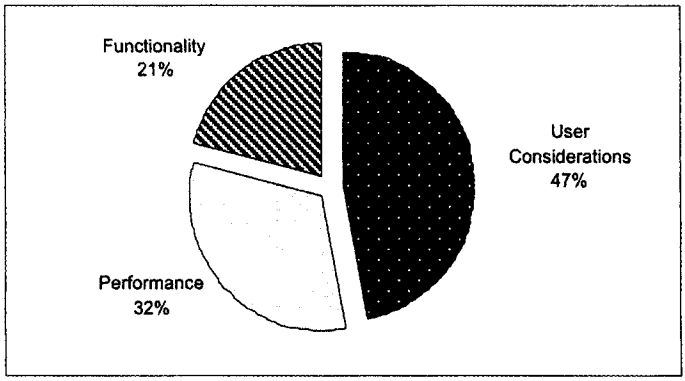

Figure 4. High level reasons why instructors were satisfied with toolkits.

comprehensiveness of the software's documentation, short learning curve, toolkit familiarity, and the user friendliness of the software interface). Almost one third of the responses related to performance variables (e.g., performed well, stability of the software, powerfulness of the toolkit) and only over one fifth of the responses pertained to functionality (e.g., the software supported what users wanted).

Figure 5 displays the overall breakdown of the comments elicited by participants, which implied dissatisfaction with the toolkit. Again, almost half of the responses pertained to user considerations variables (e.g., high complexity, substantial degree of programming required, poor software documentation). However, the ratios for functionality and performance were reversed. Almost one third of the responses related to functionality variables (e.g., lack of important functions and features). Over one fifth of the responses pertained to performance (e.g., not properly working features, instability).

There were several recurring comments made by instructors in terms of the features an "ideal" agent toolkit should possess. First, the toolkit should be user-friendly. Clear graphical user interfaces would help visualize activities and help students better understand agent concepts. A number of teaching aids should be provided. These include features such as: examples of working, collaborating agents along with their sample code; agent building templates; and strong support-

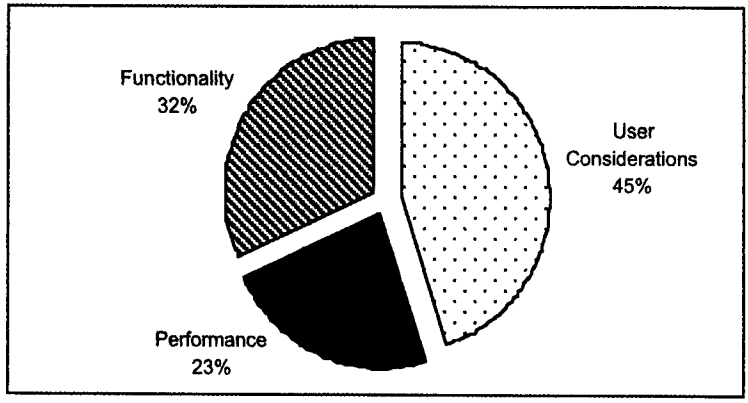

Figure 5. High level reasons why instructors were dissatisfied with agent toolkits.

ing user documentation. Such tools are critical for student success with agent toolkits. The environment should respect and accommodate different levels of student computer programming abilities. Second, a toolkit should provide a persistent, flexible, robust, and reliable environment in which agents can be constructed. Third, the environment should support artificial intelligent capabilities for agents, such as knowledge representation and problem solving. Last, the toolkit should be relatively inexpensive and easy to install to facilitate wide adoption and use throughout academia.

Figure 6 summarizes the high-level breakdown in terms of functionality, performance and user considerations.

Table 2 summarizes findings from Figures 4, 5, and 6. The table demonstrates the relative importance of user considerations characteristics in agent toolkits, as expressed by instructors utilizing toolkits in their courses.

Of the 25 instructors who utilized agent toolkits in their courses, 88% stated they would utilize such toolkits again. These instructors believed that the toolkits enriched their programs, forming an integral part of the course and providing a useful aid for teaching students about agent behavior. The instructors stated that the toolkits fostered student understanding of agent programming, technologies, and concepts by allowing students to implement their own agents. Such tasks

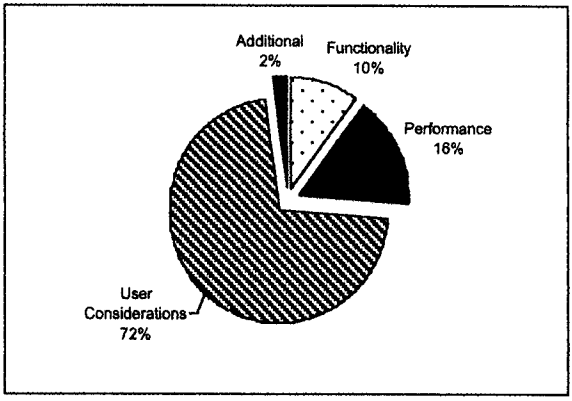

Figure 6. User considerations factors deemed desirable by instructors in an agent toolkit.

Table 2.
Summary of Instructor Responses

Usability Factor	Satisfaction Reasons	Dissatisfaction Reasons	"Ideal" Toolkit Characteristics
Functionality	21%	32%	10%
Performance	32%	23%	16%
User Considerations	47%	45%	72%

encouraged students to think at higher levels of abstraction. It was also believed that students gained valuable hands-on experience with toolkits they likely would confront when working in industry. The 12% of instructors who stated they were not going to continue utilizing agent toolkits in subsequent courses were either "not satisfied" or "somewhat satisfied" with their toolkit experience that resulted from poor performance, limited functionality, and lack of user support.

INSTRUCTORS NOT UTILIZING AGENT TOOLKITS

In contrast to the group of instructors discussed in the previous section, this set of questionnaire respondents were instructors of postsecondary agent-related courses who did not utilize agent toolkits. Figure 7 illustrates the breakdown of responses made by this group of instructors on whether agent toolkits would be beneficial to use.

As the diagram shows, the majority (61%) believed that agent toolkits would be beneficial to use in the agent-related courses they taught for the following reasons. First, given the importance of agents and agent-related research in today's Internet world, these instructors felt it was critical that students not only understand the fundamentals and underlying theory pertaining to agents, but also learn modern agent technologies and tools.

Second, these instructors believed that theoretical examples were insufficient for students and that they needed working examples of agents to convince them of the usefulness of agent technologies and research. They also believed that hands-on experience with the design, creation, and implementation of agents was the best way of enabling student comprehension of agents and of fostering student interest in agent-related research.

Third, these instructors felt that agent toolkits would allow students to concentrate more on agent issues (such as the actions an agent must perform based on another agent's communications) rather than lower-level technical issues (such as writing code to parse packets of data sent in agent communication). Lower level technical issues tend to involve too much programming and distract students from understanding higher-level agent concerns taught in the course material.

Despite this enthusiasm for toolkits, these instructors cautioned that toolkits be used with care. Agent toolkits are like any software teaching aid and need to be introduced and utilized in courses appropriately. That is, they should be used in the right context with the right material and in the right circumstances.

Of the 23% of instructors who thought that toolkits would offer little benefit to students, the majority framed this comment in context of the current agent-related course they were teaching. These tended to be introductory agent courses. Here, the instructors felt the time students would spend learning the toolkit would be better utilized on other parts of the course that placed more emphasis on basic concepts and fundamentals. A few instructors commented that agent toolkits would be better utilized in more advanced agent courses. Reasons for not utilizing agent toolkits are summarized in Figure 8.

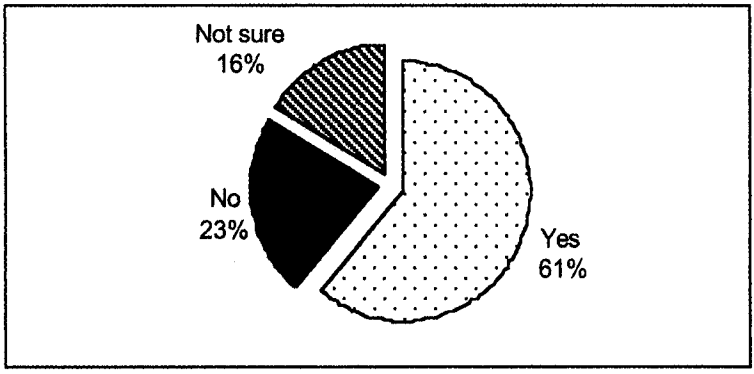

Figure 7. Breakdown of instructors on whether toolkits would be beneficial to use in the classroom.

Inappropriateness for the course was the predominant response. This pertained primarily to basic or introductory courses where most time is spent on teaching fundamentals rather than agent engineering, or where the length of the course is short (e.g., six weeks in duration) limiting the amount of time instructors can dedicate to implementing agent technologies in the classroom. Most respondents considered creating agents an advanced proposal in their courses.

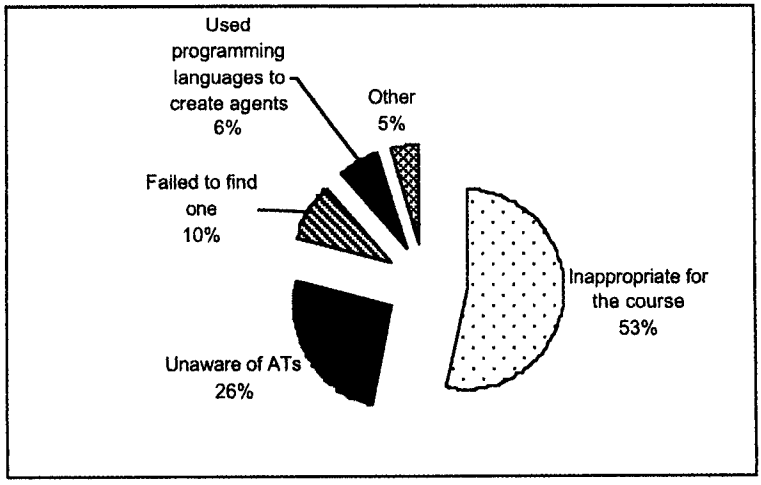

Figure 8. Reasons why instructors did not use agent toolkits.

Instructor unawareness of the existence of agent toolkits in general was the second most popular response. This was followed by the inability to find an appropriate toolkit that matched the specific course requirements. The instructors who indicated this reason also lamented the extensive reliance on advanced programming techniques in the current batch of available toolkits on the market, making the toolkits inappropriate as teaching aids for students without extensive programming knowledge. Despite these instructors' inability to find an appropriate toolkit after conducting an extensive search for one, all believed that the usage of toolkits would be beneficial for their courses.

Utilizing existing object-oriented languages and platforms to create intelligent agents was a response given by the instructors who did not want to restrict their students to any particular language or application. In this case, students had to program features in their agents that toolkits would have provided, such as the ability to communicate and send messages to other agents.

The remaining responses covered a wide range of concerns. Here, instructors had not yet investigated the use of agent toolkits for their courses, or felt they did not have enough knowledge to include one.

DISCUSSION AND CONCLUSIONS

RECALL THE FOUR PURPOSES OF THE STUDY: (1) to investigate how agent toolkits are utilized in higher education; (2) to identify the relationship between user satisfaction and agent toolkit functionality, performance, and user considerations; (3) to examine how satisfaction with an agent toolkit affects its future usage in subsequent courses; and (4) to identify the features an "ideal" toolkit should possess. The study's findings were based on responses from 87 participants who filled out an online questionnaire. The participants were randomly chosen and formed a representative sample population of instructors teaching agent-related courses.

The research demonstrated that only a minority of instructors teaching agent-related courses currently included agent toolkits in their curriculums. Of toolkits that were used, no single uniform toolkit met the specific needs of all agent-related courses. Rather, a wide variety of toolkits were utilized across disparate types of agent-related courses. Academic manufacturers developed more than half of the toolkits used. "Intelligent agents" and "multi-agent systems" courses formed the predominant category of courses employing agent toolkits. Two-thirds of the courses that utilized toolkits were non-AI focused. Overall, the average rating of toolkit satisfaction ranged between "somewhat satisfied" to "satisfied." The average satisfaction level of instructors was higher for AI focused courses and for toolkits developed by academic manufacturers.

In terms of a relationship between the level of user satisfaction and determinants of the proposed Agent Toolkits TAM, the study demonstrated the relative importance of user considerations. This implies that user considerations with the toolkit were the leading characteristic of agent toolkits that influenced satisfaction levels over those of toolkit performance and functionality. Regarding user considerations, satisfaction levels were higher when the toolkit was easy-to-use (i.e., simple, non-programming intensive, quick to learn, accompanied by a comprehensive documentation set). With respect to performance, satisfaction levels were higher when the toolkit operated well (i.e., was stable). In terms of functionality, satisfaction levels were higher when the toolkit supported basic agent concepts.

The study also revealed the direct relationship between the level of instructor satisfaction with an agent toolkit and an instructor's decision to continue utilizing the toolkit in future courses. All highly satisfied instructors believed that the toolkits enriched their programs and that they would continue utilizing or demonstrating agent toolkits in their curriculums. At the same time, the instructors who had negative personal experience with the toolkits were going to either change the toolkit or stop using toolkits in their courses altogether.

Instructors identified several characteristics of an ideal toolkit. They suggested the toolkit should provide a persistent, reliable, flex-

ible, and easy-to-use environment for agent creation and deployment. They also stated the toolkit be accompanied by a comprehensive documentation set and several teaching aids, such as working examples, sample code, and agent generation templates. The majority of instructors who did not currently utilize toolkits in the classroom felt that toolkits would be beneficial for their students.

It should be noted that agent toolkits are in the early stages of development. The pace of adoption, however, is constantly increasing. We believe that more instructors will adopt toolkits in the classroom as agent toolkits become more easy-to-use, perform better, and support a wider range of agent-related functions, and as instructors themselves become more knowledgeable about agent toolkits on the market. We also believe that no particular toolkit will be appropriate for all instructors—the chosen toolkit needs to match individual course requirements and instructor preferences.

In general, we are encouraged by the future use of agent toolkits in postsecondary education. Agent toolkits are useful and practical applications for creating, deploying, and reusing agents. Though the toolkits are not at a stage of development yet which yields enthusiastic satisfaction scores by instructors of agent-related courses, the toolkits currently available on the market do offer distinct advantages over other software development environments. Most current development platforms, packages, and compilers do not allow creating software components with agent capabilities, such as personalization, productiveness, adaptiveness, and proactiveness. They also do not address implementation of other required agent features, such as agent interaction rules, agent communication, and common knowledge bases. Agent toolkits address these concerns by providing a conceptual level of abstraction, supporting agent functionality, and offering run-time testing and debugging environments.

A limitation of the study was its sample size of 87 instructors. The small sample size of 25 instructors who actually utilized agent toolkits in the classroom makes the study's finding difficult to generalize to the larger population of instructors of agent-related courses. However, despite this limitation, the study yielded pertinent prelimi-

nary findings that provide a good test bed for future research in this area. Several avenues can be explored. For example, the level of satisfaction of students can be investigated by surveying the students directly (and not just the instructors). It would be interesting to see if student satisfaction levels differed significantly from those of instructors, and more importantly, what usability factors affect student satisfaction levels. Another idea would be to poll industry participants utilizing agent toolkits and determine those criteria that affect satisfaction. Satisfaction levels for particular toolkits could be explored, perhaps by polling instructors who use the same toolkit in similar courses.

Overall, this report has shed light on the use of agent toolkits in higher education. One of its major contributions was the development of a theoretical framework for agent toolkit satisfaction and use. It is our hope that the framework may serve as a lens of examination for future research in this area.

ACKNOWLEDGEMENTS

This paper is partially supported by a grant from the Natural Sciences and Engineering Council of Canada.

REFERENCES

AgentBuilder. (2000). "What is AgentBuilder?" San Diego, CA: Reticular Systems. [Online]. Available: http://www.agentbuilder.com/Documentation/product.html.

Al-Gahtani, S., & King, M. (1999). Attitudes, satisfaction and usage: Factors contributing to each in the acceptance of information technology. *Behavoiur & Information Technology*, 18(4), 277-297.

Bellifemine, F., Poggi, A., Rimassa, G., & Turci, P. (2000). An Object-oriented framework to realize agent systems. *Proceedings of WOA 2000 Workshop* (pp. 52-57).

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 34-43.

Chau, P.Y.K., & Hu, P.J. (2002). Examining a model of information technology acceptance by individual professionals: An exploratory study. *Journal of Management Information Systems*, 18(4), 191-229.

Chen, L., Gillenson, M., & Sherrell, D. (2002). Enticing online consumers: An extended technology acceptance perspective. *Information & Management*, 39(8), 705-719.

Davis F. (1993). User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man-Machine Studies*, 38, 475-487.

Davis F. (1989). Received usefulness, perceived ease of use and user acceptance of information technology. *MIS Quarterly*, 13(3), 319-340.

Eiter, T., & Mascardi, V. (2001). Comparing environments for developing software agents. *INFSYS Research Report* (1843-01-02). Knowledge-Based Systems Group.

Holder, L.B., & Cook, D.J. (2001). A client-server computational tool for integrated artificial intelligence curriculum. *Journal of Computing in Higher Education*, 12(2), 1-13.

Horton, R.P., Buck T., Waterson, P.E., & Clegg, C.W. (2001). Explaining intranet use with the technology acceptance model. *Journal of Information Technology*, 16(4), 237-249.

Howden, N., Ronnquist, R., Hodgson, A., & Lucas, A., (2001). JACK intelligent agents – summary of an agent infrastructure. *Proceedings of 5th International Conference on Autonomous Agents*, 2nd International Workshop on Infrastructure for Agents, MAS and Scalable MAAS. Montreal, Canada.

Jennings, N., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1), 275-306.

Jennings, N.R., & Wooldridge, M.J. (1998). *Agent Technology Foundations, Applications, and Markets*. Berlin: Springer.

Luck, M., Griffiths, N., & d'Inverno, M. (1997). From agent theory to agent construction: A case study. *Proceedings of the ECAI'96 Workshop on Agent Theories, Architectures, and Languages*: Intelligent Agents III.

Maes, P. (1999). Smart commerce: The future of intelligent agents in cyberspace. *Journal of Interactive Marketing*, 13(3), 66-76.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7). 31-40.

Maes, P., Guttman, R., & Moukas, A. (1999). Agents that buy and sell: Transforming commerce as we know it. *Communications of the ACM*, 42(3), 81-91.

Port, O. (2002, March 4). The next web. *Business Week.* pp. 96-102.

Rahman, S.M., Bignall, R.J. (Eds.). (2001). *Internet commerce and software agents.* Hershey, Pennsylvania: Idea Group.

Schoepke, S.H. (1999, July 3,). Facilitating the deployment of intelligent agents in the application development mainstream. *AgentLink Newsletter* N3, pp. 10-12.

Serenko, A., & Detlor, B. (2002). Agent toolkits: A general overview of the market and an assessment of instructor satisfaction with utilizing toolkits in the classroom (*Working Paper 455*). Hamilton, Ontario, Canada: Michael G. DeGroote School of Business, McMaster University.

Shoham, Y. (1997). An overview of agent-oriented programming. In J.M. Bradshaw (Ed.), *Software Agents* (pp. 271-90). Menlo Park, CA.: AAAI Press.

Sloman, A. (1998, July). What's an AI toolkit for? Proceedings of AAAI-98 *Workshop on Software Tools for Developing Agents.* B. Logan. J. Baxter, (Eds.). Madison.

Sun Microsystems, Inc., (2000). Java dynamic management kit. (*White Paper*). [Online]. Available: http://www.sun.com/products-n-solutions/nep/whitepapers/JDMK4_april00.pdf [2002, August].

Winikoff, M., Padgham, L., & Harland, J. (2001). Simplifying the development of intelligent agents. *Proceedings of AI2001: Advances in Artificial Intelligence.* 14th Australian Joint Conference on Artificial Intelligence (pp. 557-568). LNAI 2256, Adelaide, Australia.

Wooldridge, M., & Ciancarini, P. (2001). Agent-oriented software engineering: The state of the art. In P. Ciancarini & M. Wooldridge (Eds.), Agent-Oriented Software Engineering (pp.1-28). Berlin: Springer-Verlag.

Wooldridge, M., & Jennings, N. (1998). Pitfalls of agent-oriented development. *Proceedings of the 2nd International Conference on Autonomous Agents* (Agents 98) (pp. 385-391). Minneapolis, MI.

ABOUT THE AUTHORS

Alexander Serenko is a PhD candidate at the Michael G. DeGroote School of Business, McMaster University. He holds a MSc in Computer Science and an MBA in eBusiness. His research interests include intelligent agents and knowledge management. Alexander Serenko is the Director of the Doctoral Consortium for the World Congress on the Management of Electronic Business, Intellectual Capital and Innovation at McMaster University. E-mail: serenkav@mcmaster.ca

Brian Detlor is an Assistant Professor of Information Systems at the Michael G. DeGroote School of Business, McMaster University. Dr. Detlor specializes in intelligent agents, electronic business, and Web information systems design. His research interests include the application of intelligent agents in electronic shopping environments, electronic government, the personalization of Web page designs, Web information seeking, and knowledge management. Dr. Detlor teaches courses in information retrieval, intelligent agents, and eBusiness at both the undergraduate and graduate levels. He is currently writing a monograph entitled *Knowledge Portals and the Digital Worker: From Human Issues to Intelligent Agents* to be published by Kluwer Academic Publishers. Author's present address: Michael G. DeGroote School of Business, McMaster University, Hamilton, Ontario, Canada. E-mail: detlorb@mcmaster.ca