

ARCADE Example

An example of application of the Open Architectural Description Framework ARCADE



Erlend Stav, Ståle Walderhaug, and Babak Farshchian

Developed by SINTEF ICT



Copyright © 2013 by SINTEF

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/deed.en_US.

The following attribution must be included in all work adapted from this document:

This work is based on the “ARCADE Example: An example of application of the Open Architectural Description Framework ARCADE” by SINTEF, available from: <http://www.arcade-framework.org/>

Document information:

Title: “ARCADE Example: An example of application of the Open Architectural Description Framework ARCADE”
Version: 0.6, December 2013
Authors: Erlend Stav, Ståle Walderhaug, and Babak Farshchian
Cover picture: Babak Farshchian

Content

1 INTRODUCTION	1
1.1 ARCADE example	1
1.2 What is ARCADE?	1
1.3 Document Content	1
2 OVERVIEW	2
2.1 Viewpoints and Views	2
2.2 Stakeholders and their roles	2
2.3 Concerns	2
2.4 System Assets	2
2.5 Reference Architecture	2
2.6 Modelling Language	3
3 CONTEXT VIEW	4
3.1 Business Aspects Model	4
3.2 Environment Systems Model	6
3.3 Business to System Mapping Model	7
4 REQUIREMENT VIEW	8
4.1 Requirement Model	8
4.2 Target System Interface Model	8
5 COMPONENT VIEW	9
5.1 System Information Model	9
5.2 System Decomposition Model	10
5.3 System Collaboration Model	12
5.4 Component and Interface Specification Model	12
6 DISTRIBUTION VIEW	13

<u>6.1 System Distribution Model.....</u>	<u>13</u>
<u>6.2 Role Distribution Model.....</u>	<u>13</u>
<u>7 REALISATION VIEW.....</u>	<u>14</u>
<u>7.1 System Deployment Model.....</u>	<u>14</u>
<u>7.2 Technology Mapping Model.....</u>	<u>14</u>
<u>7.3 System Integration Test Model.....</u>	<u>14</u>
<u>BIBLIOGRAPHY.....</u>	<u>15</u>

1 Introduction

1.1 ARCADE example

This document provides an example of how to use the ARCADE architectural description framework [ARCADE 2010]. The example described in this document is from the domain of Ambient Assisted Living (AAL), and describes a system for monitoring the health condition of a patient at home using a set of sensors.

The example includes a description of the use context of the system, including which environment systems it could interact with. Further, the decomposition of the system in sub-systems and components is described.

Please note that the example provided in this document is only an imaginary system, and its design was only created in order to illustrate the use of the ARCADE framework, and not to be a complete or good architecture for this kind of application.

1.2 What is ARCADE?

ARCADE is a domain and technology independent architectural description framework for software intensive systems. ARCADE was created to assist in creating, understanding, and describing the architecture software systems.

The ARCADE framework is document in the ARCADE Handbook which is freely available from:

<http://www.arcade-framework.org/>

Beyond the illustration provided by the example, the details of the ARCADE Framework itself is not further described in this document.

1.3 Document Content

This document

2 Overview

2.1 Viewpoints and Views

The main parts of the architectural description of this example is provided in a set of views. The views follow the viewpoints defined in the ARCADE framework, and each view is presented in a separate chapter.

2.2 Stakeholders and their roles

Stakeholders are identified and described in the Context View in this example.

2.3 Concerns

To be added in a future version of this description.

2.4 System Assets

This example does not currently use any system assets such as dictionaries, standards, or patterns beyond those which ARCADE is based on. Future versions of the example may be extended with such usage.

2.5 Reference Architecture

The example uses the generic, tier-based reference architecture defined in ARCADE, which is shown in Figure 1. Details are described in the ARCADE Handbook [ARCADE 2010].

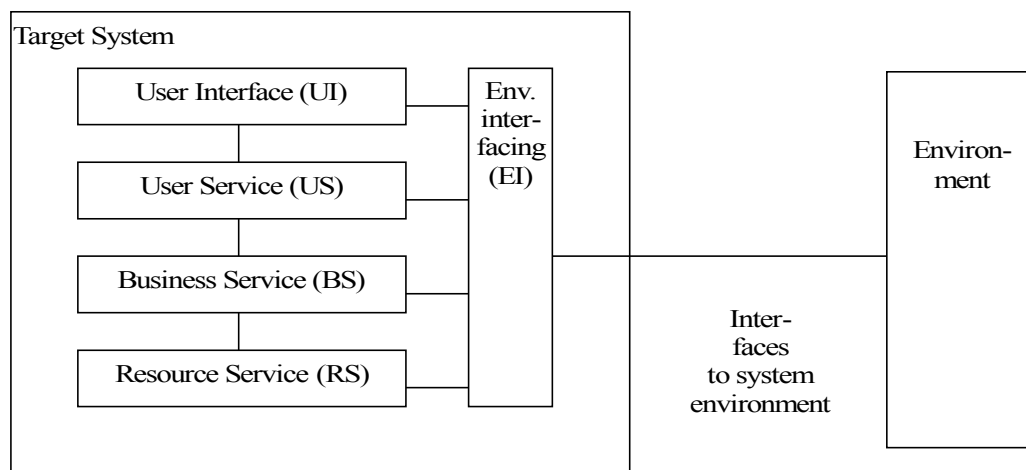


Figure 1: ARCADE generic reference architecture

2.6 Modelling Language

The models in this example are expressed using UML and supplementary free text.

3 Context View

The main purpose of the system is to enable patients to monitor health conditions at home, and to discover potential incidents based on sensor measurements. The following models describe the context in which the system will be used, including the stakeholder and processes around the system, environment systems to which the system will interface, and which processes that map to which system.

3.1 Business Aspects Model

An overview of the stakeholders for the system is shown in Figure 2. As shown in the figure, in addition to the patient, nurses and next of kin are stakeholders.

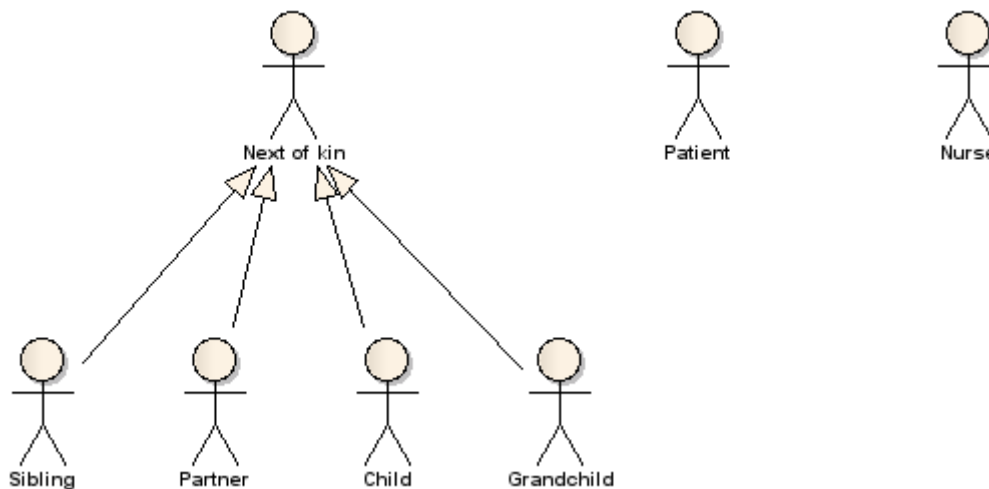


Figure 2: Stakeholders for the system

The use-case diagram in Figure 3 gives an overview of the processes each of the stakeholders is involved in. As seen from the diagram, both continuous monitoring and point measurements are needed by the patient. Also, the patient needs a way to trigger an alarm manually in case of an emergency. The patient needs to learn to use the equipment and follow up on health conditions in collaboration with the nurse. The nurse will manage a set of patients, and will regularly check the monitored data both to discover potential health problems and problems with the measurements. If an alarm goes off, the nurse will handle this – potentially in collaboration with the patient's next of kin.



Figure 3: Use cases for supporting the patients in monitoring their health condition

3.2 Environment Systems Model

In the environment of the target system, there are some other technical systems with which the target system have to interact. Firstly, the system will use existing sensors for both continuous monitoring and point measurements. Further, an existing sensor measurement storage server will be used to store the measurements, and integration is needed with an existing electronic patient record (EPR) system. All of these systems influence the implementation and operation of the target system. The environment systems are summarised in Figure 4, and the use cases the systems are involved in are shown in Figure 5.

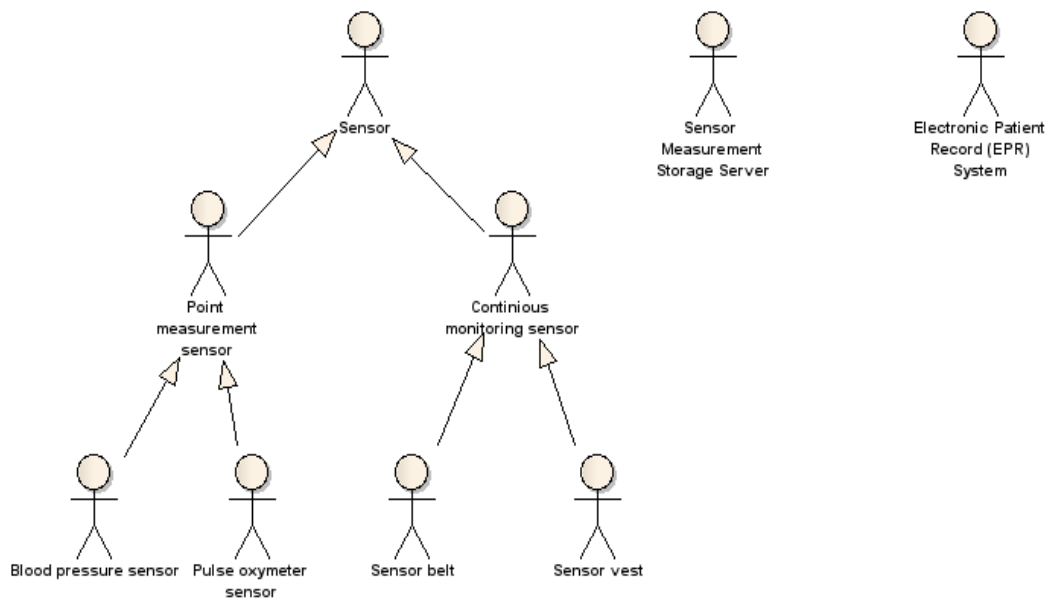


Figure 4: Overview of environment systems

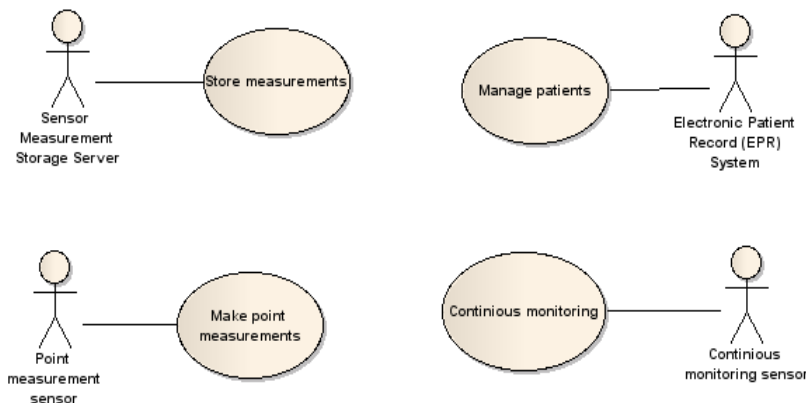


Figure 5: Use cases for environment systems

3.3 Business to System Mapping Model

Figure 6 illustrates how the use cases defined in the Business Aspects model are mapped to different systems. As shown in the figure, the sensor management storage system and the electronic patient record system will handle respectively the store measurements and manage patients. The use cases for learning to use the system and following up on health conditions are defined to be external, manual procedures that will be handled in other interaction between the nurse and the patient, and are thus not part of the target system. The remaining use cases will be implemented by the target system (shown in the box on the left side of the figure).

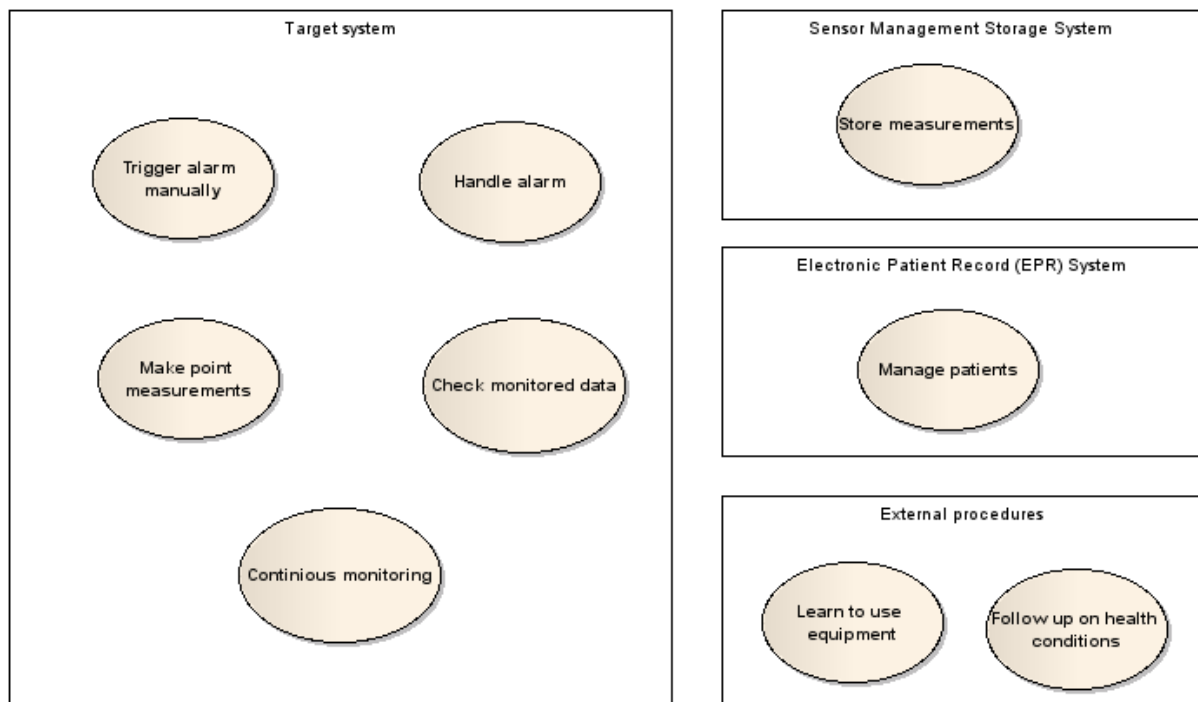


Figure 6: Mapping of the use cases to the systems in which they are implemented

4 Requirement View

This view will be described in a future release of this document.

4.1 Requirement Model

4.2 Target System Interface Model

5 Component View

The component view describes the system in terms of its subsystems and information objects, and documents how subsystem interaction and information processing is carried out.

5.1 System Information Model

The main information objects shared in our example system are defined in Figure 7 in form of a UML class diagram. Note that the Patient information is pre-defined by the Electronic Patient Record system, with which the target system will be integrated.

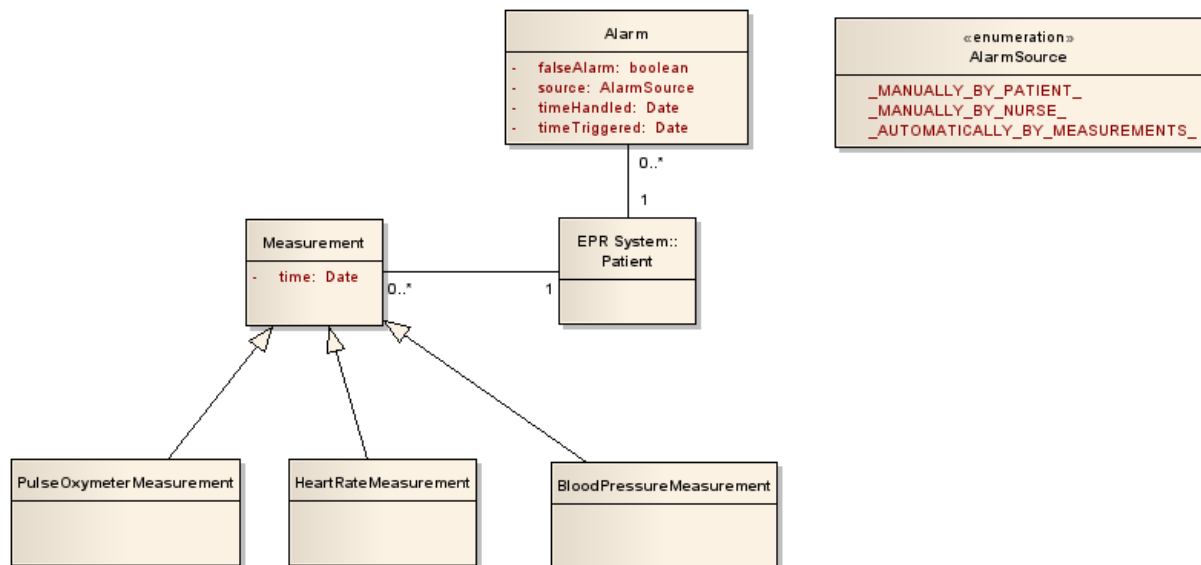


Figure 7: System information model

5.2 System Decomposition Model

The main decomposition of the target system into sub-systems is shown in the UML component diagram of Figure 8. The diagram shows each of the environment systems identified in the Environment Systems model as components (the different sensors are shown as a single component in this diagram). The target system has been divided into three subsystems: an application the user will run on a handheld device, an application used by the nurse, and an alarm central. The connection points for each component is illustrated by the ports, and the connectors and their implied direction (use vs. provide interface) show the interaction between the components.

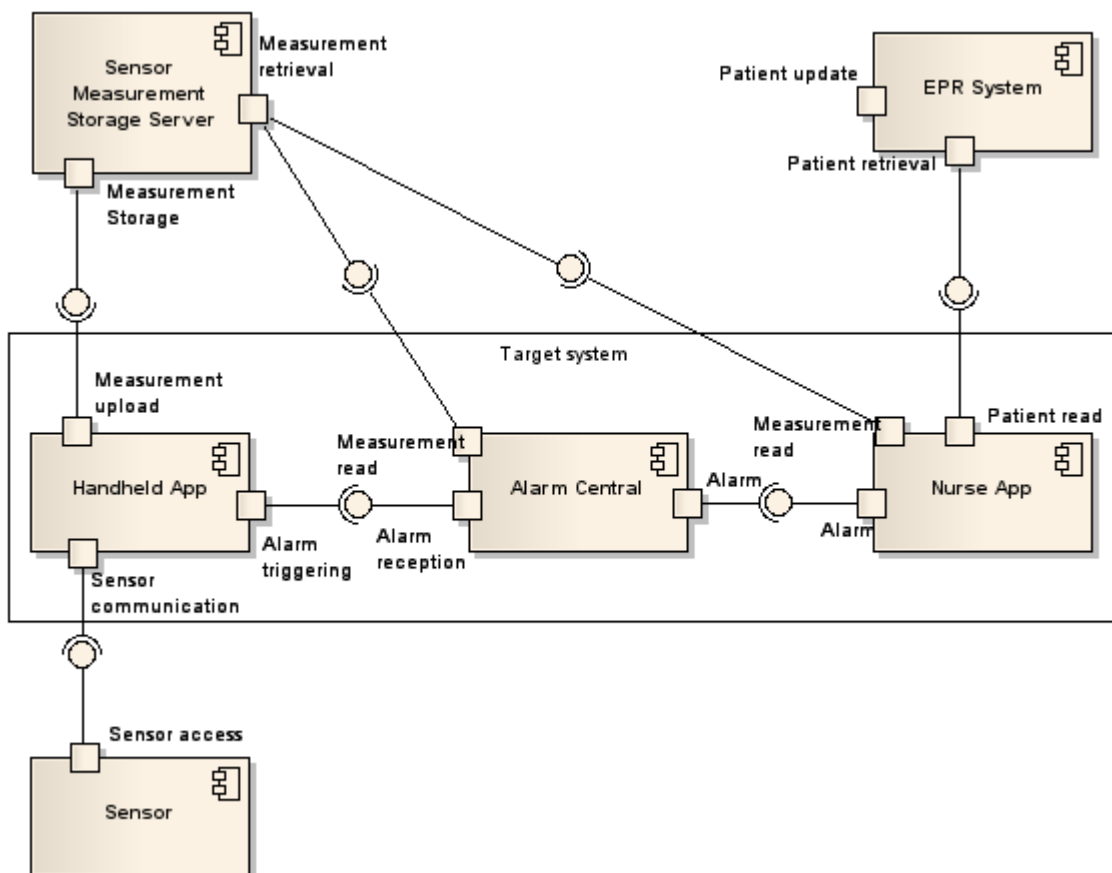


Figure 8: Decomposition into sub-systems

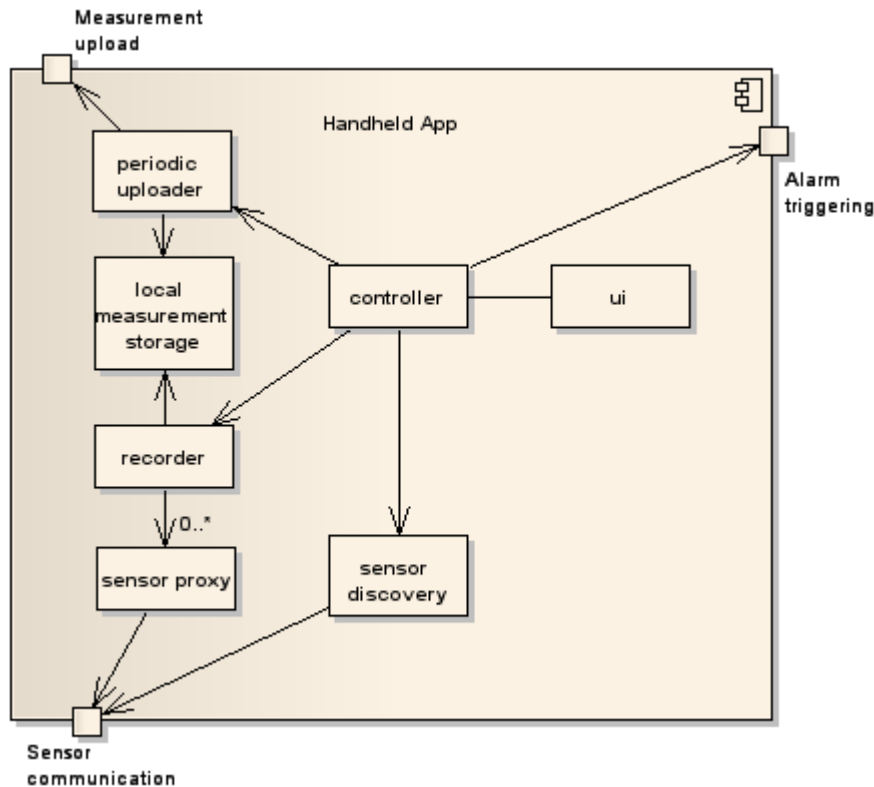


Figure 9: Decomposition of Handheld Application

Components that constitute the target system can be further decomposed using UML composite structure diagrams. For instance, the handheld application component from Figure 8 is further decomposed in Figure 9. The figure illustrates the parts from which the internals of the application will be built, and shows which of these parts will interact with the external ports of the component (which are the same as in Figure 8). Although not visible in this diagram, each of the parts is associated with an implementation class.

Mapped to the reference architecture (shown in Figure 1 in Section 2.5), the *ui* belongs to the UI tier, the *controller* belongs to the user service tier, the *periodic uploader* and *recorder* belongs are business services, the *local measurement storage* is a resource service, while the the *sensor discovery* and *sensor proxy* provide environment interfacing.

5.3 System Collaboration Model

The sequence diagram in Figure 10 gives a high level description of how the parts of the Handheld App interact to support the continuous monitoring, including how it is started and stopped. This example illustrates how sequence diagrams can be used to describe how central use cases are realized by the components of the system.

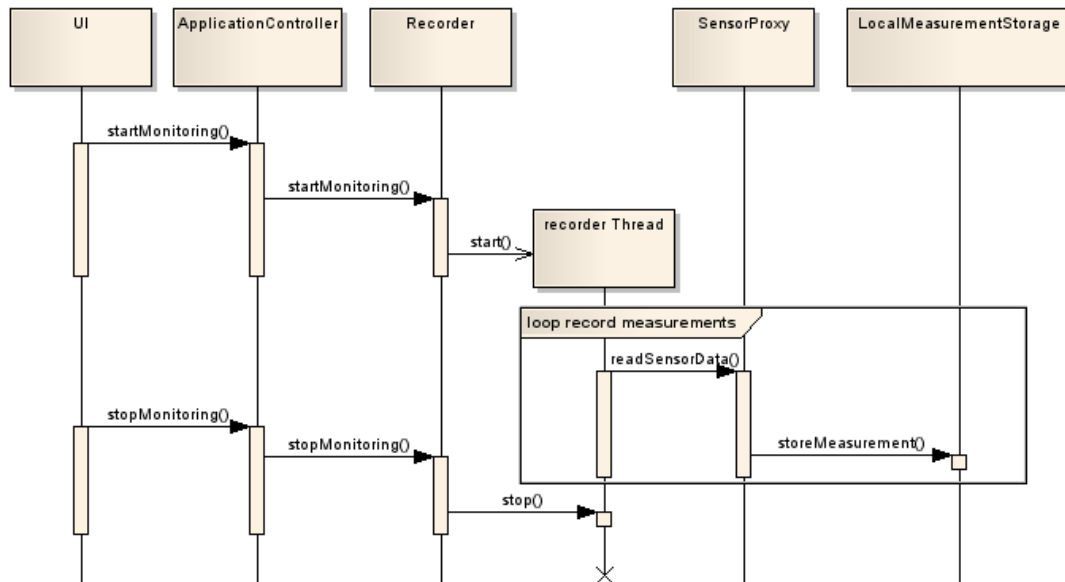


Figure 10: Sequence diagram for continuous monitoring in Handheld App

5.4 Component and Interface Specification Model

This sub-chapter will be provided in a future version of this document.

6 Distribution View

This chapter and its sub-chapters will be described in a future version of this document.

6.1 System Distribution Model

6.2 Role Distribution Model

7 Realisation View

This chapter and its sub-chapters will be described in a future version of this document.

7.1 System Deployment Model

7.2 Technology Mapping Model

7.3 System Integration Test Model

About this example

ARCADE is a domain and technology independent architectural description framework for software systems.

This document presents an example illustrating how the framework can be applied. The example includes description from each of the main viewpoints defined by the ARCADE framework, including the context in which the systems will be used, architectural requirements, component and information model description, logical distribution, and mapping to realisation.

The ARCADE Handbook and further information about the framework is available from:

<http://www.arcade-framework.org/>