Design and Modelling of the Nocturnal AAL Care System

J. C. Augusto[1], H. Zheng[1], M. Mulvenna[1], H. Wang[1], W. Carswell[1], P. Jeffers[2]

Abstract. We present the modelling of a monitoring system which provides night-time care by detecting situations of concern and therapeutic interventions as the core technological component within an Ambient Assisted Living project. The modelling of processes and interactions allows early detection of problems in the strategy to be implemented through simulation and verification.

Keywords: Intelligent Environments, AAL, safety critical.

1. Introduction

Dementia is a group of symptoms caused by specific brain disorder. The 2003 World Health Report Global Burden of Disease [1] estimated that dementia contributed to 11.2% of all years lived with disability among people aged 60 and over. In the UK, the Dementia 2010 report revealed that it is over 800,000 people sufferers a form of dementia and the annual cost of dementia is £23bn. It is estimated that the number of sufferers will pass the one million mark before 2025 [2]. In 2009, the National Dementia Strategy was published with the aim to support people with dementia and their carers to live well with dementia. Telecare and assistive technology spans a number of objectives in the national strategy [3].

Generally, people with dementia exhibit the symptoms of memory loss, problems using language, changes in personality, disorientation, problems doing usual daily activities and disruptive or inappropriate behaviour. Wandering and incontinence are the top two causes of institutionalisation. This paper presents our work

[1] University of Ulster (UK) [jc.augusto,h.zheng,md.mulvenna,hy.wang,w.carswell]@ulster.ac.uk

[2] Fold Housing Association (UK) Paul.Jeffers@foldgroup.co.uk

on the design and modelling of the Ambient Assisted Living (AAL) care system to support people with dementia at night time. We explain how tool-supported methodologies from Software Engineering can guide the development of the core monitoring and actuation process of a MAS system.

The reminder of this paper is organised as follows. In section 2, we introduce the NOCTURNAL project followed by a description of the design and modelling software package, SPIN, in section 3. NOCTURNAL model design, simulation and verification are detailed in section 4. The conclusion is presented in section 5.

2. The Nocturnal Project

AAL services and technologies are designed to help extend the time that older people can live at home by "increasing their autonomy and assisting them in carrying out activities of daily life" [4]. The services offered may include support for functional, activity, cognitive, intellectual and sensory-related activities; for example, providing alarms to detect dangerous situations that are a threat to the user's health and safety, continuously monitoring the health and well-being of the user and the use of interactive and virtual services to help support the user.

These technology-enriched services have evolved from relatively simple telecare services such as emergency fall alarm provision into more sophisticated telehealth services supporting people with long-term chronic health conditions such as Alzheimer's disease. Along with this evolution in the provision of services, there is a parallel development in the sophistication of the technology that underpins the AAL services and in the complexity and volume of the data that is harvested from the sensors that monitor the activity of the user in their home setting. Such data can include movement information, device usage information, medication compliance data and other rich data that can inform decisions for AAL services.

In the NOCTURNAL project, data representing activities of people with dementia is gathered and analysed in order to create behavioural profiles for them. The goal of the project is to develop a solution that supports older people with mild dementia in their homes, specifically during the hours of darkness. This is a relatively new area of research and was identified as a key area of need for care recipients with dementia after [5] found in their literature review of papers reporting on nighttime care of people with dementia that only 7% of papers addressed nighttime specific issues with a further 26% focusing on night and day activities together. It is also of interest because of the negative impact that lack of sleep and consequent anxiety causes for the informal carer in the home of the person with dementia. The support provided is also relatively unusual in that the AAL services are focused on identifying negative behaviours at nighttime such as restlessness and wakefulness [6] and then responding to these behaviours with interventions, e.g., guidance, that are designed to have a therapeutic impact. The design is for the AAL services to provide reassurance, aid and guidance for the general behaviour of the care recipient and to support a stable circadian rhythm. The types of therapeutic interventions include musical, visual and lighting based.

3. SPIN

SPIN [7] is one of the most well-known and used system for verification of software. It is a highly efficient and stable system with a user friendly interface, and good team support. SPIN is focused on the concept of models. Users can model a system by using a language called Promela (PROcess MEta-LAnguage), which emphasizes the role of processes and their interactions. Once a user has built a model the possible scenarios represented in the model can be simulated in various ways (e.g., randomly guided by the machine or user guided). Additionally users can perform what in Software Engineering is called Formal Verification, i.e., an exhaustive analysis of the computations implied by the model. SPIN provides a formal language for users to specify properties which can then be checked by the tool. If the property the user was trying to verify in the model does not hold then SPIN provides a counter-example (an explanation of why that property is not true for that particular model). The presentation below is more centred on the model and the simulation process (see [8, 9] for an emphasis on verification of IEs).

4. The Nocturnal Model

Our team is using SPIN to inform the design and modelling phases of the Nocturnal project. Some models consider the overall system whilst other models focus specific agents. The model of the overall architecture is depicted in Fig. 1.

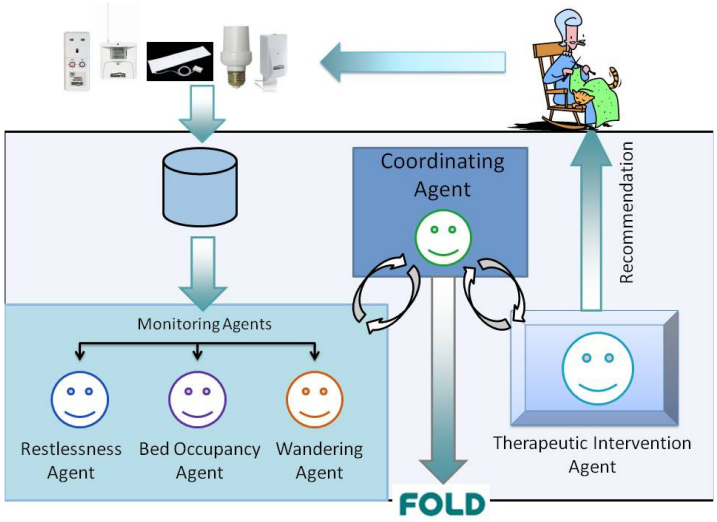

Figure 1. Nocturnal Architecture

Activities of the client trigger sensors which are recorded as events in a database. These events are fed to a group of monitoring agents specialized on night related situations (e.g., restlessness, bed occupancy and wandering). When the

number of episodes of interest detected by any single agent is above an acceptable threshold, which is dynamically adapted to the client and the context, the agent involved contacts a coordinating agent (CA) which have a holistic view of the context informed by all the single agent's reports. If appropriate, CA can trigger a therapeutic intervention with the aim of helping the client. If subsequent reports from the monitoring agents show there is still reasons for concern the coordinating agent can issue a new intervention or eventually if the situation requires it the call centre at Fold can be contacted so that a human deals directly with the situation. The system then is used as a way to increase independence with safety and to focus human interventions on those cases where is really needed.

4.1 Promela Model

Figure 1 depicted the main actors and interactions within the Nocturnal system architecture. The model was conceived to explicitly represent those elements and relationships to test the idea and use it as a framework to experiment with and discover non-trivial features which escaped the initial analysis of the team. One of the versions of the model of the overall architecture is provided in Appendix A (see comments inserted providing explanation of the model). Each main element of technology and human actors depicted in Figure 1 is represented in the Promela model by a process type (each of them has their name highlighted in boldface). Processes are autonomous entities running concurrently. Interaction amongst these elements is represented by message passing through synchronous channels. Naturally there are many features of the model that can be changed to experiment with, this model is only a snapshot in the lifetime of the system.

4.2 Simulation

This model can be used for simulation in SPIN and several different types of views can be extracted as the simulation unfolds. Figure 2 shows the Message Sequence Chart which depicts the different processes and the messages they sent each other in this specific random simulation. Yellow boxes at the top indicate the name of the processes in the system. Arrows indicate a message sent from one process to another. Fig. 2 shows a run such that when process Client activates (box number 6) a sensor (box number 7), this event is stored in the DB and passed to the monitoring agents: Restlessness (31), Bed Occupancy (10) and Wandering (37), they act according to whether is relevant or not to them. They in turn (in any possible combination, i.e., none, some or all) may or may not contact the Coordinating Agent (39). This one may (75) or many not (39) decide that a Therapeutic Intervention is needed. The therapeutic intervention may sometimes produce a response from the user (76-78). Sometimes the situation may require to contact Fold (199).

Notice this model does not focus on frequencies but rather on possibilities, i.e., whether something can be achieved or not within that architecture. Other modes

we have explored focused on different aspects of the system, for example on how the monitoring agents can effectively keep track of the frequency of restlessness episodes, detect absence from bed or wandering, during a period of time.

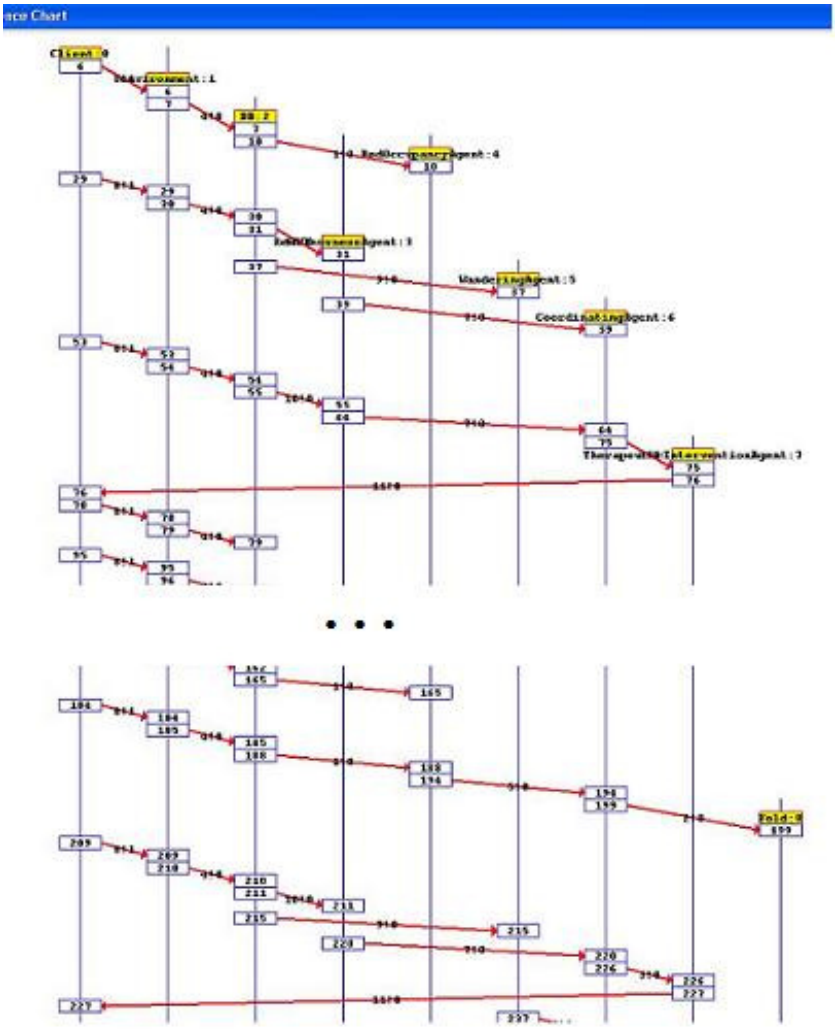

Figure 2: processes of the Overall Architecture communicate with each other

4.3 Verification

Formally specified behavioural properties (e.g., [] <> activeClient) can be explored using SPIN. These properties are usually related to the requirements of the system being examined. Examples of such properties for the Nocturnal case are:

"can the system monitor the client continuously?", "Are all sensor activations stored in the DB", "Is each emergency followed by a therapeutic intervention?".

5. Conclusions

The project Nocturnal aims at providing Ambient Assisted Living with an emphasis on night-time care. This paper has presented the underlying processes related to the project Nocturnal, and has shown how the use of well established techniques and tools from Software Engineering can be used to model these processes and to rigorously examine them during software development.

Our tool of choice provide simulation and verification capabilities. Through simulation different scenarios can be recreated either randomly by the tool or guided by the user. Once this has provided confidence to the team on the strategy being modelled other features of the tool can be used to check if the strategy represented in the model is consistent with the behavioural properties that are present in the model.

Our team has used this tool at two different levels: to model the overall architecture (the focus of this paper) and to model individual agents, e.g., restlessness agent of Fig. 1. This process allowed the team to focus on the strategy and the logic of the processes before implementation.

References

1. World Health Organization, 2003. The World Health Report 2003: Shaping the future.
2. Alzheimer's Research Trust, Dementia 2010.
3. Department of Health, Living well with dementia: A National Dementia Strategy, 2009.
4. Wojciechowski, M. & Xiong, J. (2008), "A User Interface Level Context Model for Ambient Assisted Living", Smart Homes and Health Telematics, (Online), pp. 105-112.
5. Carswell, W., McCullagh, P.J., Augusto, J.C., Martin, S., Mulvenna, M.D., Zheng, H, Wang, H.Y., Wallace, J.G., McSorley, K., Taylor, B., Jeffers, W.P., A Review of the Role of Assistive Technology for People with Dementia in the Hours of Darkness. Technology and Health Care, Volume 17, Number 4, pp. 281-304. 2009. IOS Press.
6. Wang, H, Zheng, H, Augusto, J.C., Martin, S, Mulvenna, M.D., Carswell, W, Wallace, J, Jeffers, P, Taylor, B and McSorley, K, Monitoring and Analysis of Sleep Pattern for People with Early Dementia, in Proc. of the First Workshop on Know. Eng., Discovery and Dissemination in Health, 2010.
7. Holzmann, G. J., The SPIN Model Checker : Primer and Reference Manual. Addison -Wesley, 2003.
8. Augusto, J. C.. McCullagh, P. Ambient Intelligence: Concepts and Applications. Int. Journal on Computer Science and Information Systems, 4(1):1-28, 2007.
9. Augusto J.C. Increasing Reliability in the Development of Intelligent Environments, Pro-c. of 5th Int. Conf. on Intelligent Environments (IE09), pp. 134-141. Barcelona, Spain, 2009.

Appendix A: *Nocturnal Architecture Promela Model*

```
/* Data Structures Declarations Section                                    */
   bool activeSensor= true; bool contact= true; bool takeAction= true;
   bool intervention= true; bool event = true;
  /* all channels declared synchronous, i.e., handshake guaranteed  */
   chan client2sensors  = [0] of {bool};  /* Client to Sensors */
   chan env2DB = [0] of {bool}; /*  environment to DB */
   chan db2RA = [0] of {bool};  /* DB to Restlessness Agent */
   chan db2BOA = [0] of {bool};  /* DB to Bed Occupancy Agent */
   chan db2WA  = [0] of {bool};  /* DB to Wandering Agent */
   chan ra2CA  = [0] of {bool};  /* Restlessness Agent to Coordinating Agent */
   chan boa2CA  = [0] of {bool};  /* Occupancy Agent to Coordinating Agent */
   chan wa2CA  = [0] of {bool};  /* Wandering Agent to Coordinating Agent */
   chan ca2TIA  = [0] of {bool};  /* CA to Therapeutic InterventionAgent */
   chan tia2client  = [0] of {bool};  /* Therapeutic InterventionAgent to Client */
   chan ca2fold  = [0] of {bool};  /* Emergency Notification */
/*Process Declaration Section                                             */
   active proctype Client ()  /* represents free will of human */
   { bool activeC;
   end: do
          :: tia2client?intervention -->
             if   :: client2sensors!event
                  :: skip    fi
                  :: activeC --> atomic{client2sensors!event; activeC=false}
                  :: !activeC --> activeC=true
         od }
   active proctype environment () /* generates sensor data and stores it in DB */
   { bool idle;
   end: do
          :: !idle -->atomic{client2sensors?event; env2DB!activeSensor; idle=true}
          :: idle --> idle=false
        od}
   active proctype DB () /* stores sensor data and passes it to agents*/
   { end: do
          :: env2DB?activeSensor -->
                  atomic{
                     if  :: db2RA!activeSensor
                         :: skip fi;
                     if  :: db2BOA!activeSensor
                         :: skip fi;
                     if :: db2WA!activeSensor
                        :: skip fi                        }
        od }
   active proctype RestlessnessAgent () /* detects restlessness episodes */
   { end: do
```

```
        :: db2RA?activeSensor -->
          if :: atomic{
                    printf("Restlesness Agent was interested on this information");
                    ra2CA!contact}
             :: skip   fi
        od }
active proctype BedOccupancyAgent () /* detects out of bed episodes */
{ end: do
        :: db2BOA?activeSensor -->
          if :: atomic{
                    printf("BedOcc. Agent was interested on this information");
                    boa2CA!contact}
             :: skip   fi
        od }
active proctype WanderingAgent () /* detects wandering episodes */
{ end: do
        :: db2WA?activeSensor -->
          if :: atomic{
                    printf("Wandering Agent was interested on this information");
                    wa2CA!contact}
             :: skip  fi
         od }
active proctype CoordinatingAgent ()  /* gathers advice from agents and,
when necessary, intervines in environment */
{ end: do
        :: ra2CA?contact -->
          if :: ca2TIA!takeAction
             :: ca2fold!takeAction
             :: skip fi
        :: boa2CA?contact -->
          if :: ca2TIA!takeAction
             :: ca2fold!takeAction
             :: skip fi
        :: wa2CA?contact -->
          if :: ca2TIA!takeAction
             :: ca2fold!takeAction
             :: skip fi
         od}
active proctype TherapeuticInterventionAgent () /* actuates in env. to
 achieve goals set by coordinator agent */
{ end: do :: ca2TIA?takeAction -->
                  atomic{tia2client!intervention; printf("Action Taken!")} od }
active proctype Fold () /* deals with emergencies */
{ end: do :: ca2fold?takeAction --> printf("Action Considered by Fold!") od }
```