Requirements engineering methods for an Internet of Things application: fall-detection for ambient assisted living

Sofia Meacham[1], Keith Phalp[1]

[1]Faculty of Science and Technology, Bournemouth University,
Fern Barrow, Poole, Dorset, BH12 5BB, UK
smeacham@bournemouth.ac.uk, kphalp@bournemouth.ac.uk

Abstract

In this paper,

According to the World Health Organization [1] approximately 28-35% of people aged 65 and over fall each year increasing to 32-42% for those over 70 years of age. The situation is getting worse due to the fact that elderly people often have to stay alone for long periods of time either in their own home environments or in care homes. In this context, automatic fall-detection systems can enable triggering of an alert (manual or automatic) in an emergency situation, thus enabling help when it is required, reducing deaths from falls and consequently increasing the personal feeling of security of elderly people. There are several available fall detection systems, each of which address some of the requirements, both for indoor [2] [3] and outdoor environments [4]. However, the requirements for these systems are rarely properly defined and formulated.

In this paper, an attempt to properly define the requirements for these types of systems through the use of a variety of different formalisms will be initiated.

First, Volere templates have been chosen for the initial steps. Among all the steps of the Volere requirements engineering process, the use of Volere templates is more well-known and widely adopted. The structured view of the requirements document allows for u00204(t)7(e)-3(ia-21(ly-5(e)o 0.00f)4J -4.289 0 Td [(m)13(21 u006 Tc 0 Tw (06 f14

To illustrate the approach, in Section 2, a case study will be described. In Section 3, a short description of the model-based design approach and the UML/SysML modelling languages for model-based will be initiated. In the following Section 4, the use of Volere templates is presented and the High-level/ Low-level Use Case and Requirements diagrams are presented and compared. The system design and implementation will be presented in Section 5, whereas the resulting requirements “flow” from modelling to design/implementation will be extracted in Section 6. In Section 7, reflections and evaluation of our approach will be presented. Finally, Section 8 offers some conclusions and suggestions for future research directions.

2.0 Case Study Overview

This case study was set in collaboration between Bournemouth University and the Technological Educational Institute of Western Greece (TWG), the Embedded

Last but not least, a major requirement for the future would be to add to the system "intelligent" behaviour wherever appropriate. For example, in case a fall is detected, monitoring mechanisms should be increased in order to obtain more information about the criticality of the incident and the patient's medical condition.

3.0 Model-based Design: UML/ SysML modelling

Throughout this project, Model-Based Systems Engineering (MBSE) was applied as an approach to the design and development of a number of systems. In MBSE, models take a central role, not only for analysis of these systems but also for their construction. According to INCOSE, the adoption of MBSE has several advantages [6] such as: improved communication among stakeholders, team members through diagrammatic model representations; improved quality through early identification of problems and fewer errors at the integration stage; increased productivity through reusability of existing models and reduced risk through improved estimates and on-going requirements validation and verification. Overall, it has been said to increase productivity and efficiency in the design and development mainly of comsi0 Td er ead whereas Requirements Diagrams gave a diagrammatic view that connected requirements with block diagram implementations, offering a path to traceability/verification as well as providing relationships between requirements.

4.2.1 High-level Use Case

In Fig. 1, we can see a high-level Use Case Diagram that consists of three main categories of Use Cases: Monitor Device which is used to monitor movement, location and battery levels; Manage Alert which creates an alert in case something is wrong; Manage Record which coordinates the storage and maintenance of medical information

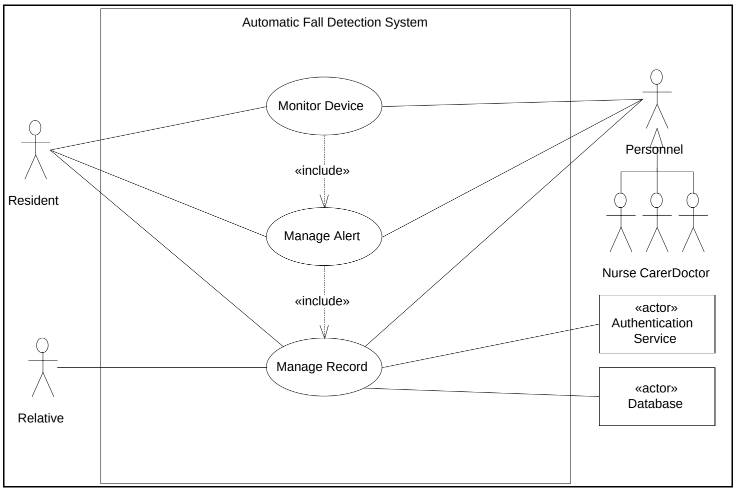
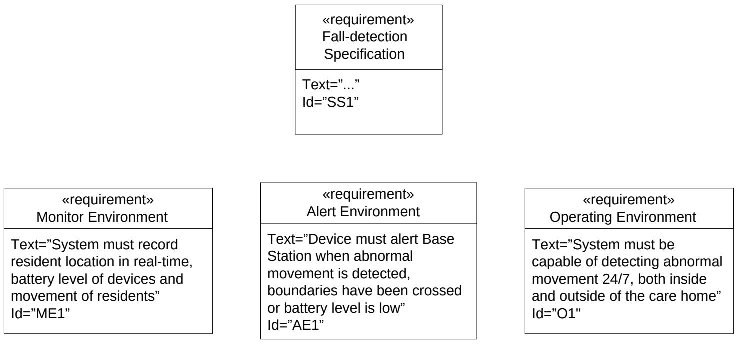


Fig. 1 High-level Use Case Diagram

In this Use Case, the system is presented from the point of view of the main actions that the actors perform (resident, relative and personnel).

4.2.2 High-level Requirements Diagram

In Fig. 2, a high-level Requirements Diagram is depicted that consists of three main categories of requirements: Monitor Environment which is used to monitor movement, location and battery levels; Alert Environment which creates an alert for five cases (manual alert, fall detection, no movement detection, out of range, low battery); Operating Environment which can be indoor, outdoor and 24/7 operating system.


«requirement»
Fall-detection Specification
Text="..."
Id="SS1"
«requirement»
Monitor Environment
Text="System must record resident location in real-time, battery level of devices and movement of residents"
Id="ME1"
«requirement»
Alert Environment
Text="Device must alert Base Station when abnormal movement is detected, boundaries have been crossed or battery level is low"
Id="AE1"
«requirement»
Operating Environment
Text="System must be capable of detecting abnormal movement 24/7, both inside and outside of the care home"
Id="O1"

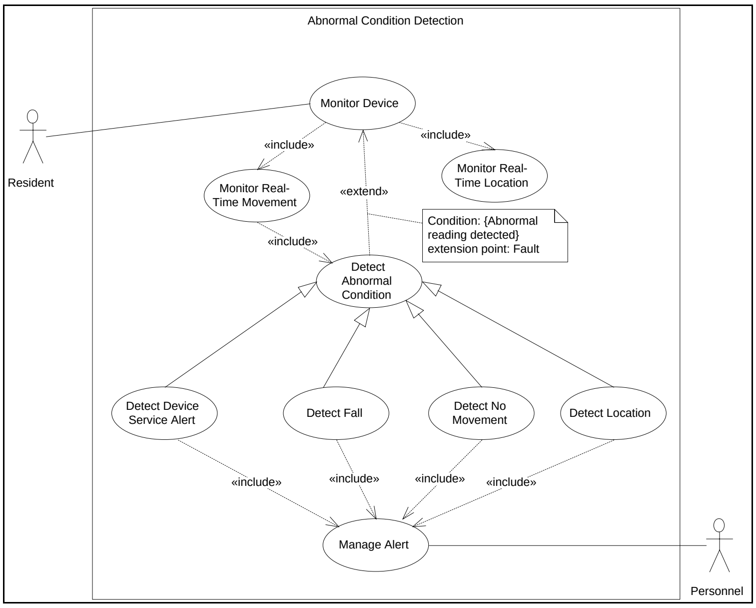


Fig. 3 Low-level Use Case Diagram

4.3.2 Low-level Requirements Diagram

In Fig. 4, a low-level Requirements Diagram is presented that describes the abnormal condition detection requirement. It consists of three main parts: movement, location and communication. The movement and the location are part of the device decision. Note that in this diagram the corresponding non-functional legal requirements are depicted that are part of the system has to follow such as Medical Device Regulation 2012 and Care Quality Commission 2009.

Fig. 4 Low-level SysML Requirements Diagram

4.2.3 Comparison of Low-level Diagrams

In the above Use Case, the system is presented from the point of view of the operations that will have to be performed. Unlike most of the Use Cases, it doe not focus on the interactions with actors but rather on the main operations, such as types of abnormal condition (fall, no movement, device send alert). On the other hand, in the above Requirements Diagram, the system is depicted from the point of view of the main structural blocks that are required by the system implementation such as defining movement, location and communication.

There is a one-to-one correspondence between some parts of the two diagrams such as the movement and location blocks. However, the two diagrams are fundamentally different in content. It is very important to note that the requirements diagram includes non-functional requirements such as legal requirements, which are not part of a Use Case diagram. This is an advantage that has been introduced by SysML and is very important for non-functional properties

5.0 System Design/implementation

In Fig. 5, the corresponding high-level SysML block diagram of the system is presented. A one-to-one correspondence between the requirements in Fig. 2 and the blocks in this diagram is apparent. In addition to this, the "flow" of events is depicted. The Monitor block monitors the system (Operating Environment block) and when something abnormal is detected it raises a Trigger to the Alert block. The le/M7 Ta Td (.675 s)-1gh

3. Dong, Q., Yang, Y., Hongjun, W. and Jian-