ELICITATION OF CONSTRAINTS AND QUALITATIVE PREFERENCES IN MULTI-ATTRIBUTE AUCTIONS

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

University of Regina

By

Shubhashis Kumar Shil

Regina, Saskatchewan

November 2013

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Shubhashis Kumar Shil, candidate for the degree of Master of Science in Computer Science, has presented a thesis titled, ***Elicitation of Constraints and Qualitative Preferences in Multi-Attribute Auctions,*** in an oral examination held on November 26, 2013. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr. Andrei Volodin, Department of Mathematics & Statistics
Supervisor:	Dr. Samira Sadaoui, Department of Computer Science
Committee Member:	Dr. Malek Mouhoub, Department of Computer Science
Committee Member:	*Dr. Lisa Fan, Department of Computer Science
Chair of Defense:	Dr. Yasser Morgan, Faculty of Engineering & Applied Science

*Not present at defense

Abstract

Multi-Attribute Reverse Auctions (MARAs) allow negotiation between the buyer and sellers over price along with other non-price attributes such as delivery time, sellers' reputation, and product quality. The seller with the most preferable product specification over multiple attributes wins. Much research has been done in the context of MARAs, but several challenging problems still need to be addressed. Eliciting the buyer's requirements, which consist of constraints and preferences, and determining the winner, which has been shown to be computationally complex, are hot research topics. Multi-Attribute Utility Theory (MAUT), which has been used in MARAs, only permits the buyer to specify his preferences quantitatively. Often, the buyer is more comfortable expressing his preferences about the product qualitatively. Moreover, there should be options for the buyer to specify constraints. As well, the constraints and preferences can both be non-conditional or conditional. On the other hand, for the purpose of efficiency, it is more suitable for the auction systems to process quantitative data. Hence, there are remaining challenges to provide the buyer with more comfort as well as to keep the system efficient. Motivated by these issues, we develop a novel MARA protocol in which the buyer can specify his non-conditional and conditional constraints as well as qualitative non-conditional and conditional preferences. Moreover, we enhance MAUT by incorporating new algorithms and conversion methods that can convert qualitative values into quantitative ones. In this way, we not only provide the buyer with more comfort, but also keep the system efficient. We design the MARA protocol with 3-layer software architecture based on the Multi-Agent technology and the Belief-Desire-Intention model. We implement it using the agent simulation framework named Jadex. The system assists the buyer to specify his requirements step by step through friendly graphical user interfaces and assists the sellers to submit bids. We conduct a case study of a reverse auction over ten attributes to demonstrate the feasibility of our MARA system. We report on a series of experiments that show that the system is able to elicit the buyer's requirements adequately and to determine the winner efficiently, in terms of processing time.

Acknowledgments

My first debt of gratitude goes to my supervisor Dr. Samira Sadaoui for providing me with constant support during my MSc studies. I would like to express my sincere thanks for her valuable guidance, financial assistance and constant encouragement. Her enthusiasm, patience and diverse knowledge helped and enlightened me on many occasions. She not only provided me with fruitful ideas but also gave me the freedom to think independently. I feel proud to be her research student and can not imagine a better supervisor.

I acknowledge the Faculty of Graduate Studies and Research for providing me with the financial means, in the form of scholarships, to complete this study.

I also thank the committee members who read my thesis and providing suggestions and useful comments for improvement.

I would like to take this opportunity to thank every member of the Department of Computer Science who has helped me throughout my studies.

Last but not least, I would like to extend my deepest gratitude to my beloved parents for their unconditional love and support throughout my entire life.

Table of Contents

Abstract i

Acknowledgments iii

Table of Contents iv

List of Tables vii

List of Figures ix

Chapter 1 Introduction 1

1.1 Problem and Motivation 1

1.2 Proposed Solution 4

1.2.1 Objectives 4

1.2.2 Features of MARA 4

1.2.3 Contributions 6

1.3 Organization of Thesis 9

Chapter 2 Related Works 10

2.1 A Survey on Auctions 10

2.2 Multi-Attribute Utility Theory 15

2.3 Preference and Constraint Elicitation 17

2.4 Multi-Agent Systems 20

Chapter 3 Multi-Attribute Reverse Auction (MARA) Protocol 22

3.1 Multi-Attribute Reverse Auction Definition 23

3.2 Buyer's Requirements 24

3.2.1 Buyer's Ordered Attributes 24

3.2.2 Buyer's Constraint Elicitation 25

3.2.3 Buyer's Preference Elicitation 27

3.3 Bid Submission and Multi-Round Auctions 30

3.4 Bid Evaluation with MAUT* 32

3.4.1 Constraint Checking Algorithm 34

3.4.2 Attribute Preference Conversion 36

3.4.3 Attribute Weight Calculation 36

3.4.4 Attribute Value Preference Conversion 38

3.4.5 Attribute Utility Function Calculation 39

3.4.6 MAUT* Algorithm 43

3.5 3-Layer Software Architecture 49

3.6 MARA Implementation 51

3.6.1 BDI Model in MARA 51

3.6.2 Execution of MARA with Jadex 57

Chapter 4 Experiments and Evaluation 59

4.1 A Case Study 59

4.1.1 Definition of a 10-Attribute Reverse Auction 59

4.1.2 Ordering the TV Attributes 62

4.1.3 Constraint Elicitation 62

4.1.4 Preference Elicitation 64

4.1.5 Bid Submission 68
4.1.6 Attribute Weight Calculation 71
4.1.7 Utility Function Value Calculation 71
4.1.8 MAUT* Utility Calculation 74
4.2 Performance Evaluation of MARA 76
4.2.1 Configuration of MARA Parameters 76
4.2.2 Experiments 80

Chapter 5 Conclusion and Future Work 89

5.1 Conclusion 89
5.2 Future Work 91

Bibliography 93

Appendix A ADFs of MARA Agents 104

List of Tables

1.1 Comparing MAUT and MAUT* 8

1.2 Comparing MARA and Other Applications 8

3.1 TV Attributes and Values 24

3.2 Ordered TV Attributes 25

3.3 Attribute Importance Levels in MARA 27

3.4 Attribute Value Likeliness in MARA 27

3.5 TV Attribute Importance Levels 28

3.6 Ordered Attribute Values for Brand 29

3.7 An Example of Attribute Preference Conversion 36

3.8 Calculation of Weights of the TV Attributes 38

3.9 Attribute Value (String) Preference Conversion for Brand 39

3.10 Attribute Value (Numeric) Preference Conversion for Price 39

3.11 Calculation of Utility Values of Attribute Values (String) for Brand . 42

3.12 Calculation of Utility Values of Attribute Values (Numeric) for Price 43

3.13 BDI Model in MARA for Admin Agent 52

3.14 BDI Model in MARA for GUI and WD Agents 53

4.1 Weight Calculation of Attributes 71

4.2 Utility Function Values of Brand, Display Technology and Resolution 72

4.3 Utility Function Values of Customer Rating, Model Year and Price . 73

4.4 Utility Function Values of Refresh Rate, Screen Size, Warranty and Weight 74
4.5 Execution Time of MAUT* by Varying Number of Attributes 80
4.6 Execution Time of MAUT* by Varying Number of Sellers 81
4.7 Execution Time of MAUT* by Varying Number of Non-Conditional Constraints 83
4.8 Execution Time of MAUT* by Varying Number of Conditional Constraints 84
4.9 Execution Time of MAUT* by Varying Number of Non-Conditional Preferences 86
4.10 Execution Time of MAUT* by Varying Number of Conditional Preferences 87

List of Figures

3.1 Bid Evaluation in MARA System 33
3.2 MARA 3-Layer Software Architecture 50
3.3 Jadex Control Center GUI 58
4.1 Attribute Selection GUI 60
4.2 Selected Attributes with Values GUI 61
4.3 Attribute Order GUI 62
4.4 Non-Conditional Constraint GUI for ncc3 63
4.5 Conditional Constraint GUI for cc2 64
4.6 Attribute Preference GUI 66
4.7 Attribute Value (String) Ordering GUI for Brand 66
4.8 Conditional Preference on Attribute Value (String) GUI for Display Technology 67
4.9 Non-Conditional Preference on Attribute Value (String) GUI for Brand 67
4.10 Conditional Preference on Attribute Value (Numeric) GUI for Weight 68
4.11 Non-Conditional Preference on Attribute Value (Numeric) GUI for Refresh Rate 68
4.12 Bid Evaluation GUI 69
4.13 Bid Submission GUI 70
4.14 Bid Status GUI 76

4.15 Attributes vs Execution Time 81

4.16 Sellers vs Execution Time 82

4.17 Non-conditional Constraints vs Execution Time 83

4.18 Conditional Constraints vs Execution Time 85

4.19 Non-conditional Preferences vs Execution Time 86

4.20 Conditional Preferences vs Execution Time 87

Chapter 1

Introduction

1.1 Problem and Motivation

Continuous progress in technologies has had a major impact on sourcing strategies and practices, in particular, on auction systems. In particular, reverse auction systems are a type of advanced development that has achieved popularity with the advent of economical and efficient electronic capabilities. Informally, a reverse auction is a market scenario where there is one buyer who wants to buy a product and many sellers who bid on the product the buyer is interested in. Over the past few years, many companies from different domains of industries have started using reverse auctions [74]. According to a survey of the Purchasing's Annual Benchmark E-Sourcing, 31% of participated companies used reverse auctions in 2006, whereas in 2003, it was 15% [53]. Moreover, a *Purchasing* magazine survey in 2000, reported that 25 percent of the respondents expected to use reverse auctions [65]. A number of procurement departments and software vendors, such as eBreviate, PurchasePro, Clarus, IBM, DigitalUnion, Moai, Menerva, and Perfect, have embraced reverse auctions [9]. In addition, GE, Boeing, Sun Microsystems, Hewlett-Packard, Proctor & Gamble, and Dow Chemical employ reverse auctions [53]. Nevertheless, there are other providers of reverse auction services and software, such as AuctionAnything.com, Reverse Auctioneer, eDynaQuote and Epiq [74]. TIScover, an Austrian destination management system, uses a multi-attribute reverse auction to match tourists and hoteliers on an accommodation market [9]. Multi-attribute auctions have also been used for contracts with high asset-specificity, such as Department of Defense (DoD) contracts [9]. Companies are also using multi-attribute auctions for the procurement of many less specific goods, e.g. Maintenance, Repair and Operations (MRO) procurement, where multiple sourcing becomes important [9].

Behind the enormous acceptance and popularity of reverse auctions, there are lots of advantages and benefits. A survey reports that the users of reverse auctions achieved 15% savings in cost and up to 90% time reduction in their buying process [68]. In general, the bidding process takes only three to eight hours [53]. Moreover, there are other benefits of reverse auctions, such as access to global suppliers, increased trading transparency, reduced purchasing costs, increased efficiency, and enhanced procurement process [53, 74, 75]. More specifically, the incentive of reducing prices by as much as 20 percent and by an average of 5 to 12 percent is the major factor that attracts buyers [65]. Many of the above advantages benefit both the buyer and sellers. They not only save both the time and costs, but also are less inexpensive and easier to operate than that of forward auctions. They increase the competition among the sellers and provide the transparency of the bidding process [74].

Multi-attribute auctions are one of the most beneficial procurement system [66] and most of the procurement decision problems are multi-attribute [49]. Multi-Attribute Reverse Auctions (MARAs) allow negotiation between the buyer and sellers over the price along with other attributes such as durability, warranty, delivery time, product quality, terms of delivery, terms of payment, and sellers' reputation [7, 9, 20, 31, 35, 70, 79]. The deal value is not the only relevant factor in a multi-attribute reverse auction; as in real-life, other aspects such as trust, reputation, empathy and fairness are also important [49]. MARAs enable higher efficiency by exchanging information concerning the requirements of the buyer and the offerings of sellers more effectively than traditional auctions [9]. Much research has been done on MARAs, but several problems still have to be solved, such as eliciting the buyer's constraints and preferences, and determining the winner which has been shown to be computationally complex [9]. Preference elicitation is a very important feature for auction systems [42]. Often, the buyer wants to express his requirements qualitatively, which is convenient. An auction system should provide facilities to a buyer so that he can feel that the system meets his needs. More precisely, the buyer should be given the ability to provide his requirements in a friendly and interactive way. However, for the purpose of efficiency, it is more suitable for an auction system to process quantitative data. So, there is a remaining challenge of giving the buyer with a satisfactory interface while providing more efficiency.

This thesis proposes the development of a novel MARA protocol that uses agent technology to elicit the buyer's requirements and determine the winner efficiently. In our system, the buyer specifies his requirements containing constraints and preferences, and then the system looks for the outcome that completely satisfies the constraints and best satisfies the preferences.

1.2 Proposed Solution

This section consists of three sub-sections. First of all, we present the objectives for the development of the proposed protocol. Then we discuss the features of the proposed protocol and finally, we focus on the contributions of this research work.

1.2.1 Objectives

Often a buyer is more comfortable specifying his preferences and constraints about the auctioned product qualitatively. In this context, we propose a Multi-Attribute Reverse Auction (MARA) protocol to (1) elicit the buyer's qualitative, conditional and non-conditional preferences as well as constraints via friendly user interfaces, and (2) determine the winner according to the buyer's requirements. To do so, we extend the MAUT (Multi-Attribute Utility Theory) method [7] by allowing the buyer to submit qualitative preferences and to take the advantage of the efficiency of MAUT at the same time. Our auction system transforms all the qualitative requirements into quantitative ones. Another main concern of our MARA protocol is to completely automate MAUT calculation since it is a challenging task for the buyer to determine all the attribute weights and utility functions quantitatively, especially when there are many attributes for the product. The target of our MARA protocol is to find the winner, which means the seller whose bid completely satisfies the constraints and best satisfies the preferences.

1.2.2 Features of MARA

The following are the main features of our MARA protocol:

- **Reverse Auction:** we consider reverse auctions in which there is one buyer and more than one seller. The buyer wants to buy a product he is interested in. He specifies qualitatively the requirements of the product including conditional and non-conditional preferences and constraints. The sellers submit bids to win. The system checks the bids whether they satisfy the constraints or not. Then, the system calculates the utility values of the valid bids and announces the winner.

- **Multiple Attributes:** in our MARA protocol, besides price we consider other non-price attributes. For example, in a scenario of a television procurement, we consider price as well as brand, customer rating, display technology, high-definition multimedia interface (HDMI) inputs, model year, refresh rate, resolution, screen size, warranty and weight. The buyer can specify his preferences and constraints on multiple attributes of the product, and then the sellers can submit bids on all these attributes. Allowing multiple attributes offers the buyer to specify his requirements about the product in a more precise way. Moreover, it provides the sellers more options to compete for. Our MARA protocol supports unlimited number of attributes.

- **Multiple Rounds:** another feature of our MARA protocol is that it allows multiple rounds. Since it is beneficial both for the buyer and sellers, the sellers get more chance to improve their bids and thus to compete more and on the other hand, the buyer gets the product with the lowest possible cost.

- **Sealed-bid:** the MARA protocol follows the characteristics of sealed-bid auctions. In this protocol, the bidding information of the bidders are kept private during the round and at the end of each round the system reveals the utility values of the submitted bids and announces the round-winner.

1.2.3 Contributions

The following are the main contributions of our MARA protocol:

- **Constraint and Preference Elicitation:** the buyer can specify constraints about the product. Our MARA protocol can deal with both non-conditional and conditional constraints. The bid which does not satisfy the constraints completely, gets disqualified and removed from the auction. Our MARA protocol is able to handle preferences specified by the buyer. We consider non-conditional and conditional preferences. Conditional preferences depend on condition(s) on different attributes. Our MARA protocol also deals with qualitative preferences. Since the buyer is more comfortable to specify his preferences about the product qualitatively, we offer the qualitative way to the buyer to specify his requirements. The bid that best satisfies the preferences scores the maximum utility value.

- **Used Methodology:** MAUT which is a common technique in multi-attribute decision making [9] is designed to handle multiple objectives. We use MAUT in our MARA protocol to calculate the overall utility value of a bid. Since MAUT only deals with quantitative preferences [77], we improve MAUT by integrating several conversion methods to transform qualitative preferences into quantitative ones.

- **Winner Determination:** the winner determination in a multi-attribute auction is a computationally hard problem [9] in which the sellers are ranked based on the preferences specified by the buyer over the product attributes [77]. Unlike a single-attribute reverse auction that announces the seller with the lowest price as a winner. Winner determination refers to finding the seller whose bid meets the buyer's preferences mostly and eventually the winner gets the contract to supply the product [9]. In a multi-attribute reverse auction, the sellers compete to win the product by satisfying the constraints and preferences on multiple attributes specified by the buyer. Our MARA protocol is able to find the winner in the context of a multi-attribute reverse auction. It can determine the winner efficiently. We evaluate the performance of the MARA protocol by considering 10 attributes of a product, 20 sellers, 5 conditional constraints, 5 non-conditional constraints, 5 conditional preferences and 5 non-conditional preferences. The MARA protocol is able to determine the winner in a reasonable processing time. We perform several experiments of our proposed solution. We change one of the parameters such as number of attributes, sellers, conditional and non-conditional preferences and constraints and keep the remaining parameters unchanged in each experiment. We report the processing time of our MAUT* and analyze the results of all experiments.

- **Friendly Graphical User Interface:** our MARA protocol provides the users with friendly graphical interfaces. It assists step by step the buyer to specify his preferences and constraints about the product, and the sellers to submit the bids. Nevertheless, the system displays the bid results and status after each round.

- **3-Layer Architecture:** we develop our MARA protocol based on a 3-layer architecture. We limit the user interactions with the system within the presentation layer. The system calculates utility values in business logic layer. We make the system secured by storing data in the data layer and allowing only the administrator to manipulate data.

- **Comparison between MAUT and MAUT*:** Table 1.1 shows the differences between MAUT and our MAUT* method. As our MAUT* method performs additional calculations and steps of conversions from qualitative to quantitative, we cannot compare the performances of these methods.

Table 1.1: Comparing MAUT and MAUT*

Method	**Constraints**	**Preferences**
MAUT	NA	Quantitative
MAUT*	Non-conditional Conditional	Qualitative Non-conditional Conditional

- **Comparison between MARA and Other Preference-Aware Interactive Applications:** the systems interacting with users have to take care of their preferences. Amazon allows a user to specify his preferences quantitatively but does not handle constraints. Netflix and SmartClient are the same as Amazon in this manner. Table 1.2 depicts some preference-aware interactive applications along with their expressiveness in handling constraints and preferences [3].

Table 1.2: Comparing MARA and Other Applications

Application	**Quantitative Preferences**	**Qualitative Preferences**	**Constraints**
Amazon	Yes	No	No
MARA	No	Yes	Yes
Netflix	Yes	No	No
SmartClient	Yes	No	No

1.3 Organization of Thesis

The rest of this thesis is organized into four chapters.

Chapter 2 reviews the relevant literature on auction systems, including forward and reverse, single and multi-attribute, single and multi-round, open and sealed-bid auctions and vickrey protocol, multi-attribute utility theory and its applications, preference and constraint elicitation, and multi-agent systems.

Chapter 3 exposes our MARA protocol. Section 3.1 presents an brief overview of a MARA scenario. Section 3.2 provides in detail the buyer's requirements step-by-step with constraint and preference elicitation. Section 3.3 presents bid submission and multi-round auctions. Section 3.4 shows bid evaluation with MAUT* and all the underlying algorithms. Section 3.5 depicts the 3-layer software architecture of our MARA system. Finally, Section 3.6 demonstrates the Belief-Desire-Intention (BDI) model in MARA along with the execution of our MARA protocol with Jadex.

Chapter 4 discusses the experiments and evaluation performed in this thesis. Section 4.1 demonstrates the feasibility of our MARA protocol through a 10-attribute reverse auction of a television as a case study. Section 4.2 illustrates the performance evaluation of the MARA protocol.

Finally, Chapter 5 lists concluding remarks as well as possible directions for future research work.

Chapter 2

Related Works

This chapter is divided into four sections. Section 2.1 contains a survey on auction systems, including forward, reverse, single-attribute, multi-attribute, multi-round, sealed-bid auctions, vickrey bidding protocol, and the applications of auctions. Section 2.2 presents a brief overview of multi-attribute utility theory. Section 2.3 consists of preference and constraint elicitation. Finally, Section 2.4 introduces multi-agent systems.

2.1 A Survey on Auctions

The auction reported in [65] as possibly one of the world's oldest commercial tools, is a market scenario in which bidders compete for an item. Auctions were used in Babylon as early as 500 BC and their usage was also found in Roman history [65]. Che defined three auction protocols such as first-score, second-score and second-preferred-offer and proposed an optimal design based on an announced scoring rule [22]. Based on Che's independent cost model, Branco derived an optimal auction mechanism when costs are correlated [13]. Auctions have become a successful approach in buying and selling products such as properties, financial instruments and services. Moreover, several other markets are being introduced for auctions, for example, mobile-phone licenses, electricity, and pollution permits are some of them [51]. Auction-based mechanisms for electronic procurement have been discussed in [20] in three categories: (1) single-item auction, (2) multi-item auction, and (3) multi-attribute auction [55]. Yu and Han [76] discussed the limitations of the existing methods of supplier selection and introduced reverse auction in it. There are two important goals in auction theory such as maximization of efficiency and maximization of utility [79]. Online auction houses allow the buyers to buy products online. Sometimes, some products are not available in local markets. Moreover, they contain lists of categories which make it easier to the buyers to find the products of their interest. Electronics, travel services, and real estate are highlighted as some popular applications of online auctions in [6]. According to [6], computer is the second, travel service is the seventh, and real estate is the twelfth most popular category in online auctions. Priceline (www.priceline.com), eBay (www.eBay.com), Dubli (www.dubli.com), and Bidz (www.bidz.com) employ reverse auctions as their business models.

Forward vs Reverse Auctions: based on the role of the bidders, auctions can be of two types such as forward and reverse. In a forward auction, there is one seller and multiple buyers. The seller offers a product to the buyers and each buyer tries to win the product. The buyer who has the highest bid wins the product. Forward auctions have benefits for the sellers, for example, the seller can sell the product for the highest price. On the contrary, in a reverse auction [24, 35, 49, 67, 78], there is one buyer and more than one seller. There are numerous applications of mechanism design in electronic market design, distributed planning, and combinatorial optimization problems [51] and reverse auction is one of the main branches of mechanism design. However, because of its effect on spectrum allocations, auction design has achieved a lot of attention [78]. In a reverse auction, the buyer is interested in purchasing a product. The sellers submit bids on that product and try to be the winner by lowering the price. Reverse auctions have several benefits for both the buyer and the sellers. A reverse auction creates a real-time contact between the buyer and the sellers [74]. The buyer wins the product for the lowest cost and the seller sells the product in a shorter time. Moreover, when the number of the sellers is large, a reverse auction works well to decrease the price that is profitable for the buyer [74]. Nevertheless, the business models of reverse auctions influence e-commerce to be more professional [74]. Moreover, a survey performed by e-Bay depicts that a successful reverse auction can cut down the procurement cost between 6.3% and 43% which is in average 18% [75]. Hence, reverse auctions are more successful than forward auctions.

Single vs Multi-Attribute Auctions: according to the negotiable aspects, auctions can be categorized as single-attribute or multi-attribute. Single-attribute auction in which the bidders can negotiate only on one attribute which is price. The mechanism of a single-attribute auction is ease for the bidders to understand. Single-attribute auctions limit the bidders to bid on one attribute to obtain the products. But, in many real world situations, competition and negotiation engage several dimensions rather than only price. This type of auction, which is an extension of the single-attribute auction, is called multi-attribute auction [10]. Multi-attribute auctions are auctions in which the bidders can submit bids on multiple attributes such as price, product quality, delivery type, seller's reputation, warranty, terms of transportation, and delivery time. With the arrival of the internet, multi-attribute reverse auctions that allow the buyer and sellers to engage in negotiation have appeared [44]. Bichler discussed the method of winner determination in multi-attribute reverse auctions and configurable offers which enable the suppliers to specify multiple values for each attribute in [9] and clarified that the bidders have chances to improve the bidding values at less cost in multi-attribute auctions in [7]. Bichler and Kalagnanam [9] presented a multi-attribute auction as a producer of high-gain because of its bidding flexibility. Teich et al. covered many other detailed reviews and design features of multi-attribute reverse auctions in [69]. The features of multiple attributes have become essential in e-commerce because many industries design their markets based on multiple attributes. The advantage of a multi-attribute auction is that it allows the bidders to negotiate on multiple attributes, rather than limiting them to a single-attribute. This feature increases the efficiency of an auction. In this way, a multi-attribute auction offers bidding flexibility and it produces higher buyer-utility and seller-profit than that of a single attribute auction [8]. Needless to mention that a multi-attribute auction has difficulty in terms of making decisions as it deals with many attributes, but still it is a very useful method [66].

Single vs Multi Round Auctions: based on the number of round, auctions can be of two types such as single or multiple round. In a single round auction, there is only one round. This limitation makes the bidding process a difficult task for the bidders. In this type of auction, the bidders have no chance to make their bids improved because there is no feedback from the previous round. Hence, the disadvantage of a single round auction is the lower potential to be the winner. On the contrary, a multi-round auction provides several rounds for the bidding process. In a multi-round auction the bidders submit their bids and the winner is announced at the end of each round which benefits the bidders to improve their bids in the next round. To determine the current winner, the bids are evaluated at the end of each round [54]. Thus, a multi-round auction provides the opportunity to the bidders to improve their bids in the next round [60] and eventually offers the higher potential to be the winner. It makes the auction more competitive. A multi-round auction continues until a termination criterion is met. The possible termination conditions are, for example, the maximum predefined number of rounds has been reached or the bids are not getting improved compared to the previous round. In a multi-round auction, the bidder does not need to collect information of the bids of the competitors, because after each round all bids are shown to all bidders. Multiple rounds make the bidders compete and as a result the seller gets a better outcome in a forward auction while the buyer gets the advantage in a reverse auction [47].

Open vs Sealed-bid Auctions: depending on the visibility granted to the participants, auctions can be classified as open or sealed-bid. An open bid auction is a type of auction in which the bids are visible to all bidders. On the other hand, a sealed-bid auction is a type of auction in which the bids are private. In a sealed-bid auction, communication is not allowed between the bidders. By comparison to a open bid auction, having the bids invisible helps a sealed bid auction to reduce the potential of trust management issues and hence makes the process more secure. Sealed-bid auctions have been widely used in e-commerce for selling products and resource allocation and its reverse version has been used in task assignment scenarios [70]. A first price sealed-bid auction is a type of sealed-bid auction in which the bidding is private. In a first price sealed-bid auction, the bidders are allowed to place only one final bid and the bids are kept secret from all other bidders. This type of invisible bidding process prevents any type of communications among the bidders to use shared information. The bidder who submits the highest bid wins and he has to pay the bid he submitted. First price sealed-bid auctions are suitable for use in public procurement [28]. Vickrey auction is a type of sealed-bid auction presented by

William Vickrey in 1961 in which the bidding process is private or closed. In a vickrey auction, the bidders bid without knowing the bids of other bidders. This feature of a vickrey auction makes the auction house secure. The bidder who submits the highest bid wins but he has to pay the second highest bid. This is the difference between a first price sealed-bid auction and a vickrey auction. Moreover, a vickrey auction differs from other auction protocols by its closing time mechanism. In a vickrey auction protocol, the closing time is not determined and for that reason, the auction can be closed whenever the bid price matches the requested price [72]. For example, eBay uses a bidding strategy similar to vickrey protocol.

2.2 Multi-Attribute Utility Theory

Multi-Attribute Utility Theory (MAUT) [25, 77] is a methodology designed to handle multiple objectives. MAUT is a very common and is the most successful technique in multi-attribute decision making [9]. It is used to tackle complex problems involving a large number of decision variables. In MAUT, the sellers are supposed to provide the attributes of the product so that the buyer can choose the ones he is interested in.

MAUT calculates the overall utility value of a bid as the sum of the weighted utilities of the attributes. For the purpose of assigning weights, besides MAUT there are other approaches [20] such as Analytic Hierarchy Process (AHP) developed by Saaty [12], Simple Multi-Attribute Rating Technique (SMART), and Weight determination based on Ordinal Ranking of Alternatives (WORA) developed by IBM research [43]. In these decision making approaches, the buyer provides the numerical estimates of weight ratios. These methods differ from each other by the assessment methods of the calculation of attribute weights. However, in MAUT, the utilities and weights of all attributes should be specified. Since, MAUT deals with only quantitative data, every preference has to be converted into a quantitative form prior to MAUT calculation. Moreover, MAUT does not handle conditional preferences and constraints. Here, the conditional preference means the preference of an attribute that depends on the preference(s) of other attribute(s). Hence, the difficulty of using MAUT is that it needs the preference of each attribute quantitatively [77]. But given the utility functions and weights of all attributes, determining the winner is very easy and efficient.

The areas of MAUT are no longer limited to decision support applications, behavioral considerations, robustness considerations, and role of heuristics, rather they have been spread out into the areas of data envelopment analysis (DEA), negotiation science, geographic information systems, engineering applications, and e-commerce including multi-attribute auctions and shopping agents [73]. In the field of multi-attribute reverse auctions, Bichler et al. [10] analyzed MAUT algorithm for bid evaluation in single-sourcing whereas Teich et al. [70] developed an online trading system for multiple-sourcing. Geoffrion and Krishnan summarized the applications of MAUT in e-commerce area in [38]. Early applications of MAUT focused on public sector decisions and policy issues. The military is also a leading user of this technique. The design of major new weapon systems always involves multi-attributes such as cost, weight, durability, survivability etc. [9]. MAUT is very popular by consumer organizations for evaluating products. Chin and Porage [25] used MAUT for travel planning. Perfect (www.perfect.com) and Frictionless Commerce (www.frictionless.com or www.sap.com) are using MAUT in their commercial sourcing software package.

2.3 Preference and Constraint Elicitation

Preference elicitation [12, 41, 46, 57] is becoming a popular research subject day by day. In many applications, it is getting more importance. Preference elicitation is a process to extract preferences from the user. Preference elicitation technique deduces important information from the preferences specified by the user [33]. The goal of preference elicitation is the acquisition of user preferences. During the elicitation process, many considerations should be made by the system in order to assist the user. Otherwise, the system may model an error-prone structure and collect false information that influences the user's preferences. Furthermore, if the system is not well structured, the user may provide the system wrong information about his preferences unintentionally. The easiest way to assist the user to specify his preferences is to provide him with the pairwise comparisons of the possible outcomes. The representation structure of a system controls the elicitation process by determining what information is needed and the order of the collection of information [33]. The primary step in a preference elicitation procedure is to determine the form of the queries. Moreover, these queries should provide additional information to make the user better understand the system. Nevertheless, elicitation queries should be as simple as possible.

Preferences and constraints co-exist in many domains [2, 57]. Thus, in many real world applications, it is of great interest to handle them together. Many attempts have been made in order to deal with both constraints and preferences in applications. Preferences can be quantitative, qualitative and conditional. MAUT is used to handle quantitative preferences [77] and conditional preference networks (CP-nets) is used to represent conditional and qualitative preferences. A constrained CP-nets is a special version of CP-nets that can handle constraints in addition of CP-nets [57]. A constraint satisfaction problem (CSP) is well-known for constraint problems. It consists of a set of variables with possible values and a set of constraints, and the solution of it is a complete assignment of values to all variables satisfying all constraints [2].

Many research works have been done in the context of preference elicitation. Some improvements of preference elicitation for decision support systems have been done in [12]. Preference elicitation plays an important role in the development of autonomous multi-agent systems in e-commerce applications. Roelofsma and Schut discussed a novel preference elicitation model based on cross modality matching in e-commerce settings [59]. How to accurately elicit user preferences has become the burning concern of current decision support systems. Chen and Pu made a survey of preference elicitation methods in [23]. In [36], we can observe the usage of preference elicitation for interface optimization which describes ARNAULD, a general interactive tool to elicit user preferences and to learn a factored cost function in decision-theoretic optimization automatically using the feedback. A comprehensive view on adaptation and personalization opportunities in the elicitation process for e-Services is developed in [46]. Understanding and representing user preferences accurately is very crucial in most decision support systems. Shao-bin et. al discussed and evaluated different methods for preference elicitation in negotiation support systems [63]. Preference elicitation in fully probabilistic design of decision strategies has been discussed in [48] which mainly opened a way to a systematic data-based preference elicitation. Winner determination based on preference elicitation in reverse auctions has been discussed in [42].

Preference elicitation performs the main role in several areas such as computer research, decision analysis, and artificial intelligence [16]. Several approaches have been developed in many application domains such as decision support systems, product configuration, recommender systems [46] and decision-theoretic artificial intelligence [57]. Preference elicitation is very important in interactive applications such as online shopping [3]. Maximizing user satisfaction can be done by considering his preferences about the product he is interested in. This type of privilege of the user preferences makes the system popular. Preference-aware interactive applications include recommender systems such as Amazon and Netflix. These systems recommend items based on the past history of the items the user was interested in. SmartClient, Teaching Salesman and PExA are some examples of the interactive systems [3]. SmartClient is a planning tool that manages the activities of travellers uses utility functions, Teaching Salesman is another interactive system which helps the user to compare products depending on his preferences and the preferences are expressed in PExA as instances of soft constraint satisfaction problem.

Decision support systems depend on the successful handling of user preferences [2]. These systems rely on preference elicitation to build the effective user models. Preference elicitation process makes no assumptions about the initial user information unlike other techniques such as collaborative filtering. The objective of preference elicitation is to construct an accurate user model by providing the necessary data so that a decision support system can assist a user efficiently. The representations of decision support systems which are capable of expressing the user preferences over multi-dimensional data set are either vector-based or network-based [33]. Since preferences play important roles in both the input and the output of all complex decision making systems, it is crucial to elicit user preferences accurately to build an efficient decision system both in theory and in practice.

User preference elicitation plays an important role in negotiations. The primary role of the preference elicitation is to allow the buyer-side agent to gather the user preference structure in an automated marketplace such as auction [43]. The acquisition of high quality user preference elicitation processes are continuously achieving importance with the increasing acceptance of the web as an effective distribution channel and the complexity of interactive e-Services that are valuable means to improve the customer relations. The quality of the outcomes depends on the capability of the system to elicit the user preferences [46].

2.4 Multi-Agent Systems

Agent oriented methodologies [45, 52, 71] have gained importance in the research field of advanced software engineering [29]. An agent, [21, 30] which can perceive and act over an environment, is an autonomous entity whose behavior is conduced by the objectives. An agent is built to solve problems in several application domains such as e-commerce, internet searchers, and intelligent tutoring systems [4]. The ideas behind the concept of agent characteristics such as autonomy, social ability, reactivity, proactivity, intelligence, rationality and cooperation provide a powerful abstraction tool in the solution of the complex distributed problems. To qualify as an agent, a software or hardware system is often required to have these properties. A Multi-Agent System (MAS) [11, 19, 32] is an instance of distributed systems composed by autonomous agents in which the agents work cooperatively to achieve their goals [4]. In recent years, MAS have achieved myriad popularity in various domains of industries [1]. In a MAS, in order to obtain the specific services, each agent interacts with its environment and other agents using high level communication languages and protocols. Agent oriented methodologies are essential to solve complex problems and lead to the development process and thus Agent Oriented Programming (AOP) is most often required where the system must be robust, autonomous and proactive. AOP seems to be suited in many applications such as e-commerce, enterprise resource planning, air traffic control systems, and personal digital assistants [17]. The designing process of a MAS involves many terminologies, for example, the system should achieve some goals and the designer should produce an accurate model for the knowledge and interactions of the agents [26]. The designer addresses different aspects of the agent solution such as social, trust, and ontology aspects when designing a MAS [26]. The existing MASs in auction systems use exact methods to solve the problem of winner determination [1].

Chapter 3

Multi-Attribute Reverse Auction (MARA) Protocol

In MARA protocol, we improve MAUT by including the following features:

1. Qualitative preferences.

2. Conditional and non-conditional preferences.

3. Conditional and non-conditional constraints.

The buyer feels more comfort to specify his preferences about the product qualitatively. Certainly it is more natural to the buyer to express preferences qualitatively rather than providing scores or digits. For example, it is more comfortable to express that *Brand* is very much important to him when purchasing a television (TV), rather than providing a score of 80% of importance level to the *Brand* attribute. Preferences and constraints are closely related types of information but are different by means of their importance. Preferences are treated as soft constraints. They are viewed as wishes or tolerant constraints [2]. On the other hand, constraints are treated as hard constraints or strict requirements that must be completely satisfied. Satisfying hard constraints is often more important than satisfying preferences. In our MARA protocol, a constraint means a hard constraint and a preference is equivalent to a soft constraint. Indeed, a bid that satisfies partially the buyer's preferences still has a chance to win but it must be deleted as soon as it violets any hard constraint. MARA protocol is discussed in detail in the following sections.

This chapter is divided into six sections. Section 3.1 provides the definition of multi-attribute reverse auction. Section 3.2 describes the buyer's requirements, including constraint and preference elicitation. Section 3.3 illustrates bid submission of the sellers and multi-round auctions. Section 3.4 depicts bid evaluation with MAUT*, including algorithms and conversion methods of MARA. Section 3.5 describes 3-layer software architecture in MARA and finally the chapter concludes with Section 3.6 providing MARA implementation, BDI model in MARA and its execution with Jadex.

3.1 Multi-Attribute Reverse Auction Definition

The buyer requests to purchase a product. Every legal attribute of the product of the buyer's interest is presented. These attributes are extracted from the product database. The buyer selects attributes according to his interests from all legal attributes. The system generates all the possible values for each attribute of interest. A reverse auction based on the buyer's request of purchasing a product is launched by the auctioneer.

Example: we assume there is a reverse auction of TV having five attributes, such as Brand, Display Technology, Price, Refresh Rate and Weight as shown in Table 3.1.

Table 3.1: TV Attributes and Values

Attribute	**Value**
Brand	LG, Panasonic, Sharp, Sony, Toshiba
Display Technology	LCD, LED, Plasma
Price ($)	[800 - 899.99], [900 - 999.99], [1000 - 1499.99], [1500 - 2000]
Refresh Rate (Hz)	60, 120, 240, 600
Weight (Kg)	[4 - 4.9], [5 - 5.9], [6 - 7]

3.2 Buyer's Requirements

The buyer can specify conditional and non-conditional constraints on attribute values. He can express preferences on attributes by assigning importance levels and also can specify conditional and non-conditional preferences on attribute values by assigning likeliness. In the following subsections we discuss these steps in detail.

3.2.1 Buyer's Ordered Attributes

Selected attributes of the buyer are shown to him alphabetically as they are stored in the product database in that manner. The buyer orders them according to his interest and then the system assigns positions to them w.r.t to the buyer's order.

Example: the buyer orders the TV attributes according to his interest and the system assigns positions as shown in Table 3.2.

Table 3.2: Ordered TV Attributes

Attribute	**Ordered by the Buyer**	**Position**
Brand	Price	1
Display Technology	Brand	2
Price	Display Technology	3
Refresh Rate	Weight	4
Weight	Refresh Rate	5

3.2.2 Buyer's Constraint Elicitation

The buyer expresses a constraint in the following form:

$$(condition_1)\ and/or,\ ...,\ and/or\ (condition_n) \Rightarrow constraint \tag{3.1}$$

where

- condition$_i$: rel(attribute, value of attribute)
- constraint: rel(attribute, value of attribute)
- rel $\in \{=, \neq, <, >, \leq, \geq\}$

If the condition clause is NULL, then the constraint is non-conditional. This type of constraint has no dependencies and has to be satisfied completely. On the other hand, if the condition clause is not NULL, then the constraint is conditional. This type of constraint has dependencies and has to be satisfied completely depending on the condition(s).

Example: let assume the buyer submits the following constraints for this auction.

(c1) $NULL \Rightarrow Brand \neq Panasonic$

(c2) $NULL \Rightarrow Brand \neq Sharp$

(c3) $(Refresh\ Rate \geq 240)$ *and* $(Weight \leq [4 - 4.9])$

$\Rightarrow Display\ Technology \neq LCD$

(c4) $(Weight \geq [5 - 5.9])$ *or* $(Display\ Technology = LED)$

and $(Refresh\ Rate \leq 120) \Rightarrow Price \leq [1000 - 1499.99]$

In the above mentioned non-conditional constraints, **(c1)** expresses that the Brand must not be Panasonic, and **(c2)** that the Brand must not be Sharp.

(c3) means that if the Refresh Rate is greater than or equal to 240 and the value of Weight is less than or equal to [4 - 4.9], then Display Technology must not be LCD. Here the constraint depends on two conditions and if these conditions are true, then this constraint must be satisfied.

The constraint, **(c4)** means if the Weight is greater than or equal to [5 - 5.9] or Display Technology is LED and Refresh Rate is less than or equal to 120, then Price must be less than or equal to [1000 - 1499.99]. Since the precedence of 'and' is greater than that of 'or', these conditions can be viewed as ($condition_1$ or ($condition_2$ and $condition_3$)) and checked in the following order: at first $condition_2$, then $condition_3$ and at last $condition_1$. If all these conditions are found true in this order, then the constraint must be satisfied. Therefore, (Display Technology = LED) is checked first followed by (Refresh Rate ≤ 120) and at last (Weight ≥ [5 - 5.9]) is executed.

3.2.3 Buyer's Preference Elicitation

The buyer expresses importance levels of attributes as well as conditional and non-conditional preferences over the attribute values qualitatively. Attribute values can be of two types such as String and Numeric. The system assigns equivalent quantitative values of the importance levels of attributes and likeliness of attribute values as shown in Tables 3.3 and 3.4.

Table 3.3: Attribute Importance Levels in MARA

Importance Level	**Abbreviated**	**Quantitative Importance Level**
Extremely Important	EI	1
Very Much Important	VMI	0.75
Important	I	0.5
Not Much Important	NMI	0.25

Table 3.4: Attribute Value Likeliness in MARA

Attribute Value Type	**Likeliness**	**Abbreviated**	**Quantitative Likeliness**
String	Highest	HS	1
	Above Average	AA	0.8
	Average	A	0.6
	Below Average	BA	0.4
	Lowest	LS	0.2
Numeric	Highest	HS	1
	Lowest	LS	0.2

A. Preferences on Attributes: the buyer specifies his preferences on attributes by assigning importance levels according to Table 3.3.

Example: the following table shows an example of specifying importance levels of TV attributes by the buyer.

Table 3.5: TV Attribute Importance Levels

Attribute	**Importance Level**
Brand	VMI
Display Technology	I
Price	EI
Refresh Rate	NMI
Weight	I

B. Order on Attribute String Values: in case of string type of attribute value, the buyer orders attribute values w.r.t his interest and then the system assigns positions to them according to the buyer's order in the same process as we discussed for attribute ordering in Section 3.2.1.

Example: assume the buyer orders the values of Brand according to his interest and the system assigns positions as shown in Table 3.6.

Table 3.6: Ordered Attribute Values for Brand

Attribute Value	**Ordered by the Buyer**	**Position**
LG	Toshiba	1
Panasonic	Sony	2
Sharp	LG	3
Sony	Sharp	4
Toshiba	Panasonic	5

C. Conditional and Non-conditional Preferences on Attribute Values: the buyer can express conditional and non-conditional preferences on attribute values. In this context, the buyer specifies the likeliness of attribute values qualitatively and the system assigns equivalent quantitative values as shown in Table 3.4. The buyer expresses conditional and non-conditional preferences on attribute values in the following form:

$$(condition_1)\ and/or,\ ...,\ and/or\ (condition_n) \Rightarrow preference \tag{3.2}$$

where

- condition$_i$: rel(attribute, value of attribute)
- rel $\in \{=, \neq, <, >, \leq, \geq\}$
- preference: attribute(value$_1$(likeliness), ... , value$_m$(likeliness)) //see Table 3.4

If the condition clause is NULL, then the preference is non-conditional that means it has no dependencies on the condition(s). On the other hand, if the condition clause is not NULL, then the preference is conditional. This type of preference has dependencies on the condition(s). In case, the buyer does not specify any preferences on an attribute values, the MARA system assigns 0 to the quantitative likeliness of those attribute values. Moreover, if the buyer specifies his conditional preferences on an attribute values but any bid does not satisfy the condition(s), then the MARA system assigns 0 to the quantitative likeliness of those attribute values for that bid.

Example: assume the buyer submits the following preferences for the TV auction.

(p1) $NULL \Rightarrow Price([800 - 899.99](HS), [1000 - 1499.99](LS))$

(p2) $(Refresh\ Rate > 120) \Rightarrow Brand(LG(BA), Panasonic(A), Sharp(A),$
$Sony(AA), Toshiba(HS))$

The above mentioned non-conditional preference, **(p1)** expresses [800 - 899.99] as the *highest* and [1000 - 1499.99] as the *lowest* preference value of Price.

(p2) means if the value of Refresh Rate is greater than 120, then LG as *Below Average*, Panasonic as *Average*, Sharp as *Average*, Sony as *Above Average* and Toshiba as *Highest* preference value of Brand. **(p1)** is an example of preference on numeric attribute value where **(p2)** is for string attribute value preference.

3.3 Bid Submission and Multi-Round Auctions

In MARA, the sellers submit bids following the Vickrey protocol [58, 64] and the winner is determined with MAUT*. Vickrey bidding protocol is a type of sealed-bid auction. According to this protocol, a bidder submits bid without knowing the bids of other bidders.

All the requirements of the buyer are displayed to the registered sellers who will compete by providing the bids. Assume the following two bids are submitted by two sellers.

bid_i**:** $Brand = Panasonic; Display\ Technology = LCD; Price = 1000;$

$Refresh\ Rate = 240; Weight = 4$

bid_j**:** $Brand = Toshiba; Display\ Technology = Plasma; Price = 1200;$

$Refresh\ Rate = 600; Weight = 4$

The first bid does not satisfy the non-conditional constraint **(c1)** of the buyer for the value of Brand. It also violets the conditional constraint **(c3)** in which the buyer specifies that the value of Display Technology must not be LCD, if Refresh Rate is greater than or equal to 240 and Weight is less than or equal to [4 - 4.9]. However, this bid respects **(c2)** and **(c4)**. But the bid is deleted as soon as it violets **(c1)** and there is no necessity to check other constraints for this bid in order to save some processing time. Moreover, there is no need to calculate further the utility value of this bid w.r.t the buyer's preferences that will also save some processing time. On the other hand, the second bid satisfies all the constraints of the buyer but does not guaranty as the best bid. MARA system will calculate its utility value according to the buyer's preferences.

MARA protocol is capable of handling multi-round auctions. In each round, after bid submission of the sellers, the system checks the submitted bids whether they maintain the constraints or not. If any bid does not satisfy the constraints, the bid will be deleted and there is no need to calculate the utility value of that bid. After each round, the sellers are informed the utility values of all bids. In the next round, each seller is expected to submit higher bid than the highest bid of the previous round. Bidding continues until the highest utility values of the two consecutive rounds remain the same.

3.4 Bid Evaluation with MAUT*

In a reverse auction, each attribute a in a bid has a value denoted by v_a. Each attribute is evaluated through an independent utility function $U_a(v_a)$. An attribute has a weight, $weight_a$, which shows its importance to the buyer. The MAUT utility of a bid is calculated as the sum of all weighted utilities of the attributes as follows [9]:

$$MAUT(bid) = \sum_{a=1}^{M} weight_a \times U_a(v_a) \quad (3.3)$$

where

- *MAUT(bid)* is the overall utility of a bid.
- M is the number of attributes.

In Figure 3.1, we present the inputs and outputs of the bid evaluation task of MARA system. It takes qualitative preferences of attributes (*Qualitative Importance Level*) as inputs and converts them into quantitative ones. It also takes qualitative preferences of attribute values (*Qualitative Likeliness*) as inputs and converts them into quantitative ones. Then it calculates the weights of attributes, utility function values of attributes and finally produces the overall utility of each submitted bid. This whole process is discussed in detail in the subsequent subsections.

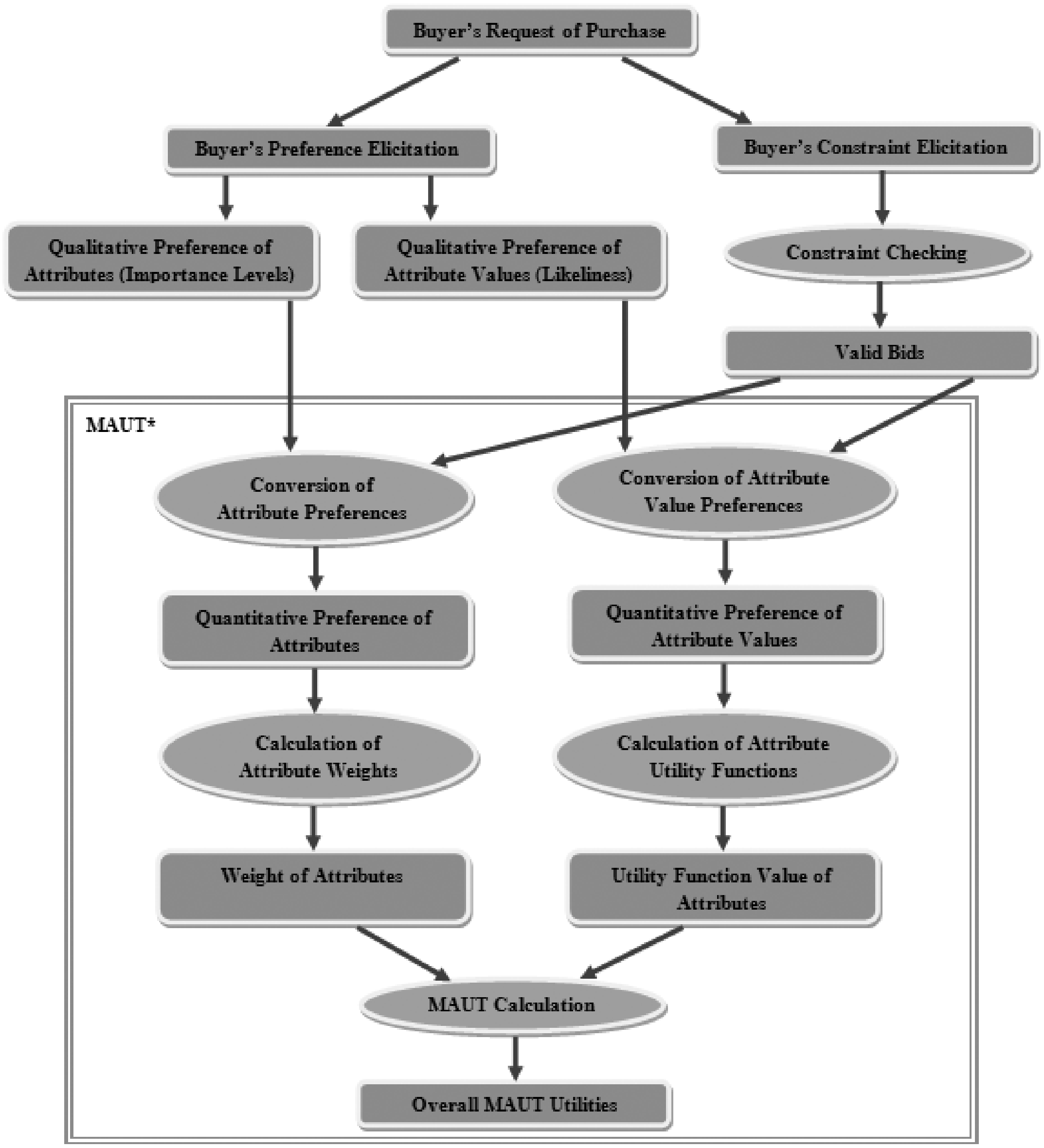


Figure 3.1: Bid Evaluation in MARA System

The remaining of this section illustrates the tasks and algorithms developed in our MARA system.

3.4.1 Constraint Checking Algorithm

ConstraintChecking algorithm, given in Algorithm 1, checks whether a submitted bid maintains the buyer's constraints or not. It deletes those bids that do not satisfy these constraints. The input and output parameters of ConstraintChecking algorithm are the following:

- $C = \{c_1, \ldots, c_P\}$ is the set of the buyer's constraints.
- $B = \{bid_1, \ldots, bid_Q\}$ is the set of submitted bids in one round.
- VB is the set of valid bids.

Algorithm 1 ConstraintChecking(C, B)

```
{ for i = 1 to Q do
   {  flag = 1; j = 1; VB = Φ;
      while (  j ≤ P and flag == 1)
     {       if ( violetBid(bid_i, c_j) == TRUE) //see Algorithm 2
                 flag = 0;
             else
                 VB = VB ∪ {bid_i};
             j++;
     }
   }
 return VB; }
```

The input parameters of violetBid algorithm are the following:

- c is a constraint.
- bid is a submitted bid.

Algorithm 2 violetBid(*bid*, *c*)

```
{ Flag = Φ; conditionFlag = TRUE; s = 1;
  if (r > 0) //r is the number of conditions in a constraint c
  { for s = 1 to r do
      { if (AValBid == AValCond or AValBid ∈ AValCond) //AValCond is
                //the value of the attribute in condition_s and AValBid is the
                //value of the same attribute in bid
            flag_s = TRUE;
        else
            flag_s = FALSE;
        Flag = Flag ∪ {flag_s};
        s++;
      }
  }
  conditionFlag = findConditionFlag(Flag, Conj);
  //Conj is the set of {and, or} to join the conditions
  if (conditionFlag == TRUE)
  { if (AValBid == AValConst or AValBid ∈ AValConst) //AValConst is
                //the value of the attribute in c and AValBid is the
                //value of the same attribute in bid
        flag_c = TRUE;
    else
        flag_c = FALSE;
  }
  if (flag_c == TRUE) return FALSE;
  else return TRUE;
```

3.4.2 Attribute Preference Conversion

The system converts the buyer's qualitative preferences of the attributes into quantitative ones by using Table 3.3. The following table depicts the attribute preference conversion of the importance levels of attributes specified by the buyer qualitatively as given in Table 3.5

Table 3.7: An Example of Attribute Preference Conversion

Attribute	**Importance Level**	**Quantitative Importance Level**
Brand	VMI	0.75
Display Technology	I	0.5
Price	EI	1
Refresh Rate	NMI	0.25
Weight	I	0.5

3.4.3 Attribute Weight Calculation

The system calculates the weights of the attributes in two steps. It first calculates the ranks of attributes followed by the calculation of their weights.

A. Attribute Ranks: in MARA, the term $rank_a$ means a relative value of an attribute a according to its position in the list of attributes sorted by the buyer. The rank is calculated with the following equation.

$$rank_a = M - position_a + 1 \tag{3.4}$$

where

- M is the number of attributes.

- $position_a$ is the position of attribute a in the attribute list ordered by the buyer according to his interests. //see Section 3.2.1

B. Attribute Weights: weight of an attribute a, $weight_a$, specifies the relative value among other attributes according to its importance to the buyer.

$$weight_a = rank_a \times quanImpLevel_a \times weightRate$$

$$/\sum_{a=1}^{M} weight_a = 1 \qquad (3.5)$$

where

- $rank_a$ is the rank of attribute a.

- $quanImpLevel_a$ is the equivalent quantitative value of the buyer's qualitative importance level of attribute a according to Table 3.3.

- $weightRate$ is the step value of weights of attributes.

Example: the following table shows an example of the calculation of weights of the TV attributes according to the above mentioned equations 3.4 and 3.5 and Tables 3.1, 3.3 and 3.5. According to the equation 3.5, $weightRate$ can be calculated for our example as follows:

$$((4 \times 0.75) + (3 \times 0.5) + (5 \times 1) + (1 \times 0.25) + (2 \times 0.5)) \times weightRate = 1$$

$\Rightarrow weightRate = 0.093$

Table 3.8: Calculation of Weights of the TV Attributes

Attribute	**Position**	**Rank**	**Quantitative Importance Level**	**Weight**
Brand	2	4	0.75	0.279
Display Technology	3	3	0.5	0.14
Price	1	5	1	0.465
Refresh Rate	5	1	0.25	0.023
Weight	4	2	0.5	0.093

It is clear from Table 3.8 that there are some effects of position, rank and importance on weight of an attribute. Price is assigned the 1st position among the five attributes as a result it gets the highest rank of 5. Moreover, it is assigned the highest importance value of 1. Therefore, it achieves the highest weight of 0.465 as well. On the other hand, Refresh Rate gets the lowest weight of 0.023 as it is assigned the lowest importance value of 0.25 and the lowest rank of 1 for the lowest position of 5.

3.4.4 Attribute Value Preference Conversion

The system converts the buyer's qualitative preferences of the attribute values into quantitative ones by using Table 3.4. The following tables depict the attribute value preference conversion of the likeliness of attribute values of Brand and Price specified by the buyer qualitatively as shown in **(p2)** and **(p1)** respectively.

Table 3.9: Attribute Value (String) Preference Conversion for Brand

Attribute Value	**Likeliness**	**Quantitative Likeliness**
LG	BA	0.4
Panasonic	A	0.6
Sharp	A	0.6
Sony	AA	0.8
Toshiba	HS	1

Table 3.10: Attribute Value (Numeric) Preference Conversion for Price

Attribute Value	**Likeliness**	**Quantitative likeliness**
[800 - 899.99]	HS	1
[900 - 999.99]	-	-
[1000 - 1499.99]	LS	0.2
[1500 - 2000]	-	-

3.4.5 Attribute Utility Function Calculation

At first we discuss the calculation of the utility value of an attribute when its value is of string type. The system converts the buyer's qualitative preferences of string value of attributes into quantitative values, ranks them and then calculates the utility function values.

A. Attribute Value Ranks: in MARA, the term $rank_{v_a}$ means a relative value of an attribute value v_a according to its position in the list of the values of an attribute that is sorted by the buyer and it is defined by the following equation.

$$rank_{v_a} = N - position_{v_a} + 1 \tag{3.6}$$

where

- N is the number of values of an attribute.
- $position_{v_a}$ is the position of the attribute value v_a in the list of the values of that attribute ordered by the buyer according to his interests. //see Section 3.2.3

B. Utility Function Value: in MARA, the term $U_a(v_a)$ means the utility function value of attribute a in respect of value v. The following equation works when the attribute value is of string type.

$$U_a(v_a) = rank_{v_a} \times quanLikeliness_{v_a} \times utilityRate \tag{3.7}$$

where

- $rank_{v_a}$ is the rank of a value of attribute a.
- $quanLikeliness_{v_a}$ is the equivalent quantitative value of the buyer's qualitative preference of value of attribute a (see Table 3.4).
- *utilityRate* is the step value of utility function values of attribute values.

Now when the value of an attribute is of numeric type, we follow another process to calculate the utility function value. The system converts the buyer's qualitative preferences of each numeric value of attributes into quantitative values, and calculates utility function values of attributes according to its each value with the following equations.

$$U_a(v_{la}) = 0; U_a(v_{ha}) = 1 \tag{3.8}$$

$$U_a(v_a) = (v_a - v_{la})/(v_{ha} - v_{la}) \tag{3.9}$$

where

- $U_a(v_{la})$ is the utility function value of attribute a in respect of value v which has the lowest likeliness specified by the buyer.
- $U_a(v_{ha})$ is the utility function value of attribute a in respect of value v which has the highest likeliness specified by the buyer.
- v_a is a value or an average value in case of value of ranges of attribute a.
- v_{la} is a value or an average value in case of value of ranges of attribute a which has the lowest likeliness.
- v_{ha} is a value or an average value in case of value of ranges of attribute a which has the highest likeliness.
- $U_a(v_a)$ is the utility function value of attribute a in respect of value v.

Then the system redefines $U_a(v_{la})$ as the second lowest utility function value divided by the total number of values of attribute a. The system also assigns the utility values of those attribute values to 0 which qualitative likelinesses specified by the buyer are out of interest.

Example: the following table shows an example of the calculation of utility values of the string type for attribute Brand according to the equations 3.6 and 3.7 and Tables 3.6 and 3.9. As the utility value of the attribute value of the highest likeliness is equal to 1, for this example the utility value of Toshiba is 1. Hence, in case of Toshiba, according to the equation 3.7, *utilityRate* can be calculated for this example as follows:

$$5 \times 1 \times utilityRate = 1$$

$$\Rightarrow utilityRate = 0.2$$

Table 3.11: Calculation of Utility Values of Attribute Values (String) for Brand

Attribute Value	**Position**	**Rank**	**Quantitative Likeliness**	**Utility Value**
LG	3	3	0.4	0.24
Panasonic	5	1	0.6	0.12
Sharp	4	2	0.6	0.24
Sony	2	4	0.8	0.64
Toshiba	1	5	1	1

Table 3.12 illustrates an example of the calculation of utility values of the numeric type values of the attribute Price according to equations 3.8 and 3.9 and Table 3.10.

Table 3.12: Calculation of Utility Values of Attribute Values (Numeric) for Price

Attribute Value	**Quantitative Likeliness**	**Utility Value**
[800 - 899.99]	1	1
[900 - 999.99]	-	0.75
[1000 - 1499.99]	0.2	0.188
[1500 - 2000]	-	-

As the values of Price are in the form of range of values, we use the average value of each range, for example, 849.995 is used for [800 - 899.99]. According to equation 3.8, the utility value of [800 - 899.99] is 1 and [1000 - 1499.99] is 0. The utility value of [900 - 999.99] is calculated according to equation 3.9 as follows where 949.995 and 1249.995 are the average values of [900 - 999.99] and [1000 - 1499.99] respectively:

$$U_{Price}([900 - 999.99]) = (949.995 - 1249.995)/(849.995 - 1249.995) = 0.75$$

Then redefine the utility value of [1000 - 1499.99] = (second lowest utility value / number of attribute values) = (0.75 / 4) = 0.188 and the utility value of [1500 - 2000] is not calculated as it is out of interest of the buyer.

3.4.6 MAUT* Algorithm

To evaluate a bid, the MAUT* algorithm is performed. MAUT* algorithm calls two other functions:

- Attribute Weight Algorithm
- Utility Function Algorithm

A. Attribute Weight Algorithm: AttributeWeight algorithm given in Algorithm 3, takes attributes' order and importance level in a qualitative form. It first converts these values into quantitative form, then calculates attributes' ranks and finally computes attributes' weights. The input and output parameters of Attribute Weight algorithm are the following:

- $OA = \{1, \ldots, \mathrm{M}\}$ is the set of ordered attributes.
- $QualImplevel_{OA} = \{qualImpLevel_1, \ldots, qualImpLevel_M\}$ is the set of qualitative importance levels of attributes.
- $Weight = \{weight_1, \ldots, weight_M\}$ is the set of weights of attributes.

Algorithm 3 AttributeWeight(OA, $QualImplevel_{OA}$)

{ $Weight = \Phi$;

  for $a = 1$ to M do

  { $position_a$ = attPosition(OA, a);

    $rank_a = M$ - $position_a + 1$;

    $quanImpLevel_a$ = attQuantitative($QualImplevel_{OA}$, a);

    $weight_a = rank_a \times quanImpLevel_a \times weightRate$;

    $Weight = Weight \cup \{weight_a\}$;

  }

return $Weight$; }

B. Utility Function Algorithm: UtilityFunction algorithm, given in Algorithm 4, takes attribute values' preferences and calculates utility function value. The input parameters of utilityFunctionValue algorithm are the following:

- $OVA = \{1, \ldots, \mathrm{N}\}$ is the set of ordered values of attributes.
- $Likeliness_{OVA} = \{likeliness_1, \ldots, likeliness_N\}$ is the set of qualitative likeliness of values of attributes.
- $U(v_a)$ is the set of utility function values of values of attributes.

Algorithm 4 UtilityFunction(OVA, $Likeliness_{OVA}$)

```
{ for a = 1 to M do
  {  U(v_a) = Φ;
     if (v_aType == String)
       { for v = 1 to N_a do
         {  pos_{v_a} = attvaluePosition(OVA, a, v);
            rank_{v_a} = N - pos_{v_a} + 1;
            likeliness_{v_a} = attvalueQuantitative(Likeliness_OVA, a, v);
            U_a(v_a) = rank_{v_a} × likeliness_{v_a} × utilityRate;
            U(v_a) = U(v_a) ∪ {U_a(v_a)};
         }
       }
     if (v_aType == Numeric)
       {  U_a(v_la) = 0;
          U_a(v_ha) = 1;
          for v = 2 to N_a − 1 do
             U_a(v_a) = (v_a - v_la)/(v_ha - v_la);
          U_a(v_la) = (second lowest utility value)/ N_a;
          for v = 1 to N_a do
             U(v_a) = U(v_a) ∪ {U_a(v_a)};
       }
  }
return U(v_a); }
```

C. MAUT* Algorithm: MAUT* algorithm given in Algorithm 5 calculates the overall utility of a valid bid. The input parameters of MAUT* Algorithm are the following:

- VB is the set of valid bids.
- $OA = \{1, \ldots, \mathrm{M}\}$ is the set of ordered attributes.
- $Impl_{OA} = \{impl_1, \ldots, impl_M\}$ is the set of qualitative importance level of attributes.
- $OVA = \{1, \ldots, \mathrm{N}\}$ is the set of ordered values of attributes.
- $Likeliness_{OVA} = \{likeliness_1, \ldots, likeliness_N\}$ is the set of qualitative importance level of values of attributes.

Algorithm 5 MAUT* (VB, OA, $Impl_{OA}$, OVA, $Likeliness_{OVA}$)

```
{ Weight = attributeWeight(OA, Impl_OA);
   // Weight is the set of weights of attributes
   U(v_a) = utilityFunctionValue(OVA, Likeliness_OVA);
   // U(v_a) is the set of utility function values of attributes
   i = 1; top = 0;
   winner = bid_i;
   for each bid_i ∈ VB do
   {  a = 1; MAUTValue = 0;
      while a ≤ M do
      {  AValUtility = findAttributeValueUtility(bid_i, U(v_a), a);
         MAUTValue += MAUTVal(Weight, AValUtility, a);
      }
      if (top < MAUTValue)
      {  top = MAUTValue;
         winner = bid_i;
      }
   }
 return winner; }
```

3.5 3-Layer Software Architecture

A 3-Layer software architecture [5, 37], used to develop a large and complex system, consists of three layers: the presentation layer, business logic layer and data access layer . The task of the presentation layer is to provide the users with the friendly user interfaces. It accesses the information of the required services of the system and translates it into the understandable format to the user. The business logic layer performs the processing of the task of the users, makes logical decisions and performs calculation. It also coordinates the application. The data access layer stores the databases of the system. 3-Layer software architecture has gained an increasing popularity in the software industry. It provides several benefits such as scalability and security. It facilitates the system by providing scalability by handling a large number of users. It provides information fast and almost with no delay. As more users access the system, a three-layer solution is more scalable than other solutions such as two or one-layer software architecture. It also provides security as the users do not have a direct access to the databases.

In Figure 3.2, we present the 3-layer software architecture of our MARA system. In the presentation layer, our MARA system provides graphical user interfaces (GUIs) to the buyer and sellers step-by-step. Graphical user interface agent controls these GUIs. Winner determination agent checks the buyer's constraints and calculates MAUT* value of bids. It exists in business logic layer. Auction and product databases are stored in data access layer. Admin agent exists in business logic layer and communicates with databases on request of graphical user interface and winner determination agents. Admin agent controls the access to the databases and thus provides the security. It also helps other agents by providing necessary information.

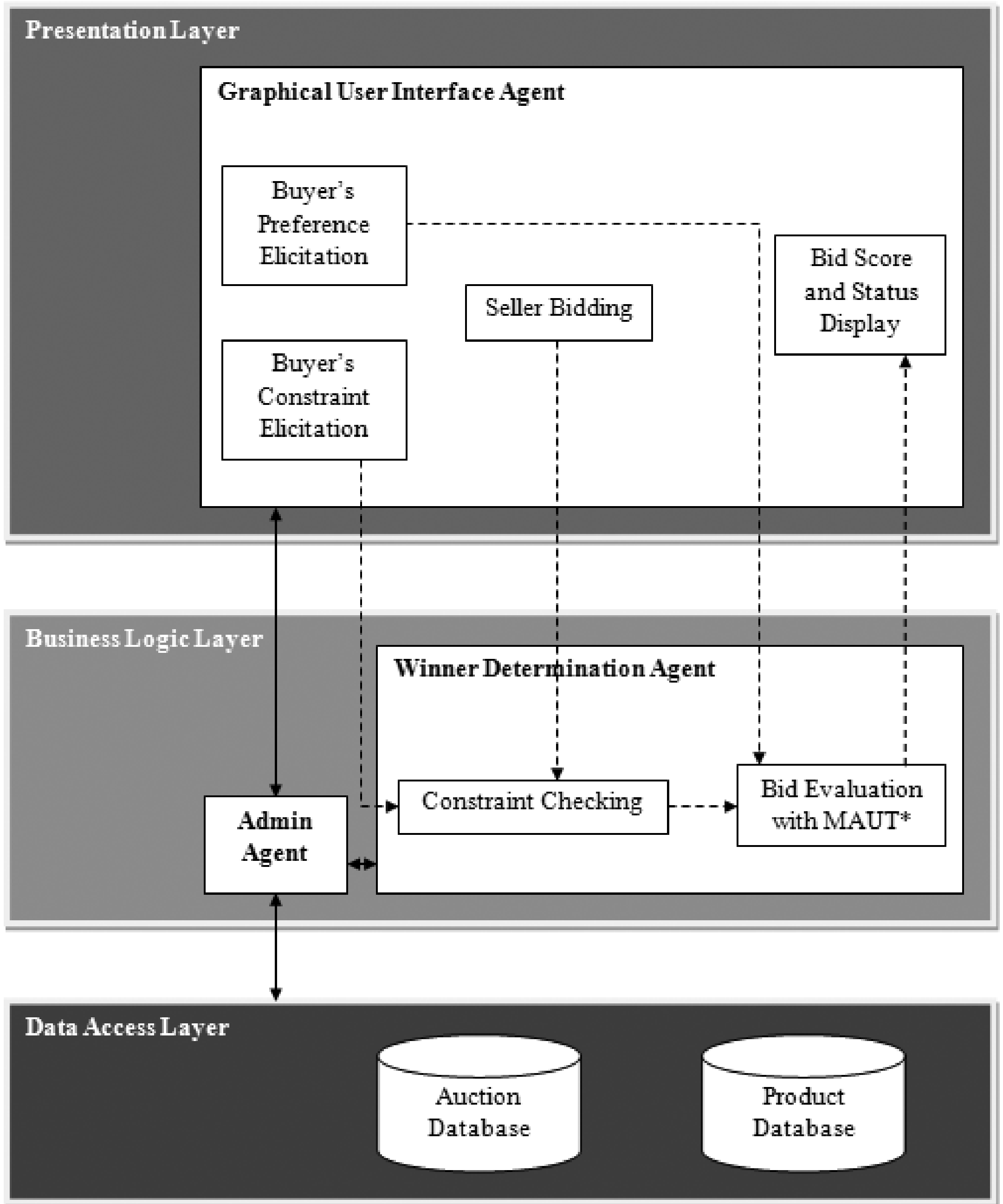


Figure 3.2: MARA 3-Layer Software Architecture

3.6 MARA Implementation

We use the concept of Belief-Desire-Intention (BDI) model [18, 34, 39, 40, 50, 61, 62] to design the agents of our MARA system. Each agent has a set of beliefs and facts to realize the current environment and status. Moreover, they have goals that are sets of desires by which they can make appropriate decisions. Furthermore, they use messages to communicate with each other and plans by which they perform their assigned tasks. Developing a multi-agent system requires an Agent Execution Environment (AEE) that provides services to create and terminate agents and a standard interface for agent interactions. We develop our MARA system by using Jadex framework [14, 15, 56] as AEE that provides core services based on FIPA specification [27], design Agent Definition Files (ADFs) by XML and implement plans of the agents and graphical user interfaces by Java. For the integrated development environment we use Eclipse IDE 3.7.2 and Java SE Development Kit (JDK 7u25).

3.6.1 BDI Model in MARA

Tables 3.13 and 3.14 show the BDI model in our MARA protocol. They highlights the plans and graphical user interface windows of agents. A brief introduction of plans which are used in MARA is discussed in the following.

Table 3.13: BDI Model in MARA for Admin Agent

Agent	**Plan**	**GUI Windows**
Admin Agent	aucAttrOrderRespPlan	-
	aucAttrPrefRespPlan	
	aucAttrRespPlan	
	aucAttrStrValOrderRespPlan	
	bidSubmissionRespPlan	
	bidUtilityRespPlan	
	condConstRespPlan	
	condNumPrefRespPlan	
	condStrPrefRespPlan	
	constChkRespPlan	
	nonCondConstRespPlan	
	nonCondNumPrefRespPlan	
	nonCondStrPrefRespPlan	
	proAttrRespPlan	
	validUserRespPlan	

Table 3.14: BDI Model in MARA for GUI and WD Agents

Agent	**Plan**	**GUI Windows**
GUI Agent	aucAttrOrderReqPlan	AttributeOrderGUI
	aucAttrPrefReqPlan	AttributePreferenceGUI
	aucAttrReqPlan	AttributeSelectionGUI
	aucAttrStrValOrderReqPlan	AttributeValueStringOrderGUI
	bidSubmissionReqPlan	AttributewithValueGUI
	bidUtilityReqTriPlan	BidEvaluationGUI
	condConstReqPlan	BidStatusGUI
	condNumPrefReqPlan	BidSubmissionGUI
	condStrPrefReqPlan	ConditionalConstGUI
	constChkReqTriPlan	ConditionalPrefNumericGUI
	nonCondConstReqPlan	ConditionalPrefStringGUI
	nonCondNumPrefReqPlan	ConstraintMainGUI
	nonCondStrPrefReqPlan	NonConditionalConstGUI
	proAttrReqPlan	NonConditionalPrefNumericGUI
	validUserReqPlan	NonConditionalPrefStringGUI
		PreferenceMainGUI
		ProductSelectionGUI
		WelcomeGUI
WD Agent	bidUtilityReqPlan	
	constChkReqPlan	

- **validUserReqPlan and validUserRespPlan:** the validUserReqPlan is instantiated by the GUI agent when the user wants to log in. By using the validUserReqPlan the GUI agent requests the Admin agent to verify the log in information of the user. The Admin agent responds with the user verification information by using the validUserRespPlan.

- **proAttrReqPlan and proAttrRespPlan:** by using the proAttrReqPlan the GUI agent requests the Admin agent for the attributes of the product. The Admin agent retrieves all legal attributes of the product from the product database by using the proAttrRespPlan.

- **aucAttrReqPlan and aucAttrRespPlan:** the GUI agent provides the selected attributes of the buyer to the Admin agent by using the aucAttrReqPlan and the Admin agent stores this information to the auction database by using the aucAttrRespPlan.

- **aucAttrOrderReqPlan and aucAttrOrderRespPlan:** the GUI agent provides the orders of the attributes of the buyer to the Admin agent by using the aucAttrOrderReqPlan and the Admin agent stores this information to the auction database by using the aucAttrRespPlan.

- **condConstReqPlan and condConstRespPlan:** the condConstReqPlan is used by the GUI agent when the buyer specifies his conditional constraints. The GUI agent requests the Admin agent to store this information in auction database and the Admin agent performs this task by using the condConstRespPlan.

- **nonCondConstReqPlan and nonCondConstRespPlan:** the nonCondConstReqPlan is used by the GUI agent when the buyer specifies his nonconditional constraints. The GUI agent requests the Admin agent to store this information in auction database and the Admin agent performs this task by using the nonCondConstRespPlan.

- **aucAttrPrefReqPlan and aucAttrPrefRespPlan:** the GUI agent provides the preferences of the attributes of the buyer to the Admin agent by using the aucAttrPrefReqPlan and the Admin agent stores this information to the auction database by using the aucAttrPrefRespPlan.

- **aucAttrStrValOrderReqPlan and aucAttrStrValOrderRespPlan:** the GUI agent provides the orders of the attribute values of string type of the buyer to the Admin agent by using the aucAttrStrValOrderReqPlan and the Admin agent stores this information to the auction database by using the aucAttrStrValOrderRespPlan.

- **condStrPrefReqPlan and condStrPrefRespPlan:** the condStrPrefReqPlan is used by the GUI agent when the buyer specifies his conditional preferences on the attribute values of string type. The GUI agent requests the Admin agent to store this information in auction database and the Admin agent performs this task by using the condStrPrefRespPlan.

- **nonCondStrPrefReqPlan and nonCondStrPrefRespPlan:** the nonCondStrPrefReqPlan is used by the GUI agent when the buyer specifies his nonconditional preferences on the attribute values of string type. The GUI agent requests the Admin agent to store this information in auction database and the Admin agent performs this task by using the nonCondStrPrefRespPlan.

- **condNumPrefReqPlan and condNumPrefRespPlan:** the condNumPrefReqPlan is used by the GUI agent when the buyer specifies his conditional preferences on the attribute values of numeric type. The GUI agent requests the Admin agent to store this information in auction database and the Admin agent performs this task by using the condNumPrefRespPlan.

- **nonCondNumPrefReqPlan and nonCondNumPrefRespPlan:** the nonCondNumPrefReqPlan is used by the GUI agent when the buyer specifies his non-conditional preferences on the attribute values of numeric type. The GUI agent requests the Admin agent to store this information in auction database and the Admin agent performs this task by using the nonCondNumPrefRespPlan.

- **bidSubmissionReqPlan and bidSubmissionRespPlan:** the GUI agent provides the bids submitted by the sellers to the Admin agent by using the bidSubmissionReqPlan and the Admin agent stores this information to the auction database by using the bidSubmissionRespPlan.

- **constChkReqTriPlan, constChkReqPlan and constChkRespPlan:** the GUI agent triggers the Winner Determination agent to check constraints by using constChkReqTriPlan. The Winner Determination agent checks the bids if they satisfy the conditional and non-conditional constraints specified by the buyer by using the constChkReqPlan with the help of the constChkRespPlan of the Admin agent.

- **bidUtilityReqTriPlan, bidUtilityReqPlan and bidUtilityRespPlan:** the GUI agent triggers the Winner Determination agent to calculate bid utility value by using bidUtilityReqTriPlan. The Winner Determination agent calculates the overall utilities of the bids submitted by the sellers by using the bidUtilityReqPlan with the help of the bidUtilityRespPlan of the Admin agent.

3.6.2 Execution of MARA with Jadex

Our MARA system has three ADF files, eighteen GUIs, thirty-two plans and two databases. According to the 3-layer architecture of our MARA protocol shown in Figure 3.2, three agents have been developed. We implement the ADF files such as AdminAgent.agent.xml, GUIAgent.agent.xml and WDAgent.agent.xml by using XML. We also develop MARAAgents.application.xml by using XML to configure these three agents. We implement GUIs and plans by using Java. We build and perform manipulation of two databases such as auction and product databases by using MySQL. We write almost eleven thousand lines of code to develop our MARA protocol. It assists the buyer by providing friendly user interfaces to specify his product of interest, conditional and non-conditional constraints and preferences step-by-step. It also assists the sellers to submit the bids.

Figure 3.3 shows the interface of Jadex Control Center (JCC) which is a Java class that provides user interface and is used to maintain the execution of the agents. The list of MARA agents is located in the model tree on top of the window on the left side of the screen. The model panel is on the top on the right side of the screen and is used to specify configurations of the agents. The panel under the model panel provides the description of the selected agent from the model tree. Figure 3.3 shows the initial state of Jadex control center for MARA system and also indicates the starting of MARA by pressing the Start button which initiates the WelcomeGUI.

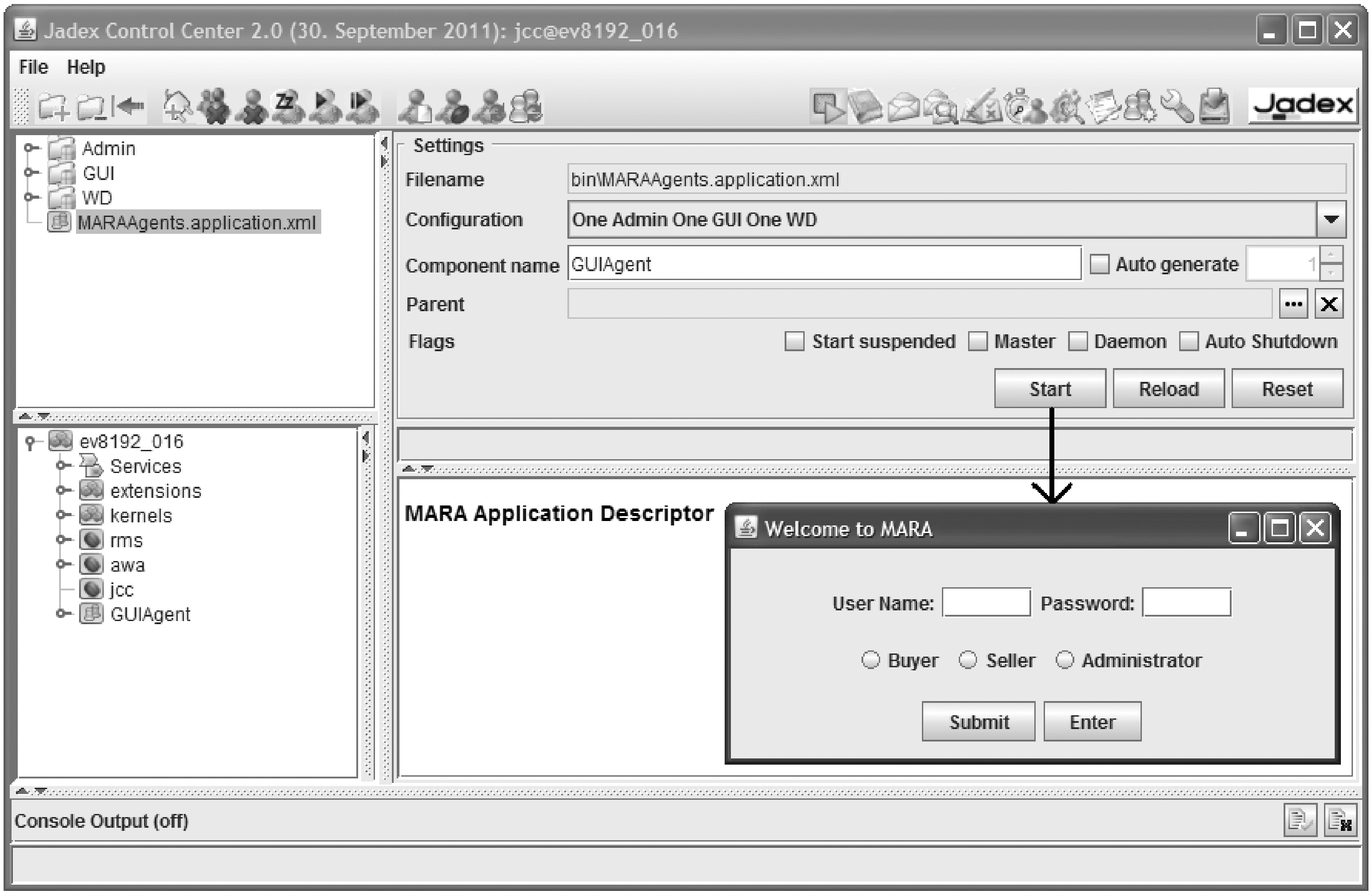


Figure 3.3: Jadex Control Center GUI

Chapter 4

Experiments and Evaluation

This chapter consists of two sections. Section 4.1 provides the case study of a complete reverse auction of a television, including the buyer's requirements, sellers' bid submissions and the calculations to determine the winner. Section 4.2 contains the performance evaluation of the MARA system.

4.1 A Case Study

We discuss how our MARA system assists the buyer to specify his requirements to purchase a product step-by-step through a case study. We also show the assistance of MARA to guide the sellers to submit their bids.

4.1.1 Definition of a 10-Attribute Reverse Auction

The buyer wants to purchase a product. The system displays all available products to the buyer. Assume the buyer selects television (TV) to purchase. Our MARA system generates all legal attributes of TV from the product database. Figure 4.1 is shown to the buyer to select attributes of his interest. Figure 4.2 depicts the selected attributes with their values.

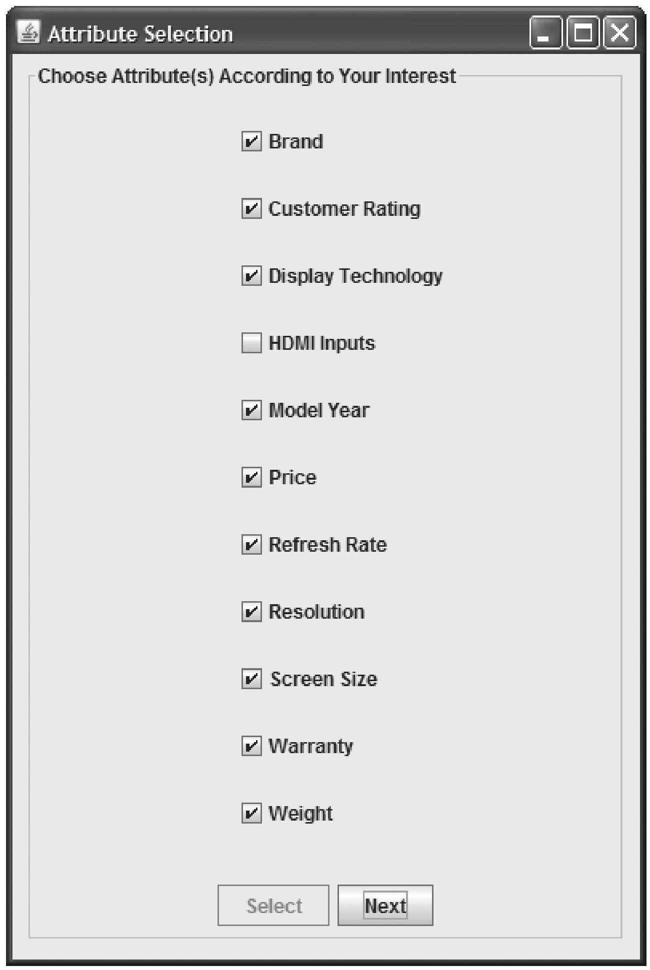


Figure 4.1: Attribute Selection GUI

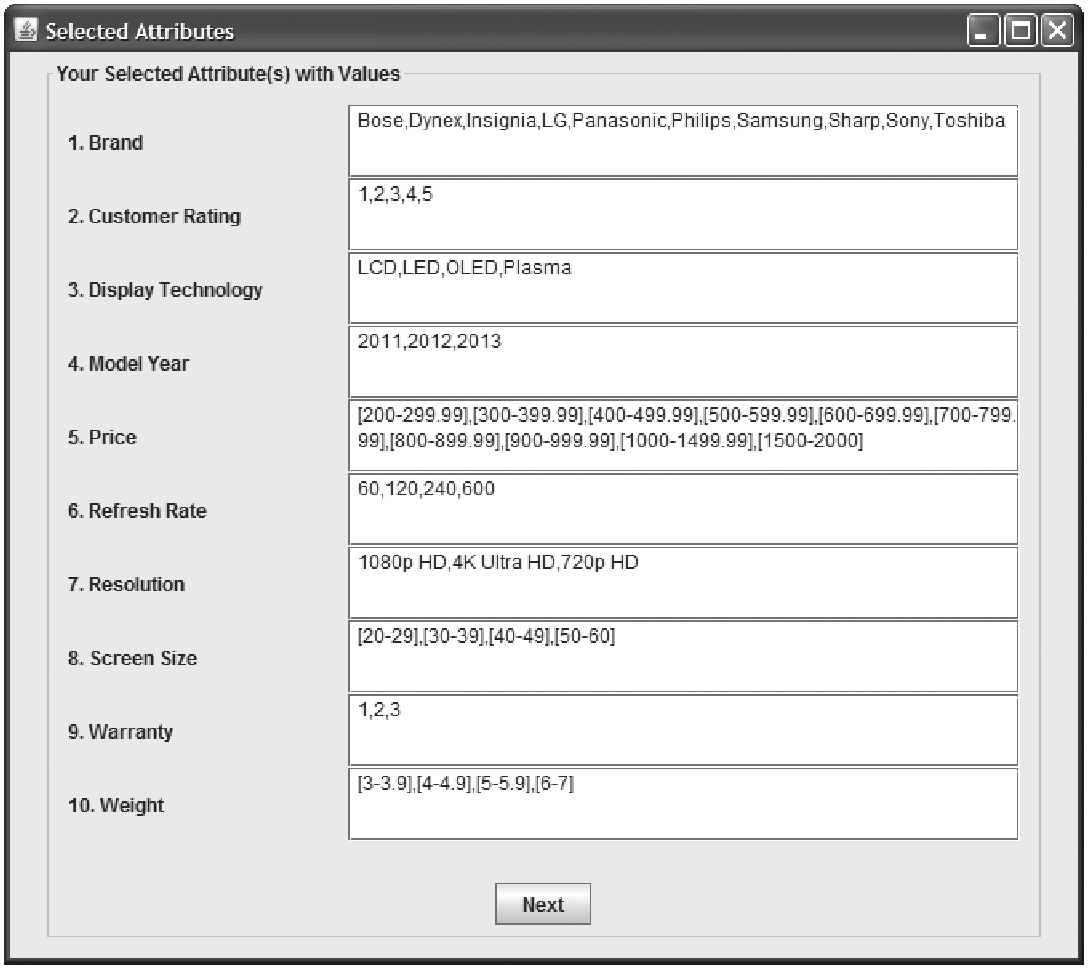


Figure 4.2: Selected Attributes with Values GUI

4.1.2 Ordering the TV Attributes

The buyer can order the attributes according to his interest as illustrated in Figure 4.3.

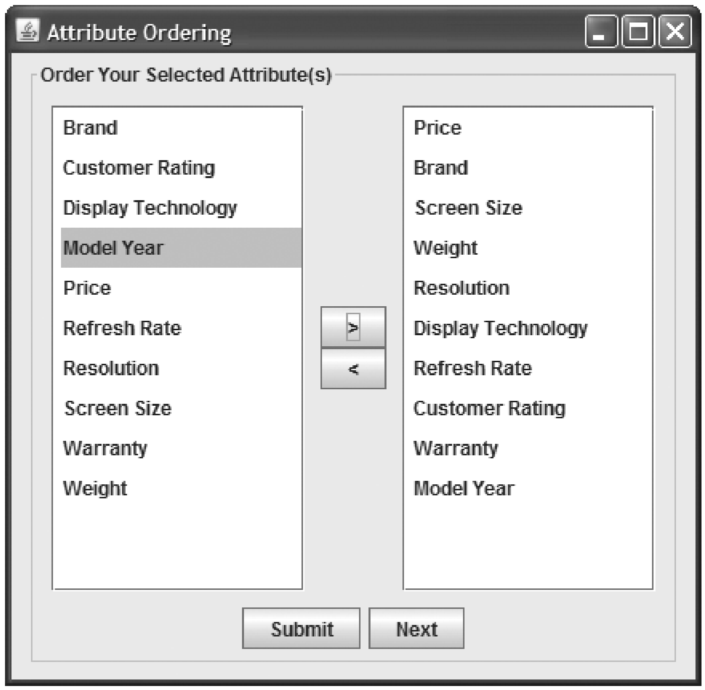


Figure 4.3: Attribute Order GUI

4.1.3 Constraint Elicitation

The buyer can now express non-conditional and conditional constraints. Assume the following are the buyer's constraints for this auction. For example, Figures 4.4 and 4.5 display how our MARA system assists the buyer to specify constraints.

Non-Conditional Constraints:

(ncc1) $NULL \Rightarrow Model\ Year \neq 2011$

(ncc2) $NULL \Rightarrow Warranty \geq 2$

(ncc3) $NULL \Rightarrow Refresh\ Rate \geq 120$

(ncc4) $NULL \Rightarrow Screen\ Size \geq [30 - 39]$

Conditional Constraints:

(cc1) $(Refresh\ Rate \leq 240) \Rightarrow Price \leq [900 - 999.99]$

(cc2) $(Brand = Panasonic)\ and\ (Resolution = 720p\ HD) \Rightarrow$

$Weight \leq [5 - 5.9]$

(cc3) $(Brand = LG)\ or\ (Resolution = 1080p\ HD) \Rightarrow$

$Screen\ Size \leq [40 - 49]$

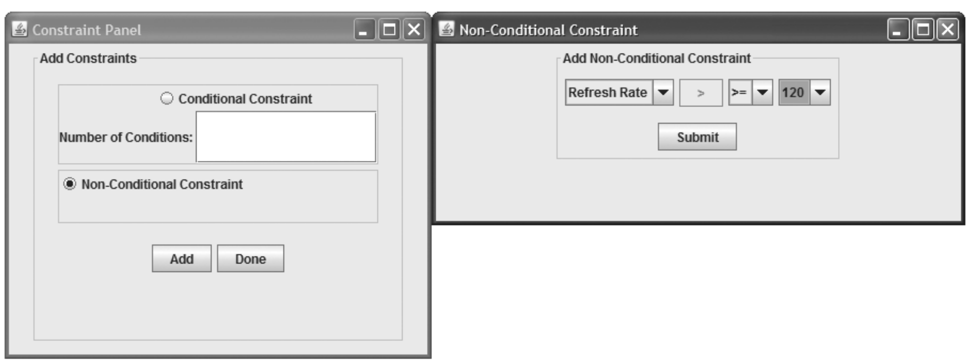


Figure 4.4: Non-Conditional Constraint GUI for ncc3

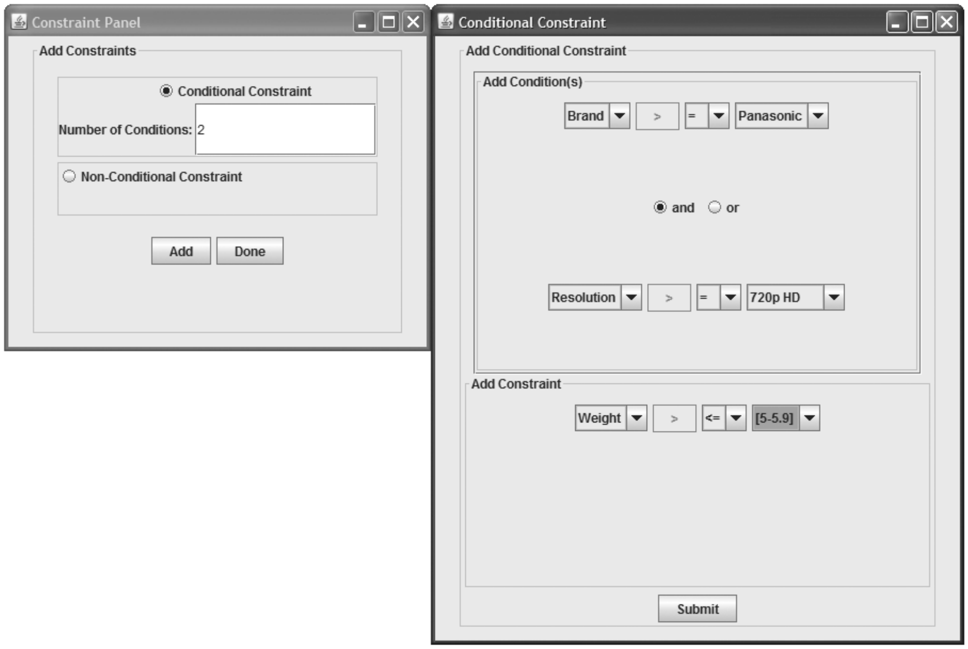


Figure 4.5: Conditional Constraint GUI for cc2

4.1.4 Preference Elicitation

The buyer can specify preferences. First, the buyer expresses attribute preferences by assigning qualitative importance levels as shown in Figure 4.6. Then, he can order attribute value of string type according to his interest as shown in Figure 4.7. Moreover, the buyer can specify conditional and non-conditional preferences on attribute values of both string and numeric types by assigning likeliness. Assume the following are the buyer's preferences for this auction. In Figures 4.8, 4.9, 4.10 and 4.11, we assist the buyer in each step of the preference elicitation process.

Non-Conditional Preferences:

(ncp1) $NULL \Rightarrow Price([300 - 399.99](HS), [1000 - 1499.99](LS))$

(ncp2) $NULL \Rightarrow Refresh\ Rate(600(HS), 120(LS))$

(ncp3) $NULL \Rightarrow Brand(Bose(BA), Dynex(LS), Insignia(BA), LG(AA),$

$Panasonic(A), Philips(A), Samsung(A), Sharp(BA),$

$Sony(AA), Toshiba(HS))$

(ncp4) $NULL \Rightarrow Screen\ Size([50 - 60](HS), [30 - 39](LS))$

(ncp5) $NULL \Rightarrow Model\ Year(2013(HS), 2012(LS))$

(ncp6) $NULL \Rightarrow Warranty(3(HS), 2(LS))$

(ncp7) $NULL \Rightarrow Customer\ Rating(5(HS), 3(LS))$

Conditional Preferences:

(cp1) $(Price > [300 - 399.99])\ and\ (Screen\ Size \geq [40 - 49]) \Rightarrow$

$Display\ Technology(LCD(BA), LED(A), OLED(AA), Plasma(HS))$

(cp2) $(Refresh\ Rate \geq 120) \Rightarrow$

$Resolution(1080p\ HD(HS), 4K\ Ultra\ HD(AA), 720p\ HD(A))$

(cp3) $(Screen\ Size \geq [30 - 39]) \Rightarrow Weight([4 - 4.9](HS), [6 - 7](LS))$

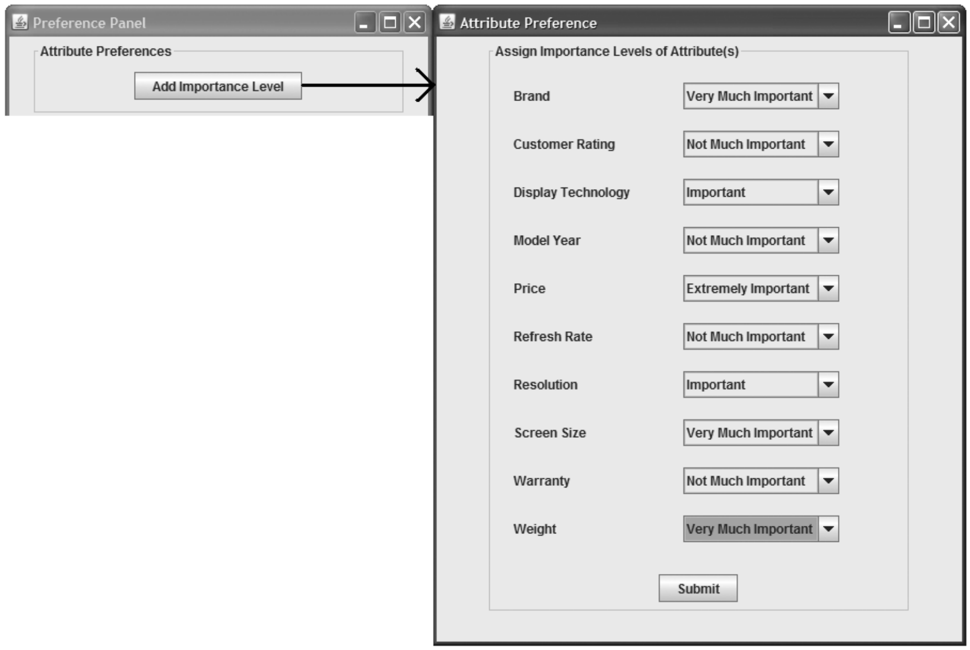


Figure 4.6: Attribute Preference GUI

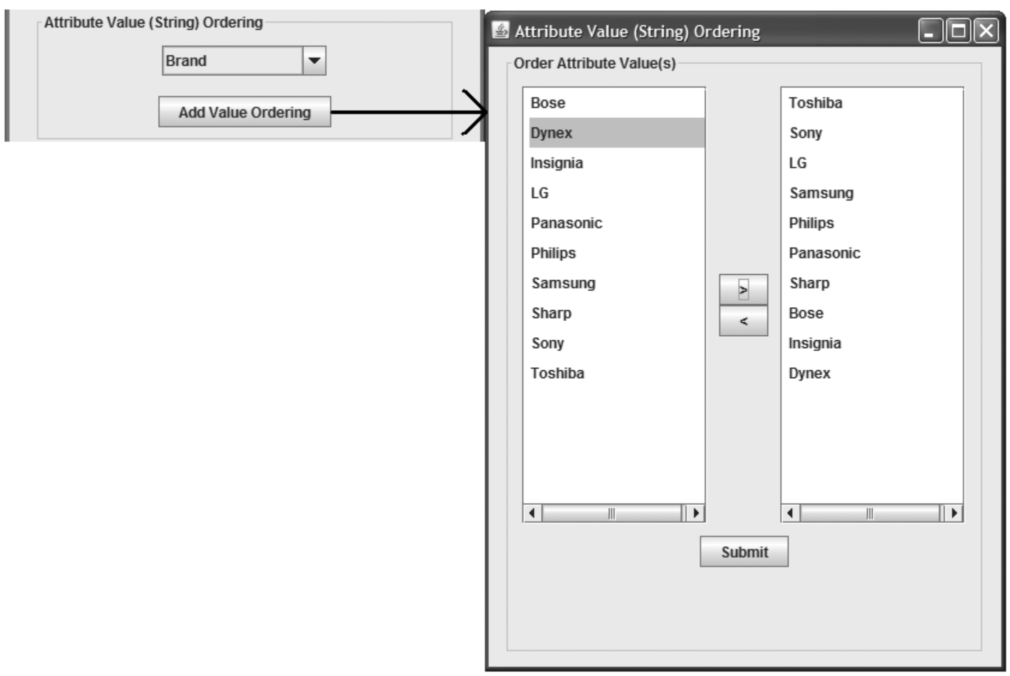


Figure 4.7: Attribute Value (String) Ordering GUI for Brand

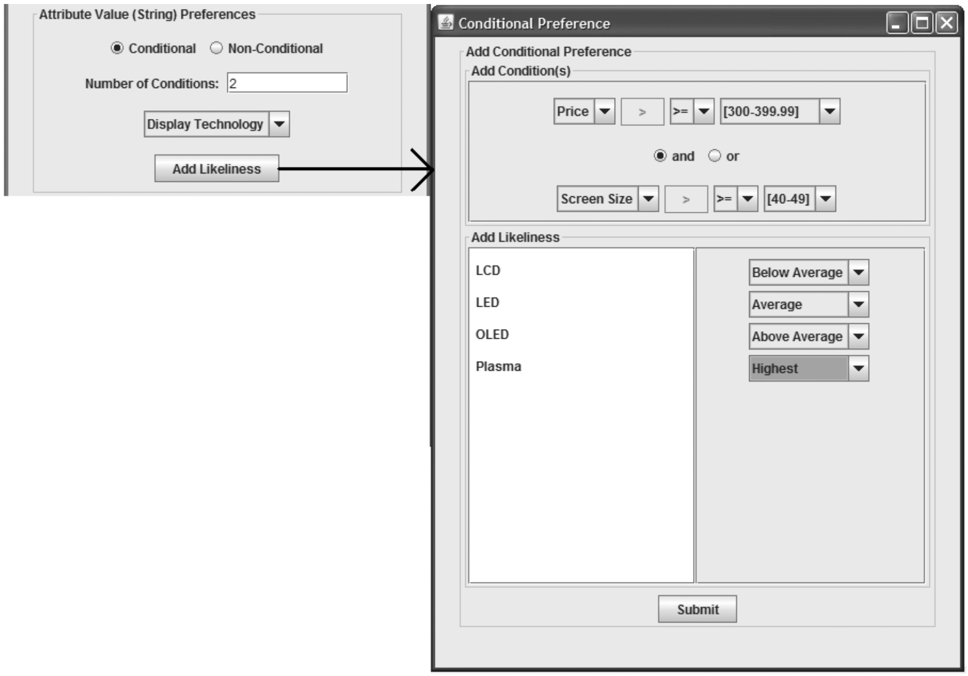


Figure 4.8: Conditional Preference on Attribute Value (String) GUI for Display Technology

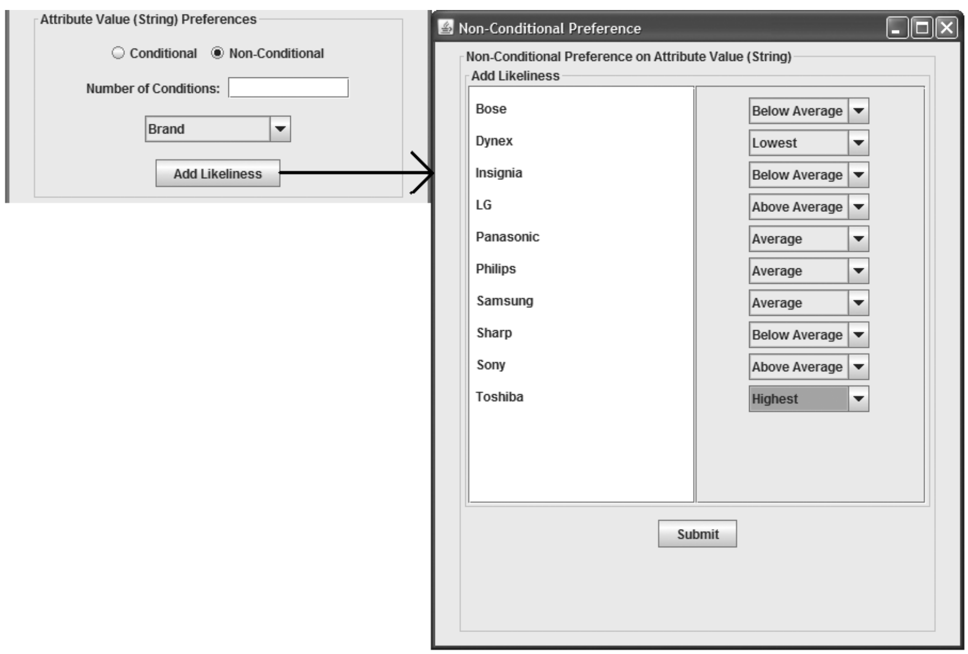


Figure 4.9: Non-Conditional Preference on Attribute Value (String) GUI for Brand

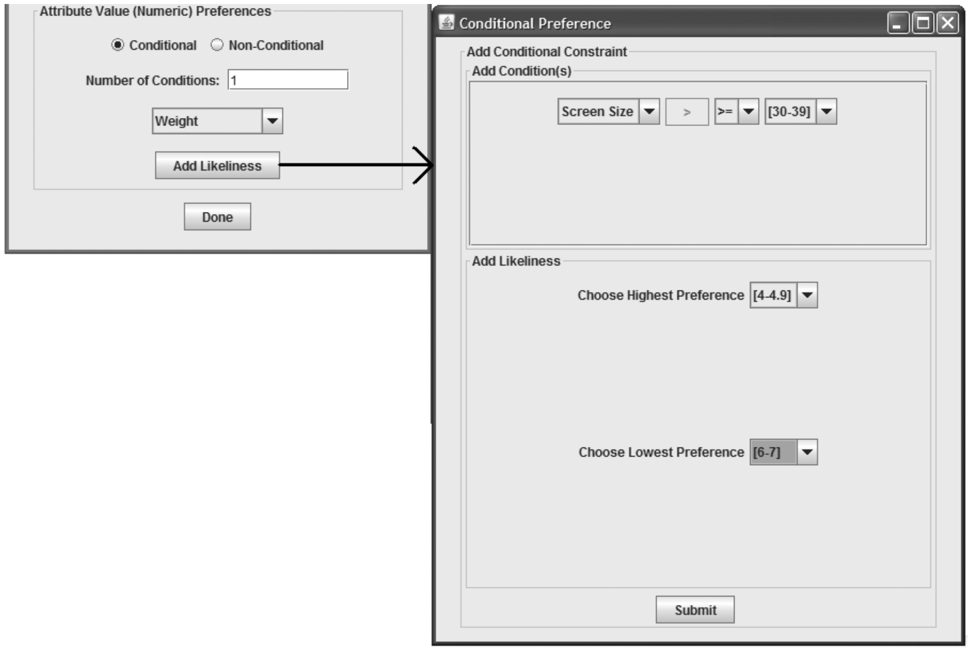


Figure 4.10: Conditional Preference on Attribute Value (Numeric) GUI for Weight

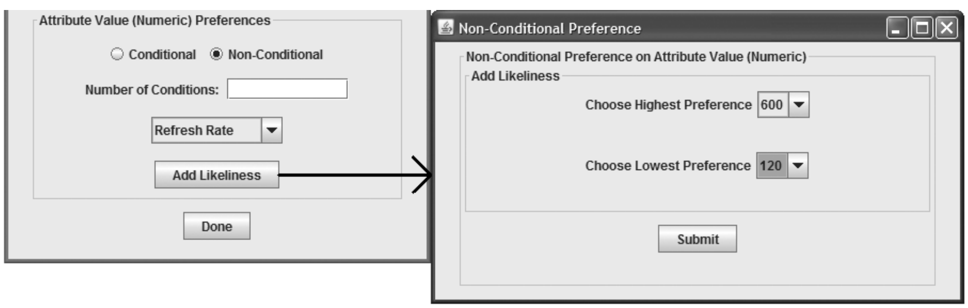
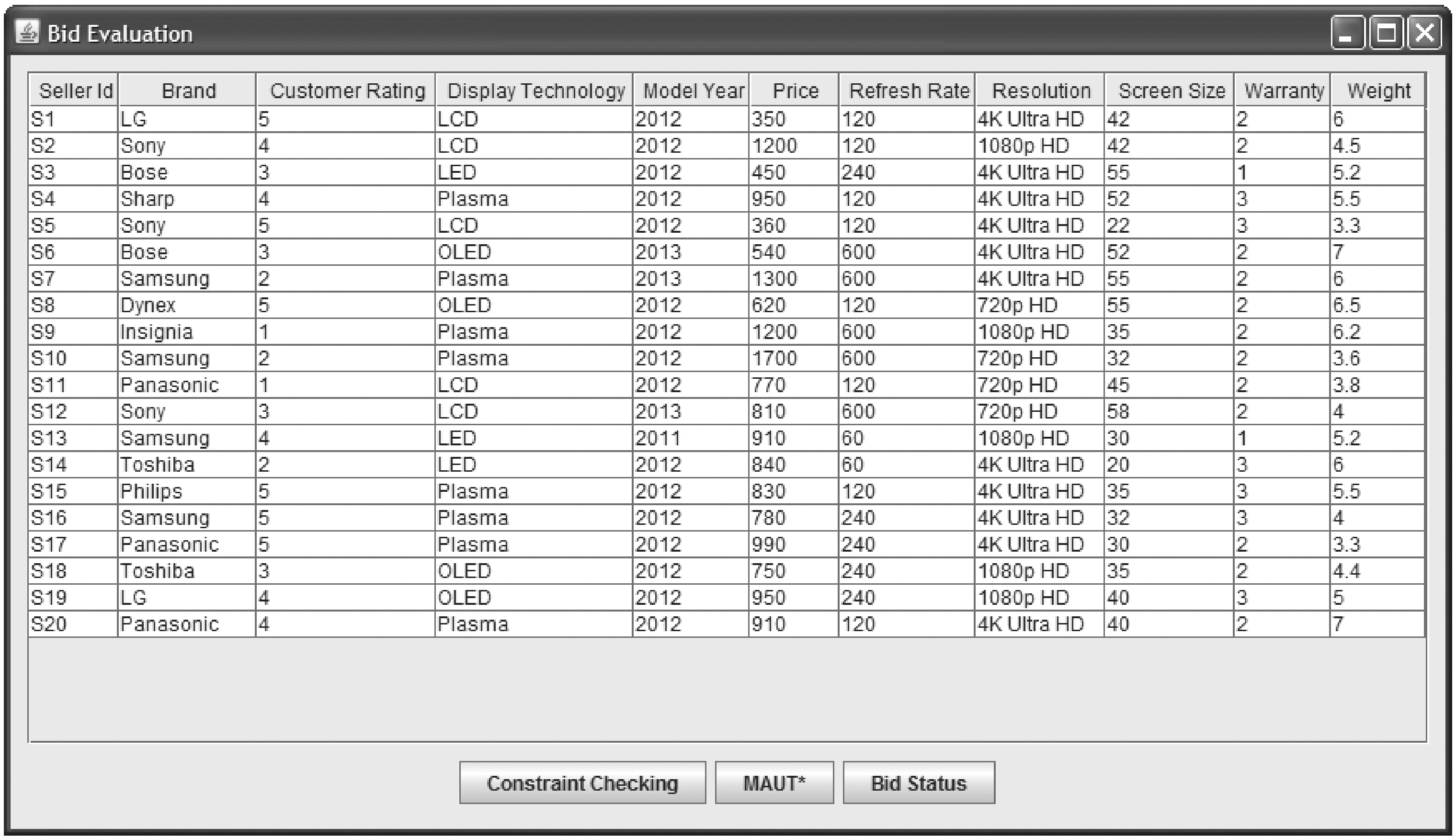


Figure 4.11: Non-Conditional Preference on Attribute Value (Numeric) GUI for Refresh Rate

4.1.5 Bid Submission

After the buyer's requirements have been submitted, the sellers can now start bidding. Assume 20 sellers participate in this multi-attribute reverse auction and they submit their first bids which are shown in Figure 4.12. For example, in Figure 4.13 our MARA system assists the seller to submit his bid.



Seller Id	Brand	Customer Rating	Display Technology	Model Year	Price	Refresh Rate	Resolution	Screen Size	Warranty	Weight
S1	LG	5	LCD	2012	350	120	4K Ultra HD	42	2	6
S2	Sony	4	LCD	2012	1200	120	1080p HD	42	2	4.5
S3	Bose	3	LED	2012	450	240	4K Ultra HD	55	1	5.2
S4	Sharp	4	Plasma	2012	950	120	4K Ultra HD	52	3	5.5
S5	Sony	5	LCD	2012	360	120	4K Ultra HD	22	3	3.3
S6	Bose	3	OLED	2013	540	600	4K Ultra HD	52	2	7
S7	Samsung	2	Plasma	2013	1300	600	4K Ultra HD	55	2	6
S8	Dynex	5	OLED	2012	620	120	720p HD	55	2	6.5
S9	Insignia	1	Plasma	2012	1200	600	1080p HD	35	2	6.2
S10	Samsung	2	Plasma	2012	1700	600	720p HD	32	2	3.6
S11	Panasonic	1	LCD	2012	770	120	720p HD	45	2	3.8
S12	Sony	3	LCD	2013	810	600	720p HD	58	2	4
S13	Samsung	4	LED	2011	910	60	1080p HD	30	1	5.2
S14	Toshiba	2	LED	2012	840	60	4K Ultra HD	20	3	6
S15	Philips	5	Plasma	2012	830	120	4K Ultra HD	35	3	5.5
S16	Samsung	5	Plasma	2012	780	240	4K Ultra HD	32	3	4
S17	Panasonic	5	Plasma	2012	990	240	4K Ultra HD	30	2	3.3
S18	Toshiba	3	OLED	2012	750	240	1080p HD	35	2	4.4
S19	LG	4	OLED	2012	950	240	1080p HD	40	3	5
S20	Panasonic	4	Plasma	2012	910	120	4K Ultra HD	40	2	7

Figure 4.12: Bid Evaluation GUI

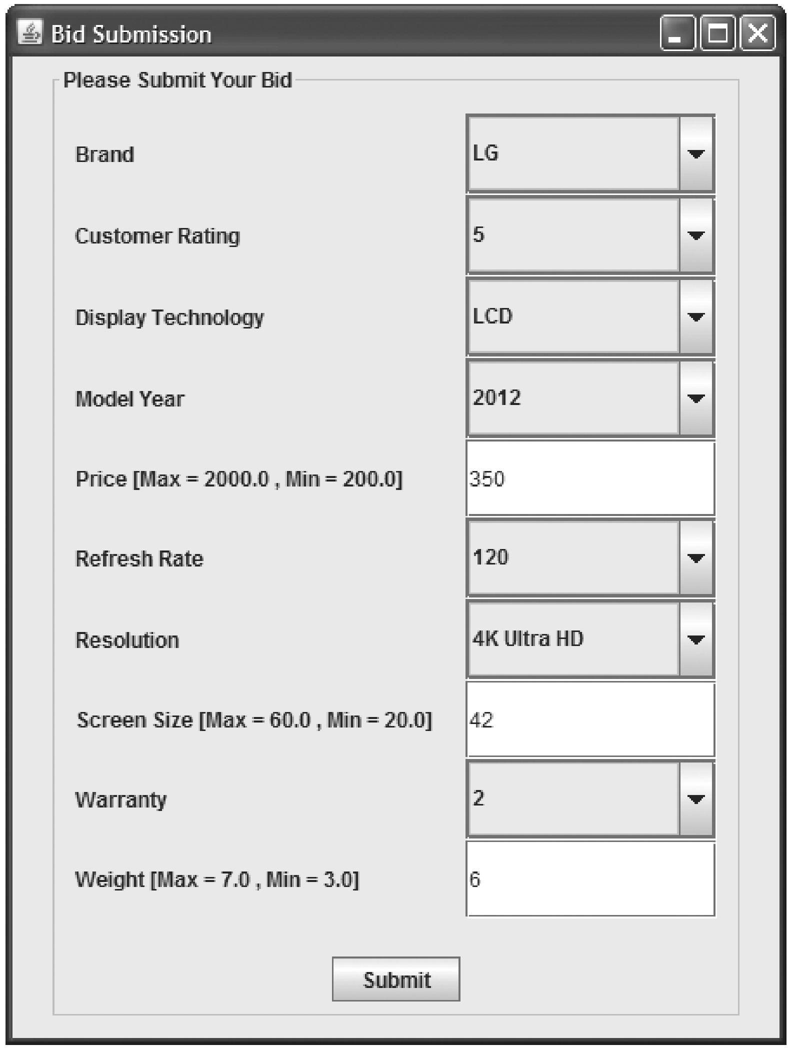


Figure 4.13: Bid Submission GUI

Our system first checks the sellers' bids with the help of constraintChecking algorithm based on the buyer's conditional and non-conditional constraints. Then, the system disqualifies the invalid bids and no further calculation will be held related to these bids.

4.1.6 Attribute Weight Calculation

Table 4.1 shows the calculation of weights of the TV attributes.

Table 4.1: Weight Calculation of Attributes

Attribute Id	**Attribute Name**	**Position**	**Rank**	**Quantitative Importance Level**	**Weight**
A1	Brand	2	9	0.75	0.188
A2	Customer Rating	8	3	0.25	0.021
A3	Display Technology	6	5	0.5	0.069
A4	Model Year	10	1	0.25	0.007
A5	Price	1	10	1	0.278
A6	Refresh Rate	7	4	0.25	0.028
A7	Resolution	5	6	0.5	0.083
A8	Screen Size	3	8	0.75	0.167
A9	Warranty	9	2	0.25	0.014
A10	Weight	4	7	0.75	0.146

4.1.7 Utility Function Value Calculation

Table 4.2 shows the utility function values of attributes of string type (Brand, Display Technology and Resolution). On the other hand, Tables 4.3 and 4.4 show the utility function values of attributes of numeric type (Customer Rating, Model Year, Price, Refresh Rate, Screen Size, Warranty and Weight).

Table 4.2: Utility Function Values of Brand, Display Technology and Resolution

Attribute Name	**Attribute Value**	**Position**	**Rank**	**Quantitative Likeliness**	**Utility Value**
Brand	Bose	8	3	0.4	0.12
	Dynex	10	1	0.2	0.02
	Insignia	9	2	0.4	0.08
	LG	3	8	0.8	0.64
	Panasonic	6	5	0.6	0.3
	Philips	5	6	0.6	0.36
	Samsung	4	7	0.6	0.42
	Sharp	7	4	0.4	0.16
	Sony	2	9	0.8	0.72
	Toshiba	1	10	1	1
Display Technology	LCD	4	1	0.4	0.1
	LED	3	2	0.6	0.3
	OLED	2	3	0.8	0.6
	Plasma	1	4	1	1
Resolution	1080p HD	1	3	1	1
	4K Ultra HD	2	2	0.8	0.533
	720p HD	3	1	0.6	0.2

Table 4.3: Utility Function Values of Customer Rating, Model Year and Price

Attribute Name	**Attribute Value**	**Quantitative Likeliness**	**Utility Value**
Customer Rating	1	-	0
	2	-	0
	3	0.2	0.1
	4	-	0.5
	5	1	1
Model Year	2011	-	0
	2012	0.2	0.333
	2013	1	1
Price	[200-299.99]	-	0
	[300-399.99]	1	1
	[400-499.99]	-	0.889
	[500-599.99]	-	0.778
	[600-699.99]	-	0.667
	[700-799.99]	-	0.556
	[800-899.99]	-	0.444
	[900-999.99]	-	0.333
	[1000-1499.99]	0.2	0.033
	[1500-2000]	-	0

Table 4.4: Utility Function Values of Refresh Rate, Screen Size, Warranty and Weight

Attribute Name	**Attribute Value**	**Quantitative Likeliness**	**Utility Value**
Refresh Rate	60	-	0
	120	0.2	0.062
	240	-	0.25
	600	1	1
Screen Size	[20-29]	-	0
	[30-39]	0.2	0.122
	[40-49]	-	0.488
	[50-60]	1	1
Warranty	1	-	0
	2	0.2	0.333
	3	1	1
Weight	[3-3.9]	-	0
	[4-4.9]	1	1
	[5-5.9]	-	0.512
	[6-7]	0.2	0.128

4.1.8 MAUT* Utility Calculation

Figure 4.14 shows the overall utility value of MAUT* and bidding status of each bid of the first round. Figure 4.14 depicts S2, S3, S5, S13 and S14 bids as Disqualified as they do not satisfy the constraints specified by the buyer. It also depicts S18 as the Winner and the remaining bids as Challenged. The overall utility value of MAUT* of S18 is the highest among all bids and for this reason this bid is announced as the

Winner. The remaining bids that named Challenged bids, score less overall utility values than the Winner bid. As an example, we show the calculation of overall utility value of MAUT* of the bid of S1.

$$
\begin{aligned}
MAUT^{*}(bid\ of\ S1) &= W_{Brand} \,.\, U_{Brand}(LG) \\
&+\ W_{Customer\ Rating} \,.\, U_{Customer\ Rating}(5) \\
&+\ W_{Display\ Technology} \,.\, U_{Display\ Technology}(LCD) \\
&+\ W_{Model\ Year} \,.\, U_{Model\ Year}(2012) \\
&+\ W_{Price} \,.\, U_{Price}(350) \\
&+\ W_{Refresh\ Rate} \,.\, U_{Refresh\ Rate}(120) \\
&+\ W_{Resolution} \,.\, U_{Resolution}(4K\ Ultra\ HD) \\
&+\ W_{Screen\ Size} \,.\, U_{Screen\ Size}(42) \\
&+\ W_{Warranty} \,.\, U_{Warranty}(2) \\
&+\ W_{Weight} \,.\, U_{Weight}(6) \\
&= (0.188 \times 0.64) + (0.021 \times 1) + (0.069 \times 0) \\
&+ (0.007 \times 0.333) + (0.278 \times 1) + (0.028 \times 0.062) \\
&+ (0.083 \times 0.533) + (0.167 \times 0.488) \\
&+ (0.014 \times 0.333) + (0.146 \times 0.128) \\
&= 0.572
\end{aligned}
$$

As the bid of S1 does not satisfy the condition(s) of the preferences for the values of *Display Technology*, the MARA system assigns 0 to the quantitative likeliness for *LCD*, thus the utility of *Display Technology* is 0 for this bid {See Section 3.2.3}.

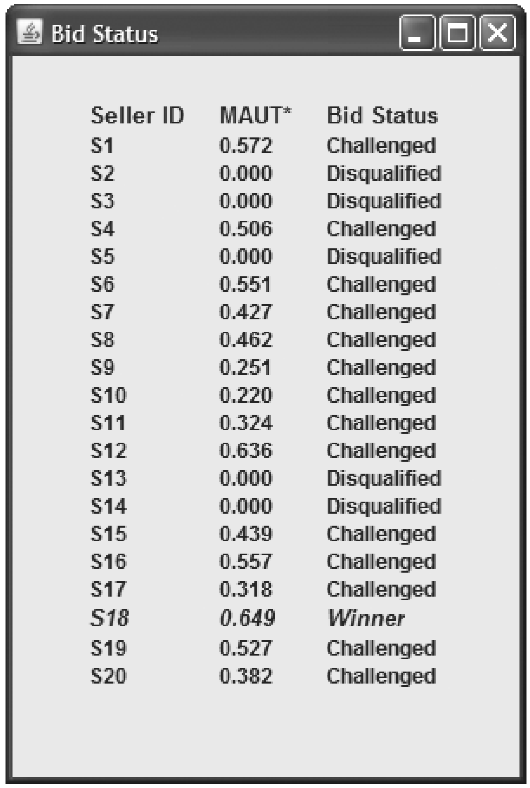


Figure 4.14: Bid Status GUI

4.2 Performance Evaluation of MARA

The execution time required by MAUT* to calculate the winner of an auction is a good measure to show the performance of our MARA protocol. Hence, we evaluate the performance of our MARA protocol by measuring the execution time (in seconds) of MAUT*. We conduct here several experiments that are executed on an AMD Athlon (tm) 64 X2 Dual Core Processor 4400+ with 3.43 GB of RAM and 2.30 GHz of processor speed.

4.2.1 Configuration of MARA Parameters

The execution time of MAUT* depends on the number of attributes of the product, the number of sellers and the number of non-conditional constraints, conditional constraints, non-conditional preferences and conditional preferences. Thus, we conduct several experiments by varying each of these parameters and report the execution time needed to return the winner. We use 10 attributes (A1-A10) {see Table 4.1}, 20 sellers (S1-S20) {see Figure 4.12} and the following non-conditional constraints, conditional constraints, non-conditional preferences and conditional preferences in all these experiments.

Non-Conditional Constraints (NCC):

(ncc1) $NULL \Rightarrow Model\ Year \neq 2011$

(ncc2) $NULL \Rightarrow Warranty \geq 2$

(ncc3) $NULL \Rightarrow Refresh\ Rate \geq 120$

(ncc4) $NULL \Rightarrow Screen\ Size \geq [30 - 39]$

(ncc5) $NULL \Rightarrow Brand \neq Dynex$

Conditional Constraints (CC):

(cc1) $(Refresh\ Rate \leq 240) \Rightarrow Price \leq [900 - 999.99]$

(cc2) $(Brand = Panasonic)\ and\ (Resolution = 4K\ Ultra\ HD) \Rightarrow$

$Weight \leq [5 - 5.9]$

(cc3) $(Brand = LG)\ or\ (Resolution = 1080p\ HD) \Rightarrow$

$Screen\ Size \leq [40 - 49]$

(cc4) $(Model\ Year = 2013)\ and\ (Warranty \geq 2) \Rightarrow$

$Brand \neq Bose$

(cc5) $(Customer\ Rating < 2)\ and\ (Model\ Year \leq 2012) \Rightarrow$

$Price \leq [500 - 599.99]$

Non-Conditional Preferences (NCP):

(ncp1) $NULL \Rightarrow Price([300 - 399.99](HS), [1000 - 1499.99](LS))$

(ncp2) $NULL \Rightarrow Refresh\ Rate(600(HS), 120(LS))$

(ncp3) $NULL \Rightarrow Brand(Bose(BA), Dynex(LS), Insignia(BA), LG(AA),$

$Panasonic(A), Philips(A), Samsung(A), Sharp(BA),$

$Sony(AA), Toshiba(HS))$

(ncp4) $NULL \Rightarrow Screen\ Size([50 - 60](HS), [30 - 39](LS))$

(ncp5) $NULL \Rightarrow Model\ Year(2013(HS), 2012(LS))$

Conditional Preferences (CP):

(cp1) $(Price \geq [300 - 399.99])$ *and* $(Screen\ Size \geq [40 - 49]) \Rightarrow$

$Display\ Technology(LCD(BA), LED(A), OLED(AA), Plasma(HS))$

(cp2) $(Refresh\ Rate \geq 120) \Rightarrow$

$Resolution(1080p\ HD(HS), 4K\ Ultra\ HD(AA), 720p\ HD(A))$

(cp3) $(Screen\ Size \geq [30 - 39]) \Rightarrow Weight([4 - 4.9](HS), [6 - 7](LS))$

(cp4) $(Price \geq [800 - 899.99]) \Rightarrow Warranty(3(HS), 2(LS))$

(cp5) $(Refresh\ Rate \geq 240)$ *or* $(Screen\ Size \geq [30 - 39]) \Rightarrow$

$Customer\ Rating(5(HS), 3(LS))$

4.2.2 Experiments

In the following, we illustrate all experiments in which Table 4.5 - Table 4.10 describe the configurations of MARA parameters and Figure 4.15 - Figure 4.20 depict the processing time of MAUT*.

Table 4.5: Execution Time of MAUT* by Varying Number of Attributes

Attributes	**Sellers**	**NCC**	**CC**	**NCP**	**CP**	**Execution Time**
6 (A1-A6)	20	3 (ncc1,ncc3, ncc5)	2 (cc1,cc5)	4 (ncp1,ncp2, ncp3,ncp5)	0	0.203
7 (A1-A7)	20	3 (ncc1,ncc3, ncc5)	2 (cc1,cc5)	4 (ncp1,ncp2, ncp3,ncp5)	1 (cp2)	0.328
8 (A1-A8)	20	4 (ncc1,ncc3, ncc4,ncc5)	3 (cc1,cc3, cc5)	5	3 (cp1,cp2, cp5)	0.437
9 (A1-A9)	20	5	4 (cc1,cc3, cc4,cc5)	5	4 (cp1,cp2, cp4,cp5)	0.438
10	20	5	5	5	5	0.453

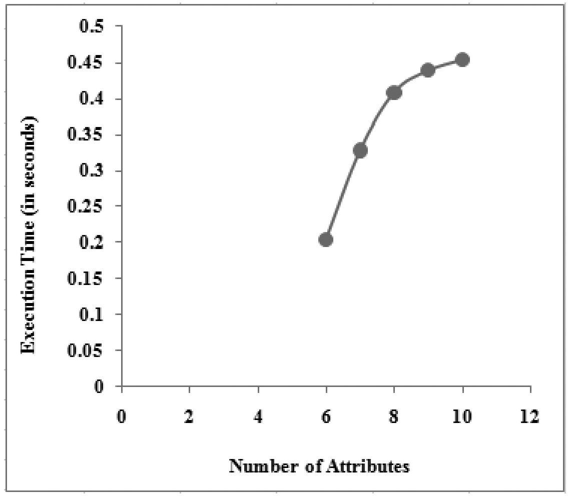


Figure 4.15: Attributes vs Execution Time

Figure 4.15 shows the execution time vs the number of attributes. As the buyer selects more attributes, the execution time of MAUT* to determine the winner increases. In this experiment, we vary the number of non-conditional constraints, conditional constraints, non-conditional preferences and conditional preferences with the change of the number of attributes but keep the number of the sellers unchanged.

Table 4.6: Execution Time of MAUT* by Varying Number of Sellers

Attributes	**Sellers**	**NCC**	**CC**	**NCP**	**CP**	**Execution Time**
10	4 (S1-S4)	5	5	5	5	0.219
10	8 (S1-S8)	5	5	5	5	0.250
10	12 (S1-S12)	5	5	5	5	0.313
10	16 (S1-S16)	5	5	5	5	0.359
10	20	5	5	5	5	0.453

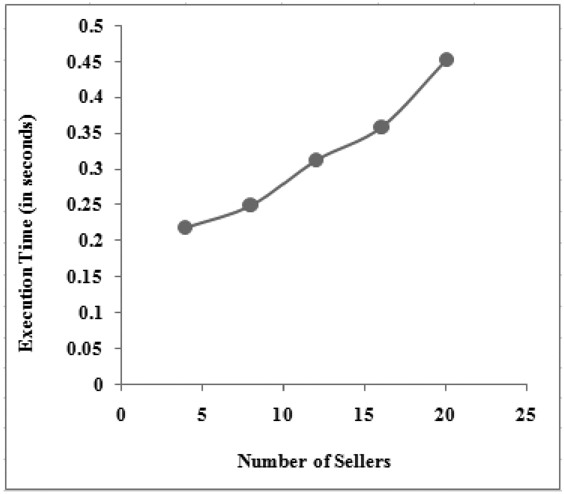


Figure 4.16: Sellers vs Execution Time

From Figure 4.16 we can deduce that as more sellers participate, the processing time of MAUT* to determine the winner increases. Here we keep the number of the attributes, non-conditional constraints, conditional constraints, non-conditional preferences and conditional preferences unchanged.

Table 4.7: Execution Time of MAUT* by Varying Number of Non-Conditional Constraints

Attributes	**Sellers**	**NCC**	**CC**	**NCP**	**CP**	**Execution Time**
10	20	1 (ncc1)	5	5	5	0.532
10	20	2 (ncc1,ncc2)	5	5	5	0.516
10	20	3 (ncc1,ncc2, ncc3)	5	5	5	0.500
10	20	4 (ncc1,ncc2, ncc3,ncc4)	5	5	5	0.469
10	20	5	5	5	5	0.453

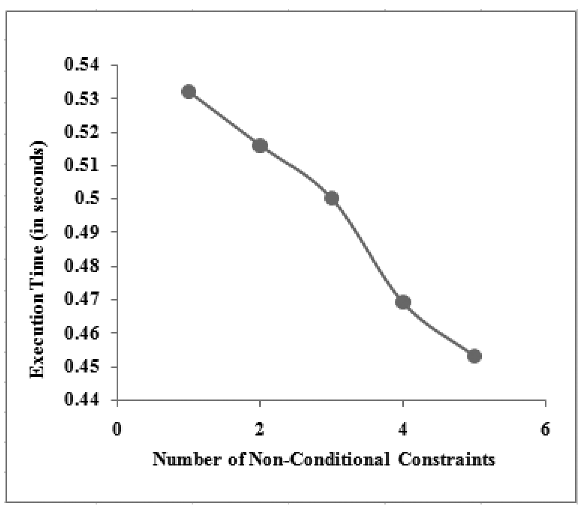


Figure 4.17: Non-conditional Constraints vs Execution Time

Figure 4.17 depicts the execution time vs the number of non-conditional constraints. As the buyer specifies more non-conditional constraints, there may be two consequences: the execution time of MAUT* may increase or remain the same. If the bids do not violet the additional non-conditional constraints, then the execution time remains the same with the increment of the non-conditional constraints. On the other hand, the increment of the non-conditional constraints creates more chances for the bids to violet them, which results to the bids to be disqualified. Thus, the execution time decreases as it needs to evaluate less bids.

Table 4.8: Execution Time of MAUT* by Varying Number of Conditional Constraints

Attributes	**Sellers**	**NCC**	**CC**	**NCP**	**CP**	**Execution Time**
10	20	5	1 (cc1)	5	5	0.578
10	20	5	2 (cc1,cc2)	5	5	0.546
10	20	5	3 (cc1,cc2, cc3)	5	5	0.546
10	20	5	4 (cc1,cc2, cc3,cc4)	5	5	0.484
10	20	5	5	5	5	0.453

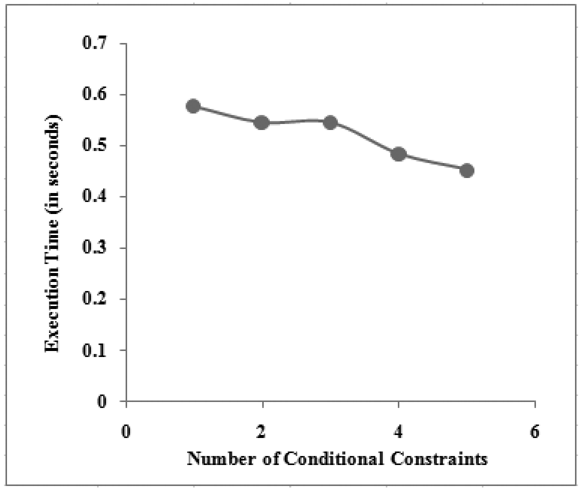


Figure 4.18: Conditional Constraints vs Execution Time

As the buyer specifies more conditional constraints, there may be two consequences: the running time of MAUT* may increase or remain the same. The increment of the conditional constraints creates more chances for the bids to violet them, which results to the bids to be disqualified. Thus, the execution time decreases as it needs to evaluate less bids. From Figure 4.18, we can see that it takes more running time when the number of the conditional constraints is 1 than that of 2, because **cc1, cc2** disqualify more bids than **cc1**. On the other hand, it takes the same running time when the number of conditional constraints are 2 and 3, because both **cc1, cc2** and **cc1, cc2, cc3** disqualify the same number of bids.

Table 4.9: Execution Time of MAUT* by Varying Number of Non-Conditional Preferences

Attributes	**Sellers**	**NCC**	**CC**	**NCP**	**CP**	**Execution Time**
10	20	5	5	1 (ncp1)	5	0.401
10	20	5	5	2 (ncp1,ncp2)	5	0.408
10	20	5	5	3 (ncp1,ncp2, ncp3)	5	0.417
10	20	5	5	4 (ncp1,ncp2, ncp3,ncp4)	5	0.437
10	20	5	5	5	5	0.453

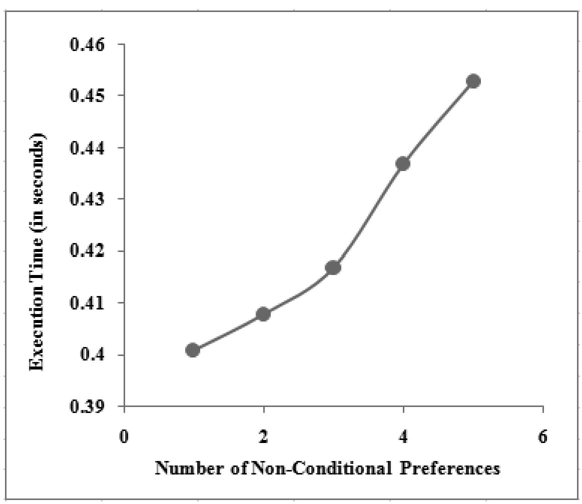


Figure 4.19: Non-conditional Preferences vs Execution Time

Figure 4.19 illustrates the execution time vs the number of non-conditional preferences. As the buyer specifies more non-conditional preferences, the execution time of MAUT* to determine the winner increases.

Table 4.10: Execution Time of MAUT* by Varying Number of Conditional Preferences

Attributes	**Sellers**	**NCC**	**CC**	**NCP**	**CP**	**Execution Time**
10	20	5	5	5	1 (cp1)	0.313
10	20	5	5	5	2 (cp1,cp2)	0.407
10	20	5	5	5	3 (cp1,cp2, cp3)	0.421
10	20	5	5	5	4 (cp1,cp2, cp3,cp4)	0.438
10	20	5	5	5	5	0.453

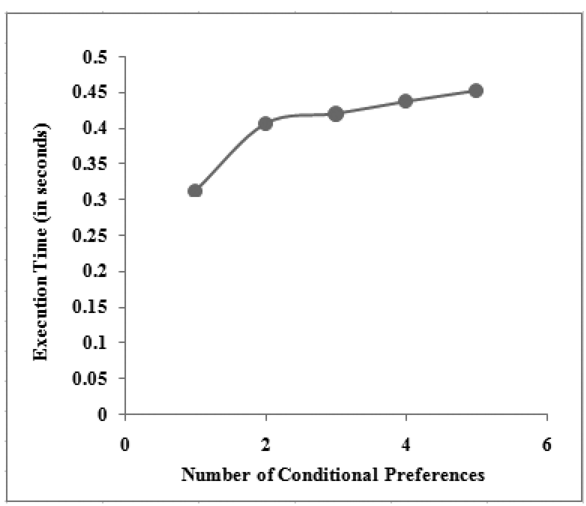


Figure 4.20: Conditional Preferences vs Execution Time

From Figure 4.20 we can note that as the buyer specifies more conditional preferences, the processing time of MAUT* to determine the winner increases.

In these experiments, we evaluate the performance of our MARA protocol by varying the number of its parameters. We illustrate all consequences considering different scenario. Based on these results and discussions, we can conclude that our MARA protocol is able to elicit the buyer's requirements and determine the winner efficiently.

Chapter 5

Conclusion and Future Work

This chapter is organized into two sections. The first section provides the nutshells of the MARA protocol and the second section suggests some promising issues for future research.

5.1 Conclusion

Constraint and preference elicitation of the buyers and winner determination in multi-attribute reverse auctions are complex problems to solve. In fact, the buyer feels comfort to specify his preferences qualitatively. On the other hand, quantitative values are efficient for calculation. This contradictory situation leads to a complex problem. Our MARA protocol can solve this problem successfully and efficiently. The system is able to elicit the buyer's non-conditional and conditional constraints as well as qualitative non-conditional and conditional preferences of a product. Requirements elicitation is one of the most challenging and important issues in multi-attribute reverse auctions. Our MARA system can convert qualitative requirements into quantitative ones by integrating several conversion methods with MAUT and determine the winner efficiently.

In this thesis, we developed a MARA system and discussed it with concrete examples to better understand the process step by step. We explored experimentally how MARA works and the way it determines the winner. We also evaluated the performance of MARA by varying the number of attributes, sellers, non-conditional constraints, conditional constraints, non-conditional preferences and conditional preferences. Moreover, we depicted how our MARA system assists both the buyers and sellers through friendly graphical user interfaces in detail.

Here we conclude this section by highlighting the main features and benefits of our MARA protocol.

- The system assists the buyer to specify his requirements through friendly graphical user interfaces. It provides the buyer with the complete guidelines and several steps from choosing the product he wants to buy to expressing his constraints and preferences.
- The system also assists the seller to submit his bid via friendly graphical user interfaces.
- MARA protocol helps the seller with realizing the valuation of his bids in each round. Since our system can have multiple rounds, it provides the seller with the opportunity to bid more wisely in the next rounds.
- In MARA, we follow vickrey protocol. By using this type of auction protocol MARA system hides the bids during each round and shows these to all the sellers after each round.

- In MARA protocol, there are three agents named Admin, Graphical User Interface and Winner Determination. By distributing specific duties and responsibilities among these agents, MARA protocol makes the system robust, reliable, scalable and secured.

- Performance is one of the most significant aspects of our MARA system. It takes into account the advantages of the notion of an agent-based system which derives from the idea of having agents carrying individual set of beliefs, desires and intentions and performing their specific tasks upon request. The MARA system can determine the winner efficiently.

- The agent-based MARA system is implemented with a generic design that has the flexibility necessary for future maintenance and extensibility architecture. The MARA system allows the addition of a new agent and removal of an existing agent from the system whose duty is to perform a specific task of the system. For instance, it is possible to have another agent that is responsible for payment of purchased item by the buyer to the existing MARA system.

5.2 Future Work

There are some promising issues of our proposed MARA protocol that can pave the ways for future research. By addressing these issues the MARA protocol can be more efficient and well-established. Some future work directions are discussed in the following.

- **Methodology:** in MARA protocol we used MAUT to elicit the buyer's requirements and to determine the winner. But it can also be possible to use some other methodologies for these tasks. Besides MAUT, there are other decision analysis techniques such as Analytic Hierarchy Process (AHP) and Conjoint Analysis (CA) [7]. Similar to MAUT, these techniques are also used in software packages for decision makers. Moreover, for the purpose of assigning weights, besides MAUT, there are other approaches such as Simple Multi-Attribute Rating Technique (SMART) and Weight determination based on Ordinal Ranking of Alternatives (WORA) [20].

- **Testing domain:** in this thesis, we evaluated the performance of the MARA protocol based on a case study of buying a TV in a multi-attribute reverse auction. This protocol can also be tested on some real world datasets of auction systems such as eBay (www.eBay.com), Bidz (www.bidz.com), and Priceline (www.priceline.com).

- **Preference:** since the buyer feels more comfort to specify his preferences qualitatively, our MARA system allows the buyer to specify his preferences only qualitatively. But the system can be improved by providing the buyer the opportunity to specify his preferences also quantitatively. By allowing the buyer to specify his preferences qualitatively on some attributes as well as quantitatively on other attributes, our MARA system can be more acceptable and well-established.

Bibliography

[1] S. Adhau, M. L. Mittal, and A. Mittal, *A Multi-Agent System for Distributed Multi-Project Scheduling: An Auction-based Negotiation Approach*, Engineering Applications of Artificial Intelligence **25** (2012), no. 8, 1738–1751.

[2] E. Alanazi and M. Mouhoub, *Managing Qualitative Preferences with Constraints*, Neural Information Processing (T. Huang, Z. Zeng, C. Li, and C. S. Leung, eds.), Lecture Notes in Computer Science, vol. 7665, Springer Berlin Heidelberg, 2012, pp. 653–662.

[3] E. Alanazi, M. Mouhoub, and B. Mohammed, *A Preference-Aware Interactive System for Online Shopping*, Computer and Information Science **5** (2012), no. 6, 33–42.

[4] H. Avancini and A. Amandi, *A Java Framework for Multi-Agent Systems*, SADIO Electronic Journal of Informatics and Operations Research **3** (2000), no. 1, 1–12.

[5] F. Bachmann, L. Bass, S. Carriere, P. Clements, D. Garlan, J. Ivers, R. Nord, and M. Little, *Software Architecture Documentation in Practice: Documenting Architectural Layers*, Special Report CMU/SEI-2000-SR-004, Software Engineering Institute, Carnegie Mellon University, 2000.

[6] C. Beam and A. Segev, *Auctions on the Internet: A Field Study*, Haas School of Business, University of California Berkeley (1998), 1–33.

[7] M. Bichler, *An Experimental Analysis of Multi-Attribute Auctions*, Decision Support Systems **29** (2000), no. 3, 249–268.

[8] M. Bichler, L. Juhnyoung, S. L. Ho, and J. Y. Chung, *ABSolute: an Intelligent Decision Making Framework for e-sourcing*, Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, WECWIS, 2001, pp. 195–201.

[9] M. Bichler and J. Kalagnanam, *Configurable Offers and Winner Determination in Multi-Attribute Auctions*, European Journal of Operational Research **160** (2005), no. 2, 380–394.

[10] M. Bichler, M. Kaukal, and A. Segev, *Multi-Attribute Auctions for Electronic Procurement*, First IBM IAC Workshop on Internet Based Negotiation Technologies, 1999, pp. 154–165.

[11] S. M. Bohte, E. Gerding, and H. L. Poutre, *Market-based Recommendation: Agents that Compete for Consumer Attention*, ACM Transactions on Internet Technology **4** (2004), no. 4, 420–448.

[12] D. P. Boulet and N. M. Fraser, *Improving Preference Elicitation for Decision Support Systems*, Intelligent Systems for the 21st Century., IEEE International Conference on Systems, Man and Cybernetics, vol. 2, 1995, pp. 1574–1579.

[13] F. Branco, *The Design of Multidimensional Auctions*, The RAND Journal of Economics **28** (1997), no. 1, 63–81.

[14] L. Braubach and A. Pokahr, *Jadex Tutorial*, Hamburg : University of Hamburg, 2003.

[15] L. Braubach, A. Pokahr, and W. Lamersdorf, *Jadex: A Short Overview*, Main Conference Net.Object Days, 2004, pp. 195–207.

[16] D. Braziunas, *Computational Approaches to Preference Elicitation*, Tech. Rep, Department of Computer Science, University of Toronto (2006).

[17] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, *A Knowledge Level Software Engineering Methodology for Agent Oriented Programming*, Proceedings of the fifth international conference on Autonomous agents (New York, NY, USA), AGENTS '01, ACM, 2001, pp. 648–655.

[18] H. Burkhard, M. Hannebauer, and J. Wendler, *Belief-Desire-Intention Deliberation in Artificial Soccer*, AI Magazine **19** (1998), no. 3, 87–93.

[19] B. Chaib-draa and F. Dignum, *Trends in Agent Communication Language*, Computational Intelligence (2002), 89–101.

[20] T. S. Chandrashekar, Y. Narahari, C. H. Rosa, D. M. Kulkarni, J. D. Tew, and P. Dayama, *Auction-Based Mechanisms for Electronic Procurement*, IEEE Transactions on Automation Science and Engineering **4** (2007), no. 3, 297–321.

[21] A. Chavez, D. Dreilinger, R. Guttman, and P. Maes, *A Real-life Experiment in Creating an Agent Marketplace*, Software Agents and Soft Computing Towards Enhancing Machine Intelligence (H. S. Nwana and N. Azarmi, eds.), Lecture Notes in Computer Science, vol. 1198, Springer Berlin Heidelberg, 1997, pp. 160–179.

[22] Y. Che, *Design Competition through Multidimensional Auctions*, Rand Journal of Economics **24** (1993), 668–680.

[23] L. Chen and P. Pu, *Survey of Preference Elicitation Methods*, Rapport technique (2004).

[24] C. Cheng, *Solving a Sealed-bid Reverse Auction Problem by Multiple-Criterion Decision-Making Methods*, Computers & Mathematics with Applications **56** (2008), no. 12, 3261–3274.

[25] D. N. Chin and A. Porage, *Acquiring User Preferences for Product Customization*, User Modeling (M. Bauer, P. J. Gmytrasiewicz, and J. Vassileva, eds.), Lecture Notes in Computer Science, vol. 2109, Springer Berlin Heidelberg, 2001, pp. 95–104.

[26] M. Cossentino, *Different Perspectives in Designing Multi-Agent Systems*, AGES '02 workshop at NODe02, 2002, pp. 8–9.

[27] M. Cossentino, *Design Process Documentation Template*, Tech. Report SC00097B, IEEE Foundation for Intelligent Physical Agents (FIPA), 2012.

[28] P. Cramton, Y. Shoham, and R. Steinberg, *An Overview of Combinatorial Auctions*, SIGecom Exchanges **7** (2007), no. 1, 3–14.

[29] P. Cuesta, A. Gomez, J. C. Gonzalez, and F. J. Rodriguez, *A Framework for Evaluation of Agent Oriented Methodologies*, Proceedings of the Conference of the Spanish Association for Artificial Intelligence, vol. 147, 2003, pp. 151–152.

[30] K. H. Dam and M. Winikoff, *Comparing Agent-Oriented Methodologies*, Agent-Oriented Information Systems (P. Giorgini, B. Henderson-Sellers, and

M. Winikoff, eds.), Lecture Notes in Computer Science, vol. 3030, Springer Berlin Heidelberg, 2004, pp. 78–93.

[31] E. David, R. Azoulay-Schwartz, and S. Kraus, *Bidding in Sealed-bid and English Multi-Attribute Auctions*, Decision Support Systems **42** (2006), no. 2, 527–556.

[32] S. A. DeLoach, M. F. Wood, and C. H. Sparkman, *Multiagent Systems Engineering*, International Journal of Software Engineering and Knowledge Engineering **11** (2001), no. 3, 231–258.

[33] B. Dennis and C. G. Healey, *A Survey of Preference Elicitation*, Tech. Rep, Knowledge Discovery Lab, Department of Computer Science, North Carolina State University (2003).

[34] M. d'Inverno, D. Kinny, M. Luck, and M. Wooldridge, *A Formal Specification of dMARS*, Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages (London, UK), ATAL '97, Springer-Verlag, 1998, pp. 155–176.

[35] R. Engelbrecht-Wiggans, E. Haruvy, and E. Katok, *A Comparison of Buyer-Determined and Price-Based Multiattribute Mechanisms*, Marketting Science **26** (2007), no. 5, 629–641.

[36] K. Gajos and D. S. Weld, *Preference Elicitation for Interface Optimization*, Proceedings of the 18th annual ACM symposium on User interface software and technology (New York, USA), UIST '05, ACM, 2005, pp. 173–182.

[37] D. Garlan and M. Shaw, *An Introduction to Software Architecture*, Tech. report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[38] A. M. Geoffrion and R. Krishnan, *Prospects for Operations Research in the e-business Era*, Interfaces **31** (2001), no. 2, 6–36.

[39] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, *The Belief-Desire-Intention Model of Agency*, Intelligent Agents V: Agents Theories, Architectures, and Languages, Springer, 1999, pp. 1–10.

[40] M. P. Georgeff and A. L. Lansky, *Reactive Reasoning and Planning*, AAAI, vol. 87, 1987, pp. 677–682.

[41] F. Ghavamifar, S. Sadaoui, and M. Mouhoub, *Winner Determination Based on Preference Elicitation Methods*, Advances in Artificial Intelligence (A. Farzindar and V. Keselj, eds.), Lecture Notes in Computer Science, vol. 6085, Springer Berlin Heidelberg, 2010, pp. 332–335.

[42] F. Ghavamifar, S. Sadaoui, and M. Mouhoub, *Preference Elicitation and Winner Determination in Multi-Attribute Auctions*, FLAIRS Conference (R. C. Murray and P. M. McCarthy, eds.), AAAI Press, 2011, pp. 93–94.

[43] Y. Guo, J. P. Muller, and C. Weinhardt, *Learning User Preferences for Multi-Attribute Negotiation: an Evolutionary Approach*, Multi-Agent Systems and Applications III, Springer, 2003, pp. 303–313.

[44] K. L. Gwebu, M. Y. Hu, and M. S. Shanker, *An Experimental Investigation into the Effects of Information Revelation in Multi-Attribute Reverse Auctions*, Behaviour & Information Technology **31** (2012), no. 6, 631–644.

[45] B. Henderson-Sellers and I. Gorton, *Agent-based Software Development Methodologies*, White Paper, Summary of Workshop at the OOPSLA, vol. 2003, 2002.

[46] D. Jannach and G. Kreutler, *Personalized User Preference Elicitation for e-Services*, The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, 2005, pp. 604–611.

[47] S. Kameshwaran, L. Benyoucef, and X. Xie, *Design of Progressive Auctions for Procurement based on Lagrangian Relaxation*, Seventh IEEE International Conference on E-Commerce Technology, 2005, pp. 9–16.

[48] M. Karny and T. V. Guy, *Preference Elicitation in Fully Probabilistic Design of Decision Strategies*, 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 5327–5332.

[49] G. E. Kersten, P. Pontrandolfo, R. Vahidov, and D. Gimon, *Negotiation and Auction Mechanisms: Two Systems and Two Experiments*, E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life (M. J. Shaw, D. Zhang, and W. T. Yue, eds.), Lecture Notes in Business Information Processing, vol. 108, Springer Berlin Heidelberg, 2012, pp. 399–412.

[50] D. Kinny, M. Georgeff, and A. Rao, *A Methodology and Modelling Technique for Systems of BDI Agents*, Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world : agents breaking away: agents breaking away (Secaucus, NJ, USA), MAAMAW '96, Springer-Verlag New York, Inc., 1996, pp. 56–71.

[51] P. Klemperer, *Auctions: Theory and Practice*, Economics Papers, Economics Group, Nuffield College, University of Oxford (2004).

[52] M. Luck, P. McBurney, and C. Preist, *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*, AgentLink/University of Southampton, 2003.

[53] G. Manoochehri and C. Lindsy, *Reverse Auctions: Benefits, Challenges, and Best Practices*, California Journal of Operations Management **6** (2008), no. 1, 123–130.

[54] L. Nie, X. Xu, and D. Zhan, *Coevolutionary Computation Based Iterative Multi-Attribute Auctions*, Enterprise Interoperability III (K. Mertins, R. Ruggaber, K. Popplewell, and X. Xu, eds.), Springer London, 2008, pp. 461–469.

[55] D. C. Parkes and J. Kalagnanam, *Models for Iterative Multiattribute Procurement Auctions*, Management Science **51** (2005), no. 3, 435–451.

[56] A. Pokahr, L. Braubach, and W. Lamersdorf, *Jadex: A BDI Reasoning Engine*, Multi-Agent Programming (R. H. Bordini, M. Dastani, J. Dix, and A. Fallah Seghrouchni, eds.), Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 15, Springer US, 2005, pp. 149–174.

[57] S. Prestwich, F. Rossi, K. B. Venable, and T. Walsh, *Constraint-based Preferential Optimization*, AAAI, vol. 5, 2005, pp. 461–466.

[58] I. Rodriguez and N. Lopez, *Analyzing the Privacy of a Vickrey Auction Mechanism*, IJEBR **2** (2006), no. 3, 17–27.

[59] P. H. M. P. Roelofsma and M. C. Schut, *Preference Elicitation without Numbers*, Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3 (Washington, DC, USA), AAMAS '04, IEEE Computer Society, 2004, pp. 1428–1429.

[60] A. R. Sadeghi, M. Schunter, and S. Steinbrecher, *Private Auctions with Multiple Rounds and Multiple Items*, Proceedings of the 13th International Workshop on Database and Expert Systems Applications, 2002, pp. 423–427.

[61] K. T. Seow and K. M. Sim, *Collaborative Assignment using Belief-Desire-Intention Agent Modeling and Negotiation with Speedup Strategies*, Information Sciences **178** (2008), no. 4, 1110–1132.

[62] M. Shajari and A. A. Ghorbani, *Application of Belief-Desire-Intention Agents in Intrusion Detection & Response*, Proceedings of the Second Annual Conference on Privacy, Security and Trust, PST, Citeseer, 2004, pp. 181–191.

[63] D. Shao-bin, F. Yu-qiang, L. Ke-xing, and S. Xun-cheng, *Preference Elicitation of the NSS in the Ecommerce*, International Conference on Management Science and Engineering, ICMSE, 2007, pp. 135–140.

[64] D. Shih, H. Huang, and D. C. Yen, *A Secure Reverse Vickrey Auction Scheme with Bid Privacy*, Information Sciences **176** (2006), no. 5, 550–564.

[65] L. R. Smeltzer and A. Carr, *Reverse Auctions in Industrial Marketing and Buying*, Business Horizons **45** (2002), no. 2, 47–52.

[66] T. R. Srinath, M. P. Singh, and A. R. Pais, *Anonymity and Verifiability in Multi-Attribute Reverse Auction*, arXiv preprint arXiv:1109.0359 (2011).

[67] D. P. Sugijarto, N. Jailani, M. Mukhtar, Y. Yahya, Z. Abdullah, and S. Abdullah, *A Framework for the Development of a Multi Attributes Reverse Auction for B2B e-Marketplace*, International Symposium on Information Technology, vol. 1, 2008, pp. 1–7.

[68] S. Talluri, R. Narasimhan, and S. Viswanathan, *Information Technologies for Procurement Decisions: A Decision Support System for Multi-Attribute e-Reverse Auctions*, International Journal of Production Research **45** (2007), no. 11, 2615–2628.

[69] J. E. Teich, H. Wallenius, J. Wallenius, and O. R. Koppius, *Emerging Multiple Issue e-Auctions*, European Journal of Operational Research **159** (2004), no. 1, 1–16.

[70] J. E. Teich, H. Wallenius, J. Wallenius, and A. Zaitsev, *A Multi-Attribute e-Auction Mechanism for Procurement: Theoretical Foundations*, European Journal of Operational Research **175** (2006), no. 1, 90–100.

[71] A. Tveit, *A Survey of Agent-Oriented Software Engineering*, NTNU Computer Science Graduate Student Conference, Norwegian University of Science and Technology, 2001.

[72] W. Vickrey, *Counterspeculation, Auctions, and Competitive Sealed Tenders*, The Journal of Finance **16** (1961), no. 1, 8–37.

[73] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb, *Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead*, Management Science **54** (2008), no. 7, 1336–1349.

[74] C. L. Williams, *An Overview of Reverse Auctions*, Southwest Decision Sciences Institute Conference, 2010.

[75] M. Yan and Y. Yuan, *A Multi-attribute Reverse Auction Decision Making Model Based on Linear Programming*, Systems Engineering Procedia (2012).

[76] J. Yu and S. Han, *Application of Reverse Auction in the Supplier Selection*, IEEE 17Th International Conference on Industrial Engineering and Engineering Management, IE&EM, 2010, pp. 1267–1270.

[77] J. Zhang and P. Pu, *Survey of Solving Multi-Attribute Decision Problems*, Tech. report, Technical Report IC/2004/54, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2004.

[78] G. Zhu, S. Sangwan, and T. Lu, *Mechanism Design of Online Multi-Attribute Reverse Auction*, 42nd Hawaii International Conference on System Sciences, HICSS, 2009, pp. 1–7.

[79] H. Zong-you and W. Ding-Wei, *Bidding Strategies in Sealed-bid Reverse Multi-attribute Auctions*, Chinese Control and Decision Conference, CCDC, 2011, pp. 1762–1767.

Appendix A

ADFs of MARA Agents

In this chapter we include the Agent Definition Files of the three agents of MARA: Admin, Graphical User Interface and Winner Determination agents with MARA application descriptor in XML format respectively.

Listing A.1: Admin Agent

```
<agent xmlns="http://jadex.sourceforge.net/jadex"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema instance"
        xsi:schemaLocation="http://jadex.sourceforge.net/jadex
                        http://jadex.sourceforge.net/jadex bdi 2.0.xsd"
        name="AdminAgent"
        package="Admin">

<imports>
    <import>java.util.*</import>
    <import>java.util.logging.*</import>
    <import>jadex.commons.*</import>
    <import>jadex.rules.rulesystem.rules.functions.*</import>
    <import>jadex.base.fipa.SFipa</import>
    <import>jadex.base.fipa.*</import>
    <import>jadex.planlib.*</import>
```

```
    <import>jadex.bridge.service.*</import>
    <import>jadex.commons.future.*</import>
</imports>

<capabilities>
    <capability name="dfcap" file="jadex.bdi.planlib.df.DF"/>
</capabilities>

<goals>
    <maintaingoalref name="df_keep_registered">
        <concrete ref="dfcap.df_keep_registered"/>
    </maintaingoalref>
</goals>

<plans>
    <plan name="validUserResp">
        <body class="validUserRespPlan"/>
        <trigger>
            <messageevent ref="request_user"></messageevent>
        </trigger>
    </plan>
    <plan name="proAttrResp">
        <body class="proAttrRespPlan"/>
        <trigger>
            <messageevent ref="request_product"></messageevent>
        </trigger>
    </plan>
    <plan name="aucAttrResp">
        <body class="aucAttrRespPlan"/>
        <trigger>
            <messageevent ref="request_attribute"></messageevent>
```

```
        </trigger>
    </plan>
    <plan name="aucAttrOrderResp">
        <body class="aucAttrOrderRespPlan"/>
        <trigger>
            <messageevent ref="request_position"></messageevent>
        </trigger>
    </plan>
    <plan name="aucAttrStrValOrderResp">
        <body class="aucAttrStrValOrderRespPlan"/>
        <trigger>
            <messageevent ref="request_stringposition"></messageevent>
        </trigger>
    </plan>
    <plan name="aucAttrPrefResp">
        <body class="aucAttrPrefRespPlan"/>
        <trigger>
            <messageevent ref="request_attpreference"></messageevent>
        </trigger>
    </plan>
    <plan name="nonCondConstResp">
        <body class="nonCondConstRespPlan"/>
        <trigger>
            <messageevent ref="request_noncondconst"></messageevent>
        </trigger>
    </plan>
    <plan name="condConstResp">
        <body class="condConstRespPlan"/>
        <trigger>
            <messageevent ref="request_condconst"></messageevent>
        </trigger>
```

```
</plan>
<plan name="nonCondStrPrefResp">
    <body class="nonCondStrPrefRespPlan"/>
    <trigger>
        <messageevent ref="request_noncondstrpref"></messageevent>
    </trigger>
</plan>
<plan name="condStrPrefResp">
    <body class="condStrPrefRespPlan"/>
    <trigger>
        <messageevent ref="request_condstrpref"></messageevent>
    </trigger>
</plan>
<plan name="nonCondNumPrefResp">
    <body class="nonCondNumPrefRespPlan"/>
    <trigger>
        <messageevent ref="request_noncondnumpref"></messageevent>
    </trigger>
</plan>
<plan name="condNumPrefResp">
    <body class="condNumPrefRespPlan"/>
    <trigger>
        <messageevent ref="request_condnumpref"></messageevent>
    </trigger>
</plan>
<plan name="bidSubmissionResp">
    <body class="bidSubmissionRespPlan"/>
    <trigger>
        <messageevent ref="request_bid"></messageevent>
    </trigger>
</plan>
```

```
    <plan name="constChkResp">
        <body class="constChkRespPlan"/>
        <trigger>
            <messageevent ref="request_constcheck"></messageevent>
        </trigger>
    </plan>
    <plan name="bidUtilityResp">
        <body class="bidUtilityRespPlan"/>
        <trigger>
            <messageevent ref="request_mautstar"></messageevent>
        </trigger>
    </plan>
</plans>

<events>
    <messageevent name="request_user" direction="receive" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.REQUEST</value>
        </parameter>
        <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("user")</match>
    </messageevent>
    <messageevent name="request_product" direction="receive" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.REQUEST</value>
        </parameter>
```

```
      <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("product")</match>
    </messageevent>
    <messageevent name="request_attribute" direction="receive" type="fipa">
      <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
      </parameter>
      <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("attribute")</match>
    </messageevent>
    <messageevent name="request_position" direction="receive" type="fipa">
      <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
      </parameter>
      <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("position")</match>
    </messageevent>
    <messageevent name="request_stringposition" direction="receive" type="fipa">
      <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
      </parameter>
      <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("stringposition")</match>
    </messageevent>
```

```
<messageevent name="request_attpreference" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("attpreference")</match>
</messageevent>
<messageevent name="request_noncondconst" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("noncondconst")</match>
</messageevent>
<messageevent name="request_condconst" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("condconst")</match>
</messageevent>
<messageevent name="request_noncondstrpref" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
```

```
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("noncondstrpref")</match>
</messageevent>
<messageevent name="request_condstrpref" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("condstrpref")</match>
</messageevent>
<messageevent name="request_noncondnumpref" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("noncondnumpref")</match>
</messageevent>
<messageevent name="request_condnumpref" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("condnumpref")</match>
```

```
</messageevent>
<messageevent name="request_bid" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed
        ">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content
        ).startsWith("bid")</match>
</messageevent>
<messageevent name="request_constcheck" direction="receive" type=
    "fipa">
    <parameter name="performative" class="String" direction="fixed
        ">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content
        ).startsWith("constcheck")</match>
</messageevent>
<messageevent name="request_mautstar" direction="receive" type="
    fipa">
    <parameter name="performative" class="String" direction="fixed
        ">
        <value>SFipa.REQUEST</value>
    </parameter>
    <match>$content instanceof String &amp;&amp; ((String)$content
        ).startsWith("mautstar")</match>
</messageevent>
<messageevent name="inform" direction="send" type="fipa">
    <parameter name="performative" class="String" direction="fixed
        ">
        <value>SFipa.INFORM</value>
```

```
        </parameter>
    </messageevent>
    <messageevent name="failure" direction="send" type="fipa">
        <parameter name="performative" class="String" direction="fixed
            ">
            <value>SFipa.FAILURE</value>
        </parameter>
    </messageevent>
</events>

<properties>
    <property name="logging.level">Level.INFO</property>
    <property name="logging.useParentHandlers">false</property>
    <property name="dfservice" class="IFuture">SServiceProvider.
         getService($scope.getServiceProvider(), IDF.class,
         RequiredServiceInfo.SCOPE_PLATFORM)</property>
</properties>

<configurations>
    <configuration name="default">
        <goals>
            <initialgoal ref="df_keep_registered">
                <parameter ref="description">
                    <value>
                        $properties.dfservice.createDFComponentDescription
                            (null,
                        $properties.dfservice.createDFServiceDescription("
                            service_admin", "admin demo", "admin agent"))
                    </value>
                </parameter>
                <parameter ref="leasetime">
```

```
                    <value>550000</value>
                </parameter>
            </initialgoal>
        </goals>
    </configuration>
</configurations>

</agent>
```

Listing A.2: Graphical User Interface Agent

```
<agent xmlns="http://jadex.sourceforge.net/jadex"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema instance"
    xsi:schemaLocation="http://jadex.sourceforge.net/jadex
                        http://jadex.sourceforge.net/jadex bdi 2.0.xsd"
    name="GUIAgent"
    package="GUI">

<imports>
    <import>jadex.base.fipa.SFipa</import>
    <import>jadex.commons.*</import>
    <import>jadex.bdi.planlib.GuiCreator</import>
    <import>jadex.rules.rulesystem.rules.functions.*</import>
    <import>jadex.base.fipa.*</import>
    <import>java.util.*</import>
    <import>jadex.bdi.planlib.protocols.*</import>
    <import>jadex.bdi.runtime.*</import>
    <import>jadex.bridge.service.clock.*</import>
    <import>jadex.bridge.ComponentIdentifier</import>
    <import>jadex.planlib.*</import>
    <import>java.util.logging.*</import>
</imports>
```

```
<capabilities>
    <capability name="procap" file="jadex.bdi.planlib.protocols.request.Request"/>
    <capability name="dfcap" file="jadex.bdi.planlib.df.DF"/>
</capabilities>

<beliefs>
    <belief name="welcomegui" class="jadex.bdi.planlib.GuiCreator"></belief>
    <beliefset name="userNamePasswordType" class="String"></beliefset>
    <belief name="product" class="String"></belief>
    <belief name="relationalOperator" class="String"></belief>
    <belief name="conjunctionOperator" class="String"></belief>
    <belief name="attributeImportanceLevel" class="String"></belief>
    <belief name="attvalueImportanceLevel" class="String"></belief>
    <belief name="selectedProduct" class="String"></belief>
    <belief name="productAttribute" class="String"></belief>
    <belief name="auctionAttribute" class="String"></belief>
    <belief name="auctionAttributeType" class="String"></belief>
    <belief name="auctionAttributeRType" class="String"></belief>
    <belief name="auctionAttributeValue" class="String"></belief>
    <belief name="auctionOrderedAttribute" class="String"></belief>
    <belief name="auctionOrderedAttributeString" class="String"></belief>
    <belief name="auctionAttributeImportanceLevel" class="String"></belief>
    <belief name="bidAttributeValue" class="String"></belief>
    <belief name="nonCondConst" class="String"></belief>
    <belief name="condConst" class="String"></belief>
```

```
<belief name="sellerName" class="String"></belief>
<belief name="auctionAttNSvalue" class="String"></belief>
<belief name="auctionAttNSvalueLikeliness" class="String"></belief>
<belief name="auctionAttCSvalue" class="String"></belief>
<belief name="auctionAttCSvalueLikeliness" class="String"></belief>
<belief name="auctionAttNNvalueLikeliness" class="String"></belief>
<belief name="auctionAttCNvalueLikeliness" class="String"></belief>
<belief name="allBids" class="String"></belief>
<belief name="auctionValidBid" class="String"></belief>
<belief name="bidAllDetail" class="String"></belief>
<belief name="numberCP" class="Integer">
    <fact>0</fact>
</belief>
<belief name="numberNP" class="Integer">
    <fact>0</fact>
</belief>
<belief name="validStart" class="Boolean">
    <fact>false</fact>
</belief>
<belief name="validUser" class="Boolean">
    <fact>false</fact>
</belief>
<belief name="productSelection" class="Boolean">
    <fact>false</fact>
</belief>
<belief name="attributeSelection" class="Boolean">
    <fact>false</fact>
```

```
</belief>
<belief name=”attributePreference” class=”Boolean”>
    <fact>false</fact>
</belief>
    <belief name=”attributePreferenceDone” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”attributeOrdering” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”attributeOrderingString” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”positionDone” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”positionDoneString” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”nonCondConstDone” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”nonCondConstInsertDone” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”condConstDone” class=”Boolean”>
    <fact>false</fact>
</belief>
<belief name=”condConstInsertDone” class=”Boolean”>
    <fact>false</fact>
</belief>
```

```
<belief name=”attNSvaluePreference” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attNSvaluePreferenceDone” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attCSvaluePreference” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attCSvaluePreferenceDone” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attNNvaluePreference” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attNNvaluePreferenceDone” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attCNvaluePreference” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”attCNvaluePreferenceDone” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”bidSubmissionDone” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”bidInsertDone” class=”Boolean”>
   <fact>false</fact>
</belief>
<belief name=”constraintCheck” class=”Boolean”>
```

```
            <fact>false</fact>
        </belief>
        <belief name="constraintCheckDone" class="Boolean">
            <fact>false</fact>
        </belief>
        <belief name="mautStarFlag" class="Boolean">
            <fact>false</fact>
        </belief>
        <belief name="mautStarFlagDone" class="Boolean">
            <fact>false</fact>
        </belief>
    </beliefs>

    <goals>
        <achievegoalref name="rp_initiate">
            <concrete ref="procap.rp_initiate"/>
        </achievegoalref>
        <achievegoalref name="df_search">
            <concrete ref="dfcap.df_search"/>
        </achievegoalref>
    </goals>

    <plans>
        <plan name="validUserReq">
            <body class="validUserReqPlan"></body>
            <trigger>
                <condition language="jcl">$beliefbase.validStart</condition
                    >
            </trigger>
        </plan>
        <plan name="proAttrReq">
```

```
    <body class=”proAttrReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.productSelection</condition>
    </trigger>
</plan>
<plan name=”aucAttrReq”>
    <body class=”aucAttrReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.attributeSelection</condition>
    </trigger>
</plan>
<plan name=”aucAttrPrefReq”>
    <body class=”aucAttrPrefReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.attributePreference</condition>
    </trigger>
</plan>
<plan name=”aucAttrOrderReq”>
    <body class=”aucAttrOrderReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.attributeOrdering</condition>
    </trigger>
</plan>
<plan name=”aucAttrStrValOrderReq”>
    <body class=”aucAttrStrValOrderReqPlan”></body>
    <trigger>
```

```
            <condition language="jcl">$beliefbase.attributeOrderingString</condition>
        </trigger>
    </plan>
    <plan name="nonCondConstReq">
        <body class="nonCondConstReqPlan"></body>
        <trigger>
            <condition language="jcl">$beliefbase.nonCondConstDone</condition>
        </trigger>
    </plan>
    <plan name="condConstReq">
        <body class="condConstReqPlan"></body>
        <trigger>
            <condition language="jcl">$beliefbase.condConstDone</condition>
        </trigger>
    </plan>
    <plan name="nonCondStrPrefReq">
        <body class="nonCondStrPrefReqPlan"></body>
        <trigger>
            <condition language="jcl">$beliefbase.attNSvaluePreference</condition>
        </trigger>
    </plan>
    <plan name="condStrPrefReq">
        <body class="condStrPrefReqPlan"></body>
        <trigger>
            <condition language="jcl">$beliefbase.attCSvaluePreference</condition>
        </trigger>
```

```
</plan>
<plan name=”nonCondNumPrefReq”>
    <body class=”nonCondNumPrefReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.attNNvaluePreference</condition>
    </trigger>
</plan>
<plan name=”condNumPrefReq”>
    <body class=”condNumPrefReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.attCNvaluePreference</condition>
    </trigger>
</plan>
<plan name=”bidSubmissionReq”>
    <body class=”bidSubmissionReqPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.bidSubmissionDone</condition>
    </trigger>
</plan>
<plan name=”constChkReqTri”>
    <body class=”constChkReqTriPlan”></body>
    <trigger>
        <condition language=”jcl”>$beliefbase.constraintCheck</condition>
    </trigger>
</plan>
<plan name=”bidUtilityReqTri”>
    <body class=”bidUtilityReqTriPlan”></body>
```

```
        <trigger>
            <condition language="jcl">$beliefbase.mautStarFlag</
                condition>
        </trigger>
    </plan>
</plans>

<properties>
    <property name="logging.level">Level.INFO</property>
     <property name="logging.useParentHandlers">false</property>
</properties>

<configurations>
    <configuration name="default">
        <beliefs>
            <initialbelief ref="welcomegui">
                <fact>
                    new jadex.bdi.planlib.GuiCreator(WelcomeGUI.class,
                    new Class[]{jadex.bdi.runtime.IBDIExternalAccess.
                        class},
                    new Object[]{$scope.getExternalAccess()})
                </fact>
            </initialbelief>
        </beliefs>
    </configuration>
</configurations>

</agent>
```

Listing A.3: Winner Determination Agent

```
<agent xmlns="http://jadex.sourceforge.net/jadex"
```

```
        xmlns:xsi="http://www.w3.org/2001/XMLSchema instance"
        xsi:schemaLocation="http://jadex.sourceforge.net/jadex
                            http://jadex.sourceforge.net/jadex bdi 2.0.xsd"
        name="WDAgent"
        package="WD">

<imports>
    <import>java.util.*</import>
    <import>java.util.logging.*</import>
    <import>jadex.commons.*</import>
    <import>jadex.rules.rulesystem.rules.functions.*</import>
    <import>jadex.base.fipa.SFipa</import>
    <import>jadex.base.fipa.*</import>
    <import>jadex.planlib.*</import>
    <import>jadex.bridge.service.*</import>
    <import>jadex.commons.future.*</import>
</imports>

<capabilities>
    <capability name="procap" file="jadex.bdi.planlib.protocols.
        request.Request"/>
    <capability name="dfcap" file="jadex.bdi.planlib.df.DF"/>
</capabilities>

<goals>
    <maintaingoalref name="df_keep_registered">
        <concrete ref="dfcap.df_keep_registered"/>
    </maintaingoalref>
    <achievegoalref name="rp_initiate">
        <concrete ref="procap.rp_initiate"/>
    </achievegoalref>
```

```
    <achievegoalref name="df_search">
        <concrete ref="dfcap.df_search"/>
    </achievegoalref>
</goals>

<plans>
     <plan name="constChkReq">
        <body class= "constChkReqPlan"/>
        <trigger>
            <messageevent ref="request_reqconstcheck"></messageevent>
        </trigger>
     </plan>
     <plan name="bidUtilityReq">
        <body class= "bidUtilityReqPlan"/>
        <trigger>
            <messageevent ref="request_reqmautstar"></messageevent>
        </trigger>
     </plan>
</plans>

<events>
    <messageevent name="request_reqconstcheck" direction="receive"
         type="fipa">
        <parameter name="performative" class="String" direction="fixed
             ">
            <value>SFipa.REQUEST</value>
        </parameter>
        <match>$content instanceof String &amp;&amp; ((String)$content
             ).startsWith("reqconstcheck")</match>
    </messageevent>
```

```
    <messageevent name="request_reqmautstar" direction="receive" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.REQUEST</value>
        </parameter>
        <match>$content instanceof String &amp;&amp; ((String)$content).startsWith("reqmautstar")</match>
    </messageevent>
    <messageevent name="inform" direction="send" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.INFORM</value>
        </parameter>
    </messageevent>
    <messageevent name="failure" direction="send" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.FAILURE</value>
        </parameter>
    </messageevent>
</events>

<properties>
    <property name="logging.level">Level.INFO</property>
    <property name="logging.useParentHandlers">false</property>
    <property name="dfservice" class="IFuture">SServiceProvider.getService($scope.getServiceProvider(), IDF.class, RequiredServiceInfo.SCOPE_PLATFORM)</property>
</properties>

```

```
<configurations>
    <configuration name="default">
        <goals>
            <initialgoal ref="df_keep_registered">
                <parameter ref="description">
                    <value>
                        $properties.dfservice.createDFComponentDescription
                            (null,
                        $properties.dfservice.createDFServiceDescription("
                            service_wd", "wd demo", "wd agent"))
                    </value>
                </parameter>
                <parameter ref="leasetime">
                    <value>550000</value>
                </parameter>
            </initialgoal>
        </goals>
    </configuration>
</configurations>

</agent>
```

Listing A.4: MARA Application Descriptor

```
<applicationtype xmlns="http://jadex.sourceforge.net/jadex"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema instance"
    xsi:schemaLocation="http://jadex.sourceforge.net/jadex
                        http://jadex.sourceforge.net/jadex
                            application 2.0.xsd"
    name="MARAAgents">

<imports>
```

```xml
    <import>Admin.*</import>
    <import>GUI.*</import>
    <import>WD.*</import>
</imports>

<componenttypes>
    <componenttype filename="AdminAgent.agent.xml" name="admin"/>
    <componenttype filename="GUIAgent.agent.xml" name="gui"/>
    <componenttype filename="WDAgent.agent.xml" name="wd"/>
</componenttypes>

<configurations>
    <configuration name="One Admin One GUI One WD">
        <components>
            <component type="admin"></component>
            <component type="gui"></component>
            <component type="wd"></component>
        </components>
    </configuration>
</configurations>

</applicationtype>
```