ePCRN-IDEA2: An Agent-Based System for Large-Scale Clinical Trial Recruitment

Seyi Feyisetan
Department of Informatics
King's College London
London, UK
seyi.feyisetan@kcl.ac.uk

Gareth Tyson
Department of Informatics
King's College London
London, UK
gareth.tyson@kcl.ac.uk

Adel Taweel
Department of Informatics
King's College London
London, UK
adel.taweel@kcl.ac.uk

Maria Vargas-Vera
Department of Informatics
King's College London
London, UK
maria.vargas-vera@kcl.ac.uk

Tjeerd Van Staa
General Practice Research Database
London, UK
tjeerd.vanstaa@mhra.gsi.gov.uk

Brendan Delaney
Department of Primary Care and Public Health Sciences
King's College London
London, UK
brendan.delaney@kcl.ac.uk

ABSTRACT

Clinical trials are a key method of evaluating the efficacy of healthcare interventions prior to public release. To allow clinical trials to take place, however, it is necessary to recruit sufficient patients for participation. Unfortunately, such recruitment poses a significant challenge though. In this paper, we discuss a novel agent-based system designed to enable patient recruitment; our system, ePCRN-IDEA, has been implemented and is currently under deployment in the UK healthcare system. Through this deployment, however, we have found a number of challenges relating to scalability; consequently, this paper focusses on an extension called ePCRN-IDEA2 that addresses these problems. Specifically, we place agents on General Practitioners' (GP) machines to dynamically compute patient eligibility in real-time during consultations, thereby enabling GUI notifications and immediate recruitment. In ePCRN-IDEA, all agents attempted to compute patient eligibility over all trials, resulting in a huge burden for a large-scale deployment (e.g. 100,000 trials). Therefore, in eCPRN-IDEA2, we have embedded the necessary intelligence in agents to dynamically compute eligibility over the trials that are most likely to match the clinic's and patient's characteristics. Through simulations, we evaluate the approach to show that our decentralised trial selection algorithm can achieve comparable performance to a global knowledge benchmark with far greater scalability.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: [Miscellaneous]; I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Intelligent agents*; J.3 [**Life and Medical Sciences**]: Medical Information Systems

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

General Terms

Design, Performance

Keywords

Clinical Trial Recruitment, Agent-Based Systems, Scalability

1. INTRODUCTION

Clinical trials are the gold standard by which medical research is evaluated. They are used to study various aspects of medical science, as well as being a vital stage in the deployment of new drug treatments. In essence, they involve the testing of new medical theories (e.g. treatments) on real patients to study the effects. For instance, before release, a new drug intended to mitigate the effects of arthritis must first be tested on patients suffering from arthritis to ensure both efficacy and safety. Clearly, however, to enable this, it is necessary to recruit sufficient patients to allow meaningful statical results to be derived. A patient sample size of 10, for example, is unlikely to offer sufficient evidence to persuade regulation boards that a new drug is appropriate for release.

When performing patient recruitment, it is necessary to locate patients who fulfil well defined eligibility criteria, e.g. age, gender, illness etc. This is because, generally, only a small proportion of the population will exhibit the necessary characteristics to make them eligible for participation. For example, testing an arthritis treatment on a healthy individual will clearly not be able to validate its effectiveness. Consequently, it is evident that the recruitment stage of a clinical trial is vital for ensuring that (i) sufficient patients are recruited to enable meaningful statistical results to be gained, and (ii) all patients fulfil the required eligibility criteria to generate accurate results. Unfortunately, however, achieving these two requirements is often highly challenging with recruitment taking up to 30% of the clinical time line, and only 15% of clinical trials finishing on schedule [12]. In fact, a review of the UK Medical Research Council found that only 31% of trials actually recruited to their planned

targets. Clearly, this creates significant overheads, which results in patient recruitment costing 30 - 40% of the entire clinical trial costs. Consequently, improving this process is of paramount importance for the future success of medical research.

The main challenge for patient recruitment is locating and establishing contact with eligible patients within sufficient time to allow them to participate. Therefore, recruitment often involves human recruiters actively visiting clinics in an attempt to locate eligible patients (e.g. by searching local records). Unfortunately, however, this is often complicated due to the geographically distributed nature of primary care clinics (i.e. eligible patients can be thinly spread across many clinics). As such, recruitment can be extremely slow and expensive, particularly for large-scale trials dealing with rare conditions.

To address the above problem, we propose replacing human recruiters with software agents that reside at local clinics. Each agent would maintain a local repository of information about active clinical trials. Whenever a patient enters a clinic for a consultation, the agent would inspect information entered about them in real-time to ascertain if they are eligible for any trials. Through this, instant notifications could be presented to the General Practitioner (GP) during a consultation to inform him/her of the patient's eligibility in real-time. Consequently, this would allow patients to be immediately recruited, negating the need for laborious effort on the part of the patient, clinical researcher or human recruiter.

We have realised the above concepts in an agent-based recruitment system called ePCRN-IDEA [21, 22], which has been implemented in two versions. The first version is under deployment in the UK healthcare system, whilst the second (ePCRN-IDEA2) is a prototype extension currently under evaluation. The second prototype, which is focussed on in this paper, has been developed to address a specific scalability challenge that we found from the first version. In this first implementation, every agent in every clinic downloaded every trial description from a central repository. This, however, resulted in significant processing and storage overheads because each agent was then required to compute the eligibility of a patient against every trial in the system; this could be a huge number, for instance, well over 100k trial are listed on clinicaltrials.gov [2]. Unfortunately, the limited resource capabilities of typical GP machines, as well as the complex nature of certain eligibility criteria, mean that such an approach is unscalable. Therefore, to address this, we have developed a second version, which allows agents to inspect and learn the characteristics of their host clinics to intelligently select the trials that they are most likely to find recruits for. This allows such agents to focus on processing the most suitable trials in a targeted manner. Further, through this intelligence, it becomes possible for agents to cluster into similar groups of trial interest, thereby allowing them to securely share their local repositories rather than using the central store. Through these extensions, we hope to move towards a far larger (pan-European) deployment of ePCRN-IDEA2.

This paper details and evaluates the components and algorithms used in ePCRN-IDEA2, with a focus on ensuring that the system can scale up with increasing numbers of clinics and trials. Specifically, our contributions are as follows:

- An assessment of traditional recruitment approaches and current Clinical Trial Alert systems highlighting their non-scalable nature.
- An extension and evaluation of the ePCRN-IDEA system to ensure scalability in the face of increasing numbers of trials.
- An extension and evaluation of the ePCRN-IDEA system to reduce the loading on a central server by allowing agents to cooperatively share trial information.

The rest of the paper is structured as follows: Section 2 gives the background to the research, leading to the design of ePCRN-IDEA2 in Section 3. Afterwards, an evaluation is then presented in Section 4, ending with the conclusion and future work in Section 5.

2. BACKGROUND

This section presents the background to the research. It first discusses clinical trial recruitment, before talking about scalability and, more generally, about agents in healthcare.

2.1 Clinical Trial Recruitment

Clinical trials are a challenging stage in the research of clinicians due to the complexity of recruiting patients for participation. Many types of trials can suffer from such difficulties; for instance, trials that have potential recruits who are widely distributed over many clinics (e.g. primary care) are extremely difficult to recruit for due to the intensive resource requirements. Studies show that 30% of participating clinics fail to even recruit a single patient [15]. Further, this can be exacerbated by many concerns, especially when dealing with complex eligibility requirements or trials that require immediate actions (e.g. a change of drug treatments).

Clinical trial recruitment is performed by first defining *eligibility criteria* that stipulates the exact characteristics that make a patient eligible for participation (e.g. gender, age, ailments etc.). Once this has taken place, it is then necessary to discover patients who match the criteria, before contacting and recruiting them. Traditionally, locating such patients is achieved using one or more of the following approaches:

- Advertising and public relations: This involves using posters, adverts and brochures to advertise eligibility criteria directly to practitioners and patients.
- Recruiters: This involves sending human recruiters to clinics, usually after feasibility modelling, analysis and site selections, in an attempt to discover patients who match the eligibility criteria.
- Practitioners: This involves doctors meeting periodically to discuss patient treatments and potential trials in an attempt to spot eligible patients during consultation.

These methods, however, are highly time consuming and expensive, particularly for trials that have high patient targets, complex eligibility criteria, rare diseases or involve emergency cases. This has led to the development of Clinical Trial Alert (CTA) systems, which alert practitioners to the eligibility of a patient when they are in consultation. Such systems then allow the practitioner to immediately discuss the trial with the patient, to enable instant recruitment in a

trusted environment (usually through a web interface). This process is achieved by automatically comparing patient information against computable eligibility criteria in real-time during consultations. However most of these systems [8, 4, 6] cater for recruiting patients to a single trial within a single clinic. Other similar techniques have also seen only limited large-scale testing [18]. The challenge of designing generic systems which can handle multiple trials, however, is hampered by the need to perform complex eligibility matching in real-time. Clearly, doing so for large numbers of diverse trials can make the process highly challenging in terms of performance. As of yet, this has led to simplistic CTA systems, which generally deal with small individual trials. We therefore believe that it is vital to address such challenges to enable the deployment of a generic scalable CTA system that can have a real impact on (global) clinical trial recruitment.

2.2 Scalability

Scalability can be defined as the ability of a system to operate within an acceptable performance range in the face of scaling up alternate system parameters (e.g. number of nodes). Currently, most systems adhere to some variation of the client-server model in which a single (logical) server handles requests issued by a number of subordinate clients (as opposed to hybrid and peer-to-peer models [20]). This, for instance, is how the above CTA systems operate, as well as the original ePCRN-IDEA implementation. Clearly, however, this does not scale as a centralised point can only possess a finite amount of resources, whilst the number of subordinate clients can continually increase with ease.

In the context of ePCRN-IDEA, there are two system parameters of importance for ensuring scalability: the number of agents and the number of trials.

As the number of *agents* (clients) increase, the loading on the server similarly increases; consequently, after a certain population is reached, the centralised resources must be upgraded to continue an acceptable quality of service.

Similarly, as the number of *trials* increase, the load on the agents also increases as it becomes necessary to compute a patient's eligibility over a larger set of eligibility criteria within a very strict time frame, i.e. before the patient has left the clinic.[1] This latter point is particularly difficult to manage because it is not possible to conveniently upgrade the resources of each agent as they are distributed throughout the entire country (there are approximately 10k clinics in the UK alone). Unsurprisingly, most GP clinics tend to utilise relatively low resource computers with limited storage capabilities, making it impossible to handle large numbers of trials (e.g. a single trial could be approx ≈1 MB). For instance, our measurements show that a typical desktop machine can take up to 100 ms to compute patient eligibility for a single trial; this means a trial repository size of 100k [2] could take over two hours to process per patient. Thus, it becomes necessary to conceive new ways to improve scalability without over-utilising or extending computing resources.

2.3 Agent Based Healthcare Systems

Agents have emerged as a prominent technology for handling a range of real-world problems [11]. Agents in healthcare have seen widespread investigation; Nealon et.al [14] discussed 11 areas in which agent technology is being applied to support and improve healthcare in Europe. Some of the areas discussed include using agents to integrate [13] heterogeneous patient records, using agents to control cardiac pacing and monitoring the elderly using agent-based teleassistance.

For example, MAID [7] is an agent-based system for integrating heterogeneous data sources within a hospital environment. The hospital studied had 24 departments, each using their own information systems. To address this, agents were constructed to interoperate with each system to monitor changes and retrieve data for insertion into a central repository. In a subsequent work, HealthAgents [9] went beyond MAID to also enable decision support, specifically for diagnosing brain tumours.

A range of agent-based systems have also been proposed for handling distributed expertise. These includes using agents to enable better communication between healthcare workers based on ambient information, e.g. their role, location etc. [17], as well as using agents to remotely monitor patients [10][16]. These systems also often involved data analysis; S(MA)^{2}D, for instance, uses statistical analysis to cluster patients into similar groups [16]. This ability to scalably perform data analysis in real-time, clearly, also shows potential for enabling the type of eligible patient identification discussed previously. Despite this, so far little work has been performed into using agents to improve clinical trial recruitment. Consequently, the rest of this paper explores exploiting the properties of agents to enable scalable patient recruitment.

3. EPCRN-IDEA2 SYSTEM DESIGN

This section presents the ePCRN-IDEA2 recruitment system, which is used to notify GPs of patients' eligibility during consultations.

3.1 Overview

The central aim of our research is to build a scalable system for clinical trial recruitment. At a high level, the system consists of two agents, as shown in Figure 1:

- Trial Agent: This resides at a central point. It holds a record of all available trials, which can then be accessed by GP Agents.
- GP Agent: This resides at a local clinic on a GP's machine. It retrieves trials from the Trial Agent and stores them locally. Whenever a patient enters a clinic, it compares his/her data with all known trials to compute if he/she is eligible for a clinical trial. If so, a pop-up is generated to notify the GP and to allow the patient's immediate recruitment through a web interface.

3.2 Trial Agent

The Trial Agent controls access to the central trial repository, holding an active record of all the trials in the system. It also manages request handling and trial transfers to the GP Agents. On startup, the Trial Agent loads all trials into its local repository. It then registers itself with a Yellow Pages service, which allows other agents to discover its services. In essence, the Trial Agent consists of two main components: the Trial Repository and the Trial Updater. We now briefly cover each of the Trial Agent's functions.

[1]On average, a consultation will last ≈10 minutes.

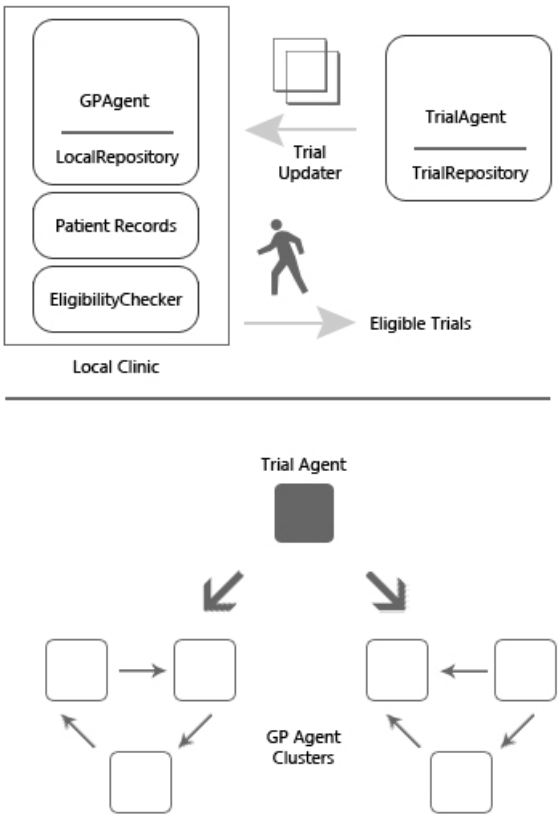

Figure 1: Agent System High Level Design

The Trial Repository component holds a description of each trial, represented using a standardised model e.g. the Biomedical Research Integrated Domain Group (BRIDG) [1] or the Primary Care Research Object Model (PCROM) [19]. These are standard information models that have been developed by clinicians to represent all relevant aspects of primary care research. Currently, PCROM is used, thereby requiring any clinical recruiters to define their trial information in this format. Put simply, PCROM is an XML schema that defines a number of attributes that must be used to describe each trial. To assist in its usage, a user interface has also been developed to automatically generate the format. In terms of ePCRN-IDEA2, the XML includes a trial description (e.g. name, brief overview etc.), a recruitment description (e.g. how many recruits are required) and the eligibility criteria (e.g. what characteristics must a patient fulfil to be considered eligible). Clearly, all clinical concepts in the system must be described using standard ontologies (e.g. Read Codes) to ensure semantic and syntactic interoperability. Collectively, these bodies of data offer the necessary information to decide if a patient is eligible and if the trial still requires recruits, before being able to present a description to the GP. Clearly, of most importance is the eligibility criteria, which can define rules for eligibility using a range of factors, including:

- Read Code(s): The patient must be associated with one or more Read Codes. Read Codes are standard codes that are used to describe clinical concepts, e.g. diagnoses, symptoms, social circumstances.
- Drug Code(s): The patient must currently be prescribed one or more drug treatments. Standard Multilex codes are used to represent drugs.
- Valid Patient List: The patient must be on a list of unique patient identifiers. These are usually generated at a central patient database (e.g. GPRD [23]), which contains collected patient records. This allows more sophisticated eligibility criteria to be pre-computed using full data sets and high performance resources. To ensure privacy, lists of patient identifiers are anonymised before being distributed using a one-way hash function. Mappings are then maintained by the organisation that generated the list of identifiers.

The Trial Agent also supports a variety of other types of criteria, as defined by PCROM. Importantly, combinations of these can be built to create more complex criteria. For instance, a typical form of eligibility criteria might include a list of potentially eligible patients plus a Read Code stipulating joint pain, i.e. to be eligible, one of the predetermined patients must enter the clinic and complain of having joint pain.

When a GP Agent wishes to retrieve trials from the Trial Agent, the request is processed by the Trial Updater component. This component is responsible for matching a GP Agent's characteristics (represented through certain parameters) to the available trials in the Trial Repository. The aim of this is to ensure each clinic retrieves the trials that they are most likely to be able to recruit on. The parameters currently consist of:

- A list of the registered patient identifiers in the clinic. This allows the Trial Agent to ensure that a trial using Valid Patient List eligibility criteria will only be sent to a clinic when the clinic contains one more eligible patients in the list.
- An ordered list of the top r most frequently observed Read Codes. This allows the Trial Agent to discern any specialisation in the clinic (e.g. cancer), to enable matching with eligibility criteria based on Read Codes.
- An ordered list of the top d most popular Drug Codes. This allows the Trial Agent to discern any tendencies to prescribe certain drugs in the clinic, to enable matching with eligibility criteria looking at particular drug usage.

These parameters therefore allow the Trial Updater to best match the clinic's characteristics to a bespoke subset of the globally stored trials. For example, a clinic that has a high number of cancer patients should receive trials that are mostly dealing with cancer. These parameters therefore determine what type of trials are forwarded to each GP Agent on an individual basis. Importantly, the parameters also determine how many trials should be sent based on the local repository size of the GP Agent. This then allows for variations in clinic resources, i.e. it allows clinics with higher capacity computers to locally process more trials. More formally, each request contains the following tuple $\langle V, R, D, n \rangle$,

where V: Set of clinic patients
R: Set of clinic top Read Codes
D: Set of clinic top Drug Codes
n: Clinic local repository size

We also define a function, $t = f(x)$ that retrieves a matching trial t based on an input search criteria x. A counter, i, also maintains the number of currently selected trials. A set, T, is then generated on each request containing all the trials to return to an agent, based on the previously discussed parameters; more formally,

T = Set of trials to be sent to the GP Agent

$$\forall v \in V : if(i < n) \longrightarrow t = f(v)$$
$$T = \{(t \notin T)\}i\text{++}$$
$$\forall r \in R : if(i < n) \longrightarrow t = f(r)$$
$$T = \{(t \notin T)\}i\text{++}$$
$$\forall d \in D : if(i < n) \longrightarrow t = f(d)$$
$$T = \{(t \notin T)\}i\text{++}$$
$$if(i < n) \longrightarrow t = f(rand(x))$$
$$T = \{(t \notin T)\}i\text{++}$$

In essence, the Trial Updater attempts to retrieve n trials by matching available trials with elements from the set V. Each successful match increments the counter i. This is repeated for the other parameters R and D until the size of set T is equal to n. Any extra slots are then filled up with randomly selected trials.

3.3 GP Agent

The GP Agent resides within the local clinic. It is responsible for sending request parameters to the Trial Agent to request relevant trials for its host clinic. It is also responsible for computing the eligibility of a patient in real-time during a consultation (and generating pop-ups). We now briefly cover each of the GP Agent's main functions.

3.3.1 Accessing Central Trial Information

First, to fill up its trial repository, the GP Agent runs an analysis of patients in its host clinic found within the patient records. These are stored in a local Electronic Healthcare Record (EHR) system; essentially, this is a database that is used to store information about each patient. Importantly, it is also actively used by GPs during consultations, thereby offering real-time information to the GP Agent. This allows the GP Agent to inspect information about a patient instantly, whilst the patient is still in consultation. The analysis on the EHR data results in a model detailing the most frequent diseases, most popular drugs and the list of patient identifiers in the clinic. These parameters are then used whenever the GP Agent requests new trials from the Trial Agent, as detailed above. Any retrieved trials are then stored in a persistent local repository to allow for fast access. Importantly, however, this local repository is of a finite size to ensure that the GP Agent is capable of both processing and storing the necessary trials. The default is 100, although this can be dynamically varied based on the memory, storage and processing capacity of the host. The above process is repeated every 12 hours to allow GP Agents to learn of any changes in the central repository.

3.3.2 Accessing Distributed Trial Information

The above section has detailed the default situation in which a GP Agent accesses trial information from the central Trial Agent. This, however, as previously mentioned, is not a scalable option as the number of GP Agents increase. This is because the central Trial Agent has only a finite amount of resources to service the GP Agents' requests. Thus, an increase in the number of GP Agents similarly requires an increase in the resources of the Trial Agent. Something which can be difficult in this domain due to the limited resources of academic research projects. Consequently, to address this concern, GP Agents are also allowed to access trial information from each other in an attempt to alleviate the burden on a central point (i.e. the Trial Agent). To achieve this, GP Agents cluster into groups of clinics that have similar characteristics, thereby allowing them to share trial information. This is because clinics with similar characteristics are likely to require similar trials, therefore allowing the decentralisation of trial distribution. Currently, the characteristics considered in this clustering are the most popular Read Codes and Drug Codes in the clinic. These clusters can be of any size based on the nature of the clinics being interconnected. Thus, to ensure security, all clinics must possess digital certificates, as well as only utilise encrypted communications.

Whenever a GP Agent starts up, it registers itself with the system's Yellow Pages service as a potential trial distributor (using the same service interface as the Trial Agent). Alongside this, it also registers its top illness and drug prescriptions in the clinic (accessed from the EHR software). Once it has done this, it then queries the Yellow Pages service to discover other GP Agents that have the same top diseases and/or drug prescriptions. If none are found, it simply utilises the central Trial Agent. However, if another GP Agent with the same top disease/drug prescription is discovered, it will simply clone its repository. When multiple are found the closest agent with the lowest loading is selected. The above process is then repeated periodically every 12 hours to ensure up-to-date trial information is maintained at each GP Agent.

Importantly, only trials based on Read and Drug codes are exchanged between the different GP Agents; this is for two reasons. First, any trials using Valid Patient List eligibility criteria will likely only be applicable to a small number of clinics (i.e. the clinics in which those patients are enrolled at). Consequently, there is (probabilistically) less benefit in sharing such trials. Second, sharing Valid Patient List eligibility criteria would likely raise certain concerns regarding patient privacy, as the inclusion of a patient on a Valid Patient List could potentially reveal a lot about that particular patient. Therefore, trials containing these lists are not shared.

3.3.3 Computing Eligibility and Recruitment

Once a GP Agent has a number of trials in its local repository, it can begin to compute eligibility for patients. This is performed in real-time whenever a patient enters the clinic. Specifically, the GP Agent is notified by the EHR system whenever a new consultation is opened. The EHR is used by the GP to enter and store information about patients, thereby offering a database of information to compute eligibility over. Importantly, any clinical information encoded in the trials (e.g. disease codes) must use the same syntax and semantics of the EHR data representation. Using this information, the GP Agent compares the patient data against the trials it is aware of to decide if the patient is eligible (e.g. are they the right age range, do they suffer from the correct illnesses etc.). If multiple eligible trials are found, a random one is simply selected; generally, patients will also only be recruited to one trial at a given time. Once

a match is found, the GP Agent generates a GUI pop-up to notify the practitioner, as shown in Figure 2. This pop-up allows the GP to register a response from the patient and to acquire extra information. Importantly, it also allows the patient to be immediately recruited through a web interface. In alternate situations, this web interface can also be used to recruit patients directly without using the GP Agent (e.g. if the GP independently decides a patient is eligible).

Figure 2: GP Agent Pop-up for Notifying Clinician of Patient Eligibility

4. EVALUATION

This section begins by taking a look at the methodology used to evaluate the system. We then seek to evaluate how effectively the system can scale up in terms of increasing numbers of trials and clinics. Specifically, we look at how effectively we can maintain performance and overheads in the face of these increasing variables.

4.1 Methodology

To evaluate ePCRN-IDEA2's scalability we have performed a number of system simulations. To achieve this, however, it is first important to create a realistic simulation environment and workload.

First, it is necessary to model how diseases are distributed throughout clinics (and patients). Second, it is necessary to understand what types of trials might typically be injected into ePCRN-IDEA2. Then, third, the characteristics and behaviour of the patients need to be modelled. Beyond this, it is also important to define the evaluative metrics that will be inspected. This section presents the evaluation methodology, looking at these four concerns. To achieve this, a prototype of the system has been built using the Java Agent Development (JADE) framework. This has then been used to perform a number of simulations with various parameter setups. An overview of the default setup is provided in Table 1; unless otherwise stipulated, these parameters are used in all experiments.

4.1.1 Modelling Disease Distribution

To present a realistic evaluation, it is important to have realistic data regarding disease frequency. This is so that the simulation can model the types of diseases that patients are likely to report as suffering from. To achieve this, we use a Zipf Distribution [5]. According to Zipf's Law, the frequency of any disease is inversely proportional to its rank in the frequency table. This results in a small number of diseases being frequently encountered (e.g. flu) and a 'long tail' of diseases that are far rarer (e.g. papillitis). This can therefore be used to model the ailments reported by patients during consultations. To validate this choice, the generated distribution of diseases was compared against data provided by the World Health Organization (WHO) [3], which confirmed its accuracy. Diseases were generated using an alpha parameter (skew) of 1 and a set size of 10,000 to mirror the WHO distribution.

Table 1: Default Parameter Setup

Parameter	Value
# Patients Per-Clinic	200
Total # Patients	5,080
# Consultations Per-Simulation	100
# Trials in Local Repository	100
Max # Trials in Global Repository	10,000
Distribution of Read/Drug Codes	Zipf
Skew of Read/Drug Codes (for Trials)	0.9
Skew of Read/Drug Codes (for Patients)	0.4
Total # Read/Drug Codes	10,000

4.1.2 Trial Workload

Clearly, it is important to model realistic trial workloads in the system. To do this, we select a variety of possible types of eligibility criteria to test ePCRN-IDEA2 with. The trials generated were of four variants with eligibility criteria consisting of (*i*) Read Codes only, (*ii*) Valid Patient Lists only, (*iii*) Valid Patients and Read Codes, and (*iv*) Valid Patients, Read Codes and Drug Codes. These represent typical trial types that are usually encountered in clinical research. The rest of this section details how each trial type was generated. Each trial's eligibility criteria was generated with at least one of the following parameters:{V, R, D}, as described below:

Eligibility Criteria with Read Codes only (R): A random value r within a pre-set range $(1 - 2)$ is selected as the size of the set R. Read Codes are then assigned to fill up set R using the Zipf distribution (with a default skew of 0.9) from a global set of 10,000 Read Codes.

Eligibility Criteria with Valid Patient Lists only (V): A random value v within a pre-set range $(5 - 15)$ is selected as the size of the set V. Patient IDs are then randomly assigned to fill up set V from a set of 5,080 patient IDs. Clearly, a real clinical trial would use a far larger list size, however, scaling down both the pre-set range and global population size allows us to tractably emulate large-scale simulations.

Eligibility Criteria with Valid Patient Lists and Read Codes (V, R): This involved the two processes listed above to assign Read Codes and valid patients. Once the Read Codes and the valid patient list have been set, they

are combined and written into the trial.

Eligibility Criteria with Valid Patients, Read & Drug Codes (V, R, D): This involved the three processes listed above. A Drug Code is selected the same way the Read Code was selected. Once the coded information and the valid patient list have been created, they are combined and written into the trial.

4.1.3 Patient Workload

Last, it is necessary to simulate patients and their characteristics when arriving at clinics. The set of patients in a clinic is generated by random selecting patient IDs from the central patient database of 5,080 patients within the range assigned to the clinic. A patient history is then built for each patient by selecting a random number of visits between one and six, then assigning treatment Read Codes for each visit (using the Zipf distribution). This is then stored in the local EHR database of the clinic.

After creating the patient history, it is also necessary to simulate patient arrivals in the clinic (as eligibility is only ever checked when a patient is in consultation). To do this, on each simulation round, a random patient is selected for a visit. Read Codes and Drug Codes are then assigned for the current visit using a Zipf distribution (with a default skew of 0.4). The skew for each clinic is assigned based on clinic type, e.g. specialist clinics have a more skewed distribution.

4.1.4 Evaluation Metrics

Alongside the above parameters, it is also necessary to define the metrics by which we measure the scalability of the system. We do this through two values: performance and overhead.

We measure the *performance* of the system by the number of pop-ups[2] generated. Clearly, the ideal result is that every patient who enters a clinic and is eligible for a trial should be notified. However, practically speaking, this is not possible as it requires every GP Agent to know about every trial in the system (this is the original design of ePCRN-IDEA). Therefore, we use this global knowledge scenario as the benchmark by which we evaluate the effectiveness of ePCRN-IDEA2's approach of intelligently selecting trials on a per-clinic basis. Consequently, we represent the system performance as the percentage of pop-ups created in ePCRN-IDEA2 when compared against those that could have been generated if all agents knew of all trials.

We next measure the *overhead* of the GP Agent; in the above global knowledge benchmark, it is necessary to have very large local trial repository sizes, as well as massive processing capacities to compute eligibility in real-time. As previously mentioned, a global trial repository could take hours to process patient eligibility for, making the intelligent selection of trials vital. Consequently, to measure the overhead required to achieve a given performance level, we use the local repository size, as this is representative of not only the per-agent storage capacity required but also the processing costs for eligibility computation. Thus, we contrast the number of pop-ups a GP Agent can generate against the quantity of resources required to achieve them.

Last, we also measure the *overhead* of the Trial Agent, which is important when considering the feasibility of increasing the number of participating clinics in the recruitment system. To measure this, we simply use the number of active connections from GP Agents to the Trial Agent. This allows us to infer the loading that the Trial Agent has at any given time.

[2]A pop-up represents a patient who has been found eligible for an available trial.

4.2 Scaling the Number of Trials

As the number of trials increase in the system, it is important that ePCRN-IDEA2 can maintain a high number of pop-ups, whilst still ensuring each GP Agent does not get allocated too many trials to process. Ideally, GP agents will be able to keep a high number of pop-ups with only a limited size of local repository (i.e. high performance, low overhead).

To evaluate this, a number of different trial types (as described above) are tested in the system to measure their performance and overhead. Each trial type was evaluated using three different approaches to distributing trials from the central Trial Agent to the GP Agents. These are as follows: (*i*) distributing all trials to every GP Agent (global knowledge benchmark), (*ii*) retrieving a random set of n trial for each GP Agent, and (*iii*) intelligently selecting n trials based on the results of profiling the host clinic (i.e. using the algorithm presented in Section 3). The rest of the section presents results from simulating each trial type in ePCRN-IDEA2.

4.2.1 Trials with 1 Read Code (R)

The first type of trial tested simply contained eligibility criteria using a single Read Code (e.g. all patients who have diabetes). The single Read Code to be included within each trial was selected from a pool of 10,000 Read Codes using a Zipf distribution with a skew of 1. To evaluate the system, we compare the number of pop-ups with the theoretical maximum that would be possible by having all agents know about all trials.

Table 2: Trial Eligibility Criteria with 1 Read Code

All Trials		Random Trials		Selected Trials	
Trials	Pop-ups	Trials	Pop-ups	Trials	Pop-ups
1000	54	1000	4	1000	52
2000	65	2000	2	2000	52
3000	67	3000	1	3000	50
4000	70	4000	0	4000	51
5000	71	5000	0	5000	50
6000	72	6000	0	6000	51
7000	72	7000	0	7000	53
8000	73	8000	1	8000	53
9000	75	9000	0	9000	51
10000	77	10000	0	10000	50
Avg	70	Avg	1	Avg	50
Ovh	100%	Ovh	1%	Ovh	1%
Perf	100%	Perf	1.4%	Perf	71.4%

Table 2 details the results from the simulations. It can be seen that randomly selecting trials to fill up the local repository leads to a significantly lower number of pop-ups compared to the benchmark of global knowledge. This is due to the obvious difficulty of randomly selecting the most

appropriate Read Codes for a given clinic. In contrast, intelligently selecting trials with a particular Read Code based on the profile of the clinic led to a much higher number of pop-ups (an average of 50). In fact, for 10,000 trials, 65% of pop-ups could still be generated using only a local trial store of only 100. This is because the GP Agent was able to run a profile of its clinic and download trials that its patients were more likely to be eligible for. Importantly, performance remained relatively constant up to a size of 10,000 trials, without needing to extend the size of the local repository beyond 100.

4.2.2 *Trials with Valid Patient Lists (V)*

The second set of trials generated consisted of patients whose eligibility has been pre-computed at a centralised database. This form of eligibility criteria therefore simply consists of a list of all the patient identifiers who are eligible.

Table 3: Trials with Valid Patient Lists

All Trials		Random Trials		Selected Trials	
Trials	Pop-ups	Trials	Pop-ups	Trials	Pop-ups
1000	43	1000	4	1000	50
2000	72	2000	6	2000	52
3000	84	3000	6	3000	50
4000	89	4000	5	4000	54
5000	94	5000	7	5000	52
6000	98	6000	1	6000	53
7000	99	7000	4	7000	52
8000	96	8000	6	8000	51
9000	100	9000	2	9000	52
10000	100	10000	5	10000	50
Avg	87	Avg	5	Avg	52
Ovh	100%	Ovh	1%	Ovh	1%
Perf	100%	Perf	6%	Perf	60%

From Table 3, it can be seen that randomly filling the local repository led to only an average of only 5 pop-ups. This is because such trials can only recruit from a small number of clinics that the required patients are enrolled at. Consequently, random selections are highly suboptimal. In contrast, intelligently selecting trials based on the registered list of patients in the clinic resulted, on average, in 60% of the pop-ups of the global knowledge benchmark. This is because each GP Agent was able to fill its repository with most of the trials that its patients had been pre-computed as eligible for. This remained constant even when the local repository size was only 1% of the trial repository size. Consequently, even when the local repository size was increased to 200, intelligently selecting trials resulted in the same number of pop-ups as retrieving all the trials.

4.2.3 *Trials with Valid Patients and 1 Read Code (V,R)*

The third set of trial eligibility criteria generated consisted of a list of eligible patients who must also be diagnosed with a specific illness (e.g. Mr. Smith is eligible if he is also diagnosed with diabetes).

Intelligently selecting trials to fill up the local repository based on the patients in the clinic resulted, on average, in 75% of the pop-ups obtained from downloading all the trials. This was because the GP Agent was able to request trials based on its registered patient list and an analysis of its top Read Codes. This result remained constant even when the local repository size was only 1% of the central trial repository size. Even when the local repository size was increased to 200, intelligently selecting trials resulted in the same number of pop-ups as retrieving all the trials, indicating the scalability of the algorithm.

Table 4: Trials with Valid Patients and 1 Read Code

All Trials		Random Trials		Selected Trials	
Trials	Pop-ups	Trials	Pop-ups	Trials	Pop-ups
1000	4	1000	2	1000	7
2000	14	2000	2	2000	13
3000	17	3000	2	3000	18
4000	22	4000	4	4000	19
5000	23	5000	5	5000	22
6000	31	6000	2	6000	23
7000	30	7000	2	7000	24
8000	34	8000	2	8000	25
9000	34	9000	4	9000	24
10000	35	10000	2	10000	26
Avg	24	Avg	3	Avg	18
Ovh	100%	Ovh	1%	Ovh	1%
Perf	100%	Perf	12%	Perf	75%

4.3 Scaling the Number of Clinics

As the number of clinics (and GP Agents) increase, the loading on the central Trial Agent similarly increases. Consequently, to ensure scalability, we consider it necessary to better utilise the global system resources to alleviate this burden. In the context of ePCRN-IDEA2, the goal is to decentralise the distribution of trials as much as possible, thereby reducing the burden on the central Trial Agent. This involves GP Agents connecting to other GP Agents to request trials, rather than always utilising the Trial Agent. To measure this, we use the number of dependent connections to the central Trial Agent as an overhead metric. Clearly, this indicates the level of loading on the central repository and should therefore be kept low.

Figure 3 presents an exemplary graph of the GP Agent interconnections during a 10 node simulation. It can be seen that the agents are clustering together based on their clinic's characteristics. Importantly, only 2 nodes were required to directly connect to the Trial Agent. To extend these results, simulations were performed with agent populations of up to 100. In each case, only 2 connections were maintained to the Trial Agent, with each GP Agent being able to effectively utilise its peers' resources.

These overhead results can also be contrasted with the performance achieved (measured using the number of pop-ups). To achieve this, the central Trial Repository was set-up with 10,000 trials, each with eligibility criteria consisting of 1 Read Code. Each GP Agent then retrieved a set of trials by either connecting to the Trial Agent or a peer GP Agent. The average number of pop-ups for each cluster size are presented in Table 5. The results here were very similar to those in Table 2 for 10,000 trials (around 50 pop-ups). This evidences the fact that the system can maintain a similar level of performance whilst also alleviating the loading on the central Trial Agent.

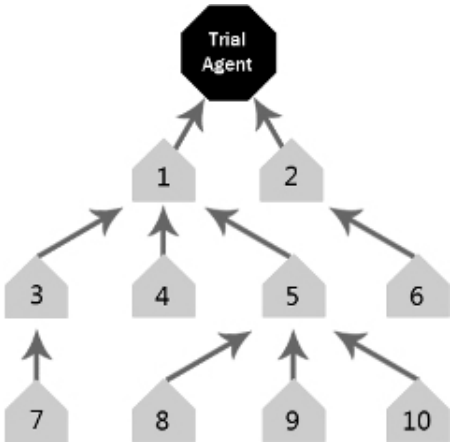

Figure 3: GP Agent Connections: Cluster of 10 GP Agents

Table 5: Trial Repository Size: 10,000 Local Repository Size: 100

All Trials		Selected Trials	
Cluster Size	Av. Pop-ups	Cluster Size	Av. Pop-ups
10	74	10	51
20	75	20	51
30	76	30	52
40	73	40	50
50	75	50	51
60	77	60	52
70	74	70	53
80	73	80	51
90	75	90	52
100	75	100	50

4.4 Summary and Discussion

Table 6 presents a summary of the scalability results; the percentages refer to the average performance and overhead levels of each selection method compared to the benchmark of global knowledge. As stated above, it can be seen that utilising random trial selections to address the global knowledge challenge resulted in consistently low performance, even though it does manage to maintain a low overhead (requiring only 1% of the global repository size). In contrast, it can be seen that ePCRN-IDEA2's approach of intelligently selecting trials results in significantly higher performance, whilst still maintaining very small local repositories. In fact, at its lowest performance, ePCRN-IDEA2 still manages to maintain 60% of the pop-ups that the global knowledge benchmark achieves.

Clearly, these results have therefore shown the unscalable nature of attempting to maintain global knowledge in a large-scale agent-based system. Our simulations have been based on repositories of 10,000 trials, however, this can easily extend well beyond this to in excess of 100,000 trials [2]. Consequently, ePCRN-IDEA2's approach is vital for ensuring scalable clinical trial recruitment. We have shown that it is possible to effectively target the distribution of trials on a per-clinic granularity. Specifically, in some settings, up to 75% of recruitment opportunities (pop-ups) can be maintained, even when reducing the local trial knowledge to just 1% of the global set. This suggests that large-scale recruitment can, indeed, be achieved without any need to upgrade local clinic computing resources. Beyond this, we have also evaluated the potential of reducing server loading at the Trial Agent by allowing GP Agents to share trial information. It has been shown that using peer GP Agents can easily reduce this load whilst, importantly, maintaining similar levels of performance.

Table 6: Summary of Performance and Overhead under Different Trial Types

	Random Trials		Selected Trials	
Trial Type	Perf	Ovh	Perf	Ovh
1 Read Code	1.4%	1%	71.4%	1%
Pat Lists	6%	1%	60%	1%
Pat List + Code	12%	1%	75%	1%

5. CONCLUSIONS AND FUTURE WORK

This paper has discussed the importance of clinical trials and the challenge of recruiting sufficient patients into them. It has looked at the current ways recruitment is carried out and the potential of using software agents to carry it out in a more scalable manner. This has led to the design and implementation of an agent-based system, ePCRN-IDEA2, which attempts to enable real-time patient recruitment on a large-scale. This system places agents in clinics with the responsibility of notifying practitioners whenever a patient, who is eligible for a clinical trial, is in consultation. This allows recruitment to be immediately performed before the patient has left the clinic. Further, to ensure long-term scalability, we have presented a way in which agents can intelligently select the trials that they consider themselves best able to recruit for (based on their host clinic). Through this, we have addressed the need for each agent to maintain global knowledge of all trials, thereby dramatically improving the ability of the system to scale up.

From this phase-2 prototype, we have identified a number of further lines of work. We believe it is important to extend the intelligence of the agents further, allowing them to gain a better understanding of their clinic. This, for instance, should involve improving the method by which agents can model a clinic's characteristics. This could also incorporate inferences regarding a given patient's likelihood to accept. Beyond this, it is clearly important to extend the evaluation to look at such things as larger trial bases, more divergent/convergent clinics and varying patient characteristics. Also, in this paper, the system has been evaluated under a synthetic workload; future evaluations should therefore endeavour to utilise more realistic setups. This will soon become possible using logging traces taken from the currently deployed ePCRN-IDEA system. Consequently, an important future step is using these traces to re-execute our simulations. Finally, it is our longer-term goal to also introduce this functionality into a working deployment, so that real results can be acquired regarding both performance and overheads.

6. REFERENCES

[1] Bridg project documentation and release notes for r3.

[2] Clinicaltrials.gov registry. http://clinicaltrials.gov/.

[3] Statistical annex 121. *The World Health Report 2004*, pages 2002–2004, 2004.

[4] L. B. Afrin, J. C. Oates, C. K. Boyd, and M. S. Daniels. Leveraging of open emr architecture for clinical trial accrual. *AMIA Annual Symposium proceedings AMIA Symposium AMIA Symposium*, 2003:16–20.

[5] S. K. Baek, S. Bernhardsson, and P. Minnhagen. Zipf's law unzipped. *New Journal of Physics*, 13(4):22, 2011.

[6] A. J. Butte, D. A. Weinstein, and I. S. Kohane. Enrolling patients into clinical trials faster using realtime recuiting. *Proceedings of the AMIA Symposium*, pages 111–115, 2000.

[7] R. Cruz-Correia, P. Vieira-Marques, P. Costa, A. Ferreira, E. Oliveira-Palhares, F. Araújo, and A. Costa-Pereira. Integration of hospital data using agent technologies - a case study. *AI Commun.*, 18, August 2005.

[8] P. J. Embi, A. Jain, J. Clark, S. Bizjack, R. Hornung, and C. M. Harris. Effect of a clinical trial alert system on physician participation in trial recruitment. *Archives of Internal Medicine*, 165(19):2272–2277, 2005.

[9] H. González-Vélez, M. Mier, M. Julià-Sapé, T. Arvanitis, J. García-Gómez, M. Robles, P. Lewis, S. Dasmahapatra, D. Dupplaw, A. Peet, C. Arús, B. Celda, S. Van Huffel, and M. Lluch-Ariet. Healthagents: distributed multi-agent brain tumor diagnosis and prognosis. *Applied Intelligence*, 30, 2009.

[10] V. 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(4):528–537, dec. 2005.

[11] M. Luck, P. McBurney, and C. Preist. A manifesto for agent technology: Towards next generation computing. *Autonomous Agents and MultiAgent Systems*, 9(3):203–252, 2004.

[12] A. McDonald, R. Knight, M. Campbell, V. Entwistle, A. Grant, J. Cook, D. Elbourne, D. Francis, J. Garcia, I. Roberts, and C. Snowdon. What influences recruitment to randomised controlled trials? a review of trials funded by two uk funding agencies. *Trials*, 7(1):9, 2006.

[13] M. Nagy and M. Vargas-Vera. *Towards an Automatic Semantic Data Integration: Multi-agent Framework Approach.* Chapter in Sematic Web. In-Tech Education and Publishing KG, 2010.

[14] J. Nealon. Agents applied in health care. *AI Communications*, page 22, 2005.

[15] R. Nitkin. Patient recruitment strategies. Training workshop conducted by National Institutes of Health, Bethesda, Md, 2003.

[16] A. Rammal, S. Trouilhet, N. Singer, and J.-M. Pécatte. An adaptive system for home monitoring using a multiagent classification of patterns. *Int. J. Telemedicine Appl.*, 2008:3:1–3:8, January 2008.

[17] M. D. Rodríguez, J. Favela, A. Preciado, and A. Vizcaíno. Agent-based ambient intelligence for healthcare. *AI Commun.*, 18:201–216, August 2005.

[18] B. L. Rollman, G. S. Fischer, F. Zhu, and B. H. Belnap. Comparison of electronic physician prompts versus waitroom case-finding on clinical trial enrollment. *Journal of General Internal Medicine*, 23(4), 2008.

[19] S. M. Speedie, A. Taweel, I. Sim, T. N. Arvanitis, B. Delaney, and K. A. Peterson. The primary care research object model: A computable information model for practice-based primary care research. *Journal of the American Medical Informatics Association*, 15(5):661–670, 2008.

[20] G. Tyson, P. Grace, A. Mauthe, G. Blair, and S. Kaune. A reflective middleware to support peer-to-peer overlay adaptation. In *Distributed Applications and Interoperable Systems (DAIS)*. 2009.

[21] G. Tyson, A. Taweel, S. Miles, M. Luck, T. V. Staa, and B. Delaney. An agent-based approach to real-time patient identification for clinical trials. In *Proc. of the 4th Intl. Conference on eHealth (eHealth)*, 2011.

[22] G. Tyson, A. Taweel, S. Zschaler, T. V. Staa, and B. Delaney. A model-driven approach to interoperability and integration in systems of systems. In *Proc. of Workshop on Model-Based Software and Data Integration (MBSDI)*, 2011.

[23] T. Williams, T. Van Staa, S. Puri, and S. Eaton. Recent advances in the utility and use of the general practice research database as an example of a uk primary care data resource. *Therapeutic Advances in Drug Safety*, 2012.