An application of augmented MDA for the extended healthcare enterprise

Valerie M. Jones* and Aart van Halteren

Department of Electrical Engineering, Mathematics and Computer Science,
University of Twente,
The Netherlands
E-mail: v.m.jones@utwente.nl
E-mail: a.t.vanhalteren@utwente.nl
*Corresponding author

Dimitri Konstantas

Advanced Systems Group,
Centre Universitaire d'Informatique,
University of Geneva,
Switzerland
E-mail: Dimitri.Konstantas@unige.ch

Ing Widya and Richard Bults

Department of Electrical Engineering, Mathematics and Computer Science,
University of Twente,
The Netherlands
E-mail: widya@cs.utwente.nl
E-mail: bults@cs.utwente.nl

Abstract: Mobile health systems extend the Enterprise Computing System (ECS) of the healthcare provider by bringing services to the patient any time and anywhere. We propose a methodology for the development of such extended ECSs which applies a model-driven design and development approach augmented with formal Validation and Verification (V&V) to address quality and correctness and to support model transformation. At the University of Twente we develop context aware m-health systems based on Body Area Networks (BANs). A set of deployed BANs are supported by a server. We refer to this distributed system as a BAN System. Development of such distributed m-health systems requires a sound software engineering approach and this is what we target with the proposed methodology. The methodology is illustrated with reference to modelling activities targeted at real implementations. BAN implementations are being trialled in a number of clinical settings including epilepsy management and management of chronic pain.

Keywords: m-health; mobile services; telemonitoring; teletreatment; body area networks; BANs; model driven architecture; MDA; modelling; verification and validation; software engineering.

Reference to this paper should be made as follows: Jones, V.M., van Halteren, A., Konstantas, D., Widya, I. and Bults, R. (2007) 'An application of augmented MDA for the extended healthcare enterprise', *Int. J. Business Process Integration and Management*, Vol. 2, No. 3, pp.215–229.

Biographical notes: Valerie M. Jones is a Senior Researcher at the University of Twente and was Co-Scientific Coordinator of the MobiHealth project. Research interests include: formal methods and modelling, e-health and m-health applications, the ICT research challenges raised by these applications and future m-health possibilities enabled by Ambient Intelligence.

Aart van Halteren is an Assistant Professor at the University of Twente. His research interests include software infrastructures for (context-aware) networked applications, body area networks and m-health applications. He is responsible for the architecture of the MobiHealth platform in several collaborative research projects.

Dimitri Konstantas is Professor at the University of Geneva (CH) and the University of Twente (NL). He has been active for 20 years in research into object oriented systems, agent technologies, multimedia applications, e-commerce services and m-health systems. His current research interests include networked and mobile applications specifically in the area of e-health.

Ing Widya is an Assistant Professor at the University of Twente.

Richard Bults is a Senior Project Engineer at the University of Twente. He was a Technical Manager of MobiHealth and is currently the Project Manager of HealthService24.

1 Introduction

Mobile health systems (m-health systems) can extend the Enterprise Computing System (ECS) of the healthcare provider by bringing the services to the patient at any time and at any place. We present a methodology for design and development of such extended ECSs. The methodology applies the design and development approach of Model Driven Architecture (MDA) (Kleppe et al., 2003; MDA Guide Version 1.0.1, 2003). The MDA approach is selected for investigation as it aims to address the complete development life cycle and promises support for portability, cross-platform interoperability, platform independence and domain specific modelling.

Further we propose to investigate augmenting MDA with formal Validation and Verification (V&V) in order to address quality and correctness of both design and implementation, and to support model transformation. The importance of quality and correctness cannot be overemphasised for the sensitive and safety-critical application domain of healthcare. We illustrate the proposed methodology with respect to Body Area Networks (BANs) for healthcare.

At the University of Twente we are developing m-health systems based on BANs. The work began with the European IST project MobiHealth (Jones et al., 2001; Konstantas et al., 2002a,b; van Halteren et al., 2003, 2004; Widya et al., 2003) and continues in the Dutch FREEBAND Awareness project and the European eTEN project HealthService24.

In MobiHealth we defined a BAN as a collection of intercommunicating devices (a computer network) which is worn on the body, providing an integrated set of personalised services to the user. One specialisation of the generic BAN concept is the *health BAN*, which incorporates a set of devices and associated software components to provide some set of health-related services. This m-health application extends the operation of the healthcare provider into the community by bringing services to the patient and by feeding back captured data into the healthcare provider's ECS.

We have previously outlined (Jones, 2006; Jones et al., 2004, 2005;) an extension of the model-driven approach wherein formal methods are used to support the process of MDA modelling and model transformation. We believe that this design and development methodology has potential to add a practical but robust dimension to verification and validation of models and of transformations. In this paper we report on ongoing modelling work relating to BANs. The models are targeted at real implementations of BANs which are trialled in a number of clinical settings including epilepsy management and management of chronic pain.

The methodological framework, and one way of instantiating it, are described in Section 2.

2 The methodology

We propose to investigate applying the model-driven approach of MDA, creating Platform Independent Models (PIMs) and transforming them to derive Platform Specific Models (PSMs), and from them implementations. The innovation of the proposed approach lies in the augmentation of MDA by the use of (tool supported) mathematical formalisms to support V&V in the context of the Model Driven Development (MDD) trajectory. First we discuss the MDA approach.

2.1 MDA and MDD

OMG's MDA and the associated MDD involve the creation of a series of models, each derived from the previous, culminating in an implementation (which we can also regard as a model). The process involves *model transformation* to derive each model from the previous one, effected by reference to a metamodel for each language involved. If model m_1 is written in language L_1 then the definition of language L_1 is in MDA terminology the metamodel of m_1. Transformation of model m_1 written in L_1 to derive model m_2 in language L_2 would involve applying to m_1 a set of transformation rules (a transformation definition) between metamodels L_1 and L_2. (There may be many different transformation definitions between any two languages). The OMG has developed a standard language for writing transformation rules, called Query/View/Transformation (QVT) (Meta Object Facility (MOF), 2005).

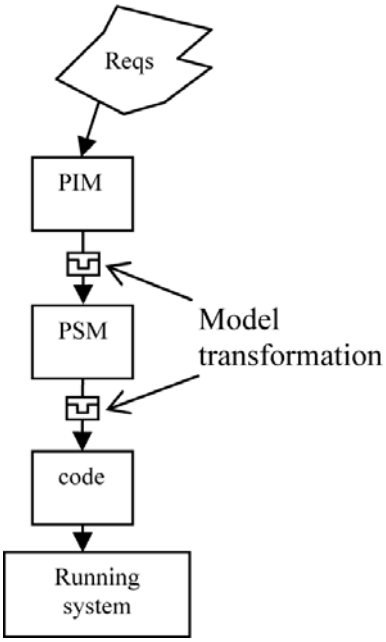
Key notions in MDA/MDD are abstraction and automation, which means that the transformation of m_1 to m_2 ideally should be automated, and that m_1 and m_2 may have a different level of abstraction. Normally, the transformation will be defined from the higher level of abstraction to the lower level, but 'reverse engineering' transformations are also useful. Automated transformation reduces the risk of human error and differences in interpretation between the modeller who created the source model and the implementer who is responsible for creating the target model (the implementation). The drawback of automation, however, is that the quality of the resulting implementation depends completely on the quality, notably the correctness, of the transformation definition.

The focus on abstraction has led to the definition of the MDA concepts of PIM, PSM and code model (code), each of which is characterised by its dependence upon the language and other artefacts used to implement the system.

Actually at each stage (PIM, PSM, code) there may be not one but a set of models at the same level of abstraction. For example, at PSM level there may be different models for different parts of the system, relating to different implementation technologies (e.g. an SQL model and a Java model). Furthermore there may be several substeps and different levels of abstraction within each step, such as class hierarchies relating to a domain ontology (an example appears in Figure 12). Adjacent models in a transformation process may be written in the same language. In that case the model transformation may be motivated by refactoring, with the goal of optimisation for example.

We ignore these complications for the moment and present a simple view of the MDD process below in Figure 1. Following (Kleppe et al., 2003) we indicate model transformation graphically by a T-shape embedded in a box, signifying a combination of a particular transformation definition from a particular source to a particular target language, (the T-shape), introduced into a transformation tool.

Figure 1 Simple view of MDA development



Put simply, the overall design and development steps of the MDA/MDD trajectory are:

- model the PIM
- derive PSMs from the PIM
- derive code from the PSMs.

where the second and third steps are performed by means of model transformation.

We now examine how we might augment MDA/MDD with formal V&V to increase confidence in the correctness and reliability of developed systems.

2.2 Some candidate V&V techniques

We consider four formal approaches to V&V. The formal techniques under consideration are selected because they represent the state of the art in formal methods and are not only practically applicable but can be automated.

2.2.1 Model checking approach

Model checking is a formal verification technique "that, given a finite state model *M* of a system and a property *P* stated in some formal notation (e.g. temporal logic) systematically checks the validity of the property" Ruys, 2001). It should be noted that in MDA and some other modelling approaches the terms 'model' and 'specification' are frequently used interchangeably; in formal methods however they are distinguished. A specification is the property which a model should satisfy, and may be expressed in a mathematical formalism such as Linear Temporal Logic (LTL) or Computation Tree Logic (CTL). *P* is referred to as 'a property' but may be a conjunction of properties which the design should satisfy. We refer below to *P* as a 'set of properties'. Furthermore the term 'model' is also used in a stricter sense; we refer to this stricter notion as a Verification Model (VM). A VM is also expressed in a mathematical formalism, for example, as a Labelled Transition System (LTS). A VM corresponding to each MDA model therefore needs to be derived. We may choose to apply model checking only to the last in the series of models, that is, the implementation, in which case the technique is known as *software model checking*.

2.2.2 Proof of implementation

This is a formal verification approach whereby the *implementation* relation must be demonstrated to hold between adjacent steps in the development trajectory; namely we must prove that each VM *implements* the preceding one. There are different formal definitions of implementation; equivalence for example, is a special (commutative) case of the implements relation. Different kinds of equivalence (e.g. testing equivalence, trace equivalence and bisimulation equivalence) can be verified.

2.2.3 Correctness by construction

Whereas the previous approach is aimed at post-hoc verification, in this approach the process whereby each model is derived from the previous one is guaranteed to be correct, for example by applying Correctness Preserving Transformations (CPTs) (Bolognesi et al., 1995). Neither this nor the previous approach verifies anything about the 'correctness' of the first model however, only about the relation that holds between adjacent models in the development trajectory.

2.2.4 Formal testing

A fourth candidate approach is the formal testing approach of (Tretmans and Belinfante, 1999), where a test suite is automatically derived from a (verification) model. The tests are applied not to another model but to the implementation. The formal testing process thus represents a kind of validation of the implementation with respect to a

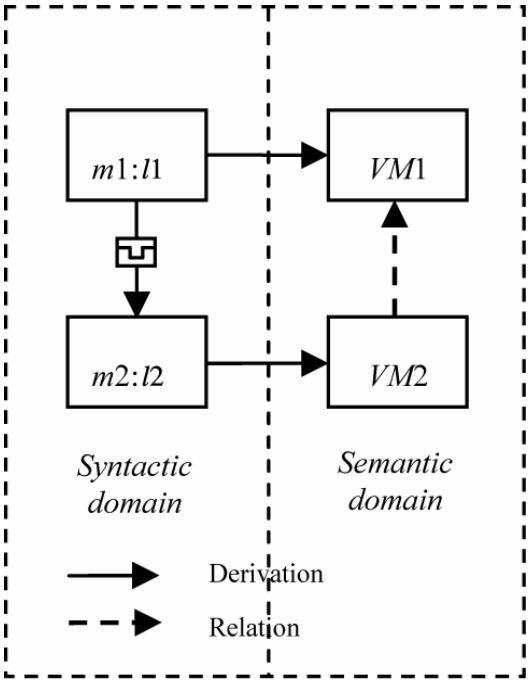
model. In MDA terms we could derive tests from the VMs corresponding to the PIM and/or the PSM(s) and apply them to the running system. This approach is a validation technique, but exceeds in a quantifiable way the coverage of traditional testing approaches. Together with use of formal models and formal verification, formal testing gives a much more solid basis for raising confidence in correctness of systems.

2.3 The A-MDA methodology

MDA model transformation operates at the syntactic level. We propose to use formal methods to introduce a semantic dimension to MDA modelling and MDD model transformation. The motivation is to be able to demonstrate that certain kinds of formal properties are maintained by the development process, thus establishing the correctness of the implementation, or at least establishing the weaker claim that the implementation satisfies a certain set of specified properties.

For any MDA model written in a well-defined language, we can derive a corresponding semantic model. The semantic model can be expressed as a LTS. So if we derive model m_2 in language L_2 from model m_1 in language L_1 by model transformation (i.e. by applying a (syntactic) model transformation from L_1 to L_2), we can introduce semantic checks into MDA by reference to the corresponding semantic models. For example, we can demonstrate the semantic equivalence of models m_1 and m_2 if we can show that m_1 and m_2 map to equivalent semantic models VM_1 and VM_2 (see Figure 2).

Figure 2 Adding the semantic domain

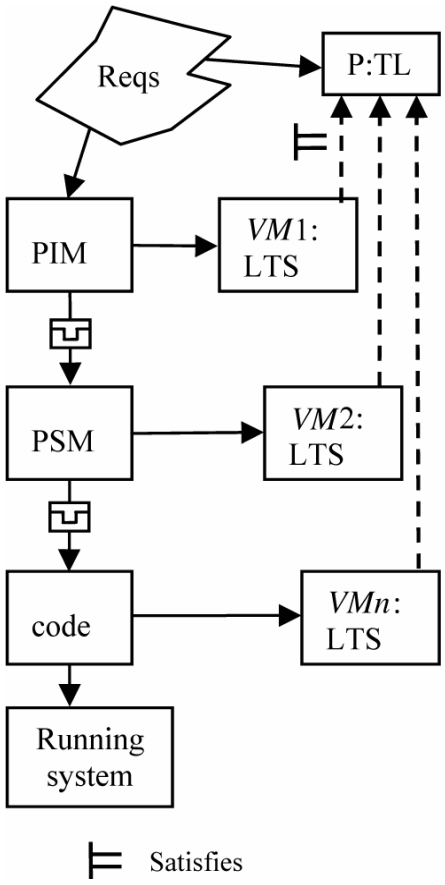


We now identify four scenarios for augmenting the MDD process with the V&V techniques identified in Section 2.2 above. Scenarios (b) and (c) involve verification, (d) involves validation. Scenario (a) could be either validation (debugging) or verification (partial or full verification).

2.3.1 Scenario (a) MDA model checking approach

Here we combine the MDA development trajectory with a model checking approach, by checking the specification (the set of properties to be verified, as derived from the requirements) against VMs derived from the MDA model(s) at one or more of PIM, PSM and code levels using model checking (see Figure 3).

Figure 3 MDA model checking

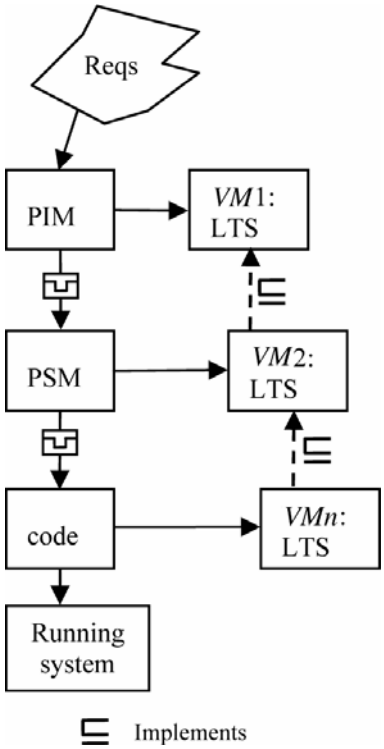


In this scenario the implementation (as code) is derived by a series of development steps conducted by conventional MDA model transformation, that is, the transformations operate on a syntactic level by application of a transformation definition for each pair of languages $<L_n, L_{n+1}>$. The first model and the set of properties to be verified are derived from the requirements. The properties P are checked against the VMs at one or more steps to check that the model satisfies the properties P. If verification is applied only between the original specification of properties and the final implementation; the process does not say anything about the semantic validity or correctness of the intermediate steps.

2.3.2 Scenario (b) Proof of implementation between adjacent model transformation steps

This scenario relies on proving a semantic relation *implements* at each transformation step, with a selected definition of the implements relation (see Section 2.2.2 above) (Figure 4).

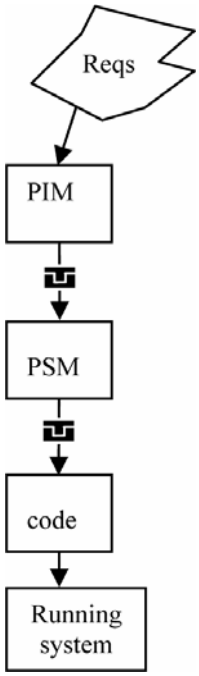
Figure 4 Proof of implementation in MDA



2.3.3 *Scenario (c) Correctness preserving model transformation*

In this scenario the semantic dimension is introduced into the transformation step explicitly, by ensuring that the MDA transformation definitions on metamodels are not only valid syntactic transformations but are also correctness preserving; hence models derived by transformation are correct by construction at each step in the chain of model derivations. We indicate this semantically augmented version of model transformation by inverting the model transformation symbol (see Figure 5).

Figure 5 Correctness preserving model transformation

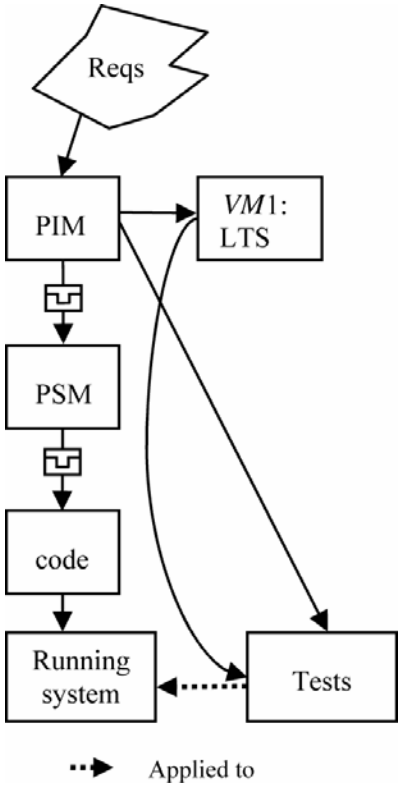


The approach presupposes that for the given source and target languages (metamodels) in each model transformation step, there exists a transformation definition which is both valid syntactically and also guarantees semantic equivalence. Creating such transformation definitions will however require significant effort.

2.3.4 *Scenario (d) Automatic test generation and formal testing in MDA*

The fourth scenario is based on application of formal testing within the MDA framework, where tests are derived from some VM in the trajectory (preferably derived from the first model in the PIM stage). The tests are automatically derived and applied to the implementation. The formal testing process thus proves (or disproves) some kind of conformance of the running system, which results from following the MDD design trajectory, with the PIM. Figure 6 shows this automatic test generation approach.

Figure 6 Formal testing in MDA

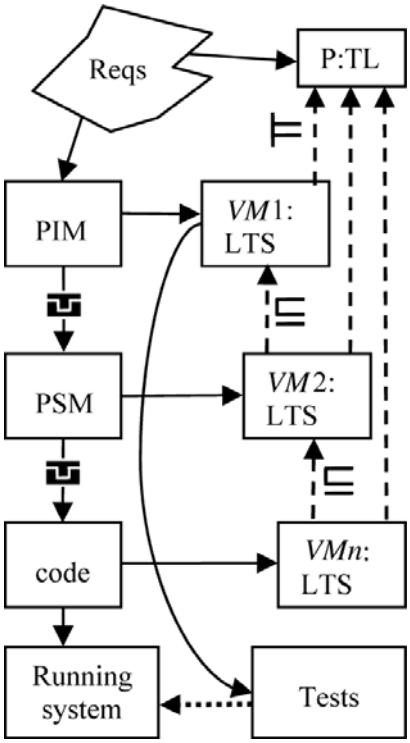


2.3.5 *Integration of approaches*

Finally, we can consider a composition of the different approaches to combining MDA with formal V&V, as illustrated in Figure 7 below.

Why do we need a combination of V&V methods? Firstly because validation methods increase confidence in the correctness of the design and/or system, but cannot give watertight guarantees. Verification techniques in principle can give proofs that certain properties hold, but they are not practical to apply exhaustively to realistic sized systems, such as m-health systems. Some of the difficulties are explained below.

Figure 7 Integrated approach



In the model checking approach (a) the property *P* has to be derived from the requirements and there is no way of knowing if everything has been anticipated; therefore *P* will often be weak or incomplete. However model checking can give a great deal of important information and confirmation that certain properties do in fact hold. In reality in current practice the use of model checking can be thought of as tending more towards debugging (thus a form of validation) than full verification. Furthermore model checking is known to give false negatives on occasion.

Scenarios (b) – proving the implementation relation – and (c) – the CPT approach – each would give complete verification in theory. If we could apply either (b) or (c) completely and perfectly then each would make the other three approaches redundant, at least for the steps in the trajectory which they address. However in practice they are both difficult to achieve completely, especially in the final step of code derivation. Similarly it is more difficult to define CPTs where the target language is a programming language rather than a 'clean' mathematical modelling language. Even for mathematical languages complete CPT schemes do not exist today. Hence we conclude that for the present a judicial combination of the more practical verification technique of model checking with validation by formal testing can give much more leverage on the quality assurance problem with coverage spanning the whole trajectory from requirements to the running system. Formal testing has the additional advantage that it applies to the running system in the execution environment. Testing can also provide a second line of defence in the case where model checking gives false negatives; when this is suspected testing can be used to demonstrate counter examples disproving the false negatives. Figure 8 below shows a high level view of the proposed A-MDA approach.

Figure 8 A-MDA methodology

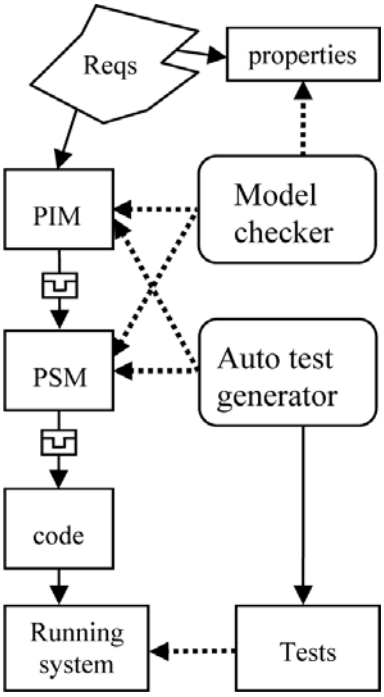
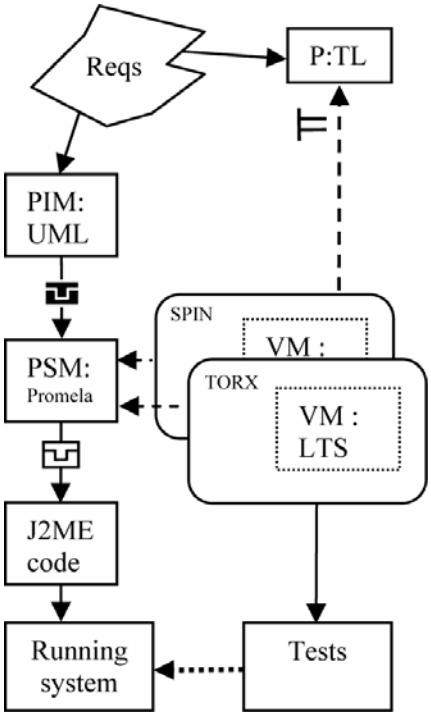


Figure 9 shows one possible instantiation of this approach using the tools SPIN for model checking and TORX for formal test generation and execution. (Note: not all V&V paths shown in Figure 7 are instantiated here.)

Figure 9 An instantiation of the integrated approach



This example is targeted at a J2ME implementation (hence for mobile devices). UML is used for modelling at the PIM step. Promela is used as an intermediate modelling language, since it is accepted by both SPIN and TORX. (Promela models are considered as lower level and arguably more suitable for PSM than PIM modelling.) TORX accepts a Promela model as input and generates a set of tests from it, which can then be applied to the final

running implementation (i.e. to executing code in a run time environment). Unlike scenario (b) where the comparison operates on the implementation as a semantic model – *a text* – (we can think of that as a white box approach) here we compare a model with the implementation as *running code* (thus a black box approach). This realisation of the integrated approach is not perfect; we would prefer to generate tests from the first rather than a subsequent model. However this practical compromise represents an improvement over the current state of the art with MDA and can serve us as a first version of A-MDA.

To summarise, we have chosen to apply two V&V methods in the context of MDA: application of model checking to models, and formal testing based on automatic test derivation. The tests are derived from the models but applied to the implementations, thus proving some form of formal equivalence between models and corresponding implementations. It is also planned to investigate the use of formal methods to address the task of model transformation. The A-MDA approach described here relates to previous work on model checking (Holzmann, 2003; Ruys, 2001; Ruys and Brinksma, 2003), formal testing (Brinksma, 1999; Tretmans and Belinfante, 1999) and transformation (Jones, 1995, 1997).

With this first approximation to the A-MDA methodology, the following additional verification and validation steps are to be performed in parallel with the MDA development trajectory:

- formulate critical properties (assertions derived from the requirements)
- model check the PSM (already cast as a VM) against formally expressed properties
- apply automatic test generation to the PSM
- apply the test suite thus derived to the implementation.

In this paper we illustrate some of the steps of the general methodology with reference to particular modelling paradigms and notations (UML, Promela, *me too*), particular tools (SPIN model checker, TORX test generator) and aim at a particular target implementation technologies (e.g. J2ME). Many other choices could be made at all steps.

3 Modelling an m-health system

In this section we describe the m-health application (BAN-based mobile healthcare services). We illustrate some initial modelling of PIMs, and discuss some of the PSMs required for this application. This modelling exercise exemplifies part of the first phase in the application of the A-MDA methodology.

3.1 BANs for healthcare

The concept of BAN originally came from work at MIT and IBM (Zimmerman, 1999) but was first discussed under the topic of Personal Area Networks (PANs) and only later distinguished by the use of the separate term BAN. Zimmerman used the term 'Intra-Body Communication' in the context of PANs to describe data exchange between body worn devices using the body itself as the communication medium. The concept was developed further by other groups, for example at Philips (van Dam et al., 2001), by the MobiHealth team at the University of Twente and at Fraunhofer. In the Wireless World Research Forum's Book of Visions, we defined a BAN as "a collection of (inter) communicating devices which are worn on the body, providing an integrated set of personalised services to the user" (Wireless World Research Forum, 2001). In the MobiHealth project we defined a BAN not by transmission technology but by physical position and range, as a computer network which is worn on the body and which moves around with the person (i.e. it is the unit of roaming). We use this definition in the remainder of this paper.

A BAN incorporates a set of devices which perform some specific functions and which also communicate via a central controlling device which we call a Mobile Base Unit (MBU). Devices may be simple devices such as simple sensors or actuators, or more complex devices such as sensor systems, multimedia devices such as cameras, microphones, audio headsets or media players such as MP3 players. The MBU may perform computation, coordination and communication functions. Communication amongst the elements of a BAN is called *intra-BAN* communication. Any external communication, that is, with other networks (which may themselves be BANs), is termed *extra-BAN* communication.

Up to now we have been discussing generic BANs; this concept can be specialised by application domain, for example to health BANs or entertainment BANs. In the MobiHealth project a prototype of a health BAN system was developed, together with several specialisations of the health BAN for telemonitoring. Different variants were trialled on different patient groups including cardiac patients, patients with chronic respiratory disease (COPD) and pregnant women. Further specialisations of the health BAN have been developed within the Awareness and HealthService24 projects, including telemonitoring BANs for epilepsy and teletreatment BANs for chronic pain management. Figure 10 shows the architecture of the BAN and Figure 11 shows the hardware components used in one of the BAN configurations.

Figure 10 Generic architecture of the MobiHealth BAN

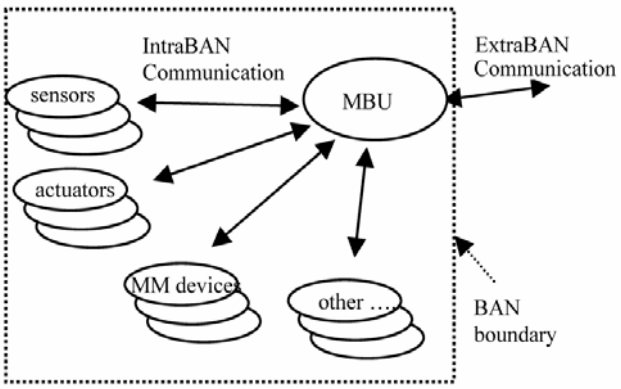


Figure 11 One configuration: PDA, front-end and sensors

Figure 11 shows a COPD BAN, where the MBU is implemented by a PDA (a Qtek). This BAN is equipped with a respiration sensor and 3-channel ECG. These are examples of front end supported sensors systems. The box in the centre is the sensor front end. This configuration represents one of many different specialisations of the generic BAN developed and realised at the University of Twente.

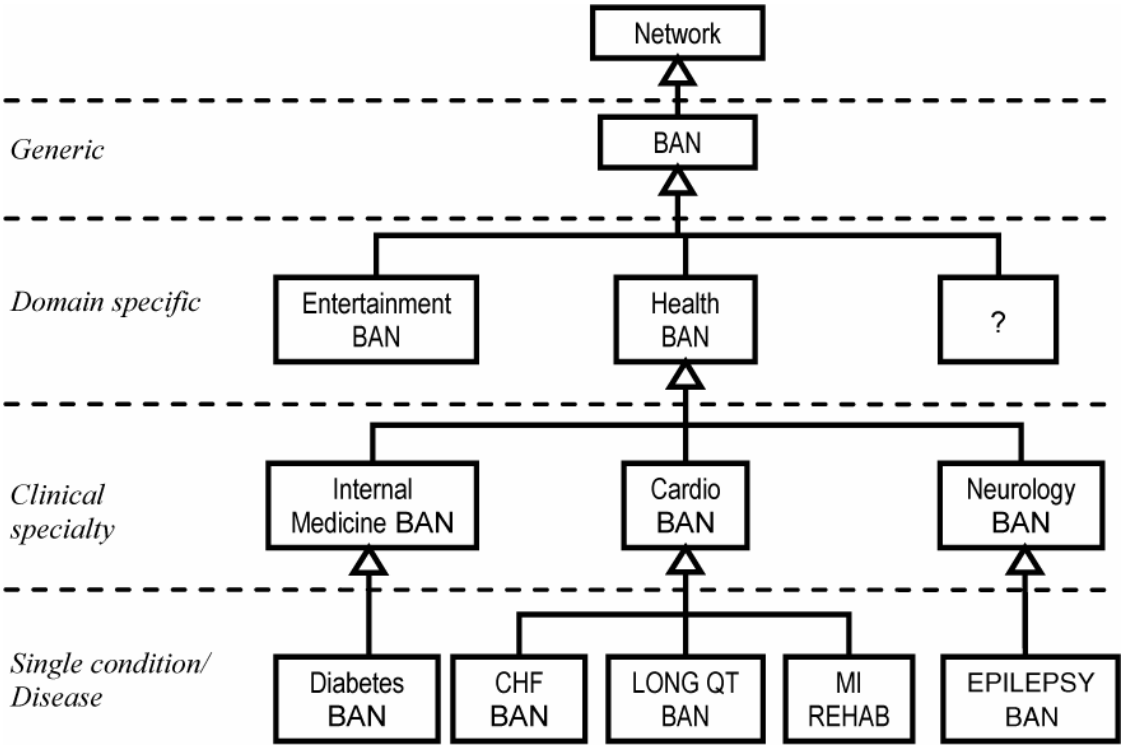
The concept space encompassing generic BANs, health BANS and specialisations of health BANs can be modelled as a class hierarchy, as shown in Figure 12. Here several levels of increasing specialisation of BANs are identified. The top level relates to generic BANs, where class (generic) BAN is seen as a specialisation of the more generic class Network.

The generic BAN can be specialised by domain (health BAN, entertainment BAN and so forth). We have chosen to distinguish the Health BAN as characterised not by use of medical devices, but rather as including *devices used for medical purposes*. By this means we can include generic devices such as cameras or GPS positioning systems in a health BAN on the grounds that they are being used for health-related purposes. Health BANs may be further specialised by clinical specialty (e.g. internal medicine or neurology), however this level of specialisation may not always be specific enough to begin to talk about services. So we distinguish a further level of specialisation: clinical condition. At this level we can begin to identify disease management services and for each service, an associated set of devices and application components. Examples of (still rather generic) services would be ECG monitoring, blood pressure monitoring, blood glucose monitoring, notification services, positioning services, medication reminders, fall detection, loss of consciousness detection and control signals to implanted devices of various kinds. Within one specialty (e.g. cardiology) we can distinguish a different set of services for patients with different conditions. A patient with Long QT syndrome (a life threatening cardiac arrhythmia) may require ECG monitoring, heart rate monitoring and defibrillation services (hence the BAN devices may include an implanted defibrillator), whereas a patient recovering from myocardial infarction may require heart rate, heart rate variability and ECG monitoring services so their BAN may include electrodes for measuring ECG (from which the other parameters may be derived). Such BANs should be generic for a class of patients, but of course may require tailoring to the needs of individual patients. Later we discuss the issue of customisation and personalisation of BANs.

3.2 Modelling the health BAN (PIM level)

In this section we present examples of modelling using two different formalisms: a linear discrete mathematics notation and UML diagrams. The goal of the modelling activity is not only to encompass all the existing specialisations of the MobiHealth BAN but also to be

Figure 12 Health BANs in UML class hierarchy



generic enough to cover the current BAN developments conducted in the Awareness and HealthService24 projects as well as many future possible instantiations of BANs, including those based on future ambient intelligence technologies such as smart sensor networks and perhaps incorporating implanted and nano-scale devices.

First we model the BAN system. There are two main categories of users of the BAN system: the patient users and the professional users. A patient wearing a BAN has a set of services available to him/her, varying with his/her current set of needs and his/her clinical conditions(s). Some of the services may be transparent to the patient and fully automatic (e.g. telemonitoring, automatic alarms) others may be patient driven (e.g. patient initiated alarms).

The professional users are the consumers of BAN captured data such as biosignals and alarms. They may be health professionals or other professional care providers. The (health) professional or (health) care provider interacts with their patients' BANs via a BAN Professional System. This system provides BAN specific services, but may interface to the healthcare provider's ECS such as a GP practice administrative system and/or Clinical Information System (CIS) or a Hospital Information System (HIS), possibly interfacing directly to the Electronic Medical Record (EMR). The professional system may itself run on a mobile system (e.g. a laptop or PDA.). Services for professionals include access operations (e.g. retrieving and viewing biosignals) and also control operations such as remotely activating a BAN, or a BAN device or altering sampling frequencies of sensors. Both patient and professional systems will have many different specialisations incorporating different functionality sets, hardware and applications.

3.2.1 The BAN system model

A great many individual patient BANs and professional BAN systems may be in operation out in the field at any one time. These components are supported by a server which knows about management of BANs and BAN applications and which mediates between the patients and the professional users. We refer to this server as the BAN Back End System (*BESys).* Together these components – BANs, Professional Systems and BESys – comprise a distributed system which we refer to as the BAN system. Communication between the components is effected via communications channels. At the most abstract level we do not distinguish further (e.g. into wired/wireless channels). Figure 13 illustrates a logical view of these components. The components to the right hand side of the dotted line are in the domain of the healthcare provider's ECS, and to the left hand side are the components of the BAN system which extends it.

The BeSYS provides, amongst others, the BAN access functions to the healthcare providers' ECS and to health professionals' mobile systems.

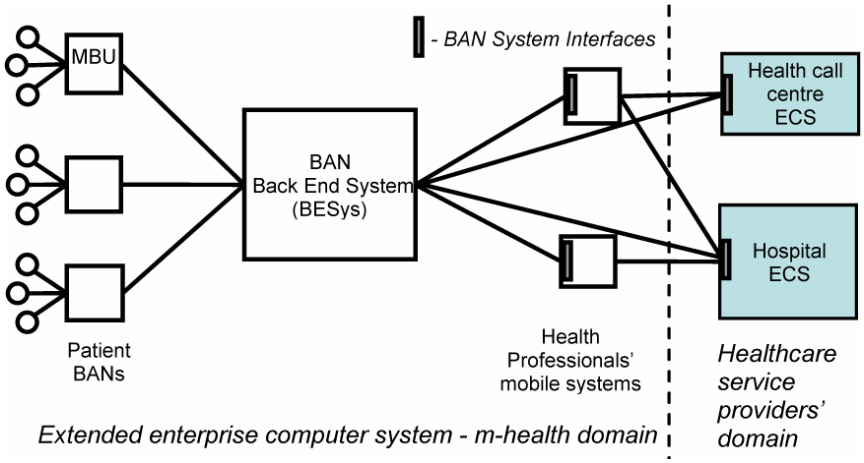
Our model identifies the classes of objects involved in a BAN System as seen in Figure 13, provides a mathematical representation of the object class BANSystem and identifies the services it offers, for example, ECG monitoring. These services will be further specified at a lower level of abstraction, depending on the clinical requirements. ECG monitoring may be specified as 3-lead or 12-lead, for example. At the time of instantiation further attributes such as sampling frequency and required mode and quality of presentation can be specified. At the highest level the model shows that a BAN System consists of one Back End System, a set of BANs, a set of BAN Professional Systems and a set of channels linking these systems.

```
BANSystem = tuple(BESys, set(BAN),
                  set(BANProfSystem), set(Chanel))
```

This is a type specification taken from a *me too* (Alexander and Jones, 1990) model. The first step in the *me too* method is identification of objects, operations and their relationships. This gives a mathematical description of the concept space and plays a role in the elaboration of the domain ontology. In further modelling steps the signatures and formal definitions of operations are given. Constraints which can be used for model checking can be expressed as predicates.

The BAN system provides services to different classes of user (patients, health professionals) and also provides system services. BAN services offered to health professionals include: Request subscription, Start BAN, Stop BAN, Show BANs, Show BAN, Show BAN Devices, View BAN Data, Call Patient, Change Sampling frequency

Figure 13 BAN system and its interfaces to the healthcare providers' EC systems



and Add Application. The *me too* model includes these as operations, specified by signature and by formal definition.

BAN services offered to patients might include: ECG monitoring, blood pressure monitoring, blood glucose monitoring, patient initiated alarm, automatic alarm, location services, medication reminders, activity monitoring, fall detection, loss of consciousness detection, epileptic seizure detection and epileptic seizure prediction. Although these are patient care services in most cases they are transparent to the patient and the only active use is by the health professional. BAN system services include: BAN/MBU discovery, BAN/MBU release, BAN service discovery, BAN service registration, add service to BAN, remove service from BAN and push sensor data.

We can view a BAN system as a network, where the nodes are BANs, professional systems and the Back End System. In terms of network topology it could be modelled as a graph. We now turn to the BAN itself.

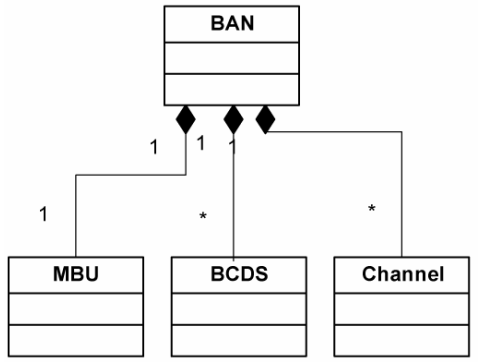
3.2.2 The BAN model

Now we look at the internal components of the BAN. The MBU or Mobile Base Unit is an (abstract) device which combines the functions of communications gateway and a computation platform. In the MobiHealth and Awareness projects the MBU functions have been implemented on PDA and smart phone platforms but in future the functionality could be implemented on a specialised chip, which could perhaps be implanted. In the network view a BAN is a kind of network where the nodes are the MBU and the other BAN devices and the channels are the (wired or wireless) links between the devices. Since the nodes may themselves be complex components or subnetworks we refer to them as BAN Connected Device Systems (BCDSs) or BAN devices for short. From a network point of view the BAN can be specified thus:

BAN = tuple(MBU, set(BCDS), set(Channel))

Figure 14 shows the corresponding UML class diagram. (Note channels could alternatively be modelled by association classes.)

Figure 14 UML BAN model



We identify three subclasses of BCDS: sensor (a device which performs some measurement), actuator (a device causing some mechanical action,) and multimedia device (such as cameras, microphones, display devices and headsets). Many more devices, such as pumps, pacemakers and defibrillators, are possible candidates and may incorporate sensors and actuators. We specify this at a high level as:

BCDS = Sensor | Actuator | MM_device|

We may extend the model to include the concept of services offered by the BAN:

BAN= tuple(MBU, set(Service),
set(BCDS), set(App), set(Channel))

where each service implies a set of hardware components (BAN devices) and an application (a set of software components).

In order to support reuse, the high level representation (at the level of the PIM) needs to cover not only devices used in the past and current projects, but should also accommodate all kinds of BAN devices that we can envisage in the future.

Constraints on permitted connectivity and attributes of nodes and channels need to be modelled at a later stage. Care should be taken to introduce constraints at the appropriate levels of abstraction and at the appropriate levels of specialisation. A channel links two network nodes and has associated attributes. The PIM attributes may represent information about required data flows and synchronisation. At PSM stage some attributes will take values relating to which technologies are used (e.g. Bluetooth, Zigbee, WLAN and WiFi).

From the possible range of devices which may be connected to a BAN, we focus now on sensors. Individual sensors may be connected directly to the MBU. In other cases a collection of sensors which is managed by its own front end device may be connected; we refer to this subsystem as a Front End Supported Sensor System (FESSS).

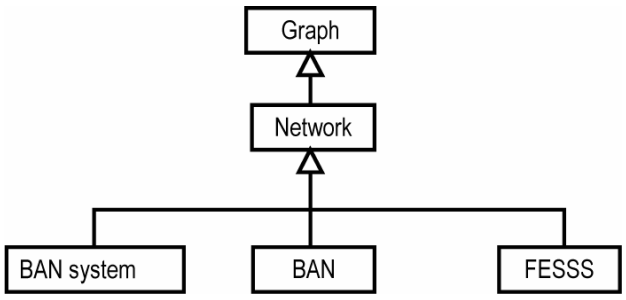
Sensor = SimpleSensor | FESSS

In an FESSS the front end device receives raw signals from one or more sensor sets and performs some processing on the signals before outputting the processed signals to a consumer component. The front end powers the sensors, handles synchronisation of signals and may be able to handle different sampling frequencies for different sensors.

FESSS = tuple(FrontEnd,
set(Sensorset), set(Channel))

Note the homomorphism between FESSS and BAN. We have already noted that the BAN system is a network, some of whose nodes (BANs) are themselves networks. Within the BAN we also see networks of devices, such as sensor systems comprising a set of sensors, a clock and a sensor front end. An FESSS (especially from the topological perspective) may also be modelled as a graph. We represent the concept of recursive networks in a UML diagram in Figure 15.

Figure 15 Nested networks



In this section we have given an outline of some of the concepts involved in the m-health application and given examples of the modelling activities whereby these concepts are being formalised as part of the PIM modelling step of the A-MDA methodology. In this case we have used UML diagrams for graphical representation and *me too* for linear textual representation.

3.3 Technologies and platforms for the health BAN (PSM level)

In this section we describe some of the technologies and platforms which may be used to realise subsystems of the BAN system model. For each of these we will require a PSM. Many kinds of PSMs will be required, addressing different aspects of the implementation, for example, modelling the target middleware technologies, programming languages and operating systems and of course the hardware components of the BAN. For each of these PSMs many choices of platform can be made for a given PIM. The models of hardware components to be integrated (such as commercially available sensor systems) can be regarded as PSMs in the sense that they refer to a particular hardware platform for implementation of a given (abstract) function. For example, a location service may be implemented using a particular commercially available GPS positioning device. Elaboration of the method of deriving PSMs from PIMs, and the validation and verification trajectory; are for future work; here we proceed directly to some outline examples of some of the components for which PSMs will be required.

3.3.1 The back end system

The Back End System was realised in MobiHealth as a proxy server using Jini technology to realise the BAN system services. The Back End System also implements other functions including a BAN Data Repository (BDR) for storing BAN captured data. For more details please refer to (Dokovsky et al., 2003). At the PSM level then we would need models and metamodels of the Jini architecture, the BDR and the other components of the Back End System. Access to BAN data is mediated by the proxy server. In a later version, jini technology has been replaced by web services technology. This situation illustrates the importance of the MDA argument for reuse enabled by PIM level models when target implementation technologies change.

3.3.2 Sensor systems

For implementing location services we select a particular positioning device, such as the GPS device from EMTAC. We define a specialisation of the class Sensor to be the class EMTAC GPS system, and specify as an attribute the service it offers (location service). This is a simple sensor so it can be connected directly to the MBU.

If ECG monitoring services are needed, we choose to implement this service using a particular FESSS, for example the Mobi from Twente Medical Systems International. The Mobi receives signals via wired connections from a number of signal sources and transmits the processed signals to a consumer over a wireless (Bluetooth) connection. We refer to the Mobi and attached sensors as a Mobi sensor system. This is an instantiation of an FESSS. The technical specification of a certain version of the Mobi includes the property that all sensor sets attached to the Mobi are synchronised with each other. Further they all operate at the same sampling frequency. This and other constraints and definitions can be expressed in the PSM and will be part of the specification which constrains the application model for BANs and BAN applications which use this version of the Mobi. Below we show the part of the PSM for (this version of) the Mobi which expresses these properties. The model fragment shown below should be read not as a requirements specification but as a specification formalising fixed properties of this device which need to be taken into account in the design and implementation of BANs which integrate instances of this device.

MOBISensorSystem

OBJECTS

MobiSensorSystem

Mobi

SetofSensorset

Sensorset

$MobiSensorSystem = pair(Mobi, SetofSensorset)$

$SetofSensorset = set(Sensorset)$

$Sensorset = set(Sensor)$

CONSTRAINTS

$\forall mss : MobiSensorSystem .$

$\quad \forall ss1, ss2 : 2(mss) . \ synch(ss1,ss2)$

$\forall s1,s2 : Sensorset .$

$\quad synch(s1,s2) \wedge samplefreq(s1) = samplefreq(s2)$

This model fragment identifies the objects concerned and shows their representations and relationships The first constraint expresses the synchronisation property and the second the constraint on sampling frequencies.

3.3.3 The MBU

Any number of PSMs of the MBU can follow from the PIM of the MBU. The PIM specifies that the MBU is the BAN's communication gateway taking care of Intra-BAN and Extra-BAN communications, and the computation platform providing BAN processing (generic BAN functions plus specific BAN services) and local storage. MBU services may be realised in one device, for example a UMTS enabled PDA or a smart phone, or they may be distributed over different devices, for example, a UMTS phone (for communications services) and a PDA (for storage and processing services). The following MBU platforms have been or are being targetted in the BAN development work at the University of Twente:

- ComPaq iPAQ 3870
- HP iPAQ 5550 / 4150 with Mobile Phone
- Qtek9000
- Qtek9090
- QBIC (Belt Integrated Computer).

In addition to a set of PSMs for the selected MBU device(s), we need PSMs for the software implementation technology, for example, J2ME or C++. Future development at Twente will target the Windows Mobile 2005 operating system, and hence a range of PDA and smart phone platforms.

3.3.4 A PSM of a condition-specific BAN

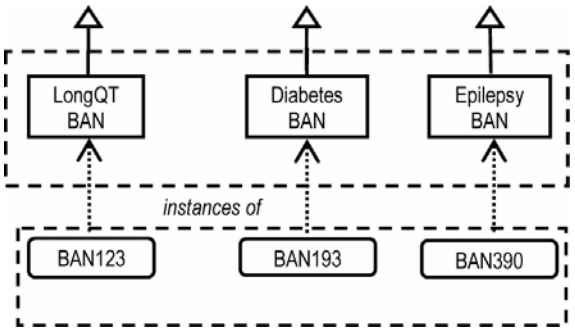
A condition-specific BAN provides a set of services associated with a particular disease or condition (see Figure 16). A BAN for Long QT syndrome patients might include ECG monitors and an implanted defibrillator. A BAN for diabetes management might include a blood glucose monitor and an implanted insulin pump. Each service implies a set of devices and associated application components to be configured on the MBU. Applications may in fact be distributed across the BAN and BeSys, even migrating dynamically between them. The purpose of the Epilepsy BAN developed in the Awareness project is to detect (and perhaps even predict) epileptic seizures. The services required are ECG monitoring, motion detection and location detection. Onset of seizure is detected by means of analysis of ECG signals including patterns of heart rate and heart rate variability (parameters derivedfrom the ECG signals). ECG information is analysed in the light of contextual information, for example, the patient's movements as detected by the motion sensor. In general the analysis software forms part of the accompanying condition specific application, which may also include disease management functions such as medication reminders and alarms. The location service gives positioning information so that the patient's geographical location can be pinpointed and assistance can be sent if necessary.

The PSM supports implementation of condition specific BANs by realisation with particular hardware components and associated software components. In realising instances of the Epilepsy BAN the actual hardware components might be

- QTEK 9090 (MBU)
- EMTAC GPS (simple sensor) for location
- TMSI Mobi delivering 4-lead ECG (an FESSS)
- Xsens MT9 motion sensor.

The PSMs corresponding to each selected device would form part of the Epilepsy BAN PSM.

Figure 16 Condition specific BANs



3.3.5 Creation of a personalised BAN

The model of a condition specific BAN represents a class of BANs which address the core needs of a group of patients with a certain condition. However every instantiation of the BAN may need to be customised for an individual patient by providing the set of services they require at a certain time. By personalisation we mean adjusting to the preferences of patient and the treating health professional team.

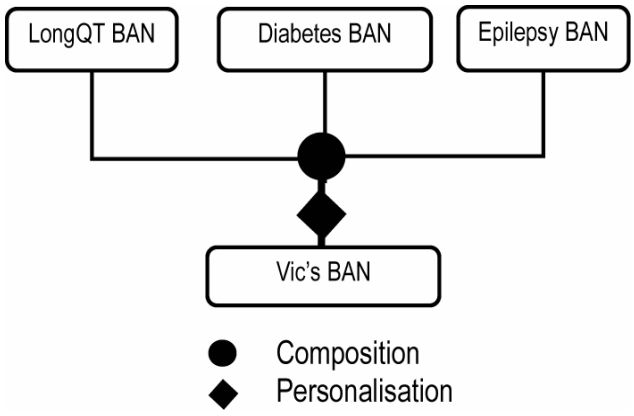
Customisation may involve fine tuning service parameters, or adding additional services and devices. Many patients suffer from multiple conditions (comorbidities) and therefore may need some combination of two or more condition-specific BANs. The methodology should support creation of personalised BANs. One approach would be by composition of condition-specific BANs as shown in Figure 17. Figure 17 represents the BAN needed by our hypothetical patient, Vic, who suffers from the life threatening cardiac arrhythmia know as Long QT syndrome. He is also an insulin dependent diabetic and suffers from epilepsy. Figure 17 shows that the BAN needed by Vic is some combination of the generic LongQT, Diabetes and Epilepsy BANs. In Figure 17 we introduce some graphical conventions, using a black circle to represent composition of BANs and a black diamond to represent personalisation. The figure should be read thus: Vic's BAN is derived from the generic LongQT, Diabetes and Epilepsy BANs by a process of composition followed by personalisation.

This procedure to create a BAN for Vic by composition of condition specific BANs could be modelled at a high level as:

personalise(compose(LongQTBAN,
DiabetesBAN, EpilepsyBAN))

Exploration of the issues involved in BAN composition is a question for future research.

Figure 17 Composition and personalisation



4 Discussion

BAN based applications are among the many potential new applications for the extended enterprise systems of the health sector enabled by wireless technologies. Healthcare systems for use by the public require high levels of safety, reliability, performance and ease of use and must be based on sound design and development paradigms. High standards are enforced by certification procedures. In response we investigate the use of formal models and methods and view formal verification as an absolute requirement. In the context of MDA, this exposes the need for further research since, for example, the issue of correctness preservation across transformations remains an open question. Tools to support MDA/MDD are available which support some degree of automatic code generation and which therefore can be presumed to incorporate some kind of transformation between pairs of languages. However development of these transformation modules represents a significant effort and investment for companies and consequently the transformation rules are often inaccessible to scrutiny. Hence it may not be clear whether these tools implement a 'pure' version of model transformation, and further the correctness of the transformations cannot be independently validated.

Developments in wearable devices proceed at a rapid pace, implying an urgent need for a methodology that supports platform shifts and offers flexibility. But one of the strengths of MDA – separation of platform independent from platform specific issues – brings with it an inherent problem. Platform models driving the PIM-to-PSM transformation are needed to support each new technology development. Who has the business incentive to provide these in this rapidly changing situation? Despite our efforts to be generic, we note that our formulation of the BAN PIM given above needs further generalisation to permit some possible future BAN configurations where the MBU as a device may disappear completely, for example, when future smart sensor networks distribute processing and storage functionality between their nodes. Many other questions arise, such as whether to model FESSS as a PIM or PSM level construct.

The construction of customised BANs by composition is not straightforward for a number of reasons. Firstly, the mapping between devices and conditions is many-to-many. In the examples shown above the Epilepsy BAN and the Long QT BAN both provide ECG monitoring services. However in such cases parameters such as sampling frequency and number of leads/electrodes may vary. Even more complex is the question of composition of application components. Combining components may lead to unpredicted conflicts, inconsistencies, performance degradation and perverse behaviour. Composition should be handled in such a way that correct behaviour, reliability, performance and safety of the resulting composed functionality can be assured. Where should the composition be expressed? At implementation time? At the model stage? Should each individual customised instance of a BAN have its own PSM? One approach is to consider not composition of BANs but composition of services. The problem then can be reexpressed as one of service composition and orchestration.

One major implication of implementation of BAN-based m-health services is the scaling issue, both technical and (health) service oriented. Rollout of BAN services across the population would require automated analysis of BAN data since health services could not dedicate staff to observe BAN data from large number of patients 24/7. Different BAN applications would involve different levels of sophistication in the algorithms and inferencing procedures needed to analyse (multiple) biosignal streams and other BAN data together with context information. For many conditions automated analysis would require development and quality assurance of very sophisticated analysis software. This further reinforces the need for the development and application of sound formally based software engineering methods in order to reach the high levels of confidence in the quality and robustness of designs and of implementations derived from them.

Related work, including approaches to some of the problems identified here, includes (Almeida, 2006) (methodological support to distinguish platform independent and platform specific concerns); (Kurtev, 2005) (adaptability of model transformations in model driven engineering); (Eshuis, 2002) (verification of UML activity diagrams); as well as the work of OMG, especially the QVT specification adopted in 2005.

We have described a design and development methodology based on a model driven approach and illustrated it with respect to an m-health application. The methodology is intended to provide a robust method for designing and developing m-health applications. At this early stage the methodology seems promising and we plan to continue to develop and apply it. Here we have described some initial modelling work at PIM and PSM levels; however much work remains to be done to arrive at a first complete application of the methodology.

In the future we plan to complete the modelling work for this application and address the other parts of the trajectory, including transformation based on formal metamodels and verification and validation steps based on model checking and formal testing. Formal methods can also be brought to bear on the question of how to perform safe composition of services and components.

Acknowledgement

This work is part of the Freeband AWARENESS project (http://awareness.freeband.nl). Freeband is sponsored by the Dutch government under contract BSIK 03025. The BAN development conducted by the Awareness and HealthService24 projects is a further evolution of earlier work of the project MobiHealth (IST-2001-36006) (http://www.mobihealth.org) funded by the European Commission. The HealthService24 project (http://www.healthservice24.com) is funded by the European Commission under the eTEN Programme (eTEN-517352). Grateful thanks go to Ed Brinksma, Arend Rensick, Anneke Kleppe and Theo Ruys for their invaluable inputs and feedback.

References

Alexander, H. and Jones, V.M. (1990) *Software Design and Prototyping Using Me Too*, London: Prentice Hall International, ISBN 0-13-820259-1.

Almeida, J.P.A. (2006) 'Model-driven design of distributed applications', PhD Thesis, ISSN 1381-3617-; No. 06-85, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, May.

Bolognesi, T., De Frutos, D., Langerak, R. and Latella, D. (1995) 'Correctness preserving transformations for the early phases of software development', in T. Bolognesi, J. Van de Lagemaat and C.A. Vissers (Eds). *LOTOSphere: Software Development with LOTOS*, Kluwer Academic Publishers, pp.348–368.

Brinksma, E. (1999) 'Formal methods for conformance testing: theory can be practical!', in N. Halbwachs and D. Peled (Eds). *Computer Aided Verification (CAV)*, Vol. 1633 of *Lecture Notes in Computer Science*, Trento, Springer, pp.44–46.

Dokovsky, N., van Halteren, A. and Widya, I.A. (2003) BANip: enabling remote healthcare monitoring with body area networks', *International Workshop on Scientific Engineering of Distributed Java Applications*, November, Luxembourg, Luxembourg.

Eshuis, R. (2002) 'Semantics and verification of UML activity diagrams for workflow modelling', CTIT PhD Thesis Series No. 02-44, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, ISBN: 90-365-1820-2.

Holzmann, G.J. (2003) *The Spin Model Checker: Primer and Reference Manual*, Addison-Wesley, ISBN 0-321-22862-6.

Jones, V., Rensink, A. and Brinksma, E. (2005) 'Modelling mobile health systems: an application of augmented MDA for the extended healthcare enterprise', *Proceedings Ninth IEEE International EDOC Enterprise Computing Conference (EDOC 2005)*, pp.58–69, 19–23 September, Enschede, The Netherlands, IEEE Computer Society, ISBN 0-7695-2441-9, ISSN 1541-7719.

Jones, V.M. (1995) 'Realization of CCR in C', in T. Bolognesi, J. Van de Lagemaat and C.A. Vissers (Eds). *LOTOSphere: Software Development with LOTOS*, Kluwer Academic Publishers, pp.348–368.

Jones, V.M. (1997) *Engineering An Implementation of the OSI CCR Protocol Using the Information Systems Engineering Techniques of Formal Specification and Program Transformation*, University of Twente, CTIT Technical Report Series No. 97-19, ISSN 1381-3625.

Jones, V.M. (2006) 'Model driven development of m-health systems (with a touch of formality)', *PerCom 2006, Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, Pisa – Italy, 13–17 March, *1st IEEE International Workshop on Ubiquitous and Pervasive Health Care (UbiCare 2006)*, 13 March 2006.

Jones, V.M., Bults, R.G.A., Konstantas, D.M. and Vierhout, P.A.M. (2001) 'Healthcare PANs: personal area networks for trauma care and home care', *Proceedings Fourth International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 9–12 September, Aalborg, Denmark, Available at: http://wpmc01.org/, ISBN 87-988568-0-4.

Jones, V.M., Rensink, A., Ruys, T., Brinksma, E. and van Halteren, A. (2004) 'A formal MDA approach for mobile health systems', *Proceedings EWMDA-2, Second European Workshop on Model Driven Architecture (MDA) with an emphasis on Methodologies and Transformations*, September, Canterbury, England.

Kleppe, A., Warmer, J. and Bast, W. (2003) *MDA Explained: The Model Driven Architecture™: Practice and Promise*, Addison Wesley Professional.

Konstantas, D.M., Jones, V.M., Bults, R.G.A. and Herzog, R. (2002a) 'Mobi-Health – innovative 2.5/3G mobile services and applications for healthcare', *11th IST Mobile and Wireless Telecommunications Summit 2002*, 16–19 June, Thessaloniki, Greece.

Konstantas, D.M., Jones, V.M., Bults, R.G.A. and Herzog, R. (2002b) 'Mobi Health-wireless mobile services and applications for healthcare', *International Conference on Telemedicine – Integration of Health Telematics into Medical Practice*, 22–25 September, Regensburg, Germany.

Kurtev, I. (2005) 'Adaptability of model transformations', PhD Thesis, University of Twente, Enschede, The Netherlands, ISBN 90-365-2184-X.

MDA Guide Version 1.0.1, © 2003 (2003) 'OMG, omg/2003-06-01', Available at: http://www.omg.org/docs/omg/03-06-01.pdf.

Meta Object Facility (MOF) 2.0 (2005) *Query/View/Transformation Specification*, Object Management Group, ptc/05-11-01,

Ruys, T.C. (2001) 'Towards effective model checking', PhD Thesis, University of Twente, Enschede, The Netherlands.

Ruys, T.C. and Brinksma, E. (2003) 'Managing the verification trajectory', *Software Tools for Technology Transfer (STTT)*, Vol. 4, No. 2, pp.246–259.

Tretmans, J. and Belinfante, A. (1999) 'Automatic testing with formal methods', *EuroSTAR'99: 7th European International Conference on Software Testing, Analysis and Review*, Barcelona, Spain, 8–12 November, EuroStar Conferences, Galway, Ireland.

van Dam, K., Pitchers, S. and Barnard, M. (2001) 'Body area networks: towards a wearable future', *Proceedings WWRF Kick off Meeting*, Munich, Germany, 6–7 March, Available at: http://www.wireless-world-research.org/.

van Halteren, A., Bults, R.G.A., Widya, I.A., Jones, V.M. and Konstantas, D.M. (2003) 'Mobihealth-wireless body area networks for healthcare', *Proceeding New Generation of Wearable Systems for Ehealth: Towards a Revolution of Citizens' Health and Life Style*, 11–14 December, Il Ciocco Castelvecchio Pascoli Lucca, Tuscany, Italy, pp.121–126.

van Halteren, A., Bults, R.G.A., Widya, I.A., Jones, V.M. and Konstantas, D.M. (2004) 'Mobihealth-wireless body area networks for healthcare', in A. Lymberis and D. de Rossi (Eds). *Wearable eHealth Systems for Personalised Health Management: State of the Art and Future Challenges. Volume 108 Studies in Health Technology and Informatics*, ISBN: 1 58603 449 9.

Widya, I.A., van Halteren, A., Jones, V.M., Bults, R.G.A., Konstantas, D.M. Vierhout, P. and Peuscher, J. (2003) 'Telematic requirements for a mobile and wireless healthcare system derived from enterprise models', *Proceedings IEEE ConTel 2003: 7th International Conference on Telecommunications*, June, Zagreb, Croatia.

Wireless World Research Forum (2001) *The Book of Visions 2001: Visions of the Wireless World*, Version 1.0, December, Available at: http://www.wireless-world-research.org/.

Zimmerman, T.G. (1999) 'Wireless networked devices: a new paradigm for computing and communication', *IBM Systems Journal*, Vol. 38, No. 4.