Abstract
Individuals’ online activities are getting more and more complex with the growing opportunities over the Internet environment. Greater portions of daily activities such as shopping, socializing, and working are being transferred to the Internet environment. Over the past few decades, a new kind of software application has appeared based on a synthesis of ideas from artificial intelligence, human-computer interaction, and electronic transactions and people started to use software agents in their Internet applications. Autonomous software agents are one of these software agents and E-Commerce applications provide a popular application area for them. This paper focuses on the Autonomous Agent Based E-Commerce Systems (AABECS) and develops a metaphor for them to explain this new technology to various stakeholders. Metaphors are powerful instruments that can be used to explain new complex technological concepts to masses. To illustrate, the desktop metaphor was developed for operating systems and contributed to the popularity of personal computers. Successful metaphors help marketers to promote new technologies effectively and increase the product acceptance among consumers. The re-examination of the AABECS from the metaphoric perspective also triggers the creativity of the designers and increases the product quality.

Introduction
A new kind of software application has appeared based on a synthesis of ideas from artificial intelligence, human-computer interaction, and electronic transactions: software agents. In today’s language of computer interface, agents are used for a system that can serve as a go-between, because of possessing some specialized skill (Minsky, 1994). They can help mediate Electronic Commerce (E-Commerce) activities and are highly adaptive and personalized. There are other definitions of Software Agents that mention different characteristics of them as stated below:

“Software Agents are software components that communicate with their peers by exchanging messages in an expressive agent communication language” (Genesereth and Ketchpel, 1994).

“Software Agents are semi-intelligent computer programs that can employ Artificial Intelligence techniques to provide active assistance to a user with computer-based tasks” (Maes, 1994).

Capabilities of Software Agents are various. Without unnecessary or undesired human intervention, they can set up schedules, make reservations, arrange transportation, buy personal interest objects such as books, CDs, select gifts, etc. Briefly, by depending on the discussion above, the expression “software agent” is a computer program that has expertise areas, in order to help users on specific tasks.

Agent Based E-Commerce Systems
As business opportunities grow over the Internet, commercial actions take place in cyberspace increasingly. E-Commerce provides the environment for these Internet-based commercial actions. The main reason for the growth of this area is the great effort and time required by the “traditional commerce”, which is the “real world commerce.” The number of consumers who prefer shopping in the E-Commerce environment rather than in the real world increases dramatically every day.

In the real world, basic commerce actions can be classified as “consumer actions”, and “retailer actions.” Main consumer actions include deciding a product to purchase, searching the related retailers according to the price, quality, and included services of the product, selecting the product, and purchasing. Main retailer actions include determining a product scale and services, introducing this scale in a presentable area, advertising in order to have consumer interest, selling the products.

In addition to these basic classifications, consumers may want to sell some goods to other consumers by means of classified ads, or with a kind of personal advertisement such as yard sale signs, or telling people about it. Retailers may also communicate with each other in order to form
partnerships, make contracts, sell their products to each other, etc. Additionally, in the real world, there are large companies whose actions are much more complex than many retailers. All of these commercial actions include negotiations on price, quality and services as well. Negotiation is a form of decision – making where two or more parties jointly search a space of possible solutions with the goal of reaching a consensus (Rosenschein and Zlotkin, 1994).

Software Agents are useful tools to help individuals on their actions in E-Commerce environment. Both consumers and retailers can create Software Agents to communicate with each other in certain ways. Main communications that currently exist in E-Commerce can be classified as “business to consumer”, consumer to consumer”, and “consumer to business” communications.

Software agents promise especially high value by acting as mediators for the processes of selecting products and merchants (product brokering, merchant brokering), and for negotiations. By supporting or even automating activities such as information filtering and retrieval, evaluations, complex coordination tasks, and time-based interactions Software Agents have great potential to support the earlier stages of a purchasing operation (requisitioning to contract signing). So far, Software Agents have mainly been developed in the area of “business-to-consumer” E-Commerce.

**Product brokering** is the act where a buyer (consumer or corporate buyer) determines what to buy. Software Agents can help retrieve and evaluate product information and as a result reduce search costs and improve decision quality. **Merchant brokering** is the act where a buyer determines from whom to buy. Software Agents can help evaluate and compare the offers from different merchants by price or other features and again improve decision quality and reduce search costs.

**Developing a Metaphor for E-Commerce Systems**

“Metaphor” comes from the Greek word *metapherein* that means, “to transfer.” There is a sufficient distance between the metaphor and the ‘thing’ itself, but it still shares as many aspects of ‘the thing’ as possible. A ‘rich’ Metaphor can often lead to an intuitive grasping of what is being discussed without once mentioning the name of the thing. The definition of Metaphor, in rhetoric, is that it is a figure of speech in which one class of things (the target) is referred to as if it is belonged to the source (The Columbia Encyclopedia, 1993).

Personal Computer’s Desktop Metaphor is a well-known example of Metaphors used in Computer Systems. Software designers use Metaphors frequently when they design user-interfaces for their software. Although the Metaphors are widely used in design processes in software industry, there is not enough systemic methodology regarding the use of Metaphors in design projects. Choosing correct Metaphors and the successful usage of them can improve the design process.

Generating Metaphors for computer systems is very difficult. Since the computer systems and services have new and very complex contexts, it is hard to find a similar context in physical life. Autonomous Agent Based E-Commerce (AAEC) is a difficult subject for suggesting Metaphors. The common Metaphors for Software Agents are “social insects”, and “personal assistant”. In social insect metaphor, the insects are the population of information gathering organisms. Personal Assistant Metaphor, the insects are the population of large groups of agents who have no individual intelligence and autonomy. Personal assistant Metaphor enhances assistance function but lacks the autonomy.

Our Metaphor suggestion for Autonomous Agent Based E-Commerce environment is “Cloning Metaphor.” Cloning is the asexual reproduction of organisms from one ancestor. When we use a software agent in buying, selling or negotiating process, what we really desire is to multiply our own identity in the Internet environment. We want a software agent to mimic our decision pattern. This resembles the cloning process in genetic engineering. In Cloning Metaphor, Software Agents are clones of a user. As in genetic engineering, genetic materials of clones can be changed intentionally so they do not need to be the same with their ancestor’s. Since we want our Software Agents to know some subjects more than us and have some extra skills, their genes are modified for special purposes. For example a clone can be an expert in finding the appropriate used car and the other clone can be an expert negotiator. However, they still inherit our original identity and decision pattern.

Madsen (1994) provides a framework of guidelines and evaluation criteria for the development of metaphors. Evaluation of the cloning metaphor is realized by using his framework as explained below:

**Richness of the Structure of the Metaphor.**

Madsen recommends choosing a Metaphor with a rich structure. Cloning is originated from genetic engineering discipline. Its process has a rich context that includes chemistry, physics, medicine, pharmacology, sociology, justice and moral sciences. Cloning process also has a lot of sub - groups and
components, which can be used to model the AAEC environment.

**Applicability of the Metaphor.** The Cloning Metaphor covers the relevant aspects of the problem. Autonomous Agents in E-Commerce environment are designed to decide on buying, selling and negotiating processes without necessitating the user’s approvals. The users want those agents’ decisions represent their original expectations. The clones meet these expectations effectively since they carry their ancestors’ inheritance in their decision patterns.

**Suitability for the Audience.** Cloning is a popular subject in media. A lot of related news is being covered; numerous science fiction novels, films and literature are being produced. There is a general understanding about this subject throughout the world. Therefore, the Cloning Metaphor is suitable for the most of the intended audience.

**Conceptual Distance between the Source and the Metaphor.** Madsen proposes that “the power of metaphorical design is to help people see things in a new way, which is obviously easier if they are seen as something fairly different”. Therefore it is desired to have a Metaphor which has a distance from the source. In our cloning Metaphor, it refers to totally different discipline (genetic engineering).

**Bridging Concept.** In Metaphorical design, at least one concept that builds a bridge between the target domain and the source domain is needed (bridging concept). In suggested Metaphor, bridging concept is between the software agent and the clone. Since the audience has an idea about what a clone is, they will automatically use their knowledge and perspective in understanding the Software Agents.

**Looking for new meanings for the concept.** Metaphors suggest new meanings for existing concepts. In our Cloning Metaphor, an Autonomous Software Agent is an organism that can evolve, be mutated, and has ability to produce off springs. In the Cloning Metaphor, organic structure is introduced for the Software Agents. Software Agents become more human-like in this suggested metaphoric perspective.

**Remodeling the system.** According to Madsen, the new relations formed by the Metaphor have to be regrouped, reordered and renamed. A new system model can be built based on the developed Metaphor. In this step, we talk about target domain as if it were the source domain. In this phase assumptions have to be elaborated. It is necessary to make explicit what the suggested Metaphor hides and what it highlights. Unused parts of the Metaphor must be identified. Aspects, features and properties of the source domain have to be discussed to better understand the target domain. Understanding the source domain helps the understanding of the target domain.

In the Cloning Metaphor, ancestors’ genes represent the user’s identity and personal preferences. In order to transfer this information pattern to the clones, the ancestors’ genes must be isolated and decoded. This process corresponds the learning phase of the Autonomous Software. In this step, the user’s personal preferences and behavioral model are coded into the clone’s genes. Once this step completed, extra genes inserted in the clone’s gene pattern in order to add extra expertise knowledge to clone’s behavioral model. These inserted genes consist of ready-to-use expert information, which can be used for desired E-Commerce practices.

When these two steps are completed, the creation of the “baby clone” is achieved. The following step involves the testing the baby clone in simulated problem context. The clones that pass this testing phase become “adult clones”, which are ready-to-be used in the real E-Commerce applications. Performance evaluations of the clones are done continuously. The Ancestor sets indices of performance. The clones, which fail in the evaluations, are taken into gene therapy treatment in order to have their gene pattern modified. In genetic engineering, “Gene Therapy” is defined as the process of introducing new genes into the DNA of a person’s cells to correct a genetic disease or flaw. The clones, which are too sick and with the high cost of treatment can be considered as “dead clones” and isolated from the systems. The healthy clones reproduce their clones with breeding with other successful clones. For specific E-Commerce tasks, clone populations can be formed, and simulated evolution can be applied to determine the best decisions. “Mutations” are used to add variety and flexibility to the agent populations. Mutations are defined as heritable change in DNA sequence.

System model is based on the “inheritance concept” that represents the reflection of the user’s original behavioral model in the final decision pattern. The suggested model introduces new concepts such as mutations, gene therapy and reproduction of clones. These concepts help to maintain the flexibility and the adaptability of the agents.

**Conclusions**

In this paper, a metaphor has been developed for Agent Based E-Commerce. The proposed “Cloning Metaphor” is evaluated by following Madsen’s
guidelines. As a result of this evaluation, it has been
demonstrated that the selected metaphor presents an
elaborate concept and useful for the designers of such
systems.

References
Genesereth, M; Ketchpel, S., “Software Agents”,
37, No. 7.
Maes, P., “Agents that Reduce Work and
Information”, Communication of ACM, Vol. 37,
Madsen, K., “Metaphorical Design”,
37, No. 12.
Minsky, Marvin, “A Conversation with Marvin
Minsky About Agents”, Communications of the
Rosenschein, J; Zlotkin, G., “Rules of Encounter:
Designing Conventions for Automated
Negotiation among Computers”, MIT Press,
1994.

About the Authors
Erol Ozan received his Ph.D. in Engineering
Management & Systems Engineering from Old
Dominion University, his MS in Applied Physics
from University of Istanbul and his BS in Electrical
and Electronics Engineering from Middle East
Technical University (Turkey). He is currently an
assistant professor in the Department of Industrial
Technology at East Carolina University. His research
interests include computer visualization, technology
management, and decision support systems.

Yesim Sireli received her Ph.D. from the Department
of Engineering Management & Systems Engineering
at Old Dominion University, and M.S. and B.S.
degrees in Electrical Engineering from Istanbul
Technical University, Istanbul, Turkey. She is
currently an assistant professor at the Engineering
Management Program at the University of North
Carolina at Charlotte. Her research interests include
technology management, decision models, project
management and product development.

Paul Kauffmann received his Ph.D. from Penn State
in Industrial Engineering. He holds a MENG in
Mechanical Engineering and a B.S. degree in
Electrical Engineering from Virginia Tech. He was
formerly Department Chair of Engineering
Technology at Old Dominion University and is
currently Department Chair of Industrial Technology
at East Carolina University. His research interests
include technology management and managerial
decision methods. His industrial career with Philip
Morris USA included positions as project engineer,
plant manager, and engineering director.