A Generic Agent-Mediated Electronic Commerce Framework for Multi-Attribute Negotiation and Trading

by

Gaurav Tewari

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degrees of Bachelor of Science in Computer Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology May 11, 2001

Copyright 2001 Massachusetts Institute of Technology. All rights reserved.
A Generic Agent-Mediated Electronic Commerce Framework for Multi-Attribute Negotiation and Trading

by

Gaurav Tewari

Submitted to the
Department of Electrical Engineering and Computer Science

May 11, 2001

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Current electronic marketplaces emphasize the singular importance of price in matching buyers and sellers. Unable to adequately express their underlying preferences for all relevant attributes, buyers and sellers are forced to partake in an inflexible and antagonistic price-centric interplay. The result is an unfortunate, zero-sum circumstance of adversarial competitiveness, in which one party can benefit only at the expense of the other. “Multi-Attribute Resource Intermediaries” (MARI) is a project which proposes to improve online marketplaces. MARI is a domain-independent, generic agent-based architecture for electronic markets, which can be instantiated in different product domains. MARI represents an advance over current e-markets in the following ways: (i). It permits buyers and sellers to express multi-attribute preferences; (ii). It is multi-tiered, allowing buyers and sellers to express preferences not only for the attributes of the underlying product or service, but also for each other; and (iii). It facilitates significant automation, allowing software agents to act as proxies for buyers and sellers. MARI transcends bid and ask prices to include all relevant attributes as dimensions for consideration and differentiation, making it possible for both buyers and sellers alike to more holistically and comprehensively specify relative preferences, and making price just one of a multitude of possible factors influencing the choice of trading partner and the decision to trade. Buyers and sellers express their preferences, which permits agents to represent them and their “utility functions.” The system then computes matches between buyer and seller agents in a Pareto optimal fashion, meaning that no one can be made better off without making someone else worse off, and implying that no potential gains are left unexploited. This thesis presents the theory behind MARI, the generic architecture, and an evaluation through the instantiation of the system in a services marketplace.

Keywords: Electronic Commerce; Software Agents; Intermediaries; Highly Mediated Communications; Product Brokering; Merchant Brokering; Preference Modeling; Decision Support; Negotiation; Utility Theory; Electronic Markets.

Thesis Supervisor: Dr. Pattie Maes
Title: Associate Professor of Media Arts and Sciences, MIT Media Laboratory
This thesis is dedicated to my advisor Pattie, to my parents, Satish and Balvinder, and to my sisters, Geetika and Kanu. For everything I am and will become, I thank you sincerely and with the utmost gratitude.
## CONTENTS

1 INTRODUCTION .......................................................................................................... 16  
1.1 Overview ............................................................................................................. 16  
1.2 Contribution ......................................................................................................... 16  
1.3 Roadmap ............................................................................................................. 17  

2 THE MARI SYSTEM ............................................................................................... 20  
2.1 Introduction .......................................................................................................... 20  
2.2 Motivation ........................................................................................................... 20  
2.3 The MARI System .............................................................................................. 21  
2.4 Example Usage ................................................................................................... 24  
2.5 Application Domain: Service Marketplaces ...................................................... 28  
2.6 The Language Translation Marketplace ............................................................. 28  

3 RELATED WORK .................................................................................................... 30  
3.1 Game Theory and Economic Modeling ............................................................. 30  
3.2 Existing Systems ................................................................................................. 33  

4 ALGORITHMS AND TECHNOLOGIES USED .................................................... 36  
4.1 Modeling User Preferences .............................................................................. 36  
  4.1.1 Overview ...................................................................................................... 36  
  4.1.2 Capturing User Preferences ......................................................................... 36  
  4.1.3 Modeling User Utility Functions .................................................................. 38  
4.2 Matching Buyers and Sellers ............................................................................. 43  
  4.2.1 Overview ...................................................................................................... 43  
  4.2.2 Generating Data Points from Functional Approximations........................... 45  
  4.2.3 Exploring the Attribute Space to identify the Optimal Transaction Configuration (OTC) ............................................................................................................ 48  
  4.2.4 Delineating Transaction Partners to Maximize Aggregate Surplus .......... 49  
  4.2.5 Presenting Results ....................................................................................... 53  

5 THEORETICAL PRINCIPLES AND METHODOLOGICAL RATIONALE .......... 56  
5.1 Modeling Preference Functions ........................................................................ 56  
  5.1.1 Assumptions .................................................................................................. 56  
  5.1.2 Mathematical Formulation of Utility Functions ............................................. 57
11.4 Schematic Depiction of Database Components and Their Interrelationships..... 138

12 APPENDIX B: IMPLEMENTATION BACK END COMPONENTS ................. 142

12.1 Introduction ........................................................................................................ 142

12.2 Back-End Components................................................................................... 144

12.2.1 Login.asp ...................................................................................................... 144

12.2.2 Welcome.asp ........................................................ ........................................ 144

12.2.3 Specific-market.asp ...................................................................................... 145

12.2.4 Create-agent.asp ......................................................................................... 145

12.2.5 Choose-profile.asp ...................................................................................... 146

12.2.6 Start-request.asp ......................................................................................... 147

12.2.7 Market-parameters.asp ................................................................................ 148

12.2.8 Visit 0: ........................................................................................................ 148

12.2.9 Visit 1: ........................................................................................................ 149

12.2.10 Visit 2: ...................................................................................................... 150

12.2.11 Visit 3: ..................................................................................................... 150

12.2.12 Market-watch.asp ..................................................................................... 150

12.2.13 Market-history.asp .................................................................................... 151

12.2.14 Process-request.asp ................................................................................... 151

12.2.15 Matching.asp ............................................................................................ 152

12.2.16 Logout.asp ................................................................................................. 154

12.2.17 Faq.asp .................................................................................................... 154

12.2.18 About.asp ................................................................................................ 155

12.2.19 Contact-info.asp ....................................................................................... 155

13 APPENDIX C: SUGGESTIONS FOR FURTHER READING ABOUT MARI...... 156

14 BIBLIOGRAPHY ........................................................................................................ 158

14.1 Works Referenced ............................................................................................ 158

14.2 Works Consulted ............................................................................................... 161
List of Figures

Figure 1: Agents Manifest Their Owners' Preferences, and are Optimally Matched by the System ................................................................. 24
Figure 2: Bids and Asks Corresponding to User Preferences .................................................. 25
Figure 3: Value Created in a Traditional Market ................................................................. 27
Figure 4: Value Created by MARI ................................................................................ 27
Figure 5: Specifying the Ideal Offer and Permissible Ranges for Flexible Attributes ..... 38
Figure 6: The Market Maker Visually Associates Utility Functions with Flexible Attributes .................................................................................. 39
Figure 7: Details of the "deal" designated by the system for a matched agent ................ 54
Figure 8: Details of potential deals with other qualified transaction partners that were considered, with corresponding scores, and whether these prospective transaction partners were matched with someone else ............................................................. 54
Figure 9: Architecture Design Schematic ........................................................................... 70
Figure 10: Flowchart Schematically Illustrating Major MARI Front-End Modules ...... 77
Figure 11: Flowchart Illustrating Major MARI Back-End Modules .................................. 78
Figure 12: Choosing a Buyer Profile .................................................................................. 80
Figure 13: Specifying Permissible Ranges for Flexible Attributes ..................................... 81
Figure 14: Selecting Utility Functions ................................................................................ 82
Figure 15: Editing a Selling Agent .................................................................................... 85
Figure 16: Viewing Matched Agents .................................................................................. 88
Figure 17: Viewing Details of the "Deal" the System has Selected for a Matched Agent 88
Figure 18: Viewing Alternatives that Were Taken Into Account for a Matched Agent ..... 89
Figure 19: View Market History ........................................................................................ 92
Figure 20: Theoretically Calculated Surplus Values Corresponding to each Potential Match .................................................................................. 100
Figure 21: Actual System Generated Surplus Values Corresponding to each Potential Match ............................................................................... 100
Figure 22: S2's owner sees details of the "deal" associated with his matched agent .... 102
Figure 23: S2’s owner sees details of potential deals with other qualified transaction partners that were considered and whether these prospective transaction partners were matched with someone else ............................................................. 102
Figure 24: Details Seen by B1’s Owner ............................................................................. 103
Figure 25: System Generated Surplus Values Corresponding to each Potential Match 105
Figure 26: Value Created in a Traditional Market ............................................................. 106
Figure 27: Value Created by MARI .................................................................................. 107
Figure 28: System Generated Surplus Values Corresponding to each Potential Match 109
Figure 29: Details Seen by S1’s Owner ............................................................................. 110
Figure 30: S1’s Owner Sees Details of Potential Deals with other Qualified Transaction Partners ............................................................................... 110
Figure 31: System Generated Surplus Values Corresponding to each Potential Match 112
Figure 32: Details Seen by B1’s Owner ............................................................................. 113
Figure 33: B1’s Owner Sees Details of Potential Deals with other Qualified Transaction Partners ............................................................................... 113
Figure 34: Table ‘Players’ and its Interaction with Other Tables ........................................ 129
Figure 35: Table ‘Markets’ and its Interaction with Other Tables.......................... 130
Figure 36: Table ‘MarketToXToAttributeMap’ and its Interaction with Other Tables.. 131
Figure 37: Interaction Schema between Major MARI Database Components........... 139
Figure 38: Flowchart Illustrating Major Back End Components and Programmatic
           Control Flow ........................................................................................................... 143
List of Tables

Table 1: MARI's Predefined Generic Utility Functions (x is a place-holder variable for a value of the flexible attribute over its permissible range) ........................................... 41
Table 2: Profile of a "Professional Buyer" BPRO ......................................................... 46
Table 3: Profile of a "Professional Seller" SPRO ........................................................ 48
Table 4: Utility Functions Represented by Degree One (Linear) Polynomials ................. 59
Table 5: Utility Functions Represented by Degree Two (Quadratic) Polynomials ............. 61
Table 6: Other Utility Functions ................................................................................. 61
Table 7: Scenario 1, Buyer 1 (B1) ............................................................................. 99
Table 8: Scenario 1, Buyer 2 (B2) ............................................................................. 99
Table 9: Scenario 1, Buyer 2 (B2) ............................................................................. 99
Table 10: Scenario 1, Seller 1 (S1) .......................................................................... 99
Table 11: Scenario 1, Seller 2 (S2) .......................................................................... 99
Table 12: System Generated Closing Prices Corresponding to Each Potential Pairing ....... 101
Table 13: Scenario 2, Buyer 1 (Professional Buyer) .................................................... 104
Table 14: Scenario 2, Buyer 2 (Casual Buyer) .......................................................... 104
Table 15: Scenario 2, Seller 1 (Professional Seller) .................................................... 104
Table 16: Scenario 2, Seller 2 (Casual Seller) .......................................................... 104
Table 17: System Generated Closing Prices Corresponding to Each Potential Pairing ....... 105
Table 18: Scenario 3, Seller 1 (Monopolist) ............................................................. 107
Table 19: Scenario 3, Buyer 1 .................................................................................. 108
Table 20: Scenario 3, Buyer 2 .................................................................................. 108
Table 21: Scenario 3, Buyer 3 .................................................................................. 108
Table 22: Details of Potential Matches for S1 Considered by the System ...................... 109
Table 23: Scenario 4, Buyer 1 .................................................................................. 111
Table 24: Scenario 4, Seller 1 .................................................................................. 111
Table 25: Scenario 4, Seller 2 .................................................................................. 111
Table 26: Scenario 4, Seller 3 .................................................................................. 112
Table 27: Details of Potential Matches for B1 Considered by the System ...................... 113
Table 28: Interface Experiment Results .................................................................... 119
Table 29: System Experiment Results ...................................................................... 121
List of Equations

Equation 1 ................................................................. 40
Equation 2 ................................................................. 42
Equation 3 ................................................................. 47
Equation 4 ................................................................. 48
Equation 5 ................................................................. 50
Equation 6 ................................................................. 50
Equation 7 ................................................................. 51
Equation 8 ................................................................. 51
Equation 9 ................................................................. 51
Equation 10 ............................................................... 51
Equation 11 ............................................................... 66
Equation 12 ............................................................... 66
Equation 13 ............................................................... 66
Equation 14 ............................................................... 67
Equation 15 ............................................................... 67
ACKNOWLEDGEMENTS

This thesis could not have been accomplished without the assistance, support and encouragement of a number of people and institutions.

I would like to recognize the invaluable assistance provided by undergraduate students Alexander Berkovich, Vladislav Gabovich, Shuehan Liang and Anand Ramakrishnan in helping to implement this system and in making the project a success.

Additionally, I want to convey my sincere gratitude and appreciation to Professor Pattie Maes, for having given me the opportunity to work directly with her and for allowing me to have benefited from her advice and guidance throughout my tenure at the Media Lab. Pattie has had a profound impact on me personally as well as professionally and, going forward, I sincerely hope that I will continue to have the good fortune to interact with her.

I would like to thank Professors Jeremie Gallien, Dan Ariely, Chris Dellarocas, Benjamin Grosof, Hal Varian and David Autor and Mr. Michael Schrage for providing useful and timely comments, suggestions and guidance. Their discussions with me helped to substantially improve the quality and contribution of this thesis.

I would like to recognize the members of the Software Agents Group at the MIT Media Laboratory, as well as the Media Lab community at large, for their objective, yet supportive, critiques of my work and valuable feedback. I would like to thank MIT, and specifically the MIT Media Laboratory for use of equipment, laboratory space and, in general, for providing, an atmosphere conducive to doing research.

Words are not enough to express my appreciation and respect for my sisters, Geetika and Kanu, my father, and my mother for their unconditional love, support and patience. Without their constant blessings, prayers, and encouragement, I could not have accomplished anything.

Finally, I recognize and feel blessed by the influence upon my life of that omniscient, omnipresent and omnipotent higher power we call God. Mata Gayatri and Sri Krishna, I am constantly cognizant of your manifest, divine protection over me at every moment, everywhere.
1 INTRODUCTION

1.1 Overview

This thesis describes the design, implementation and evaluation of “Multi-Attribute Resource Intermediaries” (MARI) – a research project conducted in the Software Agents Group at the MIT Media Laboratory. MARI is an agent-based architecture intended as a generalized platform for the specification and brokering of heterogeneous goods and services. MARI proposes to radically improve online marketplaces, specifically those that involve the buying and selling of non-tangible goods and services. Our work has focused on developing a general, domain-independent infrastructure that will make it easier for human users to customize the behavior of software agents (self-contained, executing segments of code designed to perform particular tasks), that represent them in the marketplace. Our system makes it possible for users to quantify their preferences for goods and services along multiple, heterogeneous attributes or dimensions rather than just on the basis of price, as is currently the case in state of the art systems. This, in turn, makes it possible for participants in online marketplaces to be represented by software agents who find suitable products and trading partners on behalf of their owners, autonomously generate valuations for the product in question, and ultimately negotiate the terms of the transaction.

MARI represents an advance over state of the art electronic marketplaces in the following ways: (i). It permits buyers and sellers to express multi-attribute preferences; (ii). It is multi-tiered, allowing buyers and sellers to express preferences not only for the attributes for the underlying product or service, but also for each other; and (iii). It facilitates significant automation, allowing agents to act as proxies for buyers and sellers.

1.2 Contribution

MARI attempts to overcome many of the deficiencies of contemporary electronic markets and represents a fundamentally novel paradigm for gathering and modeling user preferences, and for matching buyers and sellers. It provides automation and facilitates
multi-attribute considerations that help transcend price as the only negotiative dimension. Ultimately, MARI provides a framework for value creation, in a strategic sense, by allowing buyers and sellers to sufficiently differentiate their preferences and offerings from other players in the market, and thereby establish a competitive advantage. The unique contributions of this thesis include:

1. Formulation and implementation of the theory of multi-attribute matching;
2. Design of a general architecture that includes:
   a. A mechanism for expressing and modeling utility functions;
   b. Matching algorithms;
   c. Dynamic, domain-independent marketplace creation. The design of the system is sufficiently flexible that creating an entirely new market merely entails editing two XML files.
3. Evaluation of the system via simulations of market scenarios and user testing.

1.3 Roadmap

Chapter 2 provides a functional overview of the application domain and example usage of MARI, and delivers a high-level view of the contribution, importance and relevance of the system. Chapter 3 presents related work in transaction brokering and user preference-modeling technologies in contemporary electronic markets, along with the limitations of these approaches. This chapter underscores the compelling need for system such as MARI, and situates our work relative to other research efforts.

Chapter 4 details how the MARI system actually works, specifying the algorithms and technologies employed in the process of capturing and modeling user preferences, and matching buyers and sellers. Chapter 5 elucidates the theory and rationale behind our algorithms and technologies, and validates these choices from a theoretical perspective. Chapter 6 discusses the functional design of the MARI system, giving an overview of the major functional modules and top-level user-interaction schema. Chapter 7 describes MARI's front-end Web based interface, the user experience, and major user-driven operations, with the help of flowcharts and screenshots.
Chapters 8 and 9 discuss how MARI has been evaluated via simulations of market scenarios and via actual user-testing with human participants. Our simulations effectively illustrate the range of MARI’s functionality in actual market settings and allow the reader to gain further insight into how participants in electronic markets may benefit from such a system. Bringing the human element into the picture, findings from our user studies shed light on the extent to which users feel satisfied with the system, and provide important cues for future improvements.

Chapter 10 concludes the thesis and discusses potential future research directions. Appendices A and B detail the actual implementation of MARI’s data-model, and back-end algorithms, respectively. In addition to presenting implementation-level flowcharts and module-dependency diagrams, these chapters discuss the rationale behind implementation decisions and elucidate tradeoffs. Appendix C provides suggestions for further reading about MARI, and cites some of the most important publications related to the project.
2 THE MARI SYSTEM

2.1 Introduction

MARI embodies a trend, expected to be key to the electronic marketplaces of tomorrow. Specifically, we believe that negotiations will be highly complex and participants will engage in negotiation over various aspects of a transaction, price being only one of many considerations.

MARI represents a general purpose architecture that is capable of supporting multiple sellers and buyers within multiple product domains. To evaluate the project, we deployed the MARI infrastructure in the context of a “services marketplace” in which language translation services are bought and sold. Hence, MARI is specifically encoded with a “language translation service” ontology and suitable complementary data.

2.2 Motivation

Contemporary electronic markets are deficient for two major reasons: (a). They accentuate the importance of price in matching buyers and sellers; and (b). They severely lack any form of automation, often mandating a significant time commitment on the part of the human participant.

State of the art online marketplaces, such as Chemdex [2], Priceline [35], Elance [11], etc., do not provide any means for market participants to be able to transcend the singular importance of price and convey the full “value proposition” of the holistic product offering. When making buying decisions, we all make tradeoffs across product features (additional RAM in lieu of a faster processor when purchasing a computer, for instance) and also associate different levels of importance with different features (number of stopovers being much more important than the brand of the airline when buying an air ticket, for instance). Unfortunately, current electronic markets provide no mechanism by which market participants may express such aspects of their preferences. Further accentuating this problem, online auction systems additionally have a tendency to foster a spirit of adversarial competitiveness in the buying process. In systems such as Ebay [10]
and Amazon Auctions [2], not only must a buyer first undergo the burden of uniquely identifying the exact product she is seeking, but furthermore, she must then enter into an inflexible and antagonistic bidding interplay with the seller.

Contemporary electronic markets also suffer from a severe lack of automation. While some level of automation has begun to be introduced, such as “bidding agents” in E-bay [10], which can be programmed to automatically increment bids up to a user defined threshold at pre-specified increments, as e-commerce becomes more pervasive, much more automation will be needed in both generating bids and asks on behalf of market participants, as well as facilitating the actual matching between buyers and sellers.

In general, we have found that there is no existing satisfying system that allows buyers and sellers to express their underlying multi-attribute preferences in an intuitive and understandable fashion, and which uses these preferences to facilitate automatic, yet non-binding, matching of transaction partners. Our work represents a step in the direction of filling this void.

2.3 The MARI System

MARI is a general market architecture that can be instantiated in different domains. MARI attempts to overcome many of the deficiencies of contemporary electronic markets and represents a novel paradigm for gathering user preferences, and for matching buyers and sellers.

MARI allows market participants to express their underlying preferences with greater expressiveness. Market participants specify what they are ideally looking for, and associate a price with this ideal configuration. Additionally, market participants are able to specify permissible ranges for feature values they are willing to be flexible on and, furthermore, they are able to visually, in an intuitive fashion, associate preference functions with attributes. Based upon this input, MARI asks the user a small number of simple questions, that allow the system to generate an accurate mathematical model of the user’s underlying preferences.

MARI facilitates automation via the instantiation of buyer and seller agents that act as manifestations of their owners’ revealed preferences. These agents autonomously
generate bids and asks on behalf of their owners, which the system uses to “optimally” match transaction partners. Figure 1 schematically illustrates this notion. Our current heuristic of optimality is that aggregate “surplus,” which is currently taken to be the cumulative bid-ask spread across all buyer-seller pairs, and which can be shown to be equivalent to the economic notion of aggregate “welfare,” [15, 38] is maximized. In practice, the choice of optimization heuristic is configurable by the “market maker.” The final matching produced by our system can be shown to be Pareto efficient [15, 38], meaning that no one can be made better off without making someone else worse off. A strong condition, Pareto efficiency guarantees that our system does not leave any potential gains unexploited.

MARI facilitates matching of buyers and sellers on multiple attributes, rather than just price. MARI is unique in the sense that it allows both the buyer as well as the seller to mutually exercise control and, furthermore, allows market participants to express preferences for the attributes of the transaction partner (for instance, a buyer can express preferences for desired seller reputation or skill), in addition to the attributes of the underlying product. Our matching algorithm balances the myopic, selfish interests of individualistic agents at a local level with the desire to achieve “welfare” maximization and Pareto efficiency at a global level. Our model of user preferences allows us to predict how much a given user is willing to bid (or ask) for a given transaction partner, as well as to gauge how “far” the transaction partner lies from the user’s desired ideal offer configuration.

For each user we look at each qualified transaction partner to see whether a deal is possible and, if so, what the “Optimal Transaction Configuration” (OTC) would be. Exploring the space of possible deals is accomplished by exploring the space defined by flexible attribute ranges, as specified by the buyer and seller, and is equivalent to automated negotiation over the underlying attribute values. We explore the attribute space to find a deal configuration that is mutually agreeable and beneficial from both the buyer’s and seller’s perspective. The OTC is simply a particular configuration of attributes, or a “deal,” that is consistent with the permissible attribute ranges that both the buyer and the seller are willing to tolerate, and is the “deal” which lies “closest” to both the buyer’s and seller’s ideal offers. The notion of the OTC captures the “deal”
configuration that is myopically optimal in the context of a specific buyer-seller pairing, from the self-interested perspectives of the buyer and seller agents involved in this pairing. Calculating the surplus corresponding to the OTC allows us to gauge how the “goodness” (in terms of aggregate welfare or surplus) of this pairing might compare with that of other pairings. Since, for any given user, an OTC will be computed for every qualified transaction partner, comparing the relative “goodness” of the various OTCs gives us a global heuristic of surplus maximization by which we may decide which of the various qualified transaction partners the user ought to be ultimately paired with.

Even after having delineated tentative transaction partners, MARI gives market participants flexibility in choosing to confirm or reject the proposed match. A proposed match between a buyer and seller can only be consummated by explicit mutual confirmation from both parties involved. Moreover, without mutual confirmation, buyer and seller agents continue to remain “active,” or eligible to be matched in future market cycles, thereby maximizing the potential set of suitable alternatives.

Ultimately, MARI provides a framework by which buyers and sellers may differentiate themselves to create value. The system facilitates value creation, in a strategic sense, by allowing buyers and sellers to sufficiently differentiate their preferences and offerings from other players in the market, and thereby establish a competitive advantage.

MARI is intended to be instantiated by a “market maker” within a specific product domain. The design of the system is sufficiently flexible that creating an entirely new market merely entails editing two XML files. Within the context of any specific product domain, the market maker can use these XML files to specify relevant domain specific knowledge, such as the domain ontology (vocabulary used to describe product features), and user “profiles” (stereotype driven “profiles” of default user preferences). We assume that the market maker must independently deal with implementation specific issues relating to things such as security, scalability, fulfillment guarantees, etc.
2.4 Example Usage

To illustrate the functionality of the system, we now turn towards a simplified example of a language translation marketplace in which there are two types of buyers and two types of sellers. The buyers and sellers are differentiated in the following ways:

1. **Professional Buyer** ($B_{PRO}$): Wants a fast translation, is sensitive to seller reputation and skill. Possesses a high reputation rating.

2. **Casual Buyer** ($B_{CAS}$): Is not very sensitive to time, seller reputation or skill. Possesses a low reputation rating.

3. **Professional Seller** ($S_{PRO}$): Does not mind completing a translation job under time pressure, is sensitive to buyer reputation. Possesses a high reputation and skill rating.

---

1 This results presented in this example are concretely substantiated via a market simulation, in Chapter 8.
4. Casual Seller ($S_{CAS}$): Faces a large opportunity cost if asked to complete a translation under time pressure, is not very sensitive to buyer reputation. Possesses a lot reputation and skill rating.

MARI makes it feasible for these buyers and sellers to express their differential preferences for prospective transaction partners. Figure 2 shows system generated\(^2\) bids and asks corresponding to these preferences.

From Figure 2 we see that the Professional Buyer is willing to pay less for the Casual Translator than for the Professional Translator, since the Casual Translator possesses a lower reputation and skill rating. The Casual Buyer on the other hand does not care for the higher skill and reputation ratings of the Professional Translator has, and is hence unwilling to pay any premium over what he is willing to pay for the Casual Translator.

\(^2\) Numerical values used are based on actual system generated data. See Chapter 8 for precise details of an analogous simulation that we conducted.
The Professional Translator is willing to charge less to the Professional Buyer than the Casual Buyer, as a consequence of the former’s higher reputation rating. The Casual Translator, on the other hand, actually wants to charge more to the Professional Buyer than the Casual Buyer, since the former mandates a faster translation, which imposes a higher opportunity cost on the Casual Translator.3

Based upon these preferences, MARI matches $B_{PRO}$ with $S_{PRO}$ at a clearing price of $90$ (midway in between the $100$ bid and $80$ ask), and matches $B_{CAS}$ with $S_{CAS}$ at a clearing price of $45$. The aggregate surplus, which in this case corresponds to the total value created in the marketplace, is maximized and the pairing is Pareto efficient.

In traditional markets, there is no mechanism by which a professional buyer ($B_{PRO}$), for instance, can instantiate an agent with variable degrees of preference and valuation for different (professional and casual) sellers. The buyer can merely assume that there is some mixture of the two types of sellers in the market and bid something in between his differentiated bids corresponding to different sellers. For the sake of simplicity lets assume that in a price-matched marketplace $B_{PRO}$ would bids $85$ (halfway in between the differentiated bids). Using the same reasoning, $B_{CAS}$ would bid $50$, $S_{PRO}$ would ask $90$ and $S_{CAS}$ would ask $60$. In this case, $B_{PRO}$ will be undesirably matched with $S_{CAS}$.

Figure 3 and Figure 4 illustrate how MARI facilitates greater value creation, and more appropriate allocation of scarce resources compared to traditional marketplaces.

---

3 Chapter 8 shows in greater detail how these buyers and sellers would express these valuations.
Figure 3: Value Created in a Traditional Market

Total Value Created = $25, Inappropriate Allocation of Scarce Resources

Figure 4: Value Created by MARI

Total Value Created = $30, Appropriate Allocation of Scarce Resources
2.5 Application Domain: Service Marketplaces

A substantial motivation for the choice of “services marketplaces” in general, and a language-translation marketplace in particular, as the application domain for this project lies in our belief [33] that in the future, as people become increasingly networked, it will be easier for individuals to mutually help one another. People who share each other’s notions of quality, and other such intangible attributes, are in a much better position to be helpful to one another, at least until agents become truly “intelligent,” and can be the ones helping people with complex tasks. For instance, given the extremely rudimentary capabilities of current state of the art automated translation systems [1], a (networked) person would be much better off if she could receive help with a language translation problem from a human expert located somewhere else. In this context, it is reasonable to postulate that the importance of technologies that mediate communications amongst end users will emerge as being of critical importance.

2.6 The Language Translation Marketplace

In order to be able to participate in the MARI marketplace, a “seller” creates a “selling agent” that is aware of its owner’s level of expertise, availability, compensation expectations, and other special constraints, such as requirements for the buyer. Similarly, a “buyer” creates a “buying agent” that understands the exact needs of its owner such as degree of expertise desired, time-sensitivity or urgency with which information is needed, range and type of compensation that the buyer is willing to offer the seller, and other special constraints, such as minimum requirements on the seller’s reputation level. Additionally, the buying and selling agents also encapsulate information on how different qualified sellers or buyers, respectively, can be rank ordered in degree of relative preference. Subsequently, the “market” automatically matches buyers and sellers. Once a match has been made, other media (such as email, cell-phone, as well as richer media) may be employed to implement the “knowledge transfer relationship” in practice.

A benefit of focusing specifically on “services marketplaces” and “information goods” is that it allows us to concentrate more on the attributes relevant to the parties attempting to engage in the transaction without getting overwhelmed by the details
inherent to the product itself. For many information goods, the characteristics of the seller actually serve to define the good itself. For instance, in a language translation marketplace, the fact that the seller of the translation service is considered an expert, and has a high reputation rating associated with her to substantiate this claim, implicitly conveys the nature and quality of the "good," in this case the translation service. Indeed, one can argue that in the context of services marketplaces, in which the service being bought and sold lacks tangible manifestation and is hence not as easily susceptible to objective evaluation, the ability to be able to ontologically segregate and prioritize the various subtle impinging factors gains significance and relevance. Hence, the choice of services marketplaces, in which intangible services and information are bought and sold, is an appropriate and fitting choice as the target application domain for this project.

The electronic services marketplace built with MARI allows us to instantaneously match buyers and sellers in real-time. The resulting transactions capture both the marginal utility that the buyer derives, as well as the marginal cost that the seller incurs, ensuring market efficiency from an economic point of view\textsuperscript{4}. What is remarkable about this infrastructure is that, by using a highly mediated approach, we are able to successfully coordinate buyers and sellers in real-time so as to facilitate ad-hoc interactions. Ultimately, market forces will push the system towards an equilibrium where the time and efforts of a true expert are optimally used for just those translation jobs that cannot easily be dealt with by anyone else. Resources will be bid up or down to reflect their true worth based upon a continuously updating balance between supply and demand of the scarce resource, and will thus be optimally allocated.

\textsuperscript{4} These terms are explained further in Chapter 4.
3 RELATED WORK

MARI broadly relates to: (i). Theoretical work done in the sphere of game theory and economic modeling; and, (ii). Practical and research systems deployed in the context of contemporary electronic markets. We sequentially consider each of these categories.

3.1 Game Theory and Economic Modeling

A substantial amount of work in the realm of game theory and economic modeling has been done to understand and devise methods for agents to interact in various contexts. For our purposes, auction mechanisms, market-oriented programming, coalition-formation techniques and contracting methods represent the most important related areas of relevance. We briefly discuss each of these approaches in relation to the MARI system.

Historically, auctions have been a popular means for brokering transactions. While it is possible to formulate many different kinds of auction mechanisms, the most common patterns of interaction have been one-to-many auction protocols [32], where one agent initiates an auction and several other agents can bid, and many-to-many protocols [46], where several agents initiate an auction and several agents may bid. Auction mechanisms typically differ in the type of protocol used in the auction (ascending price, descending price, etc.) [15], and the bidding strategy adopted by the participating agents. The MAGMA system [23, 22] is an example of an architecture for an agent-based virtual market that relies upon an auction mechanism. Currently, MAGMA uses a Vickery auction mechanism as a negotiation method, but it is capable of incorporating other protocols. In MARI we chose not to adopt an auction mechanism, since one of the explicit design goals of the system has been to circumvent the adversarial competition between buyers and sellers that auctions typically induce. The main difference between auction-based marketplaces and MARI is that the former match based on a single attribute, namely price, while MARI uses multiple-attributes as dimensions for negotiation and brokering. Furthermore, auction mechanisms exhibit a great deal of variability depending upon what protocol (English, Dutch, Vickery etc.) [15, 40] is used.
Hence, using an auction mechanism to create a general, domain-independent system such as MARI, while trying to preserve desirable economic invariants such as Pareto optimality, would not be feasible without imposing significant and specialized restrictions upon participating agents. This would run counter to one of MARI’s design philosophies, that individual agents can be relatively naïve and lightweight, since most processing and computation is performed centrally on the MARI server.

Recently, there has been some work in the area of multi-attribute, combinatorial auctions [42]. Such multi-attribute auction schemes usually mandate that buyers and sellers specify several mutually exclusive “bundles” of what they are looking for. A combinatorial auction is then used to allocate a suitable bundle for each buyer-seller pair. The difference between such schemes and MARI, is that our system does not require that market participants explicitly articulate every single acceptable bundle. Rather, MARI searches over the attribute space and automatically finds an acceptable “deal” or transaction configuration, and derives the associated transaction price.

MARI also relates to work done in the area of market-oriented programming, an approach to distributed computation based on market price mechanisms [4, 5, 24, 25, 43, 50]. Market-oriented programming exploits the institution of markets and models markets to solve problems that fundamentally pertain to distributed resource allocation. Participating agents in such systems typically act in a very restricted manner — by offering to buy or sell quantities of commodities at fixed prices. The system is eventually expected to reach an equilibrium, at which point the market has computed an allocation of resources corresponding to each participating agent [5, 24, 25]. However, such approaches are applicable only when there are several units of each kind of good and when the number of agents is large — conditions which would clearly be restrictive in a system such as MARI [40]. Another issue is that there can be situations in which reaching an equilibrium takes an inordinate amount of time, and the system cannot be guaranteed to converge [24].

Another technique for resource allocation in agent-based markets has been that of coalition formation [30, 44, 45]. The formation of coalitions has been studied both in the context of multi-agent systems and distributed problem solving environments. Typically agents are always built to maximize the overall performance of the system, and design of
the coalition based mechanism typically hinges on how coalitions are formed or structured so as to maximize the overall expected utility of the agents [40]. However, finding the coalition structure that maximizes the overall utility of the system is an NP complete problem [40], which creates computational bottlenecks. Moreover, in a multi-agent environment, such as MARI, the problem of how joint utility should be ultimately divided amongst the participating self-interested agents itself becomes a difficult problem to resolve. In addition to these reasons, we chose not to adopt a coalition-oriented mechanism since deploying it in the context of MARI would impose further constraints, such as increased communication costs and computation time, not to mention the complexity of devising algorithms and metaphors for coalition formation in the first place.

Contracting methods are yet another approach towards inducing cooperation amongst agents in multi-agent systems. Applied to electronic markets, contract based mechanisms typically entail that one self-interested agent try to convince another self-interested agent to help it, in return for some promise of reward [40]. Perhaps the best known framework for automated contracting is the Contract Net protocol [39, 46]. This protocol provides a formalization of the bidding and decision processes based on marginal cost calculations by localized agents. The main question that arises when thinking about a contract metaphor in a setting such as MARI is how one agent might be able to convince another agent to undertake a contract for it when the agents do not share a global task and when the agents are strictly self-interested.

In summary, game theoretic and economics techniques provide useful metaphors for development of transactional agent-based e-commerce systems. However, the choice of a specific technique for a given domain depends on a variety of factors including: whether the agents are self-interested, the number of agents in the environment, the type of agreement, if any, that agents need to reach, and the amount and type of information that agents have about one another [40]. Given our assessment of the most popular game theoretic approaches, we did not feel that a direct application of any of the above-mentioned metaphors would be appropriate for a system such as MARI. Even though we borrow significantly from the field of market-oriented allocation mechanisms, given our
high-level objectives and design philosophy, we have devised a novel and, in many ways, a simpler and less restrictive schema.

3.2 Existing Systems

MARI relates to online negotiation systems and auctions, such as Kasbah [19] and AuctionBot [32], and to commercial systems provided by Moai [26], TradingDynamics [47] and others. It differs from these in its bilateral, multi-attribute methodology for matching of buyers and sellers. Our model, implicitly incorporating negotiation, provides a richer alternative to these systems and successfully circumvents the adversarial competitiveness of online auctions.

Unlike most online shopping systems which generally operate in only one stage of the online shopping process [31], MARI operates in three core stages – namely product brokering, merchant brokering, and (implicitly) negotiation – to provide a unified experience that better facilitates economically efficient and socially desirable transactions. MARI amalgamates features of the ‘Market Maker’ [9] and ‘Tête-à-Tête’ [37] projects at the Media Lab, and extends these to create a more comprehensive solution.

MARI relates to first generation price-comparison systems such as BargainFinder [3] and Jango [17], but goes much further than the rudimentary functionality afforded by such tools. MARI goes beyond just bid and ask prices to include the attributes of the transaction parties as dimensions for consideration and differentiation. MARI relates to second generation value comparison shopping systems such as Personalogic [34], MySimon [27], and the Frictionless ValueShopper [12] in that it integrates ideas from multi-attribute utility theory to meaningfully facilitate the exchange of complex and heterogeneous products. It differs from these systems in that: (i). It is two-way, allowing both the buyer and seller to express preferences; and (ii). It facilitates significant automation, allowing agents to act as proxies for buyers and sellers.

MARI also relates to overtly agent-based systems such as Lost Wax’s “e-Market” platform [21], a software solution targeted specifically at the business-to-business (B2B) e-commerce space. Akin to MARI, systems such as Lost Wax’s “e-Market” espouse the
philosophy that individual intelligent agents ought to automate negotiation and aspects of transactions between e-market participants. Most such systems, and “e-Market” in particular, rely upon some derivation of the standard agent Belief Desire Intention (BDI) architecture to construct and evaluate plans for agents [41]. Subsequently, participating agents are expected to formulate offers, evaluate counteroffers, and make decision about accepting or declining negotiated contracts. MARI differs from such systems in several ways. Most obviously, MARI mitigates the complexity that needs to be engineered in individual agents, and does not mandate that agents be sophisticated enough to be able to evaluate and formulate offers. This is desirable, since in many application domains, particularly in mobile computing, individual agents face severe computational limitations. Further, by delineating deals and transaction partners in a centralized fashion, MARI substantially mitigates the computational time and overhead that is incurred in the process of negotiating aspects of deals and matching transaction partners, while, at the same time, affording individual agents a great deal of flexibility in consummating deals and gaining exposure to potential transaction partners. Finally, by having a centralized entity mediate interactions between participating agents, we are able to significantly reduce the scope of potential security threats and possibly damaging behavior on the part of individual agents, while guaranteeing the existence of desirable global optimality metrics, such as welfare maximization.
4 ALGORITHMS AND TECHNOLOGIES USED

4.1 Modeling User Preferences

4.1.1 Overview

When a user (buyer or seller) initially logs onto MARI, he or she must specify whether her intent is to put a product or service up for sale or to purchase a product or service. Our intent is to gauge the user's multi-attribute utility function so as to be able to accurately assess how the user would value products that have not been explicitly seen or "rated" before. Being able to make such inferences is integral to the successful functioning of MARI's core matching algorithm.

MARI's market structure most closely resembles a "monopolistic competition" [38] – each seller has the ability to differentiate her products or services from those of other sellers. The market structure is monopolistic in the sense that each seller has the ability to set her own price, rather than merely accept the prevalent "market price" as under perfect competition and, thus, can be said to exercise market power. On the other hand, each seller must still compete, in terms of price and the range of products offered, with other sellers since they are all effectively trying to find transaction partners from a common underlying set of buyers. Moreover, there are no barriers to entry, and new sellers are free to enter the market. In this way, the market structure also resembles that of a competitive industry.

4.1.2 Capturing User Preferences

Each distinct buyer or seller is represented within MARI by an agent. The "buyer agent" embodies the buyer's revealed preferences with respect to the desired resource.

---

5 We use the term "user" to refer specifically to a buyer or seller. By contrast, we use the term "market maker" to refer to the system administrator who instantiates MARI within the context of a specific product domain.

6 Since we specifically focus on complex products and services that consists of multiple, often non-tangible, attributes (such as seller reputation, for example), one could argue that the merchant offerings are differentiated a priori. Indeed, one can reasonably argue that in such product domains, it is extremely difficult, if not simply infeasible, for a given seller to perfectly replicate another seller's product offering.
Similarly, “seller agents” embody the preferences and interests of sellers. Each agent is customized to the needs and desires of its owner, and attempts to advocate on the owner’s behalf when finding suitable transaction partners. These agents are then used by the system to coordinate the preferences and interests of each party involved. MARI’s interaction with the user can be decomposed into several steps, and results in an agent being created to represent the user.

**Step 1. Specifying the Ideal Offer** (see Figure 5): The user specifies a “referential” or “preferred” configuration, or offer, which consists of specific product and transaction partner attribute values, as derived from the underlying domain ontology. Each ontology-specific attribute has a predefined “default” value associated with it, and the user can accept or override these defaults. The user can also modify which attributes are fixed and which are flexible and must also associate a monetary valuation (“bid” or “ask”) with this offer (referred to as the pbsvalue).

In general, an attribute can be either “fixed” or “flexible”. A fixed attribute is one whose value, as specified by the user, is used for transaction party qualification. By contrast, flexible attributes have associated ranges, and are used for transaction party valuation. For instance, in the example of language translation services (buyer’s perspective), the number of words to be translated could be a fixed attribute, while the reputation of the seller, the degree of expertise of the seller, and the amount of time within which the translation should be completed could be flexible attributes. Each fixed attribute has a predefined set of permissible values, and the user must select acceptable values from this set. For instance, the permissible values for ‘number of words to be translated’ might be the set of non-negative integers.

**Step 2. Gathering Ranges of Flexible Attributes** (see Figure 5): Users must also associate a permissible range of values with each flexible attribute. Flexible attributes essentially embody the tradeoffs that a given user is willing to make. Associated utility functions (discussed in section 4.1.3) define how the user’s valuation changes as flexible attributes vary over their permissible ranges.
4.1.3 Modeling User Utility Functions

For each flexible attribute, the aim is to gather sufficient information from the user so as to be able to accurately infer how her (uni-dimensional) utility might change as the flexible attribute varies over its permissible range, while all other attributes are held at their ideal (offer) values. As such, we require the user to specify the range of permissible valuations, referred to as \( \text{maxvalue} \) and \( \text{minvalue} \) in Equation 1, associated with the flexible attribute being held at the high and low endpoints of its permissible range, respectively, while all other attributes are held fixed at their optimal or offer values. Doing so enables us to accurately assess how the user would value product offerings and transaction partners that have not been explicitly seen or “rated” before. Based upon the

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Possible range</th>
<th>Your preferred value</th>
<th>Permissible range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seller Reputation</td>
<td>From 0 (worst) to 10 (best)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Seller Expertise</td>
<td>From 0 (worst) to 5 (best)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Task Completion Time</td>
<td>From 10 minutes to 100 minutes</td>
<td>40 minutes</td>
<td></td>
</tr>
</tbody>
</table>
market maker’s configuration parameters, MARI models the user’s utility function as follows:

Step 1). Visually Selecting Utility Functions: When first instantiating MARI, the market maker is required to visually associate a generic (pre-defined) mathematical function with each flexible attribute (see Figure 6). Users could be given the option of being able to override these “default” functions during the offer specification process.

![Figure 6: The Market Maker Visually Associates Utility Functions with Flexible Attributes](image)

Step 2). Quantifying Utility Functions: Using the generalized equation form of the utility function selected by the market maker (see Table 1), in conjunction with the pbsvalue, maxvalue, and minvalue parameters specified by a given user, MARI is able to compute a mathematical approximation to the utility function corresponding to each flexible attribute. The polynomial used to represent the function is usually a quadratic. Higher order polynomial approximations can optionally be calculated using Lagrange interpolation, as described in Step 3 below.
For example, let us assume that a given buyer is willing to accept a “seller reputation” ranging from 6 to 10. Assume that in her “referential offer” the buyer specifies a preferred value of 6. Further, say the market maker has pre-associated UF$_2$ with this flexible attribute as it varies over its range – the choice of this utility function would reflect the fact that the buyer is willing to bid higher as the seller’s reputation increases, and that her valuation increases exponentially as reputation approaches the maximum possible. In this case we can derive the equation which captures the change in the buyer’s utility as reputation varies, as:

\[
UF_2(x) = \left( \frac{\text{maxvalue} - \text{pbsvalue}}{(x_{hi} - x_{low})^2} \right) x^2 + \left( \frac{(-2)(\text{maxvalue} - \text{pbsvalue})(x_{low})}{(x_{hi} - x_{low})^2} \right) x + \left( \frac{\text{pbsvalue} + (\text{maxvalue} - \text{pbsvalue})(x_{low})^2}{(x_{hi} - x_{low})^2} \right)
\]

Equation 1

Where:

- $x_{\text{low}} =$ the value of the attribute specified in the referential offer (i.e. 6);
- $x_{\text{hi}} =$ high endpoint of the permissible range (i.e. 10).

---

7 This function is derived using standard algebraic techniques along with special properties of quadratic functions. In particular, we have used the fact that the global minima of a quadratic, of the form $y = ax^2 + bx + c$, occurs at $-\frac{b}{2a}$, that the quadratic function corresponding to UF$_2$ is monotonically increasing, and that the user has already revealed two data points, $(x, y)$, on the curve: $(x_{\text{low}}, \text{pbsvalue})$ and $(x_{\text{hi}}, \text{maxvalue})$.  


Table 1: MARI's Predefined Generic Utility Functions (x is a place-holder variable for a value of the flexible attribute over its permissible range)

<table>
<thead>
<tr>
<th>Name</th>
<th>Functional Generalization</th>
<th>Graph Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF₁</td>
<td>(ax ± b)</td>
<td></td>
</tr>
<tr>
<td>UF₂</td>
<td>(ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₃</td>
<td>(-ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₄</td>
<td>(-ax ± b)</td>
<td></td>
</tr>
<tr>
<td>UF₅</td>
<td>(ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₆</td>
<td>(-ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₇</td>
<td>x &lt; (midpoint of permissible range) ? (ax ± b) : (-ax ± b)</td>
<td></td>
</tr>
<tr>
<td>UF₈</td>
<td>x &lt; (midpoint of permissible range) ? (ax² ± bx ± c) : (ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₉</td>
<td>x &lt; (midpoint of permissible range) ? (-ax² ± bx ± c) : (-ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₁₀</td>
<td>x &lt; (midpoint of permissible range) ? (-ax ± b) : (ax ± b)</td>
<td></td>
</tr>
<tr>
<td>UF₁₁</td>
<td>x &lt; (midpoint of permissible range) ? (ax² ± bx ± c) : (ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₁₂</td>
<td>x &lt; (midpoint of permissible range) ? (-ax² ± bx ± c) : (-ax² ± bx ± c)</td>
<td></td>
</tr>
<tr>
<td>UF₁₃</td>
<td>x = (high end point of permissible range) ? 1 : 0</td>
<td></td>
</tr>
<tr>
<td>UF₁₄</td>
<td>x = (high end point of permissible range) ? 0 : 1</td>
<td></td>
</tr>
<tr>
<td>UF₁₅</td>
<td>x = 1</td>
<td></td>
</tr>
<tr>
<td>UF₁₆</td>
<td>x = (low end point of permissible range) ? 1 : 0</td>
<td></td>
</tr>
<tr>
<td>UF₁₇</td>
<td>x = (low end point of permissible range) ? 0 : 1</td>
<td></td>
</tr>
</tbody>
</table>

Step 3). Refining Utility Functions through Revealed Preferences (Optional): The market maker can optionally configure the system to fine-tune the quadratic utility functions captured in Step 2. This requires that the market maker iteratively “train” the system. The market maker is asked to explicitly “valuate” hypothetical product offerings strategically.
chosen to representatively span the space of all relevant product offerings. Essentially, the preferences expressed by the market maker are taken to be a benchmark set of "reasonable" preferences. Using these revealed preferences in conjunction with offer data that is specific to a particular user, the system facilitates the construction of a (iteratively refined) piecewise, linear approximation of the user's utility function.

If there are $n$ flexible attributes, then the user's utility function can be visualized as an $n$ dimensional hyper plane in $(n+1)$ dimensions (where the $(n+1)^{st}$ dimension is the numerical "monetary" valuation associated with each point on the hyper plane). Using Lagrange Interpolation, in conjunction with an iterative scheme we shall refer to as "delta scaling," it becomes feasible to determine higher degree polynomial approximations for utility functions.

In general, we can use Lagrange Interpolation to approximate the value of any $(n-1)^{st}$ degree polynomial, $f$, at any arbitrary point, $x$, provided that we already know the values of the function ($f_0, f_1, ..., f_{n-1}$) at $n$ distinct points ($x_0, x_1, ..., x_{n-1}$), by using the following expression:

\[
    f(x) = f_0 \frac{(x-x_1)(x-x_2)...(x-x_{n-1})}{(x_0-x_1)(x_0-x_2)...(x_0-x_{n-1})} + ... + f_{n-1} \frac{(x-x_0)(x-x_1)...(x-x_{n-2})}{(x_{n-1}-x_0)(x_{n-1}-x_1)...(x_{n-1}-x_{n-2})}
\]

Equation 2

The term "delta scaling" refers to the technique by which the system picks points in the attribute space to be valued by the market maker. The algorithm goes through iterations referred to as "delta phases." In each delta phase, the algorithm unidimensionally varies the value of a single flexible attribute (by "delta"), while all other attributes are held fixed at their ideal (offer) values, and asks the market maker to explicitly associate a valuation with the feature set, thus effectively obtaining an additional data point relevant to this attribute. Subsequently, the algorithm performs Lagrange interpolation to compute a higher degree (uni-dimensional) utility function for the attribute. In a given delta phase, the algorithm does this for each attribute, thereby improving the degree of each attribute's associated utility function by at least one. The
market maker thus provides MARI with an internal model of reasonable user preferences. This model can be thought of as a "template," that is subsequently adapted to user-specific data.

4.2 Matching Buyers and Sellers

4.2.1 Overview

Finding automated matches between buyer and seller agents in electronic markets has been at the forefront of research in multi-agent e-business systems. While a number of heuristics have been explored for brokering transactions, in the vast majority of systems there is potentially a conflict between maximizing global welfare or surplus\(^8\), versus allowing each agent to act in self interest to optimize its individual gains.

Maximizing aggregate welfare for a collection of buyer and seller agents is not necessarily harmonious with allowing each individual economic agent to act in self-interest\(^9\). This is equivalent to saying that a centralized entity that allocates scarce resources, seeking to maximize the aggregate profit of a "society" of agents, may leave some individual agents faring very poorly, essentially as a "sacrifice" to improve the aggregate lot of society at large. However, from the perspective of the self-interested, individualistic agent who is required to make this sacrifice, this is hardly an attractive outcome!

To illustrate this problem, let us consider a stylized example of routing traffic along major routes. For example, let's say that a fictitious country consists of two cities, C\(_1\) and C\(_2\), with two roads connecting the cities, R\(_1\) and R\(_2\), such that R\(_2\) (side-streets) is a significantly more time consuming route than R\(_1\) (freeway). Further, let's say that, at a given time, 100 agents have to travel from C\(_1\) to C\(_2\), and that the capacity of each interconnecting road is 98. In this case, each individual agent would clearly prefer to take R\(_1\) (freeway), since it is faster and quicker. However, if all 100 agents were to take

---

\(^8\) Aggregate surplus is the sum of consumer and producer surplus. Consumer surplus is defined as the difference between the amount a consumer is willing to pay for a good and the amount she actually pays. Producer surplus is defined as the difference in the market price the producer receives for a good and the marginal cost incurred in its production [5].

\(^9\) The converse is also true: allowing each agent to act in self-interest does not, in any way, guarantee that "society," as a whole, will be well off.
R₁, the freeway would exceed capacity, leading to traffic jams and thereby depleting the time and speed advantages that it could otherwise have offered. As such, a centralized entity that regulates how traffic is routed would choose to allow only 98 agents to take R₁ and would require that the remaining two agents make a "sacrifice" by taking R₂, so as to improve the lot of society at large. The resulting configuration can be shown to maximize aggregate surplus and to be "Pareto optimal" [15, 38], meaning that no agent can be made better off without making another agent worse off (transferring an agent from R₂ to R₁ would require an agent to move from R₁ to R₂, or would otherwise make things miserable for all agents on R₁ due to overcrowding). In general, it is precisely due to the outcome of welfare or "surplus" maximization and Pareto optimality that centrally mediated resource allocation techniques have been embraced in the past. However, as demonstrated in this example, this approach clearly has the potential to alienate some agents. Since each agent is concerned primarily with its own well being, and not with the benefit of society as a whole, this scheme clearly disincentives individualistic agents from participating in the first place.

It is worth contrasting the above argument with the classical laissez-faire economics contention that, given perfect information and rational behavior on the part of each economic agent, allowing each agent to act freely in its selfish self-interest will translate into society being best off (first welfare theorem of neoclassical economics) [15, 38]. However, in general, agent-based electronic markets are characterized neither by perfect information (often computationally intractable for software agents), nor perfect rationality. Hence, the laissez-faire argument cannot be assumed to hold in general, and more generically applicable techniques for matching buyer and seller agents, such as the one proposed here, are needed.

As part of the MARI system, we have developed a technique applicable to brokering transactions in agent-based electronic markets that permits us to maximize aggregate social welfare (defined as aggregate surplus) while, at the same time, permitting each agent to find a transaction partner that is myopically optimal from its self-interested perspective.

---

¹⁰ This is especially true if agents are created by third parties and simply participate in the market using some published market interface. In general, the market cannot make any assumptions about rational behavior on the part of such agents.
As discussed previously, MARI allows users to specify an ideal offer, as well as to cite ranges of attribute values over which they are willing to be flexible. Using this information, we can build a mathematical model of user utility functions. This model of user preferences allows us to predict how much a given user is willing to bid (or ask) for a given transaction partner, as well as to gauge how “far” the transaction partner lies from the user’s desired ideal offer configuration.

Specifically, for each buyer we can look at each qualified seller to see whether a deal is possible and, if so, what the “Optimal Transaction Configuration” (OTC) would be. Exploring the space of possible deals is equivalent to exploring the space defined by flexible attribute ranges, as specified by the buyer and seller. The OTC is simply a particular configuration of attributes, or a “deal,” that is consistent with the permissible attribute ranges that both the buyer and the seller are willing to tolerate, and is the “deal” which lies “closest” to the buyer’s and seller’s ideal offers.

The notion of the OTC captures the “deal” configuration that is myopically optimal in the context of a specific buyer-seller pairing, from the self-interested perspectives of the two buyer and seller agents involved in this pairing. Calculating the surplus (defined as the bid-ask spread) corresponding to the OTC allows us to gauge how the “goodness” (in terms of aggregate welfare or surplus) of this pairing might compare with that of other pairings. Since, for any given buyer, an OTC will be computed for every qualified seller, comparing the relative “goodness” of the various OTCs gives us a global heuristic of surplus maximization by which we may decide which of the various qualified sellers the buyer ought to be ultimately paired with.

4.2.2 Generating Data Points from Functional Approximations

A mathematical model of preferences allows us to predict how much a given user would be willing to bid (or ask) for a given transaction partner. We begin by generating additional data points, using the seed data provided to us by the user. In order to formulate a functional generalization for a user’s multi-dimensional utility function, we use a quadratic least squares approximation technique [20].

11 Here, as in other parts of the paper, we will take the buyer’s point of view in discussing our brokering methodology. We could just as well have adopted the seller’s perspective, without loss of generality.
To do so, we first generate an $nxm$ matrix $A$ for every user. Every row of $A$ corresponds to a hypothetical configuration of attributes, or market "bundles," derived using the user's ideal offer in conjunction with flexible attribute ranges. For example, let's consider a "Professional Buyer" $B_{\text{PRO}}$. Let's assume $B_{\text{PRO}}$ possesses a reputation of 10 and, ideally, is willing to pay $100 for a seller who can translate 4000 words in 30 minutes and who has a reputation of 10 and an expertise of 5. Further, let's say that $B_{\text{PRO}}$ is willing to be flexible on translation time (30-120 minutes) and reputation (5-10), and associates utility functions $UF_5$ and $UF_2$ (see Table 1) with these flexible attributes, respectively. $B_{\text{PRO}}$'s "profile" is summarized in Table 2.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function (see Figure 1)</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Reputation</td>
<td>10</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Expertise</td>
<td>5</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>30</td>
<td>30-120</td>
<td>$UF_2$</td>
<td>$80-$100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$UF_5$</td>
<td>$100-$65</td>
</tr>
</tbody>
</table>

Table 2: Profile of a "Professional Buyer" $B_{\text{PRO}}$

The "bid range" in Table 2 is deduced by explicitly asking the user a sequence of questions. For instance for the time attribute, the $65$ figure is obtained by asking the buyer how much she would be willing to pay if translation time were to equal 120 minutes, while all other attributes are held fixed at their optimal values (as discussed in Section 4.1.3). In other words, if $<4000, 5, 10, 30>$ is the buyer's offer bundle, we ask the buyer to "bid" on a bundle of the form $<4000, 5, 10, 120>$. With this information, in conjunction with the fact that we know that $UF_2$ (see Figure 1) is a mathematical approximation to how the buyer's valuation changes from $100$ to $65$ as time ranges from 30 to 120 minutes, we can discretize the $[30, 120]$ range to automatically generate data points of the form $<4000, 5, 10, [30...120]>$ and corresponding bids. Doing so for every flexible attribute allows us to generate the matrix $A$, and a corresponding bid vector.
b. A will thus be an \( nx4 \) matrix, where the exact value of \( n \) is configurable, depending on how many data points we generate.

Having thus generated \( A \) and \( b \), the task is to model the underlying mathematical function that maps each row in \( A \) to each row in \( b \). We can do so using least squares data fitting [20] – a well known technique in linear algebra, where the problem is to solve an over-determined system of equations \( Ax = b \), so as to deduce the vector \( x \) that maps each row of \( A \) to the corresponding entry in \( b \). The vector \( x \) can be interpreted as a set of coefficients or “weights,” which effectively defines a function that can be used to map a configuration of attributes to a bid, in a manner consistent with the buyer’s revealed preferences.

In essence, the least squares technique attempts to “solve” the \( Ax = b \) system by minimizing the residual form \( b - Ax \). More precisely,

\[
\text{Given } A \in \mathbb{C}^{m \times n}, m \geq n, b \in \mathbb{C}^m, \\
\text{We find } x \in \mathbb{C}^n \text{ such that } \|b - Ax\|_2 \text{ is minimized.}
\]

Equation 3

Here \( \|b - Ax\|_2 \) denotes taking the 2-norm [20], which corresponds to euclidean distance. As such, the geometric interpretation of Equation 3 is that we seek a vector \( x \in \mathbb{C}^n \) such that the vector \( Ax \in \mathbb{C}^m \) is the closest point in \( \text{range}(A) \) to \( b \).

Once we determine \( x \) using our automatically generated data points, we can effectively automatically estimate the buyer’s bids for various sellers in the market. Each differentiated seller offering simply represents a combination of attributes, of the same form as a row of \( A \). Since \( x \) approximates the mapping from each row of \( A \) to a bid, the process of “valuating” a given seller is simply a matter of “scaling\textsuperscript{12}” each seller attribute by the corresponding entry in \( x \).

\textsuperscript{12} We use the term scaling, since the entries of \( x \) can be interpreted as “weights,” signifying different degrees of importance that the user associates with different attributes.
4.2.3 Exploring the Attribute Space to identify the Optimal Transaction Configuration (OTC)

Subsequently, for each buyer-seller pair in the market, we explore the space of potential transactions or “deals” that can take place. Exploring the space of possible deals is equivalent to exploring the space defined by flexible attribute ranges, as specified by the buyer and seller. For instance, let’s consider a “Professional Seller,” $S_{PRO}$, whose “profile” is summarized in Table 3 (analogous to Table 2):

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function (see Figure 1)</th>
<th>Ask Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>4000-6000</td>
<td>$\text{UF}_1$</td>
<td>$$100-$150$</td>
</tr>
<tr>
<td>Expertise</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reputation</td>
<td>10</td>
<td>Fixed</td>
<td>$\text{N/A}$</td>
<td>$\text{N/A}$</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>60</td>
<td>30-120</td>
<td>$\text{UF}_6$</td>
<td>$$125-$85$</td>
</tr>
</tbody>
</table>

Table 3: Profile of a “Professional Seller” $S_{PRO}$

Comparing $S_{PRO}$ with $B_{PRO}$, clearly we see that any attribute (such as ‘time’) that is a flexible attribute for both parties with overlapping range, is a candidate for “negotiation.” A possible “deal” between $S_{PRO}$ and $B_{PRO}$ could involve any value of ‘time’ between 30 and 120. However, since, $B_{PRO}$ and $S_{PRO}$ each have different optimal values (30 and 60 minutes respectively), a point in the interval $[30, 120]$ will not be equally desirable from the perspective of both parties. Our goal is to delineate the particular “deal” (Optimal Transaction Configuration or OTC) that is myopically optimal from the self-interested perspectives of the buyer and seller agents involved in the pairing. As such, we use a distance function (see Equation 4) to assess how any given “deal” might deviate from the buyer’s and seller’s optimal “bundles:”

$$\text{Distance, } \Delta = \sqrt{\sum [(d_i - B_i)^2 + (d_i - S_i)^2]}$$

Equation 4
Where:

- \( i \), ranges over all flexible attributes;
- \( d_i \) is the value of attribute \( i \) corresponding to the specific deal, \( d \), under consideration;
- \( B_i \) is the value of attribute \( i \) corresponding to the Buyer's optimal offer;
- \( S_i \) is the value of attribute \( i \) corresponding to the Seller's optimal offer.

For each buyer-seller pair in the marketplace, we can enumerate a set of possible “deals” that can be brokered. By searching over the (discretized) underlying attribute space, corresponding to differentiated buyer and seller offerings, we can identify a particular “deal” that is suitable from the perspective of both transaction partners, in the sense that it lies close to both of their “optimal” configurations. By searching over the whole attribute space, in conjunction with our model of user preferences, we effectively explore permissible tradeoffs that users’ are willing to make, and implicitly negotiate over the holistic product offering to identify a “transaction configuration” that is suitable for both the buyer as well as the seller.

From the set of all possible “deals” for which surplus is non-negative (i.e. bid is greater than or equal to the ask) we identify the particular deal for which \( \Delta \) is minimized, and call that the Optimal Transaction Configuration (OTC). Since, for any given buyer, an OTC will be computed for every qualified seller, comparing the bid-ask spread (surplus) corresponding to each OTC gives us a global heuristic of surplus maximization by which we may decide which of the various qualified sellers the buyer ought to be ultimately paired with.

**4.2.4 Delineating Transaction Partners to Maximize Aggregate Surplus**

Finally, having found an OTC corresponding to each possible buyer-seller pair in the market, we use a heuristic of aggregate surplus maximization so as to delineate transaction pairs.

In essence, for each buyer, we evaluate the “cost” that would be incurred if the buyer were to engage in a transaction with any of the qualified sellers. Currently, we take this “cost” to be equal to the OTC “bid-ask spread,” which can be interpreted as the
aggregate surplus [15, 38] that the two parties would derive if the transaction were to take place. We use this metric of “cost” since our indicator of the “goodness” of an allocation is welfare, which, in this case, is measured by the surplus that the allocation generates.

Subsequently, we can conveniently formulate the problem of optimally pairing up buyers and sellers as a “matching” problem. Mathematically, we can represent the state of the marketplace as a graph \( G \), in which sellers and buyers represent nodes. We refer to the set of buyers and seller as \( B \) and \( S \), respectively, and to the set of arcs as \( A \). Hence,

\[
G = S \cup B; \quad \text{and,}
\]

**Equation 5**

\[ S \cap B = \emptyset. \]

**Equation 6**

Equation 5 says that all nodes in the network are either seller nodes or buyer nodes. Equation 6 captures the fact that each node in the network can uniquely be classified as a buyer node or as a seller node without ambiguity. Each individual buyer node \( b \in B \) is connected to a subset of seller nodes \( S' \subseteq S \), via some arc \((b, s)\) with associated arc “cost” \( c_{bs} \).

In general, we can expect that, in the absence of buyer coalitions or seller “cliques” \( G \) will be bipartite\(^{14}\). The bipartite matching problem is a special case of the general matching problem and allows for algorithmic simplifications in its solution methodology\(^{15}\).

---

\(^{13}\) We assume some basic knowledge of graph theory and terminology here. To learn more, we invite the reader to browse some of the Network Optimization and Graph Theory books referenced in the bibliography.

\(^{14}\) A graph \( G = (N, A) \), where \( N \) is the set of Nodes and \( A \) is the set of Arcs, is said to be bipartite if we can partition its node set into two subsets \( N_1 \) and \( N_2 \) so that for each arc \((i, j)\) in \( A \) either (i). \( i \in N_1 \) and \( j \in N_2 \), or (ii). \( i \in N_2 \) and \( j \in N_1 \) [36].

\(^{15}\) Exploring buyer and seller coalitions would be a fascinating and extremely intriguing area of research within the context of this project. Indeed, it is our expectation that, facilitating such coalitions in our system should be fairly straightforward, given our modular design and generalized treatment of the overall problem. However, in the interest of simplicity, we will resist the temptation of analyzing such a market structure in depth, for the time being.
Given this formulation, our goal is to find a sub-graph $G' \subseteq G$, such that the sub-graph represents a feasible\textsuperscript{16} pairing of buyers and sellers with the largest overall "cost" (surplus), defined as the sum of the costs of its constituent arcs. To accomplish this, our solution strategy mirrors that of a (modified)\textsuperscript{17} minimum cost flow problem. With this formulation the matching problem can now be expressed as the following linear program [36]:

Minimize $\sum_{(i,j) \in A} c_{ij} x_{ij}$

Equation 7

Subject to the constraints:

$$\sum_{(j, i) \in A} x_{ij} = 1 \text{ for all } i \in S,$$

Equation 8

$$\sum_{(i, j) \in A} x_{ji} = 1 \text{ for all } i \in B,$$

Equation 9

$$x_{ij} \geq 0 \text{ for all } (i, j) \in A.$$

Equation 10

Conveniently, we can now apply a direct implementation of the Successive Shortest Path algorithm for the minimum cost flow problem in order to find an optimal matching [36].

The successive shortest path algorithm functions by first finding the residual network, $G(x)$ corresponding to $G$\textsuperscript{18}. It then finds shortest paths from a "supply" node (which, in our formulation, conveniently corresponds to a seller node) to all other nodes in $G(x)$. The distances computed are then used to update node "potentials" (denoted by

\textsuperscript{16} In this case the "feasibility" condition maintains that for a given buyer, the seller should be qualified to serve the buyer and that the buyer should meet the qualification criteria, if any, specified by the seller. Moreover, the buyer's bid can be no less than the seller's ask.

\textsuperscript{17} Since we are actually trying to maximize the sum of our costs (aggregate surplus), we redefine costs in the min cost flow formulation to be the negative of the computed surpluses. Minimizing the sum of the negatives of the original quantities is equivalent to maximizing the sum of the original quantities.

\textsuperscript{18} We can find the residual network, $G(x)$, corresponding to a given flow $x$ in $G$, by replacing each arc $(i, j) \in G$ by two arcs: $(i, j)$ and $(j, i)$, such that $(i, j)$ has cost $c_{ij}$ and residual capacity $r_{ij} = u_{ij} - x_{ij}$ ($u_{ij}$ denotes the capacity of arc $(i, j)$ in $G$) and arc $(j, i)$ has cost $-c_{ij}$ and residual capacity $r_{ij} = x_{ij}$. $G(x)$ consists only of arcs with positive residual capacity.
and to augment flow from the supply node to a demand node. Intuitively, then, for our purposes, when the algorithm augments one unit of flow in each iteration, that corresponds to assigning a seller in $S$ to a buyer in $B$. It can be shown that the running time of the algorithm is asymptotically bounded by $O(|S| \times F(|S + B|, |A|, \max(c_j)))$, where $F(|S + B|, |A|, \max(c_j))$ denotes the time needed to solve a shortest path problem\[36\]. Consequently, the algorithm terminates within $|S|$ iterations.

The algorithm concludes by identifying buyer-seller pairings for which the aggregate surplus of transaction parties is globally maximized. The “clearing price” for any given transaction pair is, by default, set at the midpoint between the original bid and ask prices, thereby equally dividing the surplus between the buyer and the seller. The market maker can, however, modify this distribution of surplus, choosing to retain the bid-ask spread as operating profit, for instance.

As a consequence of our matching technique, we are able to ensure that the following invariants are maintained\[20\]:

1. **Optimization Invariant**: Of all the “market baskets\[21\]” that a consumer can afford, a consumer always tries to select the basket which yields the highest utility\[22\]. This is a consequence of our local optimization technique, prior to running global matching.

2. **Efficiency Invariant**: The allocations determined by the system are economically efficient and consistent with some form of utility maximization. This is a consequence of our Pareto optimality condition, which guarantees that no potential economic gains are

---

\[19\] Strictly speaking, we should only use the shortest path algorithm when our arc costs are non-negative, in order to avoid the unhappy situation of negative cost cycles. However, in this case, since, by formulation, our graph, $G$, is bipartite and all arcs are unidirectional (going from nodes in $S$ to nodes in $B$), we can guarantee that no negative cost cycles will arise in the residual network $G(x)$.

\[20\] The optimization and equilibrium invariants are based on the “optimization principle” and “equilibrium principle” discussed by Varian [15].

\[21\] Here the term “market basket” refers to any set (possibly null or singleton) of goods and services that are purchased by a given consumer.

\[22\] The claim makes some implicit judgments about rational consumer behavior, namely that a rational consumer’s behavior is necessarily oriented towards utility maximization.
left unexploited and, hence, that the final allocation or matching is economically efficient.

4.2.5 Presenting Results

All matchings delineated by the system are considered tentative or “pending” until they are mutually “confirmed” by both transaction partners. If a user’s agent has been matched, the user has the option of reviewing and confirming the deal associated with the matched agent or doing nothing. Upon mutual confirmation by both parties involved in a “pending” transaction, the transaction is said to become “closed,” and the agents involved are “Expired” from the market, meaning that they are no longer considered in future matching cycles. Agents not involved in “closed” transactions are kept “Active,” or eligible to be matched in future matching cycles. This is desirable, since it increases the set of prospective transaction partners available to a given agent, and affords a greater deal of flexibility to the parties involved.

When “reviewing” a matched agent, the owner not only sees the details of the “deal” associated with his matched agent (see Figure 7), but also sees details of “potential” deals with other qualified transaction partners that were considered and, further, is able to see whether these prospective transaction partners were matched with someone else or not. Moreover, each deal shown to the user has a score metric associated with it (see Figure 8). The score metric printed with each deal reflects the surplus associated with that potential pairing. Surplus, taken here to be the bid-ask spread, is a valid scoring metric for both buyers and seller agents. The system only considers transactions for which the spread is positive and for which the bid is greater than the ask. As such, from the buyer’s perspective, for a given bid, a larger score indicates a lower closing price for a transaction, which translates into cost savings. Symmetrically, from the seller’s perspective, for a given ask, a larger score indicates a higher closing price for a transaction, which translates into greater profits.

Finally, when “reviewing” a matched agent, the owner is able to see whether the other transaction partner involved in the matching has already “confirmed” or unilaterally accepted. The presence of a unilateral confirmation by the other side makes it clear that
the transaction will consummate (become “closed”) by mutual confirmation if the owner confirms as well.

Figure 7: Details of the “deal” designated by the system for a matched agent

Qualified Transaction Partners Not Chosen for this Matching:

BUYER_REQUEST_ID : {FC275533-CFE8-4F48-8518-3A4E64D7778B}
This Prospective Agent (BUYER) Has Been Matched With Someone Else
Score: 123.695
Closing Price: 370.47

Details:
Buyer_Reputation = 5;
Language_From : French
Language_To : German
Number_of_Words = 4000;
Seller_Reputation = 3;
Seller_Skill = 3;
Time_to_Completion = 40;

BUYER_REQUEST_ID : {532FE899-46F5-45FA-9556-C08CA392C599}
This Prospective Agent (BUYER) Has Not yet Been Matched With Anyone Else.
Score: 51.539
Closing Price: 334.39

Details:
Buyer_Reputation = 5;
Language_From : French
Language_To : German
Number_of_Words = 4000;
Seller_Reputation = 3;
Seller_Skill = 3;
Time_to_Completion = 40;

Figure 8: Details of potential deals with other qualified transaction partners that were considered, with corresponding scores, and whether these prospective transaction partners were matched with someone else
5 THEORETICAL PRINCIPLES AND METHODOLOGICAL RATIONALE

5.1 Modeling Preference Functions

5.1.1 Assumptions

Within MARI, we attempt to determine user utility functions by explicitly observing behavior via revealed attribute preferences and then inferring what kind of preferences might have generated the observed behavior. In other words, asking the user to explicitly valuate hypothetical data gives us the opportunity to generate observations on choice behavior which, in turn, allow us to determine what, if anything, is being maximized. Once we have a mathematical estimate of the function that is being maximized, we can use this to predict choice behavior in previously unseen situations, as well as to evaluate the effects of proposed policy changes in the economic environment.

As is standard practice in economic literature, we assume that user preferences are “well behaved.” More formally, in order for preference relations to be representable by mathematical utility functions, we assume that preferences are:

1. Complete: Meaning that preferences are well defined over the entire space of possible product attribute values. This assumption guarantees that individuals are able to completely understand and decide between any two alternatives. This assumption also rules out irrational preferences, in which an individual might claim that both A is preferred to B and that B is preferred to A.

2. Transitive: Meaning that, if the consumer prefers Product A to Product B, and prefers B to C, then the consumer also prefers A to C. This assumption guarantees that an individual’s choices are internally consistent.

3. Continuous: If an individual prefers A to B, then situations “close” to A must also be preferred to B. Mathematically, the purpose of this assumption is to ensure that the
corresponding utility functions should not exhibit any singularities and should have quantifiable values at all points in the attribute-space.

### 5.1.2 Mathematical Formulation of Utility Functions

All our predefined utility functions (with the exception of UF\textsubscript{15} which is constant) have the property that they are either:

(i) **Type I**: Monotonically decreasing over the permissible range (UF\textsubscript{4}, UF\textsubscript{5}, UF\textsubscript{6}, UF\textsubscript{14}, UF\textsubscript{16});

(ii) **Type II**: Monotonically increasing over the permissible range (UF\textsubscript{1}, UF\textsubscript{2}, UF\textsubscript{3}, UF\textsubscript{13}, UF\textsubscript{17});

(iii) **Type III**: Monotonically increasing till the midpoint of the permissible range, and monotonically decreasing thereafter (UF\textsubscript{7}, UF\textsubscript{8}, UF\textsubscript{9}); or,

(iv) **Type IV**: Monotonically decreasing till the midpoint of the permissible range, and monotonically increasing thereafter (UF\textsubscript{10}, UF\textsubscript{11}, UF\textsubscript{12}).

Type III (Type IV) functions can be seen as the “aggregation” of a Type II (Type I) followed by a Type I (Type II) function. In fact, this is how we represent Type III and IV functions within MARI. Given these observations, we can derive a generic mathematical formulation for the utility functions corresponding to each of the flexible attributes as shown in Table 4, Table 5 and Table 6\textsuperscript{23}. It should be noted that in each case the degree of the polynomial used to represent the function can be at most two (quadratic). Higher order polynomial approximations can subsequently be created using Lagrange interpolation.

For the purpose of generating generic mathematic forms of the functions, we assume the presence of data-points of the form \((x, UF_i(x))\). These data points are obtained directly from the user during the normal course of interaction with MARI. The front-end user interaction model ensures that the system asks the user sufficient questions so as to be able to obtain these data points. It should be noted that all Type II functions

\textsuperscript{23} In each case, Type III and Type IV functions are treated as “aggregations” of Type I and Type II functions. Further, we assume that functions are only evaluated over valid \(x\) values that are contained in the permissible interval, inclusive of endpoints.
only assume knowledge of two data-points, despite the fact that these are quadratic functions. This is a consequence of the fact that, for simplicity, we have assumed that the quadratic function reaches its minimum or maximum value at the point where it levels off (where the slope of the function is zero). In some cases, the user might voluntarily reveal three data points, more than the minimum two required by our system. This could happen, for instance, when the optimal value of a flexible attribute lies between, and is distinct from, the maximum and minimum values of the attribute. In this case, the system prompts the user to associate prices with all three attribute values. When this happens, the system automatically uses the extra data point to generate an exact quadratic representation of the utility function corresponding to the attribute.

In our prototype implementation, we have elected to retain only Type I and Type II functions for simplicity. However, for the sake of completeness, we will derive the equation forms of all utility functions, dividing the analysis into three “cases:”

Case 1: Utility functions represented by degree one (linear) polynomials, i.e., $U_{F1}$, $U_{F4}$, $U_{F7}$, $U_{F10}$. Since, two points are sufficient to define a line in a plane, we can easily calculate these linear functions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Known Data Points</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_{F1}$</td>
<td>$(x_{low}, pbs\text{value}); (x_{hi}, max\text{value})$</td>
<td>$U_{F1}(x) = (pbs\text{value}) \frac{(x-x_{low})}{(x_{hi}-x_{low})} + (max\text{value}) \frac{(x-x_{low})}{(x_{hi}-x_{low})}$</td>
</tr>
<tr>
<td>$U_{F4}$</td>
<td>$(x_{low}, pbs\text{value}); (x_{hi}, min\text{value})$</td>
<td>$U_{F4}(x) = (pbs\text{value}) \frac{(x-x_{low})}{(x_{hi}-x_{low})} + (min\text{value}) \frac{(x-x_{low})}{(x_{hi}-x_{low})}$</td>
</tr>
<tr>
<td>$U_{F7}$</td>
<td>$(x_{low}, pbs\text{value}); (x_{mid}, max\text{value}); (x_{hi}, pbs\text{value})$</td>
<td>$U_{F7}(x) = \begin{cases} (pbs\text{value}) \frac{(x-x_{mid})}{(x_{low}-x_{mid})} + (max\text{value}) \frac{(x-x_{low})}{(x_{mid}-x_{low})} &amp; \text{if } x_{low} \leq x &lt; x_{mid} \ (max\text{value}) \frac{(x-x_{hi})}{(x_{mid}-x_{hi})} + (pbs\text{value}) \frac{(x-x_{mid})}{(x_{hi}-x_{mid})} &amp; \text{if } x_{mid} \leq x \leq x_{hi} \end{cases}$</td>
</tr>
</tbody>
</table>
Table 4: Utility Functions Represented by Degree One (Linear) Polynomials

Case 2: Utility functions represented by degree two (quadratic) polynomials, i.e., UF₂, UF₃, UF₅, UF₆, UF₉, UF₁₁, UF₁₂. Given the fact that we already know the general shape of these functions, we can easily derive general form equations using elementary algebra techniques along with special properties of quadratic functions. In particular, we have used the fact that the global minima of a quadratic, of the form \( y = ax^2 + bx + c \), occurs at \( x = \frac{-b}{2a} \). The astute reader will note that many of the derived equations are simply shifted and scaled reflections of each other across the vertical axis, \( x = \frac{x_{low} + x_{hi}}{2} \), or the horizontal axis, \( y = \frac{(y_{max} + y_{min})}{2} \) (where \( y_{max} \) and \( y_{min} \) respectively denote the maximum and minimum values of the function, \( y \), over the permissible range of the flexible attribute), or some combination thereof.

<table>
<thead>
<tr>
<th>Name</th>
<th>Known Data Points ((x, UF_f(x)))</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF₂</td>
<td>((x_{low}, pbsvalue)); ((x_{hi}, maxvalue))</td>
<td>( UF_2(x) = \frac{maxvalue - pbsvalue}{(x_{hi} - x_{low})^2} x^2 + \frac{(-2)(maxvalue - pbsvalue)(x_{low})}{(x_{hi} - x_{low})^2} x + \frac{pbsvalue + (maxvalue - pbsvalue)(x_{low})^2}{(x_{hi} - x_{low})^2} )</td>
</tr>
<tr>
<td>UF₃</td>
<td>((x_{low}, pbsvalue)); ((x_{hi}, maxvalue))</td>
<td>( UF_3(x) = -\frac{maxvalue - pbsvalue}{(x_{hi} - x_{low})^2} x^2 + \frac{(2)(maxvalue - pbsvalue)(x_{low})}{(x_{hi} - x_{low})^2} x + \frac{pbsvalue + (3)(maxvalue - pbsvalue)(x_{low})}{(x_{hi} - x_{low})^2} )</td>
</tr>
<tr>
<td>UF₅</td>
<td>((x_{\text{low}}, pbs\text{value}); (x_{\text{hi}}, \text{minvalue}))</td>
<td>(UF₅(x) = \left(\frac{pbs\text{value} - \text{minvalue}}{(x_{\text{hi}} - x_{\text{low}})^2}\right) x^2 + \left(\frac{(-2)(pbs\text{value} - \text{minvalue})(x_{\text{hi}})}{(x_{\text{hi}} - x_{\text{low}})^2}\right) x + \left(\frac{\text{minvalue} + (pbs\text{value} - \text{minvalue})(x_{\text{hi}})^2}{(x_{\text{hi}} - x_{\text{low}})^2}\right))</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>UF₆</td>
<td>((x_{\text{low}}, pbs\text{value}); (x_{\text{hi}}, \text{minvalue}))</td>
<td>(UF₆(x) = -\left(\frac{pbs\text{value} - \text{minvalue}}{(x_{\text{hi}} - x_{\text{low}})^2}\right) x^2 + \left(\frac{(2)(pbs\text{value} - \text{minvalue})(x_{\text{hi}})}{(x_{\text{hi}} - x_{\text{low}})^2}\right) x + \left(\frac{\text{minvalue} + (3)(pbs\text{value} - \text{minvalue})(x_{\text{hi}})^2}{(x_{\text{hi}} - x_{\text{low}})^2}\right))</td>
</tr>
</tbody>
</table>
| UF₈  | \((x_{\text{low}}, pbs\text{value}); (x_{\text{mid}}, \text{maxvalue}); (x_{\text{hi}}, pbs\text{value})\) | \(UF₈(x) = \left(\frac{\text{maxvalue} - pbs\text{value}}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x^2 + \left(\frac{(-2)(\text{maxvalue} - pbs\text{value})(x_{\text{mid}})}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x + \left(\frac{pbs\text{value} + (\text{maxvalue} - pbs\text{value})(x_{\text{mid}})^2}{(x_{\text{mid}} - x_{\text{low}})^2}\right)\) if \(x_{\text{low}} \leq x < x_{\text{mid}}\)
| UF₉  | \((x_{\text{low}}, pbs\text{value}); (x_{\text{mid}}, \text{maxvalue}); (x_{\text{hi}}, pbs\text{value})\) | \(UF₉(x) = \left(\frac{\text{maxvalue} - pbs\text{value}}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x^2 + \left(\frac{(-2)(\text{maxvalue} - pbs\text{value})(x_{\text{mid}})}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x + \left(\frac{pbs\text{value} + (3)(\text{maxvalue} - pbs\text{value})(x_{\text{mid}})^2}{(x_{\text{mid}} - x_{\text{low}})^2}\right)\) if \(x_{\text{mid}} \leq x \leq x_{\text{hi}}\)
| UF₁₁ | \((x_{\text{low}}, pbs\text{value}); (x_{\text{mid}}, \text{minvalue}); (x_{\text{hi}}, pbs\text{value})\) | \(UF_{11}(x) = \left(\frac{pbs\text{value} - \text{minvalue}}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x^2 + \left(\frac{(-2)(pbs\text{value} - \text{minvalue})(x_{\text{mid}})}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x + \left(\frac{\text{minvalue} + (pbs\text{value} - \text{minvalue})(x_{\text{mid}})^2}{(x_{\text{mid}} - x_{\text{low}})^2}\right)\) if \(x_{\text{mid}} \leq x < x_{\text{low}}\)
| | | \(UF_{11}(x) = \left(\frac{pbs\text{value} - \text{minvalue}}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x^2 + \left(\frac{(-2)(pbs\text{value} - \text{minvalue})(x_{\text{mid}})}{(x_{\text{mid}} - x_{\text{low}})^2}\right) x + \left(\frac{\text{minvalue} + (pbs\text{value} - \text{minvalue})(x_{\text{mid}})^2}{(x_{\text{mid}} - x_{\text{low}})^2}\right)\) if \(x_{\text{mid}} \leq x < x_{\text{low}}\)
Table 5: Utility Functions Represented by Degree Two (Quadratic) Polynomials

Case 3: All other Utility functions, i.e, UF13, UF14, UF15, UF16, UF17.

<table>
<thead>
<tr>
<th>Name</th>
<th>Known Data Points</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, UF13(x))</td>
<td>(xlow, pbsvalue); (xhi, maxvalue)</td>
<td>UF13(x) = \begin{cases} \text{pbsvalue} &amp; \text{if } x_{low} \leq x &lt; x_{hi} \ \text{maxvalue} &amp; \text{if } x = x_{hi} \end{cases}</td>
</tr>
<tr>
<td>(x, UF14(x))</td>
<td>(xlow, pbsvalue); (xhi, minvalue)</td>
<td>UF14(x) = \begin{cases} \text{pbsvalue} &amp; \text{if } x_{low} \leq x &lt; x_{hi} \ \text{minvalue} &amp; \text{if } x = x_{hi} \end{cases}</td>
</tr>
<tr>
<td>(x, UF15(x))</td>
<td>(xlow, pbsvalue); (xhi, pbsvalue)</td>
<td>UF15(x) = \text{pbsvalue}</td>
</tr>
<tr>
<td>(x, UF16(x))</td>
<td>(xlow, pbsvalue); (xhi, minvalue)</td>
<td>UF16(x) = \begin{cases} \text{pbsvalue} &amp; \text{if } x = x_{low} \ \text{minvalue} &amp; \text{if } x_{low} &lt; x \leq x_{hi} \end{cases}</td>
</tr>
<tr>
<td>(x, UF17(x))</td>
<td>(xlow, pbsvalue); (xhi, minvalue)</td>
<td>UF17(x) = \begin{cases} \text{pbsvalue} &amp; \text{if } x = x_{low} \ \text{maxvalue} &amp; \text{if } x_{low} &lt; x \leq x_{hi} \end{cases}</td>
</tr>
</tbody>
</table>

Table 6: Other Utility Functions

5.2 Mechanism Design

From a functional perspective, MARI's goal is to match buyers and sellers while taking into account relevant economic criteria, such as Pareto optimality, utility functions, and surplus maximization. As such, one can envision several possible mechanism designs. Here, we identify a few candidate mechanisms we have considered, and discuss the pros and cons of each for our purposes.
1. **Sealed Bid Double Auction**: This strategy is similar to that employed in MarketMaker [9]. In this case, each buyer (seller) defines an upper and lower bound on the price he or she is willing to accept, and delineates a mathematical function that defines how the price varies from the lower (upper) bound to the upper (lower) bound with the progression of time [7]. Over time, the buyer (seller) increases (decreases) her price until there is some seller (buyer) who is willing to transact at that price, or until the upper (lower) bound is reached or until the permissible amount of time available expires. In the latter two cases, the transaction does not take place.

   The analysis of the sealed bid double auction can be reduced to a weak version of Bayesian Nash equilibrium [8]. Since, in this scheme, buyers and sellers do not directly reveal their valuations and preferences, each round of the auction will, in general, not be Pareto optimal [28]. Secondly, it should be noted that this method would violate one of the design goals intrinsic to the MARI architecture -- to transcend the adversarial winner-loser characteristic inherent to common-value auctions24.

2. **Repeated Vickrey Auction**: In this case, we individually auction the services of the sellers to the buyers [49]. The Vickrey auction is like the conventional sealed bid auction with the critical difference that although the good is awarded to the highest bidder, it is awarded at the second-highest price that is above the seller’s reservation price. It can be shown that in a Vickrey auction, it is always in each player’s interest to bid their true value [15]. It can also be shown that a Vickrey auction will always result in a Pareto optimal outcome [15], making it an attractive candidate for a resource allocation mechanism.

   However, like any other auction, a Vickrey auction is susceptible to collusion and other forms of strategic behavior. Most importantly though, for our purposes, this mechanism only maximizes the welfare of the economy if we assume that the sellers are homogenous [29]. Needless, to state, this assumption is not reasonable in the kinds of markets we are interested in focusing on.

---

24 A common-auction is defined as an auction in which the good in question has essentially the same value to all bidders, even though bidders may have different estimates of that value [15].
3. **Clearinghouse Double Auction:** Conceptually, this method is similar to the Sealed Bid Double auction except that in this case we attempt to find a single market clearing price at which many (possibly all) buyers and sellers can be matched up [7].

The Clearinghouse functions by examining the sealed bids of the buyers and sellers at predefined frequencies, and using these to compute mathematical demand and supply curves. Subsequently, the intersection of these curves is used to identify a single equilibrium or market-clearing price, which is then enforced for all the associated transactions.

One potential shortfall of this method, however, is that it would not be very accurate in thin or semi-liquid markets since, in general, to come up with meaningful demand and supply curves, we will require that some critical mass of buyers and sellers be present. Indeed, in MARI, we expect that markets for many specialized services will, in fact, be thin, and we want that our matching methodology should be robust enough to handle such situations. Secondly, one of the manifest goals of MARI is, in fact, to transcend this notion of a "single market clearing price" and make the transaction experience personalized at the individual level. Hence, the Clearinghouse Double Auction is not an appropriate choice for our system.

### 5.3 Optimization Heuristics for Transaction Brokering

We use a matching technique in conjunction with an optimization heuristic, to successfully match up buyers with sellers. Different heuristics will differ in the metrics they employ in formulating the objective function of the optimization problem. Such metrics could include, for instance: (maximizing) cumulative profits or revenues of the sellers, (maximizing) profits or revenues of a particular seller or a subset of sellers, (minimizing) cumulative prices paid by buyers, (minimizing) the price paid by a particular buyer or some subset of buyers, (maximizing) cumulative producer surplus\(^{25}\) or the surplus of some (possibly singleton) subset of sellers, (maximizing) cumulative producer surplus can be defined as the sum over all units of production of the difference between the market price of the good and the marginal cost of production (Pindyck and Rubinfeld, p.267).

\(^{25}\)
consumer surplus\(^{26}\) or the surplus of some (possibly singleton) subset of buyers, (maximizing) combined surplus of consumers and producers, etc. Clearly, there can be many different economic institutions for allocating resources. In order to be able to compare these schemes to determine which is the best, we must first define the concept of “best” in our context. A useful criteria for comparing these outcomes is by using the notion of Pareto optimality\(^{27}\). If an allocation of resources has the property that it is not possible to make any one person better off without making someone else worse off at the same time, then we say that it is Pareto Optimal, and that no Pareto Improvement is possible [15].

More precisely, a market allocation is said to be Pareto Optimal if [15]:
1. There is no way to make all the market participants involved better off; or
2. there is no way to make any one individual better off without making some other individual worse off at the same time; or
3. all possible gains from trade have been exhausted; or
4. there are no further mutually advantageous trades to be made.

Conventional economics often associates Pareto optimal with a market in which the seller is able to perfectly price discriminate. First-degree or Perfect price discrimination means that the seller is able to sell different units of output for different prices and vary these prices across different buyers [15]. Hence, prices differ across units of the good as well as across customers. perfect discrimination is associated with the producer capturing all the surplus in the market. In this scenario, the producer is seen as wanting to maximize his or her profits (producer surplus) subject to the fact that the good is sold at the buyer’s reservation price\(^{28}\). Interestingly enough, though, this means that the outcome will be Pareto optimal, since there is no way to make both the producer and the consumer better off: we cannot increase the consumer’s surplus without decreasing the producer’s profit.

\(^{26}\) Consumer surplus can be defined as the difference between the amount consumers are willing to pay for a good and the amount they actually pay [38].

\(^{27}\) Pareto efficiency is named after the 19th century economist and sociologist Vilfredo Pareto.

\(^{28}\) The highest price that the buyer is willing to pay.
(surplus), and we cannot increase the producer's profit since it is already the maximum possible.

In our system, since we use a mediated approach to matching up buyers and sellers we are able to capture desirable Pareto optimality and surplus maximization properties, while preventing all the surplus from being usurped by any one party. MARI’s default optimization heuristic and matching technique guarantee that the final matching output by the system is Pareto optimal and maximizes surplus, as measure by the cumulative bid-ask spread amongst all prospective transaction partners in the market. The market maker can then control how this surplus ought to be divided amongst the buyers, sellers and himself.

5.4 Welfare Maximization

A maximal welfare allocation must necessarily be Pareto optimal [15]. This is easy to see by contradiction. If the allocation were not Pareto optimal, then it would be possible to redistribute resources to create a new allocation such that everyone would have the same or greater utility, and one individual would have strictly greater utility. However, the condition we placed on any welfare function is that it must be an increasing function of individual utilities, meaning that aggregate welfare should increase as the utilities of individual participants increases. Hence, the new allocation would have a higher social welfare than the previous allocation, thus contradicting the fact that the previous allocation was a maximal welfare allocation. Conversely, it is also true that any Pareto optimal allocation is a welfare maximum for some welfare function [15].

In general, however, the condition of Pareto efficiency says nothing about the distribution of welfare (surplus) amongst the market participants. Thus, the notion of Pareto optimality ignores questions of fairness and how aggregate surplus might be apportioned amongst people. Pareto optimality is a desirable goal in and of itself. However, in general, there can be many Pareto optimal allocations corresponding to a market scenario and, as such, we must define criteria to choose amongst them.

---

29 This follows directly from the definition of Pareto optimality presented earlier.
30 A proof of this converse claim is left to the reader as an exercise. The reader is invited to refer to [15], Chapter 31 for additional details.
One way to obtain the "social utility" corresponding to a market allocation is simply by numerically adding individual utilities corresponding to that allocation. That is, we say that allocation $A$ is socially preferred to allocation $B$ if

$$\sum_{i=1}^{n} u_i(A) > \sum_{i=1}^{n} u_i(B)$$

Equation 11

where $n$ is the number of market participants. Summing the individual utilities, rather than multiplying them, or subjecting them to some other monotonic transformation before aggregating, is a completely arbitrary choice. The only restrictions that a "social welfare function" needs to follow is that it should depend only on individual preferences, and it should be an increasing function of each individual's utility – meaning that if all individuals prefer allocation $A$ to allocation $B$, then the social welfare function will also rank $A$ higher than $B$. As such, there are several social welfare function that we can use 15:

1. A simple "sum of utilities" welfare function:

$$W(u_1, \ldots, u_n) = \sum_{i=1}^{n} u_i$$

Equation 12

2. A "classical utilitarian" or Benthamite welfare function, generalized as a weighted-sum-of-utilities welfare function. Classic utilitarianism considers the highest good to be the maximum utility for the maximum number of people:

$$W(u_1, \ldots, u_n) = \sum_{i=1}^{n} a_i u_i$$

Equation 13

In this case the weights, $a_1, \ldots, a_n$, represent how important each individual economic agent's utility in the context of overall social welfare 32.

3. A "maximum of minimum" or Rawlsian welfare function:

31 Named after Joseph Bentham, who is often called the father of the utilitarian school of moral philosophy.
32 Of course, it is not immediately intuitive as to what criteria we might use to determine these weights.
Equation 14

\[ W(u_1,\ldots,u_n) = \min \{u_1,\ldots,u_n\} \]

The Rawlsian school of moral philosophy thus contends that the social welfare of an allocation is contingent upon the welfare of the person who is worst off in the society.

Each of the above functions represent different ways in which we can aggregate individual utilities corresponding to a given allocation of resources to deduce a societal utility metric. In general, in the context of matching buyers and sellers in our system, we would like to find a feasible allocation that simultaneously also maximizes social welfare. However, these welfare functions view individual utilities as defined over entire allocations rather than over each person’s individual, post-transaction, allocation of goods. As a consequence, within MARI, we measure social welfare by adopting a special form of the welfare function, known as the “individualistic” or Bergson-Samuelson welfare function [15]. This formulation says, that if we let \( x_i \) denote individual \( i \)’s consumption bundle, and \( u_i(x_i) \) as the individual \( i \)’s utility corresponding to the bundle \( x_i \), then the social welfare function can be written as:

\[ W = W(u_1(x_1),\ldots,u_n(x_n)) . \]

Equation 15

5.5 Algorithms for Matching Buyers with Sellers

While we have adopted a graph-based min-cost flow algorithm that uses a heuristic of aggregate surplus maximization to match buyers and sellers within our system, other techniques were also considered. In general, “matching” problems may be viewed as a subclass of combinatorial optimization problems in general, and of network flow problems in particular [36].

In general, there are three common versions of bipartite matching problems [36]:

---

33 Named after the moral philosopher John Rawls.
1. The Cardinality Problem: We wish to find a matching containing the maximum number of arcs. This corresponds to a situation in which we simply wish to maximize the number of market participants who have a transaction partner, without regard to the costs associated with the matching.

2. The Stable Marriage Problem: There is no objective function, per se, that we wish to maximize. Rather, this models a situation in which each seller has a ranking of each buyer and each buyer has a ranking of each seller. Then we seek a feasible matching of the members of the two groups, with the property that no pair of buyer and seller prefer each other to the partners they are assigned in the stable matching.

   We chose not to use a cardinality problem formulation, since this does not allow us to make any claims about Pareto efficiency or surplus maximization, which we had perceived as being desirable properties.

   The algorithm for the stable marriage problem [36] is an iterative greedy algorithm. Each buyer “proposes” to his or her most preferred seller, and each seller who receives more than one proposal rejects all except her most preferred buyer amongst all buyers who proposed to her. In general, there can be several stable matchings that correspond to a given set of rankings. The matching constructed has several interesting properties. In particular, it can be shown that every buyer is at least as well off under it as under any stable matching [36]. Hence, such a matching can be referred to as a “buyer-optimal matching.”

   Intuitively, since buyers propose to sellers in decreasing order of their preferences, and since no seller ever rejects a stable partner, each buyer is necessarily paired with the best possible stable seller. This implies that if each buyer is independently given her best stable partner, the result is a stable matching. However, this optimality from the buyer’s point of view is obtained at the expense of the sellers. In fact, it is possible to show that in a “buyer-optimal matching,” each seller obtains the worst partner that she can have in any stable matching [36].

---

34 The treatment of the formulation with sellers “proposing” to buyers is symmetrical.
35 Symmetrically, in the converse formulation where sellers propose to buyers, we would have a “seller-optimal” matching.
36 The fact that a seller never rejects a stable partner is an interesting consequence of this formulation of the stable matching algorithm. The reader is invited to refer to [36], p.474 for a formal proof of a similar claim.
Thus we see, that if we wish to optimize strictly from the buyer's perspective alone or from the seller's perspective alone, then the stable-marriage algorithm would be a good choice. However, in the current incarnation of our system, we want to take economic welfare considerations into account and optimize from a global perspective.
6 FUNCTIONAL DESIGN

6.1 Functional Schema and Modules

The MARI system consists of several major functional components. Figure 9 illustrates the major modules and the inter-relationships between them. Subsequently, we discuss each major component and its role.

Figure 9: Architecture Design Schematic
1). **User Interface Manager (UIM):** Controls the (HTML) interface that is presented to the user. The UIM allows the user to specify and initiate a buy or sell request, to examine the status of previous requests, and to view market status and history. The Valuation Managers, discussed below, are invoked by the UIM during the course of initializing a user request to buy or sell. The UIM ensures that all relevant parameters are collected from a user in the context of any request, and that only valid requests are propagated.

2). **Buyer Valuation Manager (BVM):** Gathers sufficient information from a potential buyer so as to be able to accurately infer the buyer's valuation for previously unseen products.

2.1). **Valuation Function Generalizer (VFG):** Models a buyer's utility for multiple attributes by allowing the market maker to select from generic, pre-defined mathematical functions.

2.2). **Valuation Function Trainer (VFT):** Fine-tunes the rough utility function captured by the VFG by giving the market maker the option of iteratively “training” the system. Essentially, the VFT facilitates the construction of an (iteratively refined) piecewise, linear approximation of the buyer’s utility function.

3). **Seller Valuation Manager (SVM):** Gathers information from a potential seller so as to be able to accurately infer the seller’s valuation for previously unseen products. Like the BVM, the SVM works by having the market maker initialize the VFG and VFT.

4). **Market Cycle Manager (MCM):** Manages and enforces market cycles. The frequency of market cycles is a system variable that must be pre-specified, or can be set to be triggered by the simultaneous presence of certain (pre-specified) environmental conditions (such as number of users currently waiting to be matched).

4.1). **Optimization Heuristic Manager (OHM):** Allows the specification of which optimization heuristic ought to be employed. For instance, in some cases, the market maker's aim might be to maximize the number of buyers and sellers matched, while in other cases one may want to maximize the minimum surplus amongst all transaction
partners, etc. The OHM enforces the specified optimization heuristic throughout the system by setting key global parameters appropriately, in a mutually consistent fashion. These parameters can then be referenced by other modules in the process of optimally pairing transaction partners.

The MCM invokes the Match Maker (MM) at the start of every “market cycle” (a “market cycle” is defined as the period between consecutive runs of the matching algorithm). The MCM also ensures that buying and selling requests are time-stamped and queued appropriately so that precedence and priority relationships can be established if needed. At the end of every market cycle, the MCM examines the results output by the MM and notifies the User Status Manager (USM) and User Notification Manager (UNM), so as to update the status of user requests, as appropriate.

5). **Match Maker (MM):** Invoked by the MCM, the MM optimally pairs up buyers and sellers. The exact optimization heuristic to be used by the MM is specified by the MCM at the time of invocation. The MM gets the IDs of “active” users to be included in the matchmaking process from the User Status Manager (USM).

6). **User Status Manager (USM):** The USM monitors the status of each user. The USM keeps track of which requests are “active” and ought to be included in the matchmaking process in any given market cycle. For instance, the market maker may require that any given user should remain active for at least n market cycles, or, perhaps, that a given user should remain active until she is involved in market cycles with an aggregate of at least m other users, at least m₁ of whom are sellers and m₂ are buyers.

7). **User Notification Manager (UNM):** The UNM notifies a given user of the outcome of their buying or selling request once a definitive outcome has been established or the time permitted by the user has expired. “Notification” can be “active” (sending an e-mail to the user) or “passive” (writing the outcome to a local database which is queried when the user logs in to check the status of his or her request).
8). Database Manager (DM): The DM presents each of the above components with an interface to a back-end database, thus abstracting away the specific details by which data is stored and retrieved from the rest of the system.

9). Active User List (AUL): At the beginning of each market cycle, the Match Maker (MM) reads the AUL to identify “active” buyers and sellers who need to be matched. The User Status Manager (USM) updates the AUL when a new user enters the system and at the end of each market cycle.

10). Transaction Partner List (TPL): The TPL is a list of transaction partners as determined by the Match Maker (MM) at the termination of the most recent market cycle. The AUL, TPL, and Buyer and Seller Pools effectively comprise a system “log” that capture the state of the system at the end of a Market Cycle. In the event of a system failure, we revert back to the last logged state.

6.2 Usage Schema and Interaction Model

Referring to the major modules highlighted in Figure 9, we designed the system to support the following usage schema:

Step 1: A buyer or seller client connects to the HTML front end of our system via the Internet (HTTP), and specifies which product he or she wishes to buy/sell.

Step 2: The UIM queries the DM for the appropriate product ontology and asks user to specify required parameters via the HTML interface. Subsequently, these parameters are written back to persistent storage through the DM.

Step 3: The UIM invokes the BVM/SVM as appropriate.

Step 4: The BVM/SVM invokes the VFG, which collects relevant valuation parameters from the user and writes results back to persistent storage.
Step 5: The BVM/SVM invokes the VFT, which collects relevant valuation parameters from the user and writes results back to persistent storage.

Step 6: Once the BVM/SVM has finished, the UIM notifies the USM that a new user has successfully entered the system. The USM “activates” the user’s “profile” uniquely prefixed by the user’s MARI ID number, and updates the “Active” User List (AUL), and Buyer/Seller Pool.

Step 7: The MCM initiates a new Market Cycle when some pre-defined event or set of events occur. Such an event could be something as simple as “passage of x minutes of time,” or something more elaborate, such as the simultaneous presence of certain global environmental conditions (for instance, such a condition might be that there be at least $y_1$ active buyers present and $y_2$ active sellers present).

Step 8: The MCM invokes the OHM and indicates which optimization heuristic is to be used in pairing up buyers and sellers. Subsequently, the OHM sets all global parameters appropriately, in a mutually consistent manner, so that the system will indeed optimize according to the chosen heuristic. For instance, in some cases our aim might be to maximize the number of buyers and sellers matched, while in other cases we might want to maximize the minimum surplus amongst all transaction partners (Rawlsian approach) etc.

Step 9: The OHM invokes the MM to actually generate buyer-seller pairs.

Step 10: The MM reads the AUL to identify “active” buyers and sellers.

Step 11: The MM runs a pre-defined algorithm, specified as a global variable by the OHM, and writes the Transaction Partner List (TPL).
Step 12: The USM reads the TPL. In accordance with pre-defined system variables, the USM decides which users should continue to remain “active,” and which should be considered “matched optimally.” For instance, we may require that any given user should remain active for at least $n$ market cycles, or, perhaps, that a given user should remain active until he or she is involved in market cycles with an aggregate of at least $m$ other users, at least $m_1$ of whom are sellers and $m_2$ are buyers.

Step 13: The USM updates the AUL, writes finalized transaction pairs to persistent storage through the DM, The AUL, TPL, and Buyer and Seller Pools associated with the USM effectively comprise a system “log” that capture the state of the system at the end of a Market Cycle. In the event of a system failure, we revert back to the last logged state.

Step 14: The USM asks the UNM to send notification to users whose matchings have been finalized.

Step 15: The UNM notifies users of the outcome of their requests via email over TCP/IP.
7 IMPLEMENTATION: USER EXPERIENCE

7.1 Introduction

MARI is a web application that allows users to perform complex tasks involving buying and selling of goods and services in an online marketplace. These tasks include creating customizable agents that find appropriate transaction partners based on the user’s preferences, editing these agents, viewing current market activity, and viewing market history. The user interface for MARI was designed to enable users to perform these tasks easily and effectively. Attention was given to presentational elements and aesthetics as well to create a more enjoyable user experience.

7.2 Overview

The user interface for MARI consists of HTML pages that capture all the various states that a user can be in when using MARI. These include pages for logging in, registering, creating agents, editing agents, viewing market activity, and viewing market history, as well as success pages that provide confirmation that a task has been completed successfully, and error pages that inform the user that an error has occurred. Embedded stylesheets containing information about colors, fonts, and positioning provide a basic template for these pages. Browser compatibility issues were taken into account when creating these pages, and the system is intended for use on Internet Explorer 4.0 and above, and Netscape Navigator 4.0 and above. Figure 10 schematically illustrates various front-end states that users may experience when using MARI.

The user interface was built to be integrated with a back-end component implemented as a collection of Active Server Pages (ASP) scripted in JScript. The back-end provides the functionality for matching up prospective buyers and sellers with appropriate transaction partners. The back-end also provides the functionality for carrying out all the user tasks outlined above. The data for the system is managed using a relational database, where it is accessed and updated as users perform their various tasks.
The entire system is running on Microsoft Internet Information Server (IIS) and Microsoft SQL Server.

![Flowchart](image)

**Figure 10: Flowchart Schematically Illustrating Major MARI Front-End Modules**

The integrated pages contain both HTML and ASP code. The ASP code is interpreted in the script engine of the server, which then carries out the specified instructions. These instructions can consist of querying the database for information and generating and displaying a dynamic HTML page with the information. When the user submits information through a form, that information is processed by the ASP specified as the action of the form. The information can then be passed into functions, used to perform calculations, inserted into the database, or used to make other changes in the database. Figure 11 graphically illustrates the various back-end components and the inter-relationships between these.
Figure 11: Flowchart Illustrating Major MARI Back-End Modules
7.3 Major User-Driven Functional Operations

To use MARL, the user must first either login or register as a new user. The user only needs to enter a preferred username and password to register. Once the user has logged in, he has the option of doing one of the following: Create a new agent, view existing agents, view the current market status, view the market history, or log out. These five options appear as links on every page the user visits if he is logged in.

7.3.1 Creating a New Agent

The user can choose to create a new buying agent or create a new selling agent.

a) Creating a New Buying Agent

There are five steps to creating a new buying agent.

*Step 1: Choosing a buyer profile.* Each profile has pre-associated utility functions that define how preference changes as each attribute is varied over its permissible range. The buyer has the option of changing these values in later steps, but may also leave them unchanged if he wants to. Within the context of our language translation market prototype, there are two buyer profiles, the *professional buyer* profile and the *casual buyer* profile (see Figure 12). The *professional buyer* profile represents high time sensitivity, high seller reputation sensitivity, and high seller expertise sensitivity. The *casual buyer* profile represents low time sensitivity, low seller reputation sensitivity, and low seller expertise sensitivity.
Step 2: Selecting a translation service. The buyer selects the translation service he is interested in buying by choosing the language to translate from and the language to translate to. Currently, the available languages are Spanish, French, German, Italian, Dutch, Russian, and English.

Step 3: Specifying preferred values (see Figure 13). The buyer specifies his preferred values for each of the following attributes: Number of words, seller reputation, seller expertise, and task completion time. Seller reputation can range from 0 (worst) to 10 (best), seller expertise can range from 0 (worst) to 5 (best), and task completion time can range up from 10 minutes. The buyer also specifies his maximum bid for a seller that matches all the preferred values in the ideal “offer” configuration.
**Step 4: Specifying permissible ranges** (see Figure 13). The buyer specifies permissible ranges for flexible attributes in this step. Seller reputation, seller expertise, and task completion time are flexible attributes; number of words is not a flexible attribute. The preferred value specified in Step 3 must be within the permissible range. The buyer also specifies bid ranges in this step. This is accomplished by asking the buyer a series of questions. The buyer is asked to associate a bid value with “bundles” of attributes such that, in each bundle, a particular flexible attribute is held at the lowest or highest endpoint of its permissible range, while all other attributes are held constant at their optimal values. This, in effect, allows the buyer to valuate differently qualified sellers, and the information is used to generate a mathematical approximation of the buyer’s utility function corresponding to each flexible attribute.

<table>
<thead>
<tr>
<th>Flexible Attribute</th>
<th>Possible Range</th>
<th>Your Preferred Value</th>
<th>Permissible Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seller Reputation</td>
<td>From 0 (worst) to 10 (best)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Seller Expertise</td>
<td>From 0 (worst) to 5 (best)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Task Completion Time</td>
<td>From 10 minutes to 100 minutes</td>
<td>40 minutes</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 13: Specifying Permissible Ranges for Flexible Attributes*
Step 5: Selecting utility functions. Each of the flexible attributes has a utility function associated with it that defines how the buyer’s preference changes as the attribute varies over the permissible range specified in Step 4 (see Figure 14). The utility functions are preset (by the market maker) to the values associated with the buyer profile selected in Step 1. The buyer may leave these values unchanged, or he may change them to match his preferences more accurately.

Figure 14: Selecting Utility Functions

Finishing up. At the end of step 5, the buyer may click the “Create Agent” button to finish creating the new buying agent. He is then taken to a confirmation page that informs him that the agent has been successfully created. This page also displays the
time that the next market cycle will be run, when his results (i.e. information about a suitable seller) will be emailed to him.

**b) Creating a New Selling Agent**

There are five steps to creating a new selling agent.

*Step 1: Choosing a seller profile.* Each profile has pre-associated utility functions that define how preference changes as each attribute is varied over its permissible range. The seller has the option of changing these values in later steps, but may also leave them unchanged if he wants to. There are two seller profiles, the *professional translator* profile and the *casual translator* profile. The *professional translator* profile represents low time sensitivity, moderate (linear) number of words sensitivity, and high buyer reputation sensitivity. The *casual translator* profile represents high time sensitivity, high number of words sensitivity, and low buyer reputation sensitivity.

*Step 2: Selecting a translation service.* The seller selects the translation service he wants to sell by choosing the language to translate from and the language to translate to.

*Step 3: Specifying preferred values.* The seller specifies his preferred values for each of the following values: Number of words, buyer reputation, and task completion time. Buyer reputation can range from 0 (worst) to 10 (best), and task completion time can range from 10 minutes to 100 minutes. The seller also specifies his minimum ask price for a buyer that matches all the preferred values.

*Step 4: Specifying permissible ranges.* The seller specifies permissible ranges for flexible attributes in this step. Number of words, buyer reputation, and task completion time are all flexible attributes. The preferred value specified in Step 3 must be within the permissible range. The seller also specifies ask price ranges in this step. This is accomplished by asking the seller a series of questions. The seller is asked to associate a ask value with "bundles" of attributes such that, in each bundle, a particular flexible attribute is held at the lowest or highest endpoint of its permissible range, while all other
attributes are held constant at their optimal values. This, in effect, allows the seller to valuate differently qualified buyers, and the information is used to generate a mathematical approximation of the seller’s utility function corresponding to each flexible attribute.

**Step 5: Selecting utility functions.** Each of the flexible attributes has a utility function associated with it that defines how the seller’s preference changes as the attribute varies over the permissible range specified in Step 4. The utility functions are preset (by the market maker) to the values associated with the seller profile selected in Step 1. The seller may leave these values unchanged, or he may change them to match his preferences more accurately.

**Finishing up.** At the end of step 5, the seller may click the “Create Agent” button to finish creating the new selling agent. He is then taken to a confirmation page that informs him that the agent has been successfully created. This page also displays the time that the next market cycle will be run, when his results (i.e. information about a suitable buyer) will be emailed to him.

### 7.3.2 Viewing Existing Agents

The user can choose to view, edit or delete his active agents, view his matched agents and review or confirm “pending” transactions, view his expired agents, view all active agents, or view all expired agents.

#### a) Viewing, Deleting and Editing Active Agents

The user is given a list of all his currently active buying and selling agents and some information (such as source and target translation languages) about what each agent is trying to buy or sell, as well as the time at which the agent was created. He can choose to view the details, delete or edit any of those agents by clicking on it. He is then taken to a page that displays the information for that agent (see Figure 15). If the user has never edited the agent before, the values specified when creating the agent are displayed. If the
user has edited the agent before, the most recent changes that have been made will show up in the values that are displayed.

This information is divided up into five sections corresponding to the five steps involved in creating an agent. The five sections are: buyer or seller profile selection, translation languages, preferred values for attributes (offer), permissible ranges of flexible attributes,
and utility functions corresponding to flexible attributes. At the end of each section is a link that allows the user to edit the values for that particular section. This link takes the user to a page similar to the page for the corresponding step when creating the agent. The current values for the fields in that section are pre-selected in the form and the user can then make any changes he wants to those values. When he is finished, he may click "Submit" to update the agent with his changes. He is then taken to a confirmation page that informs him that his changes have been successfully made. From there, he may choose to make more changes to the same agent, edit another agent, or delete an active agent.

Example: Editing the preferred values for a selling agent. The information page for the agent will display the five sections corresponding to the five steps to creating a selling agent. The third section, titled "Preferred Values," lists the current values for the following information: Number of words, buyer reputation, task completion time, and ask price. The user then clicks "Edit your preferred values" to make his changes. He is taken to a page with the same form he filled out in Step 3 when creating this agent, except that his current preferred values for number of words, buyer reputation, and task completion time are already filled out in the form. He can then edit these values to his liking and click "Submit" to update the agent with these changes.

b) Viewing Matched Agents
Matched agents comprise the subset of active agents that are involved in "pending" (mutually unconfirmed) transactions. The user has the option of viewing the details of a matched agent or deleting it. Additionally, the user can review and (subsequently) confirm the matching that the system has selected for the matched agent (see Figure 16).

When a user clicks on the "Review/Confirm Transaction" link adjacent to each matched agent listing, he is taken to another page which shows the details of the "deal" that the system has selected for the agent. "Transaction details" consist of a listing of relevant attributes and their values corresponding to the "Optimal Deal Configuration" (OTC) that the agent has been matched with (see Figure 17). In addition, when "reviewing" a matched agent, the user not only sees the details of the "deal" associated
with his matched agent, but also sees details of “potential” deals with other qualified transaction partners that were considered and, further, is able to see whether these prospective transaction partners were matched with someone else or not (see Figure 18). Each deal shown to the user has a score metric associated with it. The score metric printed with each deal reflects the surplus associated with that potential pairing. The idea behind showing the user this information is to follow the general user-interface heuristic that the user be given some contextual information that helps him better understand why the system made the choice it did, and what other choices were taken into consideration. The user can use this information to infer why his agent was not matched with other qualified transaction partners and, via the score metric, can objectively situate the deal chosen by the system in relation to other possibilities.

The user has the option of reviewing and confirming the deal associated with the matched agent or doing nothing. Upon mutual confirmation by both parties involved in a “pending” transaction, the transaction is said to become “closed,” and the agents involved are “Expired” from the market. When “reviewing” a matched agent, the owner is able to see whether the other transaction partner involved in the matching has already “confirmed” or unilaterally accepted. The presence of a unilateral confirmation by the other side makes it clear, for instance, that the transaction will consummate (become “closed”) by mutual confirmation if the owner confirms as well.
MATCHED AGENTS

1. Agent is trying to **buy** a Translation.
   Language_From : English
   Language_To : German
   Desired optimal price: $100
   Agent has been matched with a seller at a Price = $82.68.
   This agent was created on **Sat May 5 17:14:15 EDT 2001**.
   This agent was matched on **Sat May 5 17:19:45 EDT 2001**.
   View/Delete the agent
   Review/Confirm this transaction

2. Agent is trying to **buy** a Translation.
   Language_From : English
   Language_To : German
   Desired optimal price: $70
   Agent has been matched with a seller at a Price = $57.75.
   This agent was created on **Sat May 5 17:15:39 EDT 2001**.
   This agent was matched on **Sat May 5 17:19:49 EDT 2001**.
   View/Delete the agent
   Review/Confirm this transaction

3. Agent is trying to **sell** a Translation.
   Language_From : English
   Language_To : German
   Desired optimal price: $60
   Agent has been matched with a buyer at a Price = $57.75.
   This agent was created on **Sat May 5 17:19:09 EDT 2001**.
   This agent was matched on **Sat May 5 17:19:49 EDT 2001**.
   View/Delete the agent
   Review/Confirm this transaction

---

**Figure 16: Viewing Matched Agents**

---

You have been pre-matched. Please select one of the deals below to confirm the deal from your side. You will be sent an email as soon as both sides accept the deal.

1. TRANSACTION_ID={F0708B19-284D-4CD9-8010-1BE795263AE7}
   SELLER_REQUEST_ID : {E6958ED8-6DEA-4088-8E01-84DD06071CA2}
   Score: 251.00
   Closing Price: $398.75

   **Transaction Details:**
   Buyer_Reputation =10;
   Language_From :English
   Language_To :Spanish
   Number_of_Words =5000;
   Seller_Reputation =10;
   Seller_Skill =5;
   Time_to_Completion =45;

   Confirm the match

---

**Figure 17: Viewing Details of the “Deal” the System has Selected for a Matched Agent**
c) Viewing Expired Agents

This page displays the user’s expired agents (agents involved in “closed” or mutually confirmed transactions), the translation languages corresponding to each, the price at which the service was sold, and the date and time the agent was created. The user cannot edit his expired agents, but has the option of being able to “reactivate” these, in which case the agent’s status shifts from “Expired” to “Active,” and the agent becomes eligible to be matched in future market cycles.

d) Viewing All Active Agents

This page displays all active agents in the marketplace at the current time, the translation languages corresponding to each, the bid or ask price of the translation service the agent is trying to buy or sell, and the date and time the agent was created.
e) **Viewing All Expired Agents**

This page displays all expired agents in the marketplace, the translation languages corresponding to each, the price at which the transaction was closed, and the date and time the agent was created.

### 7.3.3 Viewing Market Status

The user can view the current marketplace activity on this page. Information that is provided includes the current number of active buying and selling agents, the number of matched buying and selling agents (active agents involved in mutually unconfirmed, “pending” transactions), and the number of expired buying and selling agents (agents involved in mutually confirmed, “closed” transactions).

The user is shown a listing of active agents with information about what each agent is trying to buy or sell and at what price. The user can optionally view more details about any individual agent to see ranges and utility functions corresponding to various attributes. Additionally, the system differentiates between active agents that have already been matched by the system and those that are currently unmatched.

Finally, the market status page also lists all pending transactions in the market, giving details of the transaction including the final attribute values for all relevant attributes associated with the deal, the final closing price, and links to detailed views of the buyer and seller agents involved.

The purpose of the Market Status page is to give a user a complete and informative snapshot of the market. Not only can a user gauge what prospective transaction partners might be available for an agent he might create, but can also implicitly receive signals about prevailing market prices, and competition his agent can expect to face from other active agents.

### 7.3.4 Viewing Market History

The user can view the transaction history of the market on this page. A list of the last ten market cycles (exact number is configurable by the MarketMaker) is displayed, along with the date and time at which each cycle was completed. The user can click on any of
these cycles to pop up a window which lists all "closed" transactions (transactions that have been mutually confirmed by the buyer and seller involved in the matching) (see Figure 19).

The detail page for each cycle displays the date and time at which it was completed and the number of matches that were made during the cycle. For each transaction, we show the details of the transaction including the final attribute values for all relevant attributes associated with the deal, as well as the closing price at which the transaction was consummated.

The purpose of the Market History page is to give the user a cumulative historical snapshot of the market. The user can thereby infer market trends, as well as gain insight into characteristics of deals that were consummated by both of the involved transaction partners. This information can help the user set realistic expectations of what his agents can hope to accomplish, and tailor his agents so as to maximize chances of eventually reaching a consummated transaction, that would be appealing from the perspective of both parties involved in a potential pairing.
7.3.5 Logging Out

The user can log out at any time while using MARI. If he is in the middle of creating or editing an agent, the process will not be completed and there will be no record of it.

7.4 Usability Considerations

The MARI user interface was designed so that it would be easy to learn, easy to understand, and easy to use. The user tasks themselves are not very complex, but the concepts behind the functionality of the system may be more difficult to understand for those that have no experience with using software agents. Therefore, care was given to making it as easy as possible to understand the system as well as making it as easy as possible to use the system.
As discussed earlier, there are four main functions that users can perform when using MARI. These functions are: creating new agents, viewing and editing existing agents, viewing current market activity, and viewing market history. Links to performing each of these tasks are displayed prominently at the top of each page the user visits after logging in. A fifth link to log out of the system is also displayed with the other links so that the user can quit at any time.

When creating and editing agents, each step is clearly labeled and a detailed explanation of what the user is expected to do in that step is given. Detailed explanations are also given when the user is dealing with a concept that may be difficult to understand. For example, when choosing a buyer or seller profile, a description of each profile is given so that users may choose the one that is more appropriate for them (see Figure 12). The user is also informed that the pre-associated utility functions for each profile can be changed later on so that he will not feel uncomfortable selecting a profile even if he is not sure it is the right one for him. The selection of a buyer or seller profile is an important usability consideration in itself because it allows inexperienced users who will have trouble selecting utility functions to skip this step altogether.

For sections with forms that have more than one field to fill out, each field, with its own instructions, is clearly separated from those around it with a horizontal line, making it easier to distinguish between them and enabling users to enter their information more quickly. Examples of these sections are specifying preferred values and permissible ranges when creating a new agent (see Figure 13).

The most complex user task in MARI is editing agents. A possible way to create the pages for editing agents was to make them exactly the same as those for creating agents. However, users may not want to change something in every section, and for these users, it would be inefficient to go through all five steps when they only want to make a change in one section. It is also inefficient for users who want to edit, for example, Steps 2 and 5, of the ‘create agent’ sequence, to have to go through all the steps, 1 through 5, sequentially. Therefore, the information page (see Figure 15) for editing an agent allows the user to choose the section they want to edit. It also displays the current values for each section so that the user can find the field they want to edit quickly. When the user
has finished editing a section, he is taken to a page that gives him the option of editing another section easily.

Pop-up windows are used when it is more convenient for the user to not leave the current page when viewing the information in the new page. When the user selects utility functions for creating or editing an agent, it is more convenient to be able to view the graphs for the utility functions and make their selections at the same time (see Figure 14). When viewing market history, it is also more convenient to view information for more than one market cycle at the same time if the user wants to make comparisons.

7.5 Conclusions

Although we are reasonably satisfied with the current state of the MARI interface, we can envision a number of changes that may be made in the future, so as to enhance usability. The addition of more explanations to help users understand the system would be especially important if MARI was to be presented to a wider audience in the future. Users with limited technical knowledge may have difficulty when creating an agent if they do not fully understand how their choices, especially when selecting utility functions, will affect the results. The option of selecting a buyer or seller profile at the start of the agent-creating process so that inexperienced users are not required to select utility functions themselves mitigates this problem, but users will still need to know how to select utility functions themselves to improve their chances of being matched up with a transaction partner they are happy with in different situations. It would also be helpful to view some sample agents with sample results so that they could learn more about how each aspect of an agent affects the end results. The pages for viewing market history provide the results of matchings, but not the details of the buying and selling agents involved. Users also need to be provided with more information on what their choices mean, for example, exactly what a seller of an expertise level of 7 implies.

Some of the choices made for the “look and feel” of the user interface also interferes with its usability. The biggest problem is the readability of white text on a red background, especially for users that have weak vision or are colorblind. Another problem is the fonts used for the heading and caption on every page, which may be
difficult to read as well. There may also be some confusion caused by the doubling of the links on the navigation bar, because both the text links and the smaller boxes underneath them link to the same pages.

Overall, the user interface design for MARI is satisfactory for the time being because the intended audience consists mainly of those that already have knowledge of software agents and are more interested in the functionality of the system than its usability. In the future, however, some changes could be made for the system to be successful with a wider, less-specialized, audience.
8 EVALUATION: MARKET SCENARIOS

8.1 Market Variables and Possible Scenarios

In order to illustrate the system’s functionality we simulated a few carefully chosen market scenarios. We identified the following underlying variables to be of interest in selecting suitable scenarios:

1. Market Participants: Market segmentability or homogeneity of participants determines the extent to which buyers and sellers are homogenous within their respective groups -- i.e. the extent to which each seller is similar to other sellers and each buyer is similar to other buyers, both in terms of their preferences for the transaction partner, as well as their own attributes. This, in turn, determines the extent to which buyers and sellers can be segmented into easily differentiable, distinct groups.

2. Market Participant Preferences: Correlated versus uncorrelated price points and valuations determine the extent to which buyers or sellers, within their respective groups, generate independent and uncorrelated bids or asks. Valuations can be said to be uncorrelated (or “private”) when, for instance, two different buyers with similar underlying preferences bid differently, in absolute monetary terms, for a seller who is equally “good” in an objective sense. Valuations can be said to be correlated (or “public”) when the two buyers somehow are able to peg the absolute level of their bids, perhaps using market signals such as ‘Market Status’ or ‘Market History’ that would provide cues as to how the seller, or similar sellers, are valued by the market at large.

In the vast majority of cases, goods and services lend themselves to public valuation, and there are generally sufficient market signals present so that buyers and sellers are able to infer the prevalent market price. Moreover, in our system, the purpose behind creating Market History and Market Status pages is precisely so that buyers and sellers will be able to gain access to such market signals. As such, even though the case of uncorrelated or private valuations is interesting in certain niche cases, such as trading...
of authentic art pieces, we will mostly limit our analysis to the more general case of correlated valuations.

3. Macro-market Factors: Market power determines the extent to which a given buyer or a seller might be able to position himself as a dominant player. For the purposes of our system, a seller, for instance, can be said to have market power when he is the only one present amongst several buyers, thus giving him more implicit negotiation power in being able to select the optimal from a variety of potential deals.

Using binary values for the above variables, we can imagine eight possible market simulations:

a). Homogenous market participants with correlated price points, no market power. This scenario is not interesting because if participants are homogenous and have correlated valuations, then they will all be exactly identical. In this case matching would be trivial since any buyer could be matched with any seller.

b). Homogenous market participants with correlated price points, market power present. This scenario is not interesting for the same reason as (a).

c). Homogenous market participants with uncorrelated price points, no market power. This case is further explored in “Scenario 1” below. This is the only case in which we examine uncorrelated or private valuations, since the concept of correlated valuations is meaningless when market participants are homogenous.

d). Homogenous market participants with uncorrelated price points, market power present. This scenario is not interesting since if the participants are homogenous (i.e. all buyers are similar and all sellers are similar), and if a given seller, say, has market power, then the seller can be trivially matched with the prospective buyer whose offer bid is the highest.
e). Heterogeneous market participants with uncorrelated price points, no market power. We will not examine this case since, in this context, the case of uncorrelated or private valuations is of niche interest and not of sufficient generality.

f). Heterogeneous market participants with uncorrelated price points, market power present. We will not examine this case for the same reason as (e).

g). Heterogeneous and segmentable market participants with correlated price points, no market power. This case is further explored in “Scenario 2” below.

h). Heterogeneous market participants with correlated price points, market power present. This scenario is interesting both for the monopoly (seller has market power) (“Scenario 3” below) and monopsony (buyer has market power) (“Scenario 4” below) cases.

From the above analysis, it is clear that only cases ‘c’, ‘g’ and ‘h’ (2 subcases) ought to be simulated. These are further discussed in the following scenarios 1-4. Case ‘h’ is discussed in both scenarios 3 (seller has market power) and 4 (buyer has market power). These scenarios effectively illustrate how our system works under different market conditions.

**8.2 First Scenario**

*Case ‘c’:* Homogenous market participants with uncorrelated price points, no market power.

*Overview:* Three Buyers (see Table 7, Table 8, and Table 9) and two Sellers (see Table 10 and Table 11), in the market for French to German Translations. The buyers and sellers are identical except for their offer bid or ask values. All the buyer agents are qualified for all the seller agents, and vice versa. This scenario illustrates how our system maximizes welfare and achieves Pareto optimality.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Skill</td>
<td>3</td>
<td>2-5</td>
<td>UF₉</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>5</td>
<td>4-10</td>
<td>UF₉</td>
<td>N/A</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>40</td>
<td>30-50</td>
<td>UF₆</td>
<td>$430-$400</td>
</tr>
</tbody>
</table>

Table 7: Scenario 1, Buyer 1 (B1)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Skill</td>
<td>3</td>
<td>2-5</td>
<td>UF₉</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>5</td>
<td>4-10</td>
<td>UF₉</td>
<td>N/A</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>40</td>
<td>30-50</td>
<td>UF₆</td>
<td>$360-$330</td>
</tr>
</tbody>
</table>

Table 8: Scenario 1, Buyer 2 (B2)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Skill</td>
<td>3</td>
<td>2-5</td>
<td>UF₉</td>
<td>N/A</td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>5</td>
<td>4-10</td>
<td>UF₉</td>
<td>N/A</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>40</td>
<td>30-50</td>
<td>UF₆</td>
<td>$460-$440</td>
</tr>
</tbody>
</table>

Table 9: Scenario 1, Buyer 2 (B2)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Ask Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Buyer</td>
<td>5</td>
<td>4-10</td>
<td>UF₉</td>
<td>$260-$230</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>40</td>
<td>30-50</td>
<td>UF₆</td>
<td>$260-$230</td>
</tr>
</tbody>
</table>

Table 10: Scenario 1, Seller 1 (S1)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Ask Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Buyer</td>
<td>5</td>
<td>4-10</td>
<td>UF₉</td>
<td>$260-$230</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>40</td>
<td>30-50</td>
<td>UF₆</td>
<td>$260-$230</td>
</tr>
</tbody>
</table>

Table 11: Scenario 1, Seller 2 (S2)
Results: Based upon the structure of our examples, we can precisely determine the theoretical surplus that would be generated between any potential buyer seller pairing. These computed surplus values are shown in Figure 20. MARI, based upon an approximation of user preferences, generates numbers that are remarkably similar (see Figure 21).

Using the heuristic of surplus maximization, MARI determines that S1 ought to be matched with B1, S2 should be matched with B3, and B2 should be left unmatched. The "closing price" corresponding to each potential transaction is shown in
Table 12. By inspection, it is clear that the (S1-B1, S2-B3) matching output by the system is Pareto optimal.

<table>
<thead>
<tr>
<th>Pairing</th>
<th>Closing Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1-B1</td>
<td>$344.70</td>
</tr>
<tr>
<td>S1-B2</td>
<td>$308.62</td>
</tr>
<tr>
<td>S1-B3</td>
<td>$361.08</td>
</tr>
<tr>
<td>S2-B1</td>
<td>$370.41</td>
</tr>
<tr>
<td>S2-B2</td>
<td>$334.39</td>
</tr>
<tr>
<td>S2-B3</td>
<td>$386.85</td>
</tr>
</tbody>
</table>

Table 12: System Generated Closing Prices Corresponding to Each Potential Pairing

Within our system, the above information is visible to each agent owner’s when they log in to “review or confirm” potential transactions. For instance, when S2’s owner logs in, he sees the details of the “deal” associated with his matched agent (see Figure 22). In addition, he sees details of potential deals, including closing price, with other qualified transaction partners that were considered (i.e. deals with B1 and B2) and, further, is able to see whether these prospective transaction partners were matched with someone else or not (see Figure 23). The score metric printed with each “deal” reflects the surplus associated with that potential pairing. Similarly, the details seen by B1’s owner are shown in Figure 24.
Figure 22: S2's owner sees details of the "deal" associated with his matched agent

Qualified Transaction Partners Not Chosen for this Matching:

BUYER_REQUEST_ID: {FC275533-CFE0-4F48-B518-E3A4E64D777B}
This Prospective Agent (BUYER) Has Been Matched With Someone Else
Score: 123.695
Closing Price: 370.47

Details:
BUYER_Reputation = 5;
Language_From: French
Language_To: German
Number_of_Words = 4000;
Seller_Reputation = 3;
Seller_Skill = 3;
Time_to_Completion = 40;

BUYER_REQUEST_ID: {E32FE309-46F5-45FA-9556-C08CA392C599}
This Prospective Agent (BUYER) Has Not Yet Been Matched With Anyone Else.
Score: 51.539
Closing Price: 334.39

Details:
BUYER_Reputation = 5;
Language_From: French
Language_To: German
Number_of_Words = 4000;
Seller_Reputation = 3;
Seller_Skill = 3;
Time_to_Completion = 40;

Figure 23: S2's owner sees details of potential deals with other qualified transaction partners that were considered and whether these prospective transaction partners were matched with someone else.
8.3 Second Scenario

Case ‘g’: Heterogeneous and segmentable market participants with correlated price points, no market power. This scenario illustrates how our system facilitates value creation via differentiation, and how scarce resources are allocated in an appropriate fashion.

Overview: Two Buyers and two Sellers, in the market for English to German Translations. The buyers and sellers are differentiated into 2 segments each: Professional Buyer (B1), Casual Buyer (B2) and Professional Seller (S1), Casual Seller (S2).
<table>
<thead>
<tr>
<th><strong>B1</strong></th>
<th><strong>Attribute</strong></th>
<th><strong>Optimal (Offer) Value</strong></th>
<th><strong>Permissible Range</strong></th>
<th><strong>Utility Function</strong></th>
<th><strong>Bid Range</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Seller Skill</td>
<td>5</td>
<td>3-5</td>
<td>UF$_3$</td>
<td>$70-$100</td>
</tr>
<tr>
<td></td>
<td>Seller Reputation</td>
<td>10</td>
<td>5-10</td>
<td>UF$_3$</td>
<td>$70-$100</td>
</tr>
<tr>
<td></td>
<td>Time (mins)</td>
<td>30</td>
<td>30-90</td>
<td>UF$_5$</td>
<td>$100-$70</td>
</tr>
</tbody>
</table>

Table 13: Scenario 2, Buyer 1 (Professional Buyer)

<table>
<thead>
<tr>
<th><strong>B2</strong></th>
<th><strong>Attribute</strong></th>
<th><strong>Optimal (Offer) Value</strong></th>
<th><strong>Permissible Range</strong></th>
<th><strong>Utility Function</strong></th>
<th><strong>Bid Range</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Seller Skill</td>
<td>3</td>
<td>3-5</td>
<td>UF$_3$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Seller Reputation</td>
<td>5</td>
<td>5-10</td>
<td>UF$_3$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Time (mins)</td>
<td>90</td>
<td>30-90</td>
<td>UF$_6$</td>
<td>$85-$70</td>
</tr>
</tbody>
</table>

Table 14: Scenario 2, Buyer 2 (Casual Buyer)

<table>
<thead>
<tr>
<th><strong>S1</strong></th>
<th><strong>Attribute</strong></th>
<th><strong>Optimal (Offer) Value</strong></th>
<th><strong>Permissible Range</strong></th>
<th><strong>Utility Function</strong></th>
<th><strong>Ask Range</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Buyer Reputation</td>
<td>10</td>
<td>5-10</td>
<td>UF$_5$</td>
<td>$100-$80</td>
</tr>
<tr>
<td></td>
<td>Time (mins)</td>
<td>30</td>
<td>30-90</td>
<td>UF$_6$</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 15: Scenario 2, Seller 1 (Professional Seller)

<table>
<thead>
<tr>
<th><strong>S2</strong></th>
<th><strong>Attribute</strong></th>
<th><strong>Optimal (Offer) Value</strong></th>
<th><strong>Permissible Range</strong></th>
<th><strong>Utility Function</strong></th>
<th><strong>Ask Range</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Buyer Reputation</td>
<td>5</td>
<td>5-10</td>
<td>UF$_3$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Time (mins)</td>
<td>90</td>
<td>30-90</td>
<td>UF$_6$</td>
<td>$90-$60</td>
</tr>
</tbody>
</table>

Table 16: Scenario 2, Seller 2 (Casual Seller)

Results: MARI determines that S1 (professional seller) ought to be matched with the B1 (professional buyer), and that S2 (casual translator) should be matched with B2 (casual buyer). The matching, as well as surplus associated with potential transactions is shown in Figure 25. The “closing price” corresponding to each potential transaction is shown in Table 17.
This scenario illustrates how MARI effectively provides a framework by which buyers and sellers may differentiate themselves to create value. The total value created by a transaction can be interpreted as being equivalent to the surplus generated by that transaction, or the difference between the buyer's willingness to pay (WTP) and the supplier's ask price. A competitive advantage, in business strategy, is typically interpreted as one that widens the gap between WTP and ask price. MARI facilitates value creation by allowing buyers and sellers to sufficiently differentiate themselves from other players in the market, and thereby create a competitive advantage.

In traditional markets, there is no mechanism by which a professional buyer (B1), for instance, can instantiate an agent with variable degrees of preference and valuation for different (professional and casual) seller segments. The buyer can merely assume that there is some mixture of the two types of sellers in the market and bid something in between his differentiated bids corresponding to different seller segments. In our scenario, for example, B1’s bids for a professional translator and a casual translator are
$100 and $70 respectively. However, if there is no mechanism by which sellers can convey their differentiated product offerings, B1 will only be willing to bid something in between $70 and $100, say $85 (halfway in between the differentiated bids). Using the same reasoning, B2’s bid will be $77.5, S1’s ask price will be $90 and S2’s ask price will be $75. In this case, B1 (professional buyer) will be undesirably matched with S2 (casual seller), leading to a circumstance akin to the classical “lemons” problem of economics, that stems from information asymmetries between buyers and sellers [48].

Figure 26 and Figure 27 illustrate how MARI facilitates greater value creation, and more appropriate allocation of scarce resources compared to traditional marketplaces.

Figure 26: Value Created in a Traditional Market
8.4 Third Scenario

Case 'h': Heterogeneous market participants with correlated price points, market power present. Seller has market power (Monopoly).

Overview: Monopoly. One Seller and three Buyers in the Market for English to Russian translations. This scenario illustrates how our system automatically identifies a suitable deal and transaction partner for a given seller from potentially many possibilities.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>B1 Value</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>5000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Seller Skill</td>
<td>5</td>
<td>3-5</td>
<td>UF₂</td>
<td>$270-$300</td>
<td></td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>10</td>
<td>5-10</td>
<td>UF₃</td>
<td>$270-$300</td>
<td></td>
</tr>
<tr>
<td>Time (mins)</td>
<td>30</td>
<td>30-90</td>
<td>UF₅</td>
<td>$300-$220</td>
<td></td>
</tr>
</tbody>
</table>

Table 19: Scenario 3, Buyer 1

<table>
<thead>
<tr>
<th>Attribute</th>
<th>B2 Value</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>7000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Seller Skill</td>
<td>3</td>
<td>3-5</td>
<td>UF₉</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>5</td>
<td>5-10</td>
<td>UF₉</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Time (mins)</td>
<td>90</td>
<td>30-90</td>
<td>UF₆</td>
<td>$450-$400</td>
<td></td>
</tr>
</tbody>
</table>

Table 20: Scenario 3, Buyer 2

<table>
<thead>
<tr>
<th>Attribute</th>
<th>B3 Value</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Seller Skill</td>
<td>4</td>
<td>4-5</td>
<td>UF₁</td>
<td>$300-$350</td>
<td></td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>8</td>
<td>8-10</td>
<td>UF₁</td>
<td>$300-$400</td>
<td></td>
</tr>
<tr>
<td>Time (mins)</td>
<td>60</td>
<td>30-60</td>
<td>UF₄</td>
<td>$400-$300</td>
<td></td>
</tr>
</tbody>
</table>

Table 21: Scenario 3, Buyer 3

Results: MARI determines that S1 ought to be matched with B3. The surplus corresponding to each potential buyer-seller pairing is shown in Figure 28. Table 22 summarizes the most important factors taken into consideration by the system in deciding upon the final pairing of S1 with B1. In particular, Table 22 illustrates how, based upon its modeling of user preference functions, the system is able to approximate what each of the buyers are willing to bid for S1 and, symmetrically, what ask price S1 is willing to take from each buyer. These bid and ask prices correspond to the myopically optimal Optimal Transaction Configuration (OTC) that the system identifies for each potential buyer-seller pair (see Figure 29 and Figure 30 for these transaction configurations).

When S1’s owner logs in, he sees the details of the “deal” associated with his matched agent (see Figure 29). In addition, he sees details of potential deals, including
closing price, with other qualified transaction partners that were considered (i.e. deals with B1 and B2) and corresponding scores (see Figure 30).

Our matching algorithm tries to find the value maximizing “deal” or optimal transaction configuration for each potential pairing. An example of this is how the time attribute is set to 45 minutes in the S1-B1 pairing (see Figure 30), a value that lies exactly midway between S1’s optimal value of 60 minutes and B1’s optimal value of 30 minutes. Overall, the system identifies the deal where the greatest value is created and where the buyer and seller stand to benefit the most in terms of the surplus they manage to capture.

![Diagram](image)

Figure 28: System Generated Surplus Values Corresponding to each Potential Match

<table>
<thead>
<tr>
<th>Pairing</th>
<th>Bid</th>
<th>Ask</th>
<th>Closing Price</th>
<th>Score (Surplus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1-B1</td>
<td>$265.00</td>
<td>$186.19</td>
<td>$225.60</td>
<td>78.810</td>
</tr>
<tr>
<td>S1-B2</td>
<td>$455.10</td>
<td>$452.62</td>
<td>$453.86</td>
<td>2.484</td>
</tr>
<tr>
<td>S1-B3</td>
<td>$456.25</td>
<td>$153.33</td>
<td>$304.79</td>
<td>302.917</td>
</tr>
</tbody>
</table>

Table 22: Details of Potential Matches for S1 Considered by the System
Figure 29: Details Seen by S1’s Owner

Figure 30: S1’s Owner Sees Details of Potential Deals with other Qualified Transaction Partners
8.5 Fourth Scenario

Case ‘h’: Heterogeneous market participants with correlated price points, market power present. Buyer has market power (Monopsony).

Overview: Monopsony. One Buyer and three Sellers in the Market for English to Spanish translations. This scenario illustrates how our system automatically identifies a suitable deal and transaction partner for a given buyer from potentially many possibilities.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Bid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reputation=10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offer_Bid=$400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>4000</td>
<td>4000-8000</td>
<td>UF$_1$</td>
<td>$400-$600</td>
</tr>
<tr>
<td>Seller Skill</td>
<td>4</td>
<td>3-5</td>
<td>UF$_3$</td>
<td>$360-$420</td>
</tr>
<tr>
<td>Seller Reputation</td>
<td>7</td>
<td>5-10</td>
<td>UF$_3$</td>
<td>$360-$420</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>60</td>
<td>30-120</td>
<td>UF$_5$</td>
<td>$450-$330</td>
</tr>
</tbody>
</table>

Table 23: Scenario 4, Buyer 1

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Ask Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reputation=10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skill=5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offer_Ask=$300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>5000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Buyer Reputation</td>
<td>10</td>
<td>5-10</td>
<td>UF$_5$</td>
<td>$375-$300</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>30</td>
<td>30-90</td>
<td>UF$_9$</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 24: Scenario 4, Seller 1

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Ask Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reputation=5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skill=3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offer_Ask=$400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>7000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Buyer Reputation</td>
<td>5</td>
<td>5-10</td>
<td>UF$_9$</td>
<td>N/A</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>90</td>
<td>30-90</td>
<td>UF$_9$</td>
<td>$600-$400</td>
</tr>
</tbody>
</table>

Table 25: Scenario 4, Seller 2
Table 26: Scenario 4, Seller 3

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optimal (Offer) Value</th>
<th>Permissible Range</th>
<th>Utility Function</th>
<th>Ask Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>4000</td>
<td>Fixed</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Buyer</td>
<td>8</td>
<td>5-10</td>
<td>UFₐ</td>
<td>N/A</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>60</td>
<td>30-90</td>
<td>UFₐ</td>
<td>$325-$250</td>
</tr>
</tbody>
</table>

Results: MARI determines that B1 ought to be matched with S1. The surplus corresponding to each potential buyer-seller pairing is shown in Figure 31. Table 27 summarizes the most important factors taken into consideration by the system in deciding upon the final pairing of B1 with S1. In particular, Table 27 illustrates how, based upon its modeling of user preference functions, the system is able to approximate what each of the buyers are willing to bid for S1 and, symmetrically, what ask price S1 is willing to take from each buyer. These bid and ask prices correspond to the myopically optimal Optimal Transaction Configuration (OTC) that the system identifies for each potential buyer-seller pair (see Figure 32 and Figure 33 for these transaction configurations).

When S1’s owner logs in, he sees the details of the “deal” associated with his matched agent (see Figure 32). In addition, he sees details of potential deals, including closing price, with other qualified transaction partners that were considered (i.e. deals with B1 and B2) and corresponding scores (see Figure 33). Overall, the system identifies the deal where the greatest value is created and where the buyer and seller stand to benefit the most in terms of the surplus captured.

Figure 31: System Generated Surplus Values Corresponding to each Potential Match
Table 27: Details of Potential Matches for B1 Considered by the System

<table>
<thead>
<tr>
<th>Pairing</th>
<th>Bid</th>
<th>Ask</th>
<th>Closing Price</th>
<th>Score (Surplus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1-S1</td>
<td>$524.29</td>
<td>$273.21</td>
<td>$398.75</td>
<td>251.08</td>
</tr>
<tr>
<td>B1-S2</td>
<td>$408.19</td>
<td>$407.99</td>
<td>$408.09</td>
<td>0.205</td>
</tr>
<tr>
<td>B1-S3</td>
<td>$441.79</td>
<td>$337.03</td>
<td>$389.41</td>
<td>104.76</td>
</tr>
</tbody>
</table>

You have been pre-matched. Please select one of the deals below to confirm the deal from your side. You will be sent an email as soon as both sides accept the deal.

1. TRANSACTION_ID: {F0708B19-234D-4D0-8D10-1BE795263AE7}
   SELLER_REQUEST_ID: {E6955ED8-60EA-4088-8E01-840D066071CA2}
   Score: 251.08
   Closing Price: $398.75
   B1-S1 Pairing

   Transaction Details:
   Buyer_Reputation = 10;
   Language_From : English
   Language_To : Spanish
   Number_of_Words = 5000;
   Seller_Reputation = 10;
   Seller_Skill = 5;
   Time_To_Completion = 45;

   Confirm the match

Figure 32: Details Seen by B1’s Owner

Qualified Transaction Partners Not Chosen for this Matching:

SELLER_REQUEST_ID : {A4617A69-97DD-4248-8157-5895686117}
This Prospective Agent (SELLER) Has Not Yet Been Matched With Anyone Else.
Score: 104.76
Closing Price: $389.41
B1-S3 Pairing

Details:
Buyer_Reputation = 10;
Language_From : English
Language_To : Spanish
Number_of_Words = 4000;
Seller_Reputation = 10;
Seller_Skill = 5;
Time_To_Completion = 60;

SELLER_REQUEST_ID : {90957764-6FAC-4129-9FF8-E9C076F944E}
This Prospective Agent (SELLER) Has Not Yet Been Matched With Anyone Else.
Score: 0.205
Closing Price: $408.09
B1-S2 Pairing

Details:
Buyer_Reputation = 10;
Language_From : English
Language_To : Spanish
Number_of_Words = 7000;
Seller_Reputation = 5;
Seller_Skill = 3;
Time_To_Completion = 86;

Figure 33: B1’s Owner Sees Details of Potential Deals with other Qualified Transaction Partners
9 EVALUATION: USER TESTING

9.1 Introduction

While modeling of market scenarios allows us to better illustrate the behavior of our system and exhibit its full range of functionality, no evaluation of such a system can be truly complete without a user study to better understand usability and comprehensibility considerations from the perspective of human users. Indeed, since MARI represents a fundamentally novel paradigm for gathering and modeling user preferences, and for matching buyers and sellers in electronic markets, a user study that addresses how well prospective buyers and sellers are able to use and understand the system is of substantial relevance and value. As such, we carried out a user study with real human participants, to complement our evaluation of MARI via simulations of market scenarios.

9.2 Experimental Design

9.2.1 Objectives

Our user study was conducted with two major objectives in mind:

i). To evaluate the user interface of MARI, from a usability perspective;

ii). To evaluate the system holistically, in terms of end user satisfaction. End user satisfaction was measured along a number of dimensions, including: how well users felt that they were able to express what they wanted, to what extent users felt that they could understand how the system works, and to what extent users felt satisfied with and understood the rationale behind the transaction partner the system ultimately matched them with.

9.2.2 Methodology

With the above objectives in mind, we decided to carry out two separate user studies, with six subjects each, corresponding to the two distinct objectives. We formulated the following (common) experimental methodology for each of the studies:
A. We started by giving each subject a 10 minute overview of MARI, and explained how the system works. We also explained the meanings of technical terms in the context of the system. Specifically, we defined the following terms for each subject: agent, attributes, utility functions, optimal values, permissible ranges, active agents, expired agents, matched agents, bids, asks, score metric (surplus), confirmation, transactions, closed transactions, and pending transactions.

B. Next, we asked subjects to create an account, log in and take a couple of minutes to navigate the major sections of the MARI Web site: View Agents, Create Agent, Market Status, Market History.

C. Having pre-initialized the system with a number of seller agents that we had created beforehand, we asked the subject to assume the role of a buyer. Specifically, we asked each subject to assume one of the following personas:

1. A *lawyer* who requires a highly accurate and timely legal translation (high seller reputation, expertise, and time sensitivity);
2. A *tourist* in a foreign city who wants real-time translation of a transcripted conversation with a street vendor (low seller reputation and expertise sensitivity, but high time sensitivity);
3. A *student* who wants to translate an email sent to him by a foreign pen-pal (variable expertise and reputation sensitivity, low time sensitivity).

Since we expect buyer requirements on seller reputation and expertise in a translation market to usually be correlated, the above three personas allow us to span all the interesting possibilities that could arise from differential (high/low) levels of the variables of interest.

D. Having assumed a persona, we asked the subject to go through the exercise of creating a buyer agent. This agent was subsequently immediately matched by the system, with one of the predefined seller agents.
E. We asked the user to look at Market Status, Market History and View Agents pages.

F. Finally, depending upon whether the subject was evaluating the interface or the system as a whole, we posed a number of questions (discussed below).

9.3 Interface Study

9.3.1 Questions Posed

This study was very subjective in its scope and sought to uncover and assess general interface-level usability considerations. For subjects who participated in this study, we posed the following questions:

1. We asked subjects to provide general written comments on what they liked and disliked or found confusing about the interface.

2. We asked subjects to rate, from an ease-of-use perspective, how easy or difficult (confusing) it was to perform a few tasks. We asked subjects to quantify their ratings on a scale of 1-7, where 1 signified “extremely easy and understandable.” and 7 signified “extremely difficult and confusing.” Specifically, we asked subjects to rate the following experiences (the experimental variable name of interest, corresponding to each question, appears in parentheses):
   a. Logging into MARI and proceeding to the specific Language Translation market (login);
   b. Navigating between different parts of the site (navigate);
   c. Initializing agent creation (create);
   d. Selecting a user profile (profile);
   e. Associating a price with the ideal offer configuration (offerprice);
   f. Specifying permissible ranges and picking utility functions (rangeuf);
   g. Answering system generated valuation questions pertaining to preferences (valquestions);
9.3.2 Findings

a) Qualitative Findings

The written comments we receive on what subjects generally liked and disliked about the interface, allowed us to gain a great deal of insight into what we could do to substantially enhance the usability and intuitiveness of our system. Based upon the aggregated comments we received, we implemented the following changes to the MARI user interface:

1. In pages with multiple sections, horizontal lines were placed between adjacent sections to clearly separate one section from the next.

2. There was originally a row of boxes in the navigation bar that served as an alternative way to navigate the site. These boxes are still there for aesthetic reasons but were unlinked to minimize confusion.

3. Subjects were a bit confused that the title and all the links on the top changed after going to the specific translation market from the generic MARI login page. The background color of Translation Marketplace pages was changed to differentiate between these pages and those outside the specific marketplace.

4. Subjects were confused by the presence of "Active Agents", "Matched Agents" etc. headers in specific-market.asp in situations when there weren't any such agents in the market. When there are no agents to list, we now display a line of text stating: “You do not have any active agents so far…” (View Agents page); “There are no active agents in the market so far…” (Market Status page), etc.

5. In market-watch.asp (Market Status pages), we now no longer underline text that is not a link, since this was apparently causing some confusion amongst subjects. We found that the Market Status page could get messy with too much information. It shows a list of all active agents followed by a list of Pending Transactions. We have created links at the top so that users can readily reach each of the sub-sections.
6. The “View Agents” link has been modified to read “View my Agents” so as to differentiate this from the information displayed in Market Status (which shows all agents in the market)

7. When looking at Pending or Closed transactions, we have now added links that allow users to see details of the agents that actually took part in transactions.

8. When showing a user a listing of matched agents in the marketplace, we now differentiate between the agents that are currently unmatched and those that are matched but unconfirmed (i.e. involved in Pending Transactions).

9. The “Back to Welcome Page” link in Market Status and Market History has been changed to read “Back to Translation Market” page and now links to a page which gives the user a personalized view of his agents in the specific market. Previously, the link would take the user back to the generic MARI login.

**b) Quantitative Findings**

A total of six subjects participated in our interface study. From the quantitative part of the questionnaire, we collected the following data (see Section 9.3.1 for interpretation of variable names and ratings):
Table 28: Interface Experiment Results

As can be seen from Table 28, we found that, overall, most aspects of our system are easy to use and understandable from the perspective of users. Associating a price with the ideal offer (offerprice) manifested itself as the most non-intuitive operation. In general, it took some time for subjects to realize that they could gain cues from the Market History and Market Status page when specifying the offer price and selecting utility functions. One way to alleviate this problem would be to provide this hint directly on the offer specification page in the future. Reassuringly, most users did not find answering system generated questions about their preferences to be tedious or difficult. In fact, this interactive mechanism of iteratively extracting preferences was praised by several subjects, who liked the dialogue style interaction schema. Subjects also did not seem to have any trouble in visually associating utility functions with attributes. At a high level, everyone seemed to grasp what the shape of the utility function conveyed. In many cases, subjects chose to retain the default utility functions pre-associated with their selected profile.
9.4 System Study

9.4.1 Questions Posed

Somewhat like the interface study, the system study was also subjective and quite high-level in its scope. Our goal was really to evaluate the functionality of the system, in terms of the end-user’s satisfaction with the results and understanding of the system. We asked subjects to provide quantifiable ratings along several dimensions using a scale of 1-7. Specifically, we asked subjects to rate the following dimensions (the experimental variable name of interest, corresponding to each question, appears in parentheses):

a. "To what extent did you feel satisfied with the transaction partner and deal the system designated for you?" (dealsatisfaction) [Rate: 1 (not satisfied at all)...7 (completely satisfied)].

b. "Based upon the preferences you had specified when creating your agent, and the contextual information provided by the system about other qualified transaction partners who were taken into consideration, to what extent did you understand the rationale behind the transaction partner and deal the system designated for you?" (dealunderstanding) [Rate: 1 (do not understand at all)...7 (completely understand)].

c. "When creating an agent, to what extent did you feel that the system allowed you to adequately express your preferences?" (expressprefs) [Rate: 1 (could not express preferences at all)...7 (could express preferences to the maximum extent)].

9.4.2 Findings

A total of six subjects participated in our system study. From the quantitative part of the questionnaire, we collected the following data (see Section 9.4.1 for interpretation of variable names):
Several conclusions may be drawn from the data in Table 29. We see that, reassuringly, the results are largely positive. The high mean values, and relatively small standard deviations, confirm that, in general, users were satisfied with the results produced by MARI, in every case they understood, at least to some extent, the rationale behind the transaction partner and deal the system designated for them, and, overall, they were reasonably satisfied with the extent to which the system allowed them to express their preferences. By observation, there appears to be some correlation between dealsatisfaction and dealunderstanding, implying that subjects who felt most satisfied with the deal and transaction partner MARI chose for them, were also the same individuals who best understood the rationale behind the system’s choice. As far as being able to adequately express preferences, in retrospect, it seems that it might have been a good idea to give users a frame of reference. While users did feel that they were able to adequately express their preferences at a high-level, and they liked the fact that the system proactively asked them questions about their preferences, it might have been useful to allow users to use the interface of Frictionless Valueshopper [12] or Priceline [35], for instance, so that they could better situate MARI relative to contemporary state-of-the-art systems.

| Subject   | dealsatisfaction | dealunderstanding | exprerpef |s  |
|-----------|------------------|-------------------|-----------|
| Subject 1 | 5                | 6                 | 5         |
| Subject 2 | 6                | 5                 | 5         |
| Subject 3 | 5                | 6                 | 5         |
| Subject 4 | 5                | 5                 | 4         |
| Subject 5 | 6                | 7                 | 5         |
| Subject 6 | 4                | 4                 | 5         |

**Table 29: System Experiment Results**

Mean: 5.17 5.50 4.83
Median: 5 6 5
Standard Deviation: 0.753 1.049 0.408
9.5 Conclusions

Overall, we are pleased with the results of our user studies. The qualitative component of the interface study allowed us to substantially benefit from direct user feedback, so as to further improve the usability of the system. The quantitative aspect of the interface study and the system study confirmed that, in general, users do find our system to be practically useful, functionally advantageous, and adequately fathomable.

For more rigorous evaluation in the future, it would be meaningful to do a study with two treatment groups: one group would use MARI to express preferences and find a suitable deal and transaction partner, while the other group would use an existing commercial system. Our goal would be to try to see whether improvements in perceived quality of the outcome and improved satisfaction can be correlated with usage of MARI, in a statistically significant manner. The biggest obstacle to embarking on such a study is that we would implicitly need to exercise a great deal of control over aspects of the other system, which might not be feasible. In order to isolate the relevant impinging factors, we would have to ensure that the only thing that differs across the experience of subjects in the two treatment groups is the system-interaction schema, or the method by which users are asked to express their preferences, and the matching technique, by which buyer-seller pairings are delineated. If we assume that our subjects will assume the role of buyers, we would, for instance, have to ensure that the potential set of sellers available to these buyers is identical in the two systems and, moreover, that the product or service being traded is inherently identical across the two cases. Needless to say, it could be cumbersome to exercise precise control over such factors.
10 CONCLUSION AND FUTURE DIRECTIONS

10.1 Conclusion

Motivated by the presence of significant deficiencies in contemporary electronic markets, this thesis presents the design, implementation and evaluation of the Multi-Attribute Resource Intermediaries (MARI) system. MARI attempts to overcome many of the shortcomings of today’s electronic markets and represents a novel paradigm for gathering and modeling user preferences, and for matching buyers and sellers. MARI facilitates dynamic, domain-independent marketplace creation. It provides automation and facilitates multi-attribute considerations that help transcend price as the only negotiative dimension. Ultimately, MARI provides a framework for value creation, in a strategic sense, by allowing buyers and sellers to sufficiently differentiate themselves from other players in the market, and thereby establish a competitive advantage. Our evaluations, in the form of market-scenario simulations, highlight the benefits that electronic markets may derive from our system. Our user-studies confirm that participants in these markets would find our system to be practically useful, functionally advantageous, and adequately understandable.

10.2 Future Directions

Going forward, as the MARI architecture and implementation is further refined, we expect to face a number of key questions. In particular, even though we have identified one set of models by which agents will interact and transaction partners will be determined, several issues remain to be addressed.

It would be instructive to explore what algorithms and technologies our system can leverage in the process of information integration and representation, decision analysis, modeling, and reasoning about utilities, heuristical learning and inference from user-interaction, facilitating inter-agent communication and negotiation, and attuning pre-existing knowledge bases in developing and managing shared product ontologies. Further, we would like to see how machine learning techniques can be used in assisting decision
support and negotiation, and could also facilitate dynamic alteration of bids and asks based upon stochastic demand and supply patterns over finite horizons [19, 20, 21].

Although we have conducted a reasonable amount of evaluation, to more comprehensively assess the viability of the ideas presented in this thesis, MARI must be more formally evaluated in the future. Lying at the intersection of several fields of study, MARI lends itself to evaluation using a wide range of multi-disciplinary techniques:

### 10.2.1 Economics and Game Theory

A formal game-theoretic analysis of MARI’s user modeling and transaction brokering techniques would shed further light on what types of strategies might be optimal for buyer and seller agents, and would also help delineate market equilibrium conditions that the system would lead to. An economic analysis of MARI’s “mechanism design” would allow us to better gauge whether the system is incentive compatible [15], such that each economic agent has an incentive to reveal her true preferences, and not benefit from systematically understating or overstating. More generally, we would be able to get a better sense of the conditions under which the system can be “gamed,” meaning that agents can systematically misstate their preferences to gain an unfair advantage.

In some sense, the flexibility that MARI gives buyers in sellers in being able to commit and de-commit from tentative transactions, as their agents are involved in alternative deals, suggests a resemblance to a non-binding, iterative, first-price, sealed-bid auction protocol. A game theoretic analysis along these lines, as well as possible comparisons with other auction paradigms, such as Vickrey (second-price, sealed bid) would be reasonable.

An alternative to purely theoretical analysis would be to execute automated simulations and compile statistical inferences. This simulations, and in particular comparison of these with other simulations of agent economies, such as those performed by Kephart [37, 44], would help us better understand how our system allocates scarce resources compared to alternative techniques.
10.2.2 User Interface Studies

MARI embodies a novel paradigm by which buyers and sellers are able to express their preferences. Although we have conducted some basic usability testing that has helped to drastically improve the usability and comprehensibility of the current interface, more extensive user testing would need to be performed in the future, so as to be able to more accurately gauge the effectiveness of the system.

The most important questions regarding MARI’s ease of use and user interaction schema have to do with whether buyers and sellers feel that the system permits them to express their true preferences with reasonable comprehensiveness, whether the results (tentative buyer-seller pairings) output by the system accurately reflect their underlying preferences, and whether they feel transaction partners are significantly differentiated using MARI versus other systems.

10.2.3 Operations Research

In the process of modeling user preferences and brokering transactions, MARI borrows significantly from Operations Research, Network Optimization and Linear Algebra. It would be interesting to see how the results produced by the system, and thus the user experience, would change if we were to implement more elaborate modeling techniques. For instance, we currently use a quadratic least square approximation for modeling multi-dimensional user utility functions. Experimenting with more complex, higher order approximations would allow us to measure, for instance, the deviations in terms of efficiency (total market surplus) that arise as a consequence of our approximations.

It would be also be interesting to undertake simulations that employ different optimization heuristics, welfare metrics, and matching algorithms, and to see how the quality of the outcome might change as we vary these parameters.

10.2.4 Software Engineering

In the process of implementing MARI, we have taken great pains to build a scalable, robust system with a modular, generic data model. MARI is wholly domain independent
and readily extensible in terms of functionality. Nevertheless, further analysis needs to be performed in order to assess the system’s performance bottlenecks. In particular, if the system is commercially deployed we would need to reassess performance limitations, and would need to consider how our algorithms, particularly the core matching algorithm, may be optimized for speed and fault-tolerance.

10.2.5 Market Research

It would be instructive to attempt to make generalized inferences about domain-specific buyer and seller preferences by analyzing the user-preference data captured by the system. For instance, aggregated buyer preferences for permissible attribute ranges, utility functions and optimal attribute values would help sellers better understand which value-added services their customers care about and which features are most important. This information could then be used by sellers to tailor their product offerings to be more compelling, given the mandate of the marketplace.

10.2.6 Value Proposition Analysis

An alternative mode of evaluation of our system could be to actually survey some current buyers, sellers and “market makers” in electronic markets in an attempt to gauge whether MARI’s value proposition is sufficiently attractive. This would also help us better understand what additional features could be added to the system so as to make it more attractive from the perspective of prospective commercial users. In fact, the most comprehensive evaluation of MARI’s concepts would come from actually deploying the system in a real world commercial setting. This would allow us to benefit from direct user feedback to address considerations ranging from incentive compatibility, market liquidity, and desirability of system-designated matchings, to more mundane concerns such as speed, accuracy, data integrity, ease of use, and scalability.
11 APPENDIX A: IMPLEMENTATION DATA

MODEL

MARI is implemented using three technologies: ASP (Active Server Pages) with Jscript, SQL (Structure Query Language), and XML (extensible Markup Language). All data is stored in the SQL tables. The exceptions are user-profile templates and market-specific questions, which are stored in XML files. ASP files handle and process all the data presented to the user. These files are completely domain-independent, and therefore it is very easy to add new marketplaces to the system. In order to add a new marketplace to the system, the administrator only needs to add two new files to the MARI root directory: <market_name>Profiles.xml and <market_name>Questions.xml.

11.1 Structured Query Language (SQL) Usage

The backbone structure of the MARI data model is comprised of a number of SQL tables. These constitute several data abstractions.

Table ‘Players’ represents the user (see Figure 34). That is where the basic information (name, email, etc.) about the user gets stored. The IS_ADMINISTRATOR field distinguishes administrators from regular users. ASP uses this field to give more detailed picture of the system, as well as additional powers, to the administrator. The most important field is PLAYER_ID, which uniquely identifies the user in the system. With this parameter, one can get all the information about the user in the other database tables. Basic personal information about administrators is hard-coded into the system by the SQL scripts. Regular users fill up the table upon registration.
Table ‘Markets’ lists all the markets in the system, assigning a unique identifier MARKET_NAME to each of them. All other tables use this field to differentiate between markets. This allows the MARI system to support multiple marketplaces simultaneously.

The following tables are used to store more specific information about each market (see Figure 35): MarketAttributes, MarketProfiles, MarketPlayersInfo, MarketQuestions, and MarketAnswers. Specific information pertaining to any particular market can easily be accessed via the MARKET_NAME field of the Markets table.

Tables PlayerProfileMap and MarketPlayerRequests map the user to his profiles and requests (equivalent to agents) respectively. Since users can have more than one profile in different Markets, the PlayerProfileMap table usually has multiple entries for each user. The MarketPlayerRequests table has MARKET_NAME as one of the fields to facilitate data handling.

Transactions (matched requests) are represented using the following tables: PendingTransactions, ClosedTransactions and TransactionAttributes. PendingTransactions keeps track of all system-matched, but not yet finalized requests (i.e. the system has matched a buyer-seller pair, but either the buyer, or the seller, or both have not yet confirmed their intent to consummate the transaction based upon the system matching). ClosedTransactions keeps track of all matchings that have been confirmed by both the matched buyer and seller. Finally, TransactionAttributes shows the exact details
of each transaction, whether closed or pending. All three tables are bound together by the common TRANSACTION_ID identifier.

![Figure 35: Table 'Markets' and its Interaction with Other Tables](image)

Table 'Attributes' represents the attributes or features of products or services being traded in MARI marketplaces. This is a generic and versatile table. It can readily hold attribute information for Profiles, for Requests, as well as the request-history information for the ClosedTransactions. In order to identify which attribute belongs to which type of data, table MarketToXToAttributeMap comes into play (see Figure 36). It maps each type of data-type (profile, request or transaction) to all the attributes associated with that particular data-type using the X_ID identifier. MarketToXToAttributeMap is a very important table that significantly simplifies the use of the other Tables.
Finally, the CorrelationCoefficients table stores the mathematical coefficients used in generating a mathematical approximation to the global utility function of the user.

11.2 Extensible Markup Language (XML) Usage

There are two data files written in XML within the system: <market_name>Profiles.xml and <market_name>Questions.xml.

<market_name>Profiles.xml has all user profiles relevant to the specific market under consideration. This file may be easily extended or modified by the administrator, as it is conveniently placed outside the code base of the system. The general structure of the file is the very simple:

```
<profile>
  <profileName></profileName>
  <profileType></profileType>
  <profileAttributes>
```

Figure 36: Table ‘MarketToXTToAttributeMap’ and its Interaction with Other Tables
Questions.xml keeps track of all the questions that are specific to the specific market under consideration, and which cannot be asked by the generic market-independent code. For instance, asking the user about source and target languages are questions specific to the language translation marketplace, and are hence encoded within the TranslationQuestions.xml file. The structure is:

```xml
<question>
  <questionValue> ... </questionValue>
  <attributeName> ... </attributeName>
  <answerType> ... </answerType>
  <answerCategory> ... </answerCategory>
</question>
```

### 11.3 Major Database Tables and Constituent Fields

Now, we iteratively discuss the structure of each major data table in the system, giving a brief overview of the functionality embodied within each:

#### 11.3.1 TABLE Players

**Overview:**
User data abstraction. Represents basic information about the user. The main key is PLAYER_ID, which can also be used to access data in other user-related tables.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAYER_ID</td>
<td>PRIMARY KEY</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td>User's first name</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td>User's last name</td>
</tr>
<tr>
<td>EMAIL</td>
<td>User's email</td>
</tr>
<tr>
<td>LOGIN_NAME</td>
<td>User's login name</td>
</tr>
<tr>
<td>LOGIN_PASSWORD</td>
<td>User's password</td>
</tr>
<tr>
<td>ONLINE_STATUS_P</td>
<td>1 if user is currently logged in; 0 otherwise</td>
</tr>
<tr>
<td>IS_ADMINISTRATOR</td>
<td>1 if user has administrator privileges; 0 otherwise</td>
</tr>
</tbody>
</table>
11.3.2 TABLE Markets

*Overview:*
Market data abstraction. A list of markets in the system. The main field is MARKET_NAME, which distinguishes markets in other tables.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARKET_ID</td>
<td>PRIMARY KEY, not user</td>
</tr>
<tr>
<td>MARKET_NAME</td>
<td>Market name. Used to uniquely identify the market in other tables; the de-facto PRIMARY KEY</td>
</tr>
<tr>
<td>MARKET_DESC</td>
<td>Description of the market</td>
</tr>
</tbody>
</table>

11.3.3 TABLE PlayerProfileMap

*Overview:*
Maps users to profiles. There is no primary key, but the system usually requests all the profiles for the current user (making PLAYER_ID the de-facto primary key).

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFILE_ID</td>
<td>Unique identifier of the profile</td>
</tr>
<tr>
<td>PLAYER_ID</td>
<td>User unique identifier (derived from the Players Table)</td>
</tr>
</tbody>
</table>

11.3.4 TABLE MarketToXToAttributeMap

*Overview:*
The MarketToXToAttributeMap table serves as mapping mechanism to link data profiles, requests (equivalent to buying or selling agents), and transactions (derived from the MarketPlayerRequests, MarketProfiles, and PendingTransactions tables respectively) to attributes (stored in the Attributes table). This approach simplifies table creation significantly. Given an id (X_ID), and a type of record it represents (X_TYPE), the MarketToXToAttributeMap table yields a list of all attributes associated with this record. X_ID serves as the de-facto primary key.
### 11.3.5 TABLE ClosedTransactions

**Overview:**

Keeps record of all committed or closed transactions (i.e. transactions that matched buyer-seller pairs have agreed to consummate) in the system. Exact values of all attributes assigned during the transaction can be obtained using TRANSACTION_ID in TransactionAttributes table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTION_ID</td>
<td>PRIMARY KEY representing the transactions</td>
</tr>
<tr>
<td>TRANSACTION_DATE</td>
<td>Time at which transaction was committed</td>
</tr>
<tr>
<td>MARKET_NAME</td>
<td>Market name, where transaction took place</td>
</tr>
<tr>
<td>BUYER_REQUEST_ID</td>
<td>Unique identifier for the buyer’s request. Used to obtain more detail</td>
</tr>
<tr>
<td></td>
<td>in MarketPlayerRequests table</td>
</tr>
<tr>
<td>SELLER_REQUEST_ID</td>
<td>Unique identifier for the seller’s request. Used to obtain more detail</td>
</tr>
<tr>
<td></td>
<td>in MarketPlayerRequests table</td>
</tr>
</tbody>
</table>

### 11.3.6 TABLE PendingTransactions

**Overview:**

Keeps record of all pending transactions is the system (i.e. the case when the system has matched a buyer and seller, but at least one of them has not yet confirmed the deal). Exact values of all attributes assigned during the transaction can be obtained using TRANSACTION_ID in TransactionAttributes table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTION_ID</td>
<td>PRIMARY KEY representing the transactions</td>
</tr>
<tr>
<td>TRANSACTION_DATE</td>
<td>Time at which system matched the buyer and seller request</td>
</tr>
<tr>
<td>MARKET_NAME</td>
<td>Market name, where transaction took place</td>
</tr>
</tbody>
</table>
BUYER_REQUEST_ID | Unique identifier for the buyer's request. Used to obtain more detail in MarketPlayerRequests table
SELLER_REQUEST_ID | Unique identifier for the seller’s request. Used to obtain more detail in MarketPlayerRequests table
BUYER_CONFIRMATION | Indicates whether the buyer confirmed the deal: 1 if yes, 0 if no
SELLER_CONFIRMATION | Indicates whether the seller confirmed the deal: 1 if yes, 0 if no

11.3.7 TABLE TransactionAttributes

**Overview:**

Keeps a detailed record of all transactions, pending as well as confirmed. Given a TRANSACTION_ID, gives the list of corresponding attributes and their values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTION_ID</td>
<td>Unique identifier representing transaction</td>
</tr>
<tr>
<td>ATTRIBUTE_NAME</td>
<td>Attribute name</td>
</tr>
<tr>
<td>ATTRIBUTE_VALUE</td>
<td>Value of the attribute given in ATTRIBUTE_NAME</td>
</tr>
</tbody>
</table>

11.3.8 TABLE Attributes

**Overview:**

Data abstraction representing Attributes or features. Attributes are basic elements of requests, profiles and transactions. An Attribute has an optimal value (specified by the buyer or seller in the ideal offer), a mathematical functional approximation represented by coefficients (A0,A1,A2), and a general form (derived from the utility function associated with the attribute by the buyer or seller)

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_ID</td>
<td>PRIMARY KEY</td>
</tr>
<tr>
<td>ATTRIBUTE_TYPE</td>
<td>Type of the attribute: 0 if personal (i.e. owned by an agent, such as the agent’s own reputation); 1 if desired (i.e. pertaining to the desired transaction partner)</td>
</tr>
<tr>
<td>ATTRIBUTE_NAME</td>
<td>Attribute name</td>
</tr>
<tr>
<td>UTILITY_FUNCTION</td>
<td>Number representing the utility function corresponding to the Attribute</td>
</tr>
</tbody>
</table>
| A0                | Constant coefficient in mathematical equation corresponding to
### 11.3.9 TABLE CorrelationCoefficients

**Overview:**
Used in least squares approximation to represent correlation coefficients between variables. This table facilitates a second-degree least-squares approximation. First-degree coefficients are stored in the COEFF_VALUE field of the attributes themselves.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_ID_1</td>
<td>Unique identifier representing attribute 1</td>
</tr>
<tr>
<td>ATTRIBUTE_ID_2</td>
<td>Unique identifier representing attribute 2</td>
</tr>
<tr>
<td>COEFF_VALUE</td>
<td>Corresponding correlation coefficient for second order least squares</td>
</tr>
</tbody>
</table>

---

### 11.3.10 TABLE MarketPlayerRequests

**Overview:**
Data abstraction representing all types of requests (a ‘request’ is equivalent to a buyer or seller agent): active (not expired, not yet matched), expired (matched but not yet confirmed) (see PendingTransactions table); and finalized (matched and confirmed) (see ClosedTransactions table).
### 11.3.11 TABLE MarketProfiles

**Overview:**
List of profiles in the market identified by the MARKET_NAME field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFILE_ID</td>
<td>PRIMARY KEY, identifies a profile in other tables</td>
</tr>
<tr>
<td>MARKET_NAME</td>
<td>Market name, where this profile is registered</td>
</tr>
<tr>
<td>PROFILE_NAME</td>
<td>Name of the profile</td>
</tr>
<tr>
<td>BUY_SELL_P</td>
<td>Indicates whether this profile represents a buyer (0) or a seller (1)</td>
</tr>
</tbody>
</table>

### 11.3.12 TABLE MarketQuestions

**Overview:**
A list of irregular questions in the market, MARKET_NAME, that cannot be stated in a generic market-independent way. For instance, asking users about source and target languages in a language translation marketplace are examples of irregular questions specific to the language translation marketplace.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARKET_NAME</td>
<td>Market name, in which the question is relevant</td>
</tr>
<tr>
<td>ATTRIBUTE_NAME</td>
<td>Name of the attribute, for which the question is asked</td>
</tr>
<tr>
<td>QUESTION_VALUE</td>
<td>The question itself</td>
</tr>
<tr>
<td>ANSWER_TYPE</td>
<td>Format of the answer (list, number, text, etc.)</td>
</tr>
</tbody>
</table>
11.3.13 TABLE MarketAnswers

Overview:
A list of possible answers to the questions in the table MarketQuestions.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARKET_NAME</td>
<td>Market name, in which the answer is relevant</td>
</tr>
<tr>
<td>ATTRIBUTE_NAME</td>
<td>Name of the attribute, for which the answer is relevant</td>
</tr>
<tr>
<td>ANSWER_VALUE</td>
<td>Possible value of the answer</td>
</tr>
<tr>
<td>ANSWER_ID</td>
<td>Integer that uniquely identifies the answer</td>
</tr>
</tbody>
</table>

11.4 Schematic Depiction of Database Components and Their Interrelationships

Having discussed, in detail, the structure and functionality of each major table within the system, we now present the MARI user interaction schema from an internal data-flow perspective. Figure 37 presents a schematic depiction of database components, illustrating the interrelationships between these components, and highlighting the flow of data within back-end tables. The numerical steps labeled in Figure 37 are discussed in detail below the figure.
Figure 37: Interaction Schema between Major MARI Database Components
Step 1: The user chooses a market in which to begin a transaction.
The *Players* table serves as the main repository of user information: biographical data, MARI access privileges, online status etc.
The *Market* table contains market specific information such as ‘Market Maker’ information and any special characteristics.

Step 2: The user chooses a profile that most closely represents his preferences.
A Profile is a pre-specified set of user preferences.

Step 3: A new player-profile pair (a user to profile mapping) is added to the database.
Since the user-profile map is many-to-many, the *PlayerProfileMap* table is used to represent this.

Step 4: Each profile has a set of associated attributes and this mapping is recorded in the *MarketToAToAttributes* table.

Step 5: MARI automatically determines what questions the user ought to be asked so as to extract his general utility function.
A ‘Request’ (equivalent to an ‘agent’), is represented internally as a modified profile, bound by the *MarketToXToAttributeMap* table to specific attribute preferences. Requests have various expiration properties, normally expiring after a transaction is completed.

Step 6: The user preferences or utility function approximation along each attribute dimension is written to the appropriate tables, and the relationship is recorded in the *MarketToXToAttributeMap* table.
The “Attributes” of a request include the user’s utility functions along each attribute dimension.
The *CorrelationCoefficients* table is used to store second order coefficients for general utility function approximation.
Step 7: Once the Matching algorithm is run, transaction suggested by MARI, but not yet confirmed by the matched parties, are stored in the *PendingTransactions* table.

Step 8: The attribute values corresponding to the final “deal” configuration between each matched buyer seller pair are written out to the *TransactionAttributes* table.

Step 9: Upon user confirmation by the matched parties, transactions are treated as ‘closed’ and stored in the *ClosedTransactions* table.
12 APPENDIX B: IMPLEMENTATION BACK END COMPONENTS

12.1 Introduction

In the creation of both the back end and the front-end specific programming and user experience goals were kept in mind. After exhaustively diagramming what user interface components were needed, programming goals were added to realize the desired functionality. Figure 38 schematically illustrates the various back-end components (ASP files) in the MARI system and the inter-relationships between them. Subsequently, we discuss each back-end file’s functionality from a programming perspective as well as the user experience perspective.
Figure 38: Flowchart Illustrating Major Back End Components and Programmatic Control Flow
12.2 Back-End Components

12.2.1 Login.asp

a) User Experience
The user is prompted for a Username and Password. The red background and yellow highlighted link scheme is repeated throughout the web page. The user is given the option to create an account in the top-most link.

b) Programming
Behind the scenes, a simple SQL query searches to see if the Username-Password pair is correctly matched. If so, the user is redirected to the welcome.asp page with the username data in the URL.

12.2.2 Welcome.asp

a) User Experience
The welcome page greets the user with his or her username and personal details. The user has the option to modify data by clicking the “Change your info” button as well as the option to go to a specific marketplace (such as the language translation market) where further modifications of the buying agents and selling agents can occur.

b) Programming

Input:
- playerid, an alphanumeric string or nonce that uniquely identifies the user

Output:
- playerid
- market, the specific market (such as translation market) the user is accessing

Effects:
- List all of the user’s information such as the username, first and last name, and email address with an option button allowing the user to change the information. The file queries the Players table based on the input of ‘playerid’ in the database to get this general user information.
The outputs above are attached to the URL string if faq.asp, about.asp, contact-info.asp, logout.asp, or selectmarket.asp are selected. If the user goes to a specific market (such as ‘language translation market’), both the playerid and the market are outputted.

12.2.3 Specific-market.asp

a) User Experience

This page presents new menu options to the user as well as a broad overview of the current outstanding requests. The user can create buying and selling agents, as well as view the current ‘market status’.

b) Programming

Input:
- playerid

Output:
- playerid
- market, the specific market the user is accessing

Effects:
- This file displays outstanding Active, Matched, and Expired requests by querying the MarketPlayerRequests table with the playerid.

12.2.4 Create-agent.asp

a) User Experience

This page gives the user the option to create a buying or selling agent.

b) Programming

Input:
- playerid

Output:
- playerid
• isBuy, a Boolean which indicates whether the agent being created is a buyer agent or a seller agent
• market

Effects:
• This file allows the user to select whether to create a buying agent or a selling agent and, in both cases, the browser redirects the user to choose-profile.asp.

12.2.5 Choose-profile.asp

a) User Experience
This page gives the user the option to choose a pre-defined buyer or seller profile.

b) Programming

Input:
• playerid
• market
• isBuy

Output:
• playerid
• market
• MenuSelectProfile, the profile selected by the user
• isExpire, a Boolean that indicates whether the agent being created can ‘expire’ after some time. Only Administrators may create non-expirable agents
• visit, a variable that keeps track of the number of user visits for internal state maintenance

Effects:
• The page gives the user the option to choose a pre-defined profile ("stereotype") for the agent he is trying to create. From this page, onwards, the user must sequentially navigate through the files: choose-profile.asp, start-request.asp, market-parameters.asp, and place-request.asp, regardless of whether a buying or selling agent is being created.
The type of profile can be chosen in a menu that accesses the MarketProfiles table for a complete list of available profiles, and queries the PlayerProfiles table for the current profile of the user (taken to be the profile the user used, the last time he created an agent).

- The page redirects the users with the output variables above to start-request.asp.

### 12.2.6 Start-request.asp

#### a) User Experience

This page asks the user any “special” questions that are specific to the market in which the user is attempting to create an agent. For instance, in the language translation market, asking the user about the source and target languages are seen as special questions.

#### b) Programming

**Input:**
- playerid
- market
- isBuy
- visit

**Output:**
- playerid
- market
- requested, a unique ID assigned to the create-agent request currently being processed
- outerror, any errors generated from previous visits to this file
- special, the attribute ID where errors, if any, are generated
- visit

**Effects:**
- The page updates the profiles of the user in the PlayerProfileMap table (maps users to profiles) and MarketProfiles table (maps profiles to a specific market).
• The page registers the request in the MarketPlayersRequest table (maps players to agents they have created). The page then writes to MarketToXToAttributeMap table with X_TYPE=0 (maps profiles to attributes)
• Lastly the page asks the user questions from the MarketQuestions table giving the user a range of possible answers to select from, from the MarketAnswers table

12.2.7 Market-parameters.asp

The market-parameters.asp file is visited multiple times. Each ‘visit’ (tracked by an internal state variable called ‘visit’) in the looping file is documented below.

12.2.8 Visit 0:

a) User Experience
This page allows the user to choose optimal values for relevant attributes, and to specify the ideal “offer”.

b) Programming

Input:
• playerid
• visit
• market
• sellProfile, the seller profile chosen by the user, if any
• buyProfile, the buyer profile chosen by the user, if any
• isBuy

Output:
• playerid
• market
• requestID, unique ID assigned to the request (agent)

Effects:
- Show a user their parameters based on the profile file and the entries corresponding to this profile in the Attribute table. Ask the user for the optimal values for these parameters (offer configuration) and corresponding bid or ask. Ask for ranges for flexible attributes.

12.2.9 Visit 1:

a) User Experience

The user is shown his selected offer configuration and corresponding bid or ask. The system then asks the user a series of questions. The user is asked to associate a bid or ask value with “bundles” of attributes such that, in each bundle, a particular flexible attribute is held at the lowest or highest endpoint of its permissible range, while all other attributes are held constant at their optimal values. This is only done for quadratic (i.e. non-linear) attributes, where the extra data-points are needed to come up with a functional approximation to the utility function.

b) Programming

Input:
- Same as Visit 0

Output:
- Same as Visit 0

Effects:
- Show chosen optimal values of attributes and the corresponding price.
- Get the price (bid or ask) from the user where a flexible attribute is held at the lower endpoint of its range and the remaining attributes are held at their optimum values; do so for each flexible attribute which has an associated Utility Function $UF$ belonging to the set $\{2,3,4,5,6,16,17\}$
- Get the price (bid or ask) from the user where a flexible attribute is held at the higher endpoint of its range and the remaining attributes are held at their optimum values; do so for each flexible attribute which has an associated Utility Function $UF$ belonging to the set $\{1,2,3,5,6,13,14\}$. 
12.2.10 Visit 2:

a) Programming

*Input:*

- Same as Visit 0

*Output:*

- Same as Visit 0

*Effects:*

- Calculate $A_0$, $A_1$, and $A_2$ coefficients, corresponding to a mathematical approximation of a quadratic utility function of the form $A_0 + (A_1)x + (A_2)x^2$. An equation is generated for the utility function corresponding to each flexible attribute.

12.2.11 Visit 3:

a) Programming

*Input:*

- Same as Visit 0

*Output:*

- Same as Visit 0

*Effects:*

- Report all errors that may have occurred if the user enters incorrect or inconsistent data.

12.2.12 Market-watch.asp

a) User Experience

This page shows the user an overview of the complete marketplace: the number of users logged in and the number of active agents, differentiating between buying agents and selling agents.
b) Programming

Input:
- playerid
- market

Output: none

Effects:
- The page displays general MARI system information by querying the MarketPlayerRequest table.

12.2.13 Market-history.asp

a) User Experience
This page shows the history of closed transactions.

b) Programming

Input:
- playerid
- market

Output: none

Effects:
- The page queries the ClosedTransactions Table. If any “closed transactions” exist, these are listed. A distinction is made between those transactions which were consummated (i.e. the buyer and seller proceeded to complete the transaction after being matched by the system) and those which were not.

12.2.14 Process-request.asp

a) User Experience
This page shows the user the attributes of a request (agent) selected from a specific market. In the case of an active (unmatched) agent, the user can review the agent, viewing all attributes, and can choose to delete the agent if desired. In the case of a
matched agent, the user can ‘confirm’ the match or can delete the agent, indicating that
the matching proposed by the system was not found to be suitable.

b) Programming

Input:
- playerid
- requestid, uniquely identifies the request or agent
- whattodo, an internal variable used to keep track of the user’s intentions
  (delete an agent, confirm a matching etc.)
- visit

Output: playerid

Effects:
- The page queries the MarketPlayerRequests, Attributes, PendingTransactions, 
  and TransactionAttributes tables to output all the information about the agent.
  The MarketToXToAttributeMap table serves as mapping vector to link data 
  profiles, requests, or transactions to the other tables mentioned above.
- If the user wants to delete the agent the MarketToXToAttributeMap, 
  MarketPlayerRequests, and PendingTransactions tables are updated by 
  deleting the agent’s id from the tables.
- If the user wants to confirm a matching, the state of the matching in the 
  PendingTransactions table is updated to reflect this. If both the buyer and 
  seller involved in a given matching have confirmed, then the transaction is 
  transferred to the ClosedTransactions table.
- Redirects to specific-market.asp upon completion.

12.2.15 Matching.asp

Input:
- marketname, an identifier for the market for which matching script is invoked

Effects:
This is the script that matches buyers and sellers in a specific market. It has two modes of execution: continuous and clocked. In continuous mode, matching is run as soon as a user completes creating his agent. In clock mode, a server-side executable is scheduled to invoke the matching script at set time periods, or the system executes the matching when certain (configurable) predetermined conditions are satisfied by the market (this allows the MarketMaker to say, for instance, that the matching should only be invoked when there are at least $n$ buyer or $m$ sellers, etc.).

The matching script has four main stages of execution:

1. Market state loading;
2. Pair-wise buyer-seller optimization to search for a myopically optimal “deal”;
3. Global buyer-seller matching;
4. User notification.

In the first stage, the existing bid requests and user information in the database is projected onto the specific market, and loaded from database tables into data structures that reside in memory. This operation effectively caches the information to be used by the matching algorithms and speeds up the operation of the script by greatly reducing database access operations.

Pair-wise buyer-seller optimization seeks to find a myopically optimal “deal” configuration for every possible buyer-seller pair. The existing optimality conditions minimize the distance (in the usual Euclidean sense, treating deal attributes as vector components) of the proposed deal from the deals considered optimal by the users (offer configurations), while keeping the profit (defined as bid-ask spread or surplus) as large as possible. The results – potential buyer-seller matches and corresponding deals – are stored in another set of data structures.

Next, global matching takes place. The algorithm borrows from Gray code techniques, and implements an iterative depth-first search on deal attributes. The algorithm maximizes the overall (in the additive sense) profit from all the matches.
Finally, the users involved in the globally optimal matching are sent email notification with details of the proposed deal.

Final potential deals to are written to the PendingTransactions and TransactionAttributes tables. Buyer and seller agents (requests) involved in these deals are expired in the MarketPlayerRequests table.

12.2.16 Logout.asp

a) User Experience
This page logs the user out.

b) Programming
Input:
- playerid
Output: none
Effects:
- The page updates the ONLINE_STATUS_P field in the Players table for the playerid given to 0, indicating that the user is no longer logged into the system.

12.2.17 Faq.asp

a) User Experience
This page archives answers to frequently asked questions posed by users.

b) Programming
Input: none
Output: none
Effects:
- None. Displays static output
12.2.18 About.asp

a) User Experience
This static page tells the user information about the MARI project, and about specific marketplaces which have been instantiated using the system.

b) Programming
   Input: none
   Output: none
   Effects:
   • None. Displays static output.

12.2.19 Contact-info.asp

a) User Experience
This static page tells the user information about whom to contact with questions about the marketplace, and about support-related issues.

b) Programming
   Input: none
   Output: none
   Effects:
   • None. Displays static output.
13 APPENDIX C: SUGGESTIONS FOR FURTHER READING ABOUT MARI

The MARI project has resulted in a number of publications that have appeared, or will soon appear, in refereed conferences, journals and books. Of the publications listed below, special attention is drawn to “Personalized Location-based Brokering Using an Agent-Based Intermediary Architecture,” which illustrates how the MARI system can readily be instantiated in a domain other than our prototype language translation marketplace. Specifically, the paper provides an in-depth discussion of personalized location-based restaurant brokering using the MARI system. Special attention is also drawn to “A Visual Preference-Modeling and Decision-Support Technique for Buyers of Multi-Attribute Products,” in which we acknowledge the compelling need for radically different interfaces for electronic markets, and briefly discuss the theoretical foundations of a novel proposal for the visualization and conceptualization of multi-attribute product spaces.


14 BIBLIOGRAPHY

14.1 Works Referenced

1 Altavista online translation system URL: <http://world.altavista.com/>

2 Amazon Auctions URL: <http://www.amazon.com/auctions/>

3 Bargainfinder: <http://bf.cstar.ac.com/bf>


6 Chemdex URL: <http://www.chemdex.com/>


10 Ebay URL: <http://www.ebay.com/>

11 Elance URL: <http://www.elance.com/>

12 Frictionless ValueShopper URL: <http://compare.frictionless.com/>


17 Jango URL: <http://www.jango.com/>

19 Kasbah URL: <http://ecommerce.media.mit.edu/Kasbah/>


21 Lost Wax URL: <http://www.lostwax.com/>


26 Moai URL: <http://www.moai.com/>


33 Pattie Maes’ statement URL:

34 Personalogic URL: <http://www.personfalogic.com>

35 Priceline URL: <http://www.priceline.com>


47 TradingDynamics URL: <http://www.tradingdynamics.com>


50 W. Walsh, M. Wellman, P. Wurman, J. MacKie-Mason. “Some Economics of Mark-

### 14.2 Works Consulted


