A Multi-Agent Platform for Automating the Collection of Patient-Provided Clinical Feedback

Zina M. Ibrahim*
King's College London
London, UK
zina.ibrahim@kcl.ac.uk

Lorena Fernández de la Cruz
Karolinska Institutet
Stockholm, Sweden
lorena.fernandez.de.la.cruz@ki.se

Argyris Stringaris
King's College London
London, UK
argyris.stringaris@kcl.ac.uk

Robert Goodman
King's College London
London, UK
robert.goodman@kcl.ac.uk

Michael Luck
King's College London
London, UK
michael.luck@kcl.ac.uk

Richard J.B. Dobson
King's College London
London, UK
richard.j.dobson@kcl.ac.uk

ABSTRACT

Clinical outcome measures are well-established quantitative assessments of medical treatment that are crucial for understanding the effectiveness of prescribed drugs. However, recent studies suggest that timely and long-term clinical outcome measures are not usually found in the patient records. The study attributes the lack of quantitative follow-up to the reluctance of the clinicians treating the patients, whose active schedules incline them to use (insufficient) brief descriptions of the cases, greatly reducing the quality of care. To ease the clinicians burden and improve the process of patient follow-up while simultaneously dealing with the complexity of our domain, we designed and deployed a multi-agent platform to automate the process of collecting patient-provided clinical outcome measures without clinicians intervention. The system also acts as a mediator between patients and scientists, by seeking patient consent to incorporate anonymised versions of the collected data into scientific studies to which the patients are found eligible. This paper presents the design, prototype and deployment results of our system, highlighting foreseen clinical merits and documenting challenges faced throughout our experience.

Categories and Subject Descriptors

1.2.11 [**Distributed Artificial Intelligence**]: Multi-agent systems

General Terms

Design, Management, Reliability, Security, Standardization

Keywords

healthcare; multi-agent; automation; electronic health records

1. INTRODUCTION

The path traveled by a drug from laboratories to pharmacy shelves is a multi-phase process unique to every drug. The pre-release stage usually includes *Randomised control trials* (RCTs), studies whereby a random sample of patients undergoes treatment for a specific period of time [9]. RCTs evaluate the drug's capacity to induce beneficial effects on the patients during the study, i.e. the drug's *efficacy*. However, in order to establish the degree to which the same drug improves patients in the uncontrolled turbulence of everyday practice, *naturalistic treatment trials* (NTTs) are used [9].

NTTs evaluate the drug's *effectiveness*, its ability to induce long-term results in a large and heterogeneous patient population. They record patients' progress via well-established measures of medical care, known as *clinical outcome measures*, during routine clinical practice. Ideally, clinicians are responsible for inputting the outcome measures into the patient's records. However, recent studies have shown that clinicians choose to rely on their impressions in assessing symptoms and (potentially life-threatening) side effects and view outcome measures as a clinically-irrelevant bureaucracy [18]. For example, a recent study evaluating the effectiveness of treatment received by children with Attention Deficit Hyperactivity Disorder (ADHD) in the largest mental health institute in Europe found that only 1% of the patients have long-term follow-up measures recorded, and attributes the small percentage to the clinicians' reluctance to record the rather lengthy outcome measures into their databases; quite concerning results [5].

As a result, NTTs take the costly and resource-consuming form of studies whereby designated personnel visit the patients on predefined occasions to record their progress. Because patient records are strictly confidential, these studies are also burdened with time delays for obtaining the appropriate consent and approvals.

Having the above in mind, we began an effort to build a system to tackle the root of the problem: the lack of timely-recorded outcome measures data in the patients' records. The aim is to enable patients to take control of the process of feedback provision to their electronic health records, and making the data usable for clinical evaluation purposes as well as scientific inquiry by authorised researchers. In collaboration with healthcare professionals through an extensive requirements elicitation process, we soon discovered that such system will not only be complex, but also involves distributed entities and is burdened by the following issues (1) patient records are confidential, with strict rules prohibiting access to anyone not directly involved in the treatment (2) patients tend to visit several healthcare institutions during the course of their treatment, distributing their records across different organisations, each with its own set of rules, regulations and internal protocols

[*] Corresponding Author

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

(3) the structure, languages and terminologies used to build electronic health records systems largely vary across hospitals, and (4) different stakeholders with different privileges have real interest in the feedback data, including clinicians, patients and authorised research scientists. The above requirements naturally led to an agent-based infrastructure, as we believe they fit the strengths and capabilities of agent-based systems. The result is APPROaCh (Agent Platform for automating patient PROvided Clinical outcome feedback). Our work relies on the idea of replacing clinicians, who are naturally reluctant to record clinical outcome measures as part of their routine work, by software agents. The multi-agent computing infrastructure supports patient-provided clinical outcome measures collection whereby the patients, guided by software agents, are capable of feeding their clinical outcomes directly back to the electronic health records of their respective organisations. We implemented a system which (1) provides a cross-organisational ontology to tackle the heterogeneity of the data found in patient records (2) provides an online tool designed to be used by patients for the provision of treatment outcome measures regardless of where they are being treated and whether they visit one or several healthcare institutions (3) automates an anonymisation pipeline designed to create research-friendly views of the patient records in real-time, eliminating the privacy issues associated with NTTs (4) enables scientists in different research institutions to design and conduct targeted studies using the anonymised version of the patient records (5) reduces cost by automating the process of obtaining patient consent in real-time, following-up and obtaining feedback from the patients at intervals predetermined by clinicians or research scientists and (6) engages with patients on timely bases to take feedback and provide detailed and well-studied guidance reports. This improves the patients' understanding of their case, facilitates self-care and increases patient involvement with feedback provision. In addition to improving data collection for research and quality control, APPROaCh is designed to increase patient awareness and ease the financial and resource burden of healthcare research. We designed APPROaCh using theO-MaSE methodology [6] and have recently deployed the majority of our design in the South London and Maudsley National Health Services Foundation Trust (SLaM)[1] in London, United Kingdom.

There have been efforts to use multi-agent systems to remedy some of the issues within NTTs and for post-marketing drug evaluation in general. SARMA [21] and *ePCRN-IDEA* [16] build multi-agent systems to automate the process of obtaining patient consent for clinical trial recruitment to eliminate the privacy and distribution issue. Those systems focus on increasing the number of patients willing to participate in NTTs using automation of consent in collaboration with clinicians and do not examine the automation of the actual trial. [14, 11] complement the clinicians' work by building systems to detect unknown adverse drug reactions by examining the records of patients prescribed these drugs.

The paper is structured as follows. After some background in Section 2, Section 3 shows the design of APPROaCh. We discuss implementation details and deployed prototype in Section 4 and conclude with a discussion in Section 5.

2. BACKGROUND

2.1 Clinical Outcome Measures

A notion central to our work is that of a *clinical outcome measure*, a well-defined numerical quantity for evaluating the quality of medical care. Clinical outcome measures numerically establish the standard against which the end result of a treatment is assessed. An extensive area of medical research is centred around the careful design of clinical outcome measures to test the efficacy and effectiveness of medical treatment for different disorders [22].

A special class of outcome measures are *patient-reported outcome measures* (PROMs), which take the form of a questionnaire to which the responses are collected from the patient. PROM questions can range from physical measurements (e.g. blood pressure) to more exploratory questions (e.g. do you find it difficult to make friends?). PROMs are established as useful tools for improving the quality of care given to patients [10].

Nevertheless, PROMs are not currently enforced as part of routine clinical practice. As mentioned in the introduction, they are seen as time-consuming and rarely acknowledge by clinicians. A study conducted in UCLA shows that long-term outcome measures are scarce in electronic health records, with availability ranging from 1-10% [23]. The study concludes that consolidating outcome measures recording with routine clinical practice is currently one of the bottlenecks of practice-based medical research.

2.2 Agents, Healthcare and Research

Healthcare systems present several requirements which make them attractive multi-agent environments. Healthcare systems are by definition distributed, spanning different hospital units and connecting with other systems (e.g. other hospitals or general practitioner clinics). They embody decision making achieved through the coordinated efforts of a set of autonomous entities, i.e. the different professionals providing care (clinicians, nurses, etc..). In addition, they are used to solve complex problems requiring the satisfaction of different and possibly conflicting constraints, e.g. organ transplant management [3], or problems which naturally decompose into (possibly interdependent) subproblems, e.g. decision support systems [7, 12, 19]. Finally, healthcare systems require the collection and management of heterogeneous data ranging from clinical notes to medical information on the web to medical ontologies [17]. As a result, multi-agent systems have found many medical application ranging from resource allocation [2] to remote care [1, 15] (see [13] for a recent review).

The requirements presented by incorporating clinical outcome measures collection into routine care, and possibly using them for auditing and research purposes, are not different from the above. To begin with, several geographically-distributed stakeholders require access to the data. These include scientists (located at a research institution), clinicians (in one of possibly many hospitals) and patients (at home, work, hospital, etc..). These stakeholders will have different privileges, e.g. scientists can only view records related to studies they are involved in and only after the data has been made anonymous, protecting the identity of participating patients. This calls for multiple views of the data made available in real time and enforces many privacy constraints on the distributed platform. Our problem also involves heterogeneous data as studies centred around specific outcome measures involve patients receiving care at different institutions, creating a need for a representation unifying the different electronic patient records used at the participating organisations. In a nutshell, our domain presents a suitable example for implementing a multi-agent framework.

2.3 The O-MaSE Methodology

We used the Organization-based Multiagent Systems Engineering (O-MaSE) methodology to design our agents [6]. The methodology guides the process of agent system development via seven steps spanning the analysis and design phases, each producing a model whose output feeds as input for the next. The analysis phase

[1] www.slam.nhs.uk

includes: (1) goal hierarchy model, capturing the goals of the system as a hierarchical structure (2) dynamic goal model, which expands the goal hierarchy by expressing dependencies and precedence relations between the goals in the hierarchy (3) organisation model (4) role model, which defines roles: the different behaviours achieving the design goals and their interactions. The design phase includes (5) agent class diagrams, assigning roles to agents (6) protocol models, defining agent interactions and (7) plan diagrams, defining plans that agents can follow to satisfy organisation goals.

3. OUR SYSTEM

Here we present the design of our system, which is naturally made up of a number of agents located in different organisations, interacting to achieve two distinct aims: 1) to incorporate patient feedback through PROMs into routine clinical work and 2) to enable scientists located at research institutions and wishing to perform targeted naturalistic treatment trial studies to find eligible subjects through our feedback system. We achieve the first aim by placing software agents in the primary care units of registered hospitals for the task of obtaining regular and timely patient feedback. These agents interact with the patients through e-mails and a web interface, providing progress reports, detailed analyses and positive messages to encourage feedback, and subsequently input their feedback into their clinical records. Moreover, we achieve the second aim by enabling clinicians to identify eligible patients in real time and discuss the study, asking for consent to include the patients' anonymous feedback in the study. Throughout our discussions, we highlight aspects of our system which enable it to deal with the data diversity, security and distribution issues presented by the domain.

3.1 Object Modelling

To tackle the diversity of terminologies and languages used in managing electronic patient records, we created an ontology-driven model to serve as a common language understood by all the agents created by the system regardless of their location or host organisation. The model is the result of the joint effort of clinicians, researchers and developers. A detailed specification of our ontology is beyond the scope of this paper. However, we summarise the most essential objects in APPROaCh.

1. **PROM Model:** defines the specifications of a single PROM object and comprises of the following submodels:

   (a) PROM description model: a free-text description of the PROM.

   (b) PROM eligibility model: a set-theoretic model comprising of a number of queryable attribute sets, each defining the domain describing one aspect of the patient's eligibility to the PROM. The sets in the model include the disorders set, the medications set, the age set and the exclusion set, which comprises of elements which, if present in the patient's records, would render the patient ineligible to the PROM, e.g. comorbidities in patients who would otherwise fit all other eligibility criteria.

   (c) PROM evaluation model: contains the attributes measured by the PROM, both mandatory and optional, their corresponding questions which guide the patient as well as the range of acceptable numerical values for each attribute. This model also contains the method for computing PROMs based on the values of the attributes supplied by patients. This model maps to polymorphic functions performing the actual calculation.

   (d) Temporal model: contains temporal constraints associated with the PROM including appropriate intervals between any PROM instances for a patient, the number of PROM instances required (if applicable) as well as appropriate response intervals.

   We define the various PROMs as instances in our ontology. The mapping between PROMs and the eligibility model is many-to-many as multiple PROMs can be found for a disorder, and PROMs with similar eligibility models can match a single patient. We register organisation preferences with respect to the PROMs used when multiple PROMs map to a disorder or a drug. Moreover, our agents are aware of their context-specific preferences and use them to assign PROMs to patients as Section 3.2 will show.

2. **Naturalistic Treatment Trial Model:** describes a single study and is composed of the following submodels:

   (a) Study description model: a free-text description of the study's purpose, length, funding body, participating organisations, target number of participants and authorised personnel.

   (b) PROMs measured by the study: A single study can involve one or more PROMs. The PROMs are described using the model we have illustrated above.

   (c) Study eligibility model: A naturalistic treatment trial studying a certain PROM may have stricter eligibility requirements than the PROM it measures. For instance, a scientist may be interested in a specific age group or a specific stage of a disease. Therefore, the study eligibility model uses an attribute set similar to the PROM attribute set to define constraints superimposed on the corresponding PROM model associated with the study.

3.2 The System Design

As space constraints curtail the level of detail we are able to provide, we summarise our design in the modified agent class diagram shown in Figure 1. The figure illustrates aspects of the system not usually shown in agent class diagrams, including organisations of the different agents, multiplicity constraints enforced on the agent types, as well as the different actors, whom we refer to as stakeholders. Solid lines indicate agent-agent interactions and dashed lines indicate interactions involving at least one stakeholder. The agent classes are also annotated with liveness policies of the agents (the $<< policy >>$ slot) and roles played by each agent (the $<< plays >>>$ keyword), which are detailed in Tables 1 - 9. The description of the goals as well as goal dependencies are described in the text of the following sections. Similarly, we do not provide figures showing the dynamics of agent interactions (e.g. protocol diagrams) and resort to a textual description.

3.2.1 Overall Flow

There are two entry points to our system. The first is initiated when a clinician enters the details of a new patient visiting a primary care clinic at a registered hospital. The *DataManager* agent, having knowledge of all the PROMs and their corresponding patient specifications, informs the *Clinician* agent in real-time if the patient details match an existing PROM. The clinician is then notified through his agent and proceeds to explain the feedback process to the patient, its benefits and procedure. The *DataManager* agent maps the patient records to our uniform representation using the ontology for guidance, and sends the representation to the *UnitManager* agent, which creates a *PatientFollowUp* agent for the patient−PROM pair. This agent locks to that patient, providing timely communications containing guidance to self-care, detailed progress-dependent reports and feedback requests on intervals defined by the PROM's specifications. The *PatientFollowUp* agent also informs clinicians through their agents when patients are not responsive and terminates upon clinician or patient requests.

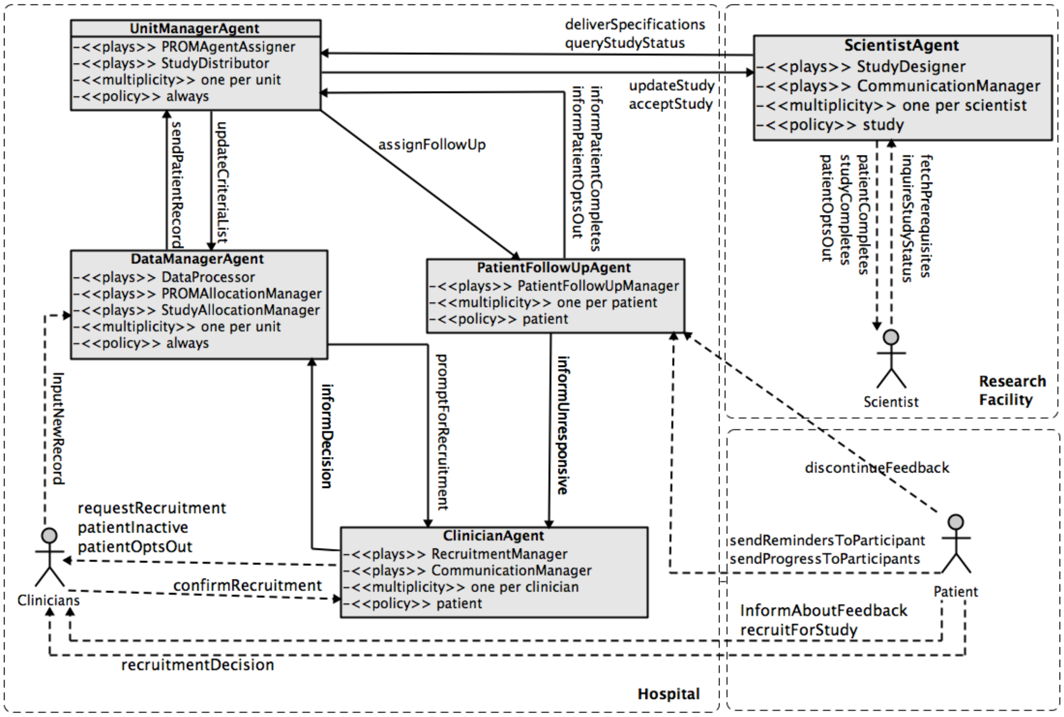

Figure 1: Our Modified Agent Class Diagram

Alternatively, scientists located at a registered research facility can initiate a naturalistic treatment trial study. When the specifications of the study are complete, the *Scientist* agent transfers them to the *UnitManager* agent, which in turn makes them available to the *DataManager* agent. In this scenario, clinicians are prompted in real time if the patient is eligible for a running study. The clinician is responsible for discussing the option of participation with the patient, obtaining consent as well as explaining the difference between patient-reported feedback being a routine process carried out to track the progress of the patient and issue interventions when required, and anonymous research studies requiring patient consent (i.e. NTTs). If the patient consents to participating in the study, the *Clinician* agent informs the *DataManager* agent, which annotates the patient records sent to the UnitManager with the study details, and the feedback process proceeds as usual with the exception that when the *PatientFollowUp* agent receives feedback, it updates the *UnitManager* agent, which subsequently updates the study and sends the anonymised data to the *Scientist* agent. If the patient has not given consent, the *DataManager* agent only attaches the PROM suitable for the patient when sending the records to the *UnitManager* agent and the *Scientist* agent remains uninvolved. We now detail the design of the different agents.

3.2.2 DataManager Agent

As discussed earlier, a crucial issue in APPROaCh is protecting the security of the patient records by prohibiting access by unauthorised personnel. However, as Figure 1 shows, patient records are shared among several distributed agents and accessed by different stakeholders. To ensure the secure transport of the data among the system's components, the *DataManager* agent, assuming the DataProcessor role (Table 1), acts to map the patient records to our internal representation and present views to the different agents based on preassigned privileges.

Role Name	DataProcessor
Description	Maps newly-retrieved patient records to a uniform representation defined by our ontology & creates an anonymised version of the records
Goals	retrieveRecord, anonymiseRecord

Table 1: The Data Processor Role

Our *DataManager* agent is always running in the background, monitoring the electronic patient records system of the hospital where it resides in real-time for incoming records. An incoming record activates the retrieveRecord goal, which maps the patient attributes and relevant treatment details to our ontology, creating an abstract representation understandable by all agents regardless of their host institution and which will be used in lieu of the actual record. The retrieveRecord goal triggers the anonymiseRecord goal, creating an anonymised version of the patient records whereby all references to the patient's identity (e.g. name, address, PatientID) have been removed. The anonymised records are the only versions of the records viewable less privileged agents, e.g. the *Scientist* agent or agents locatd outside the unit. The reader should note that the mapping between the retrieved records and our abstract representation involves knowledge of the structures of the patient records system of the hospital where the patient is receiving treatment and the extraction of terms from unstructured data (as clinicians tend to use free text in describing the case). The details of how the mapping is implemented is discussed in separate literature (see [20]) and is beyond the scope of this work.

Our *DataHander* agent assumes two additional roles. The PROMAllocationManager role (Table 2) awaits the completion of the retrieveRecord goal, triggering the allocatePatient goal, which attempts to map the patient to an appropriate PROM. The matching is made possible by the fact that our *DataManager* receives all the PROM and targeted studies specifications from the *UnitManager* agent (updateCriteriaList in Figure 1). If a match is found, the patient−PROM pair is passed to the *UnitManager* agent of the unit where the agent resides (sendPatientRecord of Figure 1). We achieve the matching between patients and PROMs using a similarity measure briefly described below.

Let $\mathcal{U}_r$ be the set of attributes representing the specifications of a given PROM, and let $\mathcal{U}_p$ be a subset of the patient attributes such that $\mathcal{M} : \mathcal{U}_p \rightarrow \mathcal{U}_r$ is a one-to-one and onto mapping between the attribute sets. The score given to the patient with respect to a PROM $\mathcal{A}(\mathcal{U}_p,\mathcal{U}_r)$ is given by:

$$\mathcal{A}(\mathcal{U}_p,\mathcal{U}_r) = \sum_{u_i \in \mathcal{U}_p} \mathcal{S}(u_i, \mathcal{M}(u_i))$$

Where $\mathcal{S}(u_i, \mathcal{M}(u_i))$ is the degree of similarity between u_i and the corresponding PROM attribute $\mathcal{M}(u_i)$. For categorical attributes (e.g. medication), $\mathcal{S}$ returns 1 if the value of the patient attribute u_i is an element of the set making up the permisible values of the corresponding PROM attribute, and 0 otherwise. Similarly for quantitative attributes (e.g. age), $\mathcal{S}$ returns 1 if the value of u_i falls within the allowed range of values of the corresponding PROM attribute. The PROM allocated to a patient should maximise her $\mathcal{A}$ score given the set of PROM specifications $\mathcal{R}$: -

$$\mathcal{A}(\mathcal{U}_p, \mathcal{R}) = \underset{\mathcal{U}_r \in \mathcal{R}}{\operatorname{argmax}} \mathcal{A}(\mathcal{U}_p,\mathcal{U}_r) \tag{1}$$

To ensure efficiency, we only compute $\mathcal{A}$ using the mandatory attributes of available PROMs for every incoming patient record (these usually include disease, medication and age group). When a patient maps to several PROMs, the preventMultipleAssignments goal is triggered, which uses the PROM's optional attributes and organisation-specific PROM preference ranking to find the match.

Role Name	PROMAllocationManager
Description	Defines the correspondence between PROMs and diagnoses, drugs and other criteria
Goals	allocatePatient, preventMultipleAssignments

Table 2: The PROM Allocation Manager Role

The *DataManager* agent also assumes the StudyAllocationManager role shown in Table 3, which oversees the allocation of patients to naturalistic treatment trial studies they qualify for (detectEligibility, which also maximises $\mathcal{A}$ for patient-study attribute sets). This agent prompts the clinician with the patient's eligibility in real time (promptForRecruitment), who communicates the details to the patient and consent (when provided by the *Clinician* agent) is communicated to the *UnitManager* agent along with the patient record. This way, the *UnitManager* agent updates the corresponding study when the patient completes a PROM.

Role Name	StudyAllocationManager
Description	Oversees the allocation of eligible patients to studies defined by the Study Designer
Goals	detectEligibility, promptForRecruitment, processRecruitment, preventMultipleAssignments

Table 3: The Study Allocation Manager Role

3.2.3 Scientist Agent

The *Scientist* agent is located at a research facility (e.g. university), acting as a mediator between researchers and the rest of the patient feedback system to assist scientists in formally designing prospective naturalistic treatment trial studies. The StudyDesigner role (Table 4) transforms scientists' requests (received via fetchPrerequisites in Figure 1) to a set of executable specifications (achieving the designStudy goal). When designStudy completes, it triggers the agent's checkStudy goal, which involves conversations with the *UnitManager* agents of prospective healthcare units to ensure that the units are capable of running the studies as Section 3.2.4 will show. The *Scientist* agent has to wait until checkStudy returns the responses of all registered *UnitManager* agents to make sure that at least one unit accepts the study. If all units reject the study, then checkStudy returns a failure notice to the scientist and the study will not run in its current state. When checkStudy successfully completes, the specifications are passed to the *UnitManager* agent of units accepting the study through deliverSpecifications.

Role Name	StudyDesigner
Description	Responsible for defining the complete specifications of a team-led study targeting a specific patient cohort.
Goals	designStudy, checkStudy,deliverSpecifications

Table 4: The Study Designer Role

This agent also assumes the *CommunicationManager* role (Table 5), informing authorised scientists of the execution state of the studies they are involved in by achieving patientCompletes, studyCompletes and patientOptsOut. These goals are implemented through eponymous polymorphic procedures, enabling them to use different levels of anonymity depending on the privileges given to the agent (only anonymised updates are seen by the *Scientist* agent).

Upon the scientist request, this agent is able to query the different *UnitManager* agents about the progress of a study through the queryStudyStatus procedure shown in Figure 1.

A single *Scientist* agent is assigned to every authorised scientist machine and is created when the scientist designs her first study. Although this agent's main task is to transform study descriptions into formal specifications understood by our agents, it also plays a role in communicating study updates to the scientist. Therefore, this agent remains alive as long studies it has helped designing are still running and only terminates when no studies are ongoing. This agent is created again when the scientist designs a new study.

Role Name	CommunicationManager
Description	Conveys timely progress messages to scientists and clinicians throughout the trial
Goals	patientCompletes, studyCompletes patientOptsOut

Table 5: The Communication Manager Role

3.2.4 UnitManager Agent

A single *UnitManager* agent is located in every registered primary care unit and always runs in the background assuming two roles. The PROMAgentAssigner role (Table 6) triggers the assignFollowUp goal when a patient− PROM or a patient−study pair is received from the *DataManager* agent, and creates a *PatientFollowUp* agent for the patient, given the PROM or study.

Role Name	PROM Agent Assigner
Description	Oversees the creation of a PatientFollowUp agent for an incoming patient
Goals	assignFollowUp

Table 6: The PROM Agent Assigner Role

In addition, because this agent is the only bridge between the *Scientist* agent and the rest of the system, it assumes the *Gateway* role (Table 7), which involves performing all tasks requiring communication with the *Scientist* agent. When *UnitManager* receives a proposal for a new study from the *Scientist* agent, it triggers the evaluateStudy goal to prevent implementing studies which are too similar in the same care unit. This is essential because if two similar studies are run in the same unit, the one with the more lenient eligibility criteria tends to overshadow the other, rendering the second study with few or no subjects. This agent uses our similarity function $\mathcal{A}$ to find the similarity between a proposed study and those running in its unit and rejects the study if it achieves a similarity above a certain threshold with one of its studies. EvaluateStudy returns a decision to the *Scientist* agent, which is processed as Section 3.2.3 discussed. This role also triggers the updateCriteriaList, which passes the specifications of an approved study to the *DataManager* agent of the unit to monitor incoming records as Section 3.2.2 discussed. Moreover, our *UnitManager* sends updates to the scientists when studies they are involved in recruit new patients, a new PROM is filled out or a patient drops (updateStudy).

Role Name	**Gateway**
Description	performs communication tasks between the scientist agent and the rest of the system
Goals	evaluateStudy,updateStudy, updateCriteriaList

Table 7: The Gateway Role

3.2.5 *The PatientFollowUp Agent*

The *PatientFollowUp* agent is created by the *UnitManager* agent when a new patient maps to a PROM or a running study and locks to the patient to guide the feedback process. This agent remains alive for the duration of the study or until follow-up is terminated for the patient. It follows a finite state machine (FSM) designed to satisfy the goals of the FollowUpManager role (Table 8) and shown in Figure 2. The FSM starts when the the patient completes the registration as guided by the agent (guidePatientRegistration completes), causing the agent to enter the *Awaiting Outcome* state in which no PROM is due yet. The agent remains at this state as long as t_i, a timer which switches on when a PROM is due for the patient, is off and as long as no new PROM is detected for the patient. As soon as t_i switches on, the agent enters the *Outcome Due* state and remains there until a new PROM is recorded. When a new PROM is found, the agent enters the *Send Report* state, where the agent sends a report to the patient, along with informative figures evaluating the patient's progress based on the PROM. The agent then goes back to the *Awaiting Outcome* (if no targeted studies have been assigned to the patient or the number of PROMs required for the study assigned to the patient have not been reached, both indicated by $c = 0$), otherwise goes to *DiscontinueFollowUp*, which also implements the informStudyComplete goal.

Role Name	**FollowUpManager**
Description	Tracks the progress of a single patient
Goals	guidePatientRegistration, remindDue sendReport ,discontinueFollowUp, informUnresponsive, informpatientCompletes, informStudyComplete, informPatientOptsOut

Table 8: The Patient Follow-up Manager Role

When in the *Outcome Due* state, the agent reacts to one signal, $R(t_f) = log_{10}(t_f)$. t_f starts as 1 and is incremented every time the waiting period for reminding the patient elapses (e.g. every week). As one can see, the agent enters the *Remind* state right after the first change in t_f. When in this state, the agent sends a reminder to the patient to fill out the new outcome measure. This state has a natural transition (ϵ) back to the *Outcome Due* state. When $R(t_f)$ exceeds 0.4 (after four reminders ave been sent to the patient), the agent enters the *Inform* state where the *Clinician* agent is notified of the lack of participation from the patient (i.e. the informUnresponsive). At this stage, the clinician can contact the patient and decides whether the trial for this record should continue or could manually kill the *PatientFollowUp* agent for this record.

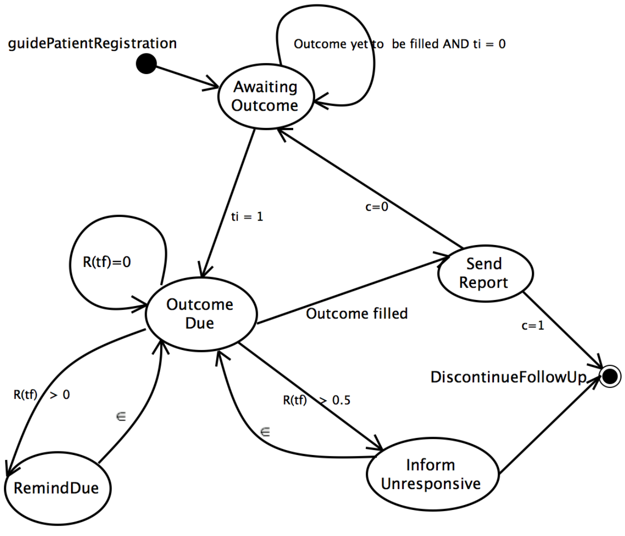

Figure 2: The *PatientFollowUp* agent Finite State Machine

When the *SendReport* state exits with $c = 1$, this indicates that the record has completed the number of time points required for the trial and the FSM goes to the final state and notifications are sent to the *UnitManager* agent. All notifications sent to *UnitManager* trigger its updateStudy goal, communicating the updates to respective *Scientist* agents.

3.2.6 *Clinician Agent*

The *Clinician* agent acts on behalf of clinicians to automate the aspects of the system requiring their involvement and to aid them in achieving timely intervention when automation is not an option. This agent assumes the RecruitmentManager role, which becomes active when the *DataManager* agent detects patient eligibility for a running naturalistic treatment trial study in real time, triggering promptForRecruitment to inform the clinician of the eligibility. When the clinician receives the alert via a pop-up message, she is responsible for discussing the study with the patient and requesting consent. The clinician inputs the patient's decision to the system, and the *Clinician* agent communicates the decision to the *DataManager* agent (through informRecruitmentDecision), which updates the patient records sent to the *UnitManager* agent. This agent also assumes the Communication Manager role, informing the clinician when a patient becomes unresponsive (informInactive) in order for the clinician to contact the patient.

4. PROTOTYPE AND STATUS QUO

In our work, we aimed to separate the design presented in Section 3 from organisation-specific implementation details. This by no means implies a large disparity between the envisioned system

Role Name	**RecruitmentManager**
Description	Oversees recruiting patients for studies
Goals	requestRecruitment, confirmRecruitment informRecruitmentDecision

Table 9: The Recruitment Manager Role

and the deployed infrastructure (with some exceptions as this section will show), as our system was developed based on day-to-day clinical requirements of UK-based healthcare trusts. However, we set the separation so that the system can be applied to different healthcare institutes and healthcare systems.

Here, we present our prototype, which targets a study whose results were the seeds of the collaboration resulting APPROaCh. The study, mentioned in Section 1, examined 8,000 health records of children diagnosed with ADHD and prescribed methylphenidate at the South London and Maudsley National Health Services Foundation Trust (SLaM). Methylphenidate is a highly-efficient drug. Nevertheless, clinicians are ambivalent about the medication as it is essentially a controlled drug (with a chemical structure similar to cocaine). The aim of the study was to investigate the long-term efficiency and side effects of methylphenidate. However, initial investigations found practically no long-term follow-up data for the 8000 patients investigated, with only 1% of the patients having timely PROMs [5]. Subsequent inspections found that the lack of outcome measures is not unique to the records investigated, drug or disorder, and is especially common among patients who leave the hospital to the community resulting no feedback. These are the clinical concerns initiating the collaboration resulting APPROaCh.

Our prototype has been recently deployed across psychiatric clinics as well as research facilities in SLaM to target the assessment of ADHD progress. The PROM of interest in our study is the well-established *strength and difficulties questionnaire* (SDQ) [8] for evaluating children and adolescents' mental health and filled by the patients or their guardians. The SDQ comprises of 25 questions measuring 25 attributes of the child's emotional, conduct and peer relationship health and whose values are added according to the SDQ formula [8] to provide the total difficulties score.

4.1 Prototype Implementation

We implemented APPROaCh using JADE (Java Agent DEvelopment Framework). The system communicates with patients through *IntelliFollow*, a Microsoft Healthvault web application[2] realised via myhealthlocker[3]: a King's-trusted application ensuring the secure transport data between the Healthvault web platform and the hospitals' internal electronic health records systems. Patients are advised to register with Microsoft HealthVault and are directed by email to *IntelliFollow*, where secure login instructions are provided. We used the JadeGateway API[4] to bridge the APPROaCh agents with the web-based *IntelliFollow*. Currently, a large portion of the overall design has been implemented, with some agents not requiring their full capabilities as will be clear shortly. We used Protégé to create our cross-platform ontology. The ontology's PROM object currently has a single instance: SDQ, as our prototype targets this specific PROM. We used JadeJessProtege[5] to transform our ontology into a JADE ontology understandable by all agents. The next section discusses agent-specific deployment.

4.2 Deployment

1. ***DataManager* agent:** So far, this agent has been deployed within the Electronic Patient Journey System (ePJS), the electronic health records system used in most of King's health partners hospitals. Four *DataManager* agents have been placed in four psychiatric clinics located in different sites. Currently, *DataManager* recognises the SDQ eligibility criteria and monitors ePJS for incoming records which match it.

   Although this agent's multiple PROM prevention capabilities have been implemented, *DataManager* does not currently make use of them as we have only deployed the SDQ PROM. As soon as the *Clinician* agent is informed of a qualifying record, it uses the anonymisation pipeline described in [4] to create a research-friendly version of the patient records whereby all references to the patient's identity are removed. The anonymised records are stored in the Clinical Record Interactive Search (CRIS) database [20], a research database containing over 250,000 detailed and anonymised records of mental health patients. CRIS is a trusted research repository established in 2008 through the National Institute of Health Research Biomedical Research Centre (NIHR BRC) at SLaM with access requiring a Disclosure and Barring Service (DBS) check. Only CRIS records are used to update the *Scientist* agent and agents residing in other hospitals with no authorisation to the hospital health records.

2. ***Scientist agent:*** We implemented our *Scientist* agent within a graphical toolkit designed to allow the design and submission of studies by authorised scientists. So far, we deployed a single instance of our agent in the Biomedical Research Council Nucleus at King's College London[6], a research insitution founded to facilitate research activities between SLaM and King's College London. The *Scientist* agent can only design trials involving authorised variations of the SDQ PROM. We impelemented checkStudy using contract Net with the *Scientist* agent being the initiator and the four *UnitManager* agents being the responders.

3. ***UnitManager agent*** Four *UnitManager* agents live in our participating clinics, with one in each clinic. These agents are triggered by messages received from the four *DataManager* agents of their respective clinics as well as our single *Scientist* agent.

4. ***PatientFollowUp agent:*** This agent is created on a per-patient basis. Projections from examining CRIS records showed that on average, 900 ADHD patients are seen annually, reflecting the anticipated number of agents created of this type. *PatientFollowUp* agents communicate with patients via regular e-mails with different purposes ranging from induction, login and account creation, provision of online resources suitable for the case, graphical views of personal progress as well as timely reminders for entering new SDQ values. Patients are directed to the *IntelliFollow* web tool, where they can view their progress and enter new SDQ values.

5. ***Clinician agent:*** We installed *Clinician* agents on 19 machines located across the four clinics. The agent communicates with clinicians by providing a pop-up if the patient is eligible for a study. Since study selection for the patient is handled by the *DataManager* agent, a single pop-up is guaranteed to appear on the clinician's machine (not applicable to our single-PROM prototype). The pop-up message has a box which can be ticked by the clinician if the patient consents to the study, and no further

[2] https://developer.healthvault.com/

[3] https://www.myhealthlockerlondon.nhs.uk/PrivacyPolicy.aspx

[4] http://jade.tilab.com/doc/api/jade/wrapper/gateway/JadeGateway.html

[5] http://sourceforge.net/projects/jadejessprotege/

[6] http://www.slam.nhs.uk/about/core-facilities/brc-nucleus

interaction is required unless the patient becomes unresponsive. When this happens, a pop-up appears on the clinician's screen to which the clinician can respond to by contacting the patient. The clinician can also mute the pop-up for a user-specified interval or indefinitely, the latter not being recommended.

4.3 Evaluation

In this work, our evaluation focuses on insuring that the different aspects of our system work as envisioned. Due to space constraints, we present two experiments, one evaluating the efficiency in recognising eligible patient records and another testing our similarity measure. We used a test (non-live) version of ePJS.

1. **Response Time:** In order to test the efficiency of patient-PROM and patient-study matching in real-time, we studied the relationship between the complexity of the PROM eligibility criteria and the time taken to complete the matching process. We experimented by varying the SDQ eligibility criteria to create 16 cases representing SDQ criteria containing 4-19 attributes. For each case, we tested our agents by populating our non-live ePJS with 20 patient records in real time and recorded the time our *ClinicianAgent* reacted with a pop-up message. Our test results show that the maximum response time was 8.38 seconds (occurred in a test set with 19 attributes), which was assessed by the clinicians in our team to be adequate with respect to consultation time. The average response time ranged from 1.2 seconds (for test sets having 4 attributes) to 7.79 seconds (for test sets with the maximum number of attributes).

2. **Study Allocation:** We tested the *UnitManager*'s effectiveness in rejecting proposed studies which are two similar to ones it is currently running. We initiated 20 fictitious NTT studies, using variations of the SDQ and whose eligibility models vary across age group, medications and gender. We designed the studies so that they comprise of pairs, with each study pair being made of two studies overlapping in one of the following two ways:

   (a) proper-subset similarity: studies whose mandatory attribute values are either subsets of supersets of other studies, e.g. a study with age group 5-7 and another with age group 4-14. In this case, our four *UnitManager* agents successfully reject proposed studies whose mandatory attribute values are subsets or supersets of studies running in their units.

   (b) intersection overlap: studies sharing similar attributes but also contain conflicting sets of attributes. Our *UnitManager* agents successfully reject studies whose mandatory attributes achieve similarity of 20% or greater with studies they are currently running. The rejection threshold was chosen based on discussions with clinical workers at SLaM.

5. CONCLUSIONS, CHALLENGES AND FUTURE WORK

We presented the design and prototype deployment of a system countering a real clinical need. The lack of measurable treatment outcome measures in clinical settings has been shown to hinder long-term investigations and patient care, scientific inquiry as well as objective quality control and auditing inquests. The initial analysis uncovered real issues which we summarise as: 1) security issues, springing from the confidential nature of patient records, making any system involving their manipulation and transport challenging to realise. This is complicated further when stakeholders not directly involved in the treatment have real interest in the data (scientists), and when patients are granted the right to add to their clinical records from outside the hospitals where they are seeking treatment 2) distribution issues, embodied by the fact that a single patient can visit one or more hospitals for treatment, making his records distributed across different units or organisations 3) data diversity issues, arising from the diversity of medical terminology and electronic health records systems used in the different organisations. These challenges made it obvious that the key to a successful design is the avoidance of a traditional healthcare system. We therefore designed and implemented APPROaCh, a multi-agent system targeting the above issues. We handled the security issue by implementing a pipeline which anonymises the patient records and granting varying privileges to the different agent types depending on their role, with only agents representing personnel involved in the patient's treatment being allowed to access the actual identifiable patient data. Other agents (e.g. *Scientist* agents) can only view the anonymised records. We dealt with the distribution and data diversity issues by equipping our system with an ontology unifying the organisational variations of data and systems to create a representation understood by all agents involved, regardless of their host institution. This representation also allows patients to interact with our web tool regardless of the hospital(s) where they are seeking treatment. Our webtool is built to understand our unified representation, which is then used to map the data entered by patients to the electronic health records systems of the respective hospitals.

We implemented a prototype targeting ADHD patients and have recently deployed it across mental health clinics in SLaM, London, U.K.; it has so far been well-received. The work presented in this paper aims at examining the technical hurdles we faced in implementing such system in a domain where privacy, security, data diversity and guidelines adherence are major bottlenecks. However, the long-term effectiveness of the automated system can only be established once long-term data is available.

In addition to validation through data collection, there are several issues which our current deployment does not validate. Since our prototype targets a single PROM, the *DataManager*'s multiple-PROM prevention functionality is not used. The same applies to situations where conflicts arising from organisational preferences with respect to PROMs require resolution for consolidation into prospective studies. The incorporation of future PROMs in our system is an ongoing long-term process and requires authorisations from clinical, technical and governance bodies. The approval process is partially dependent on the success of the current prototype in increasing patient involvement in outcome measure collection as well as anonymised clinical trials.

Our experience in working with healthcare systems has come with many legal and security challenges. In deploying our prototype, we worked in an environment where the following is available: 1) the CRIS research database, with strict access provided to authorised scientists. This resource has eased the automation of the anonymisation pipeline of the health records and provided an existing trusted reserve for new data 2) an already-approved Microsoft Healthvault web-based API for reading from and writing to the electronic health records systems of the units where our system is deployed. Obtaining the required authorisation and completing the legalities to use these two resources have been not only time-consuming, but also a deciding factor in implementing our prototype. Moreover, since these resources are limited to King's College Health Partners, any expansion to further organisations will involve tackling the bureaucracies and legal and organisational restrictions associated with setting up similar resources (or finding alternatives).

Acknowledgements

The authors would like to acknowledge the supports of the NIHR Biomedical Research Centre for Mental Health, the Biomedical Research Unit for Dementia at the South London, the Maudsley NHS Foundation Trust and King's College London, and a joint infrastructure grant from Guys and St Thomas Charity and the Maudsley Charity, London, United Kingdom.

REFERENCES

[1] M. Bagues, J. Bermudez, A. Burgos, A. Goni, A. Illarramendi, J. Rodriguez, and A. Tablado. An innovative system that runs on a PDA for a continuous monitoring of people. In *IEEE Symposium on Computer-Based Medical Systems*, pages 151–156, 2006.

[2] L. Braun, F. Wiesman, H. Van Den Herik, and A. Hasman. Agent support in medical information retrieval. In *Working Notes of the IJCAI-05 Workshop on Agents Applied in Health Care*, pages 16–25, 2005.

[3] M. Calisti, P. Funk, S. Biellman, and T. Bugnon. A multi-agent system for organ transplant management. In *Applications of Software Agent Technology in the Health Care Domain*. 2003.

[4] A. Fernandez, D. Cloete, M. Broadbent, R. Hayes, C. Change, R. Jackson, A. Roberts, J. Tsang, M. Soncul, J. Liebscher, R. Stewart, and F. Callard. Development and evaluation of a de-identification procedure for a case register sourced from mental health electronic records. *BMC Medical Informatics and Decision Making*, 13:71–85, 2013.

[5] L. Fernandez, E. Simonoff, J. McGrough, J. Halperin, E. Arnold, and A. Stringaris. Treatment of Children With Attention-Deficit/Hyperactivity Disorder (ADHD) and Irritability: Results From the Multimodal Treatment Study of Children With ADHD (MTA). *Journal of the American Academy of Child and Adolescent Psychiatry*, page In Press, 2014.

[6] J. Garcia-Ojeda, S. DeLoach, R. Walamitien, and J. Valenzuela. O-mase: A customizable approach to developing multiagent development processes. In *The 8th International Workshop on Agent Oriented Software Engineering*, 2007.

[7] H. Gonzalez-Velez, M. Mier, M. Julia-Sape, A. TN, J. Garcia-Gomez, M. Robles, P. Lewis, S. Dasmahapatra, D. Dupplaw, A. Peet, C. Arus, B. Celda, S. Huffe, and M. Lluch-Ariet. Healthagents: distributed multi-agent brain tumor diagnosis and prognosis. *Applied Intelligence*, 30(3):191–202, 2009.

[8] R. Goodman. The strengths and difficulties questionnaire: a research note. *Journal of Child Psychology and Psychiatry*, 38(5):581–586, 1997.

[9] H. Helmchen. Ethical issues in naturalistic versus controlled trials. 13(2):173–182, 2011.

[10] N. Hjollund, L. Larsen, K. Biering, S. Johnsen, E. Riiskjar, and L. Schougaard. Use of patient-reported outcome (pro) measures at group and patient levels: Experiences from the generic integrated pro system, westchronic. *Interactive Journal of Medical Research*, 3(1):e5, 2014.

[11] Z. Ibrahim, R. Mallah, and R. Dobson. Towards using rule-based multi-agent system for the early detection of adverse drug reactions. In *The International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2014.

[12] D. Isern and A. Moreno. Computer-based execution of clinical guidelines: a review. *International Journal of Medical Informatics*, 77(12):787–808, 2008.

[13] D. Isren, D. Sanchez, and A. Moreno. Agents applied in health care: A review. *International Journal of Medical Informatics*, 79:145–166, 2010.

[14] Y. Ji, H. Ying, M. Farber, J. Yen, P. Dews, R. Miller, and M. Massanari. A Distributed, Collaborative Intelligent Agent System Approach for Proactive Postmarketing Drug Sasfety Surveillance. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):826–837, 2009.

[15] V. Koutkias, I. Chouvarda, and N. Maglaveras. A multiagent system enhancing home-care health services for chronic disease management. *IEEE Transactions on Information Technology in Biomedicine*, 9(4):528–537, 2005.

[16] S. Mahmoud, G. Tyson, S. Miles, A. Taweel, T. Van Staa, M. Luck, and B. Delaney. Multi-agent system for recruiting patients for clinical trials. In *International conference on Autonomous Agents and Multi-Agent Systems*, pages 981–988, 2014.

[17] B. Orgun and J. Vu. HL7 ontology and mobile agents for interoperability in heterogeneous medical information systems. *Computers in Biology and Medicine*, 36:817–836, 2006.

[18] G. Salvi, M. Leese, and M. Slade. Routine use of mental health outcome assessments: choosing the measure. *The British Journal of Psychiatry*, 186:146–152, 2005.

[19] S. Singh, B. Ikhwan, F. Haron, and C. Yong. Architecture of agent-based healthcare intelligent assistant on grid environment. In *International Conference on Parallel and Distributed Computing:Applications and Technologies*, pages 58–61, 2005.

[20] R. Stewart, M. Soremekun, G. Perera, M. Broadbent, F. Callard, M. Denis, M. Hotopf, G. Thornicroft, and S. Lovestone. The south london and maudsley nhs foundation trust biomedical research centre (SLAM BRC) case register: development and descriptive data. *BMC Psychiatry*, 9(51), 2009.

[21] S. Treweek, E. Pearson, N. Smith, R. Neville, P. Sargeant, and B. Boswell. Desktop software to identify patients eligible for recruitment into a clinical trial: using SARMA to recruit to the road feasibility trial. *Journal of Informatics in Primary Care*, 18:51–58, 2010.

[22] P. Velentgas, N. Dreyer, P. Nourjah, S. Smith, and M. Torchia. Developing a protocol for observational comparative effectiveness research: A user's guide. Technical report, 2013.

[23] A. Wu, R. Jensen, C. Salzberg, and C. Snyder. Advances in the use of patient reported outcome measures in electronic health records including case studies. Technical report, 2013.