Information Flow Analysis in Autonomous Agent and Peer-to-Peer Systems for Self-organizing Electronic Health Records

Ben Tse[1], Raman Paranjape[1], and Samuel R.H. Joseph[2]

[1] Electronic Systems Engineering,
University of Regina and TRLabs Regina,
3737 Wascana Pkwy
Regina, SK S4V 0R4, Canada
`Raman.paranjape@uregina.ca`
[2] Laboratory for Interactive Learning Technologies,
Department of Information and Computer Sciences,
University of Hawaii,
1680 East West Road, POST 309
Honolulu, HI 96822 USA
`srjoseph@hawaii.edu`

Abstract. There are various software applications that are highly suited for development using agent technology. Typically these applications take advantage of some of the intrinsic qualities of agents that include: autonomy, reactivity/proactivity, group-action, and/or mobility. On the other hand, there are many parallels between Agent Systems and Peer-to-Peer Systems allowing the latter to be employed in similar problem domains. This paper presents an agent application in the health care record management domain and then examines how such a system might also be implemented as a Peer-to-Peer System. The management of health care records is in itself a novel use of Mobile Agent technology and in order to understand the Agent System Dynamics, the system is simulated using a limited number of agents and agent platforms; as well as being modeled mathematically. The Peer-to-Peer system is also simulated and modeled mathematically demonstrating a number of behaviors that are similar across both systems.

Keywords: autonomous agents, peer-to-peer, mobility, electronic health records.

1 Introduction

1.1 Electronic Health Records

Health record information access/retrieval is one of the major problems in modern health care systems (Moreno, 2003; Nealon & Moreno, 2003). Ideally relevant information from a patient's complete health record would be available to every practitioner at all times; however prescription information, test results and doctor's

S. Joseph et al. (Eds.): AP2PC 2006, LNAI 4461, pp. 1–18, 2008.

diagnoses are generated and stored in multiple locations such as hospitals, clinics, pharmacies and so on. In reality it is difficult to assemble the relevant information in the correct location at the right time in order to provide the best possible service to the patient. The problem is made more complex by the importance of maintaining patient privacy.

An Electronic Health Record (EHR) is an electronic version of a patient's health information and contains prescriptions, lab results, evaluations by doctors, etc. EHRs can be made easily accessible through an electronic health information network. The advantages of EHRs include: increasing effectiveness and efficiency of clinical staff and health practitioner by simplification of access to health records, rapid movement of health information for better care of patients, simple duplication and multiple/simultaneous access to patient health information, and potential increases in the profitability of the medical practices and/or facilities.

Although EHRs appear to hold great promise, there are many challenges that need to be addressed before they can be fully integrated in a health care system. These challenges include: security and confidentiality, lack of standards (data exchange, data management and data integration) or slow adaptation to existing standards, lack of government and/or private funding, especially in developing countries, complexity of medical data, rejection by health care professionals, and network bandwidth consumption (Dick & Steen, 1991; Johns, 1997).

1.2 Mobile Agent Technology

Mobile agent technology has received a fair degree of attention in academic research in recent years (Kotz et al., 2002). Mobile agents are defined as a software objects that can migrate to different computers over an IP network to perform user-assigned and self-initiated tasks. Mobile Agents are autonomous software programs that may start running on a host computer, stop what they are doing, move to another host computer, and start up from where they left off.

Mobile Agents are best understood through comparison with other related technologies such as mobile code, distributed objects, and viruses/worms. Mobile code technologies such as process migration, remote evaluation, and mobile objects are very similar to Mobile Agents but differ in that Mobile Agents autonomously initiate their own mobility during their execution process. Mobile Agents place an emphasis on location awareness that differentiates them from distributed object technologies like RMI, CORBA, and DCOM (Raj, 1998) which abstract over location. Viruses and worms are related technologies that have negative connotations; however they are essentially mobile agents that use deception or software bugs to facilitate their movement and execution instead of relying upon an agent execution environment.

The mobile agent programs run with the aid of another program called an agent execution environment that must be installed and running on a host computer before the mobile agent program can run. An agent execution environment provides the mobile agents with services for mobility, messaging, resource access, and security. The agent execution environment also provides administration services for running and monitoring the behavior of mobile agents. TEEMA (TRLabs Execution Environment for Mobile Agents) is an agent execution environment that was developed by faculty, staff and graduate students at TRLabs Regina and Electronic Systems Engineering at the University of Regina. TEEMA provides basic services such as logging, agent-to-agent messaging, agent migration, and agent naming. Customized services can be added to TEEMA without any difficulty because of its flexible architecture. More information related to TEEMA can be found in (Gibbs, 2001; Martens, 2001).

A mobile agent system may be viewed as a specific type of multi-agent system that would be classed as a Heterogeneous Communicating Multi-agent System according to Stone's taxonomy (Stone & Veloso, 2000). Stone's survey lists many multi-agent systems and classifies them into: Homogeneous Non-Communicating Multi-agent Systems, Heterogeneous Non-Communicating Multi-agent Systems, and Heterogeneous Communicating Multi-agent Systems. SWARM (Minar et al., 1996; Luna & Stefansson, 2000) and REPAST (Collier, 2003) are examples of multi-agent systems that are particularly popular for economic simulations. Multi-agent systems can be composed of many intelligent mobile agents. These multi-agent systems have been used for various applications including for example: electronic market places (Smith et al., 2001) course scheduling applications (Yang et al., 2004), and network resource management (Wei et al., 2002).

2 Agent System Dynamics and Analysis

In order to ensure that useful and effective mobile agent systems are constructed it is important to study, examine and test their system level behavior. This allows for greater understanding of the system dynamics so that once the system is implemented the dangers of unexpected system behavior are reduced. These unexpected system behaviors result from unforeseen group actions of agent groups, and agent-group behavior that was not directly coded by the agent designers.

The proposed approach is to simulate the agent system with a simplified architecture. This simplified architecture was implemented and the actual behavior of the system examined using executions of the simulation. The simplified architecture can also be modeled mathematically to define asymptotic system behavior.

Several mathematical approaches have been introduced to model and analyze system behavior in multi-agent systems (Lerman & Galstyan, 2001; Tecchia et al., 2001; Xu et al., 2002). Among these approaches Lerman & Galstyan (2001) present a general mathematical approach to analyze the global dynamic behavior of multi-agent swarm systems. Swarm systems (Bonabeau et al., 1999) are typically composed of many simple, task-oriented, objects that travel through potentially hostile environments to search for their task-related items. With no central controller directing how individual objects behave, interesting and intelligent collective behaviors emerge from the local interactions among individual agents and the interaction between individual agents and their surrounding environment.

Lerman & Shehory (2000) applied their mathematical modeling approach to a swarm system in a large scale electronic market, allowing observation of coalition formation behaviors. This behavior, however, was not explicitly programmed into each individual agent but was a spontaneously produced group-action. Similarly, much work has been done in information access/retrieval based on mobile agents; for example Smith et al. (2001) indicated that the full potential of each individual agent is not obtained during unwanted agent-group behaviors. Therefore, the behavior of agents in a multi-agent system must be carefully examined before implementation of an actual system, in order to minimize the chances of system failure and achieve superior system and individual agent performance.

2.1 The Agent-Based Electronic Health Record System

Many current health care systems are distributed among different geographical locations and patients' record are scattered throughout the Health System and could physically be anywhere such as for example in a clinic, a doctor's office, medical laboratory and/or a pharmacy. We propose an Agent-Based Electronic Health Record System using the TEEMA platform. A simplified simulation model of the system is shown in Figure 1 and follows our earlier work (Tse & Paranjape, 2006).

By using mobile agent technology, we add mobility to these records, which allows the record to move independently anywhere within the health care information system. This multi-agent system can be colloquially described as putting a mobile

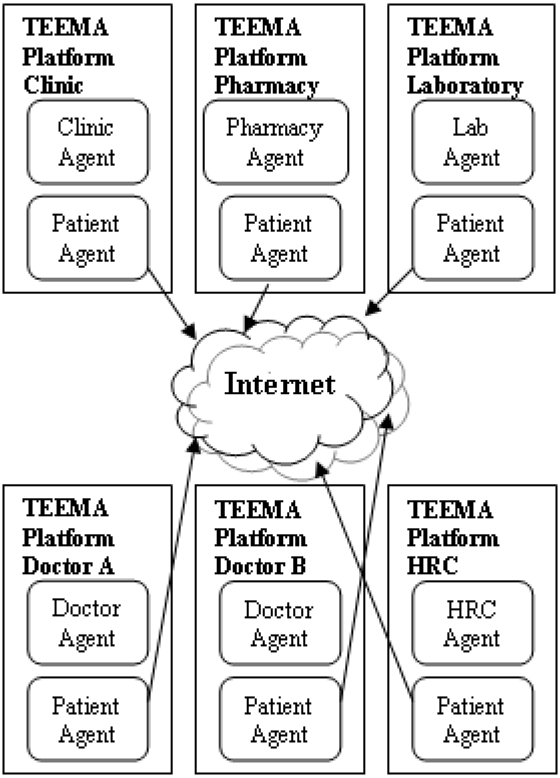


Fig. 1. An overview of the simplified Agent-based Electronic Health Record System architecture; each TEEMA platform represents a site in the system. Patient Agents visit each of these sites in the process of executing the simulation.

agent wrapper around an electronic health record fragment and instructing the agent to move the patient health record fragment to other medical facilities in order to unite and complete the patient electronic health record.

The two critically important aspects of the system are: (1) A complete health record set is defined as every piece of information in a patient's health record regardless of where it was generated united into one consistent and complete set of information. (2) Each agent in the system is self-regulated. This means that an individual mobile agent will accomplish its assigned task without any external supervision or guidance and no concept of what the group goal is. For each agent, its task is to retrieve and/or update the health record for the patient. Each individual agent has no interest in finding out what other agents within the system are doing.

2.2 System Components

Each TEEMA platform represents a certain physical location such as: a clinic, a doctor's office, a pharmacy or a laboratory. In each platform, there are a number of stationary and mobile agents. The location where all patient information is collected and collated is defined as the Health Record Central (HRC). All patient information is eventually sent to the HRC. The system contains five different types of stationary agents and one mobile agent:

Stationary Agents	
Clinic Agent (CA):	Responsible for creating an agent for a patient when the patient arrives at the clinic. The clinic agent also verifies patient identity.
Doctor Agent (DA):	Responsible for managing doctor's comment (health record) for patients.
Pharmacy Agent (PhA):	Responsible for validating the patient's identity and communicating with the patient agent when the patient pickups his/her prescription.
Lab Agent (LA):	Responsible for validating patient identity and communicating with the patient agent when the patient comes into the lab for medical tests.
Health Record Central Agent (HRCA):	Responsible for validating patient agents before they can access/modify/update the health record database.

Mobile Agents	
Patient Agent (PtA):	Is the patient's representative and it (or its clone) can migrate to different platforms to do work on the patient's behalf. It is responsible for updating patient health records, transferring new records to the HRC. If there is a prescription and/or lab test needed, the patient agent will clone itself and migrates to the pharmacy and/or laboratory and ensures that the patient fills the prescription or does the test and that the information is recorded and collected in the system.

Activity At Each Site	
Clinic:	All patients enter the simulation at the clinic. Patients Agents (PtA) are created when the patient enters the clinic. The PtA then checks if the patient health record needs to be updated. If so, it will clone itself and go to the HRC to obtain the necessary data. Then the PtA enters one of the two Doctor's offices. After the visit to one doctor's office the patient health record is updated and this new information is deposited in the HRC. The PtA will again clone itself and transmit the new information to the HRC. In addition, the Doctor may order laboratory tests, and/or medicines from the pharmacy. In this case, the PtA will also clone itself and move to the laboratory and/or pharmacy and wait for the patient to arrive.
Doctor's Office:	When the PtA arrives at the Doctor's office it interacts with a stationary doctor's office interface agent. This stationary agent relays the Doctor's instruction for the patient into the PtA. The Doctor's assessment of the patient's condition, which becomes part of the patient's health record also, is loaded into the PtA. The PtA then takes responsibility of the update of the health record and satisfying any Pharmacy or Laboratory requirements.
Laboratory:	A clone of the PtA is sent to the Laboratory on the instructions of the Doctor. The PtA clone waits for the patient to physically arrive in the Laboratory and then for Lab results to be generated. These results are assumed in this simulation to be available immediately after the patient visit but may in fact require some time to complete. The PtA clone interacts with a stationary Laboratory Agent which provides an interface to the Laboratory technician who is responsible for the operation of the Laboratory.
Pharmacy:	A clone of the PtA is sent to the Pharmacy on the instructions of the Doctor. The PtA clone waits for the patient to physically arrive in the Pharmacy and then for Pharmacy results to be generated. These results are assumed in this simulation to be available immediately after the patient visit but may in fact require some time to complete. The PtA clone interacts with a stationary Pharmacy Agent which provides an interface with the Pharmacist who is responsible for the operation of the Pharmacy.
Health Record Central (HRC):	Is the data center for the Agent-based health record system and is the place where all patient health information is stored. The HRC acts as the repository of all patient information that may be generated in the health care system even when the patient has exited the health

	care system. When the patient re-enters the health care system by coming to the clinic and a doctor's office, the HRC is used to update the patient information. The operation of the HRC is mediated by a stationary agent who task is to maintain the health record information delivered by PtA clones.

3 Experimental Validation

The architecture was implemented and experiments run to assess its behavior. These experimental results can be compared with numerical results from a general mathematical model of the system which will be discussed in section 3.4.

3.1 Simulation Structure and Conditions

Computers in the simulation were interconnected via 100Mbps Ethernet. Two computers were used that each executed a number of TEEMA platforms (the agent execution environment) Each TEEMA platform represents one specific physical medical site; in our experiment these included: 1 Clinic, 2 Doctors in the Clinic, 1 Pharmacy, 1 Laboratory and 1 HRC; leading to a total of 6 TEEMA platforms. The TEEMA platforms for the Clinic and the two Doctor's offices were executed on a single computer, and the other TEEMA platforms were executed on the other computer to represent the Pharmacy, Laboratory and the HRC.

Basic conditions and assumptions used in the experiment are listed below:

1. Doctor evaluations, prescription contents and lab test results are predefined to be the only type of data in the electronic health record. The combination of these components was considered the full health record of a patient.
2. Each TEEMA platform represents a physical medical site, so it will have its own unique IP address and port number. A configuration file is used to gather all TEEMA platforms associated with doctors, clinic, pharmacy and lab IP addresses and port numbers used in the experiment. This file is used as a reference for all the patient agents who need to migrate to different medical sites or TEEMA platforms.
3. Patients' health records are structured based on a file-system structure. So, in the HRC, each file contains an individual patient's health record.
4. A Number scheme was used for the patient name and each file was named using this scheme.
5. There are several random behaviors simulated by different kinds of random sources during the experiments:
   - Patient preference behavior – this behavior describes a patient's wish to choose a specific doctor. For simplicity a uniform distributed random number is used to represent this behavior.
   - Patient necessitated behavior - this behavior describes the need for a specific medical action. This includes the need for prescriptions and lab work. A Bernoulli random number was used to describe this type of behavior. Since the need for a prescription/lab work is

binary the chance that a patient will need this type of medical service when he/she visited the clinic is 50/50.

- Patient arrival behavior – this behavior describes rate of patient arrival at the clinic. For simplification, a constant mean rate of arrival was used and set to one patient arrival at the clinic every minute.
- Professional service behavior – this behavior describes the service time of any medical services provided to the patients. This includes physicians, pharmacists, and lab technician patient processing time. The doctors' service behavior was a uniform distributed random number between 1 and 5 (average service time of 3 minutes) while the lab and pharmacists' service behavior was a uniform distributed random number between 1 and 11 (average service time of 6 minutes).

3.2 Simulation Results

Figure 2 shows a set of graphs of the Agent population versus time for each of the sites with in the simulation. The horizontal axis on each graph shows time while the vertical axis shows number of agents. Patients and therefore Patient Agents were spawned into the system at the rate of 1 patient per minute. The experiment had duration of 30 minutes, and so involved 30 patients and their associated PtAs and clones.

We observe from Figure 2 that the number of Agents in the clinic goes up and down from zero to four agents but in general remains stable within this range. The PtA population does not show system level increases or decreases. Similarly the HRC does not show a marked development in the agent population with either zero or one agent on site throughout the experiment. On the other hand, all the other sites in the system show steady linear increases in agents congregating at the sites. Each of the Doctor's offices as well as the Pharmacy and Laboratory appear to have unsustainably long processing times and the population of patients and patient agents builds up at these sites. After studying these system behaviors the observer may be in a position to suggest mediating action such as decreasing pharmacy and laboratory wait times by adding staff.

3.3 Mathematical Modeling

A macroscopic model that treats agent population at each medical site as the fundamental unit (hence directly describing the characteristic of the system) can also be constructed. The equations used to model the system are presented below. These equations are presented in general form in Tse & Paranjape (2006) but are presented here modified for the specific context of the current experiment. The dynamics of the self-organizing processes can be examined using this model. The model contains a set of coupled rate equations that describe how the agent population at each platform evolves over time. The mathematical model contains one clinics, with two doctors in the clinic, one pharmacy, one testing laboratories and one Health Record Central.

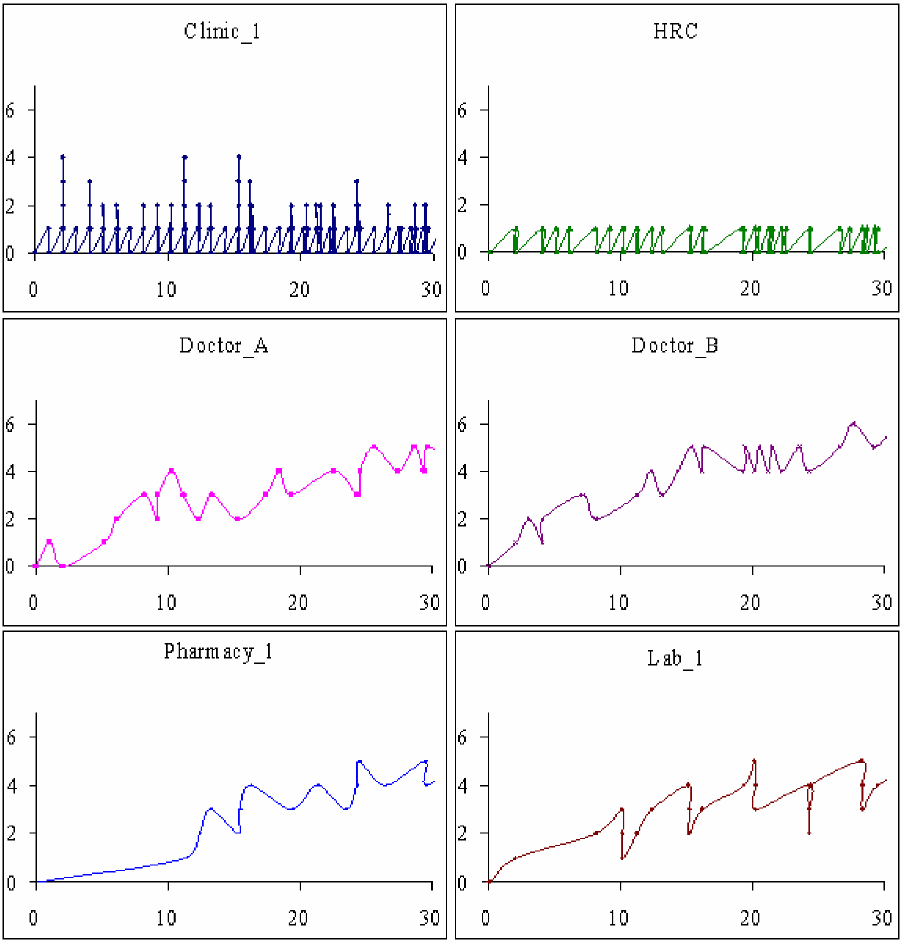


Fig. 2. Evolution of Agent Populations at each of the Sites in the Simulation. These are typical traces showing movements of patient agents in the simulation between six TEEMA platforms for a 30 minutes simulation run. The x-axis is time and the y-axis is number of agents on the site.

The dynamic variables in the model are:

- $N_C(t)$ – is the number of agents in the clinic.
- $N_{CDm}(t)$ – is the number of agents in doctor m’s office in the clinic.
- $N_P(t)$ – is the number of agents in pharmacy.
- $N_L(t)$ – is the number of agents in laboratory.
- $N_{HRC}(t)$ – is the number of agents in the Health Record Central.

The equations governing the behavior of the system are given below:

$$\begin{aligned}\frac{dN_C(t)}{dt} = \lambda + \delta - \alpha_{CD} N_C(t) - \beta_A \alpha_{CR} N_C(t) - \\ \beta_B \alpha_{CP} N_C(t - \tau_{AvgD}) - \beta_C \alpha_{CL} N_C(t - \tau_{AvgD})\end{aligned} \qquad (1)$$

$$\frac{dN_{CD1}(t)}{dt} = \alpha_{CD} N_C(t) \beta_{D1} - 1/\tau_{CD1} \tag{2}$$

$$\frac{dN_{CD2}(t)}{dt} = \alpha_{CD} N_C(t) \beta_{D2} - 1/\tau_{CD2} \tag{3}$$

$$\frac{dN_P(t)}{dt} = (\alpha_{CP} N_C(t - \tau_{AvgD}) \beta_B - 1/\tau_P) \theta(t - \tau_{AvgD}) \tag{4}$$

$$\frac{dN_L(t)}{dt} = (\alpha_{CL} N_C(t - \tau_{AvgD}) \beta_C - 1/\tau_L) \theta(t - \tau_{AvgD}) \tag{5}$$

$$\begin{aligned} \frac{dN_{HRC}(t)}{dt} = {} & \alpha_{CR} N_C(t) \beta_A + \theta(t - \tau_{AvgD} - \tau_L) 1/\tau_L + \\ & 1/\tau_{CD1} + 1/\tau_{CD2} - 1/\tau_{HRC} \end{aligned} \tag{6}$$

And the definitions of the parameters used in the model are:

- λ – the patient arrival rate at the clinic, which is the rate of agent production.
- δ - the rate of agent cloning that occurs at the clinic platform.
- τ_{CDm} - the examination time of doctor 'm' on a patient.
- τ_{AvgD} – the average of all τ_{CDn}.
- τ_P- the service time of an agent in the pharmacy (prescription fill time + prescription pickup time).
- τ_L - the service time of an agent in the lab (time for a patient to come to the lab + test result production time).
- τ_{HRC} - the service time for an agent in a HRC.
- β_A, β_B, β_C, - the probability of a patient who need an update, or a prescription, or a lab work, respectively.
- β_{Dm} - the probability of a doctor being chosen by a patient. It is set to 1/(# of doctors in the Clinic), since each doctor is to be chosen equally.
- α - the transition rates of agents between different platforms, for example: α_{CP} is the rate at which PtAs leave the Clinic platform to go to the pharmacy platform.
- $\theta(t-\tau)$ - a unit step function to ensure certain variables are zero during $t < \tau$.

For simplicity, we assume the following when solving the equations:

- all α to be uniformly distributed in some space, which set to 1.
- all β to be a constant value 0.5, except for β_{PP1} and β_{PL1} which set to 1.
- τ_{CD1} and τ_{CD2} are set to be a constant value of 3, τ_P and τ_L are set to a constant value of 6, while τ_{HRC} is set to 1. These values are the expected value of the uniform distributed random number in our parameters used in the simulation.
- $\lambda = 1$ and $\delta = 1/3$.

3.4 Modeling Results

A set of graphs indicating the solution of the set of equations above is shown in Figure 3. The graphs show the population of PtAs at each of the sites in the Agent model presented as a function of time over a 30 minute interval. The first and most important observation is that the graphs in Figure 3 correspond closely to the graphs in Figure 2 for the agent population in the simulation.

In the HRC graph, we see that there is a small oscillatory behavior which occurs in the value of $N_C(t)$ at the beginning of the experiment. The reason may be that there are many PtAs being created and they are cloning themselves at the same time, causing an increase in $N_{C1}(t)$. As the PtAs leave the Clinic platform for either of the Doctor's platform the number of Agents in the clinic platform, $N_{C1}(t)$, decreases. Thus there are forces increasing and decreasing the agent population in the Clinic platform. As time goes by, the number of agents in each platform becomes stable in the form of a straight line. This suggests that the system adjusts itself to the changes of PtA population in each platform.

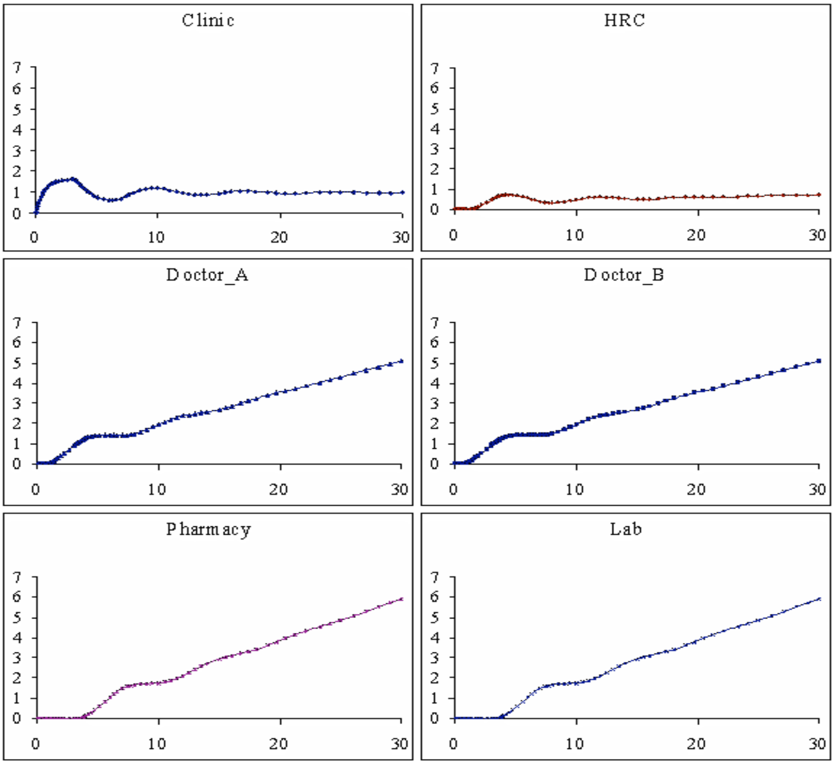


Fig. 3. Asymptotic Behavior of Agent Populations at each of the Sites in the Simulation. These are typical traces showing movements of patient agents in the mathematical model between six TEEMA platforms for a 30 minutes execution. The x-axis is time and y-axis is number of agents on the site.

We can calculate the number of agents in each Doctor platform by noting that 30 PtAs were created in the 30-minute experiment. Since the agents will divide themselves up between the two Doctors platforms each doctor will see 15 patients. Each doctor's examination time is 3 minutes on average, thus at the end of 30 minutes each will have processed only 10 patients leaving 5 patients and their corresponding PtA on each of the Doctor's platform. In fact, this is very close to what we see in Figure 3. Similar calculations for pharmacy and lab platform indicate there should be 6 PtAs in Pharmacy and 6 PtAs in the Laboratory respectively. Again there is good correspondence to what we see in Figure 3.

4 Peer to Peer

4.1 Development of an Equivalent Peer-to-Peer System

Many peer-to-peer techniques are not especially relevant to systems on the scale of the one presented above, since these techniques are designed to deal with situations where particular machines or documents needs to be tracked over very large distributed systems. A scaled up version of the health record system described in this paper could in principle rely on multiple health record centers. To the extent that there were thousands or even tens of thousands of health record centers peer-to-peer techniques such as distributed hash tables (Balakrishnan et al., 2003) could be used to ensure retrieval of a consistent individual health record for each patient. However in a system of the size considered here these techniques are redundant since any request for information can be immediately satisfied by a direct lookup against a list of available locations and dispatch of an agent.

However this does not prevent us from simulating a system that works on peer-to-peer principles, i.e. one that involves the decentralized transmission of messages as opposed to agents. In fact, arguably this kind of system is not a novel peer-to-peer system at all, but simply a decentralized messaging system just like most common network systems today. While much of the web for example relies on a client-server model layered over decentralized messaging systems, many commonly used applications such as email still use a decentralized non-client-server system and have done so for many years; long before the term peer-to-peer was associated with decentralized overlay networks as it is today.

4.2 Simulation Method

A "peer-to-peer" simulation was developed in the Ruby programming language following the specifications of the health record system described earlier in this paper. To be specific the health system described above is a largely centralized system, with most agents being required to pass through a single centralized "clinic" location, with the exception of occasional traffic between the "lab" and "hrc" locations. The possible message pathways are shown in the following diagram:

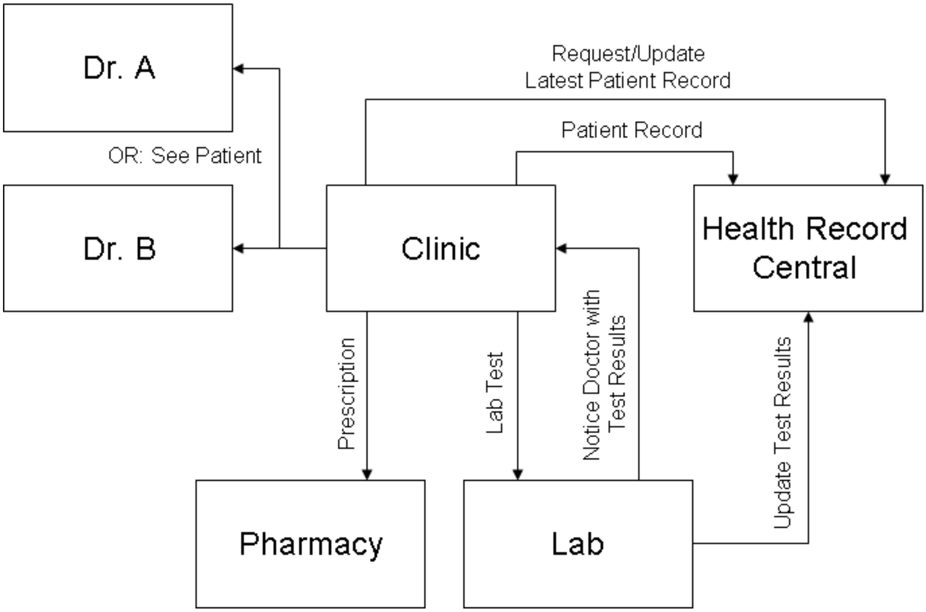
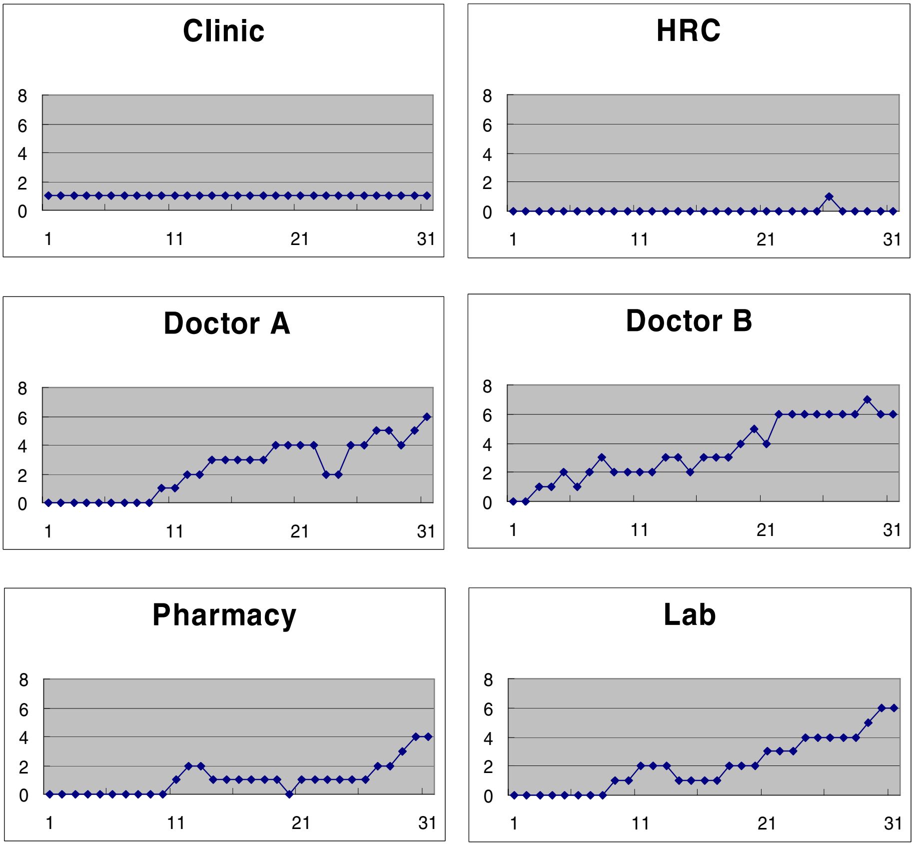


Fig. 4. Locations and information flow in the Health System application

Each location was considered to have a queue for incoming messages, and could be in a “blocked” state if it was unable to process additional messages, e.g. when a doctor was seeing a patient. Message transfer and process time was assumed to take a second, when not associated with professional service behavior, which itself followed a one to five minute Uniform distribution. A new patient arrived at the clinic every minute. The clinic peer would check the locally stored copy of the patient’s health record and request an update from the HRC peer as necessary. Assuming a patients health record was up to date patients would be assigned to the doctor of their preference, and a message containing the health record would then be sent to the doctor peer which would start to process that patient and not process any additional messages until the doctor had finished with that patient. Finished sessions with patients would lead to the doctor peer sending notification and follow up requests to the clinic peer, which would pass them on to the pharmacy and lab as appropriate, with the pharmacy and lab peers blocking as they performed their own professional service behavior. The lab and pharmacy peers would then notify the appropriate peers via the clinic peer (in the case of message to the doctor peer) or directly to the HRC peer as necessary. All probability distributions were set following the pattern described earlier in the paper.

The simulation was run on a single computer, with all “locations” virtually present in the same environment. The results of a single simulation run are shown below. Naturally any simulation should be run a repeated number of times until the expected



Clinic

HRC

Doctor A

Doctor B

Pharmacy

Lab

Fig. 5. Queue lengths over time for the different locations in the peer to peer system. X axis is time in minutes, while y axis is length of queue in messages.

error reaches a threshold level, however the mathematical analysis we shall describe in the next section fully explains the behavior of the system and makes such repeated simulations largely unnecessary. For the moment it is relevant to note that the peer simulation shows remarkably similar behavior to the agent system in that we see queues of increasing length at both doctors and at the lab and pharmacy.

Queueing theory (Allen, 1990) defines the traffic intensity (a) of a simple queueing system as the mean arrival rate (λ) divided by the mean service rate (μ), and states that the number of servers (c) must be greater than this ratio in order to avoid queues of ever increasing length, as shown in equation 7.

$$a = \frac{\lambda}{\mu} < c \tag{7}$$

If we make the plausible assumption that individual message transfer and processing times are trivial in comparison to professional service behavior the fundamental behavior of the system can be modeled in terms of individual doctors as a set of G/U/1 queuing systems. A G/U/1 queueing system has an unspecified inter-arrival time distribution (G or general), a Uniform service time distribution (U) and a single server (1). Patients in our system arrive with a deterministic or constant gap between arrivals (i.e. 1 minute), but they arrive wanting to see a particular doctor. Thus we can model the system as two queues, one for Dr. A and another for Dr. B. Given a sequence of patients wanting to see Dr. A we will eventually have a patient wanting to see Dr. B and so there is a gradually decreasing chance of longer inter-arrival times for each doctor. This amounts to a geometric distribution which counts as a general distribution in queueing theory.

Patients in the system are clearly processed in a Uniform amount of time (between 1 and 5 minutes), but it may not be so obvious how we can think in terms of a single server (the 1 in G/U/1). We can describe the system in terms of a single server because as we will show, the number of patients desiring to visit an individual doctor forms a bottleneck that makes any subsequent waiting times (e.g. at the lab or pharmacy) irrelevant in terms of the overall system behavior. Assuming for the moment that we accept the assertion that doctors create the system bottleneck it follows from our reasoning above that the two doctors should be considered as independent queues since patients have decided in advance which doctor they would like to see. As a result one doctor cannot process the others' patients, giving us effectively two independent G/U/1 type queues.

Let us consider the traffic intensity (a) for one doctor peer. We know that the service time distribution is normal and the arrival time distribution is geometric. The expected values of the service time (E[s]) and inter-arrival time (E[τ]) are thus [min+max]/2 and [1-p/p]+1 respectively, where min is the minimum service time, max is the maximum service time and p is the probability of choosing one doctor over the other. The mean service time of each patient (μ) is the inverse of the expected service time (1/ E[s]) and the arrival rate of patients to each doctor (λ) is the inverse of the expected inter-arrival time (1/ E[τ]). Given that min=1, max=5 and p = 0.5 we know the mean service time (μ) is 1/3 and the mean arrival rate (λ) is 1/2. Unfortunately for our patients this means that the traffic intensity as defined in equation 7 is 3/2 implying that a single doctor (and their patients) will always experience increasing queue lengths. Thus given that patients arrive at this rate, the only way for the system to function effectively is to have more than one doctor available for each patient. Assuming that patients had no choice as to which doctor they saw the system would still be unable to function, as although there are two doctors, the same analysis above applied to an inter-arrival time of 1 minute indicates that at least three doctors are required to ensure non-increasing queue lengths.

Thus the fundamental behavior of our computer implementation, simulation and previous mathematical analysis is explained by simple queueing theory. Our first reaction will likely be that the distributions specified for the simulations are not realistic, and that both patient arrival and service times are much more likely to be described by exponential distributions of some sort, and that it would be an uncommon health system that could always guarantee patients access to the doctor of their choice. Replacing the assumptions of the existing simulations with those found in real world settings would seems a logical next step, at which point it is likely that we would find the doctor bottleneck was removed and require a more comprehensive queueing model to explain system behaviour.

Fortunately Baskett et al. (1975) developed the BCMP Queuing network model that allows systems such as these to be modeled as a group of interconnected queues. This approach would become increasingly valuable for pinpointing particular system bottlenecks as simplifying assumptions, such as message processing time being trivial in comparison to professional service behavior, start to break down.

The particular advantage of a queuing analysis in general is that we can use it to predict precisely how many servers (doctors, labs, pharmacies, etc.) are required to support an operationally effective system rather than rely exclusively on a trial and error approach of making small changes and then repeating simulations or numerical analyses to see if the changes have had the expected effect.

5 Conclusion

In this work we have focused on the development of an agent-based mechanism to support the creation of a self-organizing electronic health record system. The method focuses on the problem of creating complete and consistent records using the strength of agent mobility. We have demonstrated that the agent system will behave as expected by employing both simulation techniques and mathematical modeling.

The second important strength of this type of modeling is that system behaviors such as the linear increase in patients at some of the sites in the health care system can be recognized and addressed before actual system implementation. This approach circumvents system problems by identifying them prior to implementation and allows for effective evaluation of mediating approaches. The approach demonstrates the advantages of test simulation and modeling in agent system design and development.

However our peer-to-peer model demonstrates that the ability to support a health record system with complete and consistent records does not explicitly require the use of mobile agents. Mobile agents are a powerful technology that has particular advantages over simple decentralized message passing systems in that they can transfer code and state along with simple message data. As described by Joseph & Kawamura (2001) there are only a limited set of circumstances in which the particular power of mobile agents can be used effectively, and we have not as yet demonstrated that the nature of the health care record maintenance challenge is one of them. We cannot rule out that as more complex health record behavior is manifested in such as system, mobile agent technology will be required. However in the absence of specific evidence and given that our somewhat simpler peer-to-peer system achieves exactly the same results as our mobile agent system, one has to ask the question of whether mobile agents are perhaps too powerful a technology to be employed for this particular application. Nonetheless we hope that our side by side comparison of an agent system implementation, a peer-to-peer simulation, a mathematical model of agents and queueing theory analysis will prove instructive for system designers in the medical informatics field.

Acknowledgments. The authors thank TRLabs and NSERC for their generous support of this project.

References

Allen, A.O.: Probability, Statistics and Queueing Theory with Computer Science Applications, 2nd edn. Academic Press, San Diego (1990)

Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I.: Looking Up Data in P2P Systems. Communications of the ACM 46(2), 43–48 (2003)

Baskett, F., Chandy, K., Muntz, R., Palacios, F.: Open, closed, and mixed networks of queues with different classes of customers. Journal of the ACM 22(2), 248–260 (1975)

Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press, New York (1999)

Collier, N.: Repast: the recursive porous agent simulation toolkit, [WWW document] (2003), `http://repast.sourceforge.net`

Dick, R.S., Steen, E.B.: The Computer Based Patient Record: An Essential Technology for Health Care. National Academy Press, Washington D.C (1991), `http://www.nap.edu/books/0309055326/html/index.html`

Gibbs, C.: TEEMA Reference Guide Version 1.0. Regina, TRLabs, Saskatchewan, Canada (2000)

Johns, M.L.: Information Management for Health Professionals, Delmar (1997)

Joseph, S., Kawamura, T.: Why Autonomy Makes the Agent. In: Liu, J., Zhong, N., Tang, Y.Y., Wang, P. (eds.) Agent Engineering, pp. 7–22. World Scientific Publishing, Singapore (2001)

Kotz, D., Gray, R., Rus, D.: Future directions for mobile-agent research. Technical Report TR2002-415, Dept. Computer Science, Dartmouth College (January 2002)

Lerman, K., Galstyan, A.: A methodology for mathematical analysis of multi-agent systems (University of California Information Sciences Technical Report ISI-TR-529). California, USA: University of California Information Sciences Institute (2001), `http://www.isi.edu/~lerman/projects/task/`

Lerman, K., Shehory, O.: Coalition formation for large-scale electronic markets [Electronic Version]. In: Proceedings of the Fourth International Conference on Multiagent Systems, Boston, MA, pp. 167–175 (2000)

Luna, F., Stefansson, B.: Economic simulations in Swarm: agent-based modelling and object programming. Kluwer Academic Publishers, Dordrecht (2000)

Martens, R.: TEEMA TRLabs Execution Environment for Mobile Agents. TRLabs, Regina, Saskatchewan, Canada (2001)

Minar, N., Burkhart, R., Langton, C., Askenazi, M.: The swarm simulation system, a toolkit for building multi-agent simulations (1996), `http://www.santafe.edu/projects/swarm/overview/overview.html`

Moreno, A.: Medical Applications of Multi-Agent Systems. In: Paper presented at the 2003 Intelligent and Adaptive Systems in Medicine Workshop, `http://cyber.felk.cvut.cz/EUNITE03-BIO/pdf/Moreno.pdf`

Nealon, J., Moreno, A.: Agent-based applications in health care. In: Paper presented at EU-LAT Workshop 2003 (2003), `http://www.etse.urv.es/recerca/banzai/toni/MAS/papers.html`

Raj, G.S.: A detailed comparison of CORBA, DCOM and Java/RMI, Tech. rep., Web Cornucopia, [WWW document] (1998), `http://my.execpc.com/~gopalan/misc/compare.html`

Smith, K., Paranjape, R., Benedicenti, L.: Agent behavior and agent models in unregulated markets [Electronic Version]. Association for Computing Machinery SIGAPP Applied Computing Review 9(3), 2–12 (2001)

Stone, P., Veloso, M.M.: Multiagent systems: A survey from a machine learning perspective. Autonomous Robots 8(3), 345–383 (2000)

Tecchia, F., Loscos, C., Conroy, R., Chrysanthou, Y.: Agent Behaviour Simulator (ABS): A Platform for Urban Behaviour Development. In: The First International Game Technology Conference and Idea Expo (GTEC 2001) in co-operation with ACM SIGGRAPH and EUROGRAPHICS, Hong Kong, pp. 17–21 (2001)

Tse, B., Paranjape, R.: Macroscopic Modeling of Information Flow in Agent-Based Electronic Health Record System, ch. 17. In: Lin, H. (ed.) Architectural Design of Multi-Agent Systems: Technologies and Techniques, Idea Group Publishing (2006)

Wei, H., Paranjape, R., Benedicenti, L.: Mobile agent network management system performance study in frame relay network. In: Proceedings of the 2002 Institute of Electrical and Electronics Engineers Canadian Conference on Electrical and Computer Engineering, Winnipeg, Manitoba, Canada, pp. 1499–1504 (2002)

Xu, D., Volz, R., Loerger, T., Yen, J.: Modeling and verifying multi-agent behaviors using predicate/transition nets. In: Association for Computing Machinery International Conference Proceeding Series, vol. 27, pp. 193–200 (2002)

Yang, Y., Paranjape, R., Benedicenti, L.: An examination of mobile agents system evolution in the course scheduling problem. In: Proceedings of the 2004 Institute of Electrical and Electronics Engineers Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 657–660 (2004)