Community care system design and development with AUML.

Wei Huang[1]
Martin D. Beer[2]
Richard Hill[2]

[1] Harrow School of Computer Science, University of Westminster
[2] School of Computing & Management Science, Sheffield Hallam University

Community Care System Design and Development with AUML

Wei Huang[1] **Martin D.Beer**[2], and **Richard Hill**[2]

Harrow School of Computer Science, University of Westminster
Northwick Park, Harrow,Greater London, HA1 3TP, United Kingdom
E-mail: huangw@wmin. ac.uk

[2]School of Computing & Management Science, Sheffield Hallam University
City Campus, Howard Street, Sheffield, S1 1WB, United Kingdom
E-mail:{m.beer, r.hill}@shu.ac.uk

Abstract

An approach to the development of an appropriate agent environment is described in which software researchers collaborate with environment builders to enhance the levels of cooperation and support provided within an integrated agent–oriented community system. Agent-oriented Unified Modelling Language (AUML) is a practical approach to the analysis, design, implementation and management of such a software agent system, whilst providing the power and expressiveness necessary to support the specification, design and organisation of a health care service. This paper describes the background of agent-based health care and the fundamental concepts of Agent-oriented UML and outlines how this refreshing approach can be used in the analysis, design, development and organization of agent-based community health care systems. Our approach to building agent–oriented software development solutions emphasizes the importance of AUML as a fundamental initial step in producing agent–based architectures and applications. This approach aims to present an effective schedule and methodology for an agent software development process, by addressing the complex agent environments decomposition, abstraction, organization and software development process activities characteristics, whilst reducing the complexity of the complex agent systems' design and development by using and exploiting AUML's productivity potential.

Keywords: Agent Technology, Intelligent agents, agent-oriented Unified Modelling Language (AUML), agent-based community health care system, INCA system.

1. Introduction

1.1 Agent–oriented Community Health Care

In recent years, the providers of public and private sector health care services have been faced with some radical changes in the society they serve (Peckham et. al., 1996), and more importantly, development in the way that traditional health care is delivered to Information Technology (I.T) based communities.

It is widely accepted that better health care results from improved health care services management. This paper considers why agent technology might be appropriate for the community care domain and why it contributes to the community care service field through reorganization of the health and social care services.

The use of multi-agent systems in this field has been prompted by several evolutionary influences:

- The desire to provide humane and effective care systems that deliver improved services to people, enhancing social interaction and enabling more effective delivery of health care.
- The ability to define high-abstraction level care management strategies by linking the health care professionals into a single framework of accountability. There are many different people or organizations (medical teams, social service, etc.) that are normally involved with different priorities and skills. It is necessary to have a lucid method to allow them to communicate, negotiate, collaborate and cooperate to achieve their common domain goals.
- The development of an in depth understanding of health information (both the data and its sources), to provide better community medicine, which includes a wide range of the related social services that are often currently neglected.
- Rule-based organisational methods allow people to operate with full managerial responsibilities. E.g. established practitioners–community health doctors, senior nurses, medical services planners and health advisors etc.
- Agent-base care expert systems can assist medical professionals in the tasks of monitoring, detection and diagnosis.
- Cooperative agents can work within community structures to change the human service programme or social service policies, by involvement in planning, scheduling and organizing (both for formal care and informal care services), in a more effective and timely manner.
- The concept of a shared multi-agent environment permits shared supervision and greater teamwork by integrating specialization experts from a number of groups into common service environments. This will allow patients to visit experts, doctors or care organizations by connecting to medical agents though an agent-based community care network

1.2 Agent–based services in a Community Care Services Environment

Community health care is dynamic, complex and progressive. Its aim is to provide such services as are necessary to maintain a client's quality of life in the community when they are unable to provide for themselves, thus maintaining their independence. It is also provided by a wide range of disparate, independent organizations and agencies, typically each having their own objectives, of which the provision of community care is only a part. The objective is to integrate this service with their other responsibilities in a coherent and efficient manner (Beer et al., 2002). Agent-based community care services are a new approach that automates the process of linking constituents with their core competencies quickly and effectively on the Internet. Important aspects of this development include the need to:

- Use agent–oriented architectures, models and methodologies to create flexible roles,
- Reach new platforms and,
- Dynamically change relationships among agents.

Huang et. al. (1995) introduced an agent–based distributed medical care system which allowed patients' own needs to guide their individualized care management. The community care service agents are able to not only support traditional services but also provide a range of utility services such as individual care planning, health service advice etc. Normally, this kind of medical care service system contains a number of complex mission-critical applications that involve a few types of agents as entities or components working in a common agent environment. In the real world, individual care providers use their own heterogeneous databases, workflows and command and control systems with little or no integration between them (Beer and Huang, 2000). This causes difficulties, not only in the provision of the most effective response to emergencies, but also in the management of routine care as the client's requirements change, often quite rapidly. Current systems do not provide the flexibility to allow such changes to be implemented as rapidly as one would desire. Under the current system an Individual Care Plan is delivered by the Social Services Department of the Local Authority. This Department is normally responsible for preparing detailed specifications and services in each case. It then contracts various agencies to actually deliver the various components of the care as appropriate. There is therefore no single agency with the overall authority to plan, manage, deliver and monitor the provision of community care. To this must be added the various health care services, and the emergency services, each of which have their own independent records and command and control structures which are essential components in the delivery of the total package (Beer et al. 2001). Similarly, there are the large numbers of informal carers (family, friends, neighbours etc.) who are currently almost totally ignored by the system, but who also provide invaluable support. Researchers in the Department of Computer Science at The University of Liverpool have created a prototype e-medical service system, called INCA.

It aims to improve:

- Current community care systems by using agent-oriented engineering solutions to design a new cooperative, coordinated, collaborative health care e-service by the adoption of agent-oriented models, platforms and methodologies.
- Coordination between social services and medical care services, organizing and managing these concurrent actions effectively and interpretatively.
- The provision of positive assistance to maintain and enhance the quality of service and the provision of routine care as specified by the Individual Care Plan.

INCA is intimately concerned with agents' cooperation, communication, organization and interaction, factors that represent an effective, co-ordinated community care system and its associated information systems without loss of autonomy or security. INCA provides a complex community care service by using Agent technology solutions to organize and allot limited resources and services to a large number of care requirements. From the released work, it became clear that to maximise the effectiveness of agent–oriented community services activities, it would be prudent to have clear methods and models for analyzing, designing, organizing, controlling and managing agents' cooperation, communication and interaction.

1. Agent-oriented UML and its Concepts

AUML (Parunak & Odell, 2001) provides the scene of new meaning, the scene of new design purposes and the scene of new way for thinking of agents and its' environment. Unified Modelling Language (UML) is a de facto industry standard established by (Booch, Rumbaugh and Jacobson, 1997). Object-oriented modelling language presents a form of notation for the object-oriented paradigm analysis and design; it provides system architects working on objects analysis and design with one consistent language for specifying, visualizing, constructing and documenting the artefacts of software systems, as well as for business modelling (OMG, 1999). UML is a meta-model that represents a collection of semantic models including static models, dynamic models, usage models, and architectural models. All the models can be constructed, viewed, developed and even evaluated during the time of systems analysis and design. it provides A semantics package is provided, which allow UML model elements and notations to work independently within different packages for different processing. It is illustrated in Figure 1 (OMG, 1999).

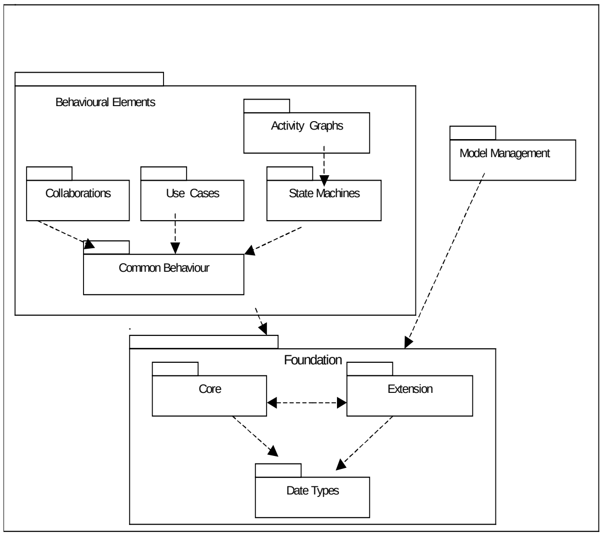


Figure 1. UML: Package Structure (OMG, 1999)

This aspect of the software development approach (e.g. representational formalism) leads us to focus on using UML notations to represent Agent concepts (e.g AUML), redesigning and reorganizing the existing Community Care & Alarm Services systems.

The use of Agent-oriented UML provides a formal platform for analyzing, designing, organizing, managing agent behaviours and modelling an agent-based system. In order to fully exploit the flexible behaviours and productivity potential of agents, AUML was selected to assist INCA's process for agent system design and development.

3. INCA System Design and Development with AUML

After considering agent concepts at an abstract level, the next step was to develop a practical design and implementation for INCA (Intelligent Community support for the elderly Architecture) systems through the use of agent-oriented Unified Modelling Language (AUML) .

First, it was necessary was to analyse all the needs of an agent–oriented community care environment and analyse the requirements with the agent models. This stage would identify the agent-based roles from all of the available care service resources (including description and identification of the agent tasks, health services and components). In practice, community care agents in INCA system could be as described in Table 1.

NAME	DESCRIPTION
Care Coordinator	The agency responsible for providing the range of services necessary to ensure that the Client is properly cared for. The Care co-ordinator is responsible for preparing a Care plan and for monitoring its effectiveness in meeting the needs of the Client. This is often the Local Authority or some other official body with a legal duty to provide the necessary care.
Care Provider	The various agencies and individuals responsible for providing the care specified by the Individual Care Plan. This will include Social Workers, Health Care Professionals, Care Assistants, Emergency Services, and Social Services etc. who can provide an extremely wide range of care services, if required
Informal Carer	The various relatives, friends, neighbours etc. who provide some form of support and assistance in an informal way (i.e. outside the Individual Care Plan), but it is often essential to allow the Client to remain resident at home. This is often flexible and responsive and can range from totally unstructured and so not recognised at all in the Individual Care Plan through to fully recognised and integrated with the efforts of the professional carers.
Client	The person who lives in their own dwelling (either an ordinary house or a sheltered home) and who receives a package of community care services. This package may range from very minimal interventions, such as social alarm systems, through to an intensive mix of community support services.
Emergency Services	This agent provides appropriate assistances and services during the emergency events. This is often flexible and responsive and through to fully recognised and integrated with the efforts of the professional emergency supports such as Ambulance, Hospital, Fire Service Department etc.

Table 1: The sample of community care agents in INCA system (Beer et. al., 2000)

During the INCA System development, agent-based UML semantics and notations are able to adequately address the overall architectural complexity across agent domains by using a set of UML graphical diagrams, such as use-case diagrams, class diagrams, agent behaviour diagrams and implementation diagrams. The sample diagram (Figure 2) is a notation diagram of Agent-based health care Service Supplying, which shows how a series of co-operational agents gathering domain operational information achieve their tasks and domain goals. The model begins with roles/agent initiation and definition, and moves to the roles 'planning for supplying emergency service' and organising agent behaviours, relations and responsibilities, followed by loading some particular class requirements. Finally, this model provides a small autonomous agent domain to generate operations such as an executable service UML package to users.

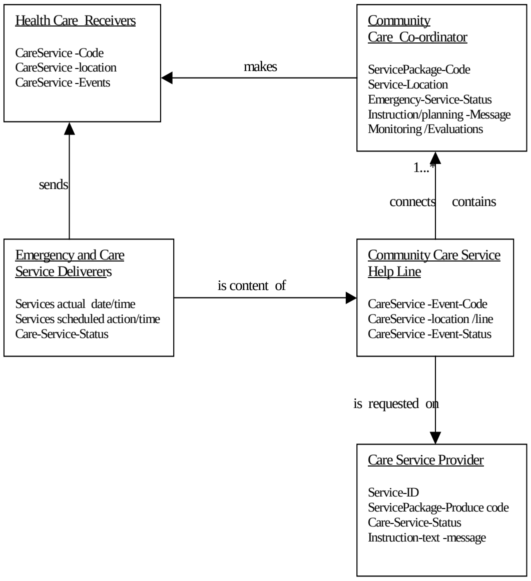


Figure 2. AUML real-time notation diagram of Community Care Services Supply

AUML is much more than a program language and modelling design diagram. It is concerned with agent domain architecture modelling and it provides a range of declarative models. Functionally, UML is a key ingredient in generating the agent architecture and informs the agent system detailed development processes. It provides the interaction and design platforms by defining all variant and invariant roles of the agents including their attributes, pre- and post-conditions for operations.

Agent-oriented UML is a modelling platform that describes the formal specification of the function, structure and behaviour of agents and multi-agent system. It has three main categories for describing system artefacts and information; static, dynamic and descriptive (OMG, 1999). Each category solves a type of domain problem. The classification can be shown as follows:

Category	Artifacts
Static	class, package, components and diagrams
Dynamic	Use-cases, interaction (sequence, collaboration, state chats, activities diagrams
Descriptive	Class descriptions

Table 2. UML artefact categories (OMG, 1999)

An agent-oriented UML model represents the static category by describing agent requirements in detail such as classes, packages, diagrams and components. The dynamic artefacts describe the communication among the components inside a shared common agent system. Descriptive artefacts are used with the other diagrams to support the description of information represented in different perspectives. The diagram below is an AUML Use Case diagram of Community Care Service Scheduling as shown in Figure 3, represents the technical roles and routing of community service supply. It allows knowledge-level agents to be linked efficiently and to share and exchange resources in a common communicated agent architecture and to achieve the "Rational Effect" domain goal.

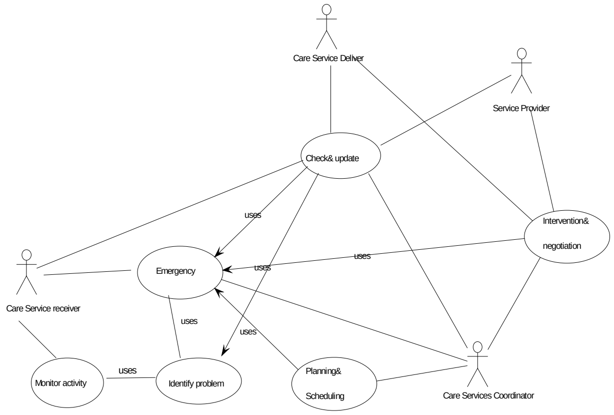


Figure 3. AUML: Use Case diagram of Care Services Scheduling in INCA system

The next stage is modelling, which involves gathering a range of state of the art techniques and models for communication, interaction, coordination and control. It can be divided as the *interaction model* and *acquaintance model* (communication, conversation policy, appropriated protocols and negotiation rules among agents). This model represents the technical roles that allow knowledge-level agents to be linked efficiently and to share and exchange resources in common agent architecture. Such interactive community care supply routing can be seen as seen in Figure 4. This is an example of how agents can provide effective routing of care requests between both official and unofficial care providers/agents (such as *Care Provider*, *Inform Care*, *Care Co-ordinator*) based on an individual care plan through negotiation.

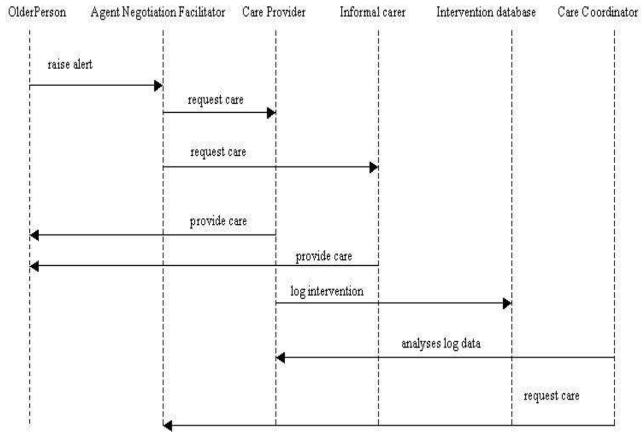


Figure 4. AUML: interaction diagram of an individual care plan and communication routing for older person

Once the individual care plan has been established, care services required to assist the Client must be managed and controlled by the agent system. Such interaction demands communication messages to be passed and routed as shown in Figure 5
.

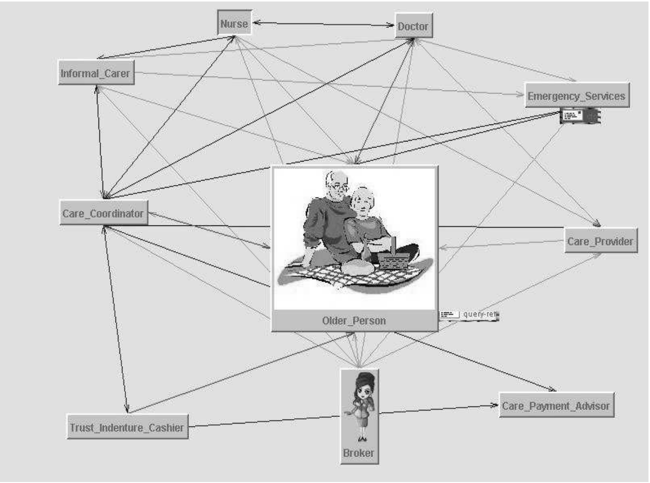


Figure 5: Community care service message passing and interactive routing

Furthermore, the main contribution of AUML to INCA work is the attempt to make the design and development of a multi-agent system as simple as possible. It greatly simplifies the configuration and organisation of the agent community, and the endowment of communication and the cooperation capabilities to the various agents. This is achieved by mimicking real-world solutions to current care service problems (Figure 6).

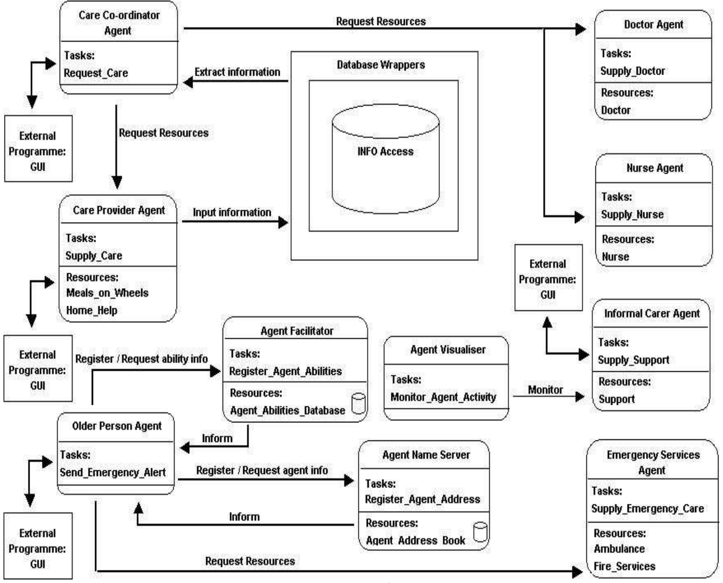


Figure 6. The Overall Architecture of the INCA Prototype

4: The Benefits of Using Agents for INCA System's Design and Development.

The integration of existing care systems facilitates the integration of a wide range of existing databases; knowledge bases and control system programmes. Agent technologies enable a 'controlled' degree of co-operative activity between these independent components. They are 'controlled' in the sense that each component within the system be it an organization or information store, needs only to communicate the information that the agent is prepared to divulge. There should be no requirement to provide additional information required for other purposes. Therefore, taking an agent-based approach it is possible for the organizations involved in the provision of care to communicate effectively yet securely, plus have access to much needed information. The compound effect leads to a more managed and effective approach to care provision, giving an increased speed of service and the elimination of unnecessary duplication of resources.

The ability of agent architectures to be distributed across heterogeneous environments has enabled the development of the INCA project's care system integration methodology. In the past a barrier to co-ordinated community care has been the incompatibility of the independent agencies systems and procedures. Agent-based methodologies like INCA enable heterogeneous systems to function together effectively in many different configurations. Thus agent technologies provide the means to integrate the different health care agency systems regardless of organizational structure and provide the means for the various agencies to interface on a common domain level.

Within the scope of the term 'agent technologies' falls a wide range of different agent typologies; for example, agents may be information agents, reactive agents or collaborative agents. In the context of the INCA architecture, the agents within the system are collaborative agents. The collaborative nature of an agent places an emphasis on autonomy and co-operation with other agents.

Autonomy means that agents have the ability to act without the need for human invention and guidance. They are able to achieve goals on behalf of the user and the system in a pro-active manner rather than a re-active manner. In other words, agents are able to take the initiative.

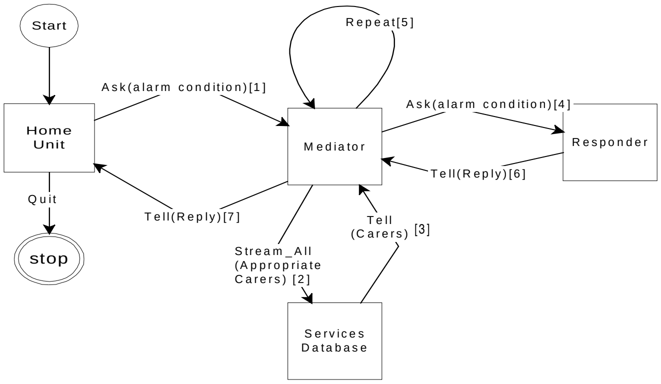


Figure 7. The Basic response Conversational Class

Co-operation means that agents are able to co-operate with other agents within the architecture in order to achieve domain goals.

The rationale for constructing an architecture of collaborative agents is that numerous agents can provide solutions to inherently distributed problems. Collaborative agents negotiate with the other agents within the system to organize and accomplish complex goals and tasks that are often beyond the capabilities of a single agent.

It can be seen that the INCA architecture enhances the co-ordination between various health care agencies by enabling the agents within the system, controlled access to the information sources incorporated within the INCA domain. That is, it is essential that the agents within the INCA architecture can interface with the existing agency information sources, including sources such as databases, log files and knowledge bases.

In order to facilitate effective co-ordination, it is necessary to allow the agent within the architecture to read and write domain information to and from the appropriate organisational information source. This can be achieved by the use of 'data wrappers', which provide an interface between agent and information source. Data wrappers enable the agents to query (read) information, by translating agent requests constructed in the communication protocol of the INCA domain into a form that is understood by the agency's own information system. For example an agent request may be translated into a SQL statement that can be executed against an organisational database. In the same manner, the responses (provided by said information sources) to agent requests are translated or mapped back into the agent communication protocol of the INCA domain.

5: The benefits of using AUML for INCA system's design and development.

A good agent model is essential for communication and interaction among agents and to assure architectural thoroughness. AUML covers a lot of essential disciplines such as system modelling, requirements and source management, design and analysis techniques, coding and fixing, programming and testing etc. It is intimately concerned with a range of domain models and deals with autonomous agent cooperation, communication, organization and interaction, in gathering domain information and factors that represent an effective collaborative and co-ordinated agent environment. It has been the preferred choice to assist INCA expanding the process and to build flexible agent architecture. The reason is that, both in theory and practice, AUML can provide a set of guidelines and methods for system design processing, agent cooperative setting and agent domain modelling. Modelling focuses on providing a formal platform for analysing, designing, organising, controlling and managing agent activities and behaviours.

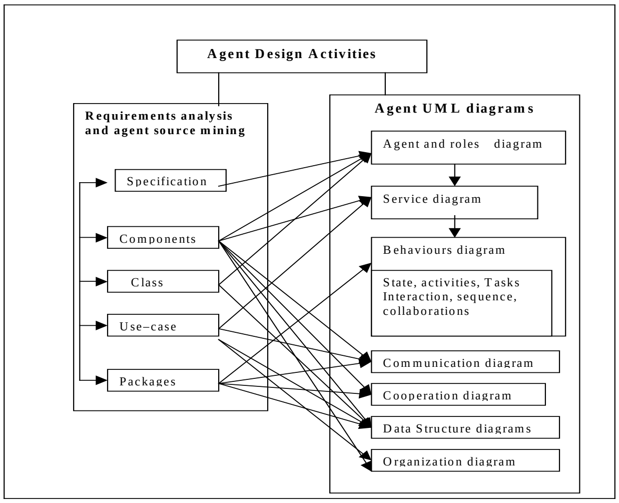
Through studies of the INCA agent development process we have found that the utilization of AUML-based design techniques is a step in the right direction for the future development of INCA. Ideally, we have found that AUML uses decomposition, abstraction, organization and development process activities characteristics, which are the attitudes for reducing the complexity of development software. It can be considered as follows:

- **Agent–oriented system decomposition:** This decomposes a system into small parts of objects, models, use-case or class, various operational actions. The structure of UML is described, together with information account of its semantics and the relationship between design notations and code (Priestley, 2000). As UML have rich design diagrams which present a view to understand the requirements of the system and a view to capture the problem space and the salutation space, it allows the developer to expose the system as a range of graphical objects/components and addresses the physical realisation of the system. System models/components can be linked and communicated. A model indicates that some kind of run-time links will exist between two instances of class shown on the same object/model. The object/model can be described or represented as classes or package diagrams and act individually, then passed to the relative source code for writing a program.

- **Agent–oriented abstraction:** a semantic level of abstraction for designing, controlling and modelling a particular system. It provides a specialized abstract view of modelling (class, use-case, diagram, interface etc.) It is used to create a set of semantics and conditions for operation and infrastructure services. It focuses on the state of activities and shows the flow from one activity to another and results in some action space. It is normally divided into a few abstract models of the program structure for producing system components, for instance, a use case view normally analyses the system requirements and resources, a process view and implementation view address the physical realisation of the system, and the deployment view focuses on system representation.

- **Agent–oriented organization:** this defines a range of elements and notations as a requirements specification for domain modelling. It aims to provide a model and an internal architecture of an agent system. It usually offers some frameworks (class, diagram, interface etc.) to show how agents can be constructed in an agent system. The modelling focuses on one aspect at a time and increases the ability to understand complex problem issues during the time of system design.

- **Agent–oriented system development process and activities:** the development process can be thought of as gathering and scheduling all the necessary methods to produce agent system components and their related program packages or source code. All of these processes aspire to collect the required resources for the identifying agent/object/components and its related case, state and activities. A technique to handle models for large systems in agent-oriented UML is providing packages; each package contains several depending objects. The purpose is to provide a practical, organizational framework for system development. The whole process of agent-base UML design activities are shown diagrammatically in Figure8



In summary, the use of AUML and its practical agent design and management platforms and methods has numerous positive characteristics as follows:

- **Inferential capability**: the ability to act on abstract roles and task specifications from perception models to designable models. It is able to provide the instruction and design platforms by defining all variant and invariant roles of the agents, describing the agent attributes and dynamically organizing conditions for operations.
- **Adaptable structure:** its modelling and structure are terse, the system is easy to operate and reconfigurable to real–end user requirements, enabling redesign and promoting model, component and task re-use.
- **Affordable communication:** built-in agent convention models, such as interaction, state diagrams exist for agent interconnections and interactions.
- **System design scalable:** it offers a set of powerful functional methods and standard design models to expand the support capability whilst providing proactive agent services and high knowledge-level systems management with a highly–flexible agent architecture.

6. Conclusions

Agent-oriented community care, such as INCA, unveils a tremendous range of challenges and opportunities to create more advanced distributed e-service systems by fully using agent capabilities and responsibilities, allowing us to open the door to new service styles which lead to reduced costs, improved communications and affect the way we live, work and do business.

From the "research and development" point of view, agent-oriented computing is a rapidly developing area and it is likely to create a lot of agent-based environments through its software devolvement activities. This paper shows that using AUML based modelling techniques can assist the development of agent–based systems and environments to further incorporate system creation within an agent domain. It can be used to help for analyzing, designing, producing and completing a complex agent-base software system.

Today, as we are in the early stage of agent research, the agent–oriented systems for software engineering development require change not only in terms of modifications and architectures, but also in terms of effective models and mechanisms of designing, organizing, modelling and communicating. Therefore, in order to design a complex agent environment, agent-oriented software engineering urgently needs more useful analysis and design models and methodologies for agent environments development and design processing. This could greatly simplify the configuration and organization of agents and the endowment of communication capability as well as cooperation capability of agents, in this way the agents and domain activities can be fully presented.

The exploration of a real pilot e-service system (e.g. INCA) and its modelling mechanism and techniques for agent-based computing is not only allowing us to find effective ways of understanding and organizing agent environments, but it also gives us a good opportunity to discover in-depth some useful methods of designing collaborative agent-oriented architectures and practical distributed information system applications, so that we can face the challenges and pressures of ever greater complexity of complex software systems' creation and development. As a result of this, an advanced distributed community care systems and their open environment will be developed and implemented very soon.

Acknowledgements

We would like to express our gratitude to all the participant researchers of INCA project for their kind help and support. Also thanks are due to Prof. Mike Wooldridge, Dr Peter McBurney and Dr Trevor Bench-Capon for their comments and assistance.

References

[1] Boooch, G., " *Object–Oriented Analysis and Design with Applications," 2nd Ed., pp*Addison-Wesley,pp18-26 1999.

[2] Beer, M.D, & Huang, W.., *"Using Agents to Build a Practical Implementation of the INCA (Intelligent Community Alarm) System"*, in conference proceedings UKMAS 2000, Oxford University, UK. December 2000 and Agents 2001, Montreal, Canada, June 2001.

[3] Beer, M.D, Huang, W. & Sixsmith, A, in book "*Intelligent Agent and their Applications"* Chapter10 (pp307-354), Springer-Verlag, Heidelberg, March 2002.

[4] Beer, M.D, Hill R, Huang.W, "Using Agents to Promote Effective Co-ordination in a Community Care Environment" in name of book "Agent Supported Collaborative Work" Kluwer, Netherlands, forthcoming winter 2002

[5] Cancarini,P. and Wooldridge,M.J.,(Eds),"*Agent-oriented Engineering"*, Springer-Verlag Lecture Notes in AI Volume 1957, January 2001.

[6] Huang.W, "An introduction to Agent-based Emergency Assistance Service System (AEAS)" paper in conference proceeding PREP2002, The University of Nottingham,UK, April 2002

[7] Object Management Group *"OMG Unified Modelling Language specification version 1.3"*, June 1999.

[8] Object Management Group website: http://www.omg.org/

[9] Priestley. M *"Practical Object-Oriented Design with UML"* McGraw-Hill Companies, Inc, 2000.

[10] Wooldridge, M.J, Jennings. & Kinny, D. "*The Gaia Methodology for Agent-Oriented Analysis and Design*" In Journal of *Autonomous Agents and Multi-Agent Systems*, 3(3): 285-312., 2000

[11] Van Dyke Parunak, H. and Odell, J. (2002), "*Representing Social Structures in UML*," *Agent-Oriented Software Engineering Workshop II*, Michael Wooldridge, Paolo Ciancarini, and Gerhard Weiss, eds., Springer, Berlin, 2002, pp. 1-16. (Lecture Notes on Computer Science - LNCS 2222; from the Second International Workshop, AOSE 2001, Montreal, Canada, May 2001, held at the Agents 2001 conference, Montreal, Canada