A conceptual framework to explain technology acceptance of electronic negotiation utilizing software agents

by

Vadivananthan Visuvalingam

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Management Sciences

Waterloo. Ontario, Canada, 2006

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Electronic negotiation (e-negotiation) is a relatively new technology that has spawned as a result of the growth of electronic commerce (e-commerce). While researchers have dealt with a variety of topics in the area of e-negotiation the acceptance of e-negotiation technology is a subject that needs further exploration. This study contributes to the research in e-negotiation, by putting forward a conceptual model that explains the factors that affect technology acceptance of e-negotiation.

We survey past works in the technology acceptance literature, and review three seminal theories – the technology acceptance model (TAM), the theory of planned behaviour (TPB) and the diffusion of innovation theory (DOI). We develop a conceptual model by identifying various factors and interrelationships amongst them that are valid for the context of our study. To test our model, we develop a web interface for participants to experience e-negotiation, and incorporate a survey instrument, adapted from previous studies, to assess participants' attitudes and perceptions towards using e-negotiation. We also test whether the presence of learning agents has an effect on perceptions of negotiation outcomes.

Regression and MANOVA tests indicate that attitude and associated attitudinal beliefs have a significant influence on acceptance of e-negotiation technology. We also find that perceptions of negotiation outcomes affect e-negotiation acceptance; however, learning agents were not found to have an influence on perceptions of negotiation outcomes.

Acknowledgements

I would like to thank Dr. R. P. Sundarraj, who was my supervisor for this thesis, for his advice and guidance in my research. I am also very grateful for his time and effort to in helping me put together my thesis. I really appreciate his words of encouragement whenever I faced obstacles in my research.

I would also like to thank Dr. S. Jeffrey and Dr. J. Goyder for spending the time to proof read my thesis and providing feedback.

I would also like to thank my family and friends for their support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Motivation

Over the years, the Internet grew and connected millions of potential customers and sellers. Along with this growth, developments in web technologies fuelled the growth of electronic commerce (e-commerce) (Wang, Tan, & Ren, 2005). The expansion of e-commerce is having an effect on the way goods and services are priced. According to Bichler et al. (2002), there has been a shift away from fixed pricing to flexible pricing, in which a seller may ask for a higher price for a brand new product and ask for a lower price for a refurbished product, thereby allowing the seller to cater to a larger segment of the customer base. Amongst the variety of mechanisms that facilitate flexible pricing, one mechanism that is becoming prominent is electronic negotiation (e-negotiation) (Choi, Liu, & Chan, 2001).

A variety of different types of systems have been developed for facilitating e-negotiation. Researchers have designed frameworks (Schoop, Jertila, & Thomas, 2003; Yuan, Rose, & Archer, 1998), which formalized negotiation processes and procedures. The presence of standard processes and procedures makes possible the use of technology to conduct e-negotiation. One genre of e-negotiation systems facilitates online negotiation between humans, by providing environments to post messages/offers securely, as well as by a variety of support tools to facilitate decision-making and process-oriented activities pertaining to negotiation. Advanced negotiation systems such as e-negotiation tables and electronic markets provide integrated communication systems and provide process support to facilitate online negotiation (Kersten, 2002). In order to

create such systems, researchers have explored the development of negotiation support systems (Schoop & Quix, 2001; Strobel, 2002). In addition to helping human-to-human negotiation, researchers have also sought to automate some of the negotiation processes through the use of software agents – continuously running programs that can understand their owners' requirements and perform tasks such as negotiations on behalf of their owners (Choi et al., 2001). Researchers have also explored the development of negotiation models (Deveaux, Paraschiv, & Latourrette, 2001; Sebenius, 1992; Wasfy & Yasser, 1998) for agents to use in their negotiations. While a variety of research areas have been explored in e-negotiation, one research area that needs further attention is the exploration of factors that would influence the acceptance of e-negotiation technology.

In this study, we postulate some factors that affect the acceptance of e-negotiation technology and administer a survey to test the factors that would hold empirically. Identification of these factors will contribute valuable information to the research in e-negotiation. By understanding what would influence potential adopters to embrace e-negotiation, researchers can gain insight into information that could guide them to develop and improve e-negotiation technology. As well, this information can also help develop guidelines for potential ways to commercialize e-negotiation. Among the factors that we will be looking at, one factor of particular interest for the study is the presence of learning agents and its effect on perceptions of negotiation outcomes, which in turn influence technology acceptance. Since learning has been noted to improve negotiation outcomes, (Deveaux et al., 2001; Mok & Sundarraj, 2005), it would be interesting to see if this additional rich feature can contribute to increase e-negotiation acceptance.

To identify our factors, we draw upon three dominant theories that have been widely used in the technology acceptance literature. These theories are: the technology acceptance model (TAM) (Davis, 1989; Davis, Bagozzi, & Warshaw, 1989), the theory of planned behaviour (TPB) (Ajzen, 1985; Ajzen & Madden, 1986) and the diffusion of innovation theory (DOI) (Rogers, 1983). TAM is an adaptation of Fishbien and Ajzen's (1975) theory of reasoned action (Ajzen & Fishbein, 1980), while TPB is an extension of the theory of reasoned action. DOI was based on characteristics identified from a survey of several studies on innovation diffusion and adoption (Rogers, 1983). These three theories have been used to explain the acceptance of a variety of commercial technologies used in both office and educational environments. Thus, our research will be grounded in these theories, although we will adapt them to suit our particular context. As indicated earlier, e-negotiation is a very infant technology and thus has not been identified for usage in any particular environment. Therefore, it represents a unique context that has not been considered in previous studies on technology acceptance and thus requires a new model that is suitable to its particular situation.

The specific research issues covered in this thesis include:

1. A conceptual model to explain the factors that affect acceptance of e-negotiation technology.
2. Empirical verification of the reliability and validity of the factors, and of the interrelationships among them.

3. Testing the effect of learning agents on perceptions of negotiation outcomes, and in turn, the influence on e-negotiation acceptance.

1.2 Overview

In this study, we first review the various factors considered by the technology acceptance model (TAM), theory of planned behaviour (TPB) and diffusion of innovation (DOI) theory. We then develop a theoretical model by taking into consideration the factors applicable to the e-negotiation context. The hypotheses proposed in the study indicate the nature of the relationships amongst the factors. The hypotheses also postulate how the presence of learning agents would influence perceptions of negotiation outcomes, which then affects e-negotiation acceptance. This contributes to the research area of user-acceptance, in the sense that we put forth a model for a technology that is not commercially available, and as a result, has no associated environment of usage.

To test our theory, we developed a web interface, based on learning algorithms put forth by Mok and Sundarraj (2005). The web interface incorporated important elements of e-negotiation, so that the study participant can experience the technology and form attitudes. Drawing from questionnaire items used in other studies, we develop a survey instrument, and conduct pretests to ensure that the instrument is comprehensible and does not cause excessive mental exertion for the participants. Then, the survey instrument is tested for reliability and validity through the use of factor analysis.

We then conduct regression analysis to test the theoretical model and a MANOVA to assess how the presence of learning agents affects perceptions of negotiation outcomes. Following this, we summarize our findings and make some conclusions. We also discuss the implications of our findings and point to some future research that can be conducted based on this study.

In chapter 2, we present the literature review on the research areas of e-negotiation and technology acceptance. Chapter 3 presents an examination of the various theories in technology acceptance and development the theoretical model and hypotheses. In chapter 4, we present the development of the e-negotiation web-interface, the research methodology used in the study and the survey instrument's reliability and validity testing results. The data analysis of the survey results is discussed in chapter 5. A summary of our findings and our conclusions can be found in chapter 6. The implications of this study and future research that can be conducted are presented in Chapter 7. Appendices A and B describes the web interface in detail and provides the software code used in the system. Appendix C presents the slides from the Powerpoint presentation that provides detailed steps to using the e-negotiation system. Appendix D presents the various communications, such as posters and letters that were used to promote the survey and to recruit participants.

Chapter 2. Literature review

The previous chapter introduced and motivated our study. In this chapter, we will discuss relevant work in the area of e-negotiation and illustrate the need to identify the factors that influence technology acceptance of e-negotiation. The area of technology acceptance will also be explored to show past work in this area and to illustrate the need for a new model that would take into account the unique context of e-negotiation. We will then discuss how our study makes a contribution to the two research areas of e-negotiation and technology acceptance.

2.1 E-negotiation

E-negotiation has emerged as a broad interdisciplinary area drawing from such disciplines as computer science, economics and psychology. Our survey of literature in the area of e-negotiation showed that research has explored a variety of topics covering such areas as design of frameworks for e-negotiation (Schoop & Quix, 2001; Strobel, 2002), development of negotiation support systems (Schoop & Quix, 2001; Schoop et al., 2003; Yuan et al., 1998) and development of negotiation models for agents (Choi et al., 2001; Deveaux et al., 2001; Faratin, Sierra, & Jennings, 1998; Mok & Sundarraj, 2005; Sebenius, 1992; Wasfy & Yasser, 1998). Frameworks provide a conceptual map for how systems can engage in negotiations, while negotiation support systems provide tools used by human negotiators to help improve their outcomes in negotiations. Thus, research in both areas is necessary for the development of systems that will enable humans to engage in e-negotiation. Software agents can perform tasks on behalf of their human principals and thus automate the negotiation process. However, development of negotiation models

is necessary, as software agents are not naturally endowed with the intelligence to engage in negotiations and thus use negotiation models to guide their negotiation behaviour. In this section, a brief coverage of these various areas is provided to give the reader some background in e-negotiation. As well, we will illustrate the need to identify the factors that influence technology acceptance of e-negotiation.

2.1.1 Frameworks for e-negotiation

Frameworks provide protocols and procedures for negotiation parties to communicate with each other about negotiation issues and offers. They also provide mechanisms for the parties to accept or decline offers put forward by the other party. Parties engaged in e-negotiation can be composed of either humans or software agents (Huhns & Singh, 1998; Wooldridge & Jennings, 1995), which are autonomous software programs that act on behalf of humans. For e-negotiation involving agents, the main goal in design of frameworks is to ensure clear exchange of messages, i.e. no ontology issues[1], such that there is not any ambiguity in the communication between agents (Schoop & Quix, 2001). An example is the SILKROAD framework[2] (Strobel, 2002), where the communication design is represented in terms of XML Schema enabling clear exchange of messages between the agents involved in the negotiation.

[1] Ontology issues arising in e-negotiation involving agents involves the representation of negotiation issues as abstract objects in the negotiation process and how those abstract objects are interpreted by the respective parties (Strobel, 2002).

[2] The SILKROAD framework is a negotiation framework, developed by Strobel (2002) for e-negotiation involving agents, that provides procedures and protocols for agents to use while they engage in e-negotiation.

In systems that facilitate e-negotiation between human parties, the systems provide tools to support human negotiation instead of engaging in automated negotiations (Schoop & Quix, 2001). Like the goal for frameworks for negotiation involving agents, the main focus in the design of the frameworks for negotiation involving humans is to provide for clear exchange of messages. An example is the DOC.COM framework (Schoop & Quix, 2001), where structured message exchange and contract management are provided to ensure clear communication of issues and offers in the negotiation between two human parties.

Frameworks are conceptual designs of how e-negotiations should be conducted and thus are not dependent of the technology used in e-negotiation. However, they are important in the overall development of e-negotiation systems as they enable the design of such systems to facilitate the negotiation between various parties. Since negotiation frameworks are conceptual designs and not the actual technology used in e-negotiation, it is not possible to assess the acceptance of negotiation frameworks on their own. However, their technology acceptance can be assessed in concert with the various technologies developed for e-negotiation. In the next two sub-sections of this section, we will discuss the research with regards to the development of negotiation support systems and the development of models for agents.

2.1.2 Negotiation support systems

Negotiation support systems are software systems that provide tools for use by human negotiators engaged in negotiations. There are two types of approaches to the

development of negotiation support systems: communication-oriented approach and document oriented approach (Schoop et al., 2003). The communication-oriented approach is aimed at developing negotiation support systems that facilitate the organizational communication processes. An example is the WebNS negotiation support system, which provides different windows for various types of communication such as negotiation discussion, informal discussion and personal notes (Yuan et al., 1998). The document-oriented approach is aimed at developing negotiation support systems that facilitate the exchange of documents and document storage such as business contract storage; for example the SmartSettle negotiation support system provides an assortment of business forms to support human negotiators (Schoop et al., 2003). Some systems such as Negoisst (Schoop et al., 2003), based on the DOC.COM framework (Schoop & Quix, 2001), combine both approaches and provide support for both organizational communication processes as well as the exchange of documents and document storage.

Some research work has been done in studying the factors that influence the adoption of negotiation support systems (Koeszegi, Vetschera, & Kersten, 2004). Koeszegi et al. (2004) classified culture, based on differences in communication patterns, as two particular types: high context culture, where a lot of the implicit information is contained in the context of an event or internalized by individuals, and low context culture, where very little information is contained in the context of the event. Koeszegi et al. (2004) studied the influence of culture on technology acceptance of the Inspire negotiation support system. More specifically they looked at the influence of high context culture like Japanese culture versus the influence of low context culture like

German culture. They found that high context culture has a strong influence on users' actual use of communication features while low context culture has a strong influence on users' perceived usefulness of analytical features. Overall they found negotiation support systems did gain technology acceptance. However, there is further need to explore acceptance of e-negotiation technology, specifically agents for e-negotiation, since it is still a very under-explored area. In the next sub-section we will discuss the research with regards to the development of negotiation models for agents.

2.1.3 Negotiation models

Negotiation models for autonomous agents provide agents with potential tactics that they can use during negotiations. A variety of approaches have been taken to develop models for autonomous agents, including but not limited to genetic algorithms (Choi et al., 2001), game theoretic algorithms (Sebenius, 1992) and behavioural models (Deveaux et al., 2001; Faratin et al., 1998; Mok & Sundarraj, 2005). All these varied models bring a unique approach to negotiation but also contain some disadvantages.

In genetic algorithms, the agents start with a population of tactics that is used to calculate an offer. Through each round of negotiation the agents gradually filter the tactics through predefined evaluation criteria to create a new population or generation of tactics (Choi et al., 2001). This particular approach continues until the agent is able to reach an optimal solution. A major disadvantage with genetic algorithms is the amount of time or generations it would take to achieve a good set of tactics (Beam & Segev, 1997). This may lead to the agents utilizing genetic algorithms taking a long time to

reach an outcome, which would not be very desirable to the agents' respective human parties.

In game theoretic algorithms, the vague nature of negotiations is transformed into a precise game between rational players or agents. Analysis of the all the potential interactions between the agents can predict equilibrium outcomes to negotiations (Sebenius, 1992). Pursuit of an equilibrium producing strategy by an agent will lead to a situation such that the other agent will not change its tactic, thereby ensuring an agreement is reached. A major disadvantage with game theoretic algorithms is that there is an assumption that all agents are perfectly rational, which may not be the case in all e-negotiation situations (Deveaux et al., 2001). For example, there can be an electronic marketplace with agents that have different models that they utilize to negotiate. In such a case, agents that do not utilize game theoretic algorithms can behave irrational which will place agents utilizing the game theoretic algorithms at a disadvantage as they will not adapt and achieve a favorable outcome for their respective human party.

In behavioural models, the agents imitate particular aspects of human behaviour while removing the disadvantage of slowness associated with human negotiation (Mok & Sundarraj, 2005). Agents that utilize behavioural models do not require information about the other agents to engage in negotiation. This overcomes the disadvantage faced by agents utilizing game theoretic models, namely the need for the other agent to act with perfect rationality. Behavioural agents employ a variety of tactics (Faratin et al., 1998) in negotiation and amongst these various tactics the time dependent tactic overcomes the

disadvantage faced by agents utilizing genetic algorithms. In the time dependent tactic the primary goal of the agents is to complete the negotiation within a given time frame. Thus behavioural agents that utilize the time dependent tactic overcome the two disadvantages that were identified with genetic and game theoretic algorithms.

Agents that are adaptive, such as those that utilize genetic algorithms, have an advantage over non-adaptive agents since they can update their internal beliefs and adjust their tactics to achieve a more favorable outcome. However, due to the limitations noted about genetic algorithms, some research work has been done to develop adaptive algorithms for behavioural agents that utilize the time dependent tactic (Mok & Sundarraj, 2005). Mok and Sundarraj (2005) developed adaptive algorithms, in which an agent can possess the capacity to "learn" about another agent by examining the various bids put forth by the other agent. Once the agent has "learned", it can adjust its own parameters to the benefit of its respective party. By incorporating adaptive algorithms into behavioural agents that utilize the time dependent tactic, a richness of features is added to a negotiation model that overcomes limitations in other negotiation models. Thus, the presence of learning agents in e-negotiation provides some promising potential for the user-acceptance of e-negotiation.

2.1.4 Importance of studying factors that influence technology acceptance

One relatively unexplored research pertains to the identification of factors that influence the technology acceptance of e-negotiation. As well of interest for study is how the presence of learning agents affects these factors. Since learning agents overcome

disadvantages held by other models for agents they provide a relative advantage over other comparable technologies and can therefore possible contribute to increasing the acceptance of e-negotiation. Technology acceptance is an important area to research as it provides valuable insight to researchers developing e-negotiation technologies.

In order to explain the technology acceptance of e-negotiations, we need to identify the factors that would influence technology acceptance and develop a model that takes into account the unique context of e-negotiation. In the next section of the chapter, we will explore the various models from other studies that explain technology acceptance and proceed to examine the contextual differences between this study and previous studies.

2.2 Technology Acceptance

In our review of the technology acceptance literature, three theories have emerged that have been used to explain user acceptance of various technologies. These three theories are TAM (Davis et al., 1989; Davis, 1989), TPB (Ajzen, 1985; Ajzen & Madden, 1986) and DOI (Rogers, 1983). We give a brief coverage of these three theories and their application to explain acceptance of various technologies. In chapter 3, we will discuss in further detail the constructs and the interrelationships introduced by these theories prior to developing our model.

2.2.1 Technology acceptance model

TAM is an adaptation of Fishbien and Ajzen's (1975) theory of reasoned action (Ajzen & Fishbein, 1980), a well-established social psychological model (Davis et al., 1989). TAM placed emphasis on the two beliefs of perceived usefulness and perceived ease of use and their effect on behavioural intention. TAM has been tested to explain or predict behavioural intention on a variety of technologies such as word processors (Adams, Nelson, & Todd, 1992; Davis et al., 1989), spreadsheet software (Adams et al., 1992), email (Adams et al., 1992; Szajna, 1996), voicemail (Adams et al., 1992), graphics software (Adams et al., 1992) and net conferencing software (Venkatesh, 1999). Thus the technology acceptance model has been shown to be valid over a variety of commercially available technologies that are primarily used in an office environment (Adams et al., 1992; Venkatesh, 1999) or educational environment (Adams et al., 1992; Davis et al., 1989; Szajna, 1996).

2.2.2 Theory of planned behaviour

TPB is an extension of the theory of reasoned action. It introduced a new dimension, namely perceived behavioural control and control beliefs as an additional antecedent to behavioural intention. TPB has been tested to explain or predict behavioural intention on a variety of technologies such as voice mail (Benham & Raymond, 1996), spreadsheet software (Mathieson, 1991) and telemedicine (Chau & Hu, 2001). Similar to TAM, TPB has been shown to be valid over a variety of commercially available technologies that are primarily used in an office environment (Chau & Hu, 2001) or educational environment (Benham & Raymond, 1996; Mathieson, 1991).

2.2.3 Diffusion of innovation

DOI was based on characteristics of diffusion of innovation identified by Rogers (1983). DOI is not as focused as TAM or TPB in explaining or predicting behavioural intention, but rather concerned with characteristics of diffusion of innovation. DOI has been used in conjunction with other theories, usually by having the various diffusion characteristics incorporated as antecedent constructs to behavioural intention, to study user acceptance of a variety of technologies such as voicemail (Benham & Raymond, 1996), personal workstations (Moore & Benbasat, 1991), smart card technology (Plouffe, Hulland, & Vandenbosch, 2001) and operating systems (Karahanna, Straub, & Chervany, 1999). The various characteristics have been shown to be valid antecedents to behavioural intention on a variety of commercially available technology that are primarily used in an office environment (Karahanna et al., 1999; Plouffe et al., 2001) or educational environment (Benham & Raymond, 1996; Moore & Benbasat, 1991).

2.2.4 Contextual differences between current study and past studies

One of the contextual differences between the current study and past studies is that much of the research work in the technology acceptance literature has often focused on commercially available technology. The three theories of TAM, TPB and DOI have been shown to be valid over a wide variety of technologies ranging from messaging technologies such as voicemail and email to online technologies such as net conferencing software. However, e-negotiation technology utilizing agents is still very much at an infancy stage in its development and not commercially available. Thus, the model that is

developed in this study will take this context into account; for example, the technology will be very new and thus such things as pre-formed opinions or views etc. will not exist for such a technology.

Another contextual difference is that past studies have focused on office or educational technologies, which have a certain mandatory requirement (e.g. a directive from management or course instructors encouraging the use of a particular type of technology). However, there are no mandatory aspects with regards to agents in e-negotiation since it is still very much at an infancy stage in its development. Thus, there has not been any organizational level recognition of this technology to elicit mandatory requirements concerning the use of this technology. In the next section of this chapter we will cover how our study makes a contribution to the research areas of e-negotiation and technology acceptance.

2.3 Contribution of this thesis

Through this study, we will contribute to the e-negotiation literature by identifying the factors that would influence e-negotiation adoption and assess how the presence of learning agents has an effect on perceptions of negotiation outcomes. In the literature we have surveyed, this type of research has not been carried out extensively, particularly whether the presence of the learning feature in agents will have any influence on the adoption of e-negotiation technology. By studying these factors, we hope to provide valuable insight to researchers for developing agents with a richer set of features that can improve e-negotiation technologies.

Relatively new technologies differ from mature ones in sense that they typically are not commercially available and that they lack any requirement of mandatory usage. Our contribution to the technology acceptance literature lies in putting forward a model that explains the adoption of a relatively new technology. This work is also important as the theoretical framework may also be used in the future to study acceptance of other new technologies under development.

In summary, our contribution through this study is primarily through two aspects. First, we put forward a conceptual model that explains acceptance of e-negotiation utilizing agents, which is an entirely new context versus the contexts of technologies assessed in past studies. Second, we examine whether of the presence of learning agents will contribute to the acceptance of e-negotiation technology.

Chapter 3. Theory and Hypotheses

Thus far, we identified the relevant research work carried out in e-negotiation and technology-acceptance areas. We also discussed the need to identify the factors that influence technology acceptance of e-negotiation. Our discussion also highlighted the contextual differences between the current study and previous ones on technology acceptance. In this chapter, the theories in technology acceptance literature and their constructs will be discussed in greater detail. In particular, from these theories, we will develop and present a model that identifies the factors influencing e-negotiation acceptance and how the presence of learning agents affects one of these factors pertaining to perceptions of negotiation outcomes.

3.1 Major Theories

TAM is one of the most widely and empirically tested theories (Chau & Hu, 2001; Davis, 1989; Moore & Benbasat, 1991; Segars & Grover, 1993) that drew on social psychological approach to explain adoption of technology and the factors that influenced individuals. Using the well-established theory of reasoned action as a basis, TAM explained the factors that would influence individuals to adopt technology. TPB is an extension of the theory of reasoned action and has been used as an alternative competing model and compared to TAM. DOI was developed from the survey of several hundred studies on technology diffusion. It identified the characteristics about a technology that would impact its diffusion, and in turn, its acceptance. By reviewing TAM, TPB and DOI, we will identify the components that will be relevant to the theoretical framework that is developed in this thesis.

3.1.1 Technology Acceptance Model

Davis et al. (1989) proposed TAM as a way to explain and predict technology acceptance of an information system by its end users. TAM is an adaptation of Fishbein and Ajzen's (1975) theory of reasoned action (Ajzen & Fishbein, 1980), which had "proven successful in predicting and explaining behaviour across a wide variety of domains" (Davis et al., 1989, p. 983).

As shown in Figure 1, TAM proposes six constructs (Davis et al., 1989): actual system use, behavioural intention to use, attitude toward using, perceived usefulness, perceived ease of use and external characteristics. The relationship between attitude toward using, behavioural intention to use and actual system use were derived from the theory of reasoned action (Davis et al., 1989). The other technology acceptance model constructs and their relationships were new ones proposed by Davis et al (1989) for explaining the beliefs that affect the attitude towards using technology and how external characteristics affect these beliefs.

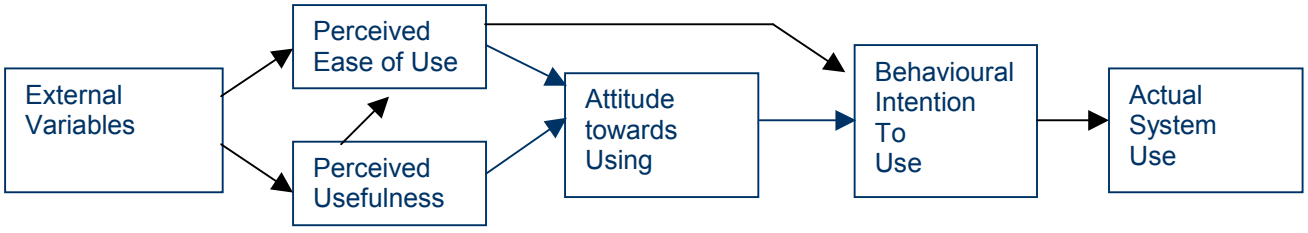


Figure 1 The technology acceptance model.

Two constructs, namely external characteristics and actual system use, were introduced to encapsulate observable components of technology adoption. External characteristics refer to all the external features of a system ranging from menus, icons to output produced by the system (Davis et al., 1989). Actual system use refers to the potential adopter's system usage behaviour. TAM explains how the external characteristics of the system affect the potential adopter's attitudes and perceptions leading to actual use of the system. The direct effect of behavioural intention on actual system usage is adapted from the theory of reasoned action. Similarly, the positive direct effect of attitude on behavioural intention is also adapted from the theory of reasoned action.

The two behavioural beliefs[3] introduced by TAM consisting of perceived ease of use and perceived usefulness was a new contribution to research in technology acceptance. Perceived ease of use is defined as "the degree to which a person believes that using a particular system would be free of effort" (Davis, 1989, p. 320). The complexity of the external characteristics of the system has a direct effect on perceived ease of use. Perceived ease of use is considered to have a positive direct effect on attitude; for example, if an individual views that using a system is fairly free of effort, their affect with regards to using the system will increase positively. Perceived usefulness is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance" (Davis, 1989, p. 320). A potential adopter's perceived usefulness is directly affected by the degree to which they perceived

[3] To some readers in other fields, the notion of behavioural beliefs may seem illogical, but this term is widely used in the Information Systems literature, particularly in discussion related to antecedents to attitude.

that the external characteristics of a system aided them in performing a task or a set of tasks. Equivalently, the ease of use of a system can also contribute to increased performance; thus, ease of use has a direct effect on perceived usefulness. Perceived usefulness is also considered to have a positive direct effect on behavioural intention; for example, if potential adopters believe that the system delivers useful outcomes, their intention to use is increased. Perceived usefulness is considered to have a positive direct effect on attitude towards using a system. When potential adopters observe that the system delivers positive outcomes this will positively increase their affect with regards to using the system.

3.1.2 Theory of Planned Behaviour

As shown in Figure 2, another major theory in technology acceptance literature is TPB (Ajzen, 1985; Ajzen & Madden, 1986). The theory of reasoned action had been developed under the assumption that a person has complete control over a behaviour; however, it became clear that situations exist where individuals lacked complete discretionary control to engage in a behaviour (Ajzen, 1991). Thus, as indicated in Figure 2, the theory of reasoned action was extended with an additional construct, namely perceived behavioural control, to account for such situations and the new theoretical model became TPB.

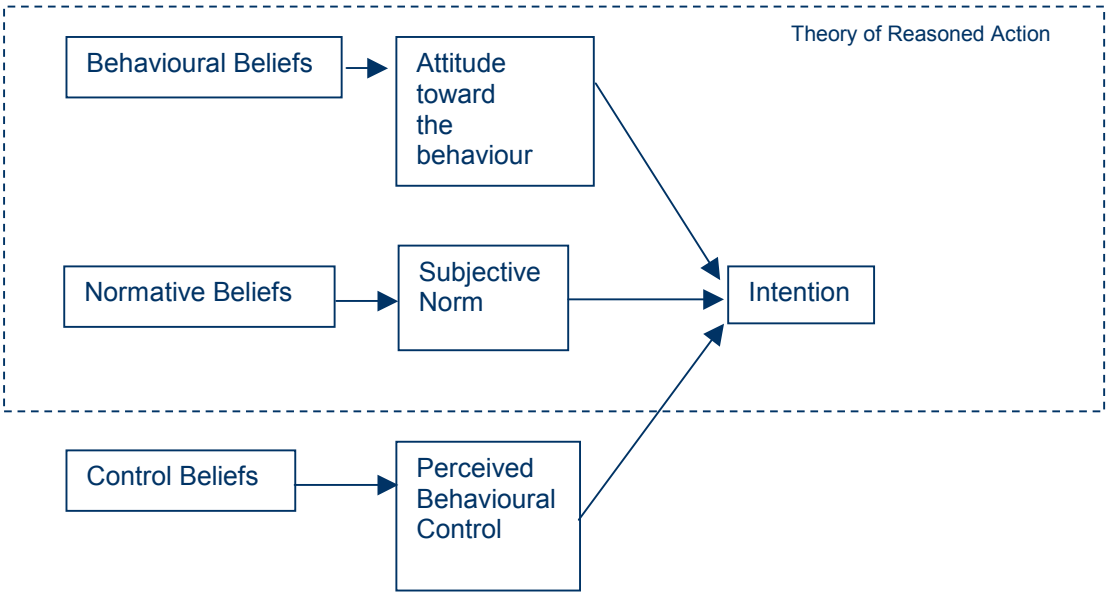


Figure 2 The theory of planned behaviour.

The general ability people feel that they have to perform certain behaviour is referred to as perceived behavioural control (Ajzen, 1985; Ajzen & Madden, 1986). The beliefs that individuals hold with regards to their ability to perform a behaviour, is referred to as control beliefs. These beliefs would include whether an individual believed they possessed the necessary skills, resources or opportunities to perform the behaviour.

A number of studies have explored using TPB to explain and predict technology acceptance (Benham & Raymond, 1996; Chau & Hu, 2001; Mathieson, 1991) and others have also compared it to TAM (Chau & Hu, 2001; Mathieson, 1991). Other studies such as Benham and Raymond (1996) proposed a research model that extended TPB by including antecedents drawn from DOI (Moore & Benbasat, 1991; Rogers, 1983).

3.1.3 Diffusion of Innovation Theory

Rogers (1983) developed DOI based on "a survey of several thousand innovation studies" (Moore & Benbasat, 1991, p. 193) and identified characteristics associated with innovation including: relative advantage, compatibility, complexity, observability and trialability. Moore and Benbasat (1991) developed an instrument, based on DOI, to measure technology adoption. Based on the empirical testing of the instrument, they introduced new DOI characteristics and modified some of the original DOI characteristics. The refined DOI characteristics include: voluntariness, image, relative advantage, compatibility, ease of use, trialability, results demonstrability and visibility. The DOI characteristic that refers to an individual's perception of the benefits of using an innovation is relative advantage.

Relative advantage refers to the degree to which an innovation is perceived to be better than earlier innovations of similar nature (Rogers, 1983). This relative advantage can be expressed in terms of economic factors, social status, or other types of benefits (Rogers, 1983). In terms of economic factors the relative advantage would occur when an innovation's overall cost were lower than earlier innovations of similar nature; i.e., a VCR (video cassette recorder) cost $1200 in 1980 whereas a few years later a similar VCR cost $200 (Rogers, 1983). In terms of social status, the relative advantage would occur when an innovation's conferred prestige is greater than earlier innovations of similar nature; i.e., the latest clothing fashions are more prestigious to wear than earlier clothing fashions (Rogers, 1983). Relative advantage is also considered equivalent to

TAM's perceived usefulness (Moore & Benbasat, 1991). Another DOI characteristic, namely compatibility, also refers to an individual's perception of the benefits of using an innovation.

Compatibility is defined as "the degree to which an innovation is perceived as being consistent with the existing values, needs, and past experiences of potential adopters" (Moore & Benbasat, 1991, p. 195). An innovation that is more compatible with previously introduced innovations and existing socio-cultural beliefs is less uncertain to potential adopters (Rogers, 1983). Innovations that offer a relative advantage but have not been compatible with existing needs and values have failed. For example, in the mid-1960s a variety of rice developed by the Rice Research Institute in the Philippines that tripled yields; however, it failed to be adopted by rice farmers since it did not taste the same as previous rice varieties (Rogers, 1983).

Moore and Benbasat (1991) introduced the construct of results demonstrability as a way to capture the degree to which the results of using an innovation are tangible and can be conveyed to others. An individual's perception of the benefits offered by the innovation is critically influenced by the tangible results that are seen by the potential adoptee (Rogers, 1983). Thus, Moore and Benbasat (1991) included this construct as an extension to Rogers' (1983) DOI.

Rogers (1983) also considered a dimension called complexity to capture an individual's perception of how difficult an innovation was to understand. Moore and

Benbasat (1991) renamed this dimension as ease of use to make it similar to the TAM's perceived ease of use as they felt that both constructs were highly similar to each other.

Other constructs considered by the Moore and Benbasat (1991) were image, visibility, trialability and voluntariness. Image is defined as "the degree to which an innovation is perceived to enhance one's image or status in one's social system" (Moore & Benbasat, 1991, p. 195). Visibility is defined as "the degree to which others can ""see"" that an innovation is being used" (Benham & Raymond, 1996, p. 6). Voluntariness is defined as "the degree to which use of the innovation is perceived as being voluntary, or of free will" (Moore & Benbasat, 1991, p. 195). Trialability is defined as "the degree to which an innovation may be experimented with before adoption" (Moore & Benbasat, 1991, p. 195).

3.2 Model Development

Mathieson (1991) compared TAM and TPB and concluded that TPB provides more useful information than TAM during the development stage of an information system. According to Mathieson (1991) while "TAM supplies very general information … TPB delivers more specific information … and identifies factors that respondents feel might be barriers to system use" (p. 187). Given this finding, we consider a TPB-based model to be more suitable than TAM for assessing acceptance of e-negotiation technology. As has been noted previously in the literature review, e-negotiation agents are currently at an infancy stage of development and are not a fully commercial product.

Mathieson (1991) also noted that, as compared to TAM, TPB is more difficult to apply across various contexts, since it needs to be tailored to each new context; for example the various behavioural beliefs, normative beliefs and control beliefs would need to be identified and defined in each new context. Benham and Raymond (1996) addressed this concern when they put forward a model by fusing TPB and DOI, to explain the adoption of voicemail. As shown in Figure 3, they adapted the DOI instrument developed by Moore and Benbasat (1991) and categorized the various perceived characteristics of innovation in DOI as behavioural beliefs, normative beliefs and control beliefs in TPB. They also introduced an additional control belief, support, which sought to capture the perceived adequacy of resources and opportunities pertinent to the use of the innovation (Benham & Raymond, 1996). As noted previously, DOI is a generic theory, based on characteristics observed in many different innovations. Utilization of agents in e-negotiation is an innovation as this is consistent with one of Rogers (1983) characterizations of an innovation, as a tool that can help solve some need or problem; agents help solve the need for automation in e-negotiation. Thus, Benham and Raymond's (1996) approach to technology acceptance is highly suitable for this thesis, since DOI provides readily validated beliefs with regards to innovation that can be used in conjunction with TPB.

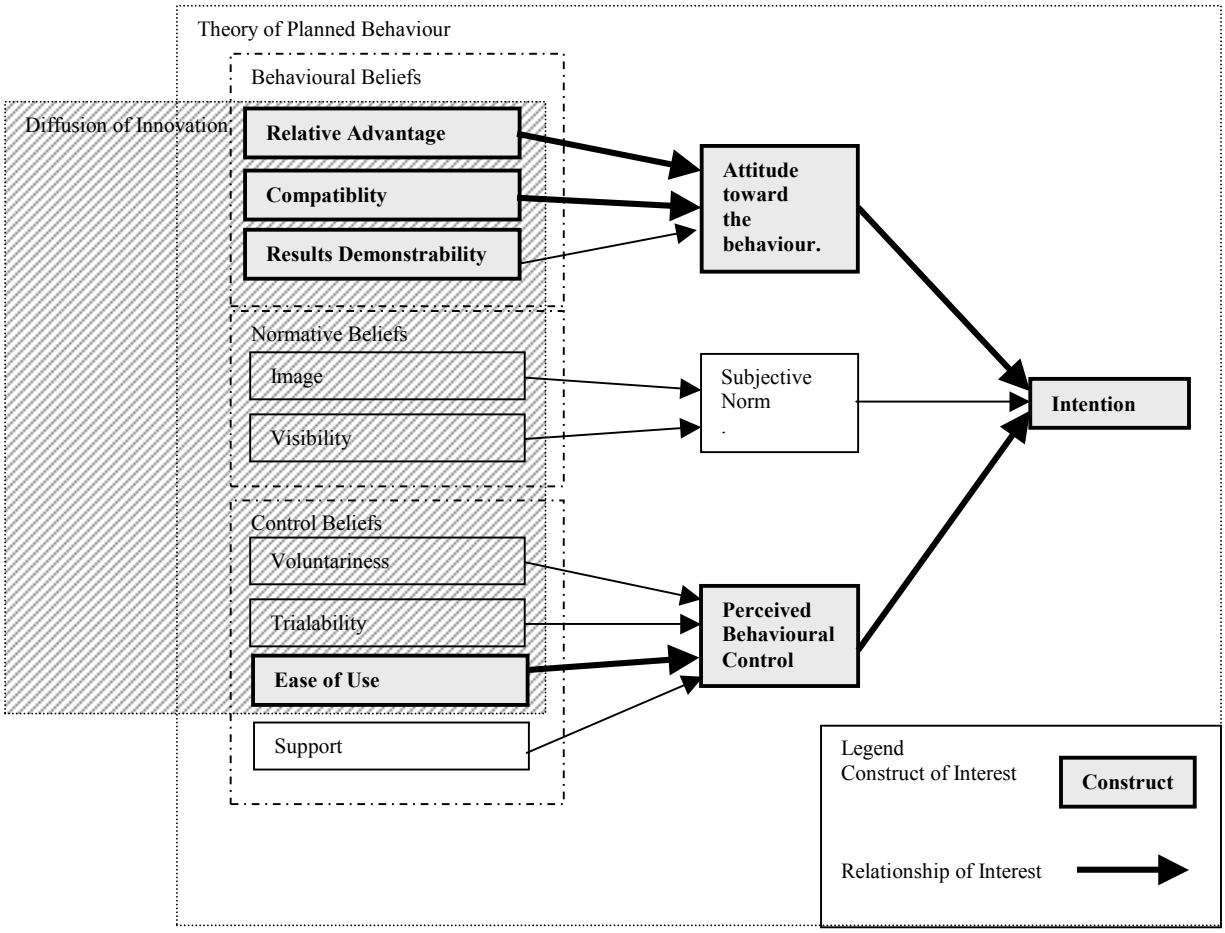


Figure 3 Benham and Raymond's adaptation of theory of planned behaviour.

Development of new theoretical frameworks does not require that established theories and constructs be ignored; rather, there is a need to re-examine what is valid and what is not valid. As Chau and Hu (2001) pointed out in their study of technology acceptance by professionals, there are "plausible limitations of TPB (theory of planned behaviour) and TAM (technology acceptance model) in a professional context" (p. 714). Similarly when the context of study changes to a non-commercial and non-mandatory

setting there is a need to develop new theoretical frameworks to predict adoption, as existing theories may not suitably account for all factors relevant to this particular contextual setting. Over the next few sections we will specify and develop our model by providing rationale as to why certain constructs and relationships are considered while others are not. A high-level overview is shown in Figure 3, in which bolded items refer to construct and relationships that are of interest to our work.

3.2.1 Dependent variable

We use *behavioural intention* as the dependent variable. This is consistent with both TAM and TPB. Further, Chau and Hu (2001) note "considerable prior studies have reported a strong and significant causal link between behavioural intention and target behaviour ... Given this strong link, use of behavioural intention as a dependent variable to examine technology acceptance is theoretically justifiable" (p. 701).

3.2.2 Attitude and behavioural beliefs

In our theoretical framework, we consider attitude to be an antecedent to behavioural intention. Past research has shown a strong link between attitude and behavioural intention (Chau & Hu, 2001; Davis et al., 1989; Mathieson, 1991). With regards to determinants of attitude, we categorized relative advantage and compatibility to be belief constructs similar to Benham and Raymond (1996). However, we consider results demonstrability to be an antecedent to relative advantage as past research work has shown a strong relationship between results demonstrability and relative advantage, (Venkatesh & Davis, 2000). We also introduce an additional construct in our model,

which we label "presence of learning agents". We consider presence of learning agents to be an antecedent to results demonstrability. Greater discussion will be given in the hypotheses section, elaborating on why we consider the presence of learning agents will have an effect on the tangible results perceived by the user.

3.2.3 Subjective norm and normative beliefs

In our theoretical framework, we do not consider subjective norm to be an antecedent to behavioural intention. Venkatesh and Davis (2000) found that subjective norm had significant influence on behavioural intention in settings where the degree of voluntariness was perceived to be low; however, subjective norm has no significant influence on behavioural intention in settings with a high degree of voluntariness. As discussed in the literature review, since the use of agents in e-negotiation has no associated degree of mandatory requirement, we consider that subjective norm will have no influence on behavioural intention. As well, subjective norm assumes existence of pre-formed opinions, which currently does not exist with regards to agents in e-negotiation, since this is a fairly new technology.

Along with omitting subjective norm from the theoretical framework, any construct identified as normative beliefs are also omitted. Benham and Raymond (1996) identified image and visibility as normative beliefs in their study of technology adoption of voicemail. These two constructs are omitted from the theoretical framework given that the theory of planned behaviour indicates normative beliefs' influence on behavioural intention is mediated by subjective norm.

3.2.4 Perceived behavioural control and control beliefs

In our theoretical framework, we consider perceived behavioural control to be an antecedent to behavioural intention. Past research has shown a strong link between perceived behavioural control and behavioural intention (Benham & Raymond, 1996; Mathieson, 1991). With regards to determinants of perceived behavioural control, we consider ease of use as a control belief, since Benham and Raymond (1996) classify ease of use as such. In some of the other studies (Chau & Hu, 2001; Davis et al., 1989; Jackson, Chow, & Leitch, 1997) ease of use has been classified as a behavioural belief; for example, perceived ease of use is considered a behavioural belief in TAM. However, TAM was adapted from the theory of reasoned action, which did not incorporate the construct of perceived behavioural control (Benham & Raymond, 1996). In comparing the technology acceptance model and the theory of planned behaviour, Matheison (1991) suggested that ease of use might be more of a control belief than a behavioural belief. As well, Davis et al (1989) suggest that "the easier a system is to interact with, the greater should be the user's sense of efficacy" (p. 987). Thus, we classify ease of use as a control belief that has an effect on perceived behavioural control.

As for the other constructs that Benham and Raymond (1996) considered to be determinants of perceived behavioural control, for reasons mentioned earlier, we do not include voluntariness in our theoretical framework. We also do not include trialability in our theoretical framework. Rogers (1983) introduced trialability as a construct for diffusion of innovation to account for the fact that an innovation may be experimented

upon before adoption (Rogers, 1983). This is a characteristic that is offered by commercial products but not by non-commercial products. Thus, the ability of a potential adoptee to experiment with e-negotiation before adoption is not something that is currently measurable in the current contextual setting. We also omit the support construct, considered by Benham and Raymond (1996) as a control belief, since it is not valid to the context of this thesis as this construct captures characteristics offered by a commercial product. Commercial products have associated resources and services pertinent to their use so as to provide support for their customers; however, non-commercial products do not have such resources and services. As well, the support construct is not a DOI construct. Its psychometric properties have not been extensively validated and thus it may not be reliable in different context (Benham & Raymond, 1996).

3.2.5 Our model

Figure 4 shows our proposed theoretical framework for explaining the technology acceptance of e-negotiation and explaining the effect of learning agents on the factors that influence technology acceptance. This theoretical framework incorporates some of the relevant constructs from TAM, TPB and DOI. The dependent variable of interest in this theoretical framework is the behavioural intention to use e-negotiation.

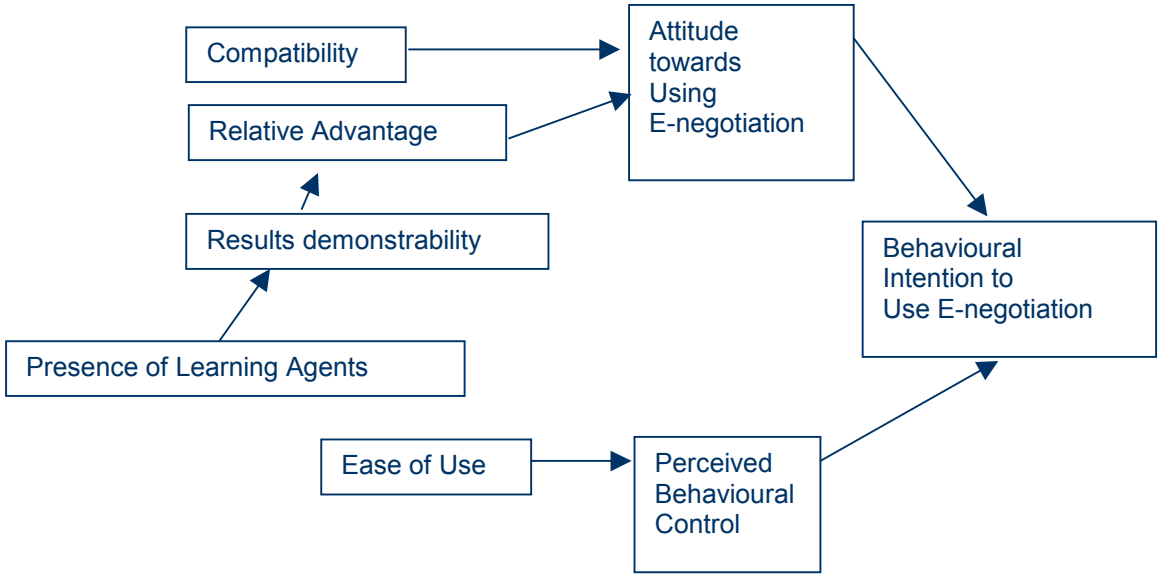


Figure 4 Our theoretical framework for testing e-negotiation acceptance.

The first set of constructs in this theoretical framework considered to influence behavioural intention is attitude and behavioural beliefs (compatibility, relative advantage, results demonstrability). The second set of constructs in this theoretical framework considered to influence behavioural intention is perceived behavioural control and control beliefs (ease of use). The presence of learning agents is considered primarily to have an effect through the results demonstrability of learning agents on e-negotiation.

Below we expand on each of these sets of constructs and outline the theoretical basis for the causal relationships proposed in the theoretical framework.

3.3 Hypotheses

The theoretical rationale and associated hypotheses will explain the causal relationships proposed in the theoretical framework and will also provide the theoretical basis for each of the hypotheses.

3.3.1 Attitude and Behavioural Beliefs

In technology acceptance literature, attitude is defined as a person's positive or negative evaluation of performing a type of behaviour (Chau & Hu, 2001). Hansen, Jensen, and Solgaard (2004) found that attitude had a strong influence on behavioural intention to use online shopping for groceries. Other past studies have also shown that attitude has a significant influence on behavioural intention to use a technology such as word processing software (Davis et al., 1989) or spreadsheet software (Mathieson, 1991). If an individual feels that there are positive consequences from using e-negotiation then the individual will have more intention to adopt e-negotiation. Conversely, feelings of negative consequences lower adoption intention. In this study, the first hypothesis we propose draws a causal link between attitude and behavioural intention in the context of e-negotiation.

Hypothesis 1: Attitude towards the use of e-negotiation will have a significant positive direct effect on the behavioural intention to use e-negotiation.

Amongst the factors considered by Benham and Raymond (1996), compatibility was considered a major determinant of attitude. Compatibility is defined as the degree of consistency with existing beliefs and needs (Rogers, 1983). By having a greater degree of consistency with existing values and needs, an innovation will reduce the uncertainty the potential adopter has about the innovation. Thus, it will positively increase the adopter's affect with regards to the using the innovation.

In their study of consumer acceptance of virtual stores, Chen, Gillenson and Sherrell (2002) found that compatibility has a significant influence on attitude. As well, Moore and Benbasat (1991) also found that compatibility has a significant influence on attitude amongst adopters in the context of personal workstations. The second hypothesis proposed asserts a causal link between compatibility and attitude in the context of e-negotiation. In the context of agents in e-negotiation, if an individual considers e-negotiation to offer at least the same level of benefits as other current technologies in e-commerce then the individual is likely to have a more positive attitude towards using e-negotiation. However, if the individual perceives e-negotiation to be not on par with other technologies, then the individual will have a more negative attitude towards using e-negotiation.

Hypothesis 2: The compatibility of e-negotiation with current e-commerce technologies will have a positive direct effect on the attitude towards using e-negotiation.

When an innovation is perceived to offer better benefits, an individual's perception of positive outcomes from using that innovation will increase. The degree to which an innovation is perceived to offer better benefits than earlier innovations of a similar nature is referred to as relative advantage (Rogers, 1983). In a longitudinal study of the adoption of Windows operating system in a single organization, Karahanna et al. (1999) found that relative advantage has a strong influence on attitude. Other studies (Benham & Raymond, 1996; Chau & Hu, 2001; Mathieson, 1991; Taylor & Todd, 1995) have also shown this relationship to exist in other contexts. We propose in our third hypothesis that relative advantage has a strong positive effect on attitude. A potential adopter's positive or negative evaluation of using e-negotiation is directly affected by the perceived improvements in benefits gained from using e-negotiation.

Hypothesis 3: The relative advantage offered by e-negotiation over other e-commerce technologies will have a significant positive direct effect on the attitude towards using e-negotiation.

The outcomes of e-negotiations using agents will strongly influence a potential adopter's perceptions about the type of results that e-negotiation can deliver. In turn, these perceptions will develop the adopter's valuation of the relative advantage offered by e-negotiation versus other e-commerce technologies. The degree to which the results of using an innovation are evident and tangible is referred to as results demonstrability (Moore & Benbasat, 1991). Venkatesh and Davis (2000) found that results demonstrability has a significant effect on relative advantage in a study of adoption of

accounting software. In the context of e-negotiation, the fourth hypothesis proposes that results demonstrability will have a positive direct effect on relative advantage.

Hypothesis 4: The results demonstrability of e-negotiation will have a positive direct effect on the relative advantage offered by e-negotiation over other e-commerce technologies.

3.3.2 Perceived Behavioural Control and Control Beliefs

An individual's sense of self-efficacy with regards to using an innovation will have a strong effect on the individual's behavioural intention to use the innovation. If individuals feel confident about their ability to use an innovation effectively their intention to use the innovation will be greatly enhanced, as they will not feel timid about using the innovation. The general ability that an individual feels that they have to engage in a certain behaviour is referred to as perceived behavioural control (Ajzen, 1985; Ajzen & Madden, 1986).

Chau and Hu (2001) found that perceived behavioural control had a significant influence on behavioural intention. Other studies (Benham & Raymond, 1996; Chau & Hu, 2001; Riemenschneider, Harrison, & Mykytn, 2003) have also shown this relationship to exist in other contexts. This implies that in the context of e-negotiation, an individual's behavioural intention to use e-negotiation will be affected by the confidence that individual has in his or her general ability to use e-negotiation. The fifth hypothesis proposed asserts a causal link between perceived behavioural control and behavioural intention to use e-negotiation.

Hypothesis 5: The perceived behavioural control with regard to e-negotiation will have a positive direct effect on the behavioural intention to use e-negotiation.

In their study of small business intending to develop a web presence, Riemenschneider, Harrison and Mykytn (2003) found that perceived behavioural control is significantly influenced by ease of use: the degree to which the use of a system is free of effort and fairly comprehensible (Davis et al., 1989). Our sixth hypothesis is that ease of use will have a positive direct effect on perceived behavioural control in the e-negotiation context. The amount of effort that is required to use an innovation will have an effect on the adopter's sense of self-efficacy with regards to using the innovation. If the individual finds that the he or she can comprehend fairly easily how to use an innovation, he or she is more likely to feel confident about using the innovation effectively. This implies, in the context of e-negotiation, the degree of simplicity that use of e-negotiation provides will significantly influence a person's confidence about his or her ability to use e-negotiation.

Hypothesis 6: E-negotiation's ease of use will have a significant positive direct effect on the perceived behavioural control with regard to e-negotiation.

3.3.3 Presence of Learning Agents

Research in human negotiation has shown that a negotiator who learns and adapts in a negotiation can significantly improve their negotiation outcomes (Raiffa, 1982). Generally, once individuals acquire information that allows them to compare their current strategy with an ideal strategy they can improve their performance (Balzer, Doherty, &

O'Connor, 1989; Hammond, McClelland, & Mumpower, 1980). Similarly, without using observation, drawing analogies or drawing on other strategies, experience alone is not very effective in negotiation (Nadler, Thompson, & van Boven, 2003). Nadler et al. (2003) compared four different types of learning in negotiation including: principle-based learning, learning via information revelation, analogical learning, and observational learning. Participants, who were exposed to principle-based learning, were taught in abstract terms the main principles in successful negotiations but were not provide sufficient applications of these principles. Those who were exposed to learning via information revelation, were provided with information about their opponent's preferences. In both of these learning situations, the participant is provided strategies or information and thus is not actively engaged in learning. Participants, who are exposed to analogical learning, were given analogical examples to illustrate negotiation principles. Those who are exposed to observational learning, were shown how a "model" negotiator would engage in negotiations. Unlike the previous two learning situations, these last two learning situations required the participants to actively engage in learning as they were not directly taught the strategies but rather had to learn it either through observation or inference.

Nadler et al. (2003) found that negotiators who were exposed to analogical learning or observational learning were relatively more effective than negotiators who used principle-based learning or learning via information revelation. This indicates that negotiators who learn from observation or adapt using a variety of strategies are more successful than those who base their negotiations on principles or voluntary information

revelation. Generally speaking, negotiators who actively adapt their strategies are more successful than those who maintain the same strategies or passively adapt their strategies. The observed effect of actively adapting new strategies in human negotiation has also been shown to hold in e-negotiation, especially when learning agents are present in the process (Mok & Sundarraj, 2005; Zeng & Sycara, 1998).

Learning agents provide the ability to "learn" in agent-based e-negotiation; i.e., a learning agent will "learn" over the course of negotiating with an opponent and will manipulate its negotiation behaviour to deliver a favourable outcome for its respective party. Zeng and Sycara (1998) incorporated Bayesian learning into their negotiation model and found that the joint utility was higher when both agents were learning than when they were not learning. Similarly, Mok and Sundarraj (2005) incorporated behavioural learning into their negotiation model and found that the joint utility was higher when both agents were learning than when they were not learning. Overall, both studies (Mok & Sundarraj, 2005; Zeng & Sycara, 1998) indicate that the presence of learning agents has a positive effect on negotiation outcomes, which will be perceived as a tangible outcome by the potential adopter. Thus, in this study we propose that the presence of learning agents will have a positive effect on the results demonstrability of e-negotiation. This proposal is put forward as hypothesis seven.

Hypothesis 7: The presence of learning agents will have a significant positive direct effect on the results demonstrability of e-negotiation.

Figure 5 presents the theoretical framework with the respective hypotheses denoted by an "H" followed by the hypothesis number.

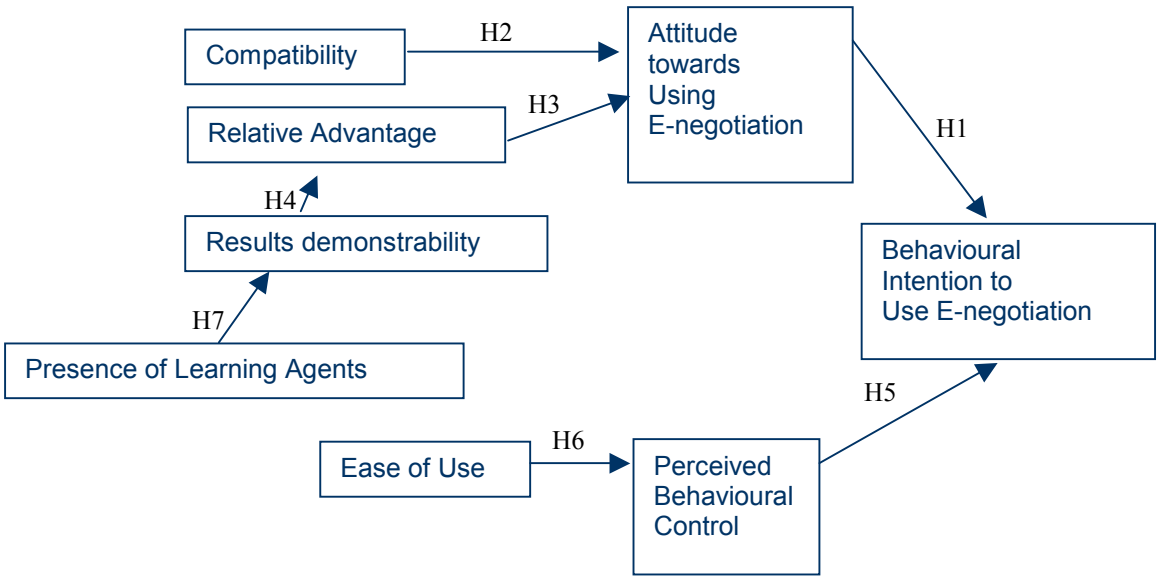


Figure 5 Theoretical framework to explain the technology acceptance of e-negotiation.

Chapter 4. Research Method

In order to study the adoption of e-negotiation, we conducted a survey study that required participants to use an e-negotiation tool and then complete an attitudinal survey. Based on algorithms in the literature (Mok & Sundarraj, 2005), we developed a custom e-negotiation tool since there was a general lack of commercially available e-negotiation tools that contained the particular technology. We developed the e-negotiation tool and the survey as web-based systems, as this ensured that the participants had the maximum accessibility to the study.

In this chapter, we will cover the development of the web-based e-negotiation tool, the deployment of the survey, and the validation of survey instrument using factor analysis.

4.1 System Design

One of the first exercises that need to be undertaken with the design of any system is to consider the major factors that can influence the system's utility to the end-user. Past work in decision support systems and negotiation (Benbasat & Lim, 1993; Thompson & Nadler, 2002) has shown the importance of two aspects. The first factor to consider is the effect of other activities other than negotiation on the overall course of the negotiation and negotiation outcomes. The second factor identified is the latent effect of support tools in the negotiation process.

Non-negotiation oriented activities can add disturbance to the negotiation process and thus potentially contribute a negative bias to the process. According to Thompson and Nadler (2002) non-negotiation task relevant activities can lead to inefficient negotiations. We placed a major focus in the design of the system interface to ensure that it offered the necessary components for an individual to engage in e-negotiation without the added distraction from other unnecessary components. One example of this is the reduction of the number of issues that required consideration and would be at stake for the participant. Past studies have considered a variety of issues ranging from price, time of delivery, quantity etc (Beam & Segev, 1997). For our study, we chose to limit the negotiation issues to one issue – price. By reducing the number of issues to negotiate on the user can more focus more on the process of negotiation versus having to consider and decide on the importance of the variety of issues. Price is also very suitable as a single issue since it is very applicable to a variety of domains on which negotiation can occur.

Support tools can aid the negotiation process and thus potentially contribute to a positive user experience in the negotiation process. According to Benbasat and Lim (1993) analytical tools in decision support systems could contribute to a positive user experience in the decision making process, an important aspect in the negotiation process. We incorporated analytical decision support interface as a major component in our system. Basically an analytic decision support interface can aid a user to assess their next step in the negotiation process (Koeszegi et al., 2004). One example of an analytic decision support interface is a graph or table that displays the history of the negotiation

offers. By having a view of past offers the user is able to assess how best to proceed with a new offer. Users were exposed to the web interface shown in Figure 6.

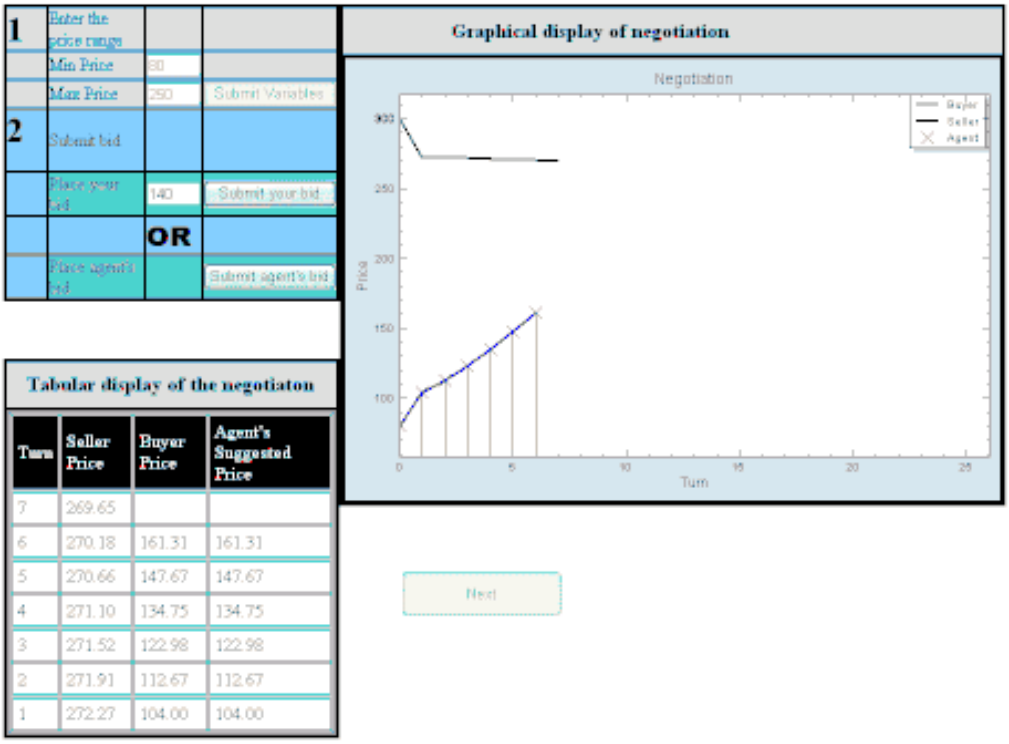


Figure 6 Web interface for e-negotiation.

As shown in Figure 6, we provided both a tabular and graphical format that would allow the user to track the history of negotiation offers. This allows the users to have both a graphical view of the negotiation offers as well as a tabular view that has the exact numerical values listed. This ensured that the user had access to two fairly common and relevant forms of analytic decision support tools.

Aside from design issues related to the presentation of the web interface, another issue that had to be considered was the negotiation technology behind the interface with

which the user was interacting. As shown in Figure 5, the user had two options when making an offer – they could enter an offer themselves or allow the agent to make an offer for them. Mok and Sundarraj (2005) developed an algorithm that allowed for the integration of learning into an autonomous negotiation agent. We adopted this algorithm for use in our e-negotiation tool and modified the algorithm so that it was possible for human mediation. We provided users the option to either make an offer or allow the agent to make an offer. At the start for each turn, the agent makes an estimate offer and then waits for users to select their choices. If the user elects to make an offer then the agent submits the user's offer. However, if the user elects to have the agent act on the user's behalf the agent submits its estimate offer.

A major concern that has been brought up in the literature, (Couper, 2000), with regards to web-based data gathering is the lack of access restrictions, which can lead to unauthorized or multiple responses, threatening the scientific validity of the results. Online surveys are readily accessible via the Internet and thus individuals or unknown participants can skew the integrity of the data collection process by accessing and completing the survey. To prevent this we introduced an authentication code system, which required participants to obtain a code from us to access the site. The code was not associated in any way to the data collection to ensure anonymity. It did however ensure that only a known population was able to access the site and unauthorized individuals were unable to access the survey. We also designed the system such that the authentication code expires once it has been used thus, preventing multiple entries by authorized participants.

4.2 Survey Instrument

The question items used in the survey instrument were adapted from past studies. Benham and Raymond (1996) had adapted the 25-item instrument from Moore and Benbasat (1991) for testing the behavioural beliefs, normative beliefs and control beliefs. They also adapted 12 items from the instrument from Mathieson (1991) for testing the TPB constructs: attitude, subjective norms and perceived behavioural control. The main modification they made was to change the context from personal workstations to voice mail (Benham & Raymond, 1996).

Our model differs from Benham and Raymond's (1996) model in that we exclude the subjective norms and related antecedents. We also do not include the control beliefs of voluntariness, trialability and support in our model. For the other constructs in the model we choose to adapt the same questionnaire items as Benham and Raymond (1996) except to change the context to e-negotiation. The questionnaire items are listed in Table A-2.

As mentioned earlier, we are also studying the effect of the presence of learning agents on acceptance of e-negotiation. However, this aspect is not measured through the use of questionnaire items, rather the presence of learning was a manipulation in a one-factor experiment. Participants were randomly assigned, by the system, to an e-negotiation where the agent employed learning (learning group) or an e-negotiation where the agent did not employ learning (non-learning group).

4.3 Pilot Study

To test the general readability and overall flow of the survey, we conducted a pretest of the e-negotiation tool and survey via a pilot study. The pilot study was conducted on four individuals, two of whom are graduate students and another two who are professionals working full-time in industry. The pilot study had the individuals completing the demographic questionnaire, engaging in the e-negotiation tool and completing the survey.

We queried the pilot study participants about the general readability, cognitive burden and comprehensiveness of the survey. We also queried about the e-negotiation experience and usability of e-negotiation interface. Participants responded overall that the questions posed were fairly self-explanatory; however, one of the participants pointed out the use of the term "negotiation behaviour" in the question COM_3 was somewhat confusing. Two of the participants commented on the need for greater detail in the instructions. They were somewhat confused initially and took a little bit of time to fully understand what they needed to do in the study.

Following the suggestions of the participants the question wording for COM_3 was changed from "my negotiation behaviour" to "the way I conduct negotiations". This improved the readability of the question item. We also introduced a PowerPoint presentation (Appendix C) to supplement existing instructions, which provided far more detail instructions and overview about the study.

Following the completion of the pilot study we submitted the survey and associated recruitment and information letters for review by the University research ethics office. The research ethics office provided provisional clearance and recommended some suggestions such as reducing the amount of wording in the initial introduction letter. They also requested rewording of some of the recruitment letters, to explicitly state where the study will take place. Following their suggestion and requests, the recruitment letters and introduction letter were modified and the package was resubmitted for final clearance. Following final clearance, the e-negotiation tool and survey were deployed and participants were recruited for the full study.

4.4 Content of Website Study

The website (for detailed specification see Appendix A and for software code see Appendix B) we developed contained six sections: introduction letter, demographic questionnaire, instructions, e-negotiation tool, attitudinal survey and a letter of appreciation. We placed the introduction letter as the first section in the website as this would provide the participants with the basic synopsis of the study. The introduction letter also included information assuring anonymity and privacy of data collected as required by the research ethics office at the University (Appendix A.3).

We included the demographic questionnaire as the second section of the website and this included questions on e-commerce experience, age, gender, income level, education level, educational background and employment status (Appendix A.4). We included the instructions as the third section before the e-negotiation tool so that the user

is given detailed directions on how to use the e-negotiation web interface (Appendix A.5).

We placed the e-negotiation web interface as the fourth section on the website which allowed the user to engage in negotiation with a seller agent (Appendix A.6). We set the attitudinal survey as the fifth section following the e-negotiation web interface (Appendix A.7). We included the letter of appreciation as the sixth and final section to wrap up the study (Appendix A.8).

4.5 Survey administration and response result

For our survey, we recruited from three different groups of subjects: undergraduate students, graduate students and full-time professionals. We promoted the survey in-class to approximately 600 undergraduate students enrolled in the various undergraduate courses offered by the department of Management Sciences. We also performed in-class promotion of the survey to approximately 50 graduate students enrolled in graduate courses offered by the department. Aside from in-class promotion, we also promoted the survey using poster advertisements and email (Appendix D).

We also promoted the survey by placing poster advertisements seeking participants throughout various information boards at the Faculty of Engineering. We promoted the survey to approximately 60 full-time professionals at three firms. We first acquired permission from company management before proceeding to contact the employees via

email. In all of our promotions, we explicitly stated that the survey was voluntary and all information collected would remain anonymous.

A total of 145 participants took part in the on-line study and completed the survey. There were 85 participants who were students and another 51 participants who were employed workers. There were 9 participants who were either unemployed, self-employed, retired or in some other employment situation. Of the 145 participants the system randomly assigned 75 participants to the learning group and it randomly assigned 70 participants to the non-learning group. From the 145 survey participants that took part in the study, 37 participants submitted survey data that was incomplete. These results are omitted from the analysis presented in the rest of this chapter.

4.6 Demographic analysis

We performed a frequency analysis on the demographic information that is presented in Table 1.

Table 1 Summary of demographic information.

Question Item	Category	Number	Percentage
E-commerce Experience*	Yes	65	60.2%
	No	43	39.8%
Age	18-25	44	40.7%
	26-35	49	45.4%
	36-45	11	10.2%
	46-55	3	2.8%
	56-65	1	0.9%
	Above 65	0	0%
Gender	Male	72	66.7%
	Female	36	33.3%
Income Level	< $10,000	30	27.8%
	$10,000-$29,999	32	29.6%

	$30,000-$49,999	16	14.8%
	$50,000-$79,999	20	18.5%
	> $80,000	10	9.3%
Education Level	Some High School	1	0.9%
	High School	25	23.2%
	College Diploma	7	6.5%
	Bachelors	42	38.9%
	Masters	29	26.9%
	Phd	1	0.9%
	Other	3	2.8%
Educational Background	Business	17	15.7%
	Engineering	50	46.3%
	Science	8	7.4%
	Liberal Arts	3	2.8%
	Information Technology	22	20.4%
	Health/Medical	1	0.9%
	Social Work	4	3.7%
	Other	3	2.8%
Employment Status	Student	59	54.6%
	Unemployed	3	2.8%
	Employed	42	38.9%
	Self-Employed	3	2.8%
	Retired	0	0%
	Other	1	1%

* E-commerce experience is defined as past/current usage of e-commerce sites such as Ebay (http://www.ebay.com) or Priceline (http://www.priceline.com)

Of the 108 participants who submitted valid responses 60.2% have past e-commerce experience while 39.8% do not have past e-commerce experience. Also of the 108 participants, 46.3% have an engineering educational background while 15.7% have a business background and 20.4% have an information technology back ground. As well, of the 108 participants, 40.7% of the participants were between the ages of 18-25 and 45.4% were between the ages of 26-35. The demographic analysis indicates that we have a sizable percentage of the participants who are experienced in e-commerce and have the right educational background for assessing a new technology.

Individuals with past e-commerce experience will be able to better judge the acceptance of new technologies such as e-negotiation which is applicable to e-commerce. Individuals with an engineering, business or information technology background have experience with handling new technologies. Thus, they may be better able to assess new technologies such as e-negotiation. Aside from statistics regarding individuals' background and experience, another statistic of relevance is that majority of the participants were under the age of 35. This is important, since this is the demographic group that will most likely use a new technology such as e-negotiation. Younger individuals are more likely to engage in e-commerce transactions and other online activities than are individuals who are middle aged or older. Thus, their increased representation is able to better reveal the adoption intention of the target population.

4.7 Reliability of survey instrument

We tested the reliability of our survey instrument by using Cronbach's Alpha to check the level of agreement between the various questionnaire items used to measure a target construct. We calculated the reliability levels for the scales of the constructs for three different groups: learning, non-learning and pooled data.

The reliability levels are shown in Table 2. According Moore and Benbasat (1991) acceptable values for Cronbach's alpha should be 0.7 or greater. During our calculation we found that the reliability level for the results demonstrability construct was lower than minimum acceptable values. Thus following the approach taken by Benham and

Raymond (1996), we dropped the fourth question item and found the newly calculated reliability levels to be acceptable values as shown in Table 2. For the perceived behavioural control construct the Cronbach's alpha are above 0.7 except for the non-learning group. We decided to proceed with using the same questionnaire items for perceived behavioural control since it is shown to be reliable in the other two groups and the value for the non-learning group is almost close to 0.7. All of our reliability levels for the other constructs are within acceptable values.

Table 2 Reliability levels for construct scales.

Construct	Learning	Non-Learning	Pooled
Relative Advantage	0.82	0.85	0.84
Compatibility	0.82	0.84	0.83
Results demonstrability	0.50	0.52	0.51
Results demonstrability*	0.84	0.86	0.85
Attitude	0.86	0.89	0.88
Ease of Use	0.80	0.82	0.82
Perceived Behavioural Control	0.75	0.67	0.71
Intention	0.91	0.89	0.89

*Results demonstrability with question item 4 dropped.

4.8 Validity of survey instrument

There are two types of validity that need to be tested: convergent and discriminant validity. We tested the validity of our survey instrument by using the multi-trait multi-

method approach (MTMM) introduced by Campbell and Fiske (1959). This has been recommended in the literature, (Adams et al., 1992; Davis, 1989; Sundarraj & Wu, 2005).

4.8.1 Convergent Validity

Convergent validity refers to the degree of common or shared variance among the various questionnaire items used to measure a target construct (Judd, Smith, & Kidder, 1986). According to Davis (1989) one way to show convergent validity is for questionnaire items intended to measure the same trait correlate highly with each other. In our case, this would mean that the questionnaire items for the various constructs would highly correlate with each other in the three different groups of learning, non-learning and pooled.

As shown in Table 3, all the correlations are statistically significant and above 0.25 except for two item pair, (REL_1, REL_5) for learning and (EOU_1, EOU_4) for learning. However, for both of the two pairs that do not correlate in the learning group, across the non-learning and pooled groups, they are highly correlated. We therefore proceed with these scales.

Table 3 Item pairs that were not statistically significant or had correlation less than 0.25.

Construct	Learning	Non-Learning	Pooled
Relative Advantage	(REL_1, REL_5)	None	None
Compatibility	None	None	None
Results demonstrability	None	None	None
Attitude	None	None	None
Ease of Use	(EOU_1, EOU_4)	None	None
Perceived Behavioural Control	None	None	None
Intention	None	None	None

4.8.2 Discriminant Validity

Discriminant validity refers to the degree of lack of common or shared variance among various questionnaire items intended to measure different target constructs (Judd et al., 1986). According to Davis (1989), the test for discriminant validity is that an item should correlate more highly with other items intended to measure the same trait than with the same item or different items measuring different traits. We analyzed the correlations between the various questionnaire items for the learning group and the non-learning group to test for discriminant validity of our survey instrument.

Let REL_{ij} represent the relative advantage construct for the two groups (i = 1 for learning group, i = 2 for non-learning group) for the questionnaire item j ($j \in 1, 2, \ldots, 5$). We test discriminant validity by first comparing the correlation between REL_{1a} and

REL_{1b} ($a \neq b$, and $a, b \leq 5$) with the correlation between REL_{1a} and REL_{2a} and with the correlation between REL_{1a} and REL_{2c} ($c \neq a$, and $c \leq 5$). Altogether there are 200 pairs of correlations to be compared for the two groups. For discriminant validity to be verified the first correlation set (REL_{1a}, REL_{1b}) has to be greater than the last two correlation sets {(REL_{1a}, REL_{2a}), (REL_{1a}, REL_{2c})}. The number of exceptions is tallied and if this is significantly less than the total number of comparisons, then discriminant validity is demonstrated. This procedure is repeated for all the constructs in our model and the summary of this analysis is shown in Table 4.

Table 4 Summary of analysis for discriminant validity.

Construct	Number of Exceptions	Number of Comparisons	Percentage of Exceptions
Relative Advantage	0	200	0%
Compatibility	0	36	0%
Results demonstrability	0	36	0%
Attitude	0	36	0%
Ease of Use	0	96	0%
Perceived Behavioural Control	0	36	0%
Intention	0	8	0%

According to Adams et al (1992), 3% or less is an acceptable level for the percentage of exceptions for discriminant validity to be demonstrated. As can be seen in Table 4 our percentage of exceptions for all the constructs meets the acceptable levels

indicated in literature and thus the discriminant validity of construct scales has been verified.

4.9 Factor Analysis

Factor analysis is a statistical technique that reveals underlying latent factors across various questionnaire items, enabling the amalgamation of different questionnaire items into a single construct scale (Norusis, 1994). Before performing factor analysis we must also evaluate the sampling adequacy of the data to ensure the factor analysis is appropriate by calculating the Kaiser-Meyer-Olkin (KMO) measure, an index that compares the magnitudes of the observed correlation coefficients to the magnitudes of the partial correlation coefficients (Norusis, 1994). A low KMO value indicates that the factor analysis may not be valid; however, a KMO value close to 1 is generally considered ideal. Generally, a KMO measure greater that 0.50 is considered sufficient for factor analysis to be valid, a value of 0.70 is considered "middling" and a value of 0.80 is considered "meritorious"(Norusis, 1994).

Initially, we conducted an exploratory factor analysis with all the question items. Three factors were extracted; however, we found that the factor loadings were not clean and the question items did not load in the particular manner that was expected. The ease of use and perceived behavioural control items loaded on one factor while the attitude and intention question items loaded another factor. Some of the relative advantage, compatibility and results demonstrability question items loaded on the one factor with attitude and intention while others loaded on another completely separate factor. One

relative advantage item and one compatibility question item cross-loaded significantly on two of the factors. Since this initial exploratory approach does not yield results that we can use in our analysis, we take an alternative factor analysis approach adopted by Benham and Raymond (1996).

Our research model can be expressed by the following equations:

$$INT = \alpha_1 + \beta_1 ATT + \beta_2 PBC \quad \text{(Equation 1)}$$

$$ATT = \alpha_1 + \beta_1 COM + \beta_2 REL \quad \text{(Equation 2)}$$

$$REL = \alpha_1 + \beta_1 RSLT \quad \text{(Equation 3)}$$

$$PBC = \alpha_1 + \beta_1 EOU \quad \text{(Equation 4)}$$

Similar to past studies (Benham & Raymond, 1996; Chin, 2006; Liao, Shao, Wang, & Chen, 1999), for each of the regression equations in our model we will conduct factor analysis to confirm the proposed composite factors. We will first attempt a factor analysis, using varimax rotation, by extracting factors with eigenvalues greater than 1. If we are unable to extract the proposed composite factors, we attempt to extract the proposed composite factors by forcing the factors. This has been done with Chin (2006): when the initial factor analysis did not yield the desired set of factors, a subsequent factor analysis was performed while forcing the number of factors. If the second approach does not yield the desired set of factors we will accept the composite factors extracted in the first factor analysis.

For equation 1, we performed factor analysis, using varimax rotation and extracted factors with eigenvalues greater than 1 on the attitude and perceived

behavioural control questionnaire items. The results of the factor analysis are shown in Table 5 and only factor loadings larger than 0.5 are shown, since these are considered significant (Hair, Andersin, Tatham, & Black, 1992).

Table 5 Factor analysis of theory of planned behaviour (attitude, perceived behavioural control) items using varimax rotation.

Question Items	Factor 1
ATT_1	.892
ATT_2	.821
ATT_3	.870
PBC_1	.658
PBC_2	.657
PBC_3	.862
Eigenvalue	3.834
% of variance	63.896%
Kaiser-Meyer-Olkin	0.835

As can be seen in Table 5, the KMO measure indicates that the sampling is adequate, however, the questionnaire items loaded on one factor rather than two. We then forced the extraction of two factors. The question items loaded on two factors with the exception of the third question item for perceived behavioural control, PBC_3. We dropped this question item and reran the factor analysis procedure by forcing two factors. The question items loaded on two factors, with attitude items loading on one factor and perceived behavioural control items loading on the second factor. The results of this factor analysis are shown in Table 6.

Table 6 Factor analysis of theory of planned behaviour (attitude, perceived behavioural control) items using varimax rotation and two factor extraction.

Question Items	Factor 1	Factor 2
ATT_1	.755	
ATT_2	.880	
ATT_3	.902	
PBC_1		.788
PBC_2		.892
Eigenvalue	2.276	1.724
% of variance	45.525%	34.480%
Kaiser-Meyer-Olkin	0.755	

As can be seen in Table 6, the KMO measure indicates that the sampling is adequate. The attitude questionnaire items all loaded on one factor while perceived behavioural control questionnaire items loaded on the second factor. Thus, we accept the first factor as attitude and the second factor as perceived behavioural control.

We applied the factor analysis to equation 2 and the results are shown in Table 7. As can be seen in Table 7, the KMO measure indicates that the sampling is adequate. However, the questionnaire items load on one factor rather than two factors. We then forced the extraction of two factors. However, the loadings were not clean, with some items from both the relative advantage questionnaire items and compatibility questionnaire items loading significantly on one factor while the rest of the items loaded significantly on the second factor. Thus, we accepted the results from the first analysis and accepted the one factor as representing relative advantage, since the relative advantage questionnaire items dominated the loadings. Benham and Raymond (1996) experienced similar results where the questionnaire items for relative advantage and compatibility loaded on one factor, which they chose to accept as representing relative

advantage. This also means that hypothesis 2 cannot be verified as we are no longer considering the compatibility construct in our model.

Table 7 Factor analysis of relative advantage and compatibility items using varimax rotation.

Question Items	Factor 1
REL_1	.634
REL_2	.786
REL_3	.799
REL_4	.815
REL_5	.737
COM_1	.730
COM_2	.823
COM_3	.865
Eigenvalue	4.826
% of variance	60.327%
Kaiser-Meyer-Olkin	0.895

We applied the factor analysis to equation 3 and the results are shown in Table 8. As can be seen in Table 8, the KMO measure indicates that the sampling is adequate. The questionnaire items load on one factor as expected; thus, we accept the questionnaire items are measuring results demonstrability factor.

Table 8 Factor analysis of results demonstrability items using varimax rotation.

Question Items	Factor 1
RSLT_1	.898
RSLT_2	.896
RSLT_3	.843
Eigenvalue	2.320
% of variance	77.329%
Kaiser-Meyer-Olkin	.717

We apply the factor analysis to equation 4 and results are shown in Table 9. As can be seen in Table 9, the KMO measure indicates that the sampling is adequate. The questionnaire items all loaded on one factor as expected; thus, we accept that the questionnaire items are measuring ease of use.

Table 9 Factor analysis of control belief items using varimax rotation.

Question Items	Factor 1
EOU_1	.754
EOU_2	.772
EOU_3	.859
EOU_4	.832
Eigenvalue	2.596
% of variance	64.893%
Kaiser-Meyer-Olkin	0.780

From the factor analysis carried out, we were able to validate equation 1, equation 3 and equation 4 since the proposed constructs were empirically legitimate. However, equation 2 had to be modified as we found that only one factor, not the two factors proposed, could be extracted from the data. Equation 2 has now been modified as the following equation:

$$ATT = \alpha_1 + \beta_1 REL \qquad \text{(Equation 2)}$$

This meant, as well that we had to revise the model as shown in Figure 7.

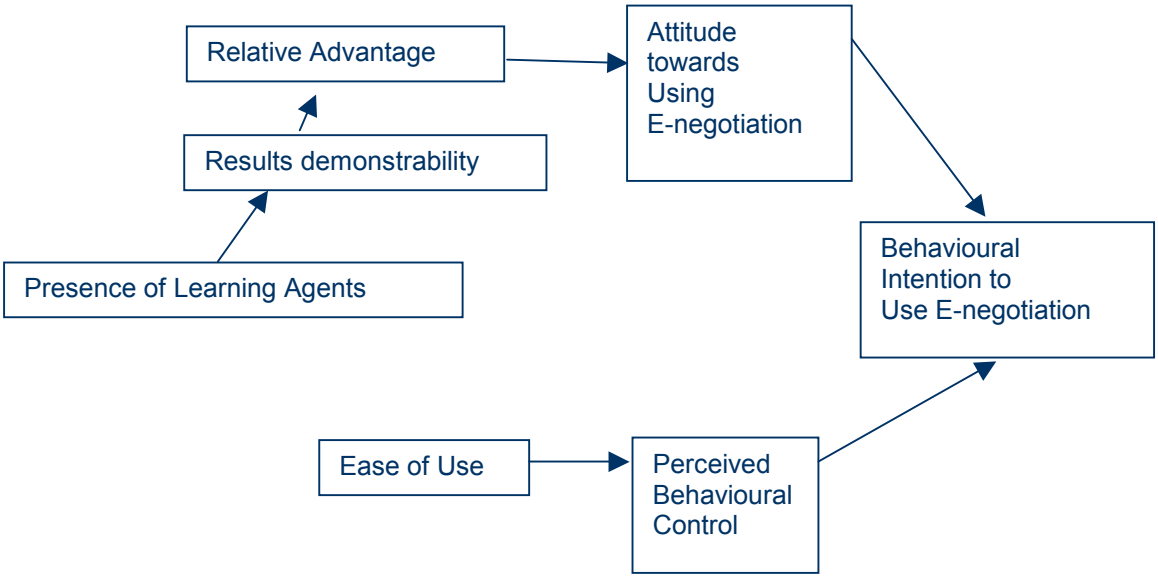


Figure 7 Revised theoretical framework that explains technology acceptance of e-negotiation.

Chapter 5. Results and Data Analysis

In this chapter we will perform data analysis and test the survey data to verify the proposed hypotheses. A regression analysis on the data is conducted to verify the hypotheses. We will also conduct a MANOVA test to assess the learning agent's effect on results demonstrability. Finally in the general discussion, the hypotheses will be looked at to see those that were shown to hold true empirically. The discussion will also cover why some of the hypotheses were not verified or did not hold true empirically.

5.1 Regression Analysis

In order to test the various hypotheses that were proposed, we performed regression analysis. Structural equation modeling had been explored, however, the goodness-of-fit indices were below acceptable levels and thus, we chose to proceed with regression analysis to verify our hypotheses. Based on the factor analysis, we dropped the compatibility construct from our model and merged the compatibility question items with the relative advantage question items to represent the relative advantage construct. As well based on the factor analysis, we dropped the third perceived behavioural question item, PBC_3, from our regression analysis. The new model is shown in Figure 7 and can also be expressed in the following equations:

$$INT = \alpha_1 + \beta_1 ATT + \beta_2 PBC \quad \text{(Equation 1)}$$

$$ATT = \alpha_1 + \beta_1 REL \quad \text{(Equation 2)}$$

$$REL = \alpha_1 + \beta_1 RSLT \quad \text{(Equation 3)}$$

$$PBC = \alpha_1 + \beta_1 EOU \quad \text{(Equation 4)}$$

The first step to preparing the data for analysis was to summate individual questionnaire item's scores to arrive at an overall score for each construct. Table 10 shows the items that were summated.

Table 10 Summated score and component items

Summated Score	Component Items
INT	INT_1 INT_2
ATT	ATT_1 ATT_2 ATT_3
PBC	PBC_1 PBC_2
REL	REL_1 REL_2 REL_3 REL_4 REL_5 COM_1 COM_2 COM_3
RSLT	RSLT_1 RSLT_2 RSLT_3
EOU	EOU_1 EOU_2 EOU_3 EOU_4

Using SPSS 14.0, we conducted the regression analysis using the overall score for each constructs as values for the variables in the equations. Table 11 summarizes the results of the regression analysis for the hypothesized relationships. In the discussion that

follows we refer to the regression coefficient of a variables as the beta value of the variable.

Table 11 Regression results for hypothesized relationships

Equation	Variables	Adjusted R-Square	Beta	t	Significance
1) INT = $\alpha_1 + \beta_1$ ATT + β_2 PBC		0.742			
	ATT		0.804	13.511	0.000
	PBC		0.099	1.670	0.098
2) ATT = $\alpha_1 + \beta_1$ REL		0.762			
	REL		0.874	18.523	0.000
3) REL = $\alpha_1 + \beta_1$ RSLT		0.625			
	RSLT		0.793	13.387	0.000
4) PBC = $\alpha_1 + \beta_1$ EOU		0.455			
	EOU		0.678	9.507	0.000

From equation 1, we see that the antecedents to behavioural intention account for most of the variance in behavioural intention (adjusted $R^2 = 0.742$). The beta value, 0.804, for the attitude construct was statistically significant ($p < 0.05$), while the beta value, 0.099, for the perceived behavioural control was mildly significant ($p < 0.10$). Therefore, in the context of e-negotiation agents, attitude has a positive effect on behavioural intention, thus verifying hypothesis 1, while perceived behavioural control has a positive effect on behavioural intention, verifying hypothesis 5.

The regression results for equation 2 indicate that behavioural beliefs sufficiently account for most of the variance in attitude (adjusted $R^2 = 0.762$). The beta value, 0.874, for the relative advantage construct was statistically significant ($p < 0.05$). This ascertains that relative advantage has a positive direct effect on attitude (hypothesis 3). From equation 3, we see that the antecedent to relative advantage accounts for most of the variance in relative advantage (adjusted $R^2 = 0.625$). The results demonstrability construct has a beta value, 0.793 that is statistically significant ($p < 0.05$). From this we can surmise that results demonstrability has a positive direct effect on relative advantage, which verifies hypothesis 4.

From regression analysis carried out on equation 4, we conclude that control beliefs do not fully account for most of the variance in perceived behavioural control (adjusted $R^2 = 0.455$). The ease of use construct has a beta value, 0.678 that is statistically significant ($p < 0.05$). This means that ease of use has a positive effect on perceived behavioural control (hypothesis 6).

5.2 Analysis of the effect of learning agents

We conduct a MANOVA test on results demonstrability questionnaire items to assess the effect of learning agents.

5.2.1 MANOVA analysis

MANOVA is a statistical technique that is very similar to ANOVA. In ANOVA, the dependent variable's values are divided into groups corresponding to each value of

the independent variable. The effect of the independent variable on the dependent variable is measured by the F-statistic (Hair et al., 1992): high F value indicates significance, while a low F value indicates otherwise.

Unlike ANOVA, MANOVA tests to check if the independent variable has an effect on two or more dependent variables. In MANOVA, the equivalent to the F-statistic is the Wilks Lambda statistic (Hair et al., 1992). For our study, the independent variable is the presence of learning agents, a binary variable, with 0 indicating no presence and 1 indicating presence. The dependent variables are results demonstrability's questionnaire items.

From our experiment, we obtain the Wilks Lambda statistic = 0.487 ($p < 0.7$). This means that the data for the learning and non-learning groups are not significantly different, indicating that the presence of learning agents has no effect on results demonstrability, thus hypothesis 7 does not hold empirically. We can infer from this that the participants' perception of the tangible outcomes, offered by e-negotiation, was not influenced by the presence of learning agents. To investigate the reason behind this, we examine the outcomes of the negotiations using learning agents versus those using non-learning ones.

5.2.2 Negotiation outcome analysis

We make three sets of comparisons to examine outcome difference. In the first two sets, we test respectively whether the learning and non-learning groups perform

better than an acceptable baseline. If either of the two groups provides outcomes better than the baseline, the participants in the respective groups would perceive some degree of tangible results. Since there is a possibility that both groups can provide outcomes better than the baseline, we also make a third comparison between the learning group and the non-learning group to see which performs better. It could be argued that the third comparison may be sufficient to answer our initial question about why the presence of learning agents has no significant effect on results demonstrability. However, even if one group provides a better outcome than the other group, individuals may not perceive tangible results if the outcomes are not better than a theoretical baseline.

Following Deveaux et al. (2001), our baseline outcome is the Nash solution, which is an equilibrium that maximizes the joint utility for both parties (Nash, 1950). From the results of the three t-tests comparisons, given in Table 12, we see that while both learning and non-learning groups differ from the baseline ($p = 0.017$ & 0.008), there were no outcome difference between the groups themselves ($p = 0.406$). Thus, learning group participants may not have perceived outcome improvement and in turn, there was no effect on results demonstrability.

Table 12 Summary of comparisons

Comparison of Negotiation Outcomes	**t**	**Significance (2-tail)**
Learning versus Nash	2.458	.017
Non-Learning versus Nash	2.758	.008
Learning versus Non-Learning	-0.835	.406

Chapter 6. General Discussion

Thus far, we presented a theoretical model that explains the technology acceptance of e-negotiation. To test the model, we designed a web interface for an e-negotiation system, as well as a survey to elicit attitudinal perceptions. We also analyzed the survey results, by using regression and MANOVA. This chapter presents a summary of the findings and discusses the conclusions that can be drawn there from.

6.1 Summary of Findings

In this thesis, we explored a number of research issues pertaining to the survey instrument, the theoretical model and results of the survey, including:

1. The reliability and validity of the survey instrument.
2. The empirical verification of the postulated factors in our model.
3. The various equations' explanatory power: how much of the variance in the dependent variable is accounted for by the independent variables.
4. The verification of the various interrelationships between the model's constructs in the theory model.
5. The effect of learning agents on results demonstrability.

The findings, in regard to each of the research issues, are summarized in the Table 13.

Table 13 Summary of findings

Research Issue	Findings
Reliability and validity survey instrument.	Reliability • Cronbach's alpha for all the constructs greater than 0.7. • For results demonstrability question item 4, RSLT_4 had to be dropped for alpha to be greater than 0.7. Convergent validity • Demonstrated since question items correlate highly with other items that measure the same trait. • Exceptions: one correlation pair for relative advantage and one correlation pair for ease of use, only in learning group. Discriminant validity • Demonstrated since all question items correlate more highly with other items that measure the same trait than with the same item or different items measuring different traits.
Empirical verification of postulated factors.	$INT = \alpha_1 + \beta_1 ATT + \beta_2 PBC$ (Equation 1) • Both attitude and perceived behavioural control are shown to be two separate factors using factor analysis. • Perceived behavioural control's 3rd question item PBC_3 was dropped, as it did not load with the rest of the question items. $ATT = \alpha_1 + \beta_1 COM + \beta_2 REL$ (Equation 2) • Compatibility and relative advantage load on one factor with relative advantage dominating. • Equation 2 is revised as $ATT = \alpha_1 + \beta_1 REL$ $REL = \alpha_1 + \beta_1 RSLT$ (Equation 3) • Results demonstrability loaded on one factor. $PBC = \alpha_1 + \beta_1 EOU$ (Equation 4) • Ease of use loaded on one factor using factor analysis.
Explanatory power of equations.	$INT = \alpha_1 + \beta_1 ATT + \beta_2 PBC$ (adjusted $R^2 = 0.742$). (Equation 1) $ATT = \alpha_1 + \beta_1 REL$ (adjusted $R^2 = 0.762$). (Equation 2) $REL = \alpha_1 + \beta_1 RSLT$ (adjusted $R^2 = 0.625$). (Equation 3) $PBC = \alpha_1 + \beta_1 EOU$ (adjusted $R^2 = 0.455$) (Equation 4)
Interrationships between various variables.	$ATT \rightarrow INT$, $\beta = 0.804$ ($p < 0.05$), Hypothesis 1 verified $COM \rightarrow ATT$, not considered, Hypothesis 2 not verified $REL \rightarrow ATT$, $\beta = 0.874$ ($p < 0.05$), Hypothesis 3 verified $RSLT \rightarrow REL$, $\beta = 0.793$ ($p < 0.05$), Hypothesis 4 verified $PBC \rightarrow INT$, $\beta = 0.099$ ($p < 0.10$), Hypothesis 5 verified $EOU \rightarrow PBC$, $\beta = 0.678$ ($p < 0.05$), Hypothesis 6 verified
Effect of learning agents on results demonstrability.	• Presence of Learning Agents $\rightarrow$ RSLT, no effect (Wilks Lambda = 0.487, $p < 0.7$), Hypothesis 7 does not hold empirically • T-tests indicated that outcomes from negotiation with learning agents did not perform better than those that had non-learning ones - both groups performed worse than their baseline.

6.2 Discussion

Cronbach's alpha values greater than 0.7, for the postulated factors (except for results demonstrability), confirmed the survey instrument's reliability. After dropping the fourth question item for results demonstrability, similar to Benham and Raymond (1996), the Cronbach's alpha for results demonstrability was also greater than 0.7. The convergent and discriminant validity analysis, using multi-trait multi-method (Campbell & Fiske, 1959), confirmed the survey instrument's validity.

We conducted a factor analysis to empirically verify that the postulated factors held empirically. For the perceived behavioural control construct, the third question item for perceived behavioural control loaded on another factor and thus was dropped. After this drop, the rest of the perceived behavioural control question items loaded on a separate factor, while attitude items loaded on another factor. The factor analysis indicated that compatibility and relative advantage constructs loaded on one factor. Although, Moore and Benbasat (1991) originally posited conceptual differences between compatibility and relative advantage, we find that empirically there is no distinction between compatibility and relative advantage, and thus arrive at the same conclusion as Benham and Raymond (1996). Therefore the compatibility and relative advantage questionnaire items were amalgamated to represent relative advantage. All the other constructs loaded on individual factors confirming that the postulated factors held empirically.

Our findings indicate that the antecedents to behavioural intention sufficiently account for its variance, which means that we can conclude that our model accounts for the factors that influence behavioural intention. Attitude was shown to have a positive direct effect on behavioural intention, which is consistent with past studies (Davis et al., 1989; Hansen, Jensen, & Solgaard, 2004; Mathieson, 1991). Perceived behavioural control did have a positive direct effect on behavioural intention, which is consistent with past work (Benham & Raymond, 1996; Chau & Hu, 2001; Riemenschneider, Harrison, & Mykytn, 2003).

In our study, the antecedent to attitude accounts for much of its variance. Consistent with previous studies (Benham & Raymond, 1996; Chau & Hu, 2001; Karahanna et al., 1999; Mathieson, 1991; Taylor & Todd, 1995), relative advantage was shown to have a positive direct effect on attitude. Our findings also indicate that the antecedent to relative advantage accounts for much of its variance. Results demonstrability was shown to have a positive direct effect on relative advantage, which is consistent with the relationship found by Venkatesh and Davis (2000).

The study findings indicate that the presence of learning agents did not have an effect on results demonstrability. A possible explanation for this was offered by the analysis carried out, which indicated the outcomes of the negotiations in the learning group and the non-learning group were not different. Thus, the participants' perception of tangible results was not significantly affected by the presence of learning agents. It is quite possible that the fact that we allowed for human mediation could have contributed

to the lack of difference in the negotiation outcomes for the learning group and the non-learning group. A lack of trust in the agent technology by participants may have led them to intervene more in the negotiation process thus preventing the learning agent from being able to deliver the superior outcomes.

Our finding also indicates that the antecedent to perceived behavioural control did not fully account for much of its variance. However, ease of use did have a positive direct effect on perceived behavioural control, which is consistent with past studies (Benham & Raymond, 1996; Riemenschneider et al., 2003). This indicates that there are some other factors that affect perceived behavioural control, which have not been accounted for by our theoretical model.

Chapter 7. Conclusions, Implications, Limitations and Future Research

7.1.1 Conclusions

For the study, we tested the validity of our theoretical model by developing developed a web interface, based on algorithms given by Mok and Sundarraj (2005). Participants used our system to gain user experience in e-negotiation. The web-based system contained essential components of e-negotiation such as an analytic decision support interface (i.e. a chart illustrating history of offers), which Benbasat and Lim (1993) identified as contributing to a positive user experience with the decision making process. The negotiation technology behind the system used behavioural models for agents to engage in negotiation. These behavioural models also incorporated the "learning" feature, the ability to alter the agent's negotiation tactics based on interaction with the other agent in the negotiation. Participants were randomly assigned by the system to either a learning or non-learning group.

A survey instrument was used to assess the participants' experience after they had finished using the e-negotiation system. This survey instrument was adapted from past studies (Davis, 1989; Mathieson, 1991; Moore & Benbasat, 1991) and pre-tested to ensure that it was comprehensible to participants and did not present a cognitive burden. The reliability and validity of this instrument was confirmed and the postulated factors were empirically verified with the exception of the compatibility construct.

All the key interrelationships in the model were verified with the exception of the compatibility and attitude relationship. In conclusion, the theoretical model identifies and explains the most of the factors that influence technology acceptance of e-negotiation. As well, the presence of learning agents did not have an effect on results demonstrability in our study.

In this study, we contributed to the e-negotiation research area by putting forward a theoretical model that identifies and explains the factors that influence the technology acceptance of e-negotiation. We also contributed by examining what effect the presence of learning agents has on perceptions of negotiation outcomes, in turn affecting acceptance of e-negotiation. Our contribution to the literature is the development of a model that explains the acceptance of an experimental technology that is not commercially available and has no pre-defined environment of usage, thus lacking any level of mandatory directive concerning its usage.

7.1.2 Implication of findings for usage of survey instrument in future studies

The confirmation of the survey instrument's reliability and validity indicates that the items adapted from past studies (Davis, 1989; Mathieson, 1991; Moore & Benbasat, 1991) are also applicable to the context of e-negotiation. This further strengthens the case for using these instruments in other technology acceptance studies. The factor analysis also validated empirically that the question items represented the postulated constructs except for the case of the compatibility construct. This means that in future

technology acceptance studies, the question items for the compatibility construct should be evaluated further.

7.1.3 Implication of findings for technology acceptance studies

The findings indicate that the theoretical model sufficiently accounts for the variance in behavioural intention to use e-negotiation. As well, most of the interrelationships and hypotheses hold empirically, thus strengthening the case for the usage or adaptation of the model in future studies of experimental technologies. One component of the model that will need some consideration in future studies are the antecedents to perceived behavioural control. The relatively lower explanatory power of the antecedent to perceived behavioural control indicates that other factors should be explored, so as to strengthen the model.

7.1.4 Implications for e-negotiation research

The influence of attitude on behavioural intention indicates that researchers should focus on developing e-negotiation technology, so that it can increase the positive evaluation of using e-negotiation. The model also points a way to achieve this as it indicates that results demonstrability has a positive direct effect on relative advantage, which in turn has a positive direct effect on attitude. The technology acceptance of e-negotiation can be positively influenced when the perceived tangible results from using e-negotiation are increased.

Although we found that the presence of learning agents did not of a significant effect on results demonstrability, we feel that researchers should not discontinue research in learning agents, as the model still empirically showed results demonstrability's influence on relative advantage. Instead, this outcome should be an impetus for researchers to continue work on learning agents, so that they can deliver better negotiation outcomes.

7.1.5 Limitations and future research

One limitation we identify in our study is the lack of explanatory power possessed by the proposed antecedents to perceived behavioural control. Thus, there is the need for further exploration and identification of factors that would influence perceived behavioural control. Future studies can ask participants to elicit thoughts on what factors contribute to their sense of self-efficacy vis-à-vis an experimental technology. These elicited factors can be further explored for commonalities and can thus lead to identification of new control belief constructs. Another limitation that we identify in our study is that the theoretical model has only been tested on one technology. We also suggest testing this theoretical model on other experimental technologies to give further validation to this model. Another limitation in our study is that we do not assess how well users trust the agents, as listed in our discussion this could have contributed to the inability of the learning agents to deliver better outcomes. Future studies should seek to assess this factor as well through the development of new items for this construct.

Appendix A: Web Interface Specification

A.1 Web Interface

The system is composed of 6 distinct areas: introduction, demographics, instructions, negotiation form, questionnaire and letter of appreciation. An overview of the system is presented in Figure A-1. In Table A-1, the six different areas are indicated with their corresponding web components and the appendix B sections, which contain the software code. For further detail not provided in the specification on how to use the system, refer to Appendix C.

Table A-1 Website areas and associated web components.

Web Areas	Web Components	Code
Introduction	default.aspx	Appendix B.4
	DBSession.vb	Appendix B.3
Demographics	Demographic.aspx	Appendix B.5
	DBSession.vb	Appendix B.3
Instructions	Instructions.aspx	Appendix B.6
	DBSession.vb	Appendix B.3
Negotiation Form	NegotiationForm.aspx	Appendix B.8
	DBSession.vb	Appendix B.3
	sellAgent.vb	Appendix B.11
	buyAgent.vb	Appendix B.2
	Negotiation.vb	Appendix B.7
Questionnaire	Questionnaire_Page_1.aspx	Appendix B.9
	Questionnaire_Page_2.aspx	Appendix B.10
	DBSession.vb	Appendix B.3
Letter of Appreciation	Thanks_For_Participation.aspx	Appendix B.12
	DBSession.vb	Appendix B.3

We ensure that only authorized participants access the site by requiring participants to authenticate themselves using a unique code provided to them. To prevent multiple entries, the code expires once the study is completed. Although objects and databases cannot be accessed from the external Internet environment, web pages are accessible externally. Thus, they are placed in a secure web area accessible only after authentication.

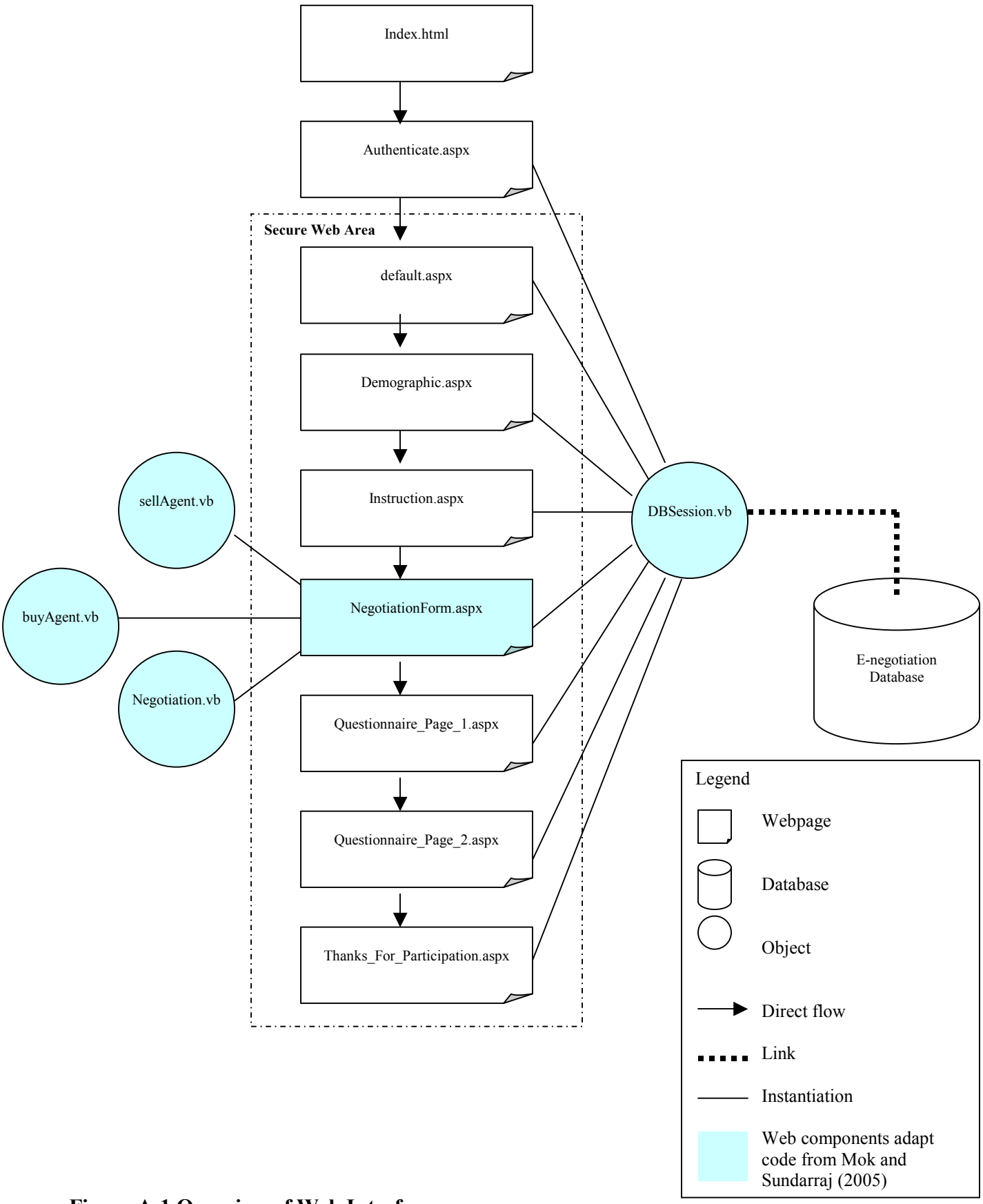


Figure A-1 Overview of Web Interface

A.2 Start Screen

Participants are first exposed to the start screen when they access the site. A screen shot of the start screen is shown in Figure A-2. From the start screen, participants can choose to access the study's web interface or view a power point presentation (Appendix C). On the start screen some information regarding the study are also posted for the participants' benefit.

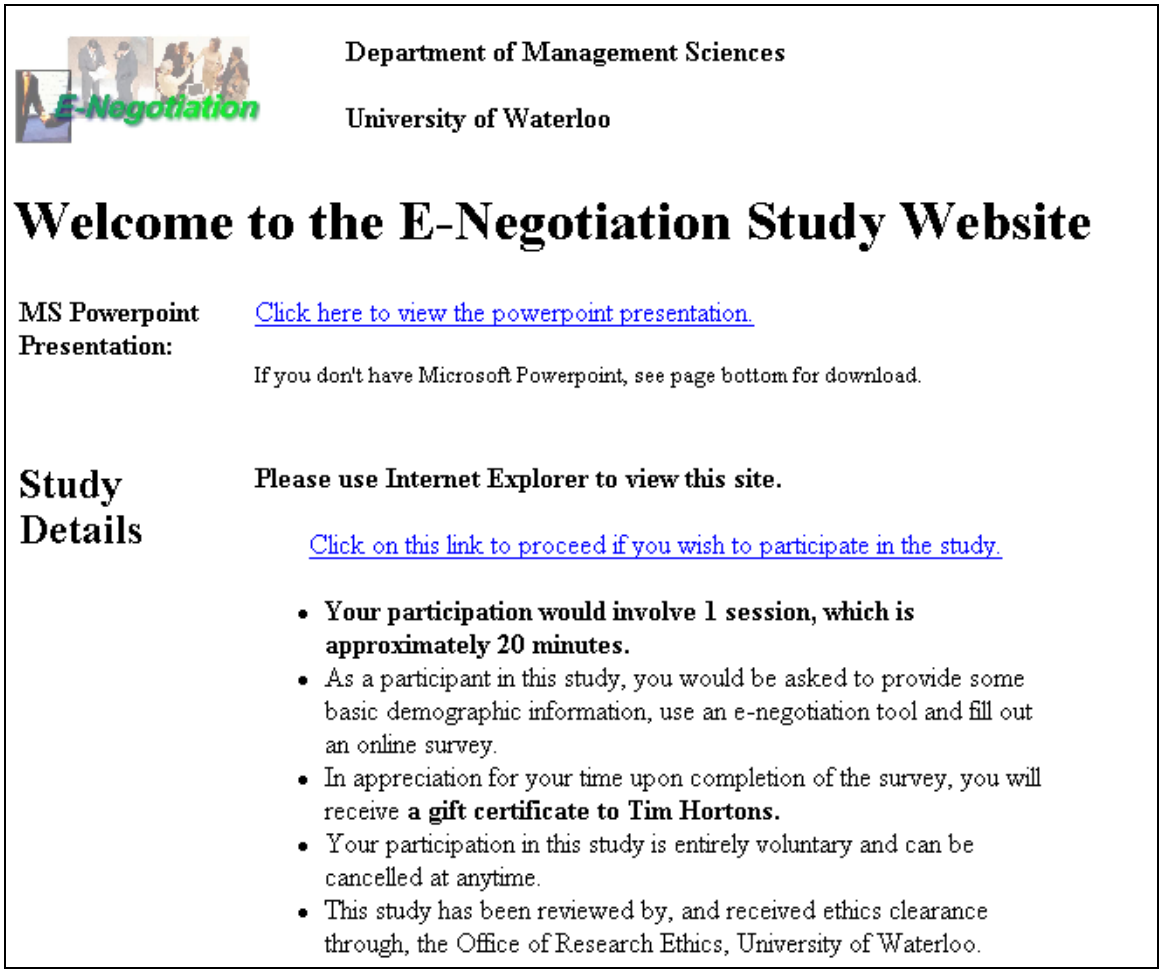


Figure A-2 Web site start screen.

A.3 Introduction

The introduction provides the participants with pertinent information with regards to the survey. Figure A-3 presents the information displayed to the participants.

Title of Project: Theoretical framework for predicting adoption of e-negotiation.
Faculty Investigator: Dr. R. P. Sundarraj, Department of Management Sciences,rsundarr@engmail.uwaterloo.ca or (519) 888-4567 Ext. 2235.

Student Investigator: Vadi Visuvalingam, Department of Management Sciences, v2visuva@engmail.uwaterloo.ca or (519) 888-4567 Ext. 6099.
This study is being conducted by Vadi Visuvalingam as part of his Master's thesis under the supervision of Dr. R. P. Sundarraj, Management Sciences of the University of Waterloo. We are conducting a research study about the factors that would influence individuals to adopt . This study wishes to identify the factors that influence user acceptance of electronic agents. In light of identifying and understanding these factors, we are asking for your participation in the study. To participate in this study, you should be in a position in which you have been, or are currently exposed to the use of new types of technologies. This study will attempt to ascertain your attitudes to this new technology of e-negotiation.

- Your participation is entirely voluntary.
- Your participation in the study will take approximately 20 minutes. Upon completion of this study you will receive a two dollar Tim Hortons gift certificate as a token of our appreciation for your time.
- If you decide to volunteer, you will be asked to complete a basic demographic questionnaire, use an e-negotiation tool and complete an attitudinal survey. The questionnaire will ask basic demographic information such as your age and level of education. The use of e-negotiation tool will require you to engage in a simulated e-negotiation where you play the part of a buyer negotiating with a seller who is represented by an autonomous software agent. The attitudinal survey will request you to rate your level of agreement or disagreement with various statements made in the survey.
- You may decline to answer any questions by selecting the n/a (not applicable) value for the question and you can withdraw your participation at any time.
- All information you provide will be confidential. Furthermore, the web site is programmed to collect responses for demographic questionnaire, e-negotiation results and attitudinal survey alone. That is, the site will not collect any information that could potentially identify you (such as machine identifiers). Additionally, if you begin entering responses to the questionnaire on the Web and then choose not to complete the questionnaire, the information that you have already entered will not be collected for the data. You may withdraw from the study at anytime by advising one of the researchers conducting the survey of this decision.
- Your participation in this study benefits the e-negotiation research community in assessing how well e-negotiation will be adopted.
- If you wish to participate, please press "I agree to participate" and continue. You will be taken to an instructions page which will provide you with instructions on how to begin.
- Thank you for considering to participate.

I would like to assure you that this study has been reviewed and received ethics clearance through the Office of Research Ethics. However, the final decision about participation is yours. If you have any comments or concerns resulting from your participation in this study, please feel free to contact Dr. Susan Sykes, Director, Office of Research Ethics, at (519) 888-4567 ext. 6005 or by email at ssykes@uwaterloo.ca.

The data collected from this study will be accessed only by the two researchers named above and will be maintained on a password-protected computer database in a restricted access area of the university. As well, the data will be electronically archived after completion of the study for possible usage in future studies for a period of 10 years. Should you have any questions about the study, please contact either Vadi Visuvalingam, Department of Management Sciences, v2visuva@engmail.uwaterloo.ca or (519) 888-4567 Ext. 6099 or Dr. R. P. Sundarraj, Department of Management Sciences, rsundarr@engmail.uwaterloo.ca or (519) 888-4567 Ext. 2235. Further, if you would like to receive a copy of the results of this study, please contact either investigator.

Figure A-3 Information presented in the introduction.

Participants need to signal their agreement in order to proceed to the next stage in their study. As shown in Figure A-4 they are presented with two buttons, one that

signals they agree to participate and another that indicates otherwise. This ensured that participants consciously elected to participate in the study.

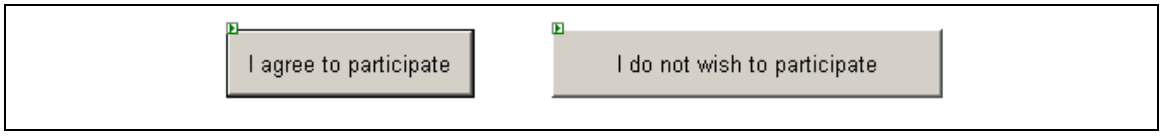


Figure A-4 Buttons that allowed the user to continue or stop the study.

A.4 Demographics

The demographics component requires the participant to enter in information with regards to their background. Response options are provided as radio buttons that participants can click to select their choice. Figure A-5 shows a screen shot of the demographic section.

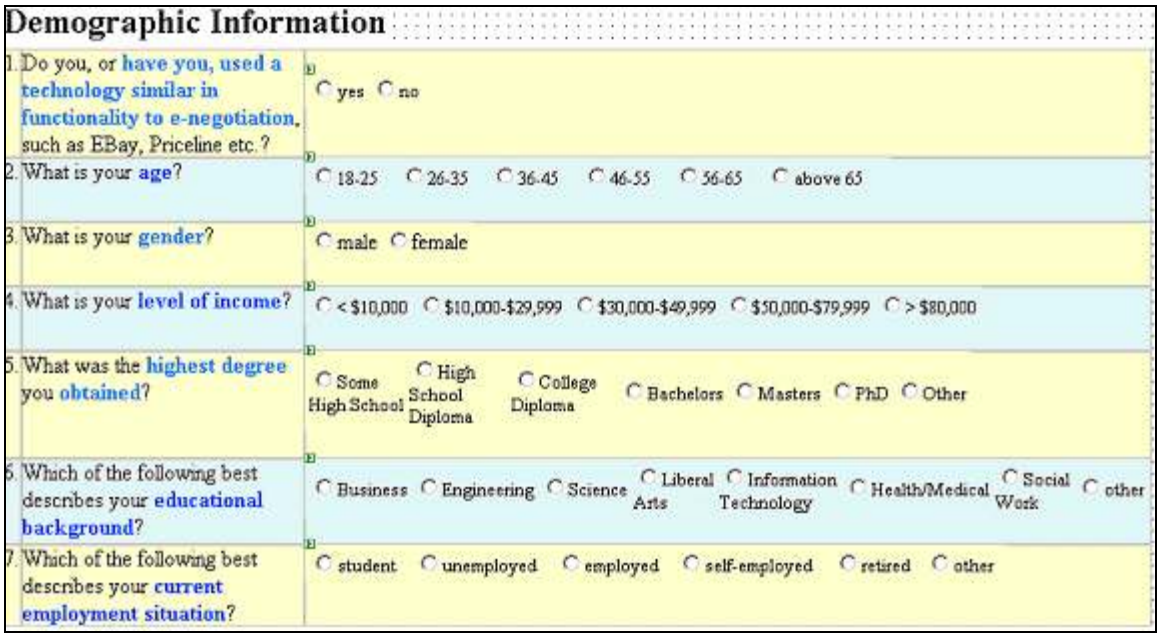


Figure A-5 Screen shot of demographic section.

A.5 Instructions

The instructions provide the participant with information that shows them how to use the negotiation interface. Key points are highlighted so that the users are aware of what steps are required of them. Figure A-6 shows a screen shot of the instructions section.

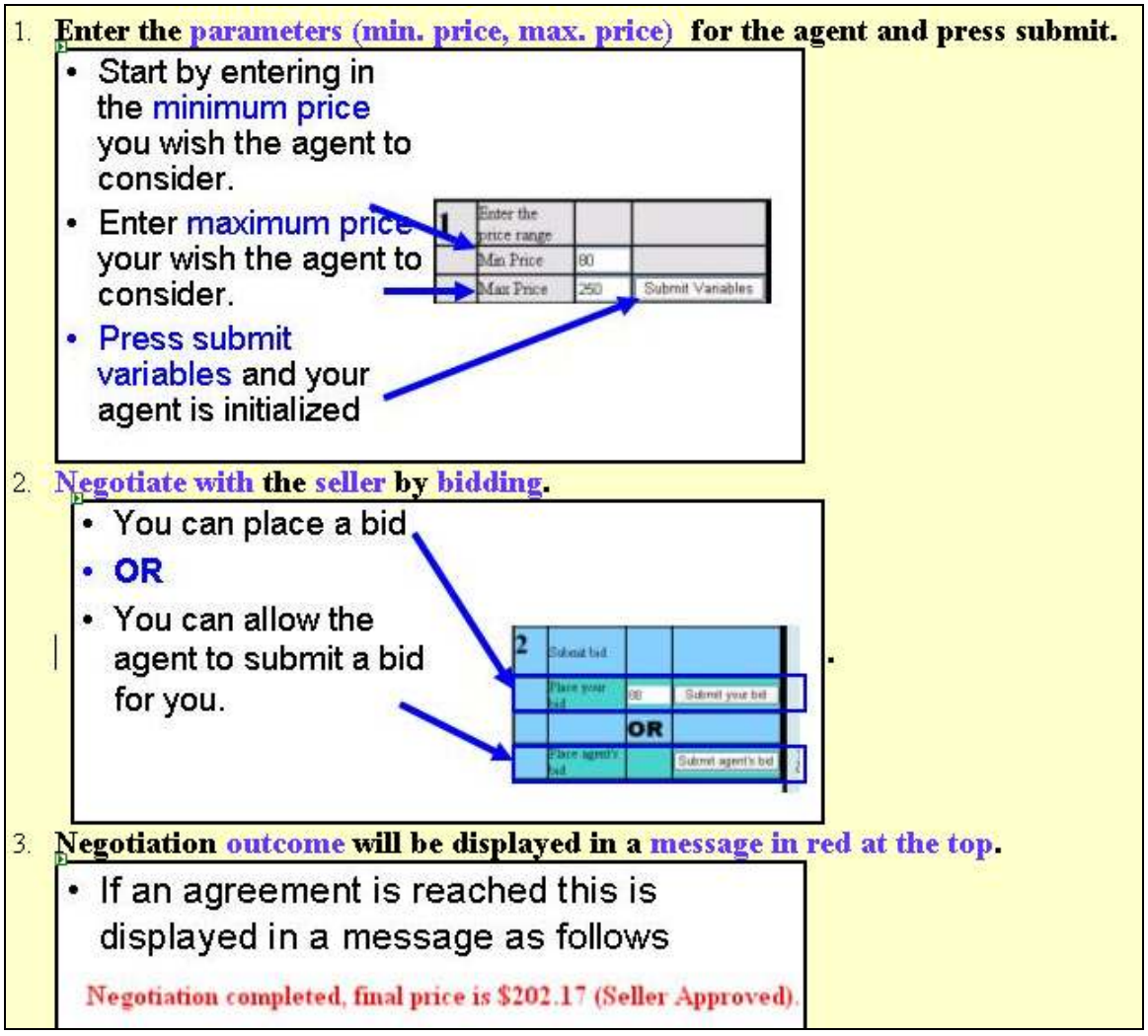


Figure A-6 Screen shot of instructions section.

A.6 Negotiation Form

The negotiation form is the major component of the study where the participants get to interact with the system and engage in e-negotiation. The first step is for the participant to enter their preferences so that the agent can be initialized. Once this has been done the participant engages in negotiation with the agent until an outcome is reached. Figure A-7 shows a screen shot of the negotiation form.

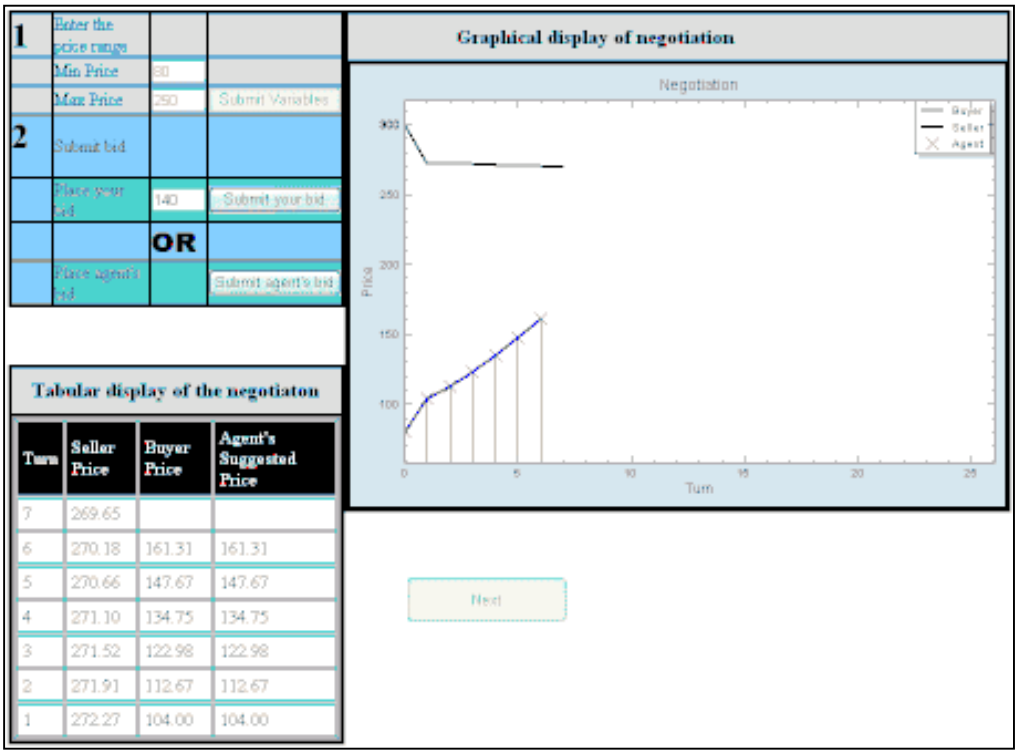


Figure A-7 Screen shot of negotiation form.

In Figure A-8, we show the overall negotiation process in a flowchart. Portions that use algorithms from Mok and Sundarraj (2005) are shaded. For software code, refer to Appendix B.8.

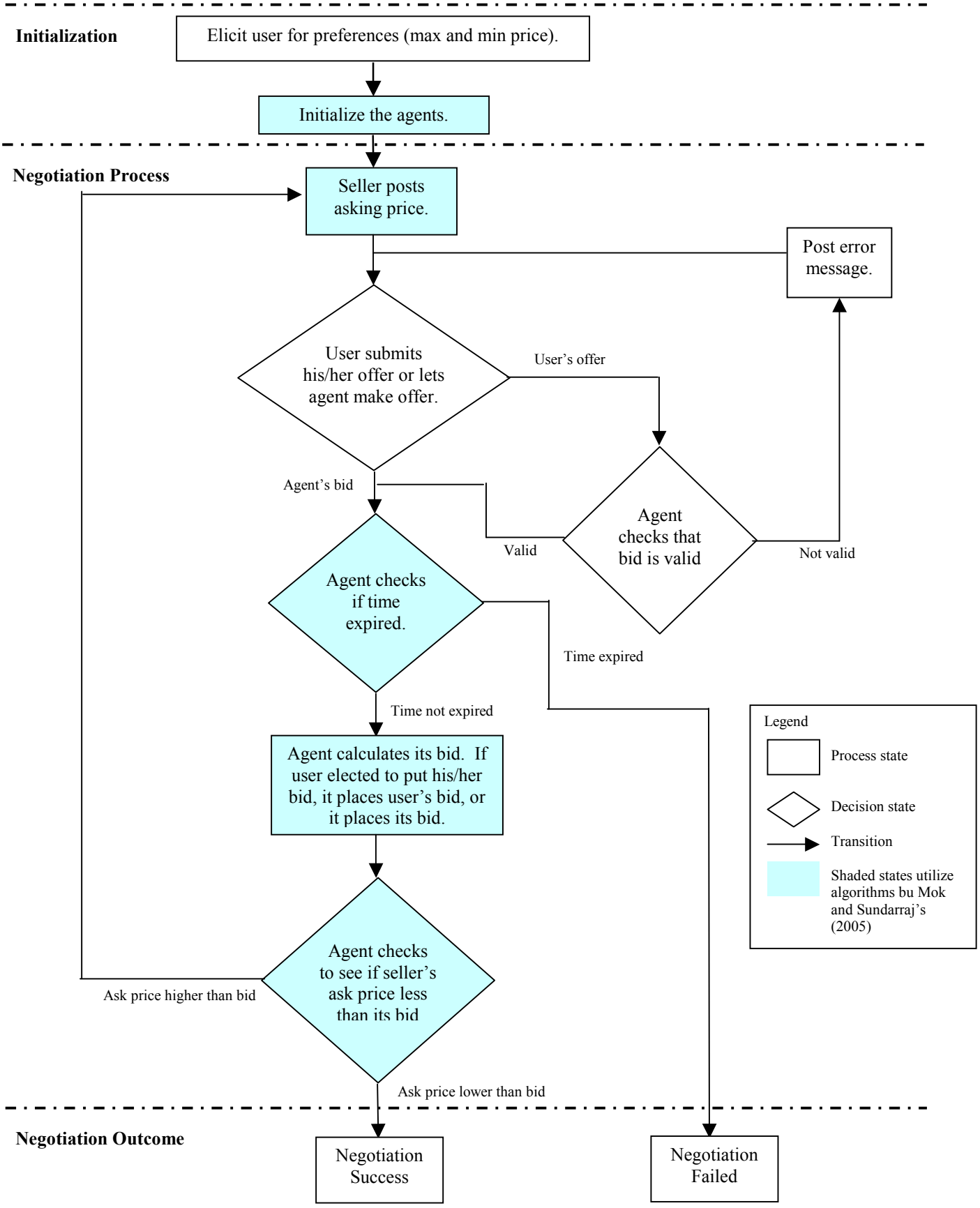
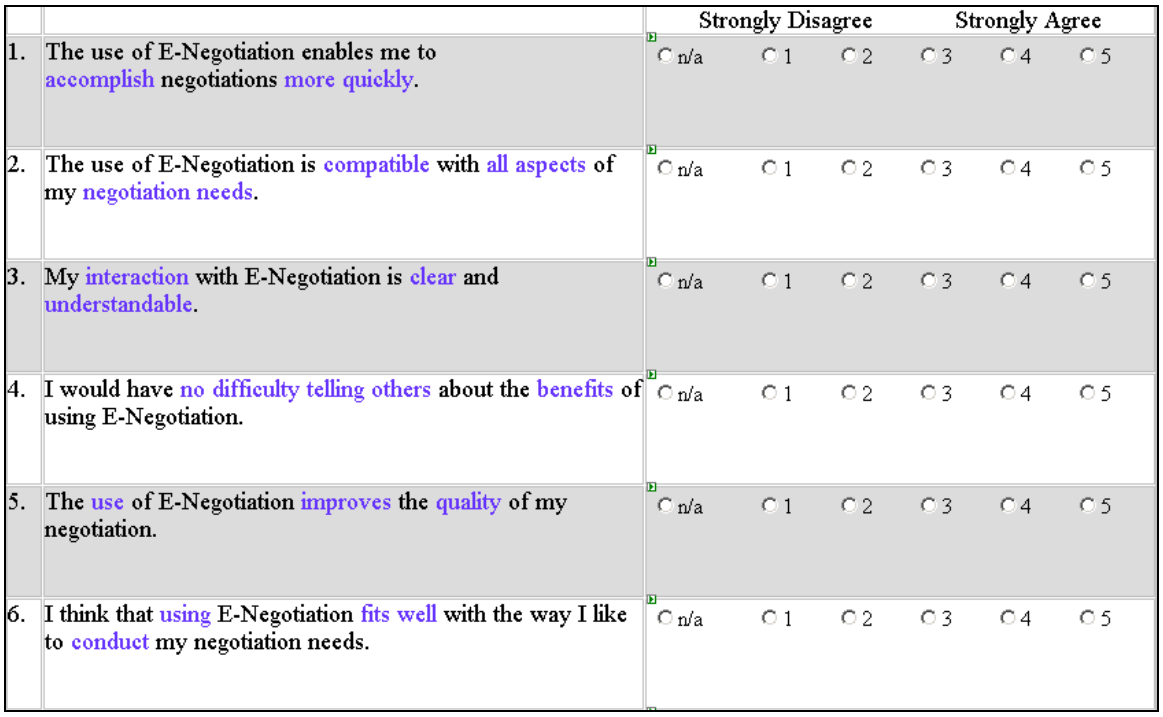


Figure A-8 Negotiation process.

A.7 Questionnaire

The questionnaire asks participants to answer questions that probe their attitudes and perceptions. The response options are provided as radio buttons that users can select to make their choice. Figure A-9 presents a screen shot of the questionnaire and Table A-2 presents the questions that the users are asked. Question 25 is asked of participants in the learning group as a check to ensure they were aware of the "learning" process.



		Strongly Disagree					Strongly Agree
1.	The use of E-Negotiation enables me to accomplish negotiations more quickly.	n/a	1	2	3	4	5
2.	The use of E-Negotiation is compatible with all aspects of my negotiation needs.	n/a	1	2	3	4	5
3.	My interaction with E-Negotiation is clear and understandable.	n/a	1	2	3	4	5
4.	I would have no difficulty telling others about the benefits of using E-Negotiation.	n/a	1	2	3	4	5
5.	The use of E-Negotiation improves the quality of my negotiation.	n/a	1	2	3	4	5
6.	I think that using E-Negotiation fits well with the way I like to conduct my negotiation needs.	n/a	1	2	3	4	5

Figure A-9 Screen shot of questionnaire.

Table A-2 Questionnaire items

Construct and Question Item	Question	Source	Additional areas used.
Relative Advantage (REL)			
REL_1	The use of E-Negotiation enables me to accomplish negotiations more quickly.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
REL_2	The use of E-Negotiation improves the quality of my negotiation.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996;

			Karahanna et al., 1999)
REL_3	The use of E-Negotiation makes it easier to perform negotiation.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
REL_4	Using E-Negotiation enhances one's effectiveness in negotiation.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
REL_5	Using E-Negotiation gives me greater control over my negotiation.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
Compatibility (COM)			
COM_1	The use of E-Negotiation is compatible with all aspects of my negotiation needs.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
COM_2	I think that using E-Negotiation fits well with the way I like to conduct my negotiation needs.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
COM_3	The use of E-Negotiation fits into my negotiation behaviour.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
Results demonstrability (RSLT)			
RSLT_1	I would have no difficulty telling others about the benefits of using E-Negotiation.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
RSLT_2	I believe I could communicate to others the benefits of using E-Negotiation.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
RSLT_3	The benefits of using E-Negotiation are apparent to me.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
RSLT_4	I would have difficulty explaining why using E-Negotiation may or may not be beneficial.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)

Attitude (ATT)			
ATT_1	It would be better for me to use E-Negotiation.	(Mathieson, 1991)	(Benham & Raymond, 1996)
ATT_2	I think it is better to be using E-Negotiation than to not be using it.	(Mathieson, 1991)	(Benham & Raymond, 1996)
ATT_3	In my opinion it is better to be using E-Negotiation.	(Mathieson, 1991)	(Benham & Raymond, 1996)
Ease of Use (EOU)			
EOU_1	My interaction with E-Negotiation is clear and understandable.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
EOU_2	I believe that it is easy to get E-Negotiation to do what I want it to do.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
EOU_3	Overall, I believe that E-Negotiation is easy to use.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
EOU_4	Learning to use E-Negotiation is easy.	(Moore & Benbasat, 1991)	(Benham & Raymond, 1996; Karahanna et al., 1999)
Perceived Behavioural Control (PBC)			
PBC_1	From the resources, opportunities, and knowledge available for E-Negotiation, I am able to use E-Negotiation effectively.	(Mathieson, 1991)	(Benham & Raymond, 1996)
PBC_2	I am able to use E-Negotiation effectively because I either have the necessary skills or I have access to the training that can provide me the necessary skills.	(Mathieson, 1991)	(Benham & Raymond, 1996)
PBC_3	I would have more flexibility and control in my negotiation if I used E-Negotiation.	(Mathieson, 1991)	(Benham & Raymond, 1996)
Behavioural Intention (INT)			
INT_1	Given that I may have access to E-Negotiation in the future, I predict that I would use it.	(Davis, 1989)	(Benham & Raymond, 1996; Davis et al., 1989; Venkatesh & Davis, 2000)
INT_2	Assuming I will have access to E-	(Davis, 1989)	(Benham &

	Negotiation in the future, I intend to use it.		Raymond, 1996; Davis et al., 1989; Venkatesh & Davis, 2000)

A.8 Letter of Appreciation

The letter of appreciation, shown in Figure A-10, provided the participants with some feedback and expressed gratitude for their partaking in the study. At this stage, the authentication code also expires and the participants are provided with a confirmation code. We required the confirmation code from them to receive their Tim Hortons gift certificate.

Thank You! Your confirmation code is <confirmation code>.
Thank you for your participation in this study and completion of this survey.

If you are interested in viewing the results of this survey, they will be posted on Jan 25, 2006 at http://enegotiation.uwaterloo.ca.

If you have any general comments or questions related to this study please contact Vadi Visuvalingam, Department of Management Sciences at v2visuva@engmail.uwaterloo.ca or (519) 888-4567 Ext. 6099 or Dr. Sundarraj, Department of Management Sciences at rsundarr@engmail.uwaterloo.ca or (519) 888-4567 Ext. 2235.

We would like to assure you that this study has been reviewed by, and has received ethics clearance through the Office of Research Ethics at the University of Waterloo. If you have any concerns regarding your participation in this study, please contact Dr. Susan Sykes, Director, Office of Research Ethics at ssykes@uwaterloo.ca or (519) 888-4567 Ext. 6005.

Thank you.

Please close this window.

Figure A-10 Thank you letter to participants.

Appendix B: Software Code

B.1 Authenticate

```
Public Class authenticate
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents txtUser As System.Web.UI.WebControls.TextBox
    Protected WithEvents litMsg As System.Web.UI.WebControls.Literal

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'Put user code to initialize the page here
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        'If FormsAuthentication.Authenticate(txtUser.Text, "test") Or
AuthenticateDB(txtUser.Text) Then
        If AuthenticateDB(txtUser.Text) Then
            FormsAuthentication.RedirectFromLoginPage(txtUser.Text, False)
        Else
            'ErrorMessage.InnerHtml = "<b>Something went wrong...</b> please re-enter
your credentials..."
            litMsg.Text = "Invalid Authenticate code."
        End If

    End Sub

    Function AuthenticateDB(ByVal strLogin As String) As Boolean
        Dim lSession As DbSession = New DbSession

        Dim strConn As String = lSession.connectionString

        Dim ocmd As SqlCommand
        Dim odtr As SqlDataReader
        Dim compare As String
        Dim strCmdText = "SELECT COMPLETED FROM UNIQUEVALTABLE WHERE USERID = '" &
strLogin & "' "

        Try
            ocmd = New SqlCommand
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
```

```
                .CommandText = strCmdText
                odtr = .ExecuteReader
            End With

            'Dim numrows As Integer = ocmd.ExecuteNonQuery()
            odtr.Read()
            compare = odtr.GetValue(0)

        Catch oexpData As Exception
            Return False

        End Try



        'odtrAboutVBDataReader.Close()
        ocmd.Connection.Close()
        ocmd.Dispose()

        Try
            If compare = "False" Then
                Return True
            Else
                Return False
            End If
        Catch oexpData2 As Exception
            Return False

        End Try
    End Function
End Class
```

B.2 BuyAgent

(Note: This code has been adapted from the work by Mok and Sundarraj (2005))

```
Option Explicit On

Public Class buyAgent

    Private m_NegotiationID As String
    Private m_id As String

    Private m_startPrice As Double
    Private m_maxPrice As Double
    'Private m_BMstartPrice As Double
    'Private m_BMmaxPrice As Double
    'Private m_initThreshold As Single
    'Private m_finalThreshold As Single
    'Private m_priceScore As Single
    'Private m_BMpriceScore As Single
    'Private m_strategy As String
    Private m_bidPrice As Double
    Private m_beta As Single
    Private m_Tmax As Integer
    Private m_cumTmax As Integer
    Private m_K As Single
    Private m_estimatedBeta As Single
    Private m_estimatedTmax As Integer
    Private m_estimatedPmin As Double
    Private m_estimatedPmax As Double
    Private m_estimatedK As Single
    Private m_learn As Boolean
    Private m_learnStrategy As String
    Private m_TmaxStrategy As String
    'Private m_decisionScore() As Variant
    'Private m_propertyScore() As Variant

    Public gSession As DbSession

    Public Sub New(ByVal a_NegotiationID As String)
        m_NegotiationID = a_NegotiationID
        gSession = New DbSession
        gSession.NegotiationID = m_NegotiationID
    End Sub

    Public Property NegotiationID() As String
        Get
            Return m_NegotiationID
        End Get
        Set(ByVal Value As String)
            m_NegotiationID = Value
        End Set
    End Property
    Public Property ID() As String
        Get
            Return m_id
        End Get
        Set(ByVal x As String)
            m_id = x
        End Set

    End Property

    Public Property learn() As Boolean
        Get
            Return m_learn
        End Get
        Set(ByVal x As Boolean)
            m_learn = x
        End Set
```

```
End Property

Public Property learnStrategy() As String
    Get
        Return m_learnStrategy
    End Get
    Set(ByVal x As String)
        m_learnStrategy = x
    End Set

End Property

Public Property TmaxStrategy() As String
    Get
        Return m_TmaxStrategy
    End Get
    Set(ByVal x As String)
        m_TmaxStrategy = x
    End Set

End Property

Public Property estimatedBeta() As Single
    Get
        Return m_estimatedBeta
    End Get
    Set(ByVal x As Single)
        m_estimatedBeta = x
    End Set

End Property

Public Property estimatedTmax() As Integer
    Get
        Return m_estimatedTmax
    End Get
    Set(ByVal x As Integer)
        m_estimatedTmax = x
    End Set

End Property

Public Property estimatedPmin() As Double
    Get
        Return m_estimatedPmin
    End Get
    Set(ByVal x As Double)
        m_estimatedPmin = x
    End Set

End Property

Public Property estimatedPmax() As Double
    Get
        Return m_estimatedPmax
    End Get
    Set(ByVal x As Double)
        m_estimatedPmax = x
    End Set

End Property

Public Property estimatedK() As Single
    Get
        Return m_estimatedK
    End Get
    Set(ByVal x As Single)
        m_estimatedK = x
    End Set

End Property
```

```
Public Property startPrice() As Double
    Get
        Return m_startPrice
    End Get
    Set(ByVal x As Double)
        m_startPrice = x
    End Set

End Property

Public Property maxPrice() As Double
    Get
        Return m_maxPrice
    End Get
    Set(ByVal x As Double)
        m_maxPrice = x
    End Set

End Property
'Public Property Get BMstartPrice() As Double
'    BMstartPrice = m_BMstartPrice
'End Property
'Public Property Let BMstartPrice(x As Double)
'    m_BMstartPrice = x
'End Property
'Public Property Get BMmaxPrice() As Double
'    BMmaxPrice = m_BMmaxPrice
'End Property
'Public Property Let BMmaxPrice(x As Double)
'    m_BMmaxPrice = x
'End Property
'Public Property Get InitThreshold() As Single
'    InitThreshold = m_initThreshold
'End Property
'Public Property Let InitThreshold(x As Single)
'    m_initThreshold = x
'End Property
'Public Property Get finalThreshold() As Single
'    finalThreshold = m_finalThreshold
'End Property
'Public Property Let finalThreshold(x As Single)
'    m_finalThreshold = x
'End Property
'Public Property Get decisionScore() As Variant
'    decisionScore = m_decisionScore
'End Property
'Public Property Let decisionScore(x As Variant)
'    m_decisionScore = x
'End Property
'Public Property Get priceScore() As Single
'    priceScore = m_priceScore
'End Property
'Public Property Let priceScore(x As Single)
'    m_priceScore = x
'End Property
'Public Property Get strategy() As String
'    strategy = m_strategy
'End Property
'Public Property Let strategy(x As String)
'    m_strategy = x
'End Property

Public Property bidPrice() As Double
    Get
        Return m_bidPrice
    End Get
    Set(ByVal x As Double)
        m_bidPrice = x
    End Set
```

```
    End Property

    'Public Property Get propertyScore() As Variant
    '    propertyScore = m_propertyScore
    'End Property
    'Public Property Let propertyScore(x As Variant)
    '    m_propertyScore = x
    'End Property
    'Public Property Get BMpriceScore() As Single
    '    BMpriceScore = m_BMpriceScore
    'End Property
    'Public Property Let BMpriceScore(x As Single)
    '    m_BMpriceScore = x
    'End Property

    Public Property beta() As Single
        Get
            Return m_beta
        End Get
        Set(ByVal x As Single)
            m_beta = x
        End Set

    End Property

    Public Property Tmax() As Integer
        Get
            Return m_Tmax
        End Get
        Set(ByVal x As Integer)
            m_Tmax = x
        End Set

    End Property

    Public Property cumTmax() As Integer
        Get
            Return m_cumTmax
        End Get
        Set(ByVal x As Integer)
            m_cumTmax = x
        End Set

    End Property

    Public Property K() As Single
        Get
            Return m_K
        End Get
        Set(ByVal x As Single)
            m_K = x
        End Set

    End Property

    Public Function Estimate(ByVal neg As Negotiation, ByVal priceArray() As Double) As
Boolean
        Dim two, three, four As Boolean
        Dim t0, t1, t2, t3, t4 As Integer
        Dim P0, P1, P2, P3, P4 As Double
        Dim beta As Single
        Dim Pmin As Double
        Dim Pmax As Double
        Dim Tmax As Integer
        Dim K As Single
        Dim cumBeta As Double
        Dim cumTmax As Integer
        Dim cumPmin As Double
        Dim cumPmax As Double
        Dim cumK As Double
        Dim sSQL As String
```

```
        Dim temp As Single
        Dim cnt As Integer
        'Dim conn As New ADODB.Connection
        'Dim rs As New ADODB.Recordset
        'conn = gSession.GetConn

        t0 = 0
        t1 = 1
        t2 = 2
        t3 = 3
        t4 = 4
        P0 = priceArray(0)
        P1 = priceArray(1)
        P2 = priceArray(2)
        P3 = priceArray(3)
        P4 = priceArray(4)

        sSQL = "SELECT * FROM Criteria ORDER BY ID"
        'rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
        'rs.MoveFirst()
        Dim strConn As String = gSession.connectionString

        Dim ocmd As SqlCommand

        Try
            ocmd = New SqlCommand
            Dim odtr As SqlDataReader
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = sSQL
                odtr = .ExecuteReader()
            End With

            Dim rs_criteria As Integer

            Do Until cnt >= 20 Or Not odtr.Read 'rs.EOF
                cnt = 0
                cumBeta = 0
                cumTmax = 0
                cumPmin = 0
                cumPmax = 0
                cumK = 0
                rs_criteria = odtr.GetValue(1)

                For Tmax = (t4 + 1) To 25 Step 1
                    For Pmax = Round(P0 + 1, 0) To 300 Step 2
                        For Pmin = 1 To Round(P4, 0) Step 2
                            K = 1 - ((P0 - Pmin) / (Pmax - Pmin))
                            beta = (Log(Log(1 - ((P1 - Pmin) / (Pmax - Pmin))) / Log(K))
/ Log(10)) / (Log(1 - t1 / Tmax) / Log(10))
                            K = Round(K, 3)
                            If K = 0 Then GoTo Nxt
                            beta = Round(beta, 3)

                            temp = Pmin + (1 - (Exp(((1 - t2 / Tmax) ^ beta) * Log(K))))
* (Pmax - Pmin)
                            If (Abs(temp - P2)) < rs_criteria Then
                                two = True

                            Else
                                two = False
                                GoTo Nxt
                            End If
                            temp = Pmin + (1 - (Exp(((1 - t3 / Tmax) ^ beta) * Log(K))))
* (Pmax - Pmin)
                            If (Abs(temp - P3)) < rs_criteria Then
                                three = True

                            Else
```

```
                                three = False
                                GoTo Nxt
                            End If
                            temp = Pmin + (1 - (Exp(((1 - t4 / Tmax) ^ beta) * Log(K))))
* (Pmax - Pmin)
                            If (Abs(temp - P4)) < rs_criteria Then
                                four = True

                            Else
                                four = False
                                GoTo Nxt
                            End If

                            If two = True And three = True And four = True Then
                                cnt = cnt + 1
                                cumBeta = cumBeta + beta
                                cumTmax = cumTmax + Tmax
                                cumPmin = cumPmin + Pmin
                                cumPmax = cumPmax + Pmax
                                cumK = cumK + K
                            End If
Nxt:                    Next Pmin
                    Next Pmax
                Next Tmax
                'rs.MoveNext()
            Loop

        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()

        If cnt >= 20 Then
            m_estimatedBeta = Round(cumBeta / cnt, 3)
            m_estimatedTmax = Round(cumTmax / cnt)
            m_estimatedPmin = Round(cumPmin / cnt, 2)
            m_estimatedPmax = Round(cumPmax / cnt, 2)
            m_estimatedK = Round(cumK / cnt, 3)
            Estimate = True

            Dim t As Integer
            Dim PredictedPrice As Double
            sSQL = "DELETE FROM PredictedSeller WHERE turn >= " & (neg.cumBuyerTurn - 4)
& " AND NegotiationID = '" & m_NegotiationID & "' "
            gSession.ExecuteNonQuery(sSQL)
            For t = 0 To m_estimatedTmax Step 1
                PredictedPrice = m_estimatedPmin + (1 - Exp(((1 - t / m_estimatedTmax) ^
m_estimatedBeta) * Log(m_estimatedK))) * (m_estimatedPmax - m_estimatedPmin)
                sSQL = "INSERT INTO PredictedSeller( NegotiationID, turn, Price ) VALUES(
'" & m_NegotiationID & "', " & (t + neg.cumBuyerTurn - 4) & ", " & PredictedPrice & ")"
                gSession.ExecuteNonQuery(sSQL)
            Next t
        Else
            Estimate = False
        End If

    End Function
    Public Sub FindOptimal(ByRef neg As Negotiation)
        Dim Tmax As Integer
        Dim turn As Integer
        Dim beta As Single
        Dim target As Double
        Dim newStartPrice As Double
        Dim i As Integer
        Dim t As Integer
        For i = m_estimatedTmax To 5 Step -1
            target = m_estimatedPmin + (1 - (Exp(((1 - i / estimatedTmax) ^
m_estimatedBeta) * Log(m_estimatedK)))) * (m_estimatedPmax - m_estimatedPmin)
```

```
            If (neg.nextBuyerLearn - neg.lastBuyerLearn) = 5 Then
                t = i + 1
            ElseIf (neg.nextBuyerLearn - neg.lastBuyerLearn) = 4 Then
                t = i
            End If
            If m_TmaxStrategy = "Fix" Then
                Tmax = m_Tmax
            ElseIf m_TmaxStrategy = "UnFix" Or m_TmaxStrategy = "Unfix" Then
                If m_Tmax <= m_estimatedTmax Then
                    Tmax = t + 1
                Else
                    Tmax = m_Tmax
                End If
            End If

            If t = Tmax And m_maxPrice = target Then
                Exit Sub
            End If

            If target > m_maxPrice Or t >= Tmax Then
                GoTo Nx
            Else
                If (neg.nextBuyerLearn - neg.lastBuyerLearn) = 5 Then
                    Tmax = Tmax - 5
                    turn = t - 5
                ElseIf (neg.nextBuyerLearn - neg.lastBuyerLearn) = 4 Then
                    Tmax = Tmax - 4
                    turn = t - 4
                End If
                newStartPrice = m_bidPrice
                'newStartPrice = (m_bidPrice - m_K * m_maxPrice) / (1 - m_K)
                If target - newStartPrice <= 0 Then GoTo Nx
                beta = (Log((Log((target - newStartPrice) / (m_maxPrice -
newStartPrice))) / Log(m_K)) / Log(10)) / (Log(1 - turn / Tmax) / Log(10))
                If beta > 0 Then
                    m_startPrice = newStartPrice
                    m_cumTmax = Tmax + neg.cumBuyerTurn
                    neg.buyerTurn = 0
                    m_beta = beta
                    m_Tmax = Tmax
                    If m_learnStrategy = "Subsequent" Then
                        If (neg.nextBuyerLearn - neg.lastBuyerLearn) = 5 Then
                            neg.nextBuyerLearn = neg.cumBuyerTurn + 4
                        ElseIf (neg.nextBuyerLearn - neg.lastBuyerLearn) = 4 Then
                            neg.nextBuyerLearn = neg.cumBuyerTurn + 5
                        End If
                    End If
                    neg.lastBuyerLearn = neg.cumBuyerTurn
                    Exit Sub
                End If
            End If

Nx:     Next i
        If m_learnStrategy = "Subsequent" Then
            If (neg.nextBuyerLearn - neg.lastBuyerLearn) = 5 Then
                neg.nextBuyerLearn = neg.cumBuyerTurn + 4
            ElseIf (neg.nextBuyerLearn - neg.lastBuyerLearn) = 4 Then
                neg.nextBuyerLearn = neg.cumBuyerTurn + 5
            End If
        End If
        neg.lastBuyerLearn = neg.cumBuyerTurn
    End Sub

End Class
```

B.3 DBSession

(Note: This code has been adapted from the work by Mok and Sundarraj (2005))

```
Public Class DbSession
    Public Const connectionString = "server=(local); database=eTestDB; User
Id=eTestWebUser; Password=waterloo"
    Private Const csDsnName = "DSN=VB01"

    Public NegotiationID As String
    'Public Function GetConn() As ADODB.Connection
    '    Dim conn As New ADODB.Connection

    '    On Error GoTo GetConn_Fail
    '    conn.Open(csDsnName)
    '    GetConn = conn
    '    Exit Function

    'GetConn_Fail:
    '        MsgBox(Err.Description, vbCritical, Err.Source)
    '    End Function
    '    Public Function GetSellerPrice() As ADODB.Recordset
    '        Dim conn As ADODB.Connection
    '        Dim rs As New ADODB.Recordset
    '        Dim sSQL As String

    '        conn = GetConn
    '        sSQL = "SELECT CurrentAsk FROM Negotiation WHERE Sender = 'Seller' ORDER BY
SellerTurn"
    '        rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
    '        GetSellerPrice = rs
    '    End Function
    '    Public Function GetBuyerPrice() As ADODB.Recordset
    '        Dim conn As ADODB.Connection
    '        Dim rs As New ADODB.Recordset
    '        Dim sSQL As String

    '        conn = GetConn
    '        sSQL = "SELECT CurrentBid FROM Negotiation WHERE Sender = 'Buyer' ORDER BY
BuyerTurn"
    '        '        rs.Open(sSQL, conn, adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
    '        GetBuyerPrice = rs
    '    End Function
    Public Function Remove()
        'Dim conn As ADODB.Connection
        Dim sSQL As String


        'conn = GetConn
        sSQL = "DELETE FROM Negotiation WHERE NegotiationID = '" & NegotiationID & "' "
        Me.ExecuteNonQuery(sSQL)
        sSQL = "DELETE FROM PredictedSeller WHERE NegotiationID = '" & NegotiationID & "'
"
        Me.ExecuteNonQuery(sSQL)
        sSQL = "DELETE FROM PredictedBuyer WHERE NegotiationID = '" & NegotiationID & "'
"
        Me.ExecuteNonQuery(sSQL)
        sSQL = "DELETE FROM FitLine WHERE NegotiationID = '" & NegotiationID & "' "
        Me.ExecuteNonQuery(sSQL)

        sSQL = "DELETE FROM Price_Seller WHERE NegotiationID = '" & NegotiationID & "' "
        Me.ExecuteNonQuery(sSQL)
        sSQL = "DELETE FROM Price_Buyer WHERE NegotiationID = '" & NegotiationID & "' "
        Me.ExecuteNonQuery(sSQL)
        sSQL = "DELETE FROM Suggested_Bid WHERE NegotiationID = '" & NegotiationID & "' "
        Me.ExecuteNonQuery(sSQL)
```

```
    End Function
    Public Function GetDeal(ByRef final As Double, ByRef turn As Integer, ByRef whom As
String) As Boolean
        'Dim conn As ADODB.Connection
        'Dim rs As New ADODB.Recordset
        Dim sSQL As String
        Dim rs_state As String
        Dim rs_finalPrice As String
        Dim rs_buyerTurn As String
        Dim rs_SellerTurn As String
        'conn = GetConn()


        sSQL = "SELECT ID, Sender, Receiver, BuyerTurn, SellerTurn, CurrentBid,
CurrentAsk, LastBid, LastAsk, FinalPrice, State FROM Negotiation WHERE state = 'Failed'
or state = 'Buyer Approved' or state = 'Seller Approved' and NegotiationID = '" &
NegotiationID & "' "
        Dim strConn As String = Me.connectionString

        Dim ocmd As SqlCommand
        Dim odtr As SqlDataReader
        Try
            ocmd = New SqlCommand
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = sSQL
                odtr = .ExecuteReader()
            End With

            odtr.Read()
            If Not odtr.IsDBNull(10) Then
                rs_state = odtr.GetValue(10)
            Else
                rs_state = ""
            End If
            If Not odtr.IsDBNull(9) Then
                rs_finalPrice = odtr.GetValue(9)
            Else
                rs_finalPrice = ""
            End If
            If Not odtr.IsDBNull(3) Then
                rs_buyerTurn = odtr.GetValue(3)
            Else
                rs_buyerTurn = ""
            End If

            If Not odtr.IsDBNull(4) Then
                rs_SellerTurn = odtr.GetValue(4)
            Else
                rs_SellerTurn = ""
            End If



        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()
        'rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
        If rs_state = "Failed" Then
            GetDeal = False
        Else
            GetDeal = True
            final = rs_finalPrice
```

```
            If rs_state = "Buyer Approved" Then
                whom = "Buyer"
                turn = rs_buyerTurn
            Else
                whom = "Seller"
                turn = rs_SellerTurn
            End If
        End If
    End Function
    Public Function ListBuyerPrice(ByVal neg As Negotiation) As Object
        'Dim conn As New ADODB.Connection
        'Dim rs As New ADODB.Recordset
        Dim sSQL As String
        Dim priceArray(4) As Double
        Dim i As Integer
        'conn = GetConn()

        i = 0
        sSQL = "SELECT CurrentBid FROM Negotiation WHERE state = 'Wait for Seller' AND
BuyerTurn BETWEEN " & neg.cumSellerTurn - 5 & " AND " & neg.cumSellerTurn - 1 & " and
NegotiationID = '" & NegotiationID & "' "
        'rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
        'rs.MoveFirst()

        'Do Until rs.EOF
        '    priceArray(i) = rs!currentBid
        '    i = i + 1
        '    rs.MoveNext()
        'Loop
        'ListBuyerPrice = priceArray
        Dim strConn As String = Me.connectionString

        Dim ocmd As SqlCommand

        Try
            ocmd = New SqlCommand
            Dim odtr As SqlDataReader
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = sSQL
                odtr = .ExecuteReader()
            End With

            Do While odtr.Read
                priceArray(i) = odtr.GetValue(0)
                i = i + 1
            Loop

            ListBuyerPrice = priceArray
        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()
    End Function
    Public Function ListSellerPrice(ByVal neg As Negotiation) As Object
        'Dim conn As New ADODB.Connection
        'Dim rs As New ADODB.Recordset
        Dim sSQL As String
        Dim priceArray(4) As Double
        Dim i As Integer
        'conn = GetConn()

        i = 0
        sSQL = "SELECT CurrentAsk FROM Negotiation WHERE state = 'Wait for Buyer' AND
SellerTurn BETWEEN " & neg.cumBuyerTurn - 4 & " AND " & neg.cumBuyerTurn & " and
NegotiationID = '" & NegotiationID & "' ORDER BY CurrentAsk DESC"
```

```
        'rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)

        'rs.MoveFirst()

        'Do Until rs.EOF
        '    priceArray(i) = rs.!currentAsk
        '    i = i + 1
        '    rs.MoveNext()
        'Loop
        'ListSellerPrice = priceArray

        Dim strConn As String = Me.connectionString

        Dim ocmd As SqlCommand

        Try
            ocmd = New SqlCommand
            Dim odtr As SqlDataReader
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = sSQL
                odtr = .ExecuteReader()
            End With

            Do While odtr.Read
                priceArray(i) = odtr.GetValue(0)
                i = i + 1
            Loop

            ListSellerPrice = priceArray
        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()
    End Function

    Function ExecuteNonQuery(ByVal strCmdText As String)
        Dim strConn As String = Me.connectionString

        Dim ocmd As SqlCommand
        Dim odtr As SqlDataReader

        Try
            ocmd = New SqlCommand
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = strCmdText
                odtr = .ExecuteReader
            End With

            'Dim numrows As Integer = ocmd.ExecuteNonQuery()
            odtr.Read()

        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        'odtrAboutVBDataReader.Close()
        ocmd.Connection.Close()
        ocmd.Dispose()
    End Function

End Class
```

B.4 Default

```
Public Class _default
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents btnNext As System.Web.UI.WebControls.Button
    Protected WithEvents ExitSurvey As System.Web.UI.WebControls.Button
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    End Sub

    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNext.Click
        Dim localhost As String = ConfigurationSettings.AppSettings.Get("URL")
        Response.Redirect(localhost + "Demographic.aspx")
    End Sub

    Private Sub ExitSurvey_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitSurvey.Click
        FormsAuthentication.SignOut()
        Response.Redirect("http://www.uwaterloo.ca")
    End Sub
End Class
```

B.5 Demographic

```
Public Class Demographic
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents QL3 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL4 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL5 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL6 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL7 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents btnNext As System.Web.UI.WebControls.Button
    Protected WithEvents litMsgBox As System.Web.UI.WebControls.Literal
    Protected WithEvents QL2 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL8 As System.Web.UI.WebControls.RadioButtonList

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        gSession = New DbSession

    End Sub

    Dim gSession As DbSession
    Dim QS1 As String
    Dim QS2 As String
    Dim QS3 As String
    Dim QS4 As String
    Dim QS5 As String
    Dim QS6 As String
    Dim QS7 As String
    Dim QS8 As String

    Private Function CheckForNullValues() As Boolean
        Dim continue = 1

        litMsgBox.Text = ""

        'If continue = 1 Then
        '    If QL1.SelectedIndex = -1 Then
        '        litMsgBox.Text += "No value has been selected for question 1."
        '        continue = 0

        '    Else
        '        QS1 = QL1.Items(QL1.SelectedIndex).Value
        '    End If
        'End If

        If continue = 1 Then
            If QL2.SelectedIndex = -1 Then
                litMsgBox.Text += "No value has been selected for question 1."
```

```
            continue = 0
        Else
            QS2 = QL2.Items(QL2.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL3.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 2."
            continue = 0
        Else
            QS3 = QL3.Items(QL3.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL4.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 3."
            continue = 0
        Else
            QS4 = QL4.Items(QL4.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL5.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 4."
            continue = 0
        Else
            QS5 = QL5.Items(QL5.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL6.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 5."
            continue = 0
        Else
            QS6 = QL6.Items(QL6.SelectedIndex).Value
        End If
    End If

    ' Question 6 is QL8
    If continue = 1 Then
        If QL8.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 6."
            continue = 0
        Else
            QS8 = QL8.Items(QL8.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL7.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 7."
            continue = 0
        Else
            QS7 = QL7.Items(QL7.SelectedIndex).Value
        End If
    End If


    If continue <> 1 Then
        Return False
    Else
        Return True
    End If


End Function
```

```
    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNext.Click

        If CheckForNullValues() Then
            Dim NegotiationID As String
            Dim sqlString As String = "" & _
            "INSERT INTO NegotiationResults (Q01, Q02, Q03, Q04, Q05, Q06, Q07) " & _
            "VALUES ('" & QS2 & "', '" & QS3 & "', '" & QS4 & "', '" & QS5 & "', '" & QS6
& "', '" & QS8 & "', '" & QS7 & "') " & _
            "SELECT @@IDENTITY IDENT "

            Dim ocmd As SqlCommand

            Try
                ocmd = New SqlCommand
                Dim odtr As SqlDataReader
                With ocmd
                    .Connection = New SqlConnection(gSession.connectionString)
                    .Connection.Open()
                    .CommandText = sqlString
                    odtr = .ExecuteReader()
                End With

                odtr.Read()
                NegotiationID = odtr.GetValue(0)
                'litMsgBox.Text = NegotiationID
                Session("NegotiationID") = NegotiationID
            Catch oexpData As OleDb.OleDbException
                ' oexpData.Message

            End Try

            ocmd.Connection.Close()
            ocmd.Dispose()

            sqlString = "" & _
            "INSERT InitTable( NegotiationID, SellerBetaBottom, SellerBetaTop,
SellerBeta, SellerTmax, SellerPmin, SellerPmax, " & _
            "SellerKBottom,   SellerKTop,    SellerK, SellerLearn, SellerSubseq,
SellerFixTmax, BuyerBeta, BuyerTmax, BuyerPmin,     BuyerPmax, " & _
            "BuyerK, BuyerLearn, BuyerSubseq, BuyerFixTmax ) " & _
            "SELECT   NegotiationID = '" & NegotiationID & "', " & _
            "SellerBetaBottom, SellerBetaTop, SellerBeta, SellerTmax, SellerPmin,
SellerPmax, SellerKBottom, " & _
            "SellerKTop, SellerK, SellerLearn, SellerSubseq, SellerFixTmax,
       BuyerBeta, BuyerTmax, BuyerPmin, BuyerPmax, " & _
            "BuyerK, BuyerLearn, BuyerSubseq, BuyerFixTmax " & _
            "FROM     InitializationValues IV " & _
            "WHERE    IV.NegotiationID = '12' "

            gSession.ExecuteNonQuery(sqlString)
            Dim localhost As String = ConfigurationSettings.AppSettings.Get("URL")
            Dim learningManipulationCheck As Boolean = False
            Dim allowSubmitVariables As Boolean = True
            Session("learningManipulationCheck") = learningManipulationCheck
            Session("allowSubmitVariables") = allowSubmitVariables
            Response.Redirect(localhost + "Instructions.aspx")

        End If
    End Sub
End Class
```

B.6 Instructions

```
Public Class Instructions
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Image2 As System.Web.UI.WebControls.Image
    Protected WithEvents Image8 As System.Web.UI.WebControls.Image
    Protected WithEvents btnNext As System.Web.UI.WebControls.Button
    Protected WithEvents Image3 As System.Web.UI.WebControls.Image

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'Put user code to initialize the page here
    End Sub


    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNext.Click
        Dim localhost As String = ConfigurationSettings.AppSettings.Get("URL")
        Response.Redirect(localhost + "NegotiationForm.aspx")
    End Sub
End Class
```

B.7 Negotiation

(Note: This code has been adapted from the work by Mok and Sundarraj (2005))

```
Option Explicit On

Public Class Negotiation

    Private m_NegotiationID As String
    Private m_buyAgent As buyAgent
    Private m_sellAgent As sellAgent
    'Private m_likability As Single
    Private m_sender As String
    Private m_receiver As String
    Private m_currentBid As Double
    Private m_currentAsk As Double
    Private m_lastBid As Double
    Private m_lastAsk As Double
    Private m_finalPrice As Double
    Private m_buyerTurn As Integer
    Private m_sellerTurn As Integer
    Private m_cumBuyerTurn As Integer
    Private m_cumSellerTurn As Integer
    Private m_lastBuyerLearn As Integer
    Private m_lastSellerLearn As Integer
    Private m_nextBuyerLearn As Integer
    Private m_nextSellerLearn As Integer
    Private m_state As String

    Public gSession As DbSession

    Public Sub New(ByVal a_NegotiationID As String)
        m_NegotiationID = a_NegotiationID
        gSession = New DbSession
        gSession.NegotiationID = m_NegotiationID
    End Sub

    Public Property NegotiationID() As String
        Get
            Return m_NegotiationID
        End Get
        Set(ByVal Value As String)
            m_NegotiationID = Value
        End Set
    End Property
    Public Property sender() As String
        Get
            Return m_sender
        End Get
        Set(ByVal x As String)
            m_sender = x
        End Set

    End Property

    Public Property receiver() As String
        Get
            Return m_receiver
        End Get
        Set(ByVal x As String)
            m_receiver = x
        End Set

    End Property

    Public Property buyAgent() As buyAgent
        Get
            Return m_buyAgent
        End Get
```

```
    Set(ByVal x As buyAgent)
        m_buyAgent = x
    End Set

End Property

Public Property sellAgent() As sellAgent
    Get
        Return m_sellAgent
    End Get
    Set(ByVal x As sellAgent)
        m_sellAgent = x
    End Set

End Property

'Public Property Get likability() As Single
'    likability = m_likability
'End Property
'Public Property Let likability(x As Single)
'    m_likability = x
'End Property

Public Property currentBid() As Double
    Get
        Return m_currentBid
    End Get
    Set(ByVal x As Double)
        m_currentBid = x
    End Set

End Property

Public Property currentAsk() As Double
    Get
        Return m_currentAsk
    End Get
    Set(ByVal x As Double)
        m_currentAsk = x
    End Set

End Property

Public Property lastBid() As Double
    Get
        Return m_lastBid
    End Get
    Set(ByVal x As Double)
        m_lastBid = x
    End Set

End Property

Public Property lastAsk() As Double
    Get
        Return m_lastAsk
    End Get
    Set(ByVal x As Double)
        m_lastAsk = x
    End Set

End Property

Public Property finalPrice() As Double
    Get
        Return m_finalPrice
    End Get
    Set(ByVal x As Double)
        m_finalPrice = x
    End Set
```

```
End Property

Public Property buyerTurn() As Integer
    Get
        Return m_buyerTurn
    End Get
    Set(ByVal x As Integer)
        m_buyerTurn = x
    End Set

End Property

Public Property cumBuyerTurn() As Integer
    Get
        Return m_cumBuyerTurn
    End Get
    Set(ByVal x As Integer)
        m_cumBuyerTurn = x
    End Set

End Property

Public Property lastBuyerLearn() As Integer
    Get
        Return m_lastBuyerLearn
    End Get
    Set(ByVal x As Integer)
        m_lastBuyerLearn = x
    End Set

End Property

Public Property nextBuyerLearn() As Integer
    Get
        Return m_nextBuyerLearn
    End Get
    Set(ByVal x As Integer)
        m_nextBuyerLearn = x
    End Set

End Property

Public Property sellerTurn() As Integer
    Get
        Return m_sellerTurn
    End Get
    Set(ByVal x As Integer)
        m_sellerTurn = x
    End Set

End Property

Public Property lastSellerLearn() As Integer
    Get
        Return m_lastSellerLearn
    End Get
    Set(ByVal x As Integer)
        m_lastSellerLearn = x
    End Set

End Property

Public Property nextSellerLearn() As Integer
    Get
        Return m_nextSellerLearn
    End Get
    Set(ByVal x As Integer)
        m_nextSellerLearn = x
    End Set

End Property
```

```
    Public Property cumSellerTurn() As Integer
        Get
            Return m_cumSellerTurn
        End Get
        Set(ByVal x As Integer)
            m_cumSellerTurn = x
        End Set

    End Property

    Public Property state() As String
        Get
            Return m_state
        End Get
        Set(ByVal x As String)
            m_state = x
        End Set

    End Property

    Public Sub InsertInRS(ByVal sender As String, ByVal receiver As String)
        'Dim rs As New ADODB.Recordset
        'Dim sSQL As String

        'sSQL = "SELECT * FROM Negotiation"
        'rs.Open(sSQL, gSession.GetConn, adOpenKeyset, adLockOptimistic)

        'rs.AddNew()
        'rs!sender = sender
        'rs!receiver = receiver
        'rs!buyerTurn = m_cumBuyerTurn
        'rs!sellerTurn = m_cumSellerTurn
        ''rs!BuyAgentLikability = m_likability
        'If m_currentBid <> 0 Then rs!currentBid = m_currentBid
        'rs!currentAsk = m_currentAsk
        'If m_lastBid <> 0 Then rs!lastBid = m_lastBid
        'If m_lastAsk <> 0 Then rs!lastAsk = m_lastAsk
        ''rs!buyAgentStartPrice = m_buyAgentStartPrice
        ''rs!sellAgentStartPrice = m_sellAgentStartPrice
        'If m_finalPrice <> 0 Then rs!finalPrice = m_finalPrice
        'rs!state = m_state
        'rs.Update()

        Dim strCmdText As String
        strCmdText &= "INSERT INTO Negotiation ( NegotiationID, Sender, Receiver,
BuyerTurn, "
        strCmdText &= " SellerTurn, CurrentBid, CurrentAsk, LastBid, LastAsk, "
        strCmdText &= " FinalPrice, State) "
        strCmdText &= " VALUES('" & m_NegotiationID & "', '" & sender & "', '" & receiver
& "', '" & m_cumBuyerTurn & "', '"
        strCmdText &= m_cumSellerTurn

        If m_currentBid <> 0 Then
            strCmdText &= "', '" & m_currentBid & "', '"
        Else
            strCmdText &= "', NULL, '"
        End If
        strCmdText &= m_currentAsk
        If m_lastBid <> 0 Then
            strCmdText &= "', '" & m_lastBid & "', "
        Else
            strCmdText &= "', NULL, "
        End If
        If m_lastAsk <> 0 Then
            strCmdText &= "'" & m_lastAsk & "', "
        Else
            strCmdText &= " NULL, "
        End If
        If m_finalPrice <> 0 Then
            strCmdText &= "'" & m_finalPrice & "', '"
```

```
        Else
            strCmdText &= "NULL, '"
        End If
        strCmdText &= m_state & "')"

        gSession.ExecuteNonQuery(strCmdText)


    End Sub


End Class
```

B.8 NegotiationForm

(Note: Portions of this code has been adapted from the work by Mok and Sundarraj (2005))

```
Option Explicit On

Public Class WebForm1
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents txtbxPmin As System.Web.UI.WebControls.TextBox
    Protected WithEvents txtbxPmax As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents chkbxLearn As System.Web.UI.WebControls.CheckBox
    Protected WithEvents SubmitVariables As System.Web.UI.WebControls.Button
    Protected WithEvents Automatic As System.Web.UI.WebControls.Button
    Protected WithEvents dgrResult As System.Web.UI.WebControls.DataGrid
    Protected WithEvents PlotSurface2D1 As NPlot.Web.PlotSurface2D
    Protected WithEvents litMsgBox As System.Web.UI.WebControls.Literal
    Protected WithEvents txtbxPrice As System.Web.UI.WebControls.TextBox
    Protected WithEvents Manual As System.Web.UI.WebControls.Button
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents Label7 As System.Web.UI.WebControls.Label
    Protected WithEvents Label8 As System.Web.UI.WebControls.Label
    Protected WithEvents Label9 As System.Web.UI.WebControls.Label
    Protected WithEvents Label10 As System.Web.UI.WebControls.Label
    Protected WithEvents Label11 As System.Web.UI.WebControls.Label
    Protected WithEvents Label12 As System.Web.UI.WebControls.Label
    Protected WithEvents Label13 As System.Web.UI.WebControls.Label
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents btnNext As System.Web.UI.WebControls.Button
    Protected WithEvents learningLabel As System.Web.UI.WebControls.Label
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'Put user code to initialize the page here

        If Not IsPostBack Then
            seller_var = New ArrayList
            buyer_var = New ArrayList
            agent_var = New ArrayList
            learning_agent_var = New ArrayList
            gSession = New DbSession
            NegotiationID = Session("NegotiationID")
            gSession.NegotiationID = Session("NegotiationID")
            gCurrent_bid = 0
            SaveSessionVariables()
        End If
```

```
        If (IsPostBack) Then
            litMsgBox.Text = ""   'Clear the message text box
            LoadSessionVariables()
        End If

        Dim allowSubmitVariables As Boolean = Session("allowSubmitVariables")
        ShowHideAgentVar(allowSubmitVariables)


    End Sub

    Dim seller_var As ArrayList
    Dim buyer_var As ArrayList
    Dim agent_var As ArrayList
    Dim learning_agent_var As ArrayList
    Public gSession As DbSession

    Dim gCurrent_bid As Double
    Public Const t_Limit As Integer = 2
    Dim NegotiationID As String
    Dim neg As Negotiation
    Dim buyerEstimate As Boolean
    Dim sellerEstimate As Boolean
    Dim priceArray() As Double


    Private Sub SaveSessionVariables()
        Session("NegotiationID") = NegotiationID
        Session("seller_var") = seller_var
        Session("buyer_var") = buyer_var
        Session("agent_var") = agent_var
        Session("learning_agent_var") = learning_agent_var
        Session("gSession") = gSession
        Session("neg") = neg
        Session("buyerEstimate") = buyerEstimate
        Session("sellerEstimate") = sellerEstimate
        Session("priceArray") = priceArray
        Session("gCurrent_bid") = gCurrent_bid
    End Sub

    Private Sub LoadSessionVariables()
        NegotiationID = Session("NegotiationID")
        seller_var = Session("seller_var")
        buyer_var = Session("buyer_var")
        agent_var = Session("agent_var")
        learning_agent_var = Session("learning_agent_var")
        gSession = Session("gSession")
        neg = Session("neg")
        buyerEstimate = Session("buyerEstimate")
        sellerEstimate = Session("sellerEstimate")
        priceArray = Session("priceArray")
        gCurrent_bid = Session("gCurrent_bid")
    End Sub

    Private Function AgentValid() As Boolean

        Dim continue As Integer = 1

        If continue = 1 Then
            If CInt(txtbxPmax.Text) > CInt(txtbxPmin.Text) And CInt(txtbxPmax.Text) <=
400 Then
                continue = 1
            ElseIf CInt(txtbxPmax.Text) > 400 Then
                litMsgBox.Text = "Max price is very high, enter a value less than or
equal to 400."
                continue = 0
            Else
                litMsgBox.Text = "Max price is less than min price."
                continue = 0
            End If
```

```
        End If

        If continue = 1 Then
            If CInt(txtbxPmin.Text) > 0 Then
                continue = 1
            Else
                litMsgBox.Text = "Min price is less than 1."
                continue = 0
            End If
        End If

        If continue = 1 Then
            Return True
        Else
            Return False
        End If

    End Function

    Private Sub ShowHideAgentVar(ByVal Enable As Boolean)
        ' txtbxTmax.Enabled = Enable
        txtbxPmin.Enabled = Enable
        txtbxPmax.Enabled = Enable
        chkbxLearn.Enabled = Enable
        SubmitVariables.Enabled = Enable


    End Sub

    Private Sub InitializeVarHolders()
        buyer_var.Clear()
        buyer_var.Add(neg.buyAgent.startPrice)
        agent_var.Clear()
        agent_var.Add(neg.buyAgent.startPrice)
        learning_agent_var.Clear()
        learning_agent_var.Add(0)
        seller_var.Clear()
        seller_var.Add(neg.sellAgent.startPrice)
    End Sub

    Private Function PriceValid() As Boolean
        'If txtbxPrice.MaxLength > 0 Then
        Return True
        'Else
        '    litMsgBox.Text = "Price Invalid"
        '    Return False
        'End If
    End Function

    Private Sub ShowHidePrice(ByVal Enable As Boolean)

        txtbxPrice.Enabled = Enable
        Manual.Enabled = Enable
        Automatic.Enabled = Enable


    End Sub

    Private Sub Adddatapoint(ByVal series As Integer, ByVal turn As Integer)
        Dim sqlString As String = "" & _
            " SELECT ps.turn AS Turn, ps.Price AS AskPrice, pb.Price AS BidPrice, a.Price
AS AgentBidPrice " & _
            " FROM (SELECT * FROM Suggested_Bid WHERE NegotiationID = '" & NegotiationID
& "') AS a " & _
            " RIGHT JOIN (SELECT * FROM Price_Seller WHERE NegotiationID = '" &
NegotiationID & "') AS ps ON a.turn = ps.turn " & _
            " LEFT JOIN (SELECT * FROM Price_Buyer  WHERE NegotiationID = '" &
NegotiationID & "') AS pb ON ps.turn = pb.turn " & _
            " WHERE ps.turn = " & turn
```

```
    Dim learn As Boolean = False
    Dim ocmd As SqlCommand

    Try
        ocmd = New SqlCommand
        Dim odtr As SqlDataReader
        With ocmd
            .Connection = New SqlConnection(gSession.connectionString)
            .Connection.Open()
            .CommandText = sqlString
            odtr = .ExecuteReader()
        End With

        odtr.Read()

        If series = 0 Then
            Dim t As Double
            Dim y As Double
            t = turn
            If Not odtr.IsDBNull(1) Then
                y = odtr.GetValue(1)
                seller_var.Add(y)
            End If

            If turn > 5 Then
                learn = True
            End If

        End If
            If series = 1 Then
                Dim t As Double
                Dim y As Double
                t = turn
                If Not odtr.IsDBNull(2) Then
                    y = odtr.GetValue(2)
                    buyer_var.Add(y)
                End If
                If Not odtr.IsDBNull(3) Then
                    y = odtr.GetValue(3)
                    agent_var.Add(y)
                    If turn >= 5 Then
                        learn = True
                        learning_agent_var.Add(y)
                    Else
                        learning_agent_var.Add(0)
                    End If
                End If

            End If

    Catch oexpData As OleDb.OleDbException
        ' oexpData.Message

    End Try

    ocmd.Connection.Close()
    ocmd.Dispose()

    DrawGraph(learn)

End Sub

Private Sub DrawGraph(ByVal Learn As Boolean)



    Dim buyer_line As New NPlot.LinePlot
    buyer_line.Label = "Buyer"
    buyer_line.OrdinateData = buyer_var
    buyer_line.Pen = New Pen(Color.Blue, 2.0F)
```

```
        Dim seller_line As New NPlot.LinePlot
        seller_line.Label = "Seller"
        seller_line.OrdinateData = seller_var

        seller_line.Pen = New Pen(Color.Black, 2.0F)

        'Dim agent_line As New NPlot.LinePlot
        'agent_line.Label = "Agent"
        'agent_line.OrdinateData = agent_var
        'agent_line.Pen = New Pen(Color.Red, 1.0F)

        Dim agent_marker As NPlot.PointPlot

        agent_marker = New NPlot.PointPlot
        agent_marker.DataSource = agent_var
        agent_marker.Label = "Agent"
        agent_marker.Marker = New NPlot.Marker(NPlot.Marker.MarkerType.Cross1, 10)
        agent_marker.Marker.DropLine = True
        agent_marker.Marker.Pen = Pens.Red
        agent_marker.Marker.Filled = False


        Dim learning_agent_marker As NPlot.PointPlot

        learning_agent_marker = New NPlot.PointPlot
        learning_agent_marker.DataSource = learning_agent_var
        learning_agent_marker.Label = "Learning in Progress"
        learning_agent_marker.Marker = New NPlot.Marker(NPlot.Marker.MarkerType.Square,
10)
        learning_agent_marker.Marker.DropLine = True
        learning_agent_marker.Marker.Pen = Pens.SteelBlue
        learning_agent_marker.Marker.Filled = False


        PlotSurface2D1.Clear()
        PlotSurface2D1.Add(buyer_line)
        PlotSurface2D1.Add(seller_line)
        PlotSurface2D1.Add(agent_marker)
        If (chkbxLearn.Checked And Learn) Then
            PlotSurface2D1.Add(learning_agent_marker)
            learningLabel.Visible = True
        End If
        'PlotSurface2D1.Legend()
        PlotSurface2D1.Title = "Negotiation"
        PlotSurface2D1.YAxis1.Label = "Price"
        PlotSurface2D1.XAxis1.Label = "Turn"
        PlotSurface2D1.YAxis1.WorldMin -= 15.0F
        PlotSurface2D1.YAxis1.WorldMax += 15.0F
        PlotSurface2D1.XAxis1.WorldMin = 0.0F
        PlotSurface2D1.XAxis1.WorldMax = 26.0F

        PlotSurface2D1.Legend = New NPlot.Legend
        PlotSurface2D1.Legend.AttachTo(NPlot.PlotSurface2D.XAxisPosition.Top,
NPlot.PlotSurface2D.YAxisPosition.Right)
        PlotSurface2D1.Legend.HorizontalEdgePlacement = NPlot.Legend.Placement.Inside
        PlotSurface2D1.Legend.VerticalEdgePlacement = NPlot.Legend.Placement.Inside
        PlotSurface2D1.Legend.XOffset = 0
        PlotSurface2D1.Legend.YOffset = 0
        PlotSurface2D1.Refresh()

    End Sub

    Private Sub FillDGR(ByVal series As Integer, ByVal turn As Integer)
        Dim myConnection As SqlConnection = New SqlConnection
        myConnection.ConnectionString = gSession.connectionString
        Dim sqlString As String = "" & _
            " SELECT TOP 10 ps.turn AS Turn, CAST(ps.Price AS DECIMAL(5,2)) AS [Seller's
Ask Price], " & _
            " CAST(pb.Price AS DECIMAL(5,2)) AS [Your Bid Price], CAST(a.Price AS
DECIMAL(5,2)) AS [Your Buyer Agent's Suggested Price] " & _
```

```
            " FROM (SELECT * FROM Suggested_Bid WHERE NegotiationID = '" & NegotiationID
& "') AS a " & _
            " RIGHT JOIN (SELECT * FROM Price_Seller WHERE NegotiationID = '" &
NegotiationID & "') AS ps ON a.turn = ps.turn " & _
            " LEFT JOIN (SELECT * FROM Price_Buyer  WHERE NegotiationID = '" &
NegotiationID & "') AS pb ON ps.turn = pb.turn ORDER BY Turn DESC"



        'Dim sqlString As String = "" & _
        '    "SELECT negotiation.sellerturn AS Turn, negotiation.currentAsk AS AskPrice,
negotiation.currentBid AS BidPrice, Suggested_Bid.Price AS SuggestedBidPrice " & _
        '    " FROM Suggested_Bid RIGHT JOIN negotiation ON negotiation.sellerturn =
suggested_bid.turn "
        Dim da As SqlDataAdapter = New SqlDataAdapter(sqlString, myConnection)
        Dim ds As DataSet = New DataSet
        da.Fill(ds)



        '********************  Commented Out Section
*******************************

        dgrResult.DataSource = ds
        dgrResult.DataBind()
        'dgrResult.Expand(-1)
        'dgrResult.ReadOnly = True
        '********************  End of Commented Out Section
*******************************
        da.Dispose()


    End Sub

    Private Sub SubmitVariables_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SubmitVariables.Click
        Dim allowSubmitVariables As Boolean = Session("allowSubmitVariables")

        If (AgentValid() And allowSubmitVariables) Then
            'txtbxDeal.Text = ""
            'txtbxFinal.Text = ""
            'txtbxTurnNum.Text = ""
            'txtbxApproved.Text = ""
            txtbxPrice.Text = txtbxPmin.Text


            Dim agentB As buyAgent
            Dim agentS As sellAgent
            'Dim neg As Negotiation

            Call gSession.Remove()
            agentB = New buyAgent(NegotiationID)
            agentS = New sellAgent(NegotiationID)
            neg = New Negotiation(NegotiationID)
            ' Call ClearOldData()
            Call Initialize(agentB, agentS)
            Call Match(neg, agentB, agentS)

            ShowHidePrice(True)

            allowSubmitVariables = False
            Session("allowSubmitVariables") = allowSubmitVariables
            ShowHideAgentVar(allowSubmitVariables)
            InitializeVarHolders()

            negot(False)

            gCurrent_bid = 0

            ' Call Negotiate(neg)
```

```
            ' Call ReturnResult(neg)
            SaveSessionVariables()

        End If
    End Sub

    Private Sub Manual_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Manual.Click
        Dim currentBid As Double = txtbxPrice.Text
        If (currentBid <= 400) Then
            If (gCurrent_bid < currentBid) Then
                negot(True) ' buyer
                negot(False) ' seller
            ElseIf (gCurrent_bid = currentBid) Then
                litMsgBox.Text = "You can not bid the same as your previous bid."
            Else
                litMsgBox.Text = "You can not bid lower than your last bid."
            End If
        Else
            litMsgBox.Text = "Your current bid is too high, give a bid less than or equal
to 400."
        End If

        'Maintain current session information
        SaveSessionVariables()

    End Sub

    Private Sub Automatic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Automatic.Click
        'Dim currentBid As Double = txtbxPrice.Text
        'If (neg.lastBid <= currentBid) Then
        negot(False) ' buyer
        negot(False) ' seller
        'Else
        '    litMsgBox.Text = "You can not bid lower than your last bid."
        'End If


        'Maintain current session information
        SaveSessionVariables()

    End Sub

    Private Sub negot(ByVal manual As Boolean)

        Dim sSQL As String
        Dim suggestedBid As Double

        'Do Until neg.state = "Buyer Approved" Or neg.state = "Seller Approved" Or
neg.state = "Failed"
        If PriceValid() Then


            If neg.state = "Wait for Seller" Then
                neg.sellerTurn = neg.sellerTurn + 1
                neg.cumSellerTurn = neg.cumSellerTurn + 1
                If neg.cumSellerTurn > neg.sellAgent.cumTmax Then
                    neg.state = "Failed"
                    sSQL = "UPDATE Negotiation SET state = 'Failed' WHERE Sender =
'Seller' AND Receiver = 'Buyer' AND BuyerTurn = " & neg.cumBuyerTurn - 1 & " AND
SellerTurn = " & neg.cumSellerTurn - 1 & " AND NegotiationID = '" & NegotiationID & "' "
                    gSession.ExecuteNonQuery(sSQL)
                    GoTo sEnd2
                End If

                If neg.nextSellerLearn = neg.cumSellerTurn Then
                    priceArray = gSession.ListBuyerPrice(neg)
                    sellerEstimate = neg.sellAgent.Estimate(neg, priceArray)
                    If sellerEstimate = False Then
                        Call InsertRecord(priceArray)
```

```
                        priceArray(4) = EstimatePrice("Estimate Buyer")
                        sellerEstimate = neg.sellAgent.Estimate(neg, priceArray)
                    End If
                    Call neg.sellAgent.FindOptimal(neg)
                End If
                neg.lastAsk = neg.currentAsk
                neg.currentAsk = GetPrice(neg.sellAgent, neg.sellerTurn)

                neg.sellAgent.askPrice = neg.currentAsk
                If neg.currentAsk = neg.lastAsk Then
                    neg.state = "Failed"
                    GoTo sEnd
                End If
                If neg.currentAsk <= neg.currentBid Then
                    neg.state = "Seller Approved"
                    neg.finalPrice = neg.currentBid
                    GoTo sEnd
                Else
                    neg.state = "Wait for Buyer"
                End If
sEnd:           Call neg.InsertInRS(neg.sellAgent.ID, neg.buyAgent.ID)
                Dim insertString As String = "INSERT INTO " & _
                    " Price_Seller(NegotiationID, turn, Price) " & _
                    " VALUES('" & NegotiationID & "', " & _
                        " '" & neg.cumSellerTurn & "'," & _
                        " '" & neg.currentAsk & "' )"
                gSession.ExecuteNonQuery(insertString)
                FillDGR(0, neg.cumSellerTurn)
                Adddatapoint(0, neg.cumSellerTurn)
sEnd2:          GoTo bEnd2     ' force it to skip evaluating the buyer
            End If

            If neg.state = "Wait for Buyer" Then
                neg.buyerTurn = neg.buyerTurn + 1
                neg.cumBuyerTurn = neg.cumBuyerTurn + 1
                If neg.cumBuyerTurn > neg.buyAgent.cumTmax Then
                    neg.state = "Failed"
                    sSQL = "UPDATE Negotiation SET state = 'Failed' WHERE Sender =
'Buyer' AND Receiver = 'Seller' AND BuyerTurn = " & neg.cumBuyerTurn - 1 & " AND
SellerTurn = " & neg.sellerTurn - 1 & " AND NegotiationID = '" & NegotiationID & "' "
                    gSession.ExecuteNonQuery(sSQL)
                    GoTo bEnd2
                End If

                If neg.nextBuyerLearn = neg.cumBuyerTurn Then
                    priceArray = gSession.ListSellerPrice(neg)
                    buyerEstimate = neg.buyAgent.Estimate(neg, priceArray)
                    If buyerEstimate = False Then
                        Call InsertRecord(priceArray)
                        priceArray(4) = EstimatePrice("Estimate Seller")
                        buyerEstimate = neg.buyAgent.Estimate(neg, priceArray)
                    End If
                    Call neg.buyAgent.FindOptimal(neg)
                End If
                neg.lastBid = neg.currentBid
                suggestedBid = GetPrice(neg.buyAgent, neg.buyerTurn)
                '*********************************************
                ' place controls for price here
                '*********************************************
                If gCurrent_bid > suggestedBid Then
                    Dim tempBuyer As buyAgent
                    tempBuyer = neg.buyAgent
                    tempBuyer.startPrice = gCurrent_bid
                    suggestedBid = GetPrice(tempBuyer, neg.buyerTurn) 'gCurrent_bid + 5
                End If
                If manual Then
                    neg.currentBid = txtbxPrice.Text
                Else
                    neg.currentBid = suggestedBid
                End If
                If neg.currentBid = neg.lastBid Then
```

```
                    litMsgBox.Text = "You have placed the same bid twice thus the seller
has declined."
                    neg.state = "Failed"
                    GoTo bEnd
                End If
                If neg.currentBid >= neg.currentAsk Then
                    neg.state = "Buyer Approved"
                    neg.finalPrice = neg.currentAsk
                    GoTo bEnd
                Else
                    neg.state = "Wait for Seller"
                End If
bEnd:           Call neg.InsertInRS(neg.buyAgent.ID, neg.sellAgent.ID)
                Dim insertString As String = "INSERT INTO " & _
                    " Price_Buyer(NegotiationID, turn, Price) " & _
                    " VALUES('" & NegotiationID & "', " & _
                        " '" & neg.cumBuyerTurn & "'," & _
                        " '" & neg.currentBid & "' )"
                gSession.ExecuteNonQuery(insertString)
                insertString = "INSERT INTO " & _
                                    " Suggested_Bid(NegotiationID, turn, Price) " & _
                                    " VALUES('" & NegotiationID & "', " & _
                                        " '" & neg.cumBuyerTurn & "'," & _
                                        " '" & suggestedBid & "' )"
                gCurrent_bid = neg.currentBid
                gSession.ExecuteNonQuery(insertString)
                FillDGR(1, neg.cumBuyerTurn)
                Adddatapoint(1, neg.cumBuyerTurn)
bEnd2:      End If

            ' Loop
        End If


        If (neg.state = "Buyer Approved" Or neg.state = "Seller Approved" Or neg.state =
"Failed") Then
            ShowHidePrice(False)

            Dim allowSubmitVariables As Boolean = False
            Session("allowSubmitVariables") = allowSubmitVariables
            ShowHideAgentVar(allowSubmitVariables)
            Call ReturnResult(neg)
            If neg.state = "Buyer Approved" Or neg.state = "Seller Approved" Then
                Dim finalprice As String
                finalprice = String.Format("{0:c}", neg.finalPrice)
                litMsgBox.Text = "Negotiation completed, final price is " & finalprice &
" (" & neg.state & "). "
            Else
                litMsgBox.Text = "Negotiation failed. YOU DON'T HAVE A DEAL!!!"
            End If
            btnNext.Enabled = True
        End If


    End Sub

    Public Sub Initialize(ByRef agentB As buyAgent, ByRef agentS As sellAgent)

        Dim SellerBetaBottom As Single
        Dim SellerBetaTop As Single
        Dim SellerKBottom As Single
        Dim SellerKTop As Single


        Dim sqlString As String = "" & _
            "SELECT SellerBeta, SellerTmax, SellerPmin, SellerPmax, SellerK, SellerLearn,
SellerSubseq, SellerFixTmax, " & _
            "BuyerBeta, BuyerTmax, BuyerPmin, BuyerPmax, BuyerK, BuyerLearn, BuyerSubseq,
BuyerFixTmax, " & _
            "NegotiationID, SellerBetaBottom, SellerBetaTop, SellerKBottom, SellerKTop "
& _
```

```
            " FROM InitTable " & _
            " WHERE NegotiationID = '" & NegotiationID & "' "

        Dim ocmd As SqlCommand

        Try
            ocmd = New SqlCommand
            Dim odtr As SqlDataReader
            With ocmd
                .Connection = New SqlConnection(gSession.connectionString)
                .Connection.Open()
                .CommandText = sqlString
                odtr = .ExecuteReader()
            End With

            odtr.Read()

            'NegotiationID = odtr.GetValue(16)


            SellerBetaBottom = odtr.GetValue(17)
            SellerBetaTop = odtr.GetValue(18)
            SellerKBottom = odtr.GetValue(19)
            SellerKTop = odtr.GetValue(20)

            Dim rand As New Random
            agentS.beta = SellerBetaBottom + ((SellerBetaTop - SellerBetaBottom) *
rand.NextDouble()) ' odtr.GetValue(0) 'SellerBeta.Text
            agentS.Tmax = rand.Next(8, 18) 'Randomize tmax between 8 and 18
odtr.GetValue(1) 'SellerTmax.Text
            agentS.cumTmax = agentS.Tmax
            agentS.startPrice = odtr.GetValue(3) 'SellerPmax.Text
            agentS.minPrice = odtr.GetValue(2) 'SellerPmin.Text
            agentS.K = SellerKBottom + ((SellerKTop - SellerKBottom) * rand.NextDouble())
'odtr.GetValue(4) 'SellerK.Text
            'If SellerLearn.Text = "Yes" Then
            If UCase(CChar(odtr.GetValue(5))) = "TRUE" Then
                agentS.learn = True
            Else
                agentS.learn = False
            End If

            'agentS.learn = odtr.GetValue(5)

            agentS.learnStrategy = odtr.GetValue(6) 'SellerSubseq.Text
            agentS.TmaxStrategy = odtr.GetValue(7) 'SellerTmax.Text

            agentB.beta = odtr.GetValue(8) 'BuyerBeta.Text
            agentB.Tmax = agentS.Tmax 'txtbxTmax.Text 'BuyerTmax.Text
            agentB.cumTmax = agentB.Tmax
            agentB.startPrice = txtbxPmin.Text 'BuyerPmin.Text
            agentB.maxPrice = txtbxPmax.Text 'BuyerPmax.Text
            agentB.K = 0.1 'BuyerK.Text
            agentB.ID = "Buyer"
            'If BuyerLearn.Text = "Yes" Then
            Dim Dice As Integer = NegotiationID Mod 2
            Dim learningManipulationCheck As Boolean
            If Dice = 1 Then
                chkbxLearn.Checked = True
                agentB.learn = True
                learningManipulationCheck = True
                Session("learningManipulationCheck") = learningManipulationCheck
            Else
                chkbxLearn.Checked = False
                agentB.learn = False
                learningManipulationCheck = False
                Session("learningManipulationCheck") = learningManipulationCheck
            End If
            agentB.learnStrategy = odtr.GetValue(14) 'BuyerSubseq.Text
            agentB.TmaxStrategy = odtr.GetValue(15) 'BuyerTmax.Text
```

```
        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()

        sqlString = " UPDATE InitTable " & _
                    " SET SellerBeta = '" & agentS.beta.ToString & "', " & _
                    " SellerTmax = '" & agentS.Tmax & "', " & _
                    " SellerK = '" & agentS.K & "', " & _
                    " BuyerTmax = '" & agentB.Tmax & "', " & _
                    " BuyerPmin = '" & txtbxPmin.Text & "', " & _
                    " BuyerPmax = '" & txtbxPmax.Text & "', " & _
                    " BuyerLearn = '" & chkbxLearn.Checked.ToString & "' " & _
                    " WHERE NegotiationID = '" & NegotiationID & "' "

        gSession.ExecuteNonQuery(sqlString)


        '*********Session Management
        '*********End of Session Management

    End Sub
    Public Function GetPrice(ByRef agent As Object, ByVal t As Integer)
        Dim alpha As Single

        alpha = Exp(((1 - t / agent.Tmax) ^ agent.beta) * Log(agent.K))
        If agent.ID = "Seller" Then
            agent.askPrice = agent.minPrice + (1 - alpha) * (agent.startPrice -
agent.minPrice)
            If agent.askPrice < agent.minPrice Then agent.askPrice = agent.minPrice
            GetPrice = agent.askPrice
        ElseIf agent.ID = "Buyer" Then
            agent.bidPrice = agent.startPrice + alpha * (agent.maxPrice -
agent.startPrice) ' txtbxPrice.Text '
            If agent.bidPrice > agent.maxPrice Then agent.bidPrice = agent.maxPrice
            GetPrice = agent.bidPrice
        End If
    End Function
    Public Sub Match(ByRef neg As Negotiation, ByRef agentB As buyAgent, ByRef agentS As
sellAgent)
        neg.buyAgent = agentB
        neg.sellAgent = agentS
        neg.buyerTurn = -1
        neg.cumBuyerTurn = -1
        neg.sellerTurn = 0
        neg.cumSellerTurn = 0
        neg.lastBuyerLearn = 0
        neg.lastSellerLearn = 0
        If neg.buyAgent.learn = True Then
            neg.nextBuyerLearn = 4
        Else
            neg.nextBuyerLearn = 0
        End If
        If neg.sellAgent.learn = True Then
            neg.nextSellerLearn = 5
        Else
            neg.nextSellerLearn = 0
        End If
        neg.currentAsk = GetPrice(agentS, neg.sellerTurn)

        neg.state = "Wait for Buyer"
        Call neg.InsertInRS(agentS.ID, agentB.ID)

        neg.buyerTurn = neg.buyerTurn + 1
        neg.cumBuyerTurn = neg.cumBuyerTurn + 1
        neg.currentBid = GetPrice(agentB, neg.buyerTurn)

        If neg.currentAsk <= neg.currentBid Then
```

```
            neg.state = "Buyer Approved"
            neg.finalPrice = neg.currentAsk
        Else
            neg.state = "Wait for Seller"
        End If
        Call neg.InsertInRS(agentB.ID, agentS.ID)
        FillDGR(1, neg.cumBuyerTurn)
    End Sub
    Public Sub Negotiate(ByRef neg As Negotiation)
        '        Dim sSQL As String
        '        Dim buyerEstimate As Boolean
        '        Dim sellerEstimate As Boolean
        '        Dim priceArray() As Double

        '        Do Until neg.state = "Buyer Approved" Or neg.state = "Seller Approved"
Or neg.state = "Failed"
        '            If neg.state = "Wait for Seller" Then
        '                neg.sellerTurn = neg.sellerTurn + 1
        '                neg.cumSellerTurn = neg.cumSellerTurn + 1
        '                If neg.cumSellerTurn > neg.sellAgent.cumTmax Then
        '                    neg.state = "Failed"
        '                    sSQL = "UPDATE Negotiation SET state = 'Failed' WHERE Sender
= 'Seller' AND Receiver = 'Buyer' AND BuyerTurn = " & neg.cumBuyerTurn - 1 & " AND
SellerTurn = " & neg.cumSellerTurn - 1
        '                    gSession.ExecuteNonQuery(sSQL)
        '                    GoTo sEnd2
        '                End If

        '                If neg.nextSellerLearn = neg.cumSellerTurn Then
        '                    priceArray = gSession.ListBuyerPrice(neg)
        '                    sellerEstimate = neg.sellAgent.Estimate(neg, priceArray)
        '                    If sellerEstimate = False Then
        '                        Call InsertRecord(priceArray)
        '                        priceArray(4) = EstimatePrice("Estimate Buyer")
        '                        sellerEstimate = neg.sellAgent.Estimate(neg, priceArray)
        '                    End If
        '                    Call neg.sellAgent.FindOptimal(neg)
        '                End If
        '                neg.lastAsk = neg.currentAsk
        '                neg.currentAsk = GetPrice(neg.sellAgent, neg.sellerTurn)

        '                neg.sellAgent.askPrice = neg.currentAsk
        '                If neg.currentAsk = neg.lastAsk Then
        '                    neg.state = "Failed"
        '                    GoTo sEnd
        '                End If
        '                If neg.currentAsk <= neg.currentBid Then
        '                    neg.state = "Seller Approved"
        '                    neg.finalPrice = neg.currentBid
        '                    GoTo sEnd
        '                Else
        '                    neg.state = "Wait for Buyer"
        '                End If
        'sEnd:           Call neg.InsertInRS(neg.sellAgent.ID, neg.buyAgent.ID)
        'sEnd2:      End If

        '            If neg.state = "Wait for Buyer" Then
        '                neg.buyerTurn = neg.buyerTurn + 1
        '                neg.cumBuyerTurn = neg.cumBuyerTurn + 1
        '                If neg.cumBuyerTurn > neg.buyAgent.cumTmax Then
        '                    neg.state = "Failed"
        '                    sSQL = "UPDATE Negotiation SET state = 'Failed' WHERE Sender
= 'Buyer' AND Receiver = 'Seller' AND BuyerTurn = " & neg.cumBuyerTurn - 1 & " AND
SellerTurn = " & neg.sellerTurn - 1
        '                    gSession.ExecuteNonQuery(sSQL)
        '                    GoTo bEnd2
        '                End If

        '                If neg.nextBuyerLearn = neg.cumBuyerTurn Then
        '                    priceArray = gSession.ListSellerPrice(neg)
        '                    buyerEstimate = neg.buyAgent.Estimate(neg, priceArray)
```

```
        '                    If buyerEstimate = False Then
        '                        Call InsertRecord(priceArray)
        '                        priceArray(4) = EstimatePrice("Estimate Seller")
        '                        buyerEstimate = neg.buyAgent.Estimate(neg, priceArray)
        '                    End If
        '                    Call neg.buyAgent.FindOptimal(neg)
        '                End If
        '                neg.lastBid = neg.currentBid
        '                neg.currentBid = GetPrice(neg.buyAgent, neg.buyerTurn)
        '                If neg.currentBid = neg.lastBid Then
        '                    neg.state = "Failed"
        '                    GoTo bEnd
        '                End If
        '                If neg.currentBid >= neg.currentAsk Then
        '                    neg.state = "Buyer Approved"
        '                    neg.finalPrice = neg.currentAsk
        '                    GoTo bEnd
        '                Else
        '                    neg.state = "Wait for Seller"
        '                End If
        'bEnd:           Call neg.InsertInRS(neg.buyAgent.ID, neg.sellAgent.ID)
        'bEnd2:      End If

        '        Loop
    End Sub
    Public Sub ReturnResult(ByVal neg As Negotiation)
        'Dim i As Integer
        'Dim rs As New ADODB.Recordset

        'i = 13
        'rs = gSession.GetSellerPrice
        'rs.MoveFirst()
        'Do Until rs.EOF
        '    ActiveSheet.Cells(i, 2) = rs!currentAsk
        '    i = i + 1
        '    rs.MoveNext()
        'Loop
        'rs = Nothing

        'i = 13
        'rs = gSession.GetBuyerPrice
        'rs.MoveFirst()
        'Do Until rs.EOF
        '    ActiveSheet.Cells(i, 3) = rs!currentBid
        '    i = i + 1
        '    rs.MoveNext()
        'Loop
        'rs = Nothing

        Dim deal As Boolean
        Dim final As Double
        Dim turn As Integer
        Dim whom As String
        deal = gSession.GetDeal(final, turn, whom)
        If deal = True Then
            'txtbxDeal.Text = "Yes"
            'txtbxFinal.Text = final
            'txtbxTurnNum.Text = turn
            'txtbxApproved.Text = whom
        Else
            'txtbxDeal.Text = "No"
        End If

        'Dim conn As New ADODB.Connection
        'Dim sSQL As String
        'conn = gSession.GetConn

        'If neg.buyAgent.learn = True Then
        '    sSQL = "SELECT * FROM PredictedSeller"
        '    rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
```

```
        '    i = 13
        '    rs.MoveFirst()
        '    Do Until rs.EOF
        '        ActiveSheet.Cells(i, 5) = rs!price
        '        i = i + 1
        '        rs.MoveNext()
        '    Loop
        '    rs = Nothing
        'End If

        'If neg.sellAgent.learn = True Then
        '    sSQL = "SELECT * FROM PredictedBuyer"
        '    rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
        '    i = 13
        '    rs.MoveFirst()
        '    Do Until rs.EOF
        '        ActiveSheet.Cells(i, 4) = rs!price
        '        i = i + 1
        '        rs.MoveNext()
        '    Loop
        '    rs = Nothing
        'End If

    End Sub
    'Public Sub ClearOldData()

    '    Range("B13:E43").Select()
    '    Selection.ClearContents()
    '    Range("G1:G4").Select()
    '    Selection.ClearContents()
    'End Sub
    Public Function EstimatePrice(ByVal choice As String) As Double
        'Dim conn As New ADODB.Connection
        'Dim rs As New ADODB.Recordset
        Dim sSQL As String
        Dim a, b As Single

        'conn = gSession.GetConn
        sSQL = "SELECT SUM(X*(LOG(Y)/LOG(10))) AS P1, AVG(X) As P2, SUM((LOG(Y)/LOG(10)))
AS P3, SUM(X*X) AS P4, COUNT(*) AS P5 FROM FitLine WHERE NegotiationID = '" &
NegotiationID & "' "
        'rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
        'rs.MoveFirst()
        Dim strConn As String = gSession.connectionString

        Dim ocmd As SqlCommand

        Dim rs_P1 As Integer
        Dim rs_P2 As Integer
        Dim rs_P3 As Integer
        Dim rs_P4 As Integer
        Dim rs_P5 As Integer

        Try
            ocmd = New SqlCommand
            Dim odtr As SqlDataReader
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = sSQL
                odtr = .ExecuteReader()
            End With

            odtr.Read()
            rs_P1 = odtr.GetValue(0)
            rs_P2 = odtr.GetValue(1)
            rs_P3 = odtr.GetValue(2)
            rs_P4 = odtr.GetValue(3)
            rs_P5 = odtr.GetValue(4)
```

```
        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()

        b = 10 ^ ((rs_P1 - rs_P2 * rs_P3) / (rs_P4 - rs_P5 * rs_P2 * rs_P2))
        a = 10 ^ ((rs_P3 / rs_P5) - rs_P2 * (Log(b) / Log(10)))

        EstimatePrice = a * (b ^ 4)

        sSQL = "DELETE FROM FitLine WHERE NegotiationID = '" & NegotiationID & "' "
        'conn.Execute(sSQL)
        gSession.ExecuteNonQuery(sSQL)
        'rs = Nothing
        'conn = Nothing
    End Function
    Public Sub InsertRecord(ByVal priceArray() As Double)
        'Dim conn As New ADODB.Connection
        Dim sSQL As String
        Dim i As Integer

        'conn = gSession.GetConn
        Do Until i > 3
            sSQL = "INSERT INTO FitLine(NegotiationID, X, Y) VALUES('" & NegotiationID &
"', " & i & ", " & priceArray(i) & ")"
            gSession.ExecuteNonQuery(sSQL)
            i = i + 1
        Loop
    End Sub




    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNext.Click
        Session("NegotiationID") = NegotiationID
        Dim localhost As String = ConfigurationSettings.AppSettings.Get("URL")
        Response.Redirect(localhost + "Questionnaire_Page_1.aspx")
    End Sub
End Class
```

B.9 Questionnaire Page 1

```
Public Class Questionnaire_Page_1
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents btnNext As System.Web.UI.WebControls.Button
    Protected WithEvents QL1 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL2 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL3 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL4 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL5 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL6 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL7 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL8 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL9 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL10 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL11 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL12 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL13 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL14 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL15 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents Form1 As System.Web.UI.HtmlControls.HtmlForm
    Protected WithEvents litMsgBox As System.Web.UI.WebControls.Literal

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        NegotiationID = Session("NegotiationID")
        gSession = New DbSession
        Session("NegotiationID") = NegotiationID
    End Sub

    Dim NegotiationID As String
    Dim gSession As DbSession
    Dim QS1 As String
    Dim QS2 As String
    Dim QS3 As String
    Dim QS4 As String
    Dim QS5 As String
    Dim QS6 As String
    Dim QS7 As String
    Dim QS8 As String
    Dim QS9 As String
    Dim QS10 As String
    Dim QS11 As String



    Private Function CheckForNullValues() As Boolean
        Dim continue = 1
```

```
litMsgBox.Text = ""

If continue = 1 Then
    If QL1.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 1."
        continue = 0
    Else
        QS1 = QL1.Items(QL1.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL2.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 2."
        continue = 0
    Else
        QS2 = QL2.Items(QL2.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL3.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 3."
        continue = 0
    Else
        QS3 = QL3.Items(QL3.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL4.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 4."
        continue = 0
    Else
        QS4 = QL4.Items(QL4.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL5.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 5."
        continue = 0
    Else
        QS5 = QL5.Items(QL5.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL6.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 6."
        continue = 0
    Else
        QS6 = QL6.Items(QL6.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL7.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 7."
        continue = 0
    Else
        QS7 = QL7.Items(QL7.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL8.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 8."
        continue = 0
    Else
        QS8 = QL8.Items(QL8.SelectedIndex).Value
```

```
            End If
        End If

        If continue = 1 Then
            If QL9.SelectedIndex = -1 Then
                litMsgBox.Text += "No value has been selected for question 9."
                continue = 0
            Else
                QS9 = QL9.Items(QL9.SelectedIndex).Value
            End If
        End If

        If continue = 1 Then
            If QL10.SelectedIndex = -1 Then
                litMsgBox.Text += "No value has been selected for question 10."
                continue = 0
            Else
                QS10 = QL10.Items(QL10.SelectedIndex).Value
            End If
        End If

        If continue = 1 Then
            If QL11.SelectedIndex = -1 Then
                litMsgBox.Text += "No value has been selected for question 11."
                continue = 0
            Else
                QS11 = QL11.Items(QL11.SelectedIndex).Value
            End If
        End If




        If continue <> 1 Then
            Return False
        Else
            Return True
        End If


    End Function

    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNext.Click

        If CheckForNullValues() Then

            Dim sqlString As String = "" & _
            "UPDATE NegotiationResults " & _
            "SET " & _
            "Q08 = '" & QS1 & "', " & _
            "Q09 = '" & QS2 & "', " & _
            "Q10 = '" & QS3 & "', " & _
            "Q11 = '" & QS4 & "', " & _
            "Q12 = '" & QS5 & "', " & _
            "Q13 = '" & QS6 & "', " & _
            "Q14 = '" & QS7 & "', " & _
            "Q15 = '" & QS8 & "', " & _
            "Q16 = '" & QS9 & "', " & _
            "Q17 = '" & QS10 & "', " & _
            "Q18 = '" & QS11 & "' " & _
            "WHERE NegotiationID = '" & NegotiationID & "' "

            Session("NegotiationID") = NegotiationID

            gSession.ExecuteNonQuery(sqlString)
            Dim localhost As String = ConfigurationSettings.AppSettings.Get("URL")
            Response.Redirect(localhost + "Questionnaire_Page_2.aspx")

        End If
    End Sub
```

```
End Class
```

B.10 Questionnaire Page 2

```
Public Class Questionnaire_Page_2
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents btnNext As System.Web.UI.WebControls.Button
    Protected WithEvents litMsgBox As System.Web.UI.WebControls.Literal
    Protected WithEvents Form1 As System.Web.UI.HtmlControls.HtmlForm
    Protected WithEvents QL16 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL17 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL18 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL19 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL20 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL21 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL22 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL12 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL13 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL14 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL15 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents learningCheck As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents learningLabel As System.Web.UI.WebControls.Label
    Protected WithEvents learningLabelNum As System.Web.UI.WebControls.Label
    Protected WithEvents QL23 As System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents QL24 As System.Web.UI.WebControls.RadioButtonList

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        NegotiationID = Session("NegotiationID")
        gSession = New DbSession
        Session("NegotiationID") = NegotiationID
        learningManipulationCheck = Session("learningManipulationCheck")
        learningLabelNum.Visible = learningManipulationCheck
        learningLabel.Visible = learningManipulationCheck
        learningCheck.Visible = learningManipulationCheck
    End Sub

    Dim NegotiationID As String
    Dim learningManipulationCheck As Boolean
    Dim gSession As DbSession

    Dim QS12 As String
    Dim QS13 As String
    Dim QS14 As String
    Dim QS15 As String
    Dim QS16 As String
    Dim QS17 As String
    Dim QS18 As String
    Dim QS19 As String
    Dim QS20 As String
    Dim QS21 As String
```

```
Dim QS22 As String
Dim QS23 As String
Dim QS24 As String
Dim QS25 As String


Private Function CheckForNullValues() As Boolean
    Dim continue = 1

    litMsgBox.Text = ""

    If continue = 1 Then
        If QL12.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 12."
            continue = 0
        Else
            QS12 = QL12.Items(QL12.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL13.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 13."
            continue = 0
        Else
            QS13 = QL13.Items(QL13.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL14.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 14."
            continue = 0
        Else
            QS14 = QL14.Items(QL14.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL15.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 15."
            continue = 0
        Else
            QS15 = QL15.Items(QL15.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL16.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 16."
            continue = 0
        Else
            QS16 = QL16.Items(QL16.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL17.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 17."
            continue = 0
        Else
            QS17 = QL17.Items(QL17.SelectedIndex).Value
        End If
    End If

    If continue = 1 Then
        If QL18.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 18."
            continue = 0
        Else
            QS18 = QL18.Items(QL18.SelectedIndex).Value
```

```
    End If
End If

If continue = 1 Then
    If QL19.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 19."
        continue = 0
    Else
        QS19 = QL19.Items(QL19.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL20.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 20."
        continue = 0
    Else
        QS20 = QL20.Items(QL20.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL21.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 21."
        continue = 0
    Else
        QS21 = QL21.Items(QL21.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL22.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 22."
        continue = 0
    Else
        QS22 = QL22.Items(QL22.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL23.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 23."
        continue = 0
    Else
        QS23 = QL23.Items(QL23.SelectedIndex).Value
    End If
End If

If continue = 1 Then
    If QL24.SelectedIndex = -1 Then
        litMsgBox.Text += "No value has been selected for question 24."
        continue = 0
    Else
        QS24 = QL24.Items(QL24.SelectedIndex).Value
    End If
End If

If learningManipulationCheck Then
    If continue = 1 Then
        If learningCheck.SelectedIndex = -1 Then
            litMsgBox.Text += "No value has been selected for question 25."
            continue = 0
        Else
            QS25 = learningCheck.Items(learningCheck.SelectedIndex).Value
        End If
    End If
End If

If continue <> 1 Then
    Return False
Else
```

```
            Return True
        End If


    End Function

    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNext.Click

        If CheckForNullValues() Then

            Dim sqlString As String = "" & _
            "UPDATE NegotiationResults " & _
            "SET " & _
            "Q19 = '" & QS12 & "', " & _
            "Q20 = '" & QS13 & "', " & _
            "Q21 = '" & QS14 & "', " & _
            "Q22 = '" & QS15 & "', " & _
            "Q23 = '" & QS16 & "', " & _
            "Q24 = '" & QS17 & "', " & _
            "Q25 = '" & QS18 & "', " & _
            "Q26 = '" & QS19 & "', " & _
            "Q27 = '" & QS20 & "', " & _
            "Q28 = '" & QS21 & "', " & _
            "Q29 = '" & QS22 & "', " & _
            "Q30 = '" & QS23 & "', "

            If learningManipulationCheck Then
                sqlString &= "Q31 = '" & QS24 & "', " & _
                             "Q32 = '" & QS25 & "' " & _
                             "WHERE NegotiationID = '" & NegotiationID & "' "
            Else
                sqlString &= "Q31 = '" & QS24 & "' " & _
                             "WHERE NegotiationID = '" & NegotiationID & "' "
            End If

            Session("NegotiationID") = NegotiationID

            gSession.ExecuteNonQuery(sqlString)
            Dim localhost As String = ConfigurationSettings.AppSettings.Get("URL")
            Response.Redirect(localhost + "Thanks_For_Participation.aspx")

        End If
    End Sub

End Class
```

B.11 SellAgent

(Note: This code has been adapted from the work by Mok and Sundarraj (2005))

```
Option Explicit On

Public Class sellAgent


    Private Const m_id As String = "Seller"

    Private m_NegotiationID As String
    Private m_startPrice As Double
    Private m_minPrice As Double
    'Private m_BMstartPrice As Double
    'Private m_BMminPrice As Double
    'Private m_BMpriceScore As Single
    'Private m_strategy As String
    Private m_askPrice As Double
    Private m_beta As Single
    Private m_Tmax As Integer
    Private m_cumTmax As Integer
    Private m_K As Single
    Private m_estimatedBeta As Single
    Private m_estimatedTmax As Integer
    Private m_estimatedPmin As Double
    Private m_estimatedPmax As Double
    Private m_estimatedK As Single
    Private m_learn As Boolean
    Private m_learnStrategy As String
    Private m_TmaxStrategy As String
    'Private m_decisionScore() As Variant
    'Private m_propertyScore() As Variant
    'Private m_carProperty() As String

    Public gSession As DbSession

    Public Sub New(ByVal a_NegotiationID As String)
        m_NegotiationID = a_NegotiationID
        gSession = New DbSession
        gSession.NegotiationID = m_NegotiationID
    End Sub

    Public Property NegotiationID() As String
        Get
            Return m_NegotiationID
        End Get
        Set(ByVal Value As String)
            m_NegotiationID = Value
        End Set
    End Property
    Public Property cumTmax() As Integer
        Get
            Return m_cumTmax
        End Get
        Set(ByVal x As Integer)
            m_cumTmax = x
        End Set
    End Property

    Public ReadOnly Property ID() As String
        Get
            Return m_id
        End Get
    End Property

    Public Property learn() As Boolean
        Get
            Return m_learn
```

```
    End Get
    Set(ByVal x As Boolean)
        m_learn = x
    End Set
End Property

Public Property learnStrategy() As String
    Get
        Return m_learnStrategy
    End Get
    Set(ByVal x As String)
        m_learnStrategy = x
    End Set

End Property

Public Property TmaxStrategy() As String
    Get
        Return m_TmaxStrategy
    End Get
    Set(ByVal x As String)
        m_TmaxStrategy = x
    End Set

End Property

Public Property estimatedBeta() As Single
    Get
        Return m_estimatedBeta
    End Get
    Set(ByVal x As Single)
        m_estimatedBeta = x
    End Set

End Property

Public Property estimatedTmax() As Integer
    Get
        Return m_estimatedTmax
    End Get
    Set(ByVal x As Integer)
        m_estimatedTmax = x
    End Set

End Property

Public Property estimatedPmin() As Double
    Get
        Return estimatedPmin
    End Get
    Set(ByVal x As Double)
        estimatedPmin = x
    End Set

End Property

Public Property estimatedPmax() As Double
    Get
        Return estimatedPmax
    End Get
    Set(ByVal x As Double)
        estimatedPmax = x
    End Set

End Property

Public Property estimatedK() As Single
    Get
        Return m_estimatedK
    End Get
    Set(ByVal x As Single)
```

```
        m_estimatedK = x
    End Set

End Property

Public Property startPrice() As Double
    Get
        Return m_startPrice
    End Get
    Set(ByVal x As Double)
        m_startPrice = x
    End Set

End Property

Public Property minPrice() As Double
    Get
        Return m_minPrice
    End Get
    Set(ByVal x As Double)
        m_minPrice = x
    End Set

End Property

'Public Property Get BMstartPrice() As Double
'    BMstartPrice = m_BMstartPrice
'End Property
'Public Property Let BMstartPrice(x As Double)
'    m_BMstartPrice = x
'End Property
'Public Property Get BMminPrice() As Double
'    BMminPrice = m_BMminPrice
'End Property
'Public Property Let BMminPrice(x As Double)
'    m_BMminPrice = x
'End Property
'Public Property Get decisionScore() As Variant
'    decisionScore = m_decisionScore
'End Property
'Public Property Let decisionScore(x As Variant)
'    m_decisionScore = x
'End Property
'Public Property Get BMpriceScore() As Single
'    BMpriceScore = m_BMpriceScore
'End Property
'Public Property Let BMpriceScore(x As Single)
'    m_BMpriceScore = x
'End Property
'Public Property Get strategy() As String
'    strategy = m_strategy
'End Property
'Public Property Let strategy(x As String)
'    m_strategy = x
'End Property
Public Property askPrice() As Double
    Get
        Return m_askPrice
    End Get
    Set(ByVal x As Double)
        m_askPrice = x
    End Set

End Property

'Public Property Get carProperty() As Variant
'    carProperty = m_carProperty
'End Property
'Public Property Let carProperty(x As Variant)
'    m_carProperty = x
'End Property
```

```
    'Public Property Get propertyScore() As Variant
    '    propertyScore = m_propertyScore
    'End Property
    'Public Property Let propertyScore(x As Variant)
    '    m_propertyScore = x
    'End Property
    Public Property beta() As Single
        Get
            Return m_beta
        End Get
        Set(ByVal x As Single)
            m_beta = x
        End Set

    End Property

    Public Property Tmax() As Integer
        Get
            Return m_Tmax
        End Get
        Set(ByVal x As Integer)
            m_Tmax = x
        End Set

    End Property

    Public Property K() As Single
        Get
            Return m_K
        End Get
        Set(ByVal x As Single)
            m_K = x
        End Set

    End Property

    Public Function Estimate(ByVal neg As Negotiation, ByVal priceArray() As Double) As
Boolean
        Dim two, three, four As Boolean
        Dim t0, t1, t2, t3, t4 As Integer
        Dim P0, P1, P2, P3, P4 As Double
        Dim beta As Single
        Dim Pmin As Double
        Dim Pmax As Double
        Dim Tmax As Integer
        Dim K As Single
        Dim cumBeta As Double
        Dim cumTmax As Integer
        Dim cumPmin As Double
        Dim cumPmax As Double
        Dim cumK As Double
        Dim sSQL As String
        Dim temp As Single
        Dim cnt As Integer
        Dim i As Integer
        'Dim conn As New ADODB.Connection
        'Dim rs As New ADODB.Recordset
        'conn = gSession.GetConn

        Dim strConn As String = gSession.connectionString

        Dim ocmd As SqlCommand


        t0 = 0
        t1 = 1
        t2 = 2
        t3 = 3
        t4 = 4
        P0 = priceArray(0)
        P1 = priceArray(1)
```

```
        P2 = priceArray(2)
        P3 = priceArray(3)
        P4 = priceArray(4)

        sSQL = "SELECT * FROM Criteria ORDER BY Criteria"
        'rs.Open(sSQL, conn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockReadOnly)
        'rs.MoveFirst()
        Try
            ocmd = New SqlCommand
            Dim odtr As SqlDataReader
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = sSQL
                odtr = .ExecuteReader()
            End With

            Dim rs_criteria As Integer
            Do Until cnt >= 20 Or Not odtr.Read 'rs.EOF
                cnt = 0
                cumBeta = 0
                cumTmax = 0
                cumPmin = 0
                cumPmax = 0
                cumK = 0
                rs_criteria = odtr.GetValue(1)

                For Tmax = (t4 + 1) To 25 Step 1
                    For Pmax = Round((P4 + 1), 0) To 300 Step 2
                        For Pmin = 1 To Round((P0 - 1), 0) Step 2
                            K = (P0 - Pmin) / (Pmax - Pmin)
                            beta = (Log(Log((P1 - Pmin) / (Pmax - Pmin)) / Log(K)) /
Log(10)) / (Log(1 - t1 / Tmax) / Log(10))
                            K = Round(K, 3)
                            beta = Round(beta, 3)
                            temp = Pmin + (Exp(((1 - t2 / Tmax) ^ beta) * Log(K)) * (Pmax
- Pmin))
                            If Abs(temp - P2) < rs_criteria Then
                                two = True
                            Else
                                two = False
                                GoTo Nxt
                            End If
                            temp = Pmin + (Exp(((1 - t3 / Tmax) ^ beta) * Log(K)) * (Pmax
- Pmin))
                            If Abs(temp - P3) < rs_criteria Then
                                three = True
                            Else
                                three = False
                                GoTo Nxt
                            End If
                            temp = Pmin + (Exp(((1 - t4 / Tmax) ^ beta) * Log(K)) * (Pmax
- Pmin))
                            If Abs(temp - P4) < rs_criteria Then
                                four = True
                            Else
                                four = False
                                GoTo Nxt
                            End If

                            If two = True And three = True And four = True Then
                                cnt = cnt + 1
                                cumBeta = cumBeta + beta
                                cumTmax = cumTmax + Tmax
                                cumPmin = cumPmin + Pmin
                                cumPmax = cumPmax + Pmax
                                cumK = cumK + K
                            End If
Nxt:                    Next Pmin
                    Next Pmax
```

```
                Next Tmax
                'rs.MoveNext()
            Loop

        Catch oexpData As OleDb.OleDbException
            ' oexpData.Message

        End Try

        ocmd.Connection.Close()
        ocmd.Dispose()

        If cnt >= 20 Then
            m_estimatedBeta = Round(cumBeta / cnt, 3)
            m_estimatedTmax = Round(cumTmax / cnt)
            m_estimatedPmin = Round(cumPmin / cnt, 2)
            m_estimatedPmax = Round(cumPmax / cnt, 2)
            m_estimatedK = Round(cumK / cnt, 3)
            Estimate = True

            Dim t As Integer
            Dim PredictedPrice As Double
            sSQL = "DELETE FROM PredictedBuyer WHERE turn >= " & (neg.cumSellerTurn - 5)
& " AND NegotiationID = '" & m_NegotiationID & "' "
            gSession.ExecuteNonQuery(sSQL)
            For t = 0 To m_estimatedTmax Step 1
                PredictedPrice = m_estimatedPmin + Exp(((1 - t / m_estimatedTmax) ^
m_estimatedBeta) * Log(m_estimatedK)) * (m_estimatedPmax - m_estimatedPmin)
                sSQL = "INSERT INTO PredictedBuyer ( NegotiationID, turn, Price )
VALUES('" & m_NegotiationID & "', " & (t + neg.cumSellerTurn - 5) & ", " & PredictedPrice
& ")"
                gSession.ExecuteNonQuery(sSQL)
            Next t
        Else
            Estimate = False
        End If




    End Function
    Public Sub FindOptimal(ByRef neg As Negotiation)
        Dim Tmax As Integer
        Dim turn As Integer
        Dim beta As Single
        Dim target As Double
        Dim newStartPrice As Double
        Dim i As Integer
        Dim t As Integer
        For i = m_estimatedTmax To 5 Step -1
            target = m_estimatedPmin + (Exp(((1 - i / estimatedTmax) ^ m_estimatedBeta) *
Log(m_estimatedK))) * (m_estimatedPmax - m_estimatedPmin)
            If (neg.nextSellerLearn - neg.lastSellerLearn) = 4 Then
                If i = 5 Then GoTo Nx
                t = i - 1
            ElseIf (neg.nextSellerLearn - neg.lastSellerLearn) = 5 Then
                t = i
            End If
            If m_TmaxStrategy = "Fix" Then
                Tmax = m_Tmax
            ElseIf m_TmaxStrategy = "UnFix" Or m_TmaxStrategy = "Unfix" Then
                If m_Tmax <= m_estimatedTmax Then
                    Tmax = t + 1
                Else
                    Tmax = m_Tmax
                End If
            End If

            If t = Tmax And m_minPrice = target Then
                Exit Sub
            End If
```

```
            If target < m_minPrice Or t >= Tmax Then
                GoTo Nx
            Else
                If (neg.nextSellerLearn - neg.lastSellerLearn) = 5 Then
                    Tmax = Tmax - 5
                    turn = t - 5
                ElseIf (neg.nextSellerLearn - neg.lastSellerLearn) = 4 Then
                    Tmax = Tmax - 4
                    turn = t - 4
                End If
                newStartPrice = m_askPrice
                'newStartPrice = (m_bidPrice - m_K * m_maxPrice) / (1 - m_K)
                If newStartPrice - target <= 0 Then GoTo Nx
                beta = (Log(Log(1 - ((target - m_minPrice) / (newStartPrice -
m_minPrice))) / Log(m_K)) / Log(10)) / (Log(1 - turn / Tmax) / Log(10))
                If beta > 0 Then
                    m_startPrice = newStartPrice
                    m_cumTmax = Tmax + neg.cumSellerTurn
                    neg.sellerTurn = 0
                    m_beta = beta
                    m_Tmax = Tmax
                    If m_learnStrategy = "Subsequent" Then
                        If (neg.nextSellerLearn - neg.lastSellerLearn) = 5 Then
                            neg.nextSellerLearn = neg.cumSellerTurn + 4
                        ElseIf (neg.nextSellerLearn - neg.lastSellerLearn) = 4 Then
                            neg.nextSellerLearn = neg.cumSellerTurn + 5
                        End If
                    End If
                    neg.lastSellerLearn = neg.cumSellerTurn
                    Exit Sub
                End If
            End If

Nx:     Next i
        If m_learnStrategy = "Subsequent" Then
            If (neg.nextSellerLearn - neg.lastSellerLearn) = 5 Then
                neg.nextSellerLearn = neg.cumSellerTurn + 4
            ElseIf (neg.nextSellerLearn - neg.lastSellerLearn) = 4 Then
                neg.nextSellerLearn = neg.cumSellerTurn + 5
            End If
        End If
        neg.lastSellerLearn = neg.cumSellerTurn
    End Sub

End Class
```

B.12 Thanks for Participation

```
Public Class Thanks_For_Participation
    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents litMsg As System.Web.UI.WebControls.Literal

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        DisplayConfirmation()
        FormsAuthentication.SignOut()


    End Sub

    Sub DisplayConfirmation()
        Dim lSession As DbSession = New DbSession

        Dim strConn As String = lSession.connectionString
        Dim strLogin As String = HttpContext.Current.User.Identity.Name
        Dim ocmd As SqlCommand
        Dim odtr As SqlDataReader
        Dim compare As String = ""
        Dim strCmdText = "SELECT ConfirmationNumber FROM UNIQUEVALTABLE WHERE USERID = '"
& strLogin & "' "

        Try
            ocmd = New SqlCommand
            With ocmd
                .Connection = New SqlConnection(strConn)
                .Connection.Open()
                .CommandText = strCmdText
                odtr = .ExecuteReader
            End With

            'Dim numrows As Integer = ocmd.ExecuteNonQuery()
            odtr.Read()
            compare = odtr.GetValue(0)

        Catch oexpData As Exception
            Return

        End Try



        'odtrAboutVBDataReader.Close()
        ocmd.Connection.Close()
        ocmd.Dispose()
```

```
        If Not compare = "" Then
            litMsg.Text = " Your confirmation code is '" & compare & "'."
            strCmdText = "UPDATE UNIQUEVALTABLE SET COMPLETED = '1' WHERE USERID = '" &
strLogin & "' "
            lSession.ExecuteNonQuery(strCmdText)
            Return
        Else
            Return
        End If
    End Sub

End Class
```

Appendix C: Powerpoint Presentation

Figure C-1 Slide 1

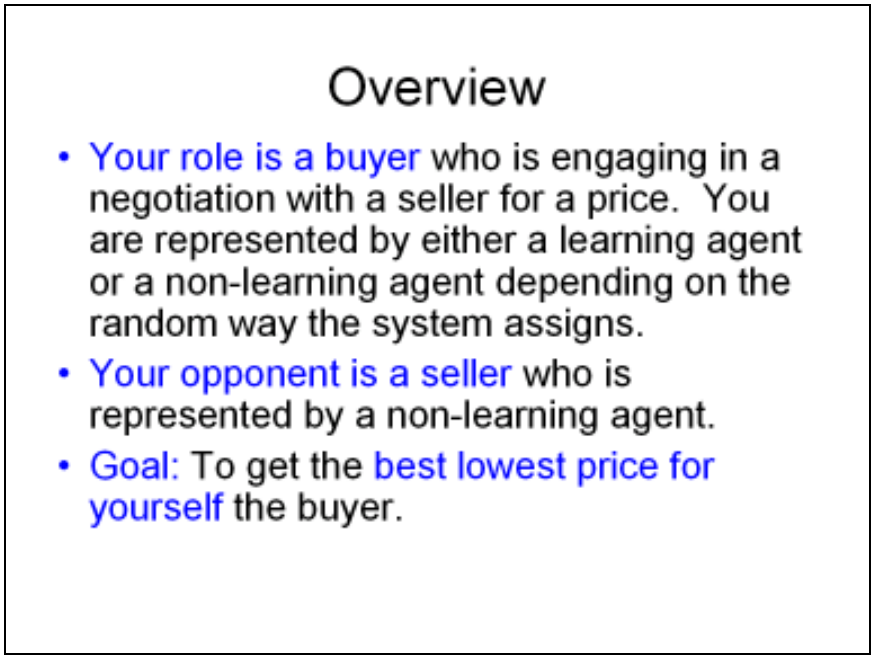


Figure C-2 Slide 2

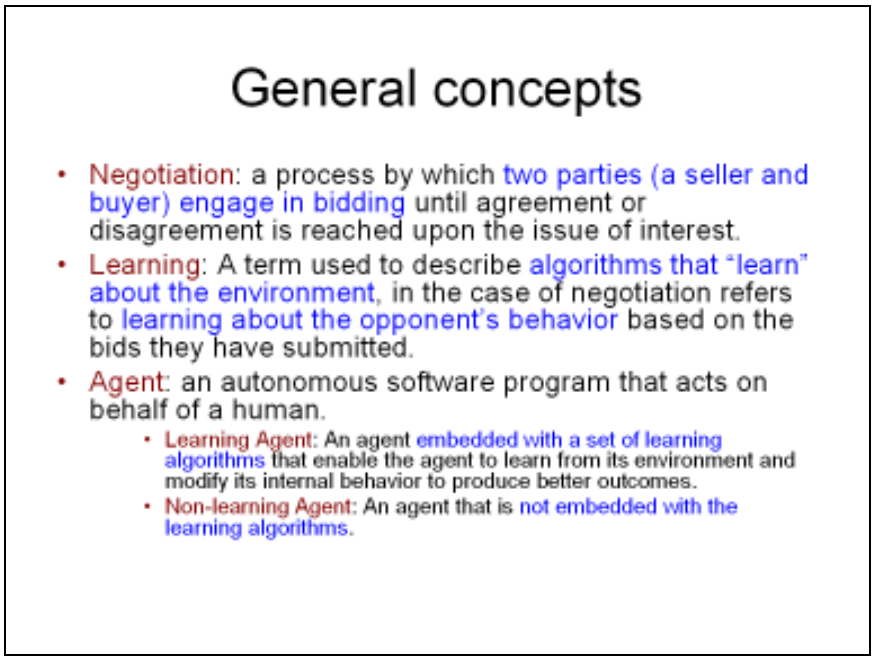


Figure C-3 Slide-3

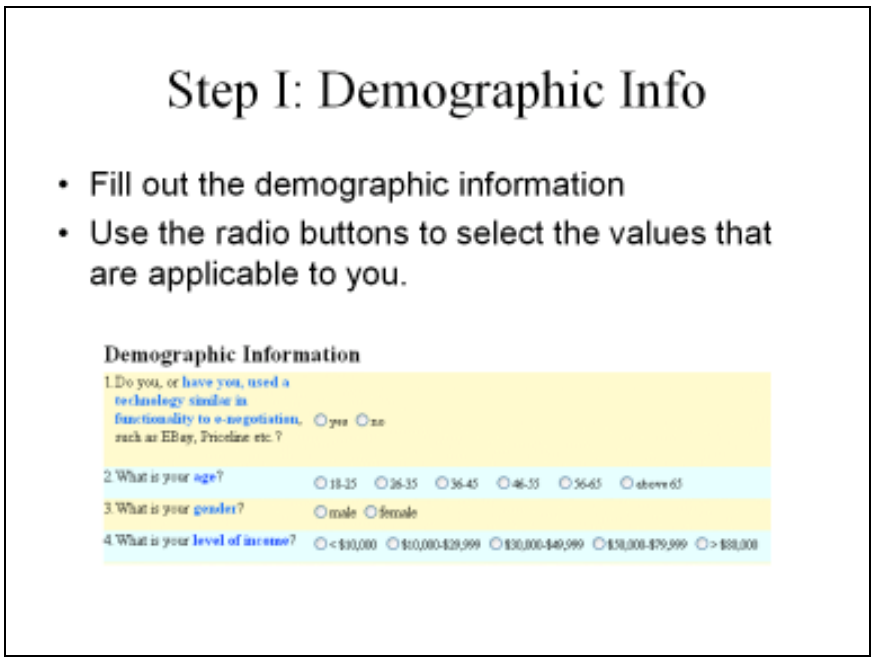


Figure C-4 Slide-4

Step II: Negotiation

- At this stage you will begin the actual negotiation.
- You will need to initialize your agent first.
- The negotiation will begin when seller's non-learning agent will submit a price bid that it considers a good price.
- You can respond by either submitting a price bid yourself **OR** allowing your agent to submit a bid for you.
- Your agent can be either learning or non-learning depending on how the system randomly assigns.

Figure C-5 Slide-5

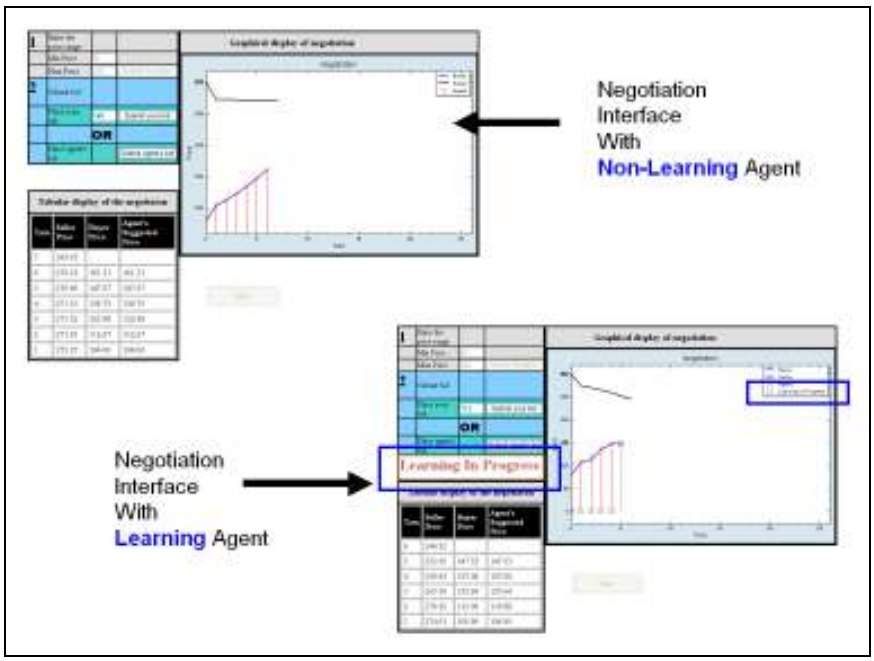


Figure C-6 Slide-6

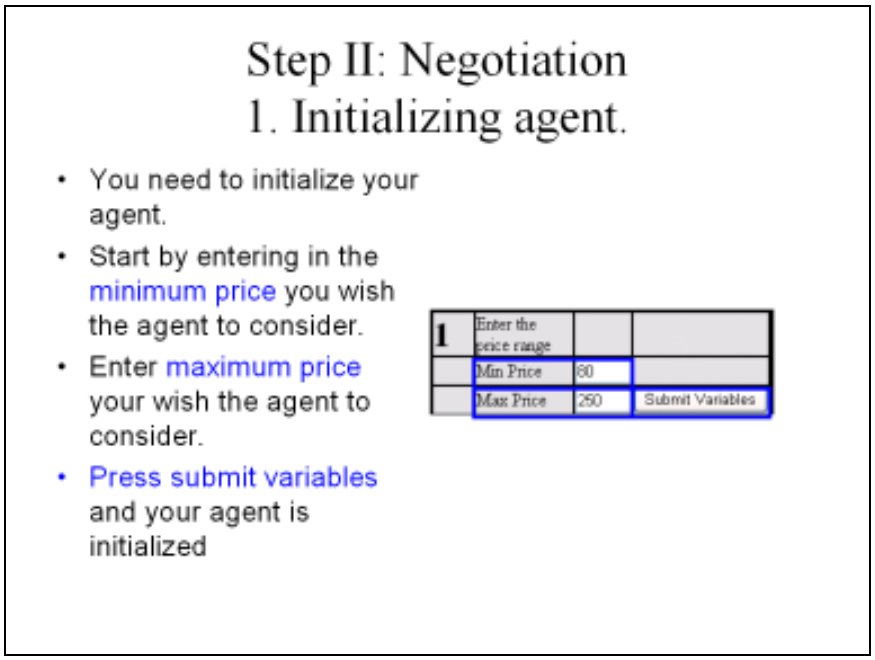


Figure C-7 Slide-7

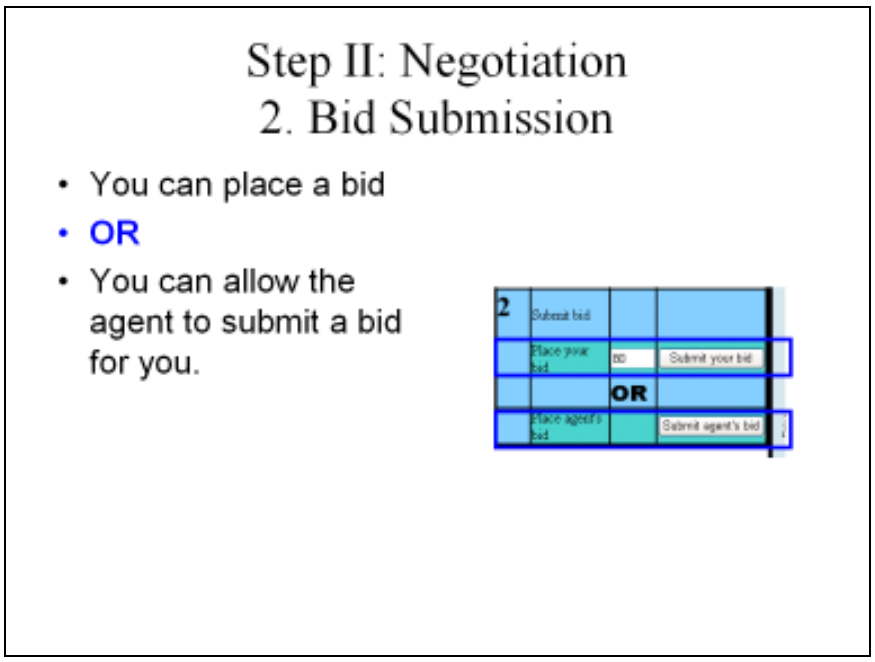


Figure C-8 Slide-8

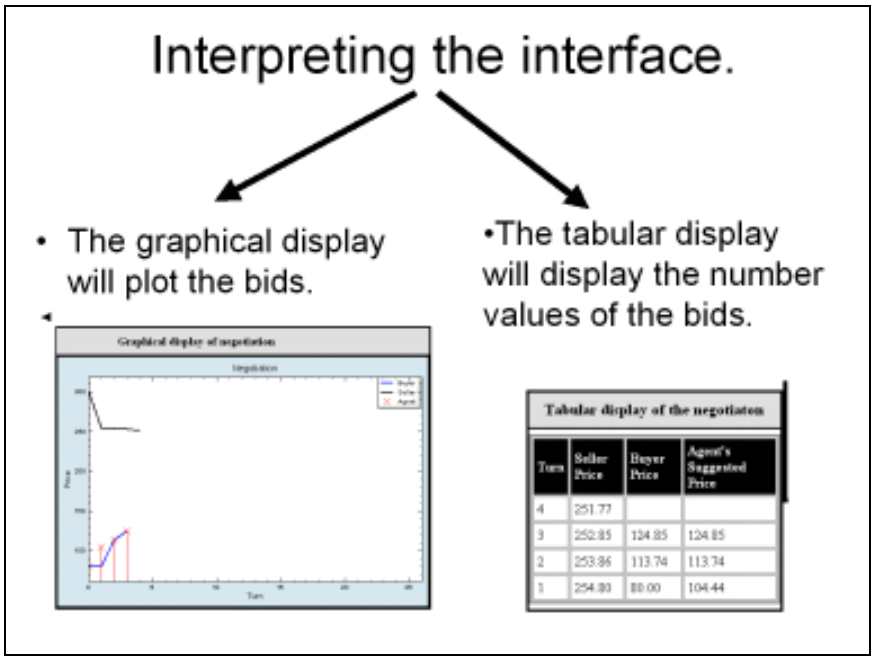


Figure C-9 Slide-9

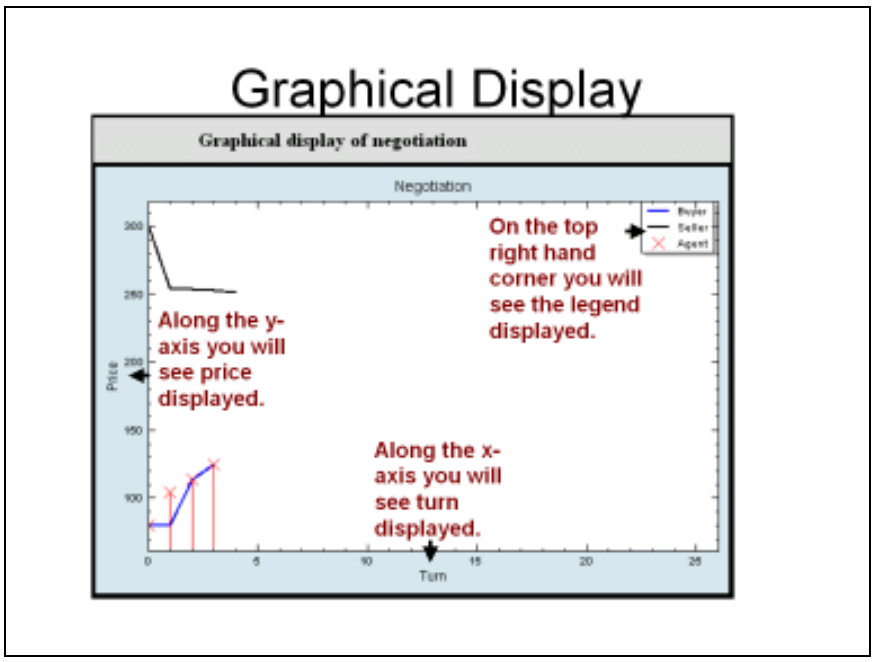


Figure C-10 Slide-10

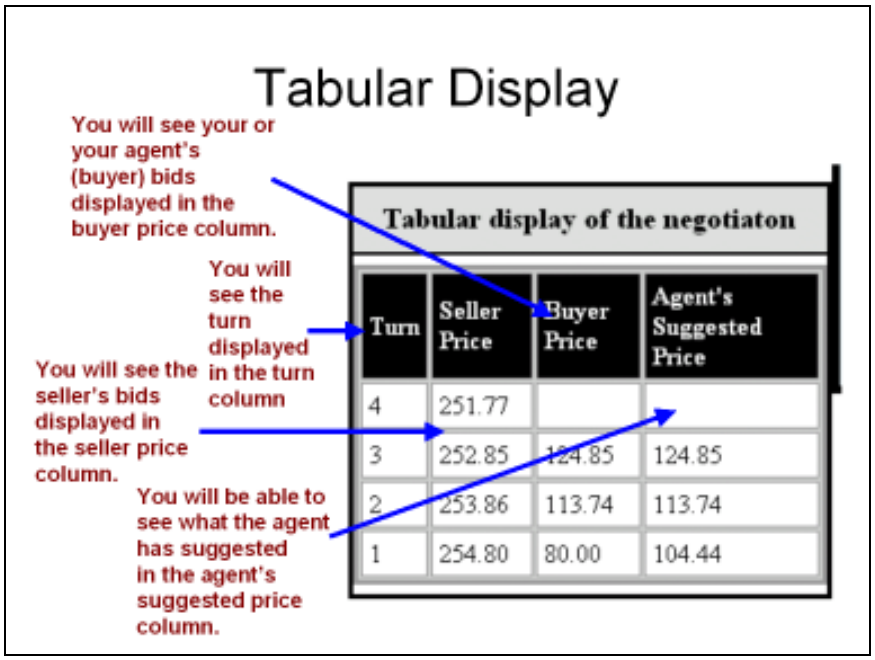


Figure C-11 Slide-11

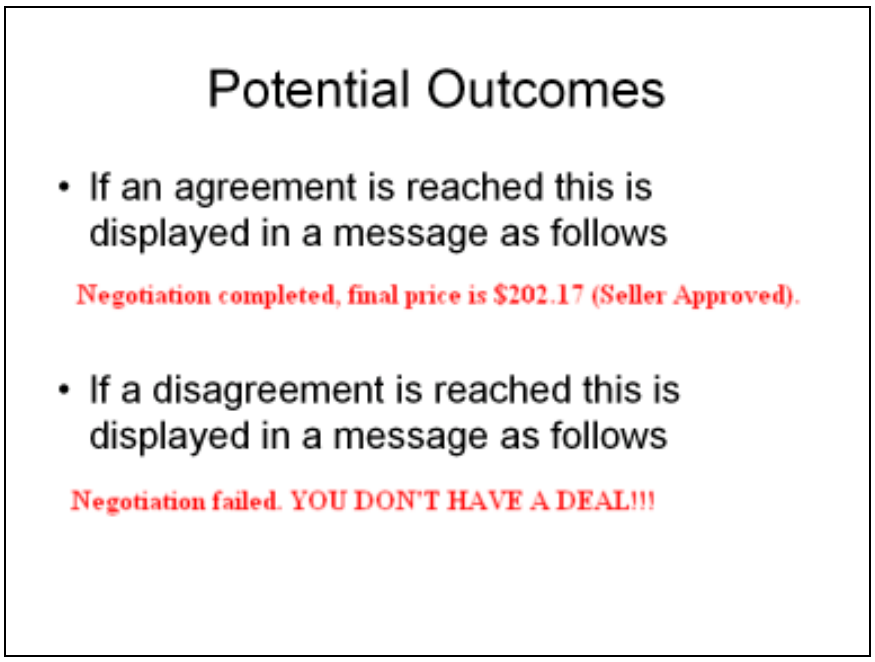


Figure C-12 Slide-12

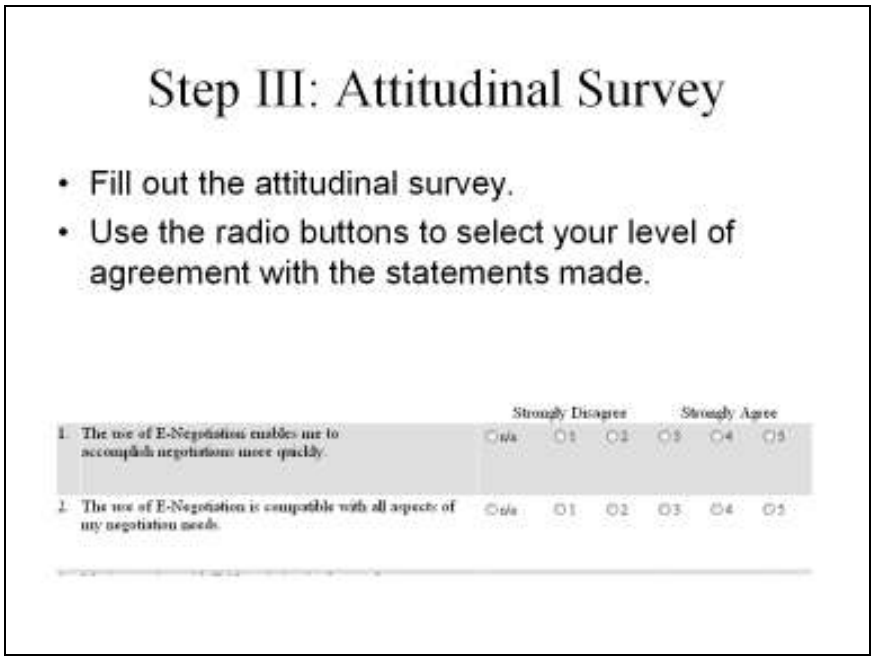


Figure C-13 Slide-13

Appendix D: Letters and Promotion

Appendix D.1 contains the survey promotion used to promote the survey. Appendix D.2 contains the email recruitment sent to managers from business organizations to seek approval to recruit from their employees. Appendix D.3 contains the email recruitment sent to employees to ask for their participation in the survey.

D.1 Survey Promotion

**** SURVEY PROMOTION ****
Department of Management Sciences
University of Waterloo
PARTICIPANTS NEEDED FOR
RESEARCH IN E-NEGOTIATION
We are looking for volunteers to take part in a study of
E-NEGOTIATION.
As a participant in this study, you would be asked to
provide some basic demographic information, use an e-negotiation tool and
fill out an online survey.
Your participation would involve 1 session,
which is approximately 20 minutes.
In appreciation for your time upon completion of the survey, you will receive
a gift certificate to Tim Hortons.
Your participation in this study is entirely voluntary and can be cancelled at anytime.
This study has been reviewed by, and received ethics clearance
through, the Office of Research Ethics, University of Waterloo.
For more information about this study, or to volunteer for this study,
please contact:
Vadi Visuvalingam
Email: v2visuva@engmail.uwaterloo.ca
Phone: (519) 888-4567 Ext. 6099.
**** SURVEY PROMOTION ****

D.2 Email Recruitment (Managers)

<Business Address>

Dear <Name of Manager>,

I am a master's student in the Department of Management Sciences at the University of Waterloo working under the supervision of Prof. R. P. Sundarraj. I am conducting research on the factors that influence the adoption of e-negotiation. E-negotiation is a new technology that can be used in e-commerce and we would like to study what factors would influence individuals to adopt a technology such as this. I would appreciate an opportunity to recruit employees of your organization to participate in the research study being conducted.

We intend to contact around 15 – 20 employees in your organization who have some exposure to the use of new technologies. We will ask that they complete, online, a basic demographic questionnaire, use an e-negotiation tool and complete an attitudinal survey. The participation of the employees is entirely voluntary and can be withdrawn at anytime. Information is obtained anonymously; no personal identifiers are attached to the information collected.

I would appreciate if you would permit me to recruit participants from your employees. The completion of the study will take no more than 20 minutes of their time and they will receive a 2 dollar gift certificate to Tim Hortons as a token of our appreciation.

The questions asked are quite general or statements of attitude that require a response of level of agreement. Participants may omit any question they prefer not to answer by selecting the n/a value for that question. There are no known or anticipated risks to participation in this study. Participation in this project is voluntary and anonymous. Further, all information provided by your employees will be considered confidential. The data collected through this study will not be associated with the machine identifiers from the computers that are used to enter the information. The information will be stored for a period of 10 years in a password protected computer in a locked computer room in the department of Management Sciences at the University of Waterloo.

If any of your employees are interested in participating in this study, please ask them to contact me at v2visuva@engmail.uwaterloo.ca or (519) 888-4567 Ext 6099 to acquire the necessary information to participate or for more information. They may also direct further questioning to my faculty supervisor Dr. Sundarraj at rsundarr@engmail.uwaterloo.ca or (519) 888-4567 ext. 2235.

I would like to assure you that this study has been reviewed and received ethics clearance through the Office of Research Ethics at the University of Waterloo. However, the final decision about participation is the employees. Should they have any comments or concerns resulting from their participation in this study, please ask them to contact Dr.

Susan Sykes in the Office of Research Ethics at (519) 888-4567 Ext. 6005.

Thank you for this opportunity to recruit from your organization.

Yours sincerely,
Vadivananthan Visuvalingam
Candidate for M.A.Sc
University of Waterloo

D.3 Email Recruitment (Employees)

<Business Address>

Dear <Name of Employee>,

We have been given permission by your manager, <Manager's name>, to contact you to ask for you participation in a research study. This research study is in no way associated with your company or affects your employment status. I am a master's student in the Department of Management Sciences at the University of Waterloo working under the supervision of Prof. R. P. Sundarraj. I am conducting research on the factors that influence the adoption of e-negotiation. E-negotiation is a new technology that can be used in e-commerce and we would like to study what factors would influence individuals to adopt a technology such as this. I would appreciate it if you would volunteer to participate in this study.

In order to participate in this study, you need to have had some exposure to the use of new technologies. We will ask you to complete, online, a basic demographic questionnaire, use an e-negotiation tool and complete an attitudinal survey. Your participation is entirely voluntary and can be withdrawn at anytime. Information is provided anonymously; no personal identifiers are attached to the information collected.

I would appreciate if you would choose to participate in this study. The completion of the study will take no more than 20 minutes of your time and you will receive a 2 dollar gift certificate to Tim Hortons as a token of our appreciation.

The survey requests your level of agreement with statements of attitudes. You may omit any question you prefer not to answer by selecting the n/a value for that question. There are no known or anticipated risks to participation in this study. Participation in this project is voluntary and anonymous. Further, all information provided by you will be considered confidential. The data collected through this study will not be associated with the machine identifiers from the computers that are used to enter the information. The information will be stored for a period of 10 years in a password protected computer in a locked computer room in the department of Management Sciences at the University of Waterloo.

If you are interested in participating in this study, please contact me at v2visuva@engmail.uwaterloo.ca or (519) 888-4567 Ext 6099 to acquire the necessary information to participate or for more information. You may also direct further questioning to my faculty supervisor Dr. Sundarraj at rsundarr@engmail.uwaterloo.ca or (519) 888-4567 ext. 2235.

I would like to assure you that this study has been reviewed and received ethics clearance through the Office of Research Ethics. However, the final decision about participation is yours. Should you have any comments or concerns resulting from

your participation in this study, please contact Dr. Susan Sykes in the Office of Research Ethics at 888-4567 Ext. 6005.

Thank you.

Yours sincerely,
Vadivananthan Visuvalingam
Candidate for M.A.Sc
University of Waterloo

References

Adams, D., Nelson, R., & Todd, A. (1992). Perceived usefulness, ease of use and usage of information technology: A replication. *MIS Quarterly, 16*(2), 227-247.

Ajzen, I. (1991). The theory of planned behaviour. *Organizational Behaviour and Human Decision Processes, 50*(2), 179-211.

Ajzen, I. (1985). From intentions to actions: A theory of planned behaviour. In J. Kuhl, & J. Beckmann (Eds.), *Action control: From cognition to behaviour* (pp. 11-39). Springer Verlag, New York:

Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behaviour*. Englewood Cliffs, New Jersey: Prentice-Hall.

Ajzen, I., & Madden, T. (1986). Prediction of goal-directed behaviour: Attitudes, intentions and perceived behavioural control. *Journal of Experimental Social Psychology, 22*, 453-474.

Balzer, W., Doherty, M., & O'Connor, R. (1989). Effects of cognitive feedback on performance. *Psychological Bulletin, 106*(3), 410-433.

Beam, C., & Segev, A. (1997). Automated negotiations: A survey of the state of the art. [Electronic version]. *Wirtschaftsinformatik, 39*(3), 263-268. Retrieved April 27, 2006, from CiteSeer database.

Benbasat, I., & Lim, L. (1993). The effects of group, task, context, and technology variables on the usefulness of group support systems. A meta-analysis of experimental studies. *Small Group Research, 24*, 430-462.

Benham, H., & Raymond, B. (1996). Information technology adoption: Evidence from a voice mail introduction. *Computer Personnel,* , 3-25.

Bichler, M., Kalagnanam, J., Katircioglu, K., King, A., Lawrence, R., & Lee, H. et al. (2002). Applications of flexible pricing in business-to-business electronic commerce. *IBM Systems Journal, 41*(2), 287-302.

Campbell, D., & Fiske, D. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin, 56*, 81-104.

Chau, P., & Hu, P. (2001). Information technology acceptance by individual professionals: A model comparison approach. *Decision Sciences, 32*(4), 699-719.

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

Chin, W. (2006). *The role of impression management in goal setting.* Unpublished Masters, University of Waterloo.

Choi, S., Liu, J., & Chan, S. (2001). A genetic agent-based negotiation system. *Computer Networks, 37*, 195-204.

Couper, M. (2000). Web surveys: A review of issues and approaches. *Public Opinion Quarterly, 64*(4), 464-494.

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

Davis, F., Bagozzi, R., & Warshaw, P. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science, 35*(8), 982-1003.

Deveaux, L., Paraschiv, C., & Latourrette, M. (2001). Bargaining on an internet agent-based market: Behavioural vs. optimizing agents. *Electronic Commerce Research, 1*(4), 371-401.

Faratin, P., Sierra, C., & Jennings, N. (1998). Negotiation decision functions for autonomous agents. *Robotics Autonomous Systems, 24*(3), 159-182.

Hair, J., Andersin, R., Tatham, R., & Black, W. (1992). *Multivariate data analysis with readings*. New York: Macmillan.

Hammond, K., McClelland, G., & Mumpower, J. (1980). *Human judgment and decision making*. New York: Praeger.

Hansen, T., Jensen, J., & Solgaard, H. (2004). Predicting online grocery buying intention: A comparison of the theory of reasoned action and the theory of planned behaviour. *International Journal of Information Management, 24*(6), 539-550.

Huhns, M., & Singh, M. (1998). *Readings in agents*. Los Altos, CA: Morgan Kaufmann.

Jackson, C., Chow, S., & Leitch, R. (1997). Toward an understanding of the behavioural intention to use an information system. *Decision Sciences, 28*(2), 357-389.

Judd, C., Smith, E., & Kidder, L. (1986). *Research methods in social relations*. New York: Holt, Rinehart and Winston.

Karahanna, E., Straub, D., & Chervany, N. (1999). Information technology adoption across time: A cross-sectional comparison of pre-adoption and post-adoption beliefs. *MIS Quarterly, 23*(2), 183-213.

Kersten, G. (2002). *The science and engineering of E-negotiation: Review of the emerging field.* Unpublished manuscript.

Koeszegi, S., Vetschera, R., & Kersten, G. (2004). National cultural differences in the use and perception of internet-based NSS: Does high or low context matter? *International Negotiation, 9*(1), 79-109.

Liao, S., Shao, Y., Wang, H., & Chen, A. (1999). The adoption of virtual banking: An empirical study. *International Journal of Information Management, 19*, 63-74.

Mathieson, K. (1991). Predicting user intentions: Comparing the technology acceptance model with the theory of planned behaviour. *Information Systems Research, 2*(3), 173-191.

Mok, W., & Sundarraj, R. (2005). Learning algorithms for single-instance electronic negotiations using the time-dependent behavioural tactic. *ACM Transactions on Internet Technology, 5*(1), 195-230.

Moore, G., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research, 2*(3), 192-222.

Nadler, J., Thompson, L., & van Boven, L. (2003). Learning negotiation skills: Four models of knowledge creation and tranfer. *Management Science, 49*(4), 529-540.

Nash, J. (1950). The bargaining problem. *Econometrica, 18*, 155-162.

Norusis, M. (1994). *SPSS professional statistics 6.1*SPSS Inc.

Plouffe, C., Hulland, J., & Vandenbosch, M. (2001). Research report: Richness versus parsimony in modeling technology adoption decisions--understanding merchant adoption of a smart card-based payment system. *Information Systems Research, 12*(2), 208-222.

Raiffa, H. (1982). *The art and science of negotiation*. MA: Harvard University Press.

Riemenschneider, C., Harrison, D., & Mykytn, P. (2003). Understanding IT adoption decisions in small business: Integrating current theories. *Information and Management, 40*(4), 269-285.

Rogers, E. (1983). *Diffusion of innovations* (Third Edition ed.). New York: The Free Press.

Schoop, M., Jertila, A., & Thomas, L. (2003). Negoisst: A negotiation support system for electronic business-to-business negotiations in e-commerce *Data and Knowledge Engineering, 47*(3), 371-401.

Schoop, M., & Quix, C. (2001). DOC.COM: A framework for effective negotiation support in electronic marketplaces *Computer Networks, 37*(2), 153-170.

Sebenius, J. (1992). Negotiation analysis: A characterization and review. *Management Science, 38*(1), 18-38.

Segars, A., & Grover, V. (1993). Re-examining perceived ease of use and usefulness: A confirmatory factor analysis. *MIS Quarterly, 17*(4), 517-525.

Strobel, M. (2002). An XML schema representation for the communication design of electronic negotiations *Computer Networks, 39*(5), 661-680.

Sundarraj, R., & Wu, J. (2005). Using information-systems constructs to study online- and telephone-banking technologies. *Electronic Commerce Research and Applications, 4*(4), 427-443.

Szajna, B. (1996). Empirical evaluation of the revised technology acceptance model. *Management Science, 42*(1), 85-92.

Taylor, S., & Todd, P. (1995). Understanding information technology usage: A test of competing models. *Information Systems Research, 6*(1), 144-176.

Thompson, L., & Nadler, J. (2002). Negotiating via information technology: Theory and application. *Journal of Social Studies, 58*, 109-124.

Venkatesh, V. (1999). Creation of favorable user perceptions: Exploring the role of intrinsic motivation. *MIS Quarterly, 23*(2), 239-260.

Venkatesh, V., & Davis, D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science, 46*(2), 186-204.

Wang, Y., Tan, K., & Ren, J. (2005). Towards autonomous and automatic evaluation and negotiation in agent-mediated internet marketplaces. *Electronic Commerce Research, 5*, 343-365.

Wasfy, A., & Yasser, A. (1998). Two-party negotiation modeling: An integrated fuzzy logic approach. *Group Decision and Negotiation, 7*, 491-518.

Wooldridge, M., & Jennings, N. (1995). Agent theories, architectures and languages: A survey. *Knowledge Engineering Review, 10*(2), 115-152.

Yuan, Y., Rose, J., & Archer, N. (1998). A web-based negotiation support system. *Electronic Markets, 8*(3), 13-17.

Zeng, D., & Sycara, K. (1998). Bayesian learning in negotiation. *International Journal of Human-Computer Studies, 48*, 125-141.