Modelling an agent-based integrated health and social care information system for older people

Haralambos Mouratidis[1,2], Gordon Manson[1], Paolo Giorgini[3], Ian Philp[2]

Abstract. This paper argues the potential of agent technology to support delivery of integrated information systems for the health and social care sector. In doing so, it points out the similarities and the mutual characteristics (such as distribution of expertise) of integrated health and social care information systems and agent technology. On the other hand, it identifies the need to develop a mature and complete agent-oriented software engineering methodology tailored to the analysis and design of agent-based medical systems. The electronic Single Assessment Process (eSAP), an electronic system to deliver the integrated assessment of health and social care needs of older people, is modelled using the *Tropos* methodology.

1 INTRODUCTION

Health Care information systems are becoming more and more computerised. A huge amount of health related information needs to be stored and analysed, and with the aid of computer systems this can be done faster and more efficiently. In a distributed health care setting different health care professionals, such as general practitioners and nurses, must cooperate in order to provide patients with appropriate care and must also work closely with social care professionals, such as social workers, because health and social care needs quite often overlap.

National policy in England is to promote the Single Assessment Process (SAP), an integrated assessment of health and social care needs of older people. The Single Assessment Process aims to create closer working for providing primary health and social care for older people and other groups. Computerising this process will help to automate some of the administration tasks (such as the management of the health and social care teams, appointments procedures and secure and anonymised sharing of medical records) of the health and social care professionals and thus leave the professionals with more time for the actual care of the older person.

Nevertheless, computerising this process is not an easy task. Only about 1% (in UK) of some health and social care professionals are using computer systems. Apart from the complexity of such a system because of the integration, security concerns and mobility there are other concerns, related to the domain of giving health care to older people, that must be taken into consideration. Thus, the fact that most of the time professionals have to use the system whilst dealing with older people is a concern that must be taken into consideration. Furthermore, the services that such a system will provide to the health and social care professionals are very important as well and plenty of thought must be given. Thus, one of the most important decisions in computerising such a process is the choice of the technology that must be used.

We are developing the electronic Single Assessment Process (eSAP), an electronic system to deliver an integrated assessment of health and social care needs of older people. The project is run jointly between the Computer Science Department of the University of Sheffield and the Sheffield Institute for Studies on Ageing (SISA), and it is funded by the RANK Foundation.

An initial exploration showed that Agent Technology looks very promising to fulfil the requirements of an integrated health and social care computerised system. The scenario of distributed health and social care suits well to an agent-based system since there is distribution of data (each professional owns their data about the patient), cooperation between the different professionals (exchange of information about the older person), and different expertise areas between the professionals.

Nevertheless, it is well recognised amongst the research community [1, 2, 3] the lack of a mature and complete analysis and design methodology that will help during the analysis and design stages of an agent-based system. The need is greater in the development of health and social care information systems [4].

We are developing an analysis and design methodology, called *Tropos*, to help in the analysis and design of agent-based health and social care information systems. *Tropos* covers four development stages: early and late requirements analysis, architectural design and detail design.

Section 2 presents the problem of the complex needs assessment and information sharing in health and social care of older people, and also it describes a solution to this problem, the Single Assessment Process. Section 3 tests the suitability of employing agent technology in the development of integrated health and social care systems. The need for a complete and mature analysis and design methodology for agent-based medical systems is introduced in Section 4, and Section 5 presents the modelling of the electronic Single Assessment Process (eSAP) using *Tropos* methodology. Conclusions and Future Work are presented in the last Section.

[1] Computer Science Department, University of Sheffield, England, email: {h.mouratidis, g.manson}@dcs.shef.ac.uk
[2] Sheffield Institute for Studies on Ageing (SISA), England, email: i.Philp@shef.ac.uk
[3] Department of Information and Communication Technology, University of Trento, Italy, email: pgiorgini@science.unitn.it

2 THE PROBLEM OF THE COMPLEX NEEDS ASSESSMENT AND INFORMATION SHARING IN HEALTH AND SOCIAL CARE OF OLDER PEOPLE AND THE SINGLE ASSESSMENT PROCESS

Older people often have a complex mixture of health and social care needs and several different health and social care professionals are involved in their care. Health and social care professionals such as General Practitioners (GPs), nurses, and social workers are mostly involved in the care of older people with assessments often undertaken in the older person's home. Currently these professionals might belong to different organisations such as GP offices, community services, and social services but national policy is to pursue close working amongst these groups.

As can be seen from the above, there is a team of different professionals working around one or more older persons. Each team consists of different specialist health care professionals, who must share their knowledge and the available information, obtained by the different assessments, in order to identify exactly the health needs of the older person and provide better care.

Thus many problems may arise amongst these professionals. One of the most important problems is communication between the professionals. Professionals must communicate in order to share their expertise. Another problem is cooperation and the exchange of the information related to the older person, such as assessment results. Furthermore, the coordination of the professionals is another problem that must be solved. All these problems finally result in duplication of assessments, lack of awareness of key concepts of need (even if known to another professional) and fragmentation of care.

In order to solve these problems long-term national policy in England is to create closer working for providing primary health and social care for older people and other groups. Ultimately this might lead to the development of Care Trusts, a single local organisation for delivering health and social care. The development of integrated health and social care information systems will support closer working and facilitate the development of Care Trusts.

With closer working, professionals will work in teams that will be responsible for the health and social care of the older person. Each team will demonstrate the following characteristics.

- Each team consists of many different professionals.
- Professionals cooperate between them.
- Professionals share information between them.
- Each professional has some expertise.
- Teams will promote person-centred care.

With the single assessment process, a common language of assessment will be used to support information sharing, and the potential to aggregate information to describe the health and social care needs of the local population of older people, which is an extremely useful tool for planning services and monitoring trends in needs and outcomes.

The single assessment process will also provide the older person and their carer with a personal copy of their care plan to support person-centred care. The single assessment process will work in three main stages. The contact, the assessment and the follow-up action. During the first stage contact assessment will provide basic information. In the second phase an overall assessment using a validated assessment instrument, such as Easy-Care [5], will take place. The third stage will provide older people with more care in particular problems (might be different problems for each individual such as housing and loneliness problems) and with more detailed assessment if appropriate. The selection of the problems is determined by the results obtained from the Easy-Care assessment instrument.

Computerising this process will help to automate some of the administration tasks, such as the appointments set up between the health and social care professionals, and the management of the health and social care teams, thus leaving the professionals with more time for the actual care of the older person. Furthermore, it will help older people to be actively involved in their health and social care, since they will have access to the system.

3 SUITABILITY OF AGENT TECHNOLOGY

An exploration of the computerised single assessment process system showed that it can be viewed as a computer system based on the agent technology. This is because of the mutual characteristics that both systems display. The single assessment process system consists of many different health and social care professionals. These professionals cooperate with each other, share information, and each of them possesses some expertise.

In a Multi-Agent System a software agent is considered a problem-solving entity. In this type of system a complex task is accomplished by combining different software agents that posses different skills (expertise). All the different software agents that co-exist in the system possess their own expertise, which can be related to the other agents of the system but it is distinct, and they use this expertise at different stages of the solving process in order to accomplish the system's aim.

A reason that makes Multi-Agent Systems very attractive to researchers is because these systems can be viewed as human societies in which the software agents play the roles of the human beings, or there is a mix between human beings and software agents that cooperate and communicate in order to solve a complex problem. Thus, a software agent in the system can be viewed either as an entity that acts on behalf of a human, or as an autonomous entity that possesses some expertise and is able to cooperate and communicate.

It is concluded from the above that the single assessment process can be modelled into a multi agent system that consists of many different software agents (and humans), which can cooperate with each other, share information (distribute information) and each of them possesses some expertise. A summary of the mutual characteristics of the Single Assessment Process and a multi-agent system is given in Table 1.

Table 1 Mutual Characteristics of the Single Assessment Process and a multi-agent system

Single Assessment Process	Agent-Based System
Professionals	Software Agents
Cooperate	Cooperate
Expertise	Expertise
Distribution	Distribution

In the eSAP system, the software agents will act on behalf of professionals. Each professional will have his/her "own" software agent, which will customise according to his/her needs. The agent will have enough information about the professional, such as personal information and professional commitments, and it will be intelligent enough (capable of analysing the information and take decisions) that will enable it to act on his/her behalf, and also negotiate for the interest of the professional. The system will have the following characteristics:

1. The system will consists of software agents as well as human professionals.
2. Each professional will have his/her software agent.
3. The professionals must able to customize their software agents through an easy-to-use interface.
4. The system must be developed with mobility in mind since many of the professionals will use it whilst in the older person's house.
5. The system must be secure.
6. The software agent will be capable of analysing information and take decisions. Also, it will have information about the professional (personal and professional) that will be able to act on his/her behalf.
7. Software Agents in the system will be able to communicate between themselves as well as with the human professionals.

4 THE NEED FOR A COMPLETE METHODOLOGY TAILORED TO THE ANALYSIS AND DESIGN OF AGENT-BASED MEDICAL SYSTEMS

From the above, it is derived that agent technology is suitable to fulfil all the characteristics required by such a system. Nevertheless, in developing such a complex system the first step is to analyse and design the system. Designing and analysing a system before it is actually implemented helps to better understand the system requirements, the user needs and thus the whole system. Starting to implement a system without designing it might work for small systems that only require a few lines of code and one developer. However, trying to implement complex distributed systems (like the health care systems), that require hundreds or thousands of lines of code, and large teams of developers, without a design proves to be a nightmare.

Thus, it is recognised amongst the agent research society [1, 2, 6, 7, 3] that there is a need for a complete analysis and design methodology for multi-agent systems. The main role of such a methodology will be to help in all the phases of the development of an agent-oriented system. And more importantly to help capture and model the unique characteristics that agent-oriented systems introduce such as flexibility, autonomous problem-solving, and the rich of interactions between the individual agents. An adequate analysis and design methodology needs to capture and model all these unique characteristics of agent oriented systems as well as the interactions between agents and non-agent elements (e.g humans), the states (internal and external) of the agents that constitute the multi-agent system, as well as the goal-directed behaviour and the actions of the agents. Also, a methodology to support design and specification of agent systems should provide a clear conceptual framework that enables the complexity of the systems to be managed by decomposition and abstraction. There are plenty of issues that must be considered when analysing and designing such systems. For example, coordination, communication and cooperation between the software agents of a Multi-Agent System are important factors in such a system and they impose some more complex issues in the design of the system. Thus the methodology must be able to deal with all these matters of the analysis and design process of a multi-agent system.

In addition, trying to model integrated health and social care systems introduces some extra requirements [4]. The need to efficiently capture the environment is essential in health and social care systems, especially when there is no existing system (like in our case), either "human" or electronic. Also some security concerns must be taken into consideration during the modelling of the system. Security is a major concern in health and social care information systems. The system will contain personal and medical information and thus must be very secure

5 APPLYING TROPOS METHODOLOGY

Tropos is a methodology, for building agent-oriented software systems, tailored to describe both the organisational environment of a system and the system itself, employing the same concepts through out the development stages. Thus, *Tropos* helps the designer to understand not only the system itself but also the environment of the system, something very important especially in health and social care systems, in which understanding of the environment plays a major role. *Tropos* adopts the i* modelling framework [8], which uses the concepts of actors, who can be (social) agents (organisational, human or software), positions or roles, goals and social dependencies (such as soft goals, tasks, and resources) for defining the obligations of actors (dependees) to other actors (dependers). This means the system (as well as its environment) can been seen as a set of actors, who depend on other actors to help them fulfil their goals. *Tropos* covers four phases of software development:

Early Requirements, concerned with the understanding of a problem by studying an existing organisational setting; the output of this phase is an organisational model, which includes relevant actors and their respective dependencies;

Late requirements, where the system-to-be is described within its operational environment, along with relevant functions and qualities; this description models the system as a (small) number of actors which have a number of dependencies with actors in their environment; these dependencies define the system's functional and non-functional requirements;

Architectural design, where the system's global architecture is defined in terms of subsystems, interconnected through data and control flows; within the framework, subsystems are represented as actors and data/control interconnections are represented as (system) actor dependencies;

Detailed design, where each architectural component is defined in further detail in terms of inputs, outputs, control, and other relevant information. *Tropos* is using elements of AUML [9] to complement the features of i*.

5.1 Early Requirements Analysis

As was mentioned above, during the early requirements analysis the analysis engineer models the goals and the dependencies between the stakeholders (actors). For this purpose, *Tropos* introduces actor diagrams. In such a diagram each node represents an actor, and the links between the different actors indicate that one depends on the other to accomplish some goals. The electronic Single Assessment Process project (eSAP) includes, among the others, the following stakeholders:

- Older Person (OP) is the patient who wishes to receive appropriate health and social care.
- Department of Health (DoH) is the government department that must provide older people with health and social care.
- R&D Agencies are agencies that wish to obtain information about older people to help them in their research.

These actors along with their goals are shown in Figure 1. The ***Older Person*** actor has a main goal to receive appropriate health and social care and a soft goal to *Maintain Good health*. The ***Older Person*** depends on the ***DoH*** to accomplish their goal and soft goal. On the other hand, ***DoH*** has a main goal to *Provide Better Health and Social Care to Elderly*. The ***R&D Agency*** has an associated goal to *Obtain Patient Information For Research*.

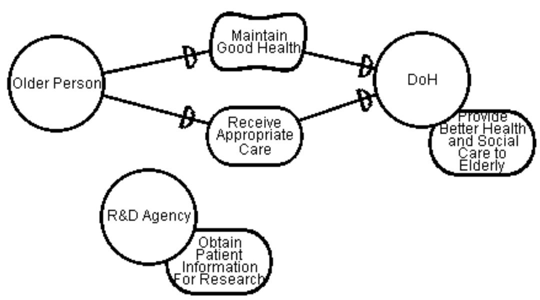

Figure 1 The Stakeholders of the eSAP

When the stakeholders, their goals and the dependencies between them have been identified, the next step of this phase is to analyse in more depth each goal relative to the stakeholder who is responsible for its fulfilment. In doing so, *Tropos* adopts *i** rationale diagrams for analysing the actors' goals through a means-end analysis. Each rationale diagram is presented as a balloon within which the goals and the dependencies are analysed. Such an analysis from the prospective of the ***Older Person*** is shown in Figure 2.

As it was mentioned, the ***Older Person*** actor has a goal to *receive appropriate care* and a soft goal to *maintain good health*. The *receive appropriate care* goal is fulfilled by the tasks *Undertake Assessment*, *Follow Care Plan*, *Get Info about Care Plan*, *Obtain Medical Info*, and *Have Appointment with Professionals*. To perform the last three tasks the ***Older Person*** must use the eSAP system, so the last three tasks are decomposed into the goal *Use eSAP System*. In addition, the *Maintain Good Health* soft goal depends on the *Follow Care Plan* and *Undertake Assessment* tasks to be fulfilled. Furthermore, the ***Older Person*** depends on the ***DoH*** to make the *Technology Infrastructure Available*, and also to make the *eSAP System Available* and Easy-to-Use.

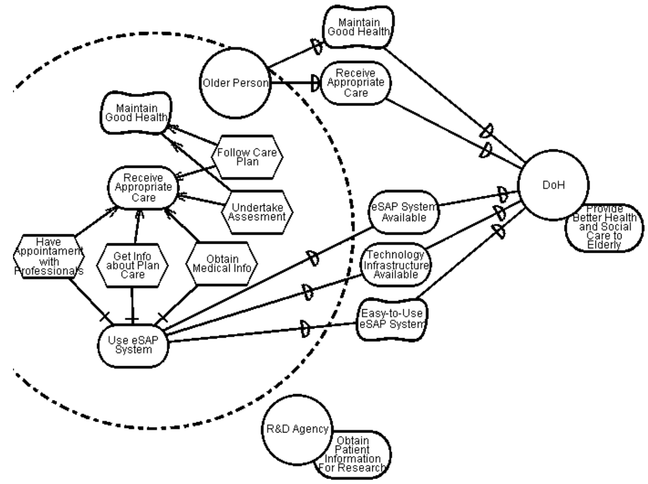

Figure 2 Means-end analysis for Older Person

Another important stakeholder of the system is the ***DoH***. The rationale diagram for the ***DoH*** is shown in Figure 3. The main goal of the Department of Health is *to Provide Better health and Social Care to Elderly*. To accomplish this goal, ***DoH*** defines a sub goal to *Make Care Person-Centred*. The latter is essential for the ***DoH*** to fulfil its main goal since the ***Older Person*** is the most important participant of the whole procedure, since they know better than anyone their difficulties and when they need health and social care. Thus, the *Make Care Person-Centred* goal is further decomposed into the two sub-goals *Promote SAP* and *Involve Elderly in their Care*. The later sub-goal depends on the task *Provide Guidelines for the Older People* to be fulfilled. The *Promote SAP* sub-goal is decomposed into the *Computerise SAP* goal and the *Provide Guidelines for the SAP* task. The later task is decomposed into two sub-tasks: *Provide Guidelines for the Different Health and Social Care Professionals* and *Provide Guidelines for the Care-Teams*. The first sub-task is further decomposed into four sub-tasks: *Provide Guidelines for GPs, Provide Guidelines for Nurses, Provide Guidelines for Other Professionals* and *Provide Guidelines for Social Workers*. In addition, to fulfil the *Provide Guidelines for the Care-Teams* goal each locality must comply with the proposed guidelines. *Provide Efficient Care* is another important goal of the ***DoH***. To accomplish this goal the sub goal *Computerise SAP* has been identified. Computerising the SAP will help health and social care professionals to automate some procedures required while caring for the older person and thus help to *Provide Efficient Care*. To accomplish the *Computerise eSAP* sub goal, *Technology Infrastructure* must be provided and the eSAP System must be available. The goal *Build eSAP* is motivated by these two goals since it has no sub-goals.

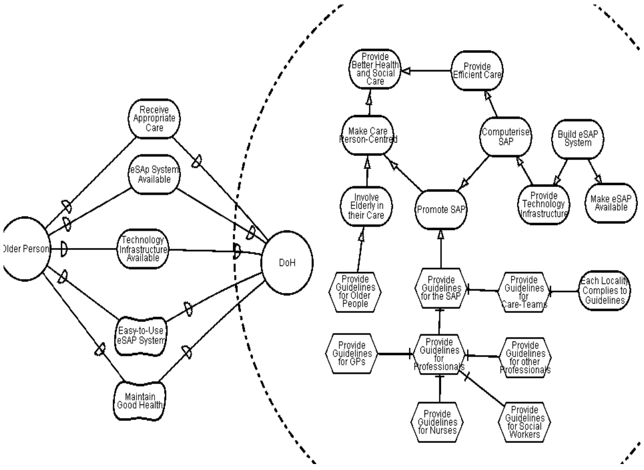

Figure 3 Means-end analysis for Department of Health (DoH)

5.2 Late Requirements Analysis

During the late requirements analysis the system-to-be (the eSAP system in our case) is described within its operation environment, along with relevant functions and qualities. The system is presented as one or more actors, who have a number of dependencies with the other actors of the organization. These dependencies define all functional and non-functional requirements for the system-to-be.

The eSAP system is introduced as another actor, as shown in Figure 4. The ***DoH*** depends on the eSAP to *Provide Efficient Care*, and also to *Make Care Person-Centred*, in order to fulfil its main goal, which is to *Provide Better Health and Social Care*. In our example we concentrate on the analysis of the goals *Provide Efficient Care, Make Care Person-Centred,* and *Usable eSAP System*.

The goal *Provide Efficient Care* aims to allow health and social care professionals more time for the actual health and social care of the older person. Plenty of time is currently spent from the health and social care professionals for administrating procedures, such as appointments set up, and communication with colleagues. eSAP will minimise the time spend on these procedures by automating most of the tasks and providing services, such as access to medical libraries, and medical records. Thus, the *Provide Efficient Care* goal is decomposed into the *Provide Services* sub-goal, which is further decomposed into four sub-goals, *Access to Medical Libraries, Access to Medical Records, Schedule Appointments* and *Access to Care Plans*. The later goal is further decomposed into the sub-goal *care plan info,* which is decomposed into four further sub-goals *Future Care Plan Appointments, Professionals Involved, Previous Assessments,* and *Future Actions*.

Another important goal of the eSAP system is to promote person-centred care. To fulfil this goal, eSAP must *Provide Facilities to Older People*. The later is decomposed into two further goals *Access to Care Plan* and *Access to Medical Info*. The soft goal *Usable eSAP System* has three positive (+) contributions from the *Easy-to-Use* soft goal, which contributes positively because the system must be Easy-to-Use to be usable, from the *Mobile* soft goal because the system must be mobile to be usable, and also from the *Secure eSAP* soft goal, which contributes positively since it makes the system secure.

The *Easy-to-Use* soft goal has two positive contributions from the *System Provides Help* and the *User Friendly Interface* soft goals. The former contributes positively since the system must help the user to be easy-to-use, and the latter contributes positively because the system must have a *User Friendly Interface* to be easy-to-use. In addition the *Easy-to-Use* soft goal has a negative (-) contribution from the *Secure eSAP* soft goal, since usually trying to make the system secure makes it more difficult to use.

The *Mobile* soft goal accepts two positive contributions from the *Portable* and the *Synchronise Data* soft goals. The former contributes positively because the system must be portable to be mobile, and the latter because the system must be able to synchronise data in order to be mobile.

Furthermore, the *Secure eSAP* soft goal receives three positive contributions. The first positive contribution comes from the *Authorise Access* soft goal, which contributes positively because the system must be able to *Authorise Access* to be secure. The other two positive contributions come from the *Secure Exchange of Data* and the *Secure Communications* soft goals. The former acts positively because the exchange of data must be secured for the system to be secure, and the latter because communications must be secured for the eSAP system to be secure.

In addition, the *Secure eSAP* soft goal has a negative contribution from the portable soft goal because a portable system is more difficult to secure.

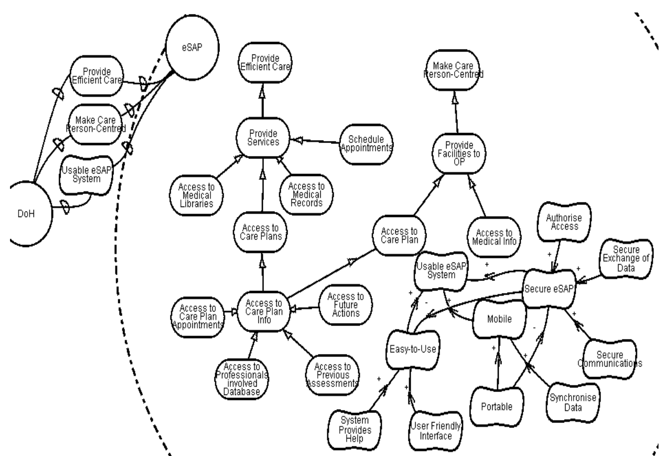

Figure 4 Means-end analysis for eSAP System

When the system goals and soft goals have been identified, new actors and sub-actors are introduced. Each of the new actors takes the responsibility to fulfil one or more goals of the system. Figure 5 shows a partial decomposition of the actors and sub-actors of the eSAP system, along with their dependencies with respect to the eSAP system.

The eSAP system depends on the ***Medical Library Manager*** to provide *Access to Medical Libraries*, on the ***Medical Record Manager*** to provide *Access to Medical Records*, on the ***Appointments Manager*** to *Schedule Appointments*, and on the ***Care Plan Manager*** to provide *Access to Care Plans*. The ***Care Plan Manager*** depends on the ***Care Plan Appointments Manager*** to Access C*are Plan Appointments*, on the ***Professionals Manager*** to provide information about the professionals involved in the care plan, on the ***Assessments Manager*** to manage *Previous Assessments* and on the ***Future Actions Manager*** to manage the *Future Actions* required by the care plan. Furthermore, the eSAP depends on ***Security Manager*** to fulfil the *Secure eSAP System* goal. The ***Security Manager*** depends on the ***Authorisation Manager*** to Authorise Access to the System, on the ***Confidentiality Manager*** to assure the confidentiality of the data of the system, and on the ***Integrity Manager*** to assure the integrity of the data.

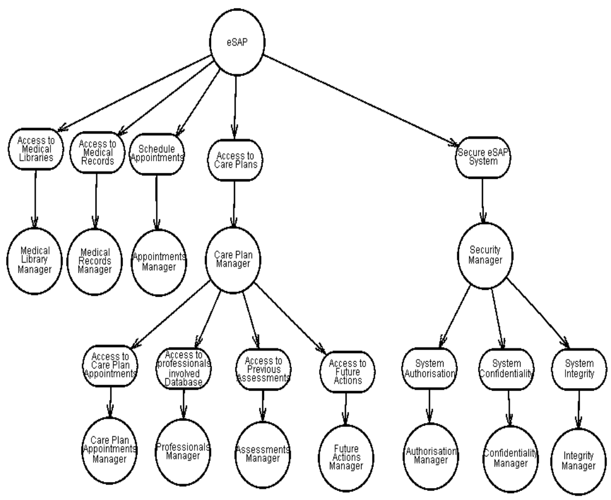

Figure 5 Sub-Actors Decomposition for the eSAP System

5.3 Architectural Design

The architectural design includes the following four steps:

- *Addition of new actors*, in which new actors are added to make the system interact with the external actors as well as to contribute positively to the fulfilment of some non-functional requirements. *Tropos* introduces the extended actor diagram in which the new actors and their dependences with the other actors are presented.
- *Actor decomposition*, in which each actor is described in detail with respect to its goals and tasks.
- *Capabilities identification*, in which the capabilities needed by the actors to fulfil their goals and tasks are identified, by analysing the extended actor diagram. Each dependency relationship can give place to one or more capabilities triggered by external events.
- *Agents assignment*, in which a set of agent types is defined assigning to each agent one or more different capabilities.

Figure 6 shows the extended actor diagram with respect to the ***Authorisation Manager***. The ***Authorisation Manager*** is responsible for authorising access to the ***Older Person (O.P)*** when trying to perform the *Get Info About Care Plan* task. When authorisation is granted the ***O.P.*** interacts with the *Care Plan Manager* in order to obtain available Care Plan Information. Then depending on the kind of information the ***O.P.*** wishes to access, he/she interacts with the appropriate manager. Again the ***Authorisation Manager*** checks for authorisation permissions and grants access to the ***Care Plan Data Manager***, which manages a repository for the care plan information. To represent the interactions of the health and social care professionals within the system we have introduced the ***Professional*** actor. The ***Professional*** interacts with the ***Authorisation Manager*** to gain the authorisation to interact with the ***Medical Library***, ***Medical Records***, and ***Appointment*** managers. In addition, the ***Authorisation Manager*** interacts with the ***R&D Agency*** actor to grant authorisation to the ***Care Plan Data Manager***.

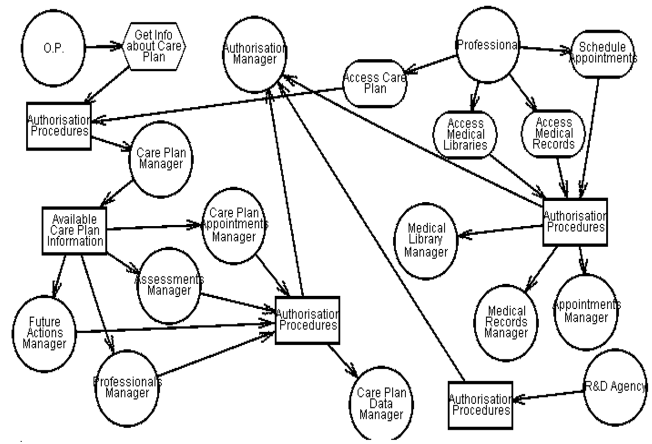

Figure 6 Extended Actor Diagram with respect to the Authorisation Manager

The second step of the architectural design is the decomposition of actors in sub-actors aiming to expand in details each actor with respect to its goals and tasks. Figure 7 shows the decomposition of the ***Authorisation Manager*** Actor with respect to the *get information about care plan* task. The ***Authorisation Manager*** is decomposed in two sub-actors: the ***Authorisation Granter*** and the ***Authorisation Checker***. The former is responsible for checking the users' authorisation details and grant access to services, while the latter is responsible for checking if the user has access and in what services. The ***Authorisation Granter*** depends on the ***Authorisation Checker*** to obtain the *Authorisation Privileges* of the user. The ***Authorisation Checker*** interacts with the *Authorisation Data* in order to obtain information about the *Authorisation Status* of the user.

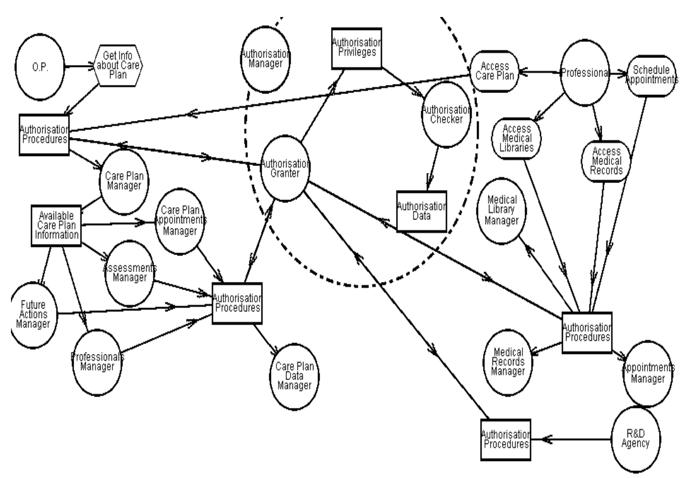

Figure 7 Extended Diagram with respect to the Authorisation Manager – Internal Decomposition

The next step of the architectural design is the capabilities identification, in which the capabilities needed by each actor to fulfil their goals and tasks are modelled. The extended actor diagram is used to identify the capabilities, since each dependency relationship can give place to one or more capabilities triggered by external events.

The last step of the architectural design is the agents' assignment. During this step each agent is assigned one or more different capabilities identified in the previous step. Table 1 illustrates the agents along with the capabilities assigned to each one of them with respect to the task *Get Info about Care Plan*, shown in Figure 7.

Table 2 Agent Types and their Capabilities w.r.t. Extended Diagram of Figure 7.

Agent	Capabilities
Authorisation Granter	Get User Authorisation Details Get User Authorisation Privileges Allow Access to Services Deny Access to Services
Authorisation Checker	Provide Info about Users' Authorisation Privileges Check User Authorisation Status
Older Person	Provide Information about Older Person Provide Authorisation Details of Older Person Provide Service Description Get Older Person query Act on Behalf of Older Person
Care Planner	Provide service description Provide available Care Plan Information Re-direct User to Appropriate service Manager
Future Actions Agent	Get Older Person query about Future Actions Get Query Results Provide Info about Future Actions
Professionals Agent	Get query about Professionals Get Query results Provide info about Professionals
Previous Assessments Agent	Get Query about Previous Assessments Get Query results Provide Info about Previous Assessments
Appointments Agent	Get Query about Appointments Get query results Provide Info about Appointments
Professional	Act on Behalf of Professional Provide Service Description Provide Info about the Professional Provide Authorisation Details of the Professional Get Professional's query
Services Facilitator	Give access to Medical Libraries Give access to Medical Records Obtain request for Appointment Provide Service Description Get request for Service

5.4 Detailed Design

Detailed design stage aims at specifying agent capabilities and interactions. Thus, during this stage internal and external events that trigger plan and the beliefs involved in agent reasoning are modelled. In our approach we have adapted a subset of the AUML diagrams proposed in [9]. These are:

- *Capability Diagrams*. We use AUML activity diagrams to model a capability or a set of capabilities for a specific actor. In each capability diagram, the starting state is represented by external events, activity nodes model plans, transition arcs model events, and beliefs are modelled as objects. An example of a capability diagram is shown in Figure 8, in which the *allow (or deny) access to services* capability of the Authorisation Granter Agent is illustrated. The ***Authorisation Granter*** initially accepts an authorisation request from the user. It compares the user's authorisation details with the user authorisation status that exist in the system, and either allow or deny access to services.
- *Plan Diagrams*. Plan Diagrams are used to further specify each plan node of a capability diagram. Figure 9 illustrates a plan diagram for the *Accept Authorisation Request from User* plan. The ***Older Person*** agent sends an authorisation request to the ***Authorisation Granter***. The ***Authorisation Granter*** checks the authorisation request integrity, by interacting with the ***Integrity Manager***, and if the integrity information is valid the ***Authorisation Granter*** accepts the authorisation request from the user and acknowledges the authorisation request, otherwise it rejects the authorisation request and notifies user about the rejection.
- *Agent Interaction Diagrams*. We apply in our case sequence diagrams modelling agent Interaction Protocols as proposed by [10]. An example of an Agent Interaction Diagram is shown in figure 10. The ***Older Person*** sends an ***Authorisation Request*** to the ***Authorisation Granter*** who acknowledges the request. Then the ***Older Person*** sends the user's authorisation details to the ***Authorisation Granter*** who forwards them to the ***Authorisation Checker***. The ***Authorisation Checker*** checks the user's ***Authorisation Status*** and replies to the ***Authorisation Granter***, who grants or refuses access.

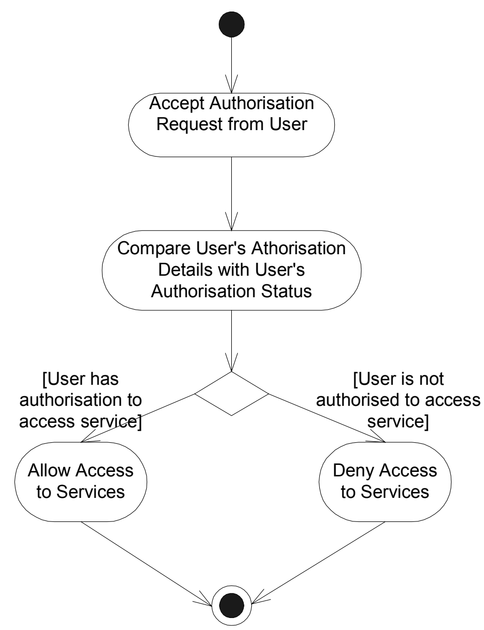

Figure 8 Capability Diagram for the *Allow or Deny Access* capability

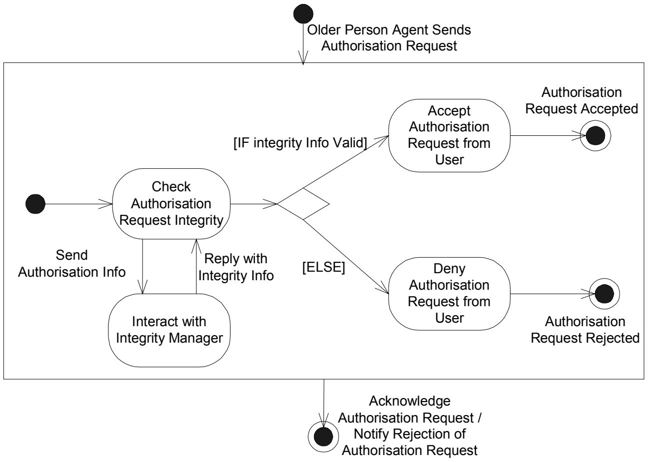

Figure 9 Plan Diagram for the Accept Authorisation Request from the User

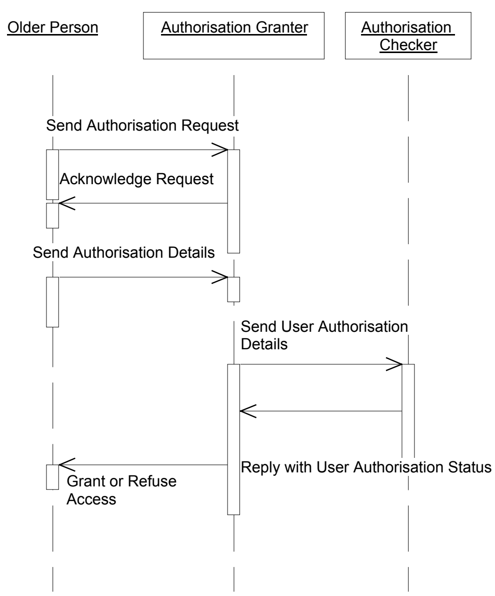

Figure 10 Agent Interaction Diagram for the Authorisation Procedure

6 CONCLUSIONS AND FUTURE WORK

This paper argues that agent technology has the potential to support the development of health and social care information systems.

However, it was pointed out the need for a methodology specifically tailored to the analysis and design of agent-based medical systems. As Kinny et al argue "If multi-agent systems are to become widely accepted as a basis for large scale applications (such as health and social care information systems), adequate agent-oriented methodologies and modelling techniques will be essential [11]". This is not only to ensure that systems are reliable, maintainable, and conformant, but to allow their design, implementation, and maintenance to be carried out by software analysts and engineers rather than researchers.

We have presented in this paper how *Tropos* can be used in the analysis and design of the electronic Single Assessment Process (eSAP), an electronic system to deliver the integrated assessment of health and social care needs of older people. One of the main difficulties in analysing the eSAP is the fact that there is not a similar system, either computerised or manual, in existence. Because of that, the capture of the requirements and also the roles (of the humans and the agents of the system) is a difficult task. *Tropos* with early requirements analysis, allows us to make explicit the reasons (why) beyond the system requirements and then to decide which is the best solution.

In our case study we tried to analyse the system inside its environment taking into consideration some security attributes. We considered security in terms of actor dependences, goals and soft goals, and we assigned capabilities to the agents of the system in order to achieve the security goals and soft goals of the system.

Nevertheless this work is by no means complete. Many open issues need to be addressed and we are working towards this. The need to introduce concepts and notation in the *Tropos* methodology in order to capture some security aspects of the system-to-be is essential. Especially for integrated health and social care systems, security is a major concern. Thus a complete and mature analysis and design methodology must provide means to capture security features.

Another important point for future work is the support of mobile agents. In an integrated health and social care system, the adoption of mobile agents may be an effective choice. Extensions must take place in *Tropos* in order to be able to capture the concept of mobile agents. Trying to give a solution to this problem, questions must be answered as to why/when/where and how an agent moves from one platform to another.

By doing that, we hope that agent technology will advance and it will be easier to use in the development of health and social care information systems providing all the advantages that are described in this paper, such as problem-solving capabilities, information sharing, encapsulation of expertise, and even more [12, 13, 14].

ACKNOWLEDGEMENTS

The first Author would like to thank the RANK Foundation for the funding of his research project, during which this work was carried out.

REFERENCES

[1] N. R. Jennings, "An agent-based approach for building complex software systems", Communications of the ACM, Vol. 44, No 4, April 2001

[2] M. Wooldridge, P.Ciancarini, "Agent-Oriented Software Engineering: The State of the Art" In P. Ciancarini and M. Wooldridge, editors, Agent-Oriented Software Engineering. Springer-Verlag Lecture Notes in AI Volume 1957, January 2001

[3] R. Evans (editor), P. Kearney, J. Stark, G. Caire, F. J. Carijo, J. J. Gomez Sanz, J. Pavon, F. Leal, P. Chainho, P. Massonet, "MESSAGE: Methodology for Engineering Systems of Software Agents", AgentLink Publication, September 2001

[4] H.Mouratidis, G.Manson, I.Philp " Testing the suitability and the limitations of agent technology to support integrated assessment of health and social care needs of older people", (to appear) 15th International Conference on Computer Based Medical Systems, Maribor, Slovenia, June 2002

[5] I. Philp. Can a medical and social assessment be combined?. Journal of the Royal Society of Medicine, 90(32), pp 11-13,1997.

[6] M. Wooldridge, N. R. Jennings, "Software Engineering with Agents: Pitfalls and Pratfalls", IEEE Internet Computing, May / June 1999

[7] M. Wooldridge, "Agent-Based Software Engineering", IEE Proceedings on Software Engineering, Vol. 144, No 1, pp 26-37, February 1997

[8] E. Yu. Modelling *Strategic Relationships for Process Reengineering*, Ph.D. thesis, Department of Computer Science, University of Toronto, Canada, 1995.

[9] B. Bauer, J. Müller, J. Odell, Agent UML: A Formalism for Specifying Multiagent Interaction. In *Agent-Oriented Software Engineering*, Paolo Ciancarini and Michael Wooldridge eds., Springer, Berlin, pp. 91-103, 2001.

[10] J. Odell and C. Bock. Suggested UML extensions for agents. Technical report, OMG, December 1999. Submitted to the OMG's Analysis and Design Task Force in response to the Request for Information enti-tled "UML2.0 RFI".

[11] D. Kinny, M. Georgeff, A. Rao, "A Methodology and Modelling Technique for Systems of BDI Agents", Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Lecture Notes in AI, Springer- Verlag, Vol. 1038, 1996

[12] J. Huang, N. R. Jennings, and J. Fox, "An Agent Architecture for Distributed Medical Care", Intelligent Agents, M. J. Wooldridge and N.R. Jennings (eds), Lecture Notes in Artificial Intelligence, Springer Verlag, 219-232, 1995

[13] J. Huang, N.R. Jennings and J.Fox, " An Agent-Based Approach to Health Care Management", Applied Artificial Intelligence: An International Journal, Taylor & Francis London, 9 (4) 1995, 401-420.

[14] A. Moreno, A. Valls, J. Bocio, "Management of hospital teams for organ transplants using multi-agent systems", 8th European Conference on Artificial Intelligence in Medicine, Portugal, July 2001