Towards Restraining Cost in Healthcare Domain: A Multiagent Approach

Saadat M. Alhashmi, Monash University, Sunway Campus
saadat.m.alhashmi@infotech.monash.edu.my

Abstract

The proliferating cost with-in the healthcare domain is forcing the researchers and practitioners alike to revisit healthcare logistics domain again and again in order to control the cost. However, controlling healthcare cost requires that limits be placed either on prices, quantities of services or both. As prices can be easily controlled by effectively focusing on the mechanism rather then involving and arguing about the services. The prototype developed has directed its attention towards coordination and intelligence in order to effectively manage the logistics while aiming for the improvement of mechanism. Coordination and user profiling has been demonstrated in multi agent environment. The idea here is to order medicines using human expertise in the form of fuzzy logics and effective coordination of different hospital pharmacies for an efficient use of medicines.

1. Introduction

The supply problem is as old as human society and rightly so supply of goods is just as ancient. The Holy Grail is how we can deliver goods by shortest route in quickest time. A supply chain management is a network of facilities and distribution options that performs the functions of procurement of materials, transformation of these materials into intermediate and finished products, and the distribution of these finished products to customers [1].

The problem of managing supply chains becomes even more critical and evident when it involves healthcare products as it involves life threatening drugs and equipments. On the other hand the health care industry has been under extreme political and public pressure to control the rapid increasing cost for treatment compared to the deliverance of services. The majority of current supply chain research has focused on refinement of existing Just in Time (JIT) systems used in the manufacturing industry but with regards to the health care industry there has been very little research directed towards implementation of the JIT concept. Most research in the healthcare industry has been directed towards process and information system improvements. It could mainly be because historically the health care industry has viewed itself as being operationally different from other businesses. As, health care practitioners have been hurt by breakdowns in the distribution, they often prepare for such events by stockpiling supplies. The main idea behind stockpiling is to avoid the liability incurred by a patient dying because critical supplies may not be available, [2], it could also mean in case of epidemic or emergency how are they going to cope with. This fear or practice has hindered the healthcare industry's efforts at effective supply-chain management the result is evident with the scale of bullwhip effect [3] but, in reality the health care industry shares many similar business processes with the manufacturing industry, especially in the areas of supply distribution, inventory control and product production. However, controlling health care costs requires that limits be placed either on prices, quantities of services, or both. Prices are measurable and more easily controlled than is quantity and consequently, health care cost containment has frequency focused on mechanisms for controlling prices.

Demand variability increases as one moves up the supply chain away from the actual requirement also know as bullwhip effect and even a small change in demand can result in large variations in orders placed upstream some of the factors which lead to this are:

- Overreaction to backlogs
- No communication up and down the supply chain
- No coordination up and down the supply chain
- Delay times for information and material flow

In this paper, a co-operative autonomous decision-making system is proposed with autonomy and co-operation facets of human intelligence. Fuzzy logic is used to depict decision-making and the system is developed on a multiagent platform to provide autonomy and co-operation.

2. The Integrated System.

In our view, a co-operative autonomous decision-making system must at least satisfy the autonomy, co-operation and intelligence. For the purpose of this research, these can be explained as:

Autonomy: The system must make decisions on the user's behalf; unlike a traditional system where the system needs a command from an input device. More specifically, the autonomous system must have control over its internal state and actions. For example, a traditional decision-making system remains passive to execute specific tasks. However, an autonomous system would execute the assigned tasks on its own for the achievement of predefined goals.

Intelligence: The decision-making system must exhibit intelligent behaviour by applying expert knowledge to manipulate the environment. This would involve recognising the relative importance of different elements in a situation and then responding in a flexible and quick manner to fulfil predefined objectives.

Co-operation: The autonomous and intelligent decision-making tool must also demonstrate co-operative behaviour. This means that the individual agents within the system would work to achieve the overall goal of the system without conflicting with each other. This builds upon the concept of socialability, i.e. the agents would autonomously communicate and negotiate with each other. We realised that the traditional technologies lack autonomy and co-operation; it was decided to propose a system by combining fuzzy logic and multiagent technologies. Fuzzy logic was chosen to build our system mainly because according to Lotfi Zadeh cited by Steimann [4, 5]

Indeed, the complexity of biological systems may force us to alter in radical ways our traditional approaches to the analysis of such systems. Thus, we may have to accept as unavoidable a substantial degree of fuzziness in the description of the behaviour of biological systems as well in their characterisation. This fuzziness, distasteful though it may be, is the price we have to pay for the ineffectiveness of precise mathematical techniques in dealing with systems comprising a very large number of interacting elements or involving a large number of variables in their decision trees.

Although many intelligent systems have incorporated the learning aspect, and are thus learning-based systems, in our view, the system we are proposing does not need to have learning capabilities if expert knowledge is already built into the system. The idea behind this argument is that the expert will monitor the behaviour of the system on a regular basis and tune the fuzzy rules if necessary, rather than the system learning on its own in this highly complex domain.

Figure 1 below is an overview of the system.

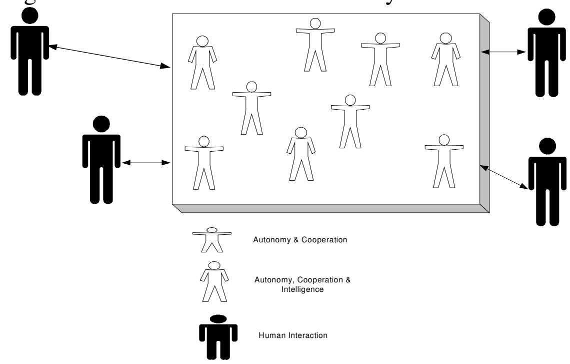

Fig 1. System Overview

The aim was to ensure that Fuzzy Logic and multiagents system complement each other and after reviewing different decision-making technologies we came up with the model shown in Figure 2. The interesting thing to reiterate at this stage is if we are applying this model in domain where we need a different set of skills i.e. learning behaviour or qualitative reasoning we can chose Genetic Algorithm or Qualitative Reasoning and complement it with multiagents.

.

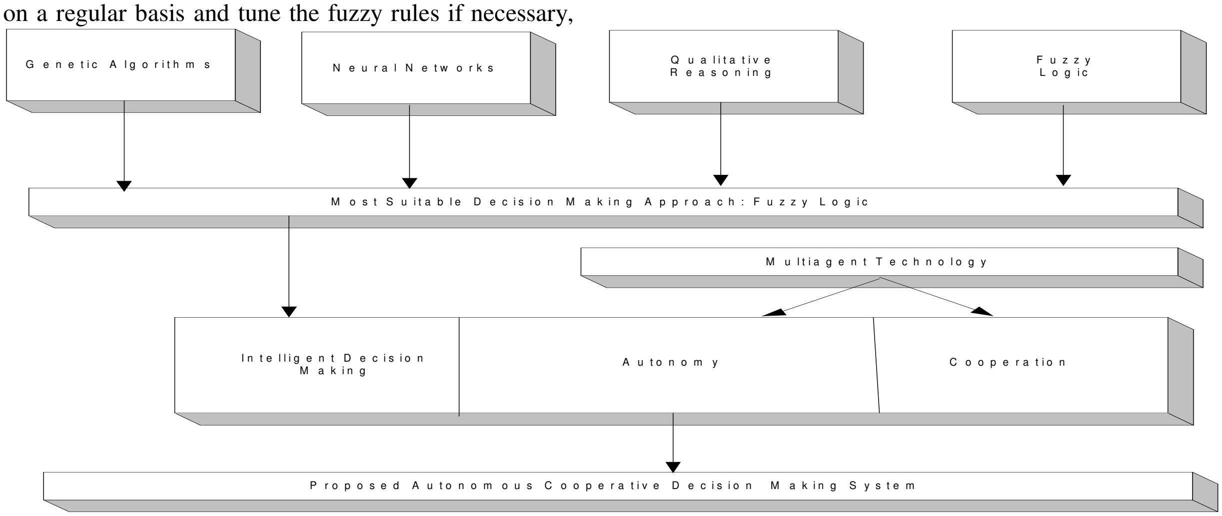

Fig 2. Autonomous co-operative decision-making system

The requirements for the system were envisaged in terms of its workings and applicability.

1. The doctor recommends medicine (generic name), dosage and time (i.e., how long a patient is supposed to take that medicine before their next appointment).
2. The local pharmacy should check if the medicine is available autonomously.
3. In case the local pharmacy runs out of stock, an autonomous negotiation takes place to
4. Procure medicine from the collaborative pharmacies.
5. A routine service to move medicine and other related services should be available to move things between the collaborative or participating pharmacies.

Ordering of medicine should be based on the pharmacist's recommendations about the supplier's delivery reliability and pattern of demand.

2.1. Assigning Agents

Our focus was on replicating the real world problem as is and mapping it onto the chosen technology. Every agent in the system replicates functions of the real world. For example, the Doctor Agent replicates the actions of the doctor as they would perform in the absence of this autonomous system. Figure 3 shows how agents are assigned.

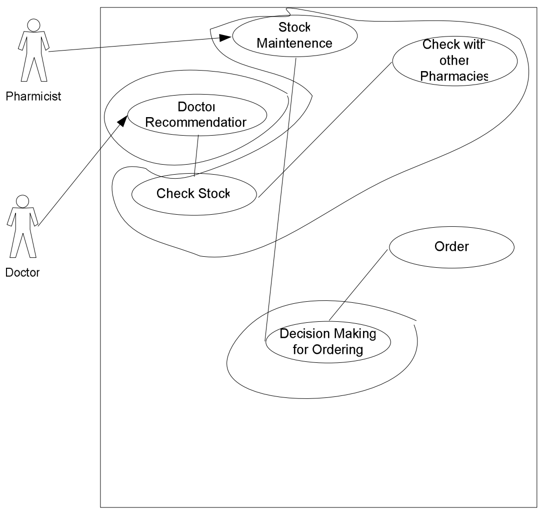

Fig 3. Assigning Agents

Three agents were identified based on similar activities: Doctor Agent Pharmacy Agent. Decision making Agent. The aim was to ensure that an adequate number of agents would be used, such that the objectives of the system were being met, that a particular agent was performing similar type of activities, no activity was being repeated or overlooked and there was clarity and coherence in the system.

2.2. System Level Design

At the agent system level design stage, two models are presented. First is the agent system architecture that describes agent interaction diagrams and second, the system plan (sequence diagram) using the Agent-based Unified Modelling Language (AUML) [6]concept. AUML is an extension of Unified Modelling Language (UML) [7], which has already gained wide acceptance for the representation of engineering artefacts. AUML basically synthesises a lot of agent-based methodologies.

2.3. System Plan Model

In the previous section, agents are identified for the system. This section shows the dynamic interaction of agents. This model in Figure 4 is adapted from AUML. The system plan model shows the different roles of agents, how they are interacting with one another and the communicative protocols between them.

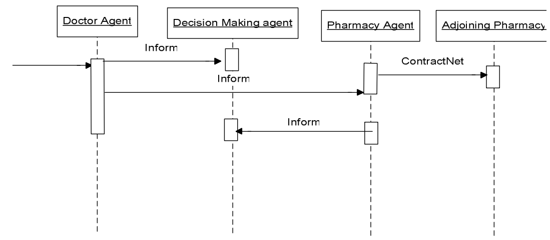

Fig 4. Agent interaction sequence diagram

In order to express in detail the communication between different agents this research has followed the FIPA [8] guidelines. The communication protocols used in this research are: inform, request and propose protocols. For example, in Figure 4, the Doctor Agent sends information about the patient's medicine requirement Decision-making agent using Inform protocol [9] etc. The nature and sequence of individual interactions is explained in detail in the next section.

2.4. Agent Level Design

This section focuses on the design of individual multiagent. We will describe the design of one agent. A Class-Responsibility-Collaborator (CRC) index card [10] is used to represent collaboration. CRC cards provide an effective technique for exploring the possible ways of allocating responsibilities and give high level descriptions of functions to classes and the collaborations that are necessary to fulfil the responsibility [11]. The strength of CRC cards lies in the fact that, when utilised by a team, the interaction between the members can highlight missing or incorrect items in the class. Also, the relationships between classes are clarified when CRC cards are used.

Pharmacy Agent: This section is an explanation of the interactions of Expert Agent shown in Figure 4. Table 1 is an Pharmacy Agent CRC card.

Responsibilities

Table 1 shows that the Pharmacy Agent has a number of responsibilities. After receiving the patient's requirements, it checks its own stock. If the medicine is unavailable, it negotiates with other pharmacy agents to check if they have the medicine. If they are out of stock as well, it then requests the Decision-making Agent for advice on ordering, and based on this advice, it orders the medicine from the supplier.

Collaborations

The Pharmacy Agent collaborates with the Doctor Agent, the Decision-making Agent and the agents of nearby pharmacies. These collaborations are shown in Table 2

Table 1: Pharmacy Agent CRC Card

Pharmacy Agent	
Responsibilities	**Collaboration**
Receive patient's requirements	Doctor Agent
Check if medicine is in stock	-
Negotiate and co-ordinate with other Pharmacy Agents	Nearby pharmacies
Request Decision-making Agent for advice	Decision-making Agent
Receive and accept decision from Decision-making Agent	Decision-making Agent
Order medicine	-

Table 2: Pharmacy agent collaborations

Pharmacy Agent	
Sending Message to	**Receiving message from**
Decision-making Agent Message type: *Request* Content: *Medicine name, Quantity Required, Supplier Reliability, Demand Pattern*	Doctor Agent Message type: *Inform* Content: *Generic name of medicine, quantity required*
-	Decision-making Agent Message type: *Inform* Content: *Medicine name, Quantity to be Ordered*

3. Development

The main goal for developing this system is to reduce the wastage of medicine, automatically order medicine. The goals can be divided into three main categories.

- Automatic Ordering of Medicine.
- Coordinating with Near-by-Pharmacies.
- Better Utilisation of Medicine.

3.1. Pharmacy Agent Plan Model

The agent plan model shows the agent's internal tasks and the data structures used for agent interaction. The goal of the agent plan model is to show the activity of each agent in detail, the state of the agent and its interaction with different agents alongwith the type of message. Standard AUML symbols are used. We will discuss the working of one agent, pharmacy agent.

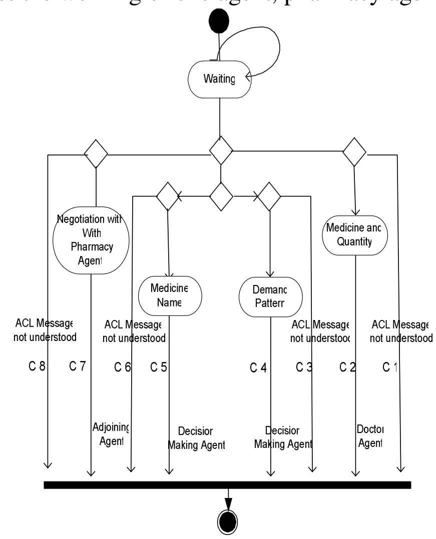

Fig 5. Pharmacy Agent - Agent Plan Model

Different communications as shown in Figure 3 are:

C 1: Throws an exception, ACL message not understood

C 2: Receives a message from Doctor Agent about the name and quantity of the medicine

C 3: Throws an exception, ACL message not understood

C 4: Receives a message from Decision-making Agent for ordering medicine

C 5: Sends message to Decision-making Agent for demand pattern of medicine for expert system

C 6: Throws an exception, ACL message not understood

C 7: Negotiates with nearby Pharmacy Agent for the medicine

C 8: Throws an exception, ACL message not understood

Table 3 explains the interactions and the type of messages the Pharmacy Agent is exchanging with other agents in more detail. For example, one of the tasks is to receive the patient's requirements for medicine quantity and the interaction protocol is Inform [12]. Similarly, another task of the agent is to negotiate with nearby pharmacies and the protocol used is FIPA ContractNet [13]. These messages are illustrated in Figure 5, along with the collaborations taking place between the Pharmacy Agent and other agents. As seen in the figure, the Pharmacy Agent is in a state of waiting unless an event happens.

Table 3: Pharmacy Agent

Pharmacy Agent				
Goal: Order Medicine				
Role: Negotiation / Co-ordination and Ordering				
Tasks	Collaborations	Communication Name		
			Message Type	Content
1.Receive Patient's Requirements	Doctor Agent	C 2	*Inform*	Generic name of medicine and quantity required.
2. Check stock of medicine	-	-	-	-
3. Engage in negotiations	Nearby- Pharmacy Agent	C 7	*ContractNet*	Name of medicine, quantity and urgency
4. Send Patient's Requirement	Decision-making Agent	C 4	*Request*	Name and quantity required of medicine, supplier reliability and demand pattern
5.Receive Decision-making Agent's Advice	Decision-making Agent	C 5	*Inform*	Name of medicine and quantity to be ordered
6.Order medicine	-	-	-	Name of medicine and quantity to be ordered

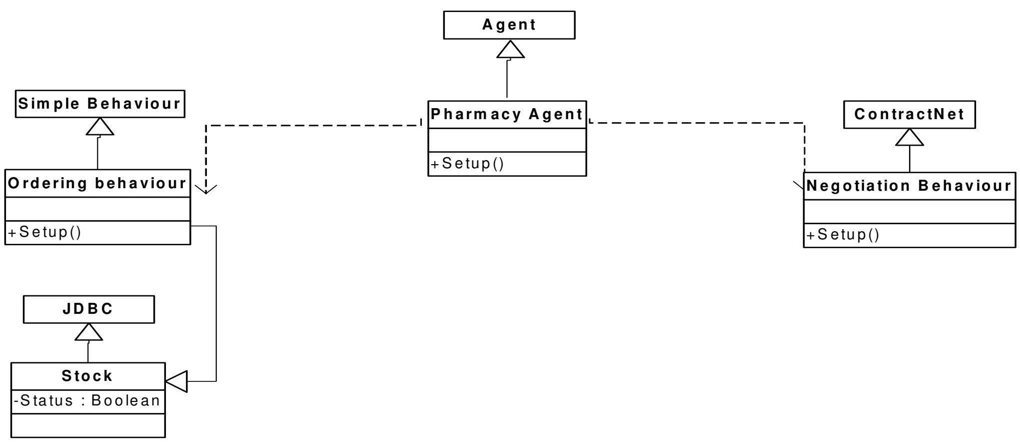

Fig 6. Pharmacy Agent Class Diagram

From Figure 5 and Table 3 we can derive the class diagram shown in Figure 6. Here, the class diagram for the Pharmacy Agent shows its internal workings for example, for ordering the medicine, a Jade [14] } simple behaviour has been used. Similarly, for negotiation with pharmacies, ContractNet behaviour is used.

3.2. Communicative Act Amongst Agents

This section explains how agents are communicating with each other. The agent interactions shown below are based on the Agent Interaction Sequence Diagram (figure 4). Here, more details are given regarding the interaction between the individual agents as well as an explanation of these interactions. This section is based on the Communication Act Library (CAL) of FIPA. According to FIPA, agent communications have to adhere to a standard procedure, as explained in the following sections.

3.3 Pharmacy Agent and Nearby Pharmacy Agents

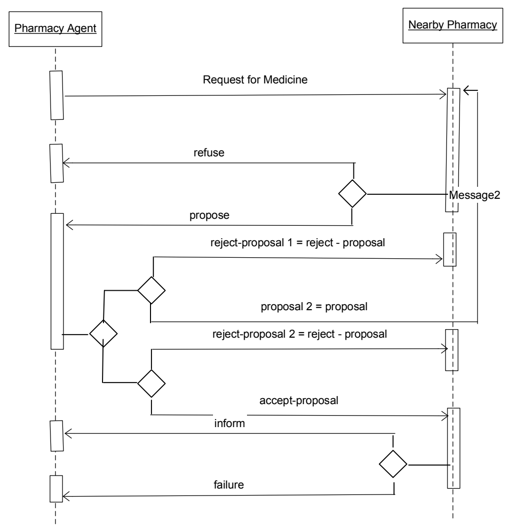

Fig 7. Pharmacy Agent and Nearby Pharmacy Agent Communicative Diagram

In Figure 7 the Pharmacy Agent communicates with nearby Pharmacy Agents using FIPA *Contract Net* protocol [13]. The content of the message is medicine quantity. The Pharmacy Agent sends a request to the nearby pharmacy agents for a particular medicine. The nearby pharmacy may either refuse this request, probably because it does not have the required medicine in stock, or doesn't have the required quantity of the medicine. The communication comes to an end with this refusal. The second possibility is that the nearby pharmacy accepts the request. It then makes an offer and sends a proposal to the Pharmacy Agent. The Pharmacy Agent either rejects this offer and makes a counter offer, or accepts it, bringing the communication to an end. This iterative process continues until both the agents either agree or disagree based on the initial criteria programmed in each individual agent. In the former case, a deal is finalised and in the latter, the Pharmacy Agent begins a negotiation process with another Pharmacy Agent

Figure 7 is a detailed explanation of the interactions shown in Figure 2. According to FIPA protocol, agents have to adhere to a standard procedure. In Figure 6 Patient Agent sends request to Expert Agent, using FIPA *Request* protocol [8] and the contents of the message are Activity level, Glucose and Diet. Once the Expert Agent receives this request it can either refuse it or agree to it. If it refuses, then that is the end of the communication. If it agrees, then, based on the knowledge or fuzzy rules the Expert Agent is linked to, it advises the Patient Agent.

Implementation and evaluation of this system is carried out mainly taking into consideration:

- Ease of use
- Effectiveness of advice

4.0 Implementation of the Prototype

This section is a system-walk-through of the developed prototype. The doctor logs in to the system, and is authenticated, after which he enters information regarding the patient's medicine requirements, i.e. the generic name of the medicine and the quantity required. This information is sent to the Pharmacy Agent. The initial stage is that an agent monitors the stock and alerts the system as soon as it goes below a certain level. Figure 8 is the screen shot of the agent monitoring the stock. Here, it is monitoring a file, and as soon as it goes below a specified level, an alert message is sent.

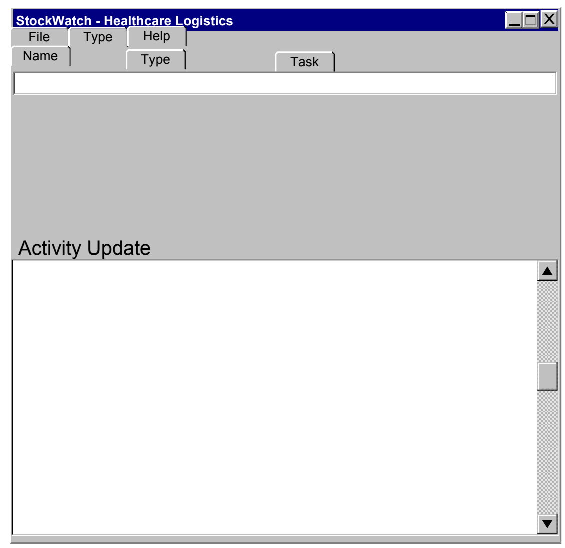

Fig 8. Screen shot of agent monitoring the stock

Figure 9 is the GUI which will appear on the screen when the agent informs the user that the stock level has fallen below the specified level.

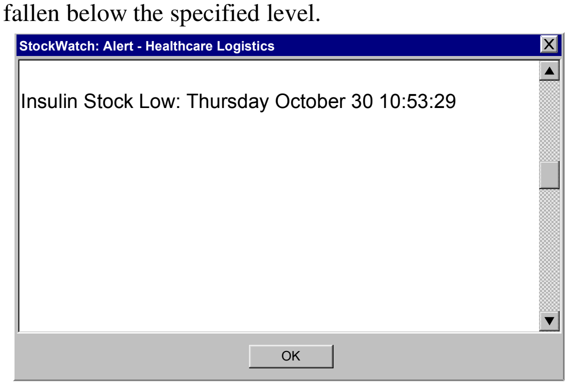

Fig 9. Screen shot of inform message regarding the status of the medicine

Once the pharmacy is informed about the status of the medicine, the next stage is to negotiate the procurement of medicine from the nearby pharmacy. Figure 10 is the GUI for the medicine negotiation box.

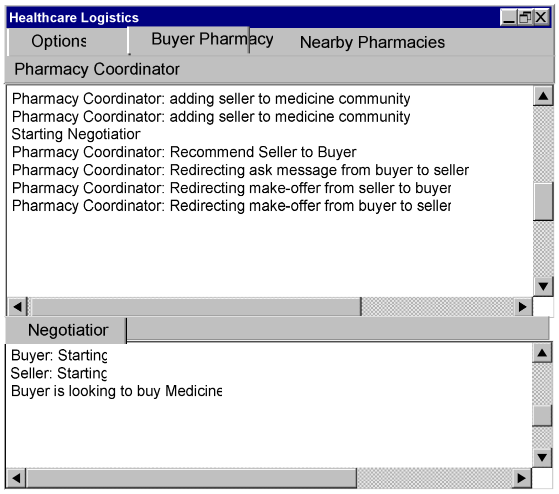

Fig 10. Screen Shot of Medicine Negotiation Box

The idea is that, once the pharmacy runs out of the required medicine, the agents will negotiate with the nearby pharmacies to get the best deal in terms of time of delivery and price. To work this out a pharmacy co-ordinator is developed as shown in Figure 9. This is like a mart where different pharmacy agents are going to negotiate with the help of the pharmacy co-ordinator. Before negotiation/co-ordination, every agent has to register itself with the pharmacy co-ordinator like the directory facilitator. Different pharmacy agents can have different strategies; some can have hard coded logics and others can use forward-chaining rules. After registering with the pharmacy co-ordinator, agents can start communicating with each other with the help of the pharmacy co-ordinator about their requirements. Pharmacy Agent One, who requires a particular medicine, informs the pharmacy co-ordinator that it needs a medicine; the pharmacy co-ordinator will inform Pharmacy Agent One about Pharmacy Agent Two, now Pharmacy Agent One can ask Pharmacy Agent Two if it has that particular medicine. In case it doesn't have the medicine, it rejects the offer. If, however, Pharmacy Agent Two has the medicine, it can then make an offer, and inform about when it can deliver, etc. Pharmacy Agent One can then either accept this offer or make a counter offer. Pharmacy Agent Two can then also accept the offer or reject the offer made by Pharmacy Agent One. Once Pharmacy Agent Two accepts the request, the negotiation is over. All the above negotiations take place with the mediation of a pharmacy co-ordinator.
If after negotiation, Pharmacy Agent One fails to get the required medicine from the nearby pharmacy then the next step is to order the medicine using the intelligent ordering system, i.e. using Fuzzy Logic. Here, the pharmacist based on his/her knowledge about the demand pattern and the reliability of supplier enters the details. The expert system uses this information to generate the recommended quantity of medicine as shown in Figure 11.

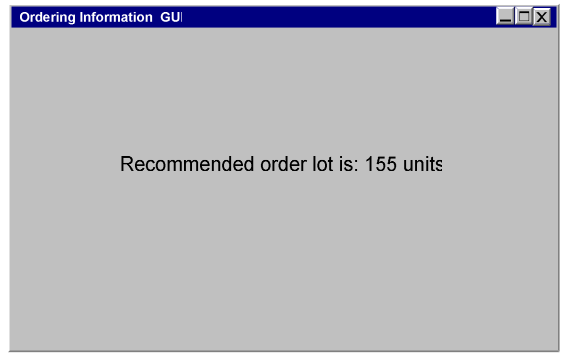

Fig 11. Recommended order quantity based on expert system

4. Conclusion and Discussion

The most obvious benefit from the prototype developed is the delegation of tasks for autonomous co-operation and intelligence. This ensures adequate interoperable communication support. Moreover, this also ensures that medicines can be used more efficiently as routine house-keeping tasks are delegated to multiagents. From a system design perspective, the developed prototype introduces a flexible concept. Multiagents can be used in a dynamic task-oriented team where humans and multiagents can work together.
The experience learned from developing this prototype so far has proved the concept that fuzzy and multiagent technologies can provide an autonomous intelligent healthcare logistics.
However, we would like to highlight some caveats in the prototype developed. After testing and evaluation of the prototype, it was realised that some factors that could have further enhanced this proof-of-concept are as follows:

- The ordering decision has been based on two factors: supplier reliability and pattern of demand. The price element could have been incorporated in the ordering decision.
- As regards supplier reliability, this was considered in terms of the frequency of the supplier's delivery of medicine. It was assumed that the supplier would bring the correct medicine.
- The prototype shows collaboration and negotiation between different pharmacies. This aspect could also have been extended to the suppliers. In other words, there could be an autonomous negotiation taking place between different suppliers on a supplier network.

The analysis and design phases in any software development cycle are critical and usually require careful attention to details. The developed prototype is no exception. It's main contribution is in the proof-of concept and the system developed.

There are issues other than those we discussed, for example, issues concerning the selection of agents, should we try for less or more. Other issues include environmental properties, for example, the security and stability aspect. Not to mention moral and ethical arguments which encompasses any debate when it comes to technological debates. All these issues would have to be addressed on a case-by-case basis and the current prototype does not offer much help. Nevertheless, it is argued that the proposed prototype would provide a framework in which these issues and questions can be identified and articulated more easily.

5. References

1. Christopher, M., *Logistics and Supply Chain Management*, ed. Anonymous. 2000, New Delhi, India: Pitman Publishing. 294.
2. Jarrett, P.G., *Logistics in the Healthcare Industry*. International Journal of Physical Distribution & Logistics., 1998. 28(9/10): p. 741-772.
3. Fransoo, J.C. and M.J.F. Wouters, *Measuring the bullwhip effect in the supply chain*. Supply Chain Management: An International Journal, 2000. 5(2): p. 78-89.
4. Phuong, N.H. and V. Kreinovich, *Fuzzy logic and its applications in medicine*. International journal of medical informatics, 2001. 62(2-3): p. 165-173.
5. Steimann, F., *Fuzzy Set Theory in Medicine*. Artificial Intelligence in Medicine, 1997. 11: p. 1-7-7.
6. Odell, J., D.V.H. Parunak, and B. Bauer, *Representing Agent Interaction Protocols in UML*. Agent_Oriented Software Engineering, 2001: p. 121â€“140.
7. Object Management, G., *Unified Modelling Language Specification Version 1.3*, in *OMG Document*. 1999.
8. Foundation For Intelligent Physical, A., *FIPA Communicative Act Library Specification*. 2001, http://www.fipa.org: Geneva, Switzerland.
9. Foundation for Intelligent Physical, A., *FIPA Request Interaction Protocol Specification*. 2002, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS: Geneva, Switzerland. p. 1-7.
10. Wirfs-Brock, R., B. Wilkinson, and L. Wiener, *Designing object-oriented software*, ed. Anonymous. 1990: Prentice-Hall. 341.
11. Bennett, S., S. McRobb, and R. Farmer, *Object-Oriented Systems Analysis and Design using UML*, ed. Anonymous. 1999: McGraw Hill.
12. Foundation For Intelligent Physical, A., *FIPA Inform Communicative Act Specification*. 2001, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS: Geneva, Switzerland.
13. Foundation For Intelligent Physical, A., *FIPA Contract Net Interaction Protocol Specification*. 2002, Foundation for Intelligent Physical Agents: Geneva, Switzerland.
14. Bellifemine, F., G. Caire, and et al., *Jade Programmer's Guide*. 2003, TILAB, formerly CSELT.