Agent-Oriented Software Engineering of Distributed eHealth Systems

Adel Taweel[1], Emilia Garcia[2], Simon Miles[1], and Michael Luck[1]

[1] King's College London, UK
{adel.taweel,simon.miles,Michael.luck}@kcl.ac.uk
[2] Universitat Politecnica de Valencia, Spain
mgarcia@dsic.upv.es

Abstract. Development of distributed ehealth systems is increasingly becoming a common necessity to work across organisations to provide efficient services. This requires healthcare information to be accessible, under appropriate safeguards, for research or healthcare. However, the progress relies on the interoperability of local healthcare software, and is often hampered by ad hoc development methods leading to closed systems with a multitude of protocols, terminologies, and design approaches. The ehealth domain, by requirements, includes autonomous organisations and individuals, e.g. patients and doctors, which would make AOSE a good approach to developing systems that are potentially more open yet retain more local control and autonomy. The paper presents the use of AOSE to develop a particular distributed ehealth system, IDEA, and evaluates its suitability to develop such systems in general.

Keywords: eHealth, Agent Architectures, Contracts, System of Systems.

1 Introduction

Scalable and autonomous distributed systems are increasingly in more demand to fulfil the emerging requirements of complex domains. *eHealth* is a typical example, which is rapidly becoming more and more dependent on large-scale integrated distributed software systems. These systems are increasingly required to provide new and innovative ways to improve patient care across dispersed organisations and interoperate with various types of other demanding systems, such as those for clinical research. The health domain presents a number of challenges and puts greater demands on a number of aspects that invite new methods and techniques to rethink the way its systems are developed. The continual expansion of need to provide services at the point of care, led to the spread and distribution of healthcare organisations and systems. Each functions autonomously, falls under its own sphere of control, and utilises its own domain techniques and speciality, yet they are all required collectively to provide an integrated and improved healthcare. A typical example is the UK National Health Service, which is, also, beyond its normal function is required to integrate with and provide the data and knowledge to drive clinical research. In this context, interactions are required to take place between components, systems, people and

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 332–341, 2013.

organisations managed by parties with different goals, possibly conflicting policies, incompatible data representations, and different needs and work methods. Not surprisingly, this can lead to serious challenges when they are required to work as a unit raising the need for integrating different systems in a trustworthy and consistent manner. This leads to the emergence of strict regulatory controls to manage not only the internal behaviour, but also the interactions that may take place between them.

Multi-agent technology has emerged over the last decade as a new software engineering paradigm for building complex, adaptive systems in distributed, heterogeneous environments. Although not mainstream, but it is provides a promising development approach. It fits well with several of the above concepts that often well addressed in the multi-agent systems, such as organisational autonomy, inherent regulatory frameworks, local control, regulated interactions and so forth. Indeed, there have been several agent-based e-health systems developed over a period of many years [1], [16], but they do not explicitly employ agent-oriented software engineering (AOSE) methodologies and, as such, do not directly evaluate the suitability of AOSE to this domain in general.

In this paper, we apply AOSE to develop a distributed ehealth system and investigate, more generally, the suitability of using AOSE for the development of distributed e-health systems. In doing so, we use a specific methodology, ROMAS (Regulated Open Multi-agent Systems) [10], [12]. 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. We apply the ROMAS methodology to a particular (real) e-health system: IDEA [20], [21], as an example, to allow exemplifying, more generally, the features of such systems, and to assess the suitability of AOSE in addressing them.

Section 2 describes the IDEA distributed ehealth system and summarizes the main challenges of the development of ehealth systems; Section 3 describes the design of the IDEA ehealth system using ROMAS; section 4 analyses the benefits that the IDEA system has obtained by means of using AOSE techniques; Section 5 discusses the suitability and general benefits of using agent methodologies in the analysis and design of e-health systems. 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, Section 6 presents some conclusions and future work.

2 IDEA eHealth System: Requirements and Challenges

Clinical trials are used to study various aspects of medical science, using evidence-based medicine [20], [21]. Currently, however, such trials are frequently unsuccessful at recruiting sufficient patients. This is because discovering and contacting eligible potential recruits is both logistically and legally challenging. IDEA is a new system, currently under deployment in the UK healthcare system, for recruiting patients for clinical trials in real-time. Fig. 1 illustrates the overall conceptual architecture of IDEA. It notifies practitioners in real-time whenever an eligible patient is in consultation. When a patient visits a practice, IDEA compares their details against a registry

of trials; if the patient is found eligible for one or more, the practitioner is prompted to help recruit the patient if they are interested. The IDEA project is discussed more fully here [20].

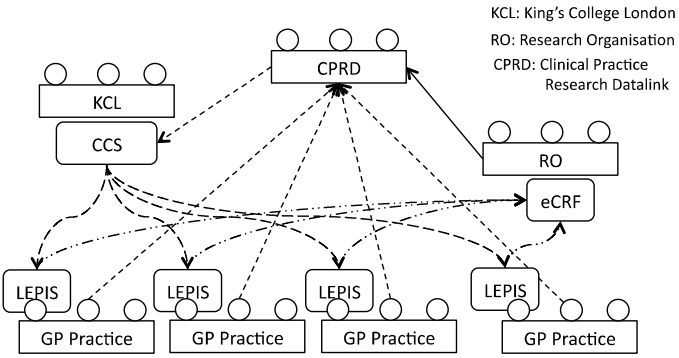


Fig. 1. IDEA Conceptual Architecture

However, to reach a more open system a more flexible development methodology needs to be used. To do so and to be able to identify, assess and use an effective methodology for the development of the IDEA system [21], its core challenges need to be considered. Below are the most important of these identified challenges to gain a better understanding of how AOSE might be more generally used effectively in the development process of such systems, which are typically of those in the medical domain.

Integration of Independent Systems. In order to recruit eligible patients, it is necessary for researchers, practitioners, patients, databases and practices 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 include [18], [19]: (i) *Distributed Data* (ii) *Technical Interoperability* (iii) *Process Interoperability* (iv) *Semantic Interoperability* (v) *Trustworthiness*. For an open and flexible healthcare system that involves multiple organizations, it must take all these aspects into account to ensure successful operation.

Regulation of Independent Systems. Healthcare systems must fulfil strict clinical and governmental regulations concerning the privacy and security of personal patient data. Moreover, each research institute and practice 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 regulatory environments.

Scalability of the System. Healthcare systems, by their nature, present scalability problems. For example, the effectiveness of the recruitment system is measured in the number of eligible patients that it is able to detect. In order to detect an adequate number of eligible patients for a specialized trial, the system must be able to get the information from as many practices as possible (more than 10000 GP practices in the UK) and manage a huge number of clinical trials requirements. However, due to the

number of potential active trials (potentially several hundreds) with the size of each trial description and eligibility criteria makes it impossible for GPs to know all active trials to assess patient eligibility during, the often short, consultation.

System Evolution. Medical institutions are constantly adapting their systems to reflect new legislation, software and medical techniques. As these autonomous organisations 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. changes in data formats). Healthcare systems that consist of multiple organizations must therefore ensure some formal procedure by which all parties understand and adhere to their responsibilities. Thus, to achieve, Institutions must also be contractually obliged to adhere to a standard interaction mechanism and data format, although their internal process or storage technology may differ [15].

3 Designing IDEA Recruitment System with ROMAS

This section details the design of the IDEA system using the ROMAS methodology [10], [12]. Fig. 2 depicts the main structure of the system in terms of agent-oriented software engineering key concepts including organisations, roles, norms and contracts. These are described below in more details.

Organizations and Processes. Several organizations are involved in the key processes performed in IDEA, as follows. When a research body wishes to create a new clinical trial, they can register 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. Each trial includes eligibility criteria of potentially eligible patients, which are managed by the *Clinical Practice Research Datalink* (CPRD), which operates a large data warehouse containing over 12 million up-to-date patient records. A software agent (named LEPIS) located on practitioners' computers communicates with CCS to obtain respective trials and their eligibility criteria for its own participating *practice* and general practitioner (GP). LEPIS agents continually listen to the interactions between the practitioner and their local Electronic Health Record (EHR), which manages patients' clinical information (e.g. diagnoses, treatments, drugs, etc.). During consultations, LEPIS agents check the eligibility of the patient to any of the registered trials. If the patient is found to be eligible for a trial, the practitioner is notified, and if the patient agrees to participate, LEPIS loads the appropriate electronic case report forms (eCRF) from a website provided by the *research organisation* responsible for the trial, allowing the patient's recruitment to be completed. Thus, the following organizations are involved: KCL, CPRD, GP practices and the research organisations.

Roles. The system is composed of six different roles presented below. The *CPRD Manager Role* is responsible for updating and controlling access to the CPRD database. It offers a service to pre-compute potential eligible patients for individual trials with complex search criteria (*CreateEligibilityList* service). The role must also offer a service to decide when a GP (and their own practice) is authorized to perform recruitment for each trial (*AuthorizeGP* service), while adhering to local and good clinical practice regulations.

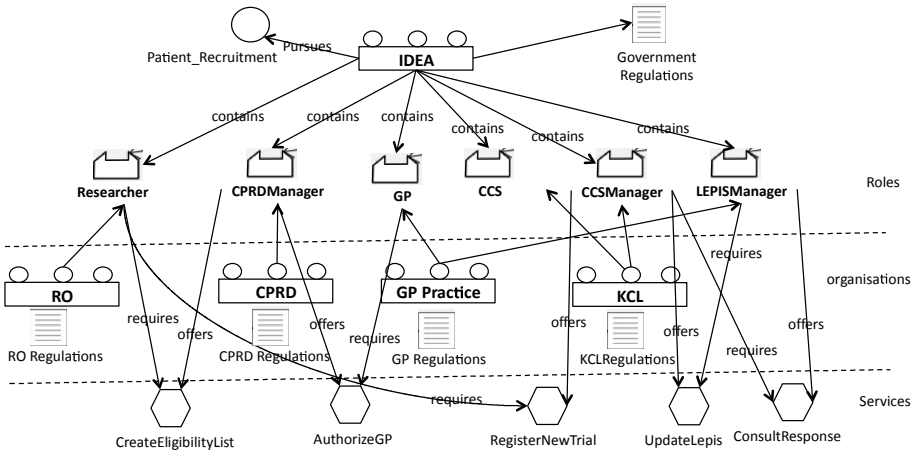


Fig. 2. IDEA Services and Organisations

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 and consented to participate in one of their own clinical trial. 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 CCS; 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 ensures consistency with LEPIS agents to update their information about active clinical trials; and (iii) a *Consult Patients Response* service that, in communication with LEPIS Agents, records the response of each consulted patient (and/or their GP) to be registered (whether they have agreed or refused to participate in a trial). In the current implementation, the CCS role is performed by an agent located at the KCL organization. Clearly, this agent must comply with established norms concerning replication of information and privacy.

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. A member of KCL undertakes this role, who complies with KCL's restrictions and rules.

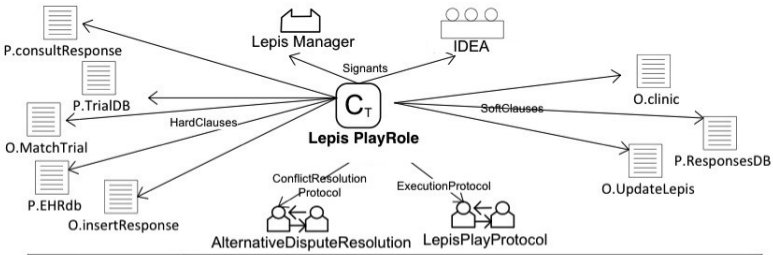
The *LEPIS Manager Role* is played by a software application that resides in each practice and investigates the eligibility of patients. LEPIS agent plays this role for each practitioner in each GP practice participating in the recruitment system. LEPIS agents continually communicate with the CCS service to acquire information about trials related to the type of patients and speciality of GPs and practices. 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 appropriate eCRF website if the patient has agreed to participate.

The *GP Role* represents a practitioner working in a practice. GPs must be previously authorized by the *CPRD Manager* before they can recruit patients into trials. This

authorization involves the acceptance of some norms related to good clinical practice, 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 IDEA system because their interaction with the system is always executed through their GP. Each role description is associated to a graphical diagram that allows a fast general overview. They have been omitted due to space restrictions.

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 IDEA system should comply with these regulations. At the same time, each institution and practice defines its own regulations, so the entities of the system should meet 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 practice-specific regulations. The rights and duties that any specific agent implementation must fulfil to play a role in 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. Due to space constraints, a comprehensive set of norms and contracts in IDEA cannot be listed; thus, we briefly cover a small number of examples.



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. LEPIS PlayRole Contracts

Fig. 3 describes the *LEPIS PlayRole* contract template. It specifies that any agent playing the LEPIS Manager role must detect changes in the EHR system and then it must check the eligibility of the patient for any of the registered trials (*Norm O.MatchTrial*). The contract template also recommends that the final 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 IDEA organization

exactly how often they should update their local registry. The remaining clauses relate to the use of the local registry and the service dependencies that LEPIS requires.

In this design, each practice can implement its own LEPIS agent (if it complies with the required contracts and norms), allowing each practice to adapt the behaviour of LEPIS in line with its own priorities. For example, a practice 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 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.

4 Benefits of AOSE Design for Distributed eHealth Systems

To reflect on the above AOSE design and the effectiveness of ROMAS for distributed ehealth systems, we revisit the design challenges listed above in Section 2.

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

Regulation of Independent Systems. The regulatory requirements of IDEA were found to fit well into the ROMAS principles. Specifically, it allowed different normative environments for each GP practice and research institution to be explicitly described and combined with global governmental norms. This allowed the behaviour of the different entities to be formally constrained --- an extremely important feature for some domains, e.g. the medical domain. Furthermore, different technologies can be used to implement the agents that play each role. For instance, each practice 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 practices from different countries.

System Evolution. More generally, ROMAS offers an effective paradigm for assisting in system evolution in 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.

5 Evaluation and Discussion

In the above sections, we have considered the use of an agent methodology to design an ehealth system, and the benefits achieved in doing so. We now consider the more general hypothesis that AOSE is more generally appropriate for the development of

ehealth systems. For the evaluation, we can draw more widely, not just on the experience from the IDEA system, but also on other multi-agent systems in ehealth. In general, AI technology including agent-based systems have been used in healthcare to tackle endemic issues such as distributed information and expertise, unpredictable dynamics, uncertainty in reasoning and simulation of systems [17], [23] [13].

We present the argument in terms of AOSE's features that make it well suited to the characteristics of distributed ehealth systems; and those that are not well addressed by AOSE. On the one hand, 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], [2], [14]. 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. [3], [7], [16]). Through the lessons learned during the development of IDEA, we now present some features of AOSE that indicate its suitability for general ehealth applications.

Autonomy. A critical aspect of ehealth systems is that they are comprised of subsystems that have their own regulations, privacy issues, localised authority, localised flexibility, and so on. For instance, in IDEA, different policies are applied in different GP practices, hospitals and regions in the UK. In this context, it is clear that ehealth systems must also take into account this diversity. The autonomy of agents and organisations assumed at the analysis stages in AOSE means that this is a particularly well-suited approach.

Openness. There are can be thousands of independent sites involved in healthcare. A common feature of large-scale ehealth systems, such as IDEA, is the expectation that more sites will join the system as they develop the technical capability to do so (e.g. new GP practices, research institutes etc.). In practice, openness is enabled by a design specifying exactly what a new party must adhere to 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 ehealth system. Regulations apply both to individual GP practices and researchers, and across the whole system due to national or potentially international laws. 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.

On the other hand, two weaknesses were found in applying ROMAS to IDEA. Although they are weaknesses of ROMAS, we believe they apply more generally to current AOSE methodologies.

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. For example, `patient' is a critical concept in the healthcare domain, modelling it as abstract as `agent' only obfuscates the intention.

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. One approach to address this issue could be to have domain constraints as an embedded part of the methodology.

6 Conclusions and Future Work

The paper presents the application of AOSE methodology for the development of complex distributed ehealth systems and assesses its suitability for the development of such systems more generally in the medical domain. To investigate this domain, we have applied the ROMAS methodology to the design of a real ehealth system for the identification of eligible patients for clinical trials in real-time. The results obtained show that the use of AOSE concepts, such as organizations, roles, norms and contracts, apply to the analysis and design of ehealth systems.

Although, it has been shown that the use of AOSE techniques offers several advantages to produce a flexible system design that can deal with the dynamics of the normative and technological environment in the healthcare domain; however, it has also shown that the current AOSE techniques and languages are not easily accessible by the domain experts and require further research to achieve greater domain-driven usability and adaption.

Acknowledgment. The authors would like to thank all colleagues from the IDEA project who contributed to this work, through discussions or providing insights.

References

1. Bajo, J., Fraile, J.A., Pérez-Lancho, B., Corchado, J.M.: The thomasarchi- tecture in home care scenarios: A case study. Expert Syst. Appl. 37(5), 3986–3999 (2010)
2. Bordini, R.H., Dastani, M., Winikoff, M.: Current issues in multi-agent systems development. In: O'Hare, G.M.P., Ricci, A., O'Grady, M.J., Dikenelli, O. (eds.) ESAW 2006. LNCS (LNAI), vol. 4457, pp. 38–61. Springer, Heidelberg (2007)
3. DeLoach, S.: Omacs a framework for adaptive, complex systems. In: Handbook of Research on Multi-AGent Systems: Semantics and Dynamics of Organizational Models, pp. 76–104. IGI Global (2009)
4. DeLoach, S.A.: Developing a multiagent conference management system using the O-maSE process framework. In: Luck, M., Padgham, L. (eds.) AOSE 2008. LNCS, vol. 4951, pp. 168–181. Springer, Heidelberg (2008)
5. DeLoach, S.A., Padgham, L., Perini, A., Susi, A., Thangarajah, J.: Using threeAOSE toolkits to develop a sample design. International Journal Agent-Oriented Software Engineering 3, 416–476 (2009)
6. Dignum, F., Dignum, V., Padget, J., Vázquez-Salceda, J.: Organizing web services to develop dynamic, flexible, distributed systems. In: iiWAS 2009, pp. 225–234 (2009)
7. Dignum, V.: A model for organizational interaction:based on agents, founded inlogic. PhD thesis, Utrecht University (2003)

8. Fernández, R.F., Magarinõ, I.G., Gómez-Sanz, J.J., Pavoń, J.: Integration of web services in an agent oriented methodology. Journal International Transactions on Systems Science and Applications 3, 145–161 (2007)
9. Fernandez, R.F., Magarinyo, I.G., Gomez-Sanz, J.J., Pavon, J.: Integration of web services in an agent oriented methodology. Journal International Transactions on Systems Science and Applications 3, 145–161 (2007)
10. Garcia, E., Giret, A., Botti, V.: Developing Regulated Open Multi-agent Systems
11. Garcia, E., Giret, A., Botti, V.: Software engineering for service-oriented MAS. In: Klusch, M., Pěchouček, M., Polleres, A. (eds.) CIA 2008. LNCS (LNAI), vol. 5180, pp. 86–100. Springer, Heidelberg (2008)
12. Garcia, E., Giret, A., Botti, V.: Regulated open multi-agent systems based on contracts. In: Information Systems Development, pp. 243–255 (2011)
13. Gonzalez-Velez, H., Mier, M., Julia-Sape, M., Arvanitis, T., Garcia-Gomez, M.R.J., Lewis, P., Dasmahapatra, S., Dupplaw, D., Peet, A., Arus, C., Celda, B., Huel, S.V., Lluch-Ariet, M.: Healthagents: distributed multi-agent brain tumor diagnosis and prognosis. Applied Intelligence 30 (2009)
14. Lin, C.-E., Kavi, K.M., Sheldon, F.T., Daley, K.M., Abercrombie, R.K.: A methodology to evaluate agent oriented software engineering techniques. In: HICSS 2007, p. 60 (2007)
15. Oren, N., Panagiotidi, S., Vázquez-Salceda, J., Modgil, S., Luck, M., Miles, S.: Towards a formalisation of electronic contracting environments. In: Hübner, J.F., Matson, E., Boissier, O., Dignum, V. (eds.) COIN@AAMAS 2008. LNCS, vol. 5428, pp. 156–171. Springer, Heidelberg (2009)
16. Paranjape, R., Sadanand, A., Paranjape, R., Sadanand, A.: Multi-Agent Systems for Healthcare Simulation and Modeling: Applications for System Improvement, 1st edn. Information Science Reference - Imprint of: IGI Publishing, Hershey (2009)
17. Rammal, A., Trouilhet, S., Singer, N., Pecatte, J.-M.: An adaptive system for home monitoring using a multiagent classification of patterns. In: International Conference on Business Process Management, pp. 3:1–3:8 (2008)
18. Taweel, A., Delaney, B., Speedie, S.: Towards achieving semantic interoperabil- ity in ehealth services. In: Watfa, M. (ed.) E-Healthcare Systems and Wireless Communications: Current and Future Challenges, pp. 388–401. IGI (2012)
19. Taweel, A., Speedie, S., Tyson, G., Tawil, A.R., Peterson, K., Delaney, B.: Service and model-driven dynamic integration of health data. In: Proceedings of the First International Workshop on Managing Interoperability and Complexity in Health Systems, MIXHS 2011, pp. 11–17. ACM (2011)
20. Tyson, G., Taweel, A., Miles, S., Luck, M., Staa, T.V., Delaney, B.: An agent- based approach to real-time patient identification for clinical trials. In: Kostkova, P., Szomszor, M., Fowler, D. (eds.) eHealth 2011. LNICST, vol. 91, pp. 138–145. Springer, Heidelberg (2012)
21. Tyson, G., Taweel, A., Zschaler, S., Staa, T.V., Delaney, B.: A model-driven approach to interoperability and integration in systems of systems. In: Modelling Foundations and Applications: MBSDI Workshop (2011)
22. Vecht, B., Dignum, F., Meyer, J.-J., Dignum, M.: Handbook of research on multi- agent systems: Semantics and dynamics of organizational models. In: Autonomous Agents Adopting Organizational Rules, pp. 314–333. IGI Global (2009)
23. Zhang, X., Xu, H., Shrestha, B.: Building a health care multi-agent simulation system with role-based modeling. In: MAS for Health Care Simulation and Modeling: Applications for System Improvement, ch. VI. IGI Global (2009)