A MultiAgent Approach Towards Solving Complex Problems of Societal Information Systems

by

Hongying Du

Bachelor of Engineering
Northwestern Polytechnical University 2006
Master of Engineering
Northwestern Polytechnical University 2009

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosphy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2014

Accepted by:

Michael N. Huhns, Major Professor

Manton M. Matthews, Committee Member

Jose Vidal, Committee Member

Munindar P. Singh, External Examiner

Kuldar Taveter, External Examiner

Lacy Ford, Vice Provost and Dean of Graduate Studies

Abstract

Complex resource allocation problems arise due to complex human societies and scarce resources to be distributed. Scarce resources could be food, water, energy, etc. Meanwhile, the size of the problem, the intersection of different areas, and possible global consequences all add to the complexity of the problems, which makes it difficult for humans to solve the problems by themselves.

For all these reasons, humans need technical help to tackle complex problems. Since humans participated in the problems usually own part of the information about the problems, and no one may see the whole picture of the problems, it is natural to use distributed systems to simulate and analyze the problems. In a distributed system, humans represented by agents know only partial information interact with each other in order to achieve a common goal while maximizing their own interests. The formed distributed system is called a multiagent system because multiple agents are involved in the systems.

In this dissertation, we studied three cases of multiagent systems to help with distributing a certain kind of resource. First we present an approach to assist grocery shopping. The aim is to help a customer to find the most economical way of shopping. A customer would save 21% or more most of the time with simulated price data and 6.7% with real price data. Robustness is also considered with deceptive stores and wrongly reported prices. Second we simulate a healthcare system in which agents are used to assist a patient to find a physician. We investigate four different strategies for assisting a person in choosing a physician and three physician-waiting strategies in three common social network models. The results show that the societal information system can decrease the number of annual sick days per person by 0.42-1.84 days compared with choosing a physician randomly. Third we investigate the influence of humans' personalities on resource allocation in mixed human-agent societies. It is shown that humans treat other humans and agents differently and humans with different temperaments behave differently, but not with significantly difference, which means fair is more important than personality types while making decisions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Because human societies are complex and they must rely on resources that are scarce, they have complex resource allocation problems. Scarce resources might be different under different circumstances, such as food, energy, medical care, clean water, etc. To solve these problems, human interact with each other, and the interaction or information flow among them forms different kinds of societal information systems. There are various kinds of societal information systems serving different purposes. Take a simple situation as an example, a patient is trying to find a doctor who could take care of him. He may have a goal to spend money as less as possible, or to cure him as soon as possible, but he has no idea which doctor fits his purpose best. Thus he turns to his friends, his friends' friends if necessary, for recommendations, incorporate these information into consideration, and make a decision. In this example, the persons involved and the information flow form a societal information system and the purpose is to find a doctor for the patient. There are more complex information systems like the ones in section 1.2.

As the scale of the system grows, more complex problems, some with global interactions which lead to global consequences, emerge. In many cases it is hard to find solutions to the problems or perform experiments on real societal information systems due to various reasons, such as difficulty of synchronization, long time span of doing the experiments, or extreme geological conditions.

Because of complexity of the systems, the huge amount of information flows and other factors that makes problems hard to solve, human need computational help in designing, implementing, and evaluating the systems. With technical help of simulated systems, the cost of experiments is reduced and the purpose of study is fulfilled. Several questions need to be considered while designing a simulated societal information system. For instance, how big should the system be and what entities are involved? What kind of information should a designer keep an eye on? What consequences to expect and what goals to achieve? We'll see more analysis in designing a simulated system in section 1.2.

Humans involved in complex problems usually only have access to partial information and they try to achieve a common goal while pursuing their own interests. This characteristic is consistent with the feature of distributed systems where agents with partial information are used to assist humans to make decisions. Such systems with multiple agents are called MultiAgent Systems (MAS). Agents are an autonomous software entities that can act on the behalf of his principle, sometimes a human in a societal information system, based on his knowledge and judgment. This natural characteristic makes an agent a good representative of a human. Also, the amount of information flow in a societal information system could be enormous potentially, which is beyond the processing ability of humans brain, thus it is better that an autonomous agent is used to gather information, communicate with other agents and make decisions on the behalf of a person. A multiagent system is a system/society that gathers multiple agents who interact with each other. There is information flow among these agents who have goals based on their principles' interest. Nowadays agent technology is used everywhere, ranging from industry such as fault detection, energy distribution, to everyday life, such as web services, security patrols. First two parts in this dissertation use multiagent systems to simulate grocery shopping scenario and health care systems.

Due to the popularity of the agents existed in our everyday life it is inevitable that human-agent mixed societies emerge. In such societies, humans and agents exchange information and work together to achieve a particular goal, compete with each other, or have more complex relationships. Examples of working together include teaching children languages or mathematics using emotional agents. An emotional agent is an agent with emotions and it expresses its emotions by expressions of its animated face on the screen or words that programmed in it. If a child answers a question correctly or performs well, the agent smiles or does other positive expressions and actions. An example of competition is that humans and agents take part in an auction and bid for some goods on the Internet.

It is important to understand how humans and agents interact in various human-agent mixed societies in different aspects. For example, will humans have the same performance in the mixed societies as previous while there's no agent involved? What factors influence humans' decisions/attitude towards agents? Do humans' personalities play a part in their decisions and how? Many researchers studied the first two questions but less studied the third question. The third part of this dissertation is trying to find an insight into those questions about relationship between personalities and decisions. Conclusions to these questions could be used in many ways. For example, predict the performance of humans in a game knowing their personalities, or assign an agent with the "proper" personality to accompany a human, etc.

This introduction continues with a description of societal information systems in section 1.1, which leads to potential complex problems in section 1.2. To find solutions for these problems, we need some technical help from the artificial intelligence domain. In section 1.3, the concept of multiagent systems is introduced and why multiagent systems fit the purpose of analyzing these problems is stated. The cases that will be studied in this dissertation are introduced briefly.

1.1 Societal Information Systems

Before doing analysis of complex problems, we should understand the environment that the problems build on, i.e., the different kinds of societal information systems. Today everyone lives in some societal information systems one way or another. Societal information systems are large-scale information systems that gather information from hundreds, perhaps thousands, of nodes, each associated with a person, and then process and use the information to affect the behaviors of the people at the nodes [61]. Societal information systems might not be noticed by ordinary people, but they are everywhere. Let's consider some other examples on top of the find-a-doctor example mentioned above.

Societal information systems are common and play an important role in our era due to increasingly complex societies, which rely on increased connectivity and global interactions. Information globalization changes human life in many ways, from communication between friends, to the way the companies operate their business. All these require technical support of information exchange. Take companies as an example, as Valacich [68] stated, information technology is important because "increasing global competitiveness has forced companies to find ways to be better and to do things less expensively. The answer for many firms continues to be to use information systems to do things better, faster, and cheaper. Using global telecommunications networks, companies can more easily integrate their operations to access new markets for their products and services as well as access a large pool of talented labor in countries with lower wages."

To mimic and analyze the problems in a societal information system, simulated systems are built. Why do we use simulated systems rather than real-world systems? There are a couple of reasons. One is that using simulation would cost less. For example, if you want to do an experiment with a hundred people, you'll have to gather subjects, tell them the rules, and do the experiment at a specific time. If you want to check the influence of the parameters in your system, you'll need to ask the subjects to do the experiment multiple times. All these experiments in real-world systems cost time and energy. Another reason is that sometimes the societal information systems are so complex as stated in section 1.2 that it's hard to perform experiments in the real-world. Simulations costs less and can mimic extreme conditions. Meanwhile, it could be a good imitation of the real-world situation if modeled well. Then computational methodologies are used to analyze and solve these problems. One of these advanced methods is to use multiagent systems, which is elaborated in section 1.3.

1.2 Complex Problems and Solutions

Humans encounter problems everyday, ranging from very personal, such as what to eat for breakfast, to very influential, such as what the best plan is for a company. Nowadays problems become more and more complicated, considering the following three factors:

- size: since the communication of people and exchange of information are very frequent today due to the development of new technologies and market needs, it is very possible that problems encountered have larger size than ever before. For example, people like to take digital pictures and put it on the Internet, and with the increasing size of digital photos today, it takes a lot of space to store these photos and more time to find specific photos. Another example is integrating several databases of huge amount of data. Because the databases are huge and there are complex relationships among them, any operation should be considered or evaluated before they are actually performed. The size of a problem matters because it may motivate new technologies to deal with new challenges brought by the size.

- intersection: a problem may involve different areas and intersect or overlap with other problem domains. For example, consider the problem of arranging the routes of goods transportation of a delivery company everyday. First a couple of key time points should be considered, such as the arrival time of goods to the company. Other things to be considered include available transportation vehicles and human labors, weather, and so on. This problem involves human resources, scheduling, in addition with the help of weather forecasting, and some other areas. For more complex problems, it is inevitable that these problems involve different areas and it is difficult to make a decision, which is why we need the help of technology.

- consequences: due to the above two factors and globalization, some problems today have more influential consequences than before. For example, an erroneous operation on databases of a large electricity company may lead to failure of several power plants, causing residents of an area short of electricity. Another example is global warming, which caused by multiple reasons. Possible reasons include increasing size of people and cars there fore more carbon dioxide, decreasing area of forests, polluted air and seas, and so on, which are all interrelated that the problem couldn't be solved only with the effort of a portion of people. Some events, such as nuclear disaster, happen on one location of the world, but continuously have global consequences, such as the release of radioactive materials after the disaster.

Due to the these factors, some problems are so complex that technical help is needed which is designed to deal with the complexity of these problems. There are two parts or aspects to the technical help:

- How the IT system represents each person and their interactions with each other and the domain.

- How the person interacts with the system.

Since multiagent systems, which is a kind of distributed systems, fit the feature of complex problems in the sense that it doesn't require information centralization and that autonomous agents could represent persons well, we used multiagent systems in our experiments.

1.3 Multiagent Systems

Researchers proposed different definitions about an agent, or an intelligent agent [22]. According to Russell and Norvig [51], an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. Jennings, Sycara and Wooldridge [31] consider an agent as a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives.

All the definitions agree that an agent should be intelligent and autonomous that he could make decisions and act on the principle's behalf on his own based on the environment he perceived [31] [51] [73], as shown in Figure 1.1. An agent could be simple, such as a thermostat which controls the air conditioner of a room and keeps the temperature stable, or something very complex, such as a robot which acts according to the environment and tries to achieve predefined goals. The thermostat perceives information about the environment using the mechanic part that detects the temperature and achieves the goal of keeping the room temperature stable by turning on/off the air conditioner. The robot perceives the environment using cameras and other sensors and takes action, e.g. moving to a specific location, based on the information he integrated. Interestingly, a human could be treated as an agent with organs perceiving the environment and a brain that integrates perceived information and makes decisions.

There are a couple of features that an agent could have as follows [31], while autonomy is the central notion of agency.

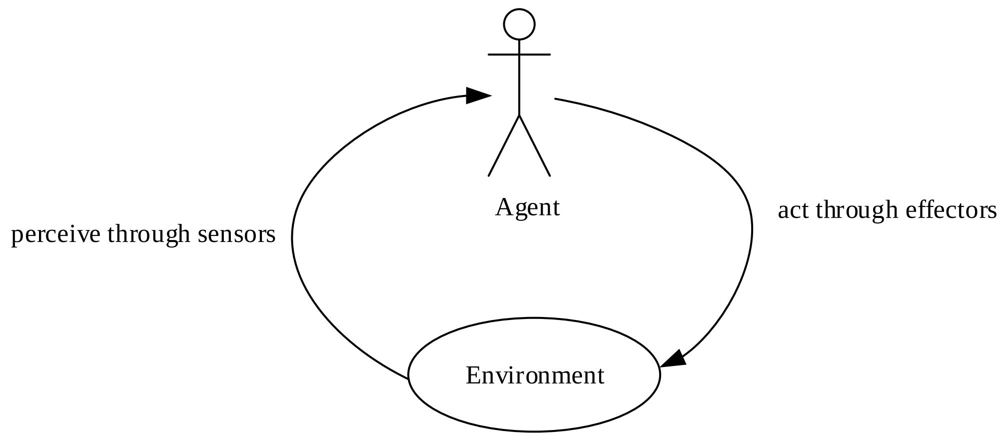

Figure 1.1 The model of an agent

- situatedness: the agent receives sensory input from its environment and it can perform actions which change the environment in some way.

- autonomy: in the sense that the system should be able to act without the direct intervention of humans or other agents.

- flexibility: which contains the following three factors:

  - responsive: the agent should perceive their environment and respond in a timely fashion to changes that occur in it.

  - pro-active: the agent should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate.

  - social: the agent should be able to interact, when appropriate, with other artificial agents and humans in order to complete their own problem solving and to help others with their activities.

There's one more possible characteristic for an agent: rational. Rational means an agent always tries to act in a way that will get him the most benefit, or reward. Humans are not always rational because humans make decisions not only based on logic. Emotions and other factors are involved while humans make decisions.

A multiagent system consists of more than one agent, and these agents interact with each other through communication. Each agent may have incomplete information or capabilities to solve the global problem in question, and they are trying to solve the problem through interactions while their primary goals are maximizing their own benefits. There are many possible ways of interaction, such as cooperation, competition, or negotiation. Due to this high-level of interaction and ability of dealing with potential complex problems, multiagent systems are good solutions to complex problems with multiple solutions/perspectives, such as those mentioned in 1.2. Multiagent systems are used in all the cases studied in this dissertation to solve specific problems.

Chapter 2

Background

Before doing simulation and analysis on the societal information systems, we need to understand the problems of the systems and possible solutions of the problems. In this chapter, we present background knowledge of societal information systems, the difference between a societal information system and a social network, and some perspectives that motivate our work. Then we introduce a complex problem of societal information systems that is related to our case studies here - the resource allocation problem.

2.1 Understanding Societal Information Systems

Societal information systems are the systems that formed by entities, either humans or agents, who exchange and process information. There are various research problems in societal information systems, such as social collaboration modeling [40]. A kind of popular societal information system is social networks. A social network is a social structure that consists of social actors, such as individuals and organizations, and relationships between the actors. Social network sites such as Facebook attract millions of users. There are differences between a societal information system and a social network:

- A societal information system consists of entities, which could be a person, an agent or anything that participates in the system; a social network usually involves persons or organizations.

- A societal information system emphasizes the information exchanged among entities no matter whether there is any connection between entities, while the relationship between two persons in a social network is usually connections, and the connections could be exposed to others if they use social network sites.

Both our first two case studies use a societal information systems, while the health care system involves is also a social network.

Crowdsourcing

One of the most promising approach to solve complex problems in societal information systems is to use crowdsourcing, which is a distributed methodology. Here is the definition from Howe [24], who coined the word "crowdsourcing" in 2006: crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers.

Crowdsourcing is used by large companies, such as Amazon and Google. Amazon's crowdsourcing platform, Amazon Mechanical Turk (AMT), allows people post or process tasks. Companies could use crowdsourcing to receive solutions quickly at relatively little cost [53] [58]. One problem that a crowdsourcing system designer should concern is the incentives [43] [63] [54].

We kind of borrow the concept of crowdsourcing in our first two case studies, but use it in a different way: both utilize the power of the crowd. In the grocery shopping scenario, we may rely on the goods information reported by customers. In the health care system, information related to physicians is passed through the network of agents and is integrated.

Mixed Human-Agent Societies

As we mentioned earlier, due to the enormous participation of agents into human societies, mixed human-agent societies are formed. For example, "social computers" which combine software and human services are constructed. Truong et al. [65] [64] propose a method to model human capabilities using cloud computing concepts and combine it with software-based services and establish clouds of hybrid services. Sierhuis et al. [7] use a human-centered perspective on teamwork and adjustable autonomy in mixed human-agent groups and integrate the Brahms [13] and KAoS [67] agent frameworks to model real work situations.

Because of the challenges in human-agent teamwork coordination [8], we want to explore ways that could improve the coordination. Among many perspectives or aspects that can be used to improve coordination, we particularly look into a psychological factor: personality in the third case study in the hope of understanding humans' attitude towards agents better and expecting conclusions that could be utilized in human-agent interaction.

2.2 Resource Allocation Problems

There is a lot of concerns on resource allocation problems in both computer science and economics fields, especially when the resource is scarce. There are many circumstances under which we need to distribute different kinds of resources, such as electricity, water, network bandwidth, among multiple entities or agents. A particular distribution of the resource is called an allocation. Since multiple agents are involved, resource allocation problems are also called *MultiAgent Resource Allocation (MARA)* problems.

Multiagent Resource allocation has a wide range of applications, such as manufacturing and scheduling [38] [1], logistics [52] and so on. Chevaleyre et al. [12] presented techniques, concepts, and four major application domains of MARA: industrial procurement, the joint exploitation of Earth Observation Satellites, manufacturing control, and grid computing. Feldman, Lai, and Zhang [20] proposes a distributed allocation scheme that converges quickly to an equilibrium while maintaining the balance of efficiency and the fairness indicated by utility uniformity and envy-freeness. Some researchers are doing resource allocation algorithms related to cloud computing. For example, Ergu et al. [18] proposes a model for task-oriented resource allocation in a cloud computing environment with ranking and a bias matrix is used to solve conflicts. In wireless networks, power, time slots, etc. are the resources that need to be allocated [19] [2] [70].

In game theory, auction is an important mechanism to provide a general solution to discrete resource allocation among selfish agents. Formally speaking, an auction is any protocol that allows agents to indicate their interest in one or more resources and that uses these indications of interest to determine both an allocation resources and a set of payments by the agents. Auction is important because [57]:

- It is widely used in real life, in consumer, corporate, as well as government settings, such as online auctions for goods.

- It provides a general theoretical framework for understanding resource allocation problems among self-interested agents.

The allocation procedure could be centralized or distributed. A centralized procedure requires a single entity that receives the agents' preferences and chooses an outcome that satisfies a certain condition, such as maximizing social welfare. One problem is that agents may lie about their private information, which happens very common in a collaborative environment [42] [41]. Also, it may not always be possible to establish a central entity. A distributed procedure doesn't need the central entity and usually involves negotiation among the agents. Schmidt et al. [55] discuss different distributed resource allocation schemes. Bachrach and Rosenschein [4] propose a distributed and random allocation procedure that converges to the optimal in terms of utilitarian social welfare.

The third case study is related to resource allocation. It asks the participants to play a "Who Gets More Cake?" game which is a variant of cake-cutting resource allocation game and continue with human-agent option after the game.

Chapter 3

A Multiagent System Approach to Grocery Shopping

3.1 Introduction

Aided by information systems for analyzing customer buying data, supermarket chains continually alter the prices of items to maximize their profits. They do this by, in essence, experimenting on their customers. For example, the price of an item might be raised at one store until customers stop buying it. This maximum price is then used at all of the stores in the chain. The customers at the supermarkets, however, do not have any comparable information systems that might aid them in price comparisons and are often at the mercy of the stores. Most stores do not post their prices online, so that consumers have to visit each store to find the prices of groceries, which makes comparison shopping prohibitive.

Imagine an online system where customers could post the prices they paid for their groceries (this could be automated by querying the RFID tags of the items) and where a prospective shopper could enter a grocery list and obtain a pointer to the store with the lowest total price. This would enable comparison shopping for groceries and would render the customer-to-store interactions fairer. It would also encourage stores to offer their true prices to avoid driving away potential customers. However, the effort required from the consumers would be substantial. To make the effort reasonable and manageable, each customer could benefit from an agent that represented his/her interests and interacted with the agents of the other customers and, possibly, with store agents.

However, there is an expense in implementing and operating such a system. Moreover, its success is dependent on prices entered by other consumers, on the availability of goods, and on prices that stores might change to yield an advantage for them to the disadvantage of consumers. Hence, it is subject to errors and manipulation. To be feasible, the potential cost savings must substantially exceed the expense and effort of its implementation.

In this chapter, we investigate the efficacy of a consumer-oriented comparison-shopping system for groceries and the trade-offs in an implementation of it. Our approach is to use real data, normalize it according to typical consumer actions, and simulate a system of stores and consumers. We introduce both random and systematic (manipulation) errors into our simulation in order to evaluate its robustness.

Our assistant agent's objective is to assist a customer by all means, especially by providing a customer with the best combination of price and quality for a list of products available at different stores and making recommendations of store(s) optimal for shopping. The whole shopping procedure contains the following four steps, shown in Figure 3.1.

- creating shopping list: a customer creates a shopping list based on his/her needs. He should specify items/products and the quantities of the items.

- finding stores: find a series of available stores according to store hours, locations, the customer's preference and other possible factors.

- deciding stores: decide which store(s) to go to with the help of an assistance agent.

- transacting: drive to the store(s) and make transactions.

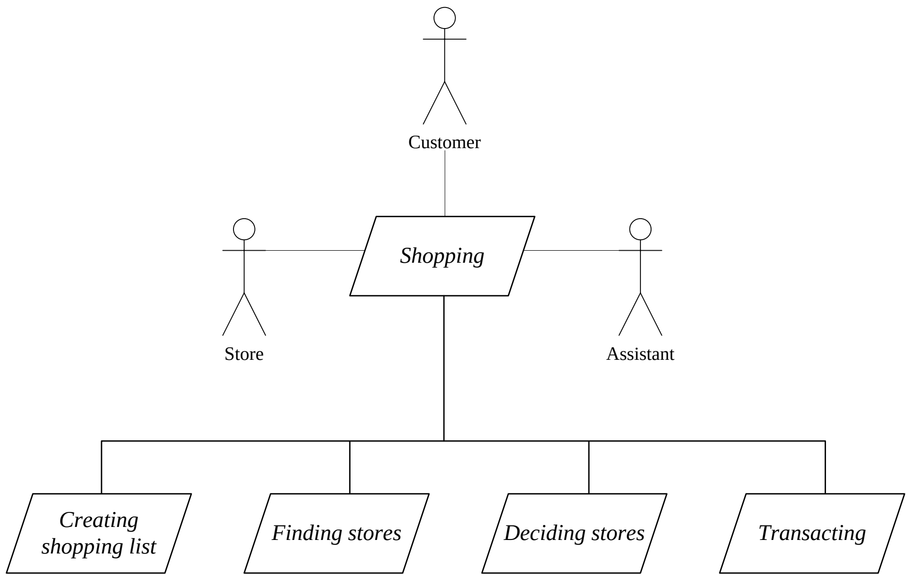

Figure 3.1 Overall goal model

3.2 Background

Price comparison services (also known as comparison shopping services) allow people to query a product's prices at online stores. The services list the product's prices in all of the stores and sort the prices to provide customers with support for their online shopping. An intelligent software agent to implement comparison shopping is called a shopbot [14].

In June 1995, the first well-known shopbot called BargainFinder [36] was released by a group of Andersen Consulting researchers as an intelligent software agent for comparison shopping. It was designed to find music CDs and had a rather simple interface. It allowed a user to enter the name of an artist and an album, searched eight online music stores, and displayed all CD prices on a web page. If the user clicked on the name of one of the stores, it would bring the user to the specific album on that store's website. Consumers gained obvious benefit from BargainFinder and it has been used widely. Nowadays, shopbots have greater functionality than before by including information about shipping expenses, taxes, vendors' rates, and product reviews. Some corporations even have their own shopbots, such as Google's Google

Product Search and eBay's shopping.com. Recently there is also a mobile application for comparison shopping called RedLaser which can scan the barcode of a product by the phone's camera, search many online stores, and show their prices on the phone.

There are typically three steps for a shopbot to deal with data. First, it retrieves data from online stores or other shopbots, possibly by using an extraction method, such as [74]. Second, the data is processed according to a user's command. Last, the results are shown to the user on a webpage in a way that can be helpful to the user. One such system lets user re-rank the results locally [10]. Other researchers are developing better algorithms to improve the behavior of shopbots and making their performance more robust to changes in the stores' websites, such as by using Semantic Web concepts [37]. Other related studies involve consumer search costs and benefits [9] [60] and price-setting strategies [23].

3.3 Analysis and Simulation

There are a number of variables in grocery shopping. Our simulation uses five parameters: customer input, customer location, store location, item price, and item quantity. Customer input is a customer shopping list that contains the items the customer wants to buy and the quantity of the items. Store location and customer location are used to calculate the fuel cost when driving to and from the stores. Item prices are those either reported by customers or by stores. We assume the quantity of a specific item in a store is either zero or infinity.

Our algorithm begins with the customer's shopping list of items and quantities. If the customer just goes to the stores with the lowest price for each item, the customer might need to go to many stores and spend more on fuel. So we search in all the stores and find the lowest price and the second lowest price of each item the customer wants to buy. The combinations of these two prices of the items may lead to the most economical way for shopping by reducing the fuel cost. We considered all the possibilities of combination of the two prices and calculate the total cost including grocery cost and the fuel cost. When calculating the fuel cost, we assume the customer goes to the nearest store he needs to go to where he has not already shopped until he gets all the items. For comparison, we also calculate the cost if the customer chooses to go to stores using three other strategies: (1) choose one store randomly and buy all the items at that store, (2) go to the nearest store, or (3) randomly go to one of the five nearest stores. Then we calculate the ratio of the grocery cost and the total cost of these three methods over that of our method to see the difference.

We next evaluate robustness. What if the stores claim their prices are lower than they actually charge if the stores themselves provide the prices? What if the customers make mistakes if they are responsible for reporting the prices to other customers? We also consider these two situations in our simulation.

The NetLogo platform [72] has been proven to be a useful environment for agent-based simulations, such as supply chain simulation [32]. We use it for our grocery shopping simulation. In our simulation, the number of stores and the number of items can be chosen by sliders in the Netlogo GUI, as shown in 3.2. In reality, a customer will usually go to one of a few familiar supermarkets, which means we do not need to indicate very many stores. Our simulation has two phases.

In the first phase, we simulate shopping according to fictitious prices generated randomly and examine the ratio of the cost of other methods over that of our method and evaluate the influence of different values for the parameters. For each combination of parameter values, we ran the simulation 100 times and used the mean of the 100 results. For deception, we assume that the deceptive stores say their prices are 10% lower than the real prices and the percentage of deceptive stores are 25%, 50% and 75% separately to see how it will affect the results.

In the second phase, we use realistic prices of items collected manually in the simulation and see whether there is a big difference between the results of simulation

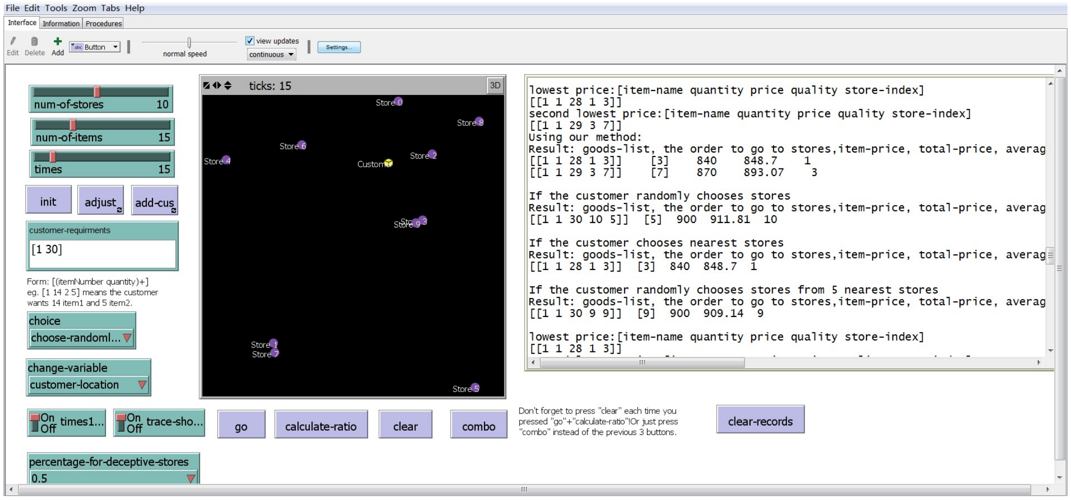

Figure 3.2 NetLogo GUI

using fictitious prices and that of using realistic prices. With realistic prices, store location, item price, and item number are fixed. We did not consider the fuel cost in this part of the simulation, since its effect would be minor compared to the money spent on the items. As for the customer input, we constructed a shopping list according to the U.S. Consumer Price Index (CPI). CPI, which is published by the U.S. Bureau of Labor Statistics, measures a price change for a constant market basket of goods and services from one period to the next within the same area (city, region, or nation) [47]. Along with CPI, the relative importance of components, which measures the importance of the items in the market basket by decimal numbers less than 1, is published. We created a realistic shopping list by selecting an item from each category according to its relative importance [46]. Since there are many categories, we did not include all of them in our shopping list, so the result was a list of 33 items. For these, we collected price data from 5 different stores, as shown in Appendix A. We compared the saving of a customer going to two stores and that of just going to one store. To measure robustness, we checked the results if there was a 10% possibility that the customers reported each digit of the prices wrong.

Table 3.1 Simulation results of changing customer location

Method	Ratio of grocery cost	Ratio of total cost
Choose randomly	1.2382	1.2328
Choose nearest	1.2446	1.2365
Choose randomly from 5 nearest stores	1.2244	1.2178

Table 3.2 Simulation results of changing store location

Method	Ratio of grocery cost	Ratio of total cost
Choose randomly	1.2385	1.2351
Choose nearest	1.2379	1.2325
Choose randomly from 5 nearest stores	1.2315	1.2269

Table 3.3 Simulation results of changing item price

Method	Ratio of grocery cost	Ratio of total cost
Choose randomly	1.2193	1.215
Choose nearest	1.2238	1.218
Choose randomly from 5 nearest stores	1.2275	1.2225

Table 3.4 Simulation results of changing item number

Method	Ratio of grocery cost	Ratio of total cost
Choose randomly	1.2679	1.2637
Choose nearest	1.3384	1.3317
Choose randomly from 5 nearest stores	1.2969	1.2911

3.4 Results and Discussion

In our NetLogo Simulation, we assume there are 12 stores and 30 kinds of items in the stores. Given 10 items a customer wants to buy, we ran the simulation 100 times for a random change in a given parameter and calculated the mean, as shown in following tables. The ratios in the table are the ratio of the grocery cost or total cost of a certain method over that of our method. We also showed what items in which store the customer should buy.

When simulating customers changing the items on their shopping list, using 30 kinds of items increases the program running time remarkably. To make this more manageable, we limit the simulation to 10 stores and 15 kinds of items, as shown in

Table 3.5 Simulation results of changing customer input

Method	Ratio of grocery cost	Ratio of total cost
Choose randomly	1.177	1.1732
Choose nearest	1.114	1.108
Choose randomly from 5 nearest stores	1.1618	1.1573

Table 3.6 Simulation results of deceptive stores

Percentage of deceptive stores	Ratio of grocery cost	Ratio of total cost
25%	1.0192	1.0191
50%	1.0164	1.0162
75%	1.0069	1.0069

Table 3.5.

As can be seen from the above tables, our approach to deciding which stores to shop at can save 21% or more in costs, except when changing customer input. Since we considered all possibilities and ran the simulation many times, it is safe to say that our approach is better than the other methods. As for changing the customer input, the savings are lower, possibly because the program generated the customer input randomly and it may contain fewer items.

We also considered deceptive stores. What if 25%, 50%, 75% stores are deceptive by claiming that their price is 10% lower than the real price? We ran the simulation with deceptive stores chosen randomly. The ratio in Table 3.6 shows the ratio of grocery cost or total cost of our approach using deceptive information over using actual information.

The difference between the cost with real price data and that of deceptive price data is 1.9% when 25% of the stores are deceptive. The difference is smaller if more stores are deceptive: 0.69% with 75% deceptive stores. So when stores are deceptive, the customer will save less than when stores are honest. However, our approach is still valuable, because even after losing 1.9% due to deception, the customer will still save more than 19.1%.

Using the real price data we collected, Table 3.7 shows the total cost of the goods

Table 3.7 Costs of shopping at one store

Store index	0	1	2	3
Cost	114.27	129.52	127.85	129.05

Table 3.8 Costs of shopping at two stores

Store index	0,1	0,2	0,3	0,4	1,2
Cost	106.94	108.72	110.05	106.58	118.95
Store index	1,3	1,4	2,3	2,4	3,4
Cost	117.6	112.05	117.79	115.1	111.85

Table 3.9 Simulation results: costs of going to one store and their frequencies of occurrence

Store index	0	2	3
Cost	114.27	127.85	129.05
Frequency	489	10	1

on the shopping list if a customer goes to just one store. The lowest price is 114.27 from store 0.

The cost of buying each item at its lowest price is 98.44, which is more than 13% lower than going to one store, but a customer would have to go to four stores to get this lowest price. Because a customer might not want to go to more than two stores, we tried all combinations of two stores and calculated the cost. Table 3.8 shows that the lowest cost of 106.58, which occurs when a customer shops at stores 0 and 4, is 6.7% lower than going to just one store.

What if the customers reported the price data wrong? We simulated this situation by giving each digit of a price a 9% possibility to change to other digits randomly, each with a 1% possibility. When the price information is wrong, the only thing changed are the stores the customer would go to. When the customer arrives at the store, he will still pay the real price. We ran the simulation 500 times and the results are shown in Tables 3.9 and 3.10. Notice that some store or store combinations are never chosen in 500 simulations, because their overall costs are too high and thus can hardly be the lowest price, even with a 9% possibility of incorrect price information.

Table 3.10 Simulation results: costs of going to two stores and their frequencies of occurrence

Store index	0,4	0,1	0,2	0,3	3,4
Cost	106.58	106.94	108.72	110.05	111.85
Frequency	315	118	35	30	2

We can see from the tables that as for the results with one store, there is a 2.2% possibility that the customer would go to another store due to the wrong price data, rather than going to the store with the lowest price. The average cost, after 500 simulation runs, is 114.57, which is very close to 114.27. For the results with two stores, there is a 37% possibility that a customer would go to different stores other than the best combination of two stores. Though the possibility is significant, the average cost is 107.04, which is very close to 106.58, the lowest price possible for two stores. So on average, a customer can still save 6.3% by going to two stores compared to going to just one store, even if the price data is incorrect.

3.5 Conclusion

A societal grocery shopping system as described in this chapter would be useful and practical, because it helps customers obtain a savings of 21% or more according to our simulation. Even with deceptive pricing by stores or incorrect price data reported by other customers, it will still be helpful for obtaining some savings. During the simulation, we considered all the parameters that may vary in real shopping experiences: customer location, store location, item price, item number, and customer input. We varied the parameters to explore this five-dimensional space and produced results consisting of the average savings achieved by customers. To validate our results further, we also used real price data in a simplified version of our simulation containing fewer stores and shopping at just two of them. The results indicate an average savings of 6.7% by choosing the best two stores. Even with incorrect price data, customers can still save 6.3% on average. An implementation of our approach would require a social infrastructure where customers could report prices they discovered and find prices reported by others. Based on both simulated and real data, and the expected costs of such an infrastructure, our system would be useful and cost-effective in practice.

Chapter 4

Simulating a Societal Information System for Healthcare

4.1 Introduction

This chapter concerns the simulation of a societal information system in the domain of healthcare. A societal information system is a large-scale information system that gathers information from hundreds or thousands of individual entities. Such systems can be abstracted as graphs with nodes representing individual entities and edges representing relationships between them. The purpose of a societal information system is to affect the behavior of a node by means of information retrieved from other nodes. Nowadays, a person's behavior is influenced by social networking services, such as Facebook. However, the amount of information to be comprehended and utilized in such services can be overwhelming for users. To further automate sharing and processing of information within a large social network or a societal information system, we are investigating supporting each node in the network by a software agent. Software agents are autonomous computational entities that can be viewed as perceiving their environment through sensors and acting upon their environment through effectors. To say that agents are computational entities simply means that they physically exist in the form of programs that run on computing devices. To say that they are autonomous entities means that to some extent they have control over their behavior and can act without the intervention of humans or other systems. Agents pursue goals or carry out tasks in order to meet their design objectives, and in general these goals and tasks can be supplementary as well as conflicting [26] [73]. Agents can form commitments and act on behalf of individuals and form multiagent systems (MAS). We view agent-based societal information systems as multiagent systems.

Societal information systems are appropriate for a wide variety of problems, including regulation (e.g., banking), allocation of scarce resources (e.g., electric power and parking spaces), distributed situation assessment (e.g., urban air quality), system control (e.g., traffic management, both vehicular and telecommunication), and decentralized decision-making (e.g., choosing medical care). This article addresses simulating a societal information system in the area of decentralized decision-making for healthcare.

Healthcare decision-making is done in many developed countries in the context of a healthcare quadruple, which consists of (1) patients, (2) healthcare providers (hospitals, health centers, labs, etc.) and provider networks, (3) insurance companies, and (4) the government. There is a variety of information systems available to support healthcare providers, provider networks, government healthcare agencies, and insurance companies, but none to support patients. Because patients are naturally distributed and are typically willing to assist each other, societal agent-based information systems instead of centralized information systems would be appropriate for fostering this mutual assistance. In such systems, each patient would be represented by a software agent. The agent would assist its principal in health-related activities, such as understanding and interpreting insurance rules, finding the most cost-effective insurer, finding a good healthcare provider, providing advice on cost-effective drugs and care, and monitoring the spread of disease symptoms and their treatments. Feedback and information sharing among patients would be used extensively in such systems.

Investigating societal information systems for healthcare is a broad research area. Moreover, it is difficult to experiment with such information systems in a society, especially because patients' health, privacy, and rights must be considered. We therefore have relied on simulations for prototyping and evaluating.

This chapter is organized as follows. First, we explain the method we use for prototyping the societal healthcare information system - agent-oriented modeling. Second, we describe briefly how agent-oriented modeling is applied to design a simulation of a societal information system for healthcare running on the NetLogo platform. Third, we analyze and explain the simulation results. We conclude by comparing the outcomes of using different strategies in the healthcare system and discussing the benefits of a societal information system for healthcare.

4.2 Related Work

Multiagent systems are widely used in different areas, such as tracking goods, traffic control, consensus knowledge, and decision-making [25]. One of the interesting areas for applying a MAS is healthcare.

Nealon and Moreno [45] analyze features of healthcare problems, including the distributed nature of the knowledge that is needed to solve a problem, coordination, complexity, and so on. They claim that a MAS is an appropriate approach to tackle healthcare problems and could be used for patient scheduling, organ and tissue transplant management, community care, information access, and decision support systems. Isern et al. [29] compare the internal architecture and communication-based coordination techniques of fifteen healthcare-related agent-based systems and claim that agent-based systems increase reusability, flexibility, and other beneficial qualities as compared with centralized software systems, such as client-server systems.

MASs are also broadly used in home-care systems. Koutkias et al. [35] present a MAS for monitoring and detecting important cases for disease management. Isern et al. [28] describe the K4Care Home Care model, which uses an agent-based platform. Charfeddine [11] introduces an agent-oriented framework to simulate the population of a chronic disease.

In the work most closely related to ours, Udupi and Singh [66] use conceptual models in a societal information system to implement a peer-to-peer network in which an agent contacts other agents to discover suitable service providers. It uses InterPol, a language and framework for supporting different kinds of interaction policies between agents. We described the modeling method of our societal information system in [61].

There are several websites, similar to RateMDs [27], where people rate doctors according to punctuality, medical knowledge, and other characteristics, and add comments. As we explain later, our approach differs from such websites and has advantages.

4.3 Methodology

We focuses on designing societal information systems of a particular kind - societal information systems for finding an appropriate physician and finding out the benefits to do so. We use the case study method [61] and explore by rapid prototyping the design of a simulation of a societal information system for healthcare. Rapid prototyping stands for implementing a proof-of-concept prototype in an agile way by directly mapping the modeling constructs to the constructs of a scripting environment like Netlogo or some agent-oriented environment like JADE. The method we use for prototyping is agent-oriented modeling. Agent-oriented modeling as described in [59] is a holistic approach for analyzing, designing, and rapid prototyping of societal information systems consisting of humans and technical components. We have chosen agent-oriented modeling because it is geared towards prototyping distributed systems that are open, adaptive, and intelligent. Societal information systems are open systems because members of the society (e.g., commuters, patients, or shoppers) may join and leave the system at any time. Societal information systems are adaptive systems, because they should react to their constantly changing environment, which

Table 4.1 The model types of agent-oriented modeling

	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behavior
Analysis	Role models and organization model	Domain model	Goal models
Design	Agent models and interaction models	Knowledge models	Behavioral scenarios
Prototyping	Interaction prototyping	Information prototyping	Behavior prototyping

for example can take the form of changes in traffic infrastructure, health insurance coverage, and product prices. We also term societal information systems as intelligent systems, because they reflect the "wisdom of crowds" when recommending a patient, for example, a healthcare provider. In addition, agent-oriented modeling meets well the requirements for purposefulness and understandability of the design.

A set of canonical models are introduced in agent-oriented modeling, whose types are shown in Table 4.1. In addition to representing each model with an abstraction layer (analysis, design, or prototyping), Table 4.1 maps each model to the vertical viewpoint aspect of interaction, information, or behavior. Each cell in the table represents a specific viewpoint. We explain these viewpoints in the following paragraphs.

From the viewpoint of interaction analysis, role models represent the properties of roles and the relationships between the roles are represented by an organization model. From the viewpoint of information analysis, a domain model represents the knowledge to be handled by the societal information system. From the viewpoint of behavior analysis, a goal model is a container of three components: goals, quality goals, and roles.

From the viewpoint of interaction design, agent models transform the abstract constructs from the analysis stage, roles, to design constructs, agent types, which will be realized in the implementation process. Interaction models are used to express interaction patterns between agents. From the viewpoint of information design,

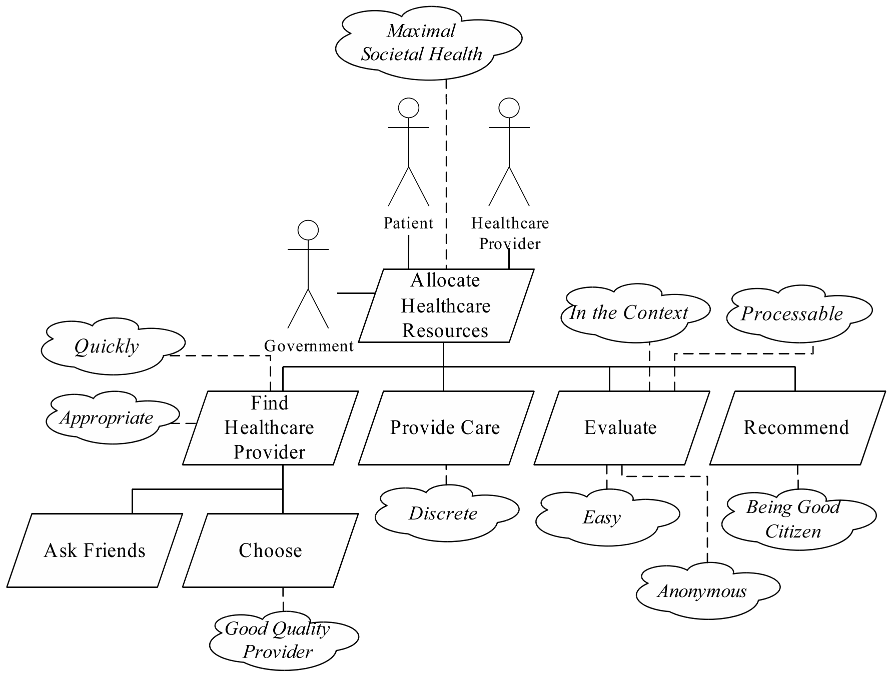

Figure 4.1 The goal model of the healthcare system

knowledge models represent both private and shared knowledge of agents. From the viewpoint of behavior design, behavioral scenarios are used to show how agents make decisions and perform activities [62].

Modeling at the abstraction layer of prototyping is explained in section 4.4.

Figure 4.1 shows the goal model of our societal information system for healthcare, in which rectangles stand for functional goals and clouds stand for quality goals. The stick figures represent roles that are required for achieving the goals. As can be seen from Figure 4.1, from the viewpoint of behavior analysis, our societal healthcare information system focuses on the purpose of "Allocate Healthcare Resources" among the members of the society. Specifically, we study the allocation of physicians - a special kind of healthcare resource. Achieving the functional goal "Allocate Healthcare Resources" is characterized by the quality goal "Maximal Societal Health", which determines the quality criterion according to which healthcare resources should be allocated in a society.

To accomplish the purpose "Allocate Healthcare Resources" of the societal information system, its four subgoals need to be achieved: finding a healthcare provider, being provided with care, evaluating the care, and recommending healthcare providers to other patients. As we demonstrate below, to fulfill the goal "Find Healthcare Provider", a patient recursively asks her friends, friends' friends, and so forth for recommendations and chooses the best physician recommended. This is represented as two subgoals of "Find Healthcare Provider:" "Ask Friends" and "Choose."

We attach a number of quality goals to the functional goals in the goal model. The meanings of the quality goals are easy to understand. For example, "Quickly" means a patient wants to find a healthcare provider as soon as possible. The "Anonymous" quality goal expresses that no evaluation by a patient should identify the patient. It should be noted that the quality goal "In the Context" attached to the functional goal "Evaluate" represents that evaluation has to occur in the context of receiving the service, preferably before leaving the facilities of the healthcare provider or at least on the same day. The "Processable" quality goal means that the evaluation should be presented in a form amenable to computer processing. In our simulation, we use a scale from 1 to 5 to measure the evaluations.

According to Figure 4.1, we model two roles for our simulation - Patient and Healthcare Provider. There is also a third role - Government. Since our work focuses on the particular aspect of the U.S. healthcare domain dealing with how a patient finds a physician, rather than modeling the healthcare domain in its full complexity, the Government role's modeling is not relevant to the simulation system being designed and we ignore the Government role in our system. Additionally, we complement the goal model with the new Assistant role, which is not shown in Figure 4.1. The Assistant role is the assistant of a person and is responsible for asking friends for recommendations, choosing a healthcare provider, and assisting in evaluating the care. In the prototypical system being designed, the role of Assistant should obviously be mapped to the Assistant Agent software agent type. Since a patient is a real human that is treated by another real human - a physician - we map both the roles Patient and Healthcare Provider to the Human Agent type. The software system boundary of the societal information system is obviously between the roles Patient and Assistant.

From the viewpoint of interaction analysis, the organization model of the societal information system being designed is decided based on the three kinds of networks that are used for representing the relationships among the members of the society:

- Random network: the relationships between pairs of patients are created randomly.

- Small-world network: most nodes are not neighbors to one another, but most nodes can be reached from any other node by a small number of hops [71].

- Scale-free network: the shortest paths between nodes flow through hubs, and if a peripheral node is deleted, it is unlikely that this will interfere with passing a message between other peripheral nodes. We use the Barabási-Albert model [5] to construct a scale-free network for our simulation. A scale-free network is a common model for a collaboration network.

After covering the viewpoints of behavior analysis and interaction analysis, we next proceed to the viewpoint of information analysis by addressing the knowledge to be represented within the system. We do this by identifying the types of knowledge entities related to the roles. As each healthcare provider has predefined capacity and efficiency, which are explained in section 4.4, we attach the Capacity and Efficiency knowledge entity types to the Healthcare Provider role.

We now proceed to the viewpoint of interaction design. Finding a physician involves interactions between Assistant Agents representing patients. We represent these interactions as an interaction protocol between agents of the type Assistant Agent. It is appropriate to remind here that the difference between an interaction protocol and other kinds of interaction models is that an interaction protocol models some aspects of the agent behaviors along with their interactions [59].

Representing the interaction protocol of the societal healthcare information system is very important, because it describes the patient's strategy of choosing a physician. We explored the following four possible strategies:

- Random strategy. The patient's Assistant Agent randomly chooses a physician.
- The "Choose one" strategy. The patient's Assistant Agent chooses the best physician according to the patient's evaluations for physicians. If the patient has no evaluations, his/her Assistant Agent asks his/her friends' Assistant Agents for recommendations. The Assistant Agent acting on behalf of the patient's friend may deal with the request in one of the following ways:
  - Reply with a recommendation.
  - Provide the requesting agent with the address of the Assistant Agent of one of its principal's friends if there is no recommendation to give. This process continues recursively until the first recommendation is received or until all the friends down to the maximum forwarding depth have been asked. The forwarding depth is defined as follows: the originator's friends are at depth 1; the originator's friends' friends at depth 2, and so on.

  Figure 4.2 presents the interaction protocol among patients' Assistant Agents for the "Choose one" strategy. It models that the Assistant Agent of a patient's friend may respond with a recommendation or recommend the Assistant Agent of the friend's friend. This means the interaction protocol is recursive, which is represented by the "Loop" behavioral construct. A friend's Assistant Agent may also ignore a request, which is not shown in the figure.

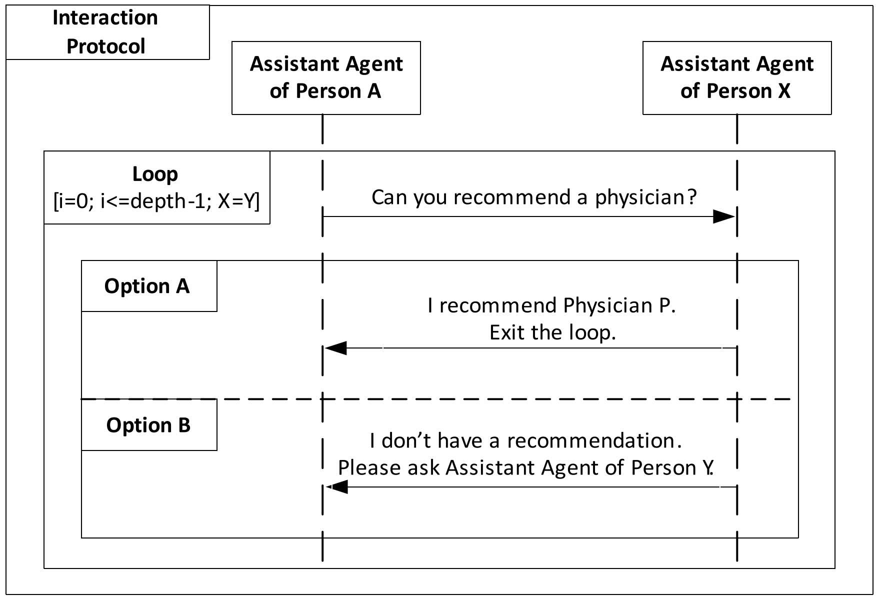

Figure 4.2 The interaction protocol for "Choose one" strategy

In addition to the random and "Choose one" strategies, we have included in our simulations the "Borda voting" and "Add and minimize" strategies. These strategies are briefly described as follows:

- The "Borda voting" strategy. The patient’s Assistant Agent asks his/her friends’ Assistant Agents, who are closer than a specified limit, for recommendations. A friend’s Assistant Agent may choose to answer or refrain from answering just like with the "Choose one" strategy. After the patient’s Assistant Agent has received all the responses, it calculates for each physician the Borda count [a single-winner election method in which voters rank candidates in order of preference, named for the 18th-century French mathematician and political scientist Jean-Charles de Borda, who devised the system in 1770], according to which a physician is given a number of points equal to the number of physicians whose evaluations are worse than the evaluations of the given physician. Thereafter the agent adds up all the points gained by the physician in question. The physician with the highest score is chosen.

- The "Add and minimize" strategy, which has the same procedure for getting recommendations as the "Borda voting" strategy. After the patient's Assistant Agent has received all the responses, it adds up all the nonzero evaluations and calculates the mean value of them for each physician. Then the Assistant Agent chooses the physician with the minimum mean evaluation. Choosing the physician with the minimum value is due to the way we define the evaluation, as described in section 4.4.

From the viewpoint of behavior design, to model the behaviors of agents of the decided types, we transform responsibilities of the roles into activities attached to the agent types. As a result, we obtain behavioral scenarios for agents playing the roles Patient, Assistant, and Physician. For example, the behavioral scenario of an agent of the type Assistant Agent playing the role Assistant models that the activities "Find a physician" and "Evaluate" are performed sequentially. In societal information system for healthcare this is always the case, because the Assistant Agent does not perform any activities between these activities while a patient is attended by a physician.

Another aspect of the Assistant Agent's behavior in choosing a physician deals with what the agent should do if the physician is not available on the given day. We have decided to consider the following three waiting strategies of a patient:

- Waiting. The patient's Assistant Agent chooses the best physician by adopting one of the strategies of choosing a physician explained above and sticks to this choice. If the physician is busy, the patient will still make an appointment with the physician and will wait until the physician becomes available.

- No waiting. If the physician chosen is busy, the patient's Assistant Agent will choose a physician randomly according to the "Random" strategy or the next best physician according to the other physician-choosing strategies until it finds an available physician.

- Waiting with limit. If the physician chosen is not available, the patient's Assistant Agent will check whether the physician could be reached in a certain number of days. If it is possible, the patient will make an appointment and wait. If not, the Assistant Agent will choose another physician according to the rules of the same waiting strategy. If no physician is available in a certain number of days, the Assistant Agent will choose a physician who has the smallest number of days required for waiting.

Finally, distinguishing between private and public knowledge entities from the viewpoint of information design is straightforward, because the knowledge entity Evaluation is private to the patient and Assistant Agent helping him/her, while the knowledge entity Recommendation is shared between different patients and instances of Assistant Agent. Similarly, the knowledge entity Efficiency is private to each Healthcare Provider, but at the same time naturally forms a basis for how patients evaluate healthcare providers. We describe models including role models, the organization model, and the domain model in detail in [61].

4.4 Evaluation

Simulation Settings

We next describe from the three viewpoints introduced in section 4.3 how we mapped agent-oriented models of the societal information system to the programming constructs of the simulation environment.

From the viewpoint of information prototyping, we represented the knowledge entities decided by agent-oriented modeling as described in section 4.3 as follows:

- The Capacity knowledge entity - in terms of the number of patients per day that a given physician can handle.

- The Efficiency knowledge entity - in terms of the number of days that it takes for a given physician to cure a patient. This number of days is generated for each physician according to a normal distribution whose mean and standard deviation can be adjusted in the user interface.

- The Evaluation knowledge entity - in terms of the following variables:

  - The number of days the physician in question failed to handle a given patient. How this value is determined is explained below.

  - The number of days that the physician needed to cure a patient. This is determined by the Efficiency knowledge item pertaining to the physician.

  - A random component representing that different patients evaluate the same physician differently.

A patient's evaluation for a specific physician is calculated by adding these three factors. For example, let us assume that a patient gets sick today and decides to visit a physician chosen by her Assistant Agent, but the physician is busy and cannot see the patient until tomorrow. In this case, the value of the first factor - the number of days the physician in question failed to handle a given patient - is 1, because the patient had to wait for 1 day to see the physician. The second factor - the number of days that the physician requires to cure the patient - is a fixed number related to the physician in question. The third factor - the random component expressing the subjective factor - is a random value that varies between -0.5 and 0.5.

The viewpoint of behavior prototyping covers the behaviors of software agents representing patients and physicians. In accordance with the behavioral scenarios modeled as a part of the design described in section 4.3, every day the patients each try to decide which physician to visit. For each patient, the Assistant Agent acting on behalf of its principal may ask Assistant Agents of the principal's friends for recommendations and then makes a decision as to which physician the principal should visit.

From the viewpoint of interaction prototyping, the exchange of messages to be implemented is modeled according to interaction diagrams, such as the one in Figure 4.2 for the case of choosing a physician according to the "Choose one" strategy. To make our simulations more realistic, we have chosen a 20% probability that a friend would ignore the patient's request.

Back to the viewpoint of behavior prototyping, the software agent corresponding to the Assistant Agent recommends physicians based on evaluations. The agent can recommend only those physicians that its principal has actually visited in the simulation. The number of days the physician in question could not handle the given patient, because of the physician's exceeded capacity, accumulates in the patient's evaluation until the patient actually visits the given physician. On each new visit the agent "forgets" its previous evaluation and updates its knowledge base with the new evaluation. The reason why the agent forgets its previous evaluation is that during the time period between the previous evaluation and the new evaluation, factors that influence the evaluation may have occurred. For example, the physician may have become more skilled. Therefore it is fairer to use the latest evaluation.

To make our simulations as realistic as possible, we used the following statistical data by the Centers for Disease Control and Prevention (CDC) from the year 2008 [21]:

- The number of physician office visits per 100 people per year: 320.1.

- The number of physicians per 10,000 people: 26.

Based on the above data, we obtained the average number of people who get sick every day by dividing the number of visits per 10,000 people by 250, which is the

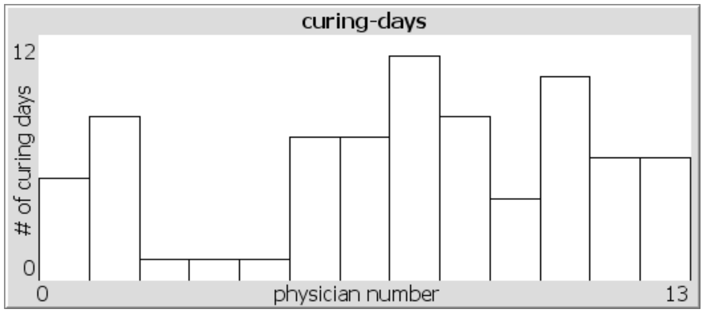

Figure 4.3 Days needed to be cured by different physicians

standard number of working days in a calendar year in the U.S. As a result, 128 people out of a population of 10,000 get sick every day.

Results and Evaluation

We simulated 365 days with 5,000 patients and 13 physicians. In our simulation, 64 random people get sick every day. The value of the local variable of each physician's software agent corresponding to the Capacity knowledge entity was set to 8 patients per day. The value of the local variable of each physician's software agent corresponding to the Efficiency knowledge entity was determined randomly according to a normal distribution with mean value 3 days and with the value of deviation as 2.0.

Figure 4.3 describes the number of days needed for curing by different physicians in our simulations.

We performed simulations by combining the types of social networks with different strategies of choosing a physician and waiting strategies, which are described in section 4.3. The results from simulations in terms of the annual sick days per person and leftover patients who were not taken care of by the end of the last day simulated are represented in Tables 4.2 - 4.7.

We can see from Tables 4.2 - 4.7 that if a patient adopts the "Waiting" strategy, the "Random" strategy will outperform all the other strategies of choosing a physician in social networks of all three kinds addressed. This is the case because in all the

Table 4.2 Average sick days for random network

	Waiting	No-waiting	Waiting-with-limit
Random	6.8	6.78	6.80
Choose one	12.18	5.21	6.34
Borda voting	34.75	6.01	8.31
Add and minimize	10.17	4.94	6.37

Table 4.3 Leftover patients for random network

	Waiting	No-waiting	Waiting-with-limit
Random	0	0	0
Choose one	427	0	80
Borda voting	4569	0	96
Add and minimize	155	0	94

Table 4.4 Average sick days for small-world network

	Waiting	No-waiting	Waiting-with-limit
Random	6.79	6.78	6.79
Choose one	10.31	5.22	6.31
Borda voting	7.63	6.58	6.83
Add and minimize	9.59	4.96	6.34

Table 4.5 Leftover patients for small-world network

	Waiting	No-waiting	Waiting-with-limit
Random	0	0	0
Choose one	407	0	71
Borda voting	76	0	25
Add and minimize	239	0	93

Table 4.6 Average sick days for scale-free network

	Waiting	No-waiting	Waiting-with-limit
Random	6.79	6.74	6.77
Choose one	13.72	5.29	6.33
Borda voting	38.56	6.49	9.92
Add and minimize	9.41	4.92	6.35

other strategies, a patient always waits for the best physician chosen by her Assistant Agent, which increases the waiting days and accordingly sick days. Also, because the Random strategy leads to even visiting of physicians, it has no leftover patients, but there are leftover patients in the other three strategies.

Table 4.7 Leftover patients for scale-free network

	Waiting	No-waiting	Waiting-with-limit
Random	1	0	2
Choose one	686	0	78
Borda voting	4449	0	110
Add and minimize	255	0	96

The performance of the "Borda voting" strategy is the worst in all three kinds of social networks addressed, except for the combination of the "Borda voting" strategy and "Waiting" strategy in the small-world network, because it uses more evaluation information than the other strategies due to its method for calculating the votes for physicians.

Differently from the "Borda voting" strategy, the "Add and minimize" strategy uses less information because it does not consider the physicians who have not been evaluated. The patients whose Assistant Agents follow the "Add and minimize" strategy therefore tend to choose physicians with fewer days required for curing, as compared with other physician choosing strategies, and then wait for that physician chosen, which increases the number of sick days.

If the "No waiting" strategy is adopted, all the other strategies will outperform the "Random" strategy. This is because the patients' Assistant Agents consider the evaluations by their principals' friends and choose the best physicians, and there is no problem of waiting.

If the "Waiting with limit" strategy is adopted, the "Choose one" and "Add and minimize" strategies of choosing a physician show the best performance. However, these strategies result in more leftover patients than the random strategy. This is reasonable, since according to "Choose one" and "Add and minimize" strategies a patient may be willing to wait for a good physician if the waiting time is less than 2 days, leading to just a few leftover patients and less average annual sick days.

According to the "Random" strategy of choosing a physician, patients' Assistant

Table 4.8 Average sick days with seven physicians

	No-waiting	Waiting-with-limit
Random	28.01	19.23
Choose one	26.15	19.24
Add and minimize	18.33	19.24

Agents just choose physicians randomly and each physician has almost the same number of patients in total. For the other three strategies, as time passes, Assistant agents gradually gather enough information about physicians, evaluate them, and recommend to their friends the best physicians they are aware of. As a result, after patients have formed their opinions about the physicians, good physicians get full capacity of patients every day and bad physicians get only a few patients. Due to space limitations, the graphs showing this trend are not shown here.

We also performed simulations with fewer physicians to check whether the claims stated above still hold. We adopted the "Choose one" and "Add and minimize" strategies of choosing a physician and "No waiting" and "Waiting with limit" strategies for conducting simulation experiments with 7 physicians. Table 4.8 shows the results in terms of average sick days. We can see that for the "No waiting" strategy, the "Choose one" and "Add and minimize" strategies of choosing a physician still perform better than the combination of "Random" and "No waiting" strategies. This is because a patient's Assistant Agent first chooses a physician who requires less days for curing and only then randomly chooses a physician if the patient has to wait.

Table 4.8 also shows that the combination of the "Waiting with limit" strategy and all strategies for choosing a physician yields almost the same result.

In addition, we investigated the performance of a system having a lower probability that the friends of a patient in small-world network will answer a request, which are shown in Tables 4.9 and 4.10. The "Random" strategy is shown here because it is not influenced by the probability.

Comparing Table 4.4 with these two tables, we discovered that the conclusions

Table 4.9 Average sick days with probability = 0.6

	Waiting	No-waiting	Waiting-with-limit
Choose one	9.86	5.31	6.29
Borda voting	7.60	6.26	6.60
Add and minimize	9.75	5.01	6.37

Table 4.10 Average sick days with probability = 0.4

	Waiting	No-waiting	Waiting-with-limit
Choose one	8.72	5.42	6.24
Borda voting	7.05	6.07	6.42
Add and minimize	10.38	5.14	6.40

Table 4.11 Changing trend with decreasing probability

	Waiting	No-waiting	Waiting-with-limit
Choose one	-	+	-
Borda voting	-	-	-
Add and minimize	+	+	+

before still hold. To clarify this, we include Table 4.11, which denotes the changing trend of the average sick days while the probability is decreasing. In the table, "+" means increasing and "" means decreasing. We can see from the table that for "Add and minimize", the average sick days increases while the probability decreases, because patients gets fewer responses from friends, leading to less informed decisions. For the "Borda voting" strategy, the average sick days decreases while the probability decreases. As mentioned before, due to the way that the "Borda voting" strategy gets the evaluation and calculates, it always gets too much information and lead to worse results than other strategies. So when the information is less with decreasing probability, we have better results of less average sick days. There's no fixed trend for a certain waiting strategy with different physician choosing strategies.

4.5 Conclusion

This chapter describes the design and rapid prototyping of a societal information system for healthcare. Agent-oriented modeling was chosen for developing our simulation because it explicitly addresses the design of societal information systems where the activities of humans are supported by software agents.

We investigated the prototyped societal healthcare information system using agent-based simulations on the NetLogo platform. In the simulation, we investigated the influence of different strategies of finding an appropriate physician and different waiting strategies in three common social network models. Our prototype revealed that if a patient adopts the "Waiting" strategy, the "Random" strategy will outperform all the other strategies of choosing a physician. On the other hand, with the "No waiting" strategy, all the other strategies will outperform the "Random" strategy. If the "Waiting with limit" strategy is adopted, the "Choose one" and "Add and minimize" strategies will show the best performance. We found that by adopting the "Choose one" or "Add and minimize" strategy, in the "No-waiting" and "Waiting-with-limit" case, the average number of sick days can be reduced by 0.42 - 1.84 days or 6.2% - 27.1%.

Our societal information system differs from RateMDs and other similar websites where people can rate and find physicians in the way that people rate the physicians and patients interact. It is difficult to compare the effect of these websites and that of our system, mostly because there are no objective evaluation statistics on the websites, such as the length of time each patient takes to get cured during a period. Such websites use more flexible criteria on which different people might have different opinions, such as punctuality, medical knowledge, and time spent on a patient, while our system uses the time it takes to cure a patient as a criterion, which is more objective and meaningful. Although patients might access more ratings online, they usually do not know the people who have rated the physicians and there is a higher possibility that the ratings will not be truthful and accurate. In our system, a patient relies on friends' recommendations, which are typically more reliable.

Chapter 5

Determining the Effect of Personality Types on Human-Agent Interactions

5.1 Introduction

Agents are used nowadays to help with people's everyday life in many ways. For example, an agent could help travelers find the cheapest ticket for a specific flight, or get elders their medications. Thus, it is not surprising that people have feelings about agents. It is reported that humans show empathy towards robots [39], evidenced by measuring their emotional and neurological change when they watched videos of dinosaur robots being abused. However, people's feelings towards agents are not always positive. There's a long-existing controversy about how the agents would behave after they have too much intelligence. Some people are afraid that robots, which are a kind of agents, might kill humans if they are intelligent enough and their interests conflict with humans' interests, despite the rule of "A robot may not injure a human being or, through inaction, allow a human being to come to harm", as stated in "The Three Laws of Robotics" [3]. Along with the technology development of many different kinds of agents, questions have risen: will humans behave preferentially towards other humans or agents? It is known that humans' personality types have impact on interactions between humans, but how about the human-agent interaction? Will a human's personality type have an impact on his/her decisions regarding other humans and agents? If we discover some relationship between personality types and decisions, how could we use these information to help with everyday life? In order to

Table 5.1 MBTI Dichotomies

Extraversion (E) - Introversion (I)
Sensing (S) - iNtuition (N)
Thinking (T) - Feeling (F)
Judging (J) - Perception (P)

Table 5.2 KTS-II dimentions

Abstract Cooperator (Idealist)	Concrete Cooperator (Guardian)
Abstract Utilitarian (Rational)	Concrete Utilitarian (Artisan)

answer these questions, we must determine a human's personality type first.

Personality Types

There are different methods to test personality types. A famous psychometric questionnaire to reveal a person's personality type is the Myers-Briggs Type Indicator (MBTI) assessment [44]. Myers used four dichotomies in MBTI theory, as shown in Table 5.1.

The result of the MBTI questionnaire is a four-letter personality type, with one letter coming from one of the four dichotomies. For example, a person with type INFP means he/she is introverted, intuitive, friendly, and more likely to probe the environment.

We chose the Keirsey Temperament Sorter-II (KTS-II) [34], which is closely associated with MBTI. KTS-II classifies people into four temperament groups according to two basic dimensions of personality: what people say (communication) and what people do (action). There are two types of communication: concrete people talk about reality while abstract people talk about ideas, shown by the columns of Table 5.2. Similarly, there are two types of action: cooperative and utilitarian. Cooperative people do what's right and utilitarian people do what works, as shown by the rows in Table 5.2.

The temperaments are Artisan, Guardian, Rational, Idealist, whose names come from Plato's book *The Republic*. They each have different traits [33]:

- *Idealists* speak mostly of what they hope for and imagine might be possible for people, and they want to act in good conscience, always trying to reach their goals without compromising their personal code of ethics. Examples of the Idealists are Mohandas Gandhi and Princess Diana.

- *Guardians* speak mostly of their duties and responsibilities, of what they can keep an eye on and take good care of, and they're careful to obey the laws, follow the rules, and respect the rights of others. Examples of the Guardians are George Washington and Mother Teresa.

- *Rationals* speak mostly of what new problems intrigue them and what new solutions they envision, and always pragmatic, they act as efficiently as possible to achieve their objectives, ignoring arbitrary rules and conventions if need be. Examples of the Rationals are Hillary Clinton and Stephen Hawking.

- *Artisans* speak mostly about what they see right in front of them, about what they can get their hands on, and they will do whatever works, whatever gives them a quick, effective payoff, even if they have to bend the rules. Examples of the Artisans are Michael Jordan and Marilyn Monroe.

Each temperament has four variants, as shown in the first two columns in Table 5.3. The third column in Table 5.3 shows the MBTI types corresponding to the KTS-II types. KTS-II describes behaviorial patterns while MBTI describes what people have in mind, which makes KTS-II suitable for our experiments in theory. For convenience, we sometimes use the letters from the MBTI dichotomies to denote the KTS-II personality types in this chapter.

Table 5.3 KTS-II types vs MBTI types

KTS-II temperament	KTS-II character types	MBTI types
Artisan (SP)	Promoter	ESTP
	Crafter	ISTP
	Performer	ESFP
	Composer	ISFP
Guardian (SJ)	Supervisor	ESTJ
	Inspector	ISTJ
	Provider	ESFJ
	Protector	ISFJ
Rational (NT)	Fieldmarshal	ENTJ
	Mastermind	INTJ
	Inventor	ENTP
	Architect	INTP
Idealist (NF)	Teacher	ENFJ
	Counselor	INFJ
	Champion	ENFP
	Healer	INFP

The Cake-Cutting Game

After the human subjects get their personality types through the KTS-II test, they are asked to play the "Who Gets More Cake?" game, which is related to the classic cake-cutting game.

In the classic cake-cutting game, players want to divide a cake in such a way that all of them believe they have received a fair amount of the cake. There are two basic measurements for a solution of the cake-cutting problem: fairness and envy-freeness. Fairness means anyone gets at least the amount that he believes is fair, while envy-freeness means anyone believes no one gets more than he has and he won't want to exchange his cake with others. If the cake is divided between two players, there is a fair and envy-free solution, which is to have one player cut the cake into two pieces and the other player choose his piece of the cake first. For three players, Selfridge-Conway discrete procedure [50] can be used to provide a fair and envy-free solution. However, our focus here is whether humans of different personality types act differently towards an agent, not dividing the cake perfectly with fairness and envy-freeness. We add a "leftover cake giveaway" part to the cake-cutting game in our "Who Gets More Cake?" game, which will be described in detail in section III.

The rest of the chapter is organized as follows. In section II, we introduce some related work. In section III and IV, the experiments are described in detail and the results are analyzed. In section V, we draw the conclusion.

5.2 Related Work

Reeves and Nass [49] claimed that people were inclined to treat media, usually computers in their studies, as if they were real people or real places. Thus we have the hypothesis that the personality types of humans would influence their behavior towards other humans and agents, just like in the interactions between humans.

Bartneck, Hoek, Mubin, and Mahmud [6] used "iCat" robots of different intelligent levels to test whether humans treat the robots differently. They showed that the robots' intelligence had a significant influence on the humans' decision in the measurement of their hesitation time to switch off the robot. While they investigated the influence of different intelligence levels towards humans' decision, we try to figure out whether the personality type of a human influences his decisions towards a person or an agent.

Many researchers who investigated the influence of personality types towards humans' decisions. For example, Schmitt, Shupp, Swope, and Mayer [56] used MBTI test to get personality types and let the human subjects play the ultimatum game. In the ultimatum game, there are two players: a proposer and a responder. The proposer first makes an offer on how to divide a given amount of money, then the responder could accept or reject. The money is divided according to the offer if the responder accepts, but none of them gets anything if the responder rejects. They discovered that the "Thinking (T)" types made lower offers than those characterized as "Feeling (F)" types and "Extraversion (E)" types indicated a willingness to accept offers that was less than "Introversion (I)" types. Peever, Johnson and Gardner [48] used the Five Factor Model to test the personality types and discovered the games a person preferred was related to his personality type.

Personality traits including those in Five Factor Model [15] and some other traits, such as public self-consciousness and shyness are considered by Von der Putten, Kramer, and Gratch [69]. In their study, subjects recruited through a website interacted with a virtual agent. They found that some personality traits, such as agreeableness, extraversion, approach avoidance, were related to humans' behavior, while some traits, gender, and age didn't affect the results.

We presented the questions of this chapter in the mixed human-agent society and some experimental results in [16] [17]. This chapter studies the impact of humans' personality types towards their behavior, while it is different from other studies because of three reasons:

- Other than MBTI or Five Factor Model, we used KTS-II test in our study, which broadens the domain of possible explanations of the influences that personality types could bring to human behavior.

- Many researchers considered the interaction between a person and an agent, sometimes just between humans, while we considered a human interacts with both a simulated human and an agent at the same time, showing the different aptitudes the human has towards the simulated human and the agent.

- We explored a different experimental setting from previous studies, which may bring new conclusions since conclusions based on previous studies might only be applied to certain studies. We developed a new game and try to figure out how humans would behave in this situation.

5.3 Experiment

As mentioned before, our experiment contains two phases:

- Test the subjects' personality types using KTS-II.

- The subjects play the "Who Gets More Cake?" game.

In our "Who Gets More Cake?" game, we have a cake for three players to divide. One player is the human subject/participant, one player is a simulated human, and the third player is an agent/robot (the robot has a way to convert the cake into the energy it needs to move). The participant was told he was playing with another person and a robot, but actually a simulated human and an agent for the reason of experimental control. Players indicate how they would like to cut the cake into three pieces, by drawing two lines/cuts on their own picture of the cake. We follow a protocol proposed by Iyer and Huhns [30], which is proved to be fair for dividing a resource among n agents, to decide how the cake is divided: whoever has drawn the left-most cut will get the left side of the cake from the edge to this cut. Of the remaining two players, whoever has drawn the right-most cut will get the right side of the cake from this cut to the right edge. The third player will get the portion in the middle indicated by that player's two cuts. Note that all players will get one of the pieces that they indicated, as proved in [30].

After the cake is divided, no player would want to trade with others, because they would get a piece that is smaller than the one they drew on the cake. However, there will be one or two portions of the cake left. To make the game more real, the participants were told one player would then be chosen randomly to give the remaining portions of the cake to one of the other players in each game. In fact, the participants were asked to whom they would give the leftover cake in every game. They could only give the leftover cake to either the simulated human or the agent, but not themselves. Each participant was asked to play the game three times, each time with a different cake and with a different simulated human. To play the part of a human as real as possible, our simulated human has different names in three games and their names are neutral to eliminate the bias of sex. At the beginning of each game, participants were asked to type a greeting sentence to the simulated human and the simulated human will type some greetings too. It takes our simulated human some time to think and draw cuts on the cake, each game with different amount of delay to mimic human thinking.

5.4 Results

73 non-computer science students with age around 20 who have little technological background participated in the experiment. They took the KTS-II personality test and played the game after being told the rules of the game. 58 of them played all three rounds of the game. In total, they played 197 games. We measure four criteria:

- The number of games in which the participants give the leftover cake to the simulated human, denoted by N_{human};
- The number of games in which the participants give the leftover cake to the agent, denoted by N_{agent};
- The number of participants who give the leftover cake to the same player (either the simulated human or the agent) in the three games they played, denoted by N_{same};
- The number of participants who give the leftover cake to different players in the three games they played, denoted by N_{diff}.

The first two criteria measure the tendency that a participant would like to choose either a person or an agent under some circumstances, which might indicates whether he would like to interact with a person or an agent, and the last two criteria measure the consistency of his choice. For the last two criteria, we only consider the participants who finished all three games.

To deal with the personality type results, we first need to understand how to interpret KTS-II test. KTS-II provides a questionnaire based on seventy questions, each with two options indicating the two aspects of a certain dichotomy. There are ten questions for E (Extraversion)-I (Introversion) dichotomy and twenty questions each for the other three dichotomies. A personality type depends on how many options you selected for the two aspects of each dichotomy. If a person choose the same number of options for the two aspects of any dichotomy, an "X" will appear for that dichotomy. For example, if a person choose 5 options for E (Extraversion) and 5 options for I (Introversion), his personality type will have an "X" in the E (Extraversion)-I (Introversion) dichotomy, such as XSTJ. If this happens, the person should read both ESTJ and ISTJ's descriptions and choose the one more like himself. In our experiments, a few participants has one or more "X"es in their personality types. We handle this by counting them as 1/2 person for one "X" situation for each possible type, 1/4 person for two "X"es situation for each possible type, and so on. For example, the above person with personality type XSTJ is counted as 1/2 person with type ESTJ and 1/2 person with type ISTJ.

In order to investigate how the personality types influence the choices the participants make, we introduce several statistic criteria to do evaluation:

- Pearson's chi-squared test (χ^2 test) or Fisher's exact test, which evaluates the degree of independence between two nominal variables.

- *Cramér's V* (V), which is an effect size measure of association between two nominal variables.

- Goodman and Kruskal's lambda, which help us to understand whether knowing a person's personality would help to predict his choice in the game (λ_1) and vice versa (λ_2).

Tendency Results

We calculated first two criteria for all the participants as a whole and have

$$N_{human} = 133,\ N_{agent} = 64. \tag{5.1}$$

The data shows the participants give the leftover cake to the simulated human in most games, which is twice as many as those in which it is given to the agent. We grouped the data by sixteen MBTI types. The data is not shown here due to space limits, which reveals that almost people of all the types give more leftover cakes to the humans than to the agents, which reveals their different aptitude towards the humans and the agents. Champion (ENFP), one of the Idealists, gives the leftover cakes to the simulated human 6 times than they give to the agent. On the other hand, Crafter (ISTP), one of the Artists, give more cake to the agent. The data is heterogeneous and it's hard to discover the pattern among all the sixteen personality types. That's one clue of suggesting us to group them in some way and analyze the results.

KTS-II Temperaments Tendency Results

Thus, we calculated the same criteria for the four KTS-II temperaments, as shown in Table 5.4 and criteria for each two aspects of the four dichotomies, as shown in Table 5.7.

Now we want to see whether the KTS-II temperaments have significant influence on the choices the participants made. Our data fits the conditions of Pearson's χ^2 test. Following the test procedure, we stated the null hypothesis as follows:

Table 5.4 Observed Frequencies of Four Temperaments

O_{freq}	Guardian	Artisan	Idealist	Rational	R_{total}
N_{human}	55.25	23.75	30	24	133
N_{agent}	26.5	13.5	11.5	12.5	64
C_{total}	81.75	37.25	41.5	36.5	197

Table 5.5 Expected Frequencies of Four Temperaments

E_{freq}	Guardian	Artisan	Idealist	Rational
N_{human}	55.19	25.15	28.02	24.64
N_{agent}	26.56	12.10	13.48	11.86

H_0: The participants' KTS-II temperaments and the choices they made are independent.

Then we represent the data in a contingency table as in Table 5.4, where R_{total} describes row total and C_{total} describes column total. The participants choices, as we observed, are called observed frequencies (O_{freq}) in statistics.

Our hypothesis is that there is no relationship between the participants' temperaments and their choices, which means they give the leftover cake to the simulated human or the agent randomly (i.e., with equal probability). Thus we get the expected frequencies (E_{freq}) proportionally, as shown in Table 5.5.

We use the following formula to calculate χ^2:

$$\chi^2 = \sum_{0<i<m,\, 0<j<n} \frac{(O_{freq}(i,j) - E_{freq}(i,j))^2}{E_{freq}(i,j)}, \tag{5.2}$$

where $O_{freq}(i,j)$ and $E_{freq}(i,j)$ denote the observed frequencies and expected frequencies in the table cell of ith row and jth column. m and n represents the total row number and total column number. Combined with degree of freedom $df = 3$, the statistical results are

$$\chi^2 = 0.72,\ P = 0.8685,\ V = 0.0606. \tag{5.3}$$

The meaning of the results is that we are $1-P$ (in the form of percentage) sure to reject the null hypothesis. Normally significant level of 0.05 or 0.1 is used, which means if

Table 5.6 Percentage Deviation of Four Temperaments

Percentage Deviation	Guardian	Artisan	Idealist	Rational
N_{human}	0.1%	-5.6%	7.1%	-2.6%
N_{agent}	-0.2%	11.6%	-14.7%	5.4%

$P < 0.05$ or $P < 0.1$ we can reject the hypothesis. In our case, $P > 0.05$ and there is 13% probability that we could reject the hypothesis, which is very low. Thus we can't reject the null hypothesis, which means we can't say there is a relationship between the participants' temperaments and their choices. V is an effect size measure which shows the inter-correlation of the variables. In this case, it measures the relationship between the participants' KTS-II temperaments with their choices. According to the convention, $V < 0.1$ means negligible relationship. In our case, $V = 0.0606$ means the association between the KTS-II temperaments and the choices is negligible.

Percentage deviation, which measures the degree to which observed frequencies differs from the expected frequencies, is calculated as follows:

$$PD(i,j) = \frac{O_{freq}(i,j) - E_{freq}(i,j)}{E_{freq}(i,j)}. \tag{5.4}$$

Table 5.6 shows the percentage deviation of the KTS-II temperaments' tendency results, from which we could see that people with different temperaments behave very differently. Artisans and Idealists are deviated more from the general public than the other two temperaments. The Guardians act just like an average person. By an average person, we refer to an imaginary person who will act as our reference data shows. For example, if this person plays our game for 197 times, he would probably end up with giving the leftover cake 133 times to the simulated human and 64 times to the agent.

At last we use Goodman and Kruskal's lambda to measure the proportional reduction in error. For example, in our case, the estimated probability of correct prediction when predicting a person's choice without knowing his temperament is

$$p_1 = \frac{133}{197} = 0.6751, \tag{5.5}$$

Table 5.7 Tendency Results and Percentage Deviation of Four Dichotomies

MBTI	N_{human}	N_{agent}	PD_{human}	PD_{agent}
E (Extraversion)	62	27	3.2%	-6.6%
I (Introversion)	71	37	-2.6%	5.5%
S (Sensing)	79	40	-1.7%	3.5%
N (iNtuition)	54	24	2.5%	-5.3%
T (Thinking)	58.5	35.5	-7.8%	16.2%
F (Feeling)	74.5	28.5	7.1%	-14.8%
J (Judging)	79.5	35.5	2.4%	-5.0%
P (Perception)	53.5	28.5	-3.4%	7.0%

while estimated probability of correct prediction when predicting what choice a person will make knowing his temperament is

$$p_2 = \frac{55.25 + 23.75 + 30 + 24}{197} = 0.6751. \tag{5.6}$$

Goodman and Kruskal's lambda of predicting choice on the basis of temperament is

$$\lambda_1 = \frac{(1 - p_1) - (1 - p_2)}{1 - p_1} = 0, \tag{5.7}$$

which means there is no difference whether or not knowing a person's temperament when predicting his choice. Also we found out lambda of predicting a person's temperament from his choice (λ_2) is 0, which means knowing a person's choice won't do any good to predicting the his temperament.

To give a hint of how the participants' choices of each dichotomy varies, table 5.7 shows the tendency results of four dichotomies, where PD_{human} is the percentage deviation of N_{human} and PD_{agent} is the percentage deviation of N_{agent}. We could see that the biggest difference from what is supposed to be with our equal probability assumption happens in the T-F dichotomy.

Then we investigated how MBTI dichotomies influence the choices the participants made. Following the same procedure, first we stated the null hypothesis for each dichotomy as follows:

- For E-I dichotomy: The participants' types in E-I dichotomy and their choices are independent;

Table 5.8 Statistical Results of Four Dichotomies for Tendency

Dichotomy	χ^2	P	V	λ_1	λ_2
E-I	0.34	0.5598	0.0417	0	0
S-N	0.17	0.6801	0.0297	0	0
T-F	2.28	0.1311	0.1077	0	0.07
J-P	0.33	0.5657	0.0409	0	0

- For S-N dichotomy: The participants' types in S-N dichotomy and their choices are independent;

- For T-F dichotomy: The participants' types in T-F dichotomy and their choices are independent;

- For J-P dichotomy: The participants' types in J-P dichotomy and their choices are independent.

Table 5.8 shows the statistic results for each dichotomy. From the table we could see that in T-F dichotomy, there is 87% possibility, which is close to the standard of rejecting the null hypothesis with a significance level of 0.1, to reject the null hypothesis. Still, we can't reject the null hypothesis, but we probably could see it get rejected with more experiments and draw a conclusion that the personality in T-F dimension has something to do with the participants' choices based on statistics. For other dimensions, there is no evidence to lead to the conclusion that we should reject the null hypothesis and say there is a relationship between a certain dichotomy and the choices.

Also, we could see from $Cram\acute{e}r's\ V$ there's a weak relationship between T-F dichotomy and the choices, and negligible relationship between any other dichotomy and the choices in the whole population based on our samples. λ_2 for T-F dichotomy is 0.07, which means that we could reduce 7% error when predicting a person's temperament with his choice known compared to that with his choice not known.

Consistency Results

Next, we measure the consistency of the participants' choices. First we calculated the consistency criteria:

$$N_{same} = 18,\ N_{diff} = 40. \tag{5.8}$$

We could see that more than two thirds of participants give the leftover cake to different players in three games, which means they don't always prefer the simulated human or the agent. Similar to the tendency results, we grouped the N_{diff} and N_{same} data according to temperaments and dichotomies, shown in the first three columns in Table 5.9 and Table 5.10.

It is not suggested to use Pearson's χ^2 test if there are small expected frequency values, so we use Fisher's exact test here to perform analysis similar to Pearson's χ^2 test for data in Table 5.9 and the result is

$$P = 0.9999. \tag{5.9}$$

We also perform Pearson's χ^2 test to get an approximate V value. Our null hypothesis is as follows:

H_0: The participants' KTS-II temperaments and the consistency results of their choices are independent.

The statistical results are as follows:

$$\chi^2 = 0.22,\ V = 0.0616,\ \lambda_1 = 0,\ \lambda_2 = 0. \tag{5.10}$$

It shows that there is no significant dependence between the participants' KTS-II temperaments and the consistency of their choices. A person's temperament has little association with the consistency of his choices. Knowing a person's temperament or the consistency of his choices won't do any help to the prediction of the consistency of his choices or his temperament.

Table 5.9 Consistency Results and Percentage Deviation of Four Temperaments

Temperaments	N_{same}	N_{diff}	PD_{same}	PD_{diff}
Guardian	7.75	16.5	3.0%	-1.3%
Artisan	3.25	7.5	-2.6%	1.2%
Idealist	3	8.5	-15.9%	7.2%
Rational	4	7.5	12.1%	-5.4%

Table 5.10 Consistency Results and Percentage Deviation of Four Dichotomies

MBTI types	N_{same}	N_{diff}	PD_{same}	PD_{diff}
E(Extraversion)	9	15.5	18.4%	-8.3%
I(Introversion)	9	24.5	-13.4%	6.0%
S(Sensing)	11	24	1.3%	-0.6%
N(iNtuition)	7	16	-1.9%	0.9%
T(Thinking)	8.5	20	-3.9%	1.8%
F(Feeling)	9.5	20	3.8%	-1.7%
J(Judging)	11.5	22	10.6%	-4.8%
P(Perception)	6.5	18	-14.5%	6.5%

Then we investigated how MBTI dichotomies influence the consistency of the choices that the participants made. Following the same procedure, first we stated the null hypothesis for each dichotomy as follows:

- For E-I dichotomy: The participants' types in E-I dichotomy and the consistency of their choices are independent;
- For S-N dichotomy: The participants' types in S-N dichotomy and the consistency of their choices are independent;
- For T-F dichotomy: The participants' types in T-F dichotomy and the consistency of their choices are independent;
- For J-P dichotomy: The participants' types in J-P dichotomy and the consistency of their choices are independent.

Table 5.10, where PD_{same} and PD_{diff} denote the percentage deviation of N_{same} and N_{diff}, gives us a hint of how the observed data deviates from what should be

Table 5.11 Statistical Results of Four Dichotomies for Consistency

Dichotomy	χ^2	P	V	λ_1	λ_2
E-I	0.64	0.4237	0.1054	0	0
S-N	0.01	0.9203	0.0105	0	0
T-F	0.04	0.8415	0.0257	0	0
J-P	0.4	0.5271	0.0833	0	0

with equal possibility assumption. E-I and J-P dichotomies deviates more than the other two dichotomies. The statistical results are shown in Table 5.11, based on which we couldn't reject any of the null hypothesis and say any dichotomy and the consistency results are not independent. Besides, knowing a person's dichotomies or the consistency of his choices won't do any help to the prediction of the consistency of his choices or dichotomies. However, $Cramér'sV$ shows there is a weak association between E-I dichotomy and the consistency of the choices.

5.5 Conclusion

In this chapter, we try to investigate whether humans' behavior towards other humans and agents is related to their personality types. We have seventy-three students participated in the experiments, by taking the KTS-II test and then playing the "Who Gets More Cake?" game.

We discovered that humans of different personality types behave differently towards other humans and agents. For example, Artisans and Idealists act more deviated from an average person; it's very likely that T-F dichotomy is not independent with the tendency results. This provides a clue in many agent-related applications. For example, an Idealist has to partner with an agent/robot as his personal assistant due to business reasons. As an Idealist, he is inclined to interact with humans more and agents less, thus he might choose an robot with less talking or interactions needed. In the next stage, we may discover agents of which personality type could cooperate well with a certain kind of person, which could be used in many domains, such as elder's personal care, team formation and so on.

Currently our experiments shows little clue of making predictions based on a person's personality. In the future, we expect that more students participate in an updated version of this experiment to draw more reliable conclusion based on our statistical criteria and explore other possibilities.

Bibliography

[1] Sunil Adhau, M. L. Mittal, and Abhinav Mittal, *A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach*, Eng. Appl. Artif. Intell. **25** (2012), no. 8, 1738–1751.

[2] E. Altman, K. Avrachenkov, and A. Garnaev, *Fair resource allocation in wireless networks in the presence of a jammer*, Performance Evaluation **67** (2010), no. 4, 338 – 349, Performance Evaluation Methodologies and Tools: Selected Papers from {VALUETOOLS} 2008.

[3] Isaac Asimov, *I, Robot*, Signet, New York, NY, USA, June 1956.

[4] Yoram Bachrach and Jeffrey S. Rosenschein, *Distributed multiagent resource allocation in diminishing marginal return domains*, Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (Richland, SC), AAMAS '08, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1103–1110.

[5] Albert-László Barabási and Réka Albert, *Emergence of scaling in random networks*, Science **286** (1999), no. 5439, 509–512.

[6] Christoph Bartneck, Michel van der Hoek, Omar Mubin, and Abdullah Al Mahmud, *"daisy, daisy, give me your answer do!": switching off a robot*, Proceedings of the ACM/IEEE international conference on Human-robot interaction (New York, NY, USA), HRI '07, ACM, 2007, pp. 217–222.

[7] JeffreyM. Bradshaw, Maarten Sierhuis, Alessandro Acquisti, Paul Feltovich, Robert Hoffman, Renia Jeffers, Debbie Prescott, Niranjan Suri, Andrzej Uszok, and Ron Van Hoof, *Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications*, Agent Autonomy (Henry Hexmoor, Cristiano Castelfranchi, and Rino Falcone, eds.), Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 7, Springer US, 2003, pp. 243–280 (English).

[8] J.M. Bradshaw, P.J. Feltovich, M.J. Johnson, L. Bunch, M.R. Breedy, T. Eskridge, Hyuckchul Jung, J. Lott, and A. Uszok, *Coordination in human-agent-*

robot teamwork, Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on, May 2008, pp. 467–476.

[9] Erik Brynjolfsson, AstridA. Dick, and MichaelD. Smith, *A nearly perfect market?*, QME **8** (2010), no. 1, 1–33 (English).

[10] Martin Buchholz, Dirk Pflüger, and Josiah Poon, *Application of machine learning techniques to the re-ranking of search results*, KI 2004: Advances in Artificial Intelligence (Susanne Biundo, Thom Frühwirth, and Günther Palm, eds.), Lecture Notes in Computer Science, vol. 3238, Springer Berlin Heidelberg, 2004, pp. 67–81.

[11] M. Charfeddine and B. Montreuil, *Integrated agent-oriented modeling and simulation of population and healthcare delivery network: Application to copd chronic disease in a canadian region*, Simulation Conference (WSC), Proceedings of the 2010 Winter, Dec 2010, pp. 2327–2339.

[12] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaitre, N. Maudet, Julian Padget, Steve Phelps, Juan A. Rodríguez-Aguilar, and P. Sousa, *Issues in multiagent resource allocation*, Informatica **30** (2006), no. 1, 3 – 31.

[13] WILLIAM J. CLANCEY, PATRICIA SACHS, MAARTEN SIERHUIS, and RON VAN HOOF, *Brahms: simulating practice for work systems design*, International Journal of Human-Computer Studies **49** (1998), no. 6, 831 – 865.

[14] D. Clark, *Shopbots become agents for business change*, Computer **33** (2000), no. 2, 18–21.

[15] J. M. Digman, *Personality Structure: Emergence of the Five-Factor Model*, Annual Review of Psychology **41** (1990), no. 1, 417–440.

[16] Hongying Du, *The effects of human personality on human-agent interactions (doctoral consortium)*, Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013) (Takayuki Ito, Catholijn Jonker, Maria Gini, and Onn Shehory, eds.).

[17] Hongying Du and M.N. Huhns, *Determining the effect of personality types on human-agent interactions*, Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, vol. 2, Nov 2013, pp. 239–244.

[18] Daji Ergu, Gang Kou, Yi Peng, Yong Shi, and Yu Shi, *The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment*, The Journal of Supercomputing **64** (2013), no. 3, 835–848 (English).

[19] A. Eryilmaz and R. Srikant, *Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control*, Networking, IEEE/ACM Transactions on **15** (2007), no. 6, 1333–1344.

[20] Michal Feldman, Kevin Lai, and Li Zhang, *A price-anticipating resource allocation mechanism for distributed shared clusters*, CoRR **abs/cs/0502019** (2005).

[21] Centers for Disease Control and Prevention, *Ambulatory care use and physician visits*, `http://www.cdc.gov/nchs/fastats/docvisit.htm`, Accessed Feb 6, 2014.

[22] Stan Franklin and Art Graesser, *Is it an agent, or just a program?: A taxonomy for autonomous agents*, Intelligent Agents III Agent Theories, Architectures, and Languages (Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings, eds.), Lecture Notes in Computer Science, vol. 1193, Springer Berlin Heidelberg, 1997, pp. 21–35.

[23] AmyR. Greenwald and JeffreyO. Kephart, *Shopbots and pricebots*, Agent Mediated Electronic Commerce II (Alexandros Moukas, Fredrik Ygge, and Carles Sierra, eds.), Lecture Notes in Computer Science, vol. 1788, Springer Berlin Heidelberg, 2000, pp. 1–23.

[24] J. Howe, *Crowdsourcing: A definition*, `http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html`, Accessed Feb 12, 2014.

[25] Michael N. Huhns, *From dps to mas to ...: Continuing the trends*, Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1 (Richland, SC), AAMAS '09, International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 43–48.

[26] Michael N. Huhns and Larry M. Stephens, *Multiagent systems*, MIT Press, Cambridge, MA, USA, 1999, pp. 79–120.

[27] RateMDs Inc., `http://www.ratemds.com/`.

[28] David Isern, Antonio Moreno, Gianfranco Pedone, and Laszlo Varga, *An intelligent platform to provide home care services*, Knowledge Management for Health

Care Procedures (David RiaÃśo, ed.), Lecture Notes in Computer Science, vol. 4924, Springer Berlin Heidelberg, 2008, pp. 149–160.

[29] David Isern, David Sánchez, and Antonio Moreno, *Agents applied in health care: A review*, International Journal of Medical Informatics **79** (2010), no. 3, 145 – 166.

[30] Karthik Iyer and Michael N. Huhns, *Multiagent negotiation for fair and unbiased resource allocation*, OTM Conferences (1) (Robert Meersman, Zahir Tari, Mohand-Said Hacid, John Mylopoulos, Barbara Pernici, Özalp Babaoglu, Hans-Arno Jacobsen, Joseph P. Loyall, Michael Kifer, and Stefano Spaccapietra, eds.), Lecture Notes in Computer Science, vol. 3760, Springer, 2005, pp. 453–465.

[31] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge, *A roadmap of agent research and development*, Autonomous Agents and Multi-Agent Systems **1** (1998), no. 1, 7–38 (English).

[32] Arkadiusz Kawa, *Simulation of dynamic supply chain configuration based on software agents and graph theory*, Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living (Sigeru Omatu, MiguelP. Rocha, José Bravo, Florentino Fernández, Emilio Corchado, Andrés Bustillo, and JuanM. Corchado, eds.), Lecture Notes in Computer Science, vol. 5518, Springer Berlin Heidelberg, 2009, pp. 346–349.

[33] D. Keirsey, *Keirsey's webpage on overview of the four temperaments*, 1996.

[34] ______, *Please understand me ii*, Prometheus Nemesis, 1998.

[35] V.G. Koutkias, I. Chouvarda, and N. Maglaveras, *A multiagent system enhancing home-care health services for chronic disease management*, Information Technology in Biomedicine, IEEE Transactions on **9** (2005), no. 4, 528–537.

[36] B. Krulwich, *The bargainfinder agent: Comparison price shopping on the internet*, 1996.

[37] Ho-Kyoung Lee, Young-Hoon Yu, Supratip Ghose, and Geun-Sik Jo, *Comparison shopping systems based on semantic web - a case study of purchasing cameras*, Grid and Cooperative Computing (Minglu Li, Xian-He Sun, Qian-ni Deng, and Jun Ni, eds.), Lecture Notes in Computer Science, vol. 3032, Springer Berlin Heidelberg, 2004, pp. 139–146.

[38] Kangbok Lee, Byung-Cheon Choi, Joseph Y. T. Leung, and Michael L. Pinedo, *Approximation algorithms for multi-agent scheduling to minimize total weighted completion time*, Inf. Process. Lett. **109** (2009), no. 16, 913–917.

[39] T. Lewis, *Humans show empathy for robots*, April 2013.

[40] Vitaliy Liptchinsky, Roman Khazankin, Stefan Schulte, Benjamin Satzger, Hong-Linh Truong, and Schahram Dustdar, *On modeling context-aware social collaboration processes*, Information Systems (2013), no. 0, –.

[41] Ahmad Kamran Malik and Schahram Dustdar, *An intelligent information sharing control system for dynamic collaborations*, Proceedings of the 8th International Conference on Frontiers of Information Technology (New York, NY, USA), FIT '10, ACM, 2010, pp. 30:1–30:6.

[42] ______, *Enhanced sharing and privacy in collaborative virtual teams.*, Journal of Information Assurance and Security **7** (2012), no. 2, 61 – 70.

[43] Winter Mason and Duncan J. Watts, *Financial incentives and the "performance of crowds"*, Proceedings of the ACM SIGKDD Workshop on Human Computation (New York, NY, USA), HCOMP '09, ACM, 2009, pp. 77–85.

[44] I.B. Myers and P.B. Myers, *Gifts differing: Understanding personality type*, Davies-Black Pub., 1980.

[45] John Nealon and Antonio Moreno, *Agent-based applications in health care*, in Applications of Software Agent Technology in the Health Care Domain , Whitestein Series in Software Agent Technologies, BirkhÃďuser Verlag, Verlag, 2003, pp. 3–18.

[46] Bureau of Labor Statistics, *Consumer price index*, http://www.bls.gov/cpi, U.S. Bureau of Labor Statistics Division of Information Services, Accessed 29 Jan 2014.

[47] ______, *Glossary*, http://www.bls.gov/bls/glossary.htm, U.S. Bureau of Labor Statistics Division of Information Services, Accessed 29 Jan 2014.

[48] Nicole Peever, Daniel Johnson, and John Gardner, *Personality & video game genre preferences*, Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System (New York, NY, USA), IE '12, ACM, 2012, pp. 20:1–20:3.

[49] Byron Reeves and Clifford Nass, *The media equation: how people treat computers, television, and new media like real people and places*, Cambridge University Press, New York, NY, USA, 1996.

[50] J. Robertson and W. Webb, *Cake cutting algorithms: Be fair if you can*, Ak Peters Series, Peters, 1998.

[51] Stuart Russell and Peter Norvig, *Artificial intelligence: A modern approach*, 3rd ed., Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.

[52] Eugene Santos, Feng Zhang, and PeterB. Luh, *Intra-organizational logistics management through multi-agent systems*, Electronic Commerce Research **3** (2003), no. 3-4, 337–364 (English).

[53] Benjamin Satzger, Harald Psaier, Daniel Schall, and Schahram Dustdar, *Auction-based crowdsourcing supporting skill management*, Inf. Syst. **38** (2013), no. 4, 547–560.

[54] Ognjen Scekic, Christoph Dorn, and Schahram Dustdar, *Simulation-based modeling and evaluation of incentive schemes in crowdsourcing environments*, On the Move to Meaningful Internet Systems: OTM 2013 Conferences (Robert Meersman, HervèŇĚ Panetto, Tharam Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter Leenheer, and Deijing Dou, eds.), Lecture Notes in Computer Science, vol. 8185, Springer Berlin Heidelberg, 2013, pp. 167–184.

[55] D.A. Schmidt, Changxin Shi, R.A. Berry, M.L. Honig, and W. Utschick, *Distributed resource allocation schemes*, Signal Processing Magazine, IEEE **26** (2009), no. 5, 53–63.

[56] Pamela Schmitt, Robert Shupp, Kurtis Swope, and Justin Mayer, *Precommitment and personality: behavioral explanations in ultimatum games*, Journal of Economic Behavior & Organization **66** (2008), no. 3-4, 597–605.

[57] Yoav Shoham and Kevin Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*, Cambridge University Press, New York, NY, USA, 2008.

[58] FLORIAN SKOPIK, DANIEL SCHALL, and SCHAHRAM DUSTDAR, *Discovering and managing social compositions in collaborative enterprise crowdsourcing systems*, International Journal of Cooperative Information Systems **21** (2012), no. 04, 297–341.

[59] Leon Sterling and Kuldar Taveter, *The art of agent-oriented modeling*, The MIT Press, 2009.

[60] Zhulei Tang, Michael D. Smith, and Alan Montgomery, *The impact of shopbot use on prices and price dispersion: Evidence from online book retailing*, International Journal of Industrial Organization **28** (2010), no. 6, 579 – 590.

[61] K. Taveter, Hongying Du, and M.N. Huhns, *Method for rapid prototyping of societal information systems*, Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on, 2012, pp. 1221–1228.

[62] Kuldar Taveter, Hongying Du, and Michael N. Huhns, *Engineering societal information systems by agent-oriented modeling*, Journal of Ambient Intelligence and Smart Environments **4** (2012), 227–252.

[63] O. Tokarchuk, R. Cuel, and M. Zamarian, *Analyzing crowd labor and designing incentives for humans in the loop*, Internet Computing, IEEE **16** (2012), no. 5, 45–51.

[64] Hong-Linh Truong, Schahram Dustdar, and Kamal Bhattacharya, *Programming hybrid services in the cloud*, Service-Oriented Computing (Chengfei Liu, Heiko Ludwig, Farouk Toumani, and Qi Yu, eds.), Lecture Notes in Computer Science, vol. 7636, Springer Berlin Heidelberg, 2012, pp. 96–110.

[65] HONG-LINH TRUONG, SCHAHRAM DUSTDAR, and KAMAL BHATTACHARYA, *Conceptualizing and programming hybrid services in the cloud*, International Journal of Cooperative Information Systems **22** (2013), no. 04, 1341003.

[66] Yathiraj B. Udupi and Munindar P. Singh, *Information sharing among autonomous agents in referral networks*, Agents and Peer-to-Peer Computing (SamuelR.H. Joseph, Zoran Despotovic, Gianluca Moro, and Sonia Bergamaschi, eds.), Lecture Notes in Computer Science, vol. 5319, Springer Berlin Heidelberg, 2010, pp. 13–26.

[67] Andrzej Uszok, JeffreyM. Bradshaw, and Renia Jeffers, *Kaos: A policy and domain services framework for grid computing and semantic web services*, Trust Management (Christian Jensen, Stefan Poslad, and Theo Dimitrakos, eds.), Lecture Notes in Computer Science, vol. 2995, Springer Berlin Heidelberg, 2004, pp. 16–26.

[68] J.S. Valacich and C. Schneider, *Information systems today: Managing in the digital world*, Prentice Hall PTR, 2009.

[69] Astrid M. von der Putten, Nicole C. Kramer, and Jonathan Gratch, *How our personality shapes our interactions with virtual characters - implications for research and development*, 10th International Conference on Intelligent Virtual Agents (Philadelphia, PA), 2010.

[70] Xin Wang and G.B. Giannakis, *Resource allocation for wireless multiuser ofdm networks*, Information Theory, IEEE Transactions on **57** (2011), no. 7, 4359–4372.

[71] Duncan J. Watts and Steven H. Strogatz, *Collective dynamics of 'small-world' networks*, Nature **393** (1998), 440–442.

[72] Uri Wilensky, *Netlogo*, http://ccl.northwestern.edu/netlogo/, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.

[73] Michael Wooldridge, *An introduction to multiagent systems*, 2nd ed., Wiley Publishing, 2009.

[74] Jaeyoung Yang, Tae-Hyung Kim, and Joongmin Choi, *An interface agent for wrapper-based information extraction*, Intelligent Agents and Multi-Agent Systems (MichaelWayne Barley and Nik Kasabov, eds.), Lecture Notes in Computer Science, vol. 3371, Springer Berlin Heidelberg, 2005, pp. 291–302.

Appendix A

The Shopping List

The Shopping List

Item index	Item	Walmart	Publix	Food Lion	BI-LO	Target
		store 0	store 1	store 2	store 3	store 4
1	Tropicana: orange juice, 64oz	2.92	3.79	2.97	3.69	2.99
2	Simply Orange juice, 1.75l	3	3.79	2.99	3	2.99
3	Corona extra: 12oz*6	8.47	8.29	7.99	8.29	6.5
4	Budlight: 12oz*6	6.97	6.49	5.99	6.99	5.25
5	Totino's: pepperoni pizza, 10.2oz	1.25	1.49	1.67	1.67	1.2
6	Coca-Cola zero, 12oz*12	4.28	4.99	3.33	3.33	4.99
7	Mtn Dew soda, 2l	1.38	1.59	1.33	1.79	1
8	Lay's: potato chips classic, 11oz	2.48	3.99	3	3.49	3.59
9	Pringles: BBQ chips, 6.38oz	1.5	1.5	1.67	1.5	1.44
10	Chicken drumsticks	1.72/LB	1.19/LB	1.49/LB	1.79/LB	-
11	Chicken breasts	3.33/LB	1.69/LB	1.99/LB	4.49/LB	-
12	Horizon organic: milk 0.5gal	3.5	3.99	3.79	3.49	3.54
13	Silk: pure almond 0.5gal	2.64	2.99	2.79	3	2.69
14	McCormick: black pepper, 2oz	1.38	2.19	1.72	1.89	0.94
15	McCormick: grill mates, 3oz	1.78	1.99	2.05	1.49	1.99
16	Kraft: sharp cheddar, 8oz	2	2.8	2.5	2.99	2.29
17	Sargento: sharp chedder, 16oz	4.74	3.99	4.89	2.99	4.38
18	Ground Beef	1.98/LB	2.89/LB	3.59/LB	4.29/LB	-
19	Whole grain bread, 24oz	3.69	3.69	3.69	3.69	2.54
20	Whole wheat bread, 16oz	2.98	2.5	2.99	3.59	2.39
21	Quaker: instant oatmeal, 15.1oz	2.88	1.99	2.99	3.79	2.89
22	Spearmint 14 ct gum	0.96	1.19	1.19	1.19	1.04
23	Doublemint 15 sticks gum	0.86	1.09	1.19	1.19	1.04
24	Ribeye steak	8.28/LB	11.99/LB	10.99/LB	9.99/LB	-
25	Beringer: Merlot, 750ml	8.97	6.59	7.49	7.99	8.99
26	Woodbridge Chardonnay, 750ml	5.47	6.99	6.49	6.5	5.00
27	Large cooked shrimp	5.71/LB	7.99/LB	10.99/LB	6/LB	-
28	Butter pecan ice cream, 1.5qt	3.5	4.41	4.69	5.49	3.99
29	Chocolate ice cream, 1.5qt	2.25	4.99	3.79	3.3	2.89
30	Maxwell House: original, 10.5oz	3	3.89	4.41	3.79	3.27
31	Starbucks: house blend 12oz	7.48	9.29	8.19	9.19	6.99
32	One dozen large eggs	1.25	1.39	1.22	1.19	1.29
33	Gala apples large	1.67/LB	1.89/LB	1.79/LB	1.99/LB	-