An Analysis of Agent-Oriented Engineering of e-Health Systems

Emilia Garcia[1]*, Gareth Tyson[2], Simon Miles[2], Michael Luck[2], Adel Taweel[2], Tjeerd Van Staa[3], and Brendan Delaney[2]

[1] Universitat Politecnica de Valencia, Spain,
[2] King's College London, UK,
[3] General Pratice Research Database, UK
mgarcia@disc.upv.es,gareth.tyson@kcl.ac.uk
simon.miles@kcl.ac.uk,michael.luck,adel.taweel@kcl.ac.uk
tjeerd.vanstaa@gprd.com,brendan.delaney@kcl.ac.uk

Abstract. Online e-health systems are being proposed and developed at an ever increasing rate. However, the progress relies on the interoperability of local healthcare software, and is often hampered by ad hoc methods leading to closed systems with a multitude of protocols, terminologies, and design approaches. AOSE seems intuitively a good approach to developing more open systems. While agent-based e-health systems have been developed, the general hypothesis of the suitability of AOSE has not been evaluated. In this paper, we test that hypothesis, including a case study of applying a regulated agent methodology to a particular real-world e-health system, and present an analysis of the strengths and weaknesses of AOSE for e-health.

Keywords: Health systems, Normative environments, Organizational Agent Architectures, Contracts, System of Systems

1 Introduction

Large-scale and flexible systems are increasingly needed to fulfil the emerging requirements of complex domains. One typical example is *healthcare*, which is rapidly becoming more and more dependent on large-scale integrated software systems. On the one hand, these systems offer new and innovative ways to improve patient care; however, on the other, they also introduce complications and risks that were never envisaged in the early days of healthcare computerisation. Clearly, these complications affect the development of related software. A particular challenge is that the healthcare domain is separated into many disparate organisations that often fall under different spheres of control. As a result, it is common for systems to be constructed out of many divergent sub-systems; this is termed a *systems of systems* (SoS). In this context, interactions can often take

* This work is partially supported by the TIN2009-13839-C03-01, CSD2007-00022, COST Action IC0801 and the FPU grant AP2007-01276

place between components that are managed by parties with conflicting goals, different policies, incompatible data representations, and so on. Not surprisingly, this can lead to serious challenges when integrating these different systems in a trustworthy, consistent manner, leading to the emergence of strict regulatory controls to manage not only the internal behaviour of organisations, but also the interactions that may take place between multiple organisations.

While multi-agent technology has emerged over the last decade as a new software engineering paradigm for building complex, adaptive systems in distributed, heterogeneous environments, it is still not mainstream in its domain application. Nevertheless, it can be observed that many properties of the healthcare domain fit well with several concepts that arise in the area of multi-agent systems, such as organisational autonomy, inherent regulatory frameworks, and so on. It thus seems appropriate that introducing such principles to the development of healthcare systems could offer many benefits. Indeed, there have been several agent-based e-health systems developed over a period of many years, but these they rarely employ explicit agent-oriented software engineering (AOSE) methodologies and, as such, do not directly evaluate the suitability of AOSE to this domain in particular.

Addressing this omission, in this paper we investigate the suitability of using AOSE, and the common underlying concepts used in AOSE design and development, for the creation of e-health systems. We wish to answer the following question: *To what extent is AOSE an approach that is appropriate to the development of e-health systems?* To represent AOSE in testing this hypothesis, we take a specific methodology, ROMAS (Regulated Open Multi-agent Systems), whose main concepts are introduced in Section 2. ROMAS is an AOSE methodology that guides developers all the way from the requirements analysis phase to the actual implementation, taking into account the notions of agents, organizations, services and contracts. As ROMAS shares many of the same fundamental concepts with other existing AOSE methodologies, it can be seen as an adequate representative for testing the hypothesis. In this paper we apply the ROMAS methodology to a particular (real) e-health system: ePCRN-IDEA [13, 12], with the resulting system design being presented in Section 3. This system allows us to exemplify the features of healthcare systems, so as to evaluate the suitability of AOSE in addressing them. Ultimately, in Section 4, we identify a number of strengths and weaknesses of AOSE for such systems, as well as suggesting improvements to better support the needs of the domain. Finally, we conclude the paper in Section 5.

2 Regulated Open Multi-agent Systems

In this section we introduce how ROMAS integrates the common AOSE concepts of agents, roles, organizations, norms and contracts. A complete description of ROMAS can be found in [9], so we address only the key aspects here. In ROMAS, *agents*, *roles* and *organizations* are defined through a formal social structure based on a service-oriented open MAS architecture, whose main features are

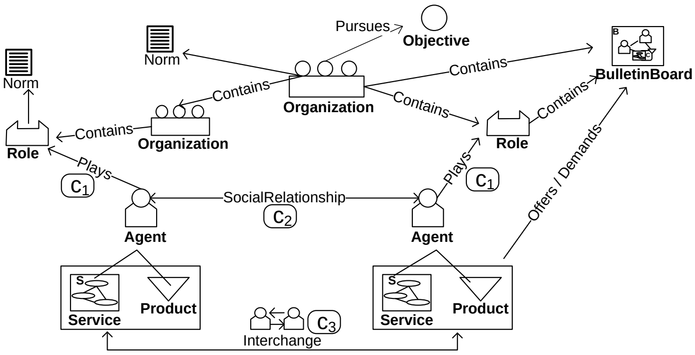


Fig. 1. Overview of ROMAS architecture

summarized in Figure 1. Here, organizations represent a set of individuals and institutions that need to coordinate resources and services across institutional boundaries. In this context, agents represent individual parties who take on roles in the system, within a given organization (e.g. a company), they can both offer and consume services as part of the roles they play. Beyond this, virtual organizations can also be built to coordinate resources and services across institutional boundaries. Importantly, each of these concepts must be strictly defined, alongside their interrelations. Organizations are conceived as an effective mechanism for imposing not only structural restrictions on their relationships, but also normative restrictions on their behaviour. These restrictions are formalized in ROMAS by means of norms and contracts.

Norms in ROMAS are specified using the model described in [3], which defines norms that control agent behaviour, the formation of groups of agents, the global goals pursued by these groups and the relationships between entities and their environment. Specifically, it allows norms to be defined: (i) at different social levels (e.g. interaction and institutional levels); (ii) with different norm types (e.g. constitutive, regulative and procedural); (iii) in a structured manner; and (iv) dynamically, including later derogation. Figure 1 shows two types of norms: (i) those that are are associated with each organization; and (ii) those that are associated with each role. Clearly, the former must be complied with by any organization member, while the latter must be complied with by all agents playing that role.

Finally, ROMAS also allows interactions to be fomalized by means of *contracts*. These are necessary when working in an open regulated system, to be able to specify the expected behaviour of others without compromising their specific implementation. ROMAS involves two types of contracts: *social contracts* and *contractual agreements*. Social contracts can be defined as a statement of intent that regulates behaviour among organizations and individuals. As shown in Figure 1, social contracts are used to formalize relationships: (i) between an agent playing a role and its host organization (as indicated by the contract labelled c_1); and (ii) between two agents providing and consuming services (as indicated by c_2). Social order, thus, emerges from the negotiation of contracts about the rights and duties of participants, rather than being given in advance. In contrast, contractual agreements represent the commitments between several entities in order to formalize an interchange of services or products (c_3).

3 Application Case-Study: ePCRN-IDEA System

In this section we present a system architecture for recruiting patients for clinical trials in real-time. First, Sections 3.1 and 3.2 introduces the domain context and its challenges. Second, Section 3.3 describes the system design using ROMAS. Finally, Section 3.4 analyzes how ROMAS deals with the main challenges of the system.

3.1 ePCRN-IDEA Overview

Clinical trials are experiments by which the efficacy of medical treatments are explored. They involve recruiting patients with specific characteristics to undergo new treatments, so that the effectiveness and safety of those treatments can be tested. However, a key challenge in this is recruiting sufficient patients to ensure the results are meaningful. This has long been a difficult problem as the requirements for participation are often very strict, making it difficult to locate eligible patients. ePCRN-IDEA [12] is a new system under deployment in the UK healthcare system that notifies practitioners in real-time whenever an eligible patient is in consultation. When a patient visits a clinic, ePCRN-IDEA compares their details against a database of trials; if the patient is eligible for one or more, the practitioner is prompted to try to immediately recruit the patient if they are interested.

3.2 Challenges in ePCRN-IDEA's Development

Development of the ePCRN-IDEA system [13] has identified a number of core challenges, which are typical of similar systems in the health domain. In this light, this section briefly covers the most important of these identified challenges to gain a better understanding of how AOSE might be able to benefit the development process of such systems.

Integration of Independent Systems. In order to recruit eligible patients, it is necessary for researchers, practitioners, patients, databases and clinics to interact. This means that several independent institutions, which are completely autonomous and have their own independent goals, must cooperate to achieve a common objective. However, the integration of multiple heterogeneous and autonomous systems can be a complicated and resource-consuming task. Some of the issues that must be solved are: (i) *Distributed Data* – the required data is spread widely across all organizations, frequently using different schemas; (ii) *Technical Interoperability* – different organizations often use different (potentially incompatible) technologies; (iii) *Process Interoperability* – different organizations often employ divergent (potentially incompatible) processes to achieve

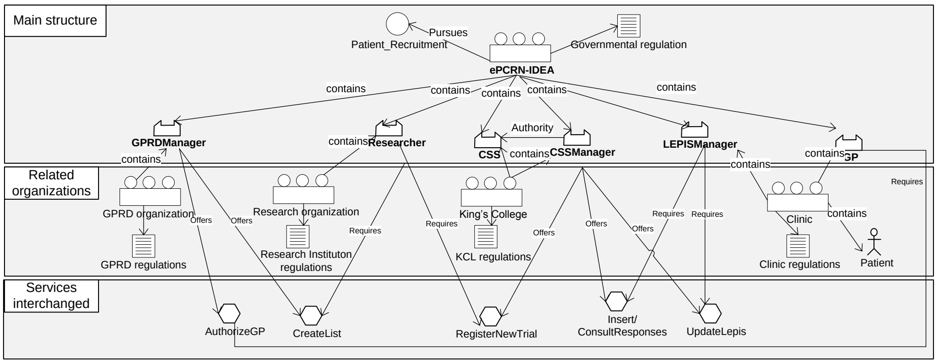


Fig. 2. ePCRN-IDEA organizational structure.

their goals; (iv) *Semantic Interoperability* – different organizations often utilise different vocabularies and coding schemes, making it difficult to understand the data of others; (v) *Trustworthiness* – little trust exists between different organizations, particularly those with conflicting goals and interests. In consequence, healthcare systems that consist of multiple organizations must take all these aspects into account to ensure successful operation.

Regulation of Independent Systems. Healthcare systems must fulfil strict governmental regulations concerning the privacy and security of personal patient data. Moreover, each research institute and clinic has its own regulations, specific goals, priorities and restrictions to regulate the behaviour of each of its members. Healthcare systems must therefore often take into account several regulation environments.

System Evolution. Medical institutions are constantly adapting their systems to reflect new legislation, software and medical techniques. As these autonomous organizations often operate with a range of aims and priorities, it is possible that changes may take place without necessarily propagating to all other parts of the system. In this respect, a change within one sub-system could result in violations of responsibilities in another sub-system (e.g. by changing data formats). Healthcare systems that consist of multiple organizations must therefore ensure some formal procedure by which all parties understand and adhere to their responsibilties. To enable practical deployment, institutions must also be contractually obliged to adhere to a standard interaction mechanism and data format, although their internal process or storage technology changes.

3.3 Developing ePCRN-IDEA Recruitment System with ROMAS

In this section, we present the ePCRN-IDEA system design following the ROMAS methodology. Figure 2 shows the main structure of ePCRN-IDEA in terms of the key concepts of organizations, roles, norms and contracts, detailed below.

Organizations and Processes. Several organizations are involved in the key processes performed in ePCRN-IDEA, as follows. When a research body wishes to create a new clinical trial, they can inject it through a service called the Central Control Service (CCS), which is hosted at *King's College London* (KCL). The CCS stores trials within a large database in a pre-defined format that all researchers must adhere to. Associated with each trial is a list of potentially eligible patients; these lists are generated by the *General Practice Research Database* (GPRD), which operates a large data warehouse containing over 12 million up-to-date patient records in the UK. Following this, the trials and their eligibility lists are distributed to software agents (called LEPIS agents) that operate on clinicians' PCs at each participating *clinic*. LEPIS agents then listen to the interactions between the practitioner and their local Electronic Health Record (EHR) database, which is used to store information about patients (e.g. diagnoses, treatments, demographic data etc.). During consultations, LEPIS agents compare the patient information against the eligibility lists of all known trials. If a patient is found to be eligible for a trial, the practitioner is notified, and if the patient is interested, the system loads a Random Clinical Trial (RCT) website provided by the *research body* responsible for the trial, allowing the patient's recruitment to be completed. Consequently, the following organizations are involved: KCL, GPRD, the clinics and the research bodies.

Roles. The system is composed of six different roles presented below.

The *GPRD Manager Role* is responsible for updating and controlling access to the GPRD database. It offers a service to pre-compute lists of eligible patients for individual trials based on complex search criteria (*CreateList* service). The role must also offer a service to decide when a GP is authorized to perform recruitment for each trial (*AuthorizeGP* service). The agent that plays the GPRD Manager role must also play a role in the governmental body (represented as the *GPRD organization*), so it must follow the special governmental legislation related to the management of this kind of data.

The *Researcher Role* is responsible for defining the specific features of each trial under its jurisdiction. Researchers are also responsible for inserting these trials into the CCS database by means of the service offered by the CCS role (described below). They are not allowed to directly contact patients unless they have agreed to participate in a clinical trial under their supervision. For obvious reasons, each researcher should be part of a specific research institution and follow its specific normative restrictions.

The *CCS Role* is a software application responsible for controlling the CCS database, which stores data about active clinical trials. It offers three services to the other members of the system: (i) a *Register New Trial* service that allows researchers to inject new clinical trials in the database; whenever a Researcher tries to inject a new trial into the CSS database, the CSS role must verify that this trial follows the specified standards and regulations; (ii) an *Update LEPIS Database* service that allows the clinic's local database to update its information about the active clinical trials; and (iii) an *Insert/Consult Patients Response* service that allows the response of each patient to be registered (whether they agree or refuse to participate in a trial). The current implementation of the CCS role is performed by an agent that is part of the KCL organization. Clearly, this agent must comply with established norms concerning replication of information, privacy and programmed machines maintenance.

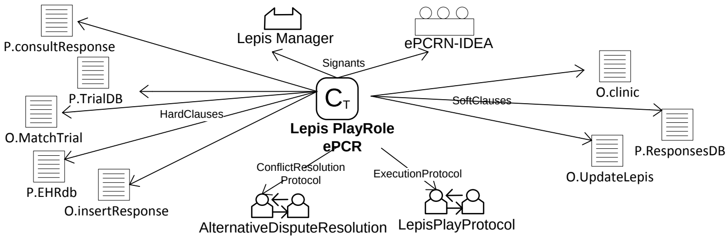
The *CCS Manager Role* is responsible for controlling the information in the CCS (i.e. it has control over the CCS Role). Due to the specific requirements described by the domain expert, there must be a human responsible for this. This role must be played by a member of KCL, who must therefore comply with the restrictions and rules that KCL establishes.

The *LEPIS Manager Role* is played by a software application that resides at a clinic and investigates the eligibility of any present patient. There is thus a LEPIS agent playing this role for each clinic participating in the recruitment system. LEPIS agents use the CCS service to acquire information about the clinical trials related to the type of patients that in which its clinic is specialized. LEPIS agents also provide the GP with a simple interface to notify them of a patient's eligibility, as well as the option to launch the RCT website if the patient is interested.

The *GP Role* represents a practitioner working in a clinic. If a GP wants to recruit patients for trials, they must be previously authorized by the GPRD Manager. This authorization involves the acceptance of some norms related to privacy, and specific restrictions described for each clinical trial. Clearly, each GP must also comply with the rules of their own clinic. Finally, patients are considered external entities for the ePCRN-IDEA system because their interaction with the system is always executed through their GP.

Norms and Contracts. The *Governmental regulations* related to the privacy of patient data and clinical trials are described at a system-wide level; i.e., every agent playing a role inside ePCRN-IDEA should comply with them. At the same time, each institution and clinic defines its own regulations, so the entities of the system should follow the general governmental regulations and the restrictions established by the institution to which they pertain. For instance, each LEPIS agent should follow both global and clinic-specific regulations. The rights and duties that any specific agent implementation must fulfil to play a role in ePCRN-IDEA are formalized by means of a *Social Contract*. Even though contracts are dynamic entities that cannot be completely defined at the design stage, designers can specify the predefined restrictions that all final contracts of a specific type should follow. These restrictions are defined in a *Contract Template*, where *Hard Clauses* indicates mandatory clauses that any contract of this type must contain and *Soft Clauses* indicate more flexible recommendations. Clearly, due to space constraints, a comprehensive set of norms and contracts in ePCRN-IDEA cannot be listed; thus, we briefly cover a small number of examples.

Figure 3 describes the *LEPIS PlayRole* contract template. It specifies that any agent playing the LEPIS Manager role must detect changes in the EHR database and after that it must check the suitability of this patient for any trials (*Norm O.MatchTrial*). The contract template also recommends that the final



NORM ID	NORM DESCRIPTION (Deontic,Target,Activation,Expiration,Action,Sanction,Reward)
O.MatchTrial	(OBLIGED, Lepis, Event(changesEHR), - , Match_Trial_Historical,-,-)
O.UpdateLepis	(OBLIGED, Lepis,DAILY, - , Request(UpdateLepis service),-,-)
P.EHRdb	(PERMITTED, Lepis, -,-, Read(EHR database),-,-)
P.TrialDB	(PERMITTED, Lepis, -,-, Read(Lepis trial database),-,-)
P.ResponsesDB	(PERMITTED, Lepis, -,-, Write(Lepis patient responses database),-,-)
P.consultResponse	(PERMITTED, Lepis, -,-, Request(ConsultPatientAnswer service),-,-)
O.insertResponse	(OBLIGED, Lepis, GPInsertResponse, - ,Request(InsertPatientResponse service),-,-)
O.clinic	(OBLIGED, Lepis, -,-, Pertain(Clinic),-,-)

Fig. 3. Phase 2: Lepis PlayRole social contract template

contract includes a norm specifying that the local LEPIS database must be updated with new clinical trials every day (*Norm O.UpdateLepis*). This clause is merely a recommendation so that at runtime, LEPIS agents are able to negotiate with the ePCRN-IDEA organization exactly how often they should update their local database. The remaining clauses relate to the use of the local LEPIS databases and the service dependencies that LEPIS requires. In this way, each clinic can implement its own LEPIS agent (if it complies with the required contracts and norms), allowing each clinic to adapt the behaviour of LEPIS in line with its own priorities. For example, a clinic could decide that its LEPIS agent should not increase patient queues; e.g. GPs should not be notified during busy periods. Similarly, each entity that plays any role in ePCRN-IDEA can be adapted to the different requirements and restrictions of its own institution. Each institution would thus maintain its own technology, with different implementations of each role interacting independently of the technological differences.

3.4 Benefits of multi-agent regulation for ePCRN-IDEA

In this section, we revisit the design challenges listed in Section 3.2 to see how effective ROMAS has been.

Integration of Independent Systems. ROMAS offers an effective design platform for modelling and integrating the different ePCRN-IDEA systems by enforcing a high level of abstraction, using many real-world concepts (e.g. organizations). First, this helps domain experts, who are typically not familiar with the relevant technology, to gain a better understanding of the system. Beyond this, it also provides well defined boundaries between different agents and organizations, allowing individual objectives and regulations to be specified, as well as maintaining the privacy of each institution's data and processes. Importantly, technical and semantic interoperability challenges are also addressed by means of standardized web service interfaces.

Regulation of Independent Systems. The regulatory needs of ePCRN-IDEA fit well into the ROMAS principles. Specifically, it allows different normative environments for each clinic and research institution to be explicitly described and combined with global governmental norms. This allows the behaviour of the different entities to be formally constrained — an extremely important feature in the medical domain. Furthermore, different vendors and technologies can be used to implement the agents that play each role. For instance, each clinic could specify and implement its own LEPIS agents according to its aims, restrictions and priorities, while maintaining the stability of the system through global governmental regulations. This is particularly important when potentially deploying agents across multiple research institutions and clinics from different countries.

System Evolution. ROMAS offers an effective paradigm for assisting in system evolution in ePCRN-IDEA. Through norm and contract regulation, each sub-system can evolve while ensuring that it does not compromise its responsibilities to other parties. Common examples include adaptation to new internal regulations or to the use of a new software technology. Moreover, global system evolution can also take place by publishing new contracts and norms, thereby forcing sub-systems to adapt.

4 Discussion

In the previous sections, we have considered how an existing e-health system could have been developed using an existing agent methodology, and the benefits of doing so. We now consider the more general hypothesis presented in Section 1: that AOSE is highly appropriate for the development of e-health systems.

4.1 Beneficial features of AOSE

AOSE methodologies commonly include analysis and design based on a few key ideas: *agents* as autonomous, pro-active, flexible and social entities; *interactions* of a flexible and well-defined nature between those agents; and *organizations* in which agents operate, modelled either implicitly or explicitly [5, 1, 10]. The functionality that agents enact in such designs is sometimes modelled as *services* [8]. Other features present in some methodologies, including ROMAS, are the assumptions of *openness* in the system, and of *regulation* to be followed by agents (e.g. norms, responsibilities, rights, contracts, etc. [2, 4, 6, 11, 14]). Through the lessons learned during the development of ePCRN-IDEA, we now present some features of AOSE that indicate its suitability for general e-health applications.

Assumption of autonomy. A critical aspect of e-health systems is that they are comprised of sub-systems that have their own regulations, privacy issues, localised authority, localised flexibility, and so on. For instance, in ePCRN-IDEA, different policies are applied in different clinics and regions in the UK. In this context, it is clear that e-health systems must also take into account this diversity. This stems from factors such as the need to preserve patient confidentiality, the commercial sensitivity of drug development, and from government involvement and regulation at a local level. The autonomy of agents and organizations assumed at the analysis stages in AOSE means that this is a particularly well-suited approach.

Allowance for openness. There are millions of independent sites involved in healthcare in various capacities worldwide (with varying levels of system computerization). A common feature of large-scale e-health systems, such as ePCRN-IDEA, is the expectation that more sites will join the system as they develop the technical capability to do so (e.g. new clinics, research institutes etc.). This means that methodologies with an assumption of an open system are well tailored to e-health. In practice, openness is enabled by a design specifying exactly how a new party must behave in order to join the system, such as through contracts (as in ROMAS) or roles, as well as lower level concerns such as interfaces and interaction protocols.

Explicit norms. Due to the confidentiality issues mentioned above, healthcare is highly regulated at all levels, and these regulations must be considered as a primary influence on any e-health system. Regulations apply both to individual clinics and researchers, and across the whole system due to national or international laws. For instance, in ePCRN-IDEA each clinic and practitioner must be individually authorised to recruit for each trial. Clearly, e-health also includes norms of good practice that are not strict regulations but with which it is preferable to comply. The advantage of a norm-based design approach is that there is a ready way for developers to specify these regulations explicitly in the development process, such that they become part of the design. Implementing the system in a norm-aware platform can ensure their fulfilment, even if the system has been externally implemented by different providers. For instance, if the system deals with critical restrictions, a *regimented* agent platform like [7] should be used. On the other hand, if the domain of application allows the violation of norms, an enforcement architecture like [3] should be used.

Domain-like concepts. Agents, norms and organisations directly map to the important features of the healthcare environment at a high level, for the reasons described above. That is, healthcare specifically concerns people (patients and clinicians), the organisations they work for, and the regulations they must comply with. This aids discussion with domain experts, thereby easing such things as requirements elicitation and verification (though there are limits, as discussed below).

4.2 Other development approaches

In theory, an e-health system such as ePCRN-IDEA could be designed by a single organization in a centralized manner, following one of the many methodologies tailored to single, non-distributed systems. However, as described above, the required data and functionality is distributed among clinics, and their autonomy makes this unrealistic. Also, for most applications, the number of patients, clinics, trials etc. could produce a scalability problem.

Turning to more comparable development views, a service-oriented approach to development is clearly appropriate in some respects. It assumes some autonomy, in that services can be separately hosted and maintained, and allows for some openness, as existing published interfaces may be implemented by new services. However, a service-oriented application is generally controlled ultimately by a single client, the interfaces only partially specify how a service should behave, and there are no explicit norms (though service-level agreements can act as contracts for low-level quality of service demands).

Methodologies based on concepts of objects or components, regardless of the particularities of the methods themselves, also suffer from having less domain-like concepts than AOSE. This point is not healthcare-specific, but significant in any domain in which the requirements relate to user interaction rather than merely system component interaction. Objects and components will normally have parallels in the domain, but these will be of less direct concern than the people, organisations and regulations.

The comparison above is not to say that services or objects are irrelevant to developing e-health systems, but are inadequate in themselves compared to an AOSE approach. Many AOSE methodologies, including ROMAS, utilise service-oriented and object-oriented specifications of the functionality performed by agents.

4.3 AOSE weaknesses

There are two weaknesses of the current approach in applying ROMAS to ePCRN-IDEA. Although they are weaknesses of ROMAS, we believe them to apply to current AOSE methodologies more generally.

First, while conceptualizing the system in terms of agents, organizations and norms was found to be intuitive by domain experts, the language itself was not. There are terms in different areas of healthcare that are commonly used, and it would help the requirements and analysis process if software engineering principles could adopt these rather than agent abstractions. For example, when ‘patient’ is so critical a concept to the healthcare domain, modelling them as abstract ‘agents’ only obfuscates the intention. Similarly, the context in which the clinical researcher operates may be an organization, but for medics, such organizations are quite distinct from the ‘sites’, such as clinics or hospitals, from which patients are recruited.

Second, while there are explicit regulations in the domain, there are also many implicit good practices for medicine and healthcare. Capturing these as part of the engineering process is possible but prone to accidental exclusion. It is unclear why these need to be captured every time, and could instead be an embedded part of the methodology. In consequence, in our ongoing work, we are investigating how to address both weaknesses, and provide an AOSE methodology tailored more specifically to e-health.

5 Conclusions and future work

This paper explores the suitability of AOSE techniques for the development of complex systems in the healthcare domain. To investigate this domain, we have designed a real-time system for the identification of eligible patients for clinical trials based on an AOSE methodology. The results obtained show that the use of high level AOSE concepts, such as organizations, roles, norms and contracts, is beneficial to analyze and design health systems. Furthermore, it has been shown that the use AOSE techniques will produce flexible systems that can deal with the dynamics of the normative and technological environment.

References

1. R. H. Bordini, M. Dastani, and M. Winikoff. Current issues in multi-agent systems development. In *Engineering societies in the agents world VII*, pages 38–61, 2007.
2. A. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Modeling and reasoning about service-oriented applications via goals and commitments. *ICST Conference on Digital Business*, 2009.
3. N. Criado, E. Argente, and V. Botti. A normative model for open agent organizations. In *ICAI*, volume 1, pages 101–107. CSREA Press, 2009.
4. S. A. DeLoach. Developing a multiagent conference management system using the o-mase process framework. In *Proc. Int. Conf. on Agent-oriented software engineering VIII*, pages 168–181, 2008.
5. S. A. DeLoach, L. Padgham, A. Perini, A. Susi, and J. Thangarajah. Using three aose toolkits to develop a sample design. In *International Journal Agent-Oriented Software Engineering*, volume 3, pages 416–476, 2009.
6. V. Dignum, J. Meyer, F. Dignum, and H. Weigand. Formal Specification of Interaction in Agent Societies. *Formal Approaches to Agent-Based Systems*, 2003.
7. M. Esteva, B. Rosell, J. A. Rodriguez, and J. L. Arcos. AMELI: An agent-based middleware for electronic institutions. In *AAMAS04*, pages 236–243, 2004.
8. R. F. Fernandez, I. G. Magarinyo, J. J. Gomez-Sanz, and J. Pavon. Integration of web services in an agent oriented methodology. *Journal International Transactions on Systems Science and Applications*, 3:145–161, 2007.
9. E. Garcia, A. Giret, and V. Botti. Regulated open multi-agent systems based on contracts. In *Information Systems Development*, pages 243–255, 2011.
10. C.-E. Lin, K. M. Kavi, F. T. Sheldon, K. M. Daley, and R. K. Abercrombie. A methodology to evaluate agent oriented software engineering techniques. In *HICSS '07*, page 60, 2007.
11. P. R. Telang and M. P. Singh. Comma: A commitment-based business modeling methodology and its empirical evaluation. In *Proc. 11th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS).*, 2012.
12. 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 *In Proc. 4th Intl. Conference on eHealth*, 2011.
13. 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 *Modelling Foundations and Applications: MBSDI workshop*, 2011.
14. J. Vázquez-Salceda, R. Confalonieri, I. Gomez, P. Storms, S. P. Nick Kuijpers, and S. Alvarez. Modelling contractually-bounded interactions in the car insurance domain. *DIGIBIZ 2009*, 2009.