Formal verification of a telerehabilitation system through an abstraction and refinement approach using Uppaal

Farid Arfi, Anne-lise Courbis, Thomas Lambolais, François Bughin, Maurice Hayot

Received: 3 June 2022 | Revised: 10 April 2023 | Accepted: 17 May 2023

IET Software

DOI: 10.1049/sfw2.12128

The Institution of Engineering and Technology WILEY

CASE STUDY

Formal verification of a telerehabilitation system through an abstraction and refinement approach using UPPAAL

Farid Arfi[1] | **Anne-Lise Courbis**[2] | **Thomas Lambolais**[2] | **François Bughin**[3] | **Maurice Hayot**[3]

[1]University of Montpellier, Montpellier, France

[2]Euromov DHM, University of Montpellier, IMT Mines Ales, Ales, France

[3]PhyMedExp, University of Montpellier, INSERM, CNRS, Montpellier CHRU, Montpellier, France

Correspondence

Anne-Lise Courbis, Ecole des Mines d'Alès, 6 Avenue de Clavières, Alès Cedex 30319, France.
Email: anne-lise.courbis@mines-ales.fr

Funding information

European Regional Development Fund, Grant/Award Number: 2014FR16MOOP006

Abstract

Formal methods are proven techniques that provide a rigorous mathematical basis to software development. In particular, they allow the quality of development to be effectively improved by making accurate and explicit modelling, so that anomalies like ambiguities and incompleteness are identified in the early phases of the software development process. Semi-formal UML models and formal Timed Automata models are used to design a telerehabilitation system through a practical approach based on abstraction and refinement. The formal verification of expected properties of the system is performed by the UPPAAL tool. The motivation of this work is threefold: (i) showing the usefulness of formal methods to satisfy the validation needs of a medical telerehabilitation system; (ii) demonstrating our approach of system analysis through refinements to guide the development of a complex system; and (iii) highlighting, from a real-life experience, the usefulness of models to involve the stakeholders all along the design of a system, from requirements to detailed specifications.

KEYWORDS

formal specification, formal verification, health care, software engineering, software reliability, unified modelling language

1 | INTRODUCTION

Over these last decades, telerehabilitation has been proven to be increasingly useful for many chronic diseases, such as chronic obstructive pulmonary disease (COPD) or obstructive sleep apnoea (OSA). Telerehabilitation consists of using Information and Communication Technologies (ICT) to support remote rehabilitation, which is defined as a set of interventions designed to facilitate the process of recovery from injury, illness, or disease to as normal a condition as possible.

Therapeutic studies published on COPD and OSA show that telerehabilitation is possible and safe, and can achieve an equivalent improvement in outcomes compared to traditional centre-based rehabilitation [1, 2]. Therefore, these studies encourage the development of telerehabilitation systems, especially since, according to the World Health Organization's estimates, COPD is predicted to become the third non communicable disease leading cause of death by 2030. In accordance with these recommendations, the m-Rehab project[1] aims at developping a telerehabilitation application focusing on the management of COPD and OSA patients.

The primary objective of software engineering is to enable the development of systems that meet the functional needs of the specification but also non-functional properties (reliability, performance, maintenance etc.) and, at the same time, keeping costs and deadlines under control. However, one of the main problems encountered during the development is the missing of a comprehensive requirements document faithfully

[1]The m-Rehab project is supported by Occitanie region—FEDER (*Fonds Européen de DEveloppement Régional*). Partners are laboratories: *Physiologie et Médecine Expérimentale du Cœur et des Muscles* (PhyMedExp, Univ. Montpellier, Fr), Montpellier Research in Management (MRM, Univ. Montpellier, Fr), Euromov-Digital Health in Motion (Euromov-DHM, Univ. Montpellier, IMT mines Ales, Fr), and KORIAN group (Fr).

transcribing stakeholders' needs. Indeed, anomalies such as ambiguities or inconsistencies are generally detected later during the testing or integration phases of the software, or even during its use, which leads to additional costs. These difficulties are important in the case of healthcare systems because they implement medical practices, such as medical guidelines or pathways, which mainly involve human know-how and human interactions between physicians and patients. It is therefore necessary to combine the informal specification phase with a formal one in order to carry out a rigorous analysis throughout the development cycle.

We have chosen UML as a semi-formal language for modelling m-Rehab requirements because UML diagrams, mainly the class diagrams, state machines and sequence diagrams, are easily understood by non-experts and provide a good visual support for discussing and refining requirements.

Moreover, in order to cope with the quantity and complexity of the specifications, we propose a methodology based on abstraction and refinement paradigms that are guides for both the modelling phase and the verification one. The refinement paradigm is well known in the formal community. However, there is little work showing how it is implemented during the design phase on a real health system both from theoretical and practical points of view, nor work highlighting the benefits of this approach for the stakeholders involved in the application design.

The main contribution of this article is threefold:

- promoting formal approaches during the design of safe medical software systems.
- combining formal and informal modelling through abstraction and refinement to improve the requirement and detailed specification phases in partnership with stakeholders involved in the telerehabilitation system definition and implementation.
- pointing out benefits of this approach, based on feedbacks of stakeholders of a real-life application development.

The rest of this paper is organised as follows: Section 2 presents works focusing on formal modelling and verification of medical systems and highlights the gap between these works and our proposal. We present in Section 3 our approach of abstraction and refinement applied for both modelling and verifying complex systems.

Section 4 focuses on the elementary concepts of the formalism chosen for formal modelling and verification. In Section 5, we present the m-Rehab telerehabilitation application in an informal way and we focus on one of its processes, the nutrition pathway, selected to demonstrate our approach. We also highlight the main phases of abstraction and refinements that have been defined to face the complexity of the system.

Section 6 points out how our modelling and verification approach is applied on the nutrition process, following the predefined abstraction and refinement steps. Finally, Section 7 reports the strengths and weaknesses of the approach. We conclude this paper and provide future directions in Section 8.

2 | RELATED WORK

As the field of formal verification is very broad, we initially focused on its use in the health domain. Among these works, we distinguish those dealing with single-process systems (guidelines or connected devices) and those dealing with complex systems involving multiple processes, whether for single user (a physician) or, in the case of telerehabilitation, for multiple remote users (patients and physicians). We present a set of representative works taking into account this classification and depending on the goal of cited works, we point out the gap between their proposal and our purpose. We then briefly review the well-known notions of abstraction and refinement both for semi-formal (UML) and formal models as they are the foundations of our approach.

2.1 | Formal approaches applied to medical domain

Formal methods are increasingly used in the medical domain for modelling, verification, validation and even for code generation. A systematic literature review has been performed in 2018 precisely in this area [3]. After enriching the bibliography, we observe that most of these methods are still applied on critical devices that require safe design methods and some of them for modelling or verifying simple processes (guidelines or single-process pathways). However, there is relatively little works to guide the design and the verification of telehealth systems, including the crucial phase of requirement collect.

Telehealth is a general term that covers a broad range of healthcare applications that can be performed remotely using ICT infrastructure in the different fields of medicine such as telemedicine, telerehabilitation or homecare.

Such applications are usually characterised by multiple processes which have to be synchronised, automatically or by a human action, and which are supervised by many stakeholders involving various professions (physicians, physiotherapists, nurses etc.) whose requirements are related to their speciality. Developing such applications requires a strong methodology supported by efficient tools and facilities to discuss experts' requirements and find solutions when some properties are not satisfied.

Representative works are listed in the following according to their purpose.

2.1.1 | Formal modelling and verification of connected devices and guidelines

Many are the works that use timed automata for modelling medical devices and the UPPAAL tool for verification, such as, for instance, the pacemaker [4–8] and the infusion pump [9, 10]. With regards to modelling single-process, some authors [11, 12] have studied a clinical pathway of the COPD disease. They use the Petri net (resp. Bigraphs) formalism for modelling, and the Maude tool for verification. In other works such

as [13, 14] the timed automata model and UPPAAL tool are used to deal with the problem of improving the quality of guidelines.

All these work are interesting from a formal point of view. However, they are limited in analysing a single formal process and they do not address the methodology for modelling complex medical processes starting from requirements expressed by medical stakeholders.

2.1.2 | Formal modelling of telehealth systems without verification

Some authors propose frameworks only for formal modelling of interaction processes in telehealth systems, without addressing their formal verification, or pointing out what properties may be verified. For example, in Ref. [15], the author proposes a formal axiomatic model for specifying the requirement actions of a healthcare system. In Ref. [16] an extension of timed automata with multiple time characteristics is used to model the components interaction in a telehealth system. In Ref. [17] a reconfigurable Petri net model is proposed to handle changes of services in a distributed telehealth environment. All these works suffer from a lack: they do not point out what kind of verification is performed and what is the purpose of formal modelling with respect to expected properties expressed by medical stakeholders.

2.1.3 | Correctness and safety of telehealth systems

One of the major concerns when designing telehealth systems is to ensure the correctness and the safety of processes. Formal analyses have been applied in some safety-critical telehealth systems such as in Ref. [18] in order to confirm robustness, accuracy and efficiency of the system modelled in Vienna Development Method Specification Language (VDM-SL). Work of Ref. [19] aims at ensuring reliability aspects of the standard Fast Healthcare Interoperability Resources modelled by the Continuous Time Markov Chain model. Another aspect of telehealth system verification is proposed by Ref. [20] in order to check the safety of the communication protocols of a telehealth system modelled by high-level Petri nets.

These works face one of our issue, verifying processes involved in a telehealth system. However, they do not point out how modelling and verification activities are integrated in the early phases of the system development, leading to the revision of requirements in collaboration with stakeholders.

2.1.4 | Data security and privacy in telehealth systems

Telehealth applications have raised serious concerns regarding security and privacy of health data. In fact, the increasing demands of accessing health data highlight critical questions and challenges concerning the confidentiality of electronic patient records and their access efficiency. Some authors address this problem and provide secure and efficient access to electronic patient records using formal methods. In Ref. [21], the authors propose a new protocol formally verified based on a model named Casper and its verification tool Casper/FDR2. In Ref. [22], it is the timed automata that are used for modelling and the UPPAAL tool for verification. In Ref. [23], the authors propose a verification tool called Automated Validation of Internet Security Protocol and Applications (Avispa) which is based on formal notations. These works are of high interest. However, the design and the verification of all the processes involved in a telerehabilitation system is out of their concerns.

2.1.5 | Quality of services

Formal methods have also been applied for improving the quality of services in telehealth applications. In Ref. [24], the authors propose an ICT application for better managing the coordination of care services to patients at home within homecare organisations. The challenge of their application is to enable the scheduling of the care plans over a common available resources, that is, the human resources and the medical constraints, in order to quantify the effectiveness of the provided services. They use the timed automata formalism for modelling care plans and, the UPPAAL tool to check the consistency and schedulability of a care plan. In Ref. [25], the same context is dealt using recursive ECATNets (timed Petri nets augmented with data types manipulation) for modelling and the Maude tool for verification. These works address optimisation issues and do not deal with design of multi-process applications.

2.1.6 | Formal approaches and medical software development

In the context of software development, formal methods are intended to systematically introduce a rigorous approach in all phases of design. However, the acceptance and regular use of formal methods is still far less than what proponents would want. Formal method researchers constantly try to convince the developers for the usefulness of formal methods, in particular to design safety-critical systems as it should be for the medical domain. For the latter, few authors have been interested for this purpose. For instance, in Ref. [26], the Z formalism is used at the early stage of a telehealth application development to verify some safety properties. The authors then show the benefits of a formal design against a traditional one during the development process of this application. In Ref. [27], the authors point out some lacks in Z formalism and propose the use of VDM-SL specification language for modelling the same telehealth application.

This review points out that little work focuses on the overall process for developing medical systems. For the best of our knowledge, there is no conceptual work based on a real experience in developing a telerehabilitation application that

shows how to combine requirements collection, detailed specification modelling and verification by involving medical stakeholders. Our proposal aims to fill this gap.

2.2 | Abstraction and refinements processes

Abstraction is a well-known paradigm widely developed since the seventies for programme development and verification: '*abstract interpretation formalizes the idea that formal proof can be done at some level of abstraction where irrelevant details about the semantics and the specification are ignored*' [28]. It is a well-known mechanism to cope with the complexity of systems. During the development of specifications leading to define a model close to the implementation, abstract models must be refined.

Refinement is also a well-known paradigm to face the complexity for modelling and analysing complex systems. For example, in Ref. [29] the authors point out the benefits of refinement both on data and processes. We have demonstrated in previous work [30, 31] that refinement and extension are mechanisms to support the incremental construction of models and also to formally verify the preservation of safety and liveness properties during refinements.

Refinement may also be applied on semi-formal languages such as UML [30, 32]. Formal languages such as B [33], Event-B [34] or Z [35] are based on the refinement theory, meaning that they allow refinements to be explicitly defined by modellers and automatic checking to be performed.

Many work focused on the translation of UML behavioural models (state charts and sequence diagrams) into formalisms integrating refinement, such as UML-B [36] or, UML transformation into Object-Z [37]. Some works point out the use of refinement in medical applications such as Event-B for verifying a pacemaker [38] or a haemodialysis machine [39, 40].

Abstraction and refinement paradigms are often considered difficult to handle mainly because they require advanced mathematical skills and tools. Hence, they are mainly reserved to critical systems such as automotive, railways or avionic systems.

Unlike works that use a specification language to support a refinement by construction such as Event-B, our approach is to deal with informal refinement at the early stage of design, during the UML design, in order to be able to switch between informal and formal specifications that may be discussed with the experts of the system under study. Furthermore, we focus on the verification of explicit properties expressed by experts (medical stakeholders), independently from the system model.

3 | OVERVIEW OF OUR APPROACH BASED ON ABSTRACTION AND REFINEMENT

Our goal is therefore to define a new approach allowing specifications to be verified using abstraction and refinement in such a way that medical and care stakeholders may be involved in the process. We demonstrate through a target process of a real application how these paradigms can be applied and what are the benefits.

Figure 1 gives an overview of the approach we have followed. The informal specifications are organised in such a way as to identify the processes structuring the application and their high-level interactions. One process is selected (the Target process in Figure 1) to be analysed first in an abstract model and then through refinements. Other processes can be represented in a very abstract way, as processes with only start and stop states, even if they have already been defined in details by stakeholders.

The target process is then modelled in UML, in the form of class diagrams and state machines. These diagrams are used

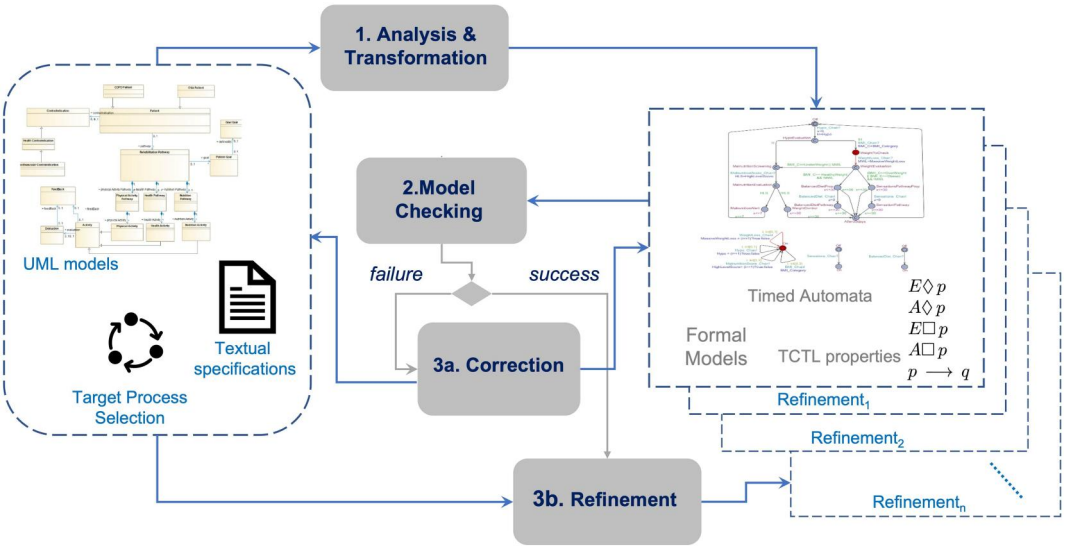


FIGURE 1 Overview of our methodology: formal verification through abstraction and refinement processes.

as a support for discussing with experts to understand and refine the process. The natural language is used to define a set of expected properties. These models are then transformed into timed automata formal models. The originality of the proposed approach is to consider an initial formal model at a higher abstraction level than defined in the experts' description: they are first built in a very abstract way (activity 1 in Figure 1) and details are gradually added after the verification stage.

When the properties are verified using a model checking technique (activity 2 in Figure 1), the formal model is refined (activity 3b), otherwise a discussion has to be engaged with the expert to analyse the failure: this may lead to correct the specification or the property (activity 3a). Note that the chosen property language guarantees that properties are either satisfied or not. A new formal model is then generated at the same level of abstraction, until all properties are verified. The process of modelling-refinement-correction is repeated until a detailed level of specification is obtained which is close to the algorithmic level.

Even if abstraction and refinement are well-known concepts within the computer science community, their implementation cannot be applied in a simple way, especially in a context where the stakeholders are doctors, unaccustomed to such reasoning. We have thus experimented this approach of iterative refinements of formal models correlated with abstract views of semi-formal modelling and pointed out how corrections on both models and textual requirements is helpful to enhance requirements and to involve stakeholders.

4 | PRELIMINARIES

4.1 | UPPAAL timed automata

UPPAAL is a tool for modelling, verifying and simulating real time systems. This tool which is based on the theory of timed automata and temporal logics has been successfully used in many industrial case studies like medical domain as referred in Section 2. In this section, we give a short informal presentation to the major UPPAAL concepts used in this paper. We refer the reader to [41] for an introduction to the theory of timed automata.

In UPPAAL, a system is modelled as a closed network of parallel timed automata called processes [42]. These processes can be seen as finite state machines extended with clocks to model the time progression, and variables of simple types (boolean and bounded integers), or structured types (arrays). The specification of a system in UPPAAL consists of (1) a set of timed automata which define processes, (2) a set of local or global declarations of clocks, variables, channels, constants, and functions and (3) a system declaration composed of parallel processes. The global variables can be used for communication between processes. Unlike asynchronous communication, synchronous one is expressed between processes while synchronisation on the channels is taking place between them.

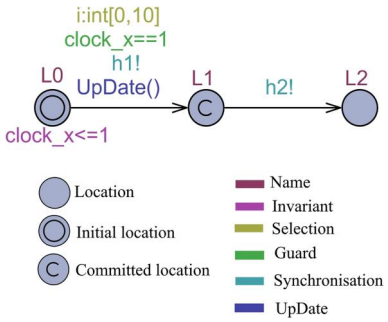


FIGURE 2 A simple example of a timed automaton in UPPAAL.

A timed automaton is composed of a set of locations and edges. Figure 2 gives a brief example for the graphical representation in UPPAAL of a timed automaton whose main concepts are explained in the following.

- Locations are shown as simple circles, except the initial one which is shown by a double circle. Each location may have a name. A location may be labelled by an invariant condition (boolean expression) expressed over clocks and variables which must be satisfied as long as time elapses in this location. Locations can also be labelled with one of the attributes committed (c) or urgent (u) respectively. In both cases, no time delay is possible before this location is left. A committed location has also a priority over other possible edges.
- Edges are represented by arrows. An edge may be labelled with: (1) a selection which is used to select a value among a given range, (2) a guard which is a boolean expression over clocks and variables indicating when the edge is enabled and can be fired, (3) updates of clocks and variables which are executed when the edge is fired and (4) synchronisation of the form h! or h? over a channel h. By default, a channel declaration is unicast. Then, an edge labelled with h! in one of the processes (the sender) may synchronise with an edge labelled with h? in another process (the receiver) leading that both edges are enabled. If several combinations are enabled, a synchronisation pair is selected according to a non-deterministic choice. However, for a broadcast channel h, an enabled edge labelled with h! may synchronise with an arbitrary number of enabled edges labelled with h? in other processes. If there are no receivers, then the sender can still execute the h! action, that is, broadcast sending is never blocking.

A state of the system is expressed by the current location for each automaton and the values of clocks and variables.

4.2 | UPPAAL property specification language

To specify properties, UPPAAL [42] uses a subset of Timed Computation Tree Logic, where formula expression primitives are variables, clocks and also locations names. There are two

quantifiers used in the outer-most of a formula: 'for every path' (A), or 'there is a path' (E) followed by all states ($\Box$) or at least one state ($\Diamond$) quantifiers over the considered path. The leads-to operator $\rightarrow$ can also be used between two formulae in UPPAAL. However, all the previous forms of specification cannot be nested. The properties can be classified into reachability, safety and liveness [42] which are defined by formulae over the modelling paths. Properties expressed in Section 5.4 are of the two first types:

- Reachability properties: starting from the initial state, they are used to check whether a given formula ϕ can be satisfied by some reachable state. $E\Diamond\ \phi$ is the syntax used in UPPAAL for this property.
- Safety properties are used to verify that something unwanted will never happen. In UPPAAL, this kind of properties is formulated positively, for example, something good is always (invariantly) true. There are two forms of writing these properties. $A\Box\ \phi$ means that a given formula ϕ should be true in every reachable locations, and $E\Box\ \phi$ expresses that there should exist a path such that ϕ is always true for its locations.

5 | m-Rehab CASE STUDY

We have been involved in the early stages of the design of the m-Rehab application to support the specification process before the implementation phase. The specification has involved many stakeholders both in the medical field and in the fields of physical activity and health data management. This process resulted in a textual document over 500 pages and a UML model that includes more than 300 concepts. It is a challenge to verify such a specification, and a significant risk to wait for the implementation of the application, even partial, to check its consistency.

We give a general presentation of the m-Rehab application and an overview of its UML modelling. Then, the nutrition pathway is detailed since it is the target process chosen to demonstrate our modelling and verification approach.

5.1 | Presentation of m-Rehab

m-Rehab is a telerehabilitation system for BPCO and OSA patients which aims at motivating patients to perform appropriate activities whose results are under the supervision of a team of experts. m-Rehab offers patients several care pathways in order to improve their health condition and the way to manage their disease (Figure 3). The pathways concern three types of activities: health, nutrition and physical activities. These activities are customised according to the patients' profile defined during the inscription process and updated all over their telerehabilitation process. Patients are equipped with Wi-Fi devices such as watches, scales, podometers, blood pressure monitors and positive airway pressure monitors, to automatically get relevant data. Every pathway is structured into activities or challenges for which a personal feedback is systematically delivered to patients through comments or progress charts.

The pathways dedicated to health, nutrition, physical activities as well as inscription and administrative processes have been precisely defined through interviews conducted with experts and formalised both in textual form and UML models.

5.2 | UML modelling

The modelling phase has started in the same time than the writing of the specifications. Its main interest is to provide a synthetic view of concepts and processes, which can be easily

FIGURE 3 Overview of the m-Rehab system.

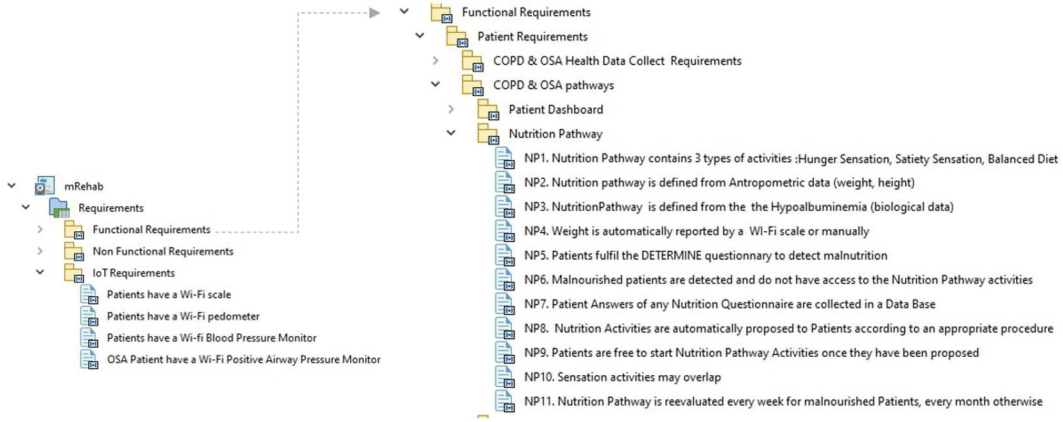


FIGURE 4 Requirements of m-Rehab (excerpt of functional requirements).

discussed by the involved experts. We have followed the early steps of the system engineering approach [43] to highlight the expert needs and domain properties, translated into system requirements. Requirements are organised into three packages (Figure 4, left part): functional, nonfunctional and IoT requirements. For example, IoT requirements state that patients are equipped with a Wi-Fi scale. An excerpt of functional requirements related to the nutrition pathway is given in the right part of the same figure. For example, there are three types of activities proposed in the nutrition pathway: hunger sensation, satiety sensation and balanced diet (NP1); patients fulfil the DETERMINE questionnaire to detect malnutrition (NP5).

Requirements are guidelines to define logical components of the system and their associated processes. We have used UML class diagrams to highlight both concepts and their relationships. The class diagram was used during plenary meetings in order to point out the progress of the project, share the knowledge of the expert who was involved in the description of his/her expertise to the entire committee. This diagram was still under construction until the end of the specification phase. For example, Figure 5 is an excerpt of the class diagrams related to the classification of m-Rehab pathways. It shows the two kinds of patients (COPD and OSA). A patient may have several contraindications (for example a cardiovascular one) for accessing to some activities. This class diagram points out that a telerehabilitation pathway requires that patients define their personal goals (at least one), these goals being defined through a specific questionnaire. We find again the three types of care pathways defining m-Rehab (administrative processes are not represented here), and their common features: they are constituted by activities and every activity leads to an evaluation and a feedback.

5.3 | The Nutrition Pathway

The appropriate pathway fitting the patient's profile is periodically proposed all over the rehabilitation phase. It is defined according to an algorithm set up by the nutritionist of the m-Rehab project. This algorithm (Figure 6) fits the NP8 requirements listed in Figure 4. It analyses the patient's biological data such as hypoalbuminemia (NP3 requirement) and his/her body mass index (BMI) computed from his/her height and weight (NP2 requirement): $\mathrm{BMI} = \frac{weight(kg)}{height(m)^2}$

The BMI is conventionally classified into four categories (see Table 1).

In case of a massive weight loss (criteria not detailed here), hypoalbuminemia or underweight, a specific questionnaire (NP5 requirement), defined by the American Academy of Family Physicians and the American Dietetic Association, named DETERMINE, is fulfilled by the patient in order to detect a possible malnutrition. In this case (left branch of the process in Figure 6), a notification is sent to the care manager and to the patient to inform him/her that the Nutrition Pathway is not accessible (NP6 requirement), since at present, no specific programme is implemented in m-Rehab to deal with malnutrition. When the patient is detected as being overweight or obese, but not in a massive weight loss (right branch of the process in Figure 6), the appropriate activity is the Sensation Pathway. In other situations, the patient may follow the Balance Diet Pathway. Such a diagram was a support to share the knowledge of the nutritionist with other stakeholders of the project. It was a support to validate the requirements and points out that the Nutrition Pathway has to be closed for malnourished patients. This was not known to other medical stakeholders of the project.

We go into details of the Sensation Pathway in order to illustrate our approach of refinement and get a relevant model for formal verification. This process is divided into two processes: Hunger and Satiety. The patient must start the Nutrition Pathway with the Hunger Pathway (Figure 7): during 3 days, he/she fulfils questionnaires during the meals. Before starting, explanations are given about the goal and the procedure to be followed in the activity. After 3 days, answers of the questionnaires are analysed. If there is not enough answers, the patient is offered to start again the activity. In other cases, his/her profile is determined depending on the maximum occurrences of four types of situations identified: no hunger sensation, slight sensation, moderate to severe sensation and

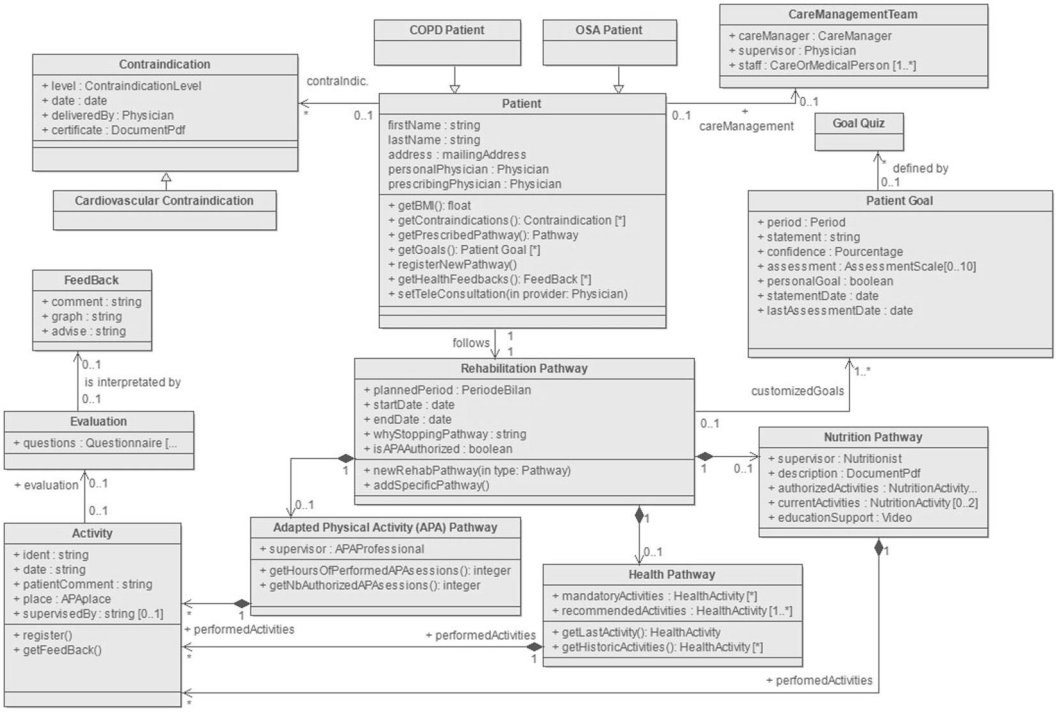


FIGURE 5 m-Rehab class diagram (excerpt) pointing out rehabilitation pathways.

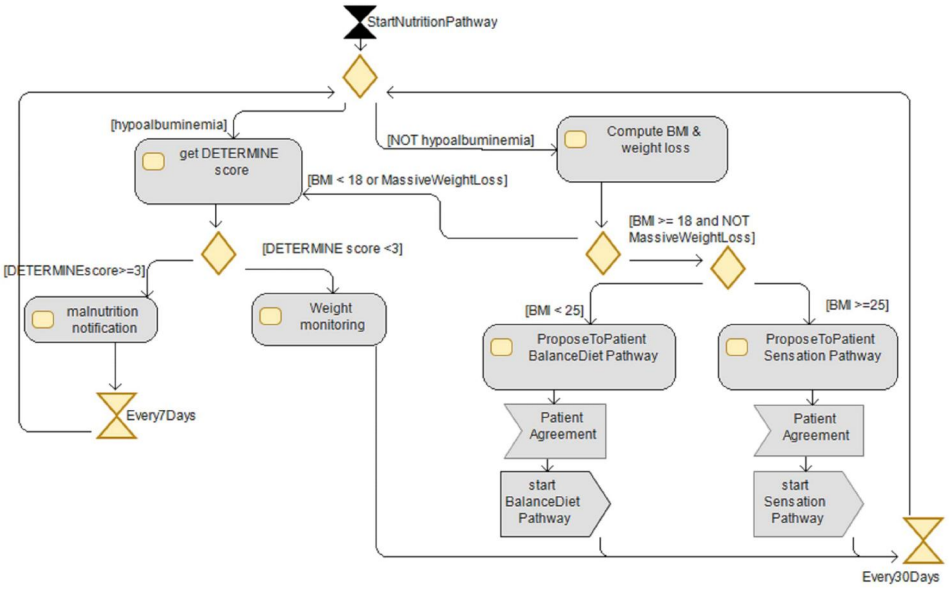


FIGURE 6 Algorithm to determine patients' appropriate nutrition pathway.

TABLE 1 Body mass index (BMI) categories.

BMI	BMI interpretation	BMI category
Less than 18	Underweight	0
Between 18 and 25	Normal (healthy weight)	1
Between 25 and 30	Overweight	2
Over 30	Obese	3

painful sensation. Patients who have no hunger sensation or a light sensation have to start again the Hunger Pathway. Other ones may start the Satiety Pathway and also continue at will the Hunger Pathway. Note that according to the decision procedure defined in Figure 6, the patient's profile is re-evaluated every 7 days for malnourished patients, and 30 days otherwise (NP11 requirements), in order to define the appropriate nutrition activity.

5.4 | Properties of the Nutrition Pathway

We have conducted discussions with the nutritionist to set up the textual specifications and the UML models. This led us to state some properties related to the Nutrition Pathway, and its sub-processes, that was not initially expressed. They have been added in a textual form, in order to have a better understanding of the expert's logic. We have selected five of them:

[P_1] *'A malnourished patient may be overweight'.* (reachability property)
[P_2] *'A malnourished patient should not have access to the Sensation Pathway nor the Balance Diet Pathway'.* (safety property)
[P_3] *'A patient can always begin the Sensation Pathway 1 week after being offered'.* (reachability property)
[P_4] *'A patient can follow both Hunger and Satiety pathways at the same time'.* (reachability property)
[P_5] *'The Satiety pathway can be opened only when the Hunger pathway has been performed at least once'.* (safety property)

We will see in Section 6 how these properties are formally expressed and verified in UPPAAL.

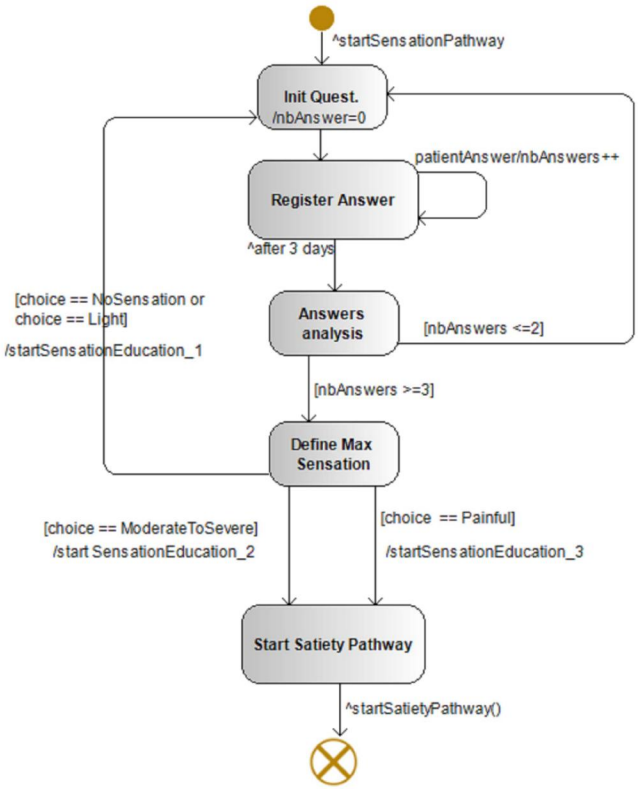


FIGURE 7 Questionnaire-based algorithm of the hunger pathway.

5.5 | Refinements of the Nutrition Pathway

In order to formalise the Nutrition Pathway before verification, we apply our methodology represented in Figure 1. The starting point of the formal modelling process is an abstract view pointing out that the Nutrition Pathway interacts with another process named Abstract Patient Data which is expected to deliver patients' data (Figure 8). At this step, we assume that data are available and have been set up during the Administrative Pathway and the Health Pathway performed before the Nutrition Pathway.

We give in this section the sequence of refinements we have followed in Section 6 to set up the formal model of the Nutrition process and make the appropriate verification.

5.5.1 | First refinement of the Nutrition Pathway

The Abstract Nutrition Pathway is refined into a Global Nutrition Pathway (Figure 9) taking into account the initial specification that define four data required from the Patient Data Process: massive weight loss, hypoalbuminemia, BMI and malnutrition score. These data are mentioned in the textual and UML specifications as well as the way to get or compute them. However, at this abstraction level, they are modelled as abstract knowledge and their computation is ignored. Their value is randomly selected by the Abstract Patient Nutrition Data into an interval of appropriate values. Let us note that the Global Nutrition Pathway does not require the precise value of BMI, but only its category defined in Table 1. Therefore, the BMI category can replace the BMI in the processes modelling.

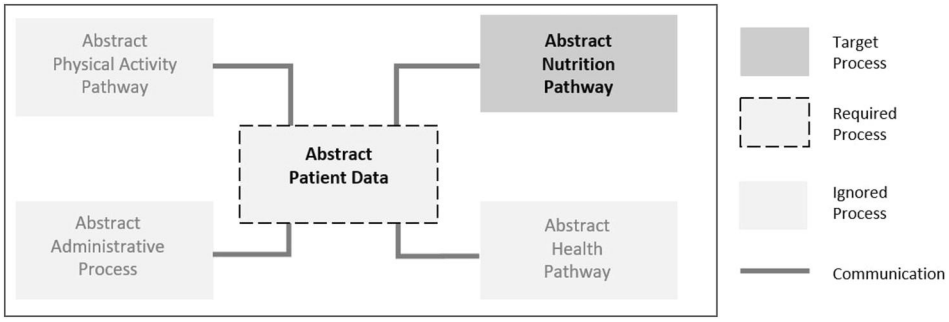


FIGURE 8 Abstract view of the four m-Rehab processes connected to the abstract patient data.

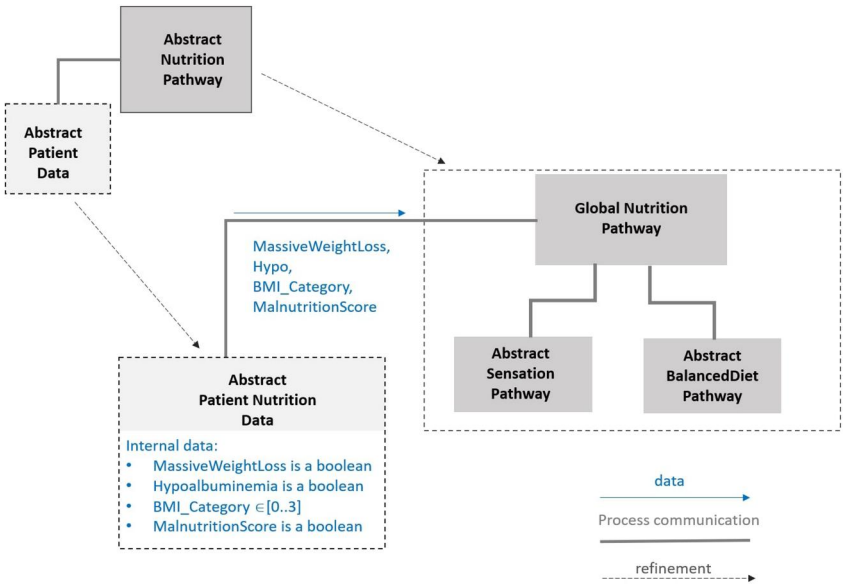


FIGURE 9 Refinement of the abstract nutrition pathway and the abstract patient data process.

The Global Nutrition Pathway supervises two sub-processes that are not described at this step.

5.5.2 | Refinement of the BMI category computation process

We focus on the hypothesis considering the BMI category as an abstract knowledge. In fact, BMI category is defined according to concrete data that must be delivered by the Patient Data Pathway, that is, the patient's weight and height. The random process to set BMI category is thus refined (Figure 10) into two steps: the first one computes the BMI value according to the patient's data (function f1) and the second one associates the BMI category (function f2) in accordance with Table 1. Let us note that this refinement does not impact the data exchanged through the synchronisation with the Global Nutrition Pathway.

5.5.3 | Refinement of the Sensation Pathway

The last refinement considers the Sensation Pathway. As defined in the specification, the Sensation Pathway is starting with a Hunger Pathway (Figure 7) that may start the Satiety Pathway. This refinement (Figure 11) leads to make an extension of the Patient Nutrition Data in order to introduce new data relating to the nutrition questionnaires (see TabQuiz[ ] variable in Figure 11). These data are again considered as abstract knowledge without considering the way to get them. This will be the purpose of future refinements that are not addressed here.

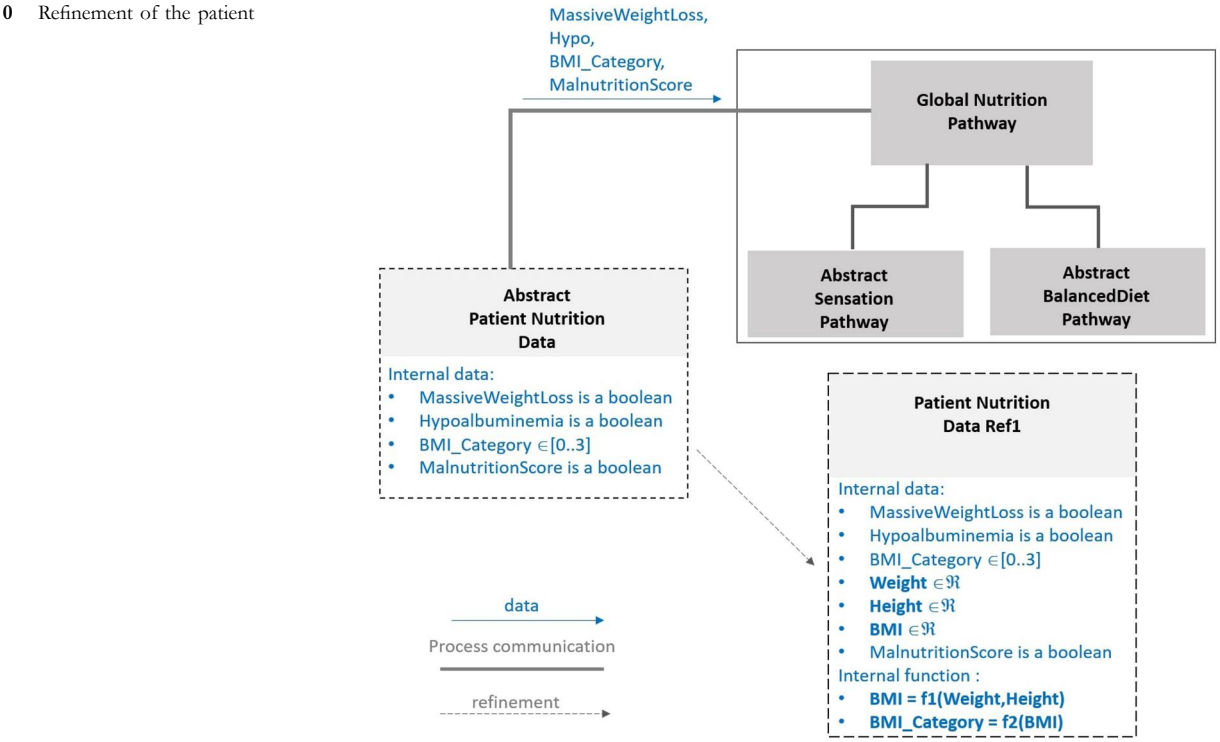
Having defined the sequence of refinements set up for dealing with the Nutrition Pathway, we present in the next section some preliminaries about the formalism and the tool we have used for formal verification, followed by the section about their application to our case study.

6 | FORMAL MODELLING AND VERIFICATION OF THE NUTRITION PATHWAY

Most of the time, even if specifications have been analysed and partially modelled in UML, they suffer from: (1) incompleteness, that is, insufficient or missing information; (2) inconsistency, for example, the Nutrition Pathway may result in different/conflicting decision given the same patient data, (3) ambiguity, for example, there are several interpretations of some term used in the Nutrition Pathway and (4) redundancy, for example, there might be elements of the nutrition description that can be removed without any effect in the resulting recommendations and thus can be skipped for the formal model interpretation and the implementation of the application. Although, several anomalies become apparent during the UML representation of the Nutrition Pathways, others may persist given the semi-formal nature of UML. Therefore, the chosen timed automata formalism should be able to assist the designer in fixing these problems. In fact, during the formal modelling phases, we have found several anomalies. Here is an example of ambiguity problem:

> In the Hunger Pathway, the patient completes a food questionnaire indicating his/her habits during food consumptions for a given period of time. Then, depending on his/her *majority* habit, a treatment is proposed. The ambiguity here is which treatment should be given to the patient if there is no specific habit which owns the *majority*. In this case: should one be randomly selected among those with the same maximum score? Should the patient be asked to repeat the questionnaire until a typical consumption habit owns the majority?

The ambiguity has been corrected by explaining the problem to the nutritionist who gave the initial specification: the recommended solution was to repeat the questionnaire.



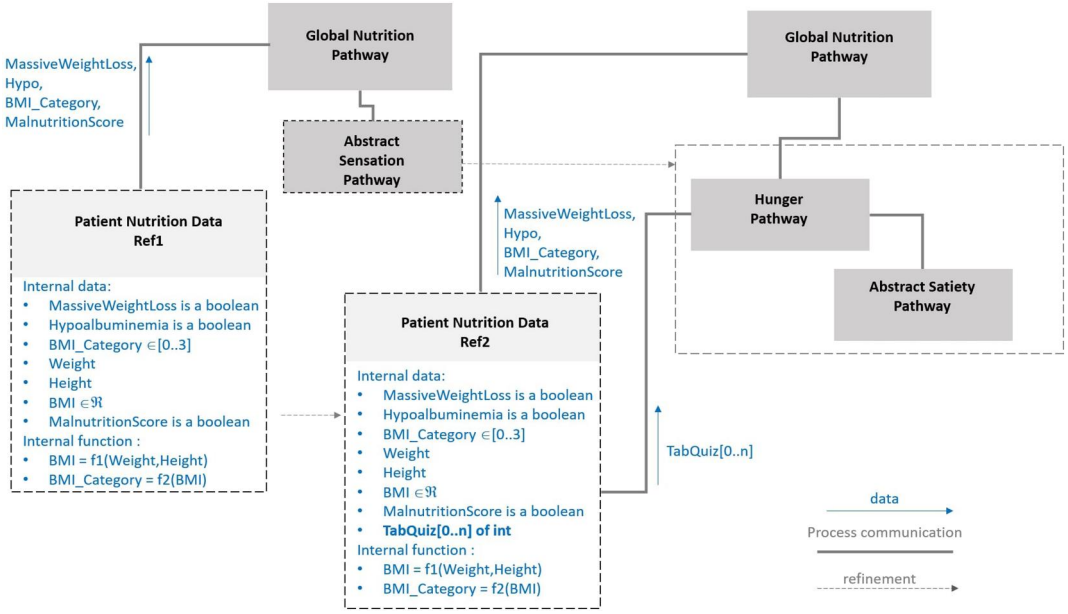


FIGURE 10 Refinement of the patient nutrition data.

FIGURE 11 Refinement of the abstract Sensation Pathway and patient nutrition data Ref1.

Basically, the first transformation step performs the initial structural validation of the Nutrition Pathway by representing it using the chosen formalism as an abstract process. Then, a gradual refinement technique from the abstract modelling is introduced in order to progressively reducing the abstraction level in a top-down design approach.

When the Nutrition Pathway is formally modelled, it is possible to perform an automatic verification of the expected properties which is another way to detect anomalies in the processes definition. We have presented in Section 5.4 some of them selected for this paper.

As shown in Figure 1, the non-satisfaction of a property leads to the modification of the specification and the corresponding formal modelling.

In the following, we describe the three abstract models obtained through the three refinement steps presented in Sections 5.5. We start with a synthetic view of the Nutrition Pathway. Then, we give an example of a process refinement

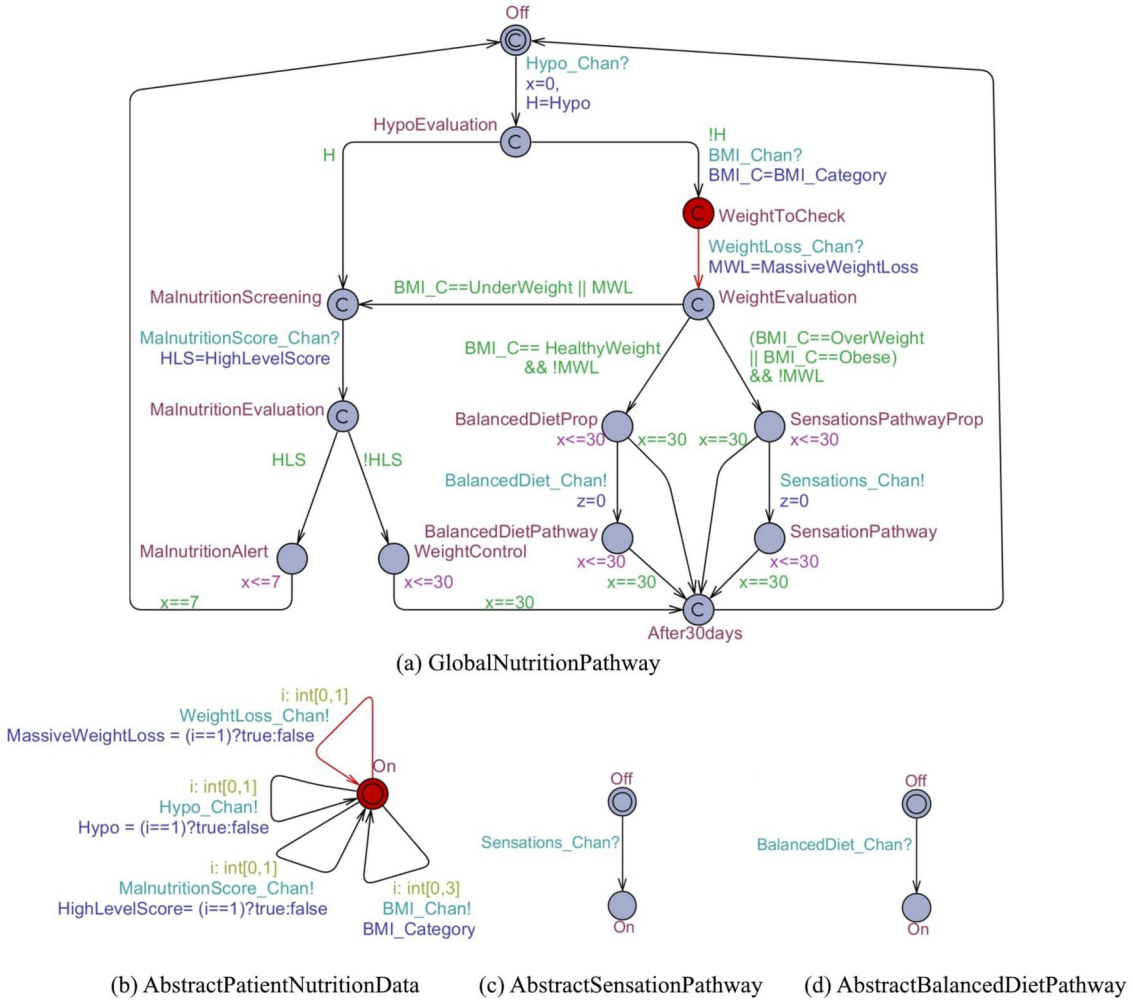


FIGURE 12 Abstract modelling of the Nutrition Pathway.

based on data refinement. Finally, we show a partial refinement of a sub-process of the Nutrition Pathway.

6.1 | Abstract modelling and verification of the Nutrition Pathway

Processes implied in the Nutrition Pathway are modelled by four UPPAAL timed automata (Figure 12):

GlobalNutritionPathway, AbstractPatientNutritionData, AbstractSensationPathway and AbstractBalancedDietPathway. The timed automata exchange data via shared global variables whose declarations are given in Figure 13.

The GlobalNutritionPathway automaton (Figure 12a) models the determination process of Nutrition Pathway specified by UML state machine in Figure 6. It is based on patient data obtained through a synchronous communication with the AbstractPatientNutritionData automaton (Figure 12b). At this level of abstraction, the patient data are modelled by an abstraction on integer or boolean variables. According to the provided data, the GlobalNutritionPathway automaton can be synchronised with one of the two timed automata: BalancedDietPathway (Figure 12c) or SensationPathway (Figure 12d) within 30 days. The 30-day constraint is represented by the invariant $x \leq 30$ in the BalancedDietProp and SensationPathwayProp locations of the GlobalNutritionPathway.

As an example of synchronous data communication, let us consider the case of the synchronisation channel WeightLoss_Chan between the two automata in Figure 12a,b achieved through the two red transitions. The communication via this channel allows PatientNutritionData to write the data about patient's massive weight loss, which is boolean, to the shared variable MassiveWeightLoss. Thereby, GlobalNutritionPathway receives this information on its local variable MWL. MassiveWeightLoss is chosen by the AbstractPatientNutritionData automaton in an abstract way by an arbitrary choice, true or false according to a random value of i, $i \in \{0, 1\}$.

The level of details of these processes, even if they are very abstract, allows propertied P_1, P_2 and P_3 given in the Section 5.4 to be expressed. P_4 and P_5 require one of the sub-processes to be refined, that will be done in a next step. Properties P_1, P_2 and P_3 are expressed by the following UPPAAL formulae:

```
// global declarations
chan Hypo_Chan, BMI_Chan, WeightLoss_Chan, Malnutrition_Chan, BalancedDiet_Chan, Sensations_Chan;
bool Hypo, MassiveWeightLoss, HighLevelScore;
int BMI_Category;
const int UnderWeight=0, HealthyWeight=1, OverWeight=2, Obese=3;

// local declarations: GlobalNutritionPathway.
bool H, MWL, HLS;
int BMI_C;
clock x,z;
```

FIGURE 13 Declaration of synchronisation channels, variables and clocks.

- P_1: $E\Diamond$ GlobalNutritionPathway.MalnutritionAlert && GlobalNutritionPathway.BMI_C == Obese.
- P_2: $A\Box$ GlobalNutritionPathway.MalnutritionAlert imply (AbstractBalancedDietPathway.Off && AbstractSensationPathway.Off).
- P_3: $E\Diamond$ GlobalNutritionPathway.SensationPathway && GlobalNutritionPathway.x − GlobalNutritionPathway.z == 7.

The verification in UPPAAL concludes that both properties P_1 and P_3 are satisfied while the property P_2 is not. Figure 14 shows a sequence diagram of a counter-example generated by UPPAAL where the property P_2 is not satisfied. In other words, the GlobalNutritionPathway automaton can be in the location MalnutritionAlert which means that the patient is malnourished while at least one of the automata AbstractBalancedDietPathway or AbstractSensationPathway is not in the location Off, which means that the corresponding pathway is open to the patient, this is the case of the AbstractBalancedDietPathway in the counter-example.

The incompleteness of the initial specifications is then corrected following an interview with specialists. Therefore, the nutritionist recommends that the Balanced Diet and Sensation Pathways should be closed to the patients once they become malnourished.

In order to take into account this new requirement, modifications to the automata of Figure 12 are proposed in the Figure 15. The first modification (Figure 15a) is applied to the GlobalNutritionPathway automaton: it stipulates that the transition which leads to the location MalnutritionAlert must be synchronised on the additional broadcast channel StopPathways_Chan with the two automata AbstractSensationPathway and AbstractBalancedDietPathways in order to stop them if they have started. On the other hand, the modifications over automata AbstractSensationPathway (Figure 15b) and AbstractBalancedDietPathway (Figure 15c) are used to return to the inactive location Off when this synchronisation takes place.

6.2 | Refinement of the BMI category computation and verification

Data refinement between models refers to the introduction of new variables into the refined model which are linked to the variables of the abstract model through invariant relations. The behaviour of the refined model must be consistent with that defined in the abstract one. Unlike in the abstract model where the BMI category is chosen randomly, in this refinement the BMI category is calculated based on the possible choices of the patient's height and weight. Figure 16 shows the refinement of the transition of AbstractPatientNutritionData process that provides the BMI category into a new process PatientNutritionDataRef1. The refinement leads to two transitions: one computes the BMI according to the patient's weight and height (function f_1 of Figure 10) and the other one associates a BMI category (function f_2). We have selected with experts some representative values into two tables H (Height) and W (Weight) as shown in Figure 17.

Two specific properties of BMI are proposed to validate the consistency between abstract and refined modelling. The first property ($Pref_{1.1}$) is a safety one: it states that BMI category still belongs to one of the types defined in the abstract model, that is, UnderWeight, HealthyWeight, OverWeight and Obese. The second property ($Pref_{1.2}$) is a reachability one, which aims to show that each of these types remains reachable in the refined modelling:

- $Pref_{1.1}$: $A\Box$ $exists(x : int[0, 3])$ GlobalNutritionPathway.WeightEvaluation imply GlobalNutritionPathway.$BMI\_C == x$.
- $Pref_{1.2}$: $E\Diamond$ GlobalNutritionPathway.WeightEvaluation && GlobalNutritionPathway.$BMI\_C == 0$.

Both properties are verified. In the same way, the second property that is verified for the Underweight BMI category ($BMI\_C == 0$) is also verified for the other BMI categories. These verifications show the consistency between the refined and abstract models so the properties P_1, P_2 and P_3 are still verified, and on the other hand, highlight the choices of the experts for the representative values of patients' weights and heights.

6.3 | Refinement of the Sensation Pathway and verification

As mentioned in the Section 5.3, the Sensation process is divided into two sub-processes, namely the Hunger process and the Satiety process. Figure 18 shows the refinement of this process. At this abstraction level, the Satiety Pathway is represented in an abstract way by a process with two states: on

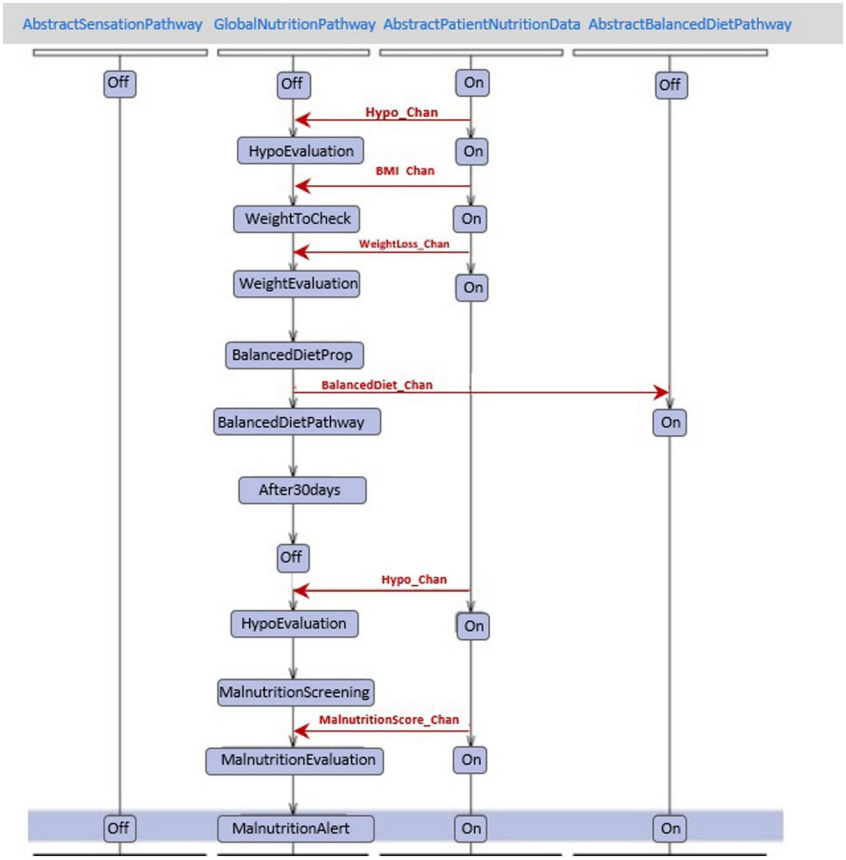


FIGURE 14 UPPAAL counter-example illustrating the failure of property P_2.

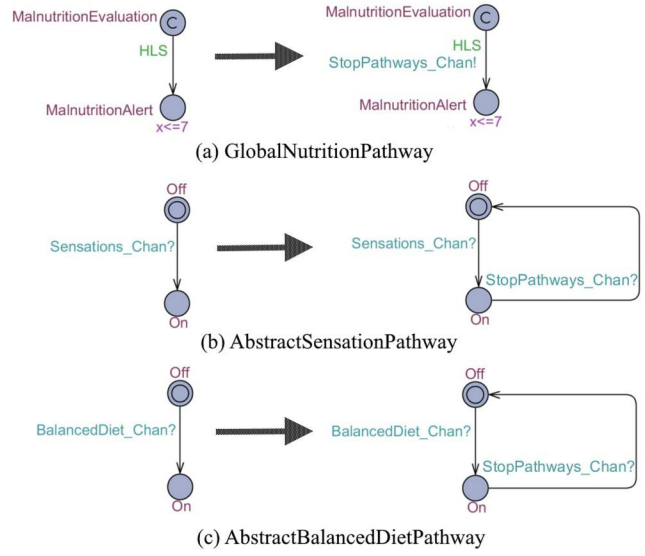


FIGURE 15 Corrections on the abstract modelling.

and off; it is expected to be refined in a future stage (not detailed here). Since the Satiety process is a sub-process of the Sensation process, stopping the later process also implies stopping the Satiety process in the refined model. The same reasoning is applied to the second sub-process, that is, the Hunger process, from all the locations that represent its active states.

The synchronisation on the channel Sensation_chan to start the Sensation process induces the start of the Hunger process in the refined model. From the Start location of this process, the patient can begin the hunger activity by answering to the first questionnaire. The transition to the FillOutQuest location represents the start of the activity: the clock y is reset to 0 and the number of questionnaires N is reset to 1. The questionnaire is simplified here by coding the patient's responses about his/her hunger sensation in four categories: (1) (no sensation), (2) (slight), (3) (moderate to severe) and (4) painful. The patient's response is modelled in PatientNutritionDataRef2 process (not represented in figure) which is the refinement of the PatientNutritionDataRef1 process by an additional transition that synchronises with Hunger process over Quest_chan channel. At each step of synchronisation, PatientNutritionDataRef2 updates the global variable *TabQuiz* [ ] according to the chosen sensation type in order to memorise the number of sensations reached for each category during the activity.

In the FilloutQuest location, the patient can continue for 3 days to answer the questionnaires while taking his/her meals (transition back to the same location). To handle a reasonable finite number of meals over 3 days, we have limited the number of meals to 9 (specified by the guard condition $N \leq 9$). After 3 days, if the number N of his/her responses is lower than the minimum required (three responses at least), the process returns to the Start location in order to allow the patient to start again the activity, and *TabQuiz*[ ] is reset to

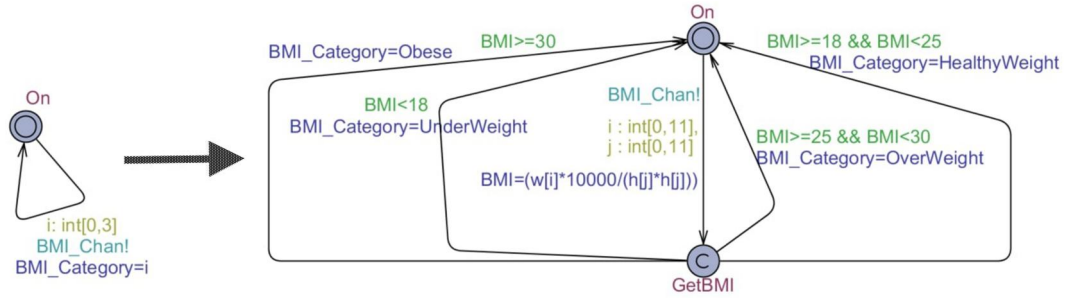


FIGURE 16 Refinement of the Body Mass Index category computation.

FIGURE 17 Local variable declarations of the automaton PatientNutritionDataRef1.

```
int W[12] = (50,60,70, 80, 90, 100, 110, 120, 130, 140, 150, 160) ;  // in kilograms
int H[12] = ([150,155,160, 165, 170, 175, 180, 185, 190, 195, 300, 205) ;  // in centimeters
int BMI ;
```

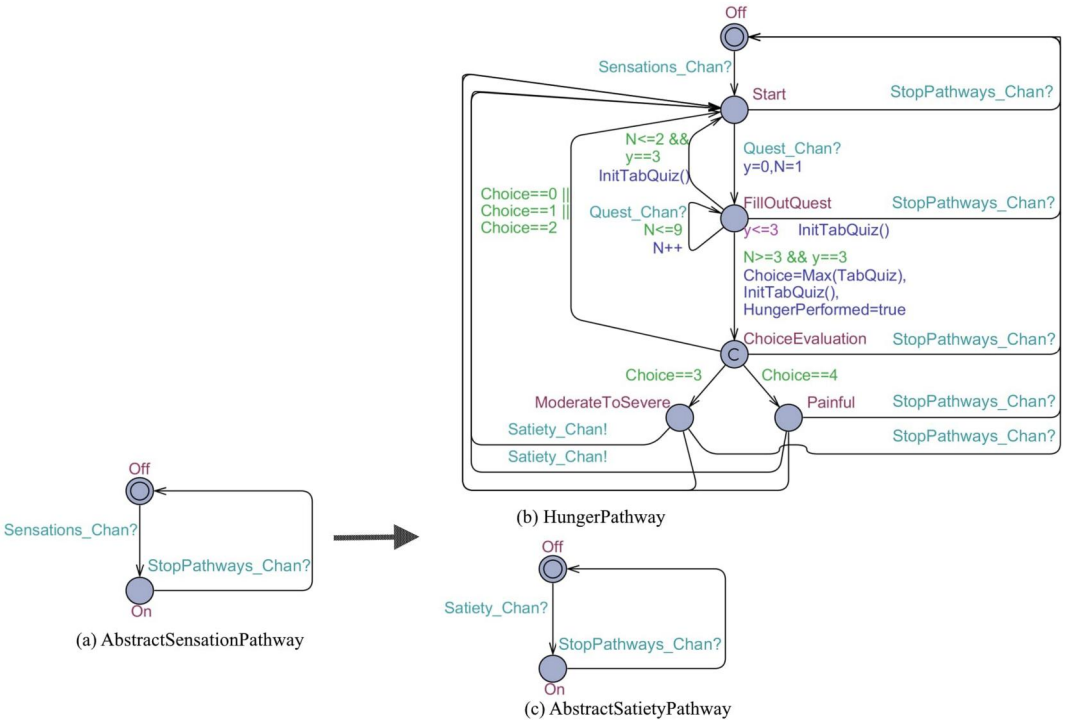


FIGURE 18 Refinement of the AbstractSensationPathway into two sub-processes.

0 (InitTabQuiz()). Otherwise, the HungerPathway activity is done: local variable HungerPerformed is assigned to true and the process passes to the location ChoiceEvaluation.

The location ChoiceEvaluation determines the patient's profile according to the four types of sensation (the choice with a majority of the answers). The assignment Choice = nutrition(TabQuiz) in the transition leading to the ChoiceEvaluation location is used to determine this choice using the variable Choice on which the evaluation is performed. When Choice equals 0, meaning there is no type of sensation has a majority holding, the process returns to the Start location so that the patient can repeat the activity. The same reasoning is applied in the case where Choice equals 1 or 2 which models the cases of no hunger sensation or slight sensation. In the case where Choice equals 3 or 4 which models moderate to severe sensation or painful sensation, the patient can start the Satiety Pathway by synchronisation with the Satiety process and/or return to the Start location to eventually restart the Hunger process. An educational content is sent to the Patient according to his/her sensation profile (Choice value) in order to improve his/her nutritional behaviour (not presented here).

Three specific properties are proposed to validate the consistency between abstract and refined modelling. The first

property ($Pref_{2.1}$) is a safety one: it states that starting Sensation Pathway leads to start Hunger Pathway. This is explained by the fact that the two processes implied in the refinement are sequentially started, Hunger being the first process. The second property ($Pref_{2.2}$) is a reachability one, which aims to show that Satiety Pathway is a living process. The third property ($Pref_{2.3}$) is a safety one, stating that stopping Sensation Pathways leads to stop both Hunger and Satiety Pathways:

- $Pref_{2.1}$: $A\Box$ GlobalNutritionPathway.SensationPathway `imply` !HungerPathway.Off.
- $Pref_{2.2}$: $E\Diamond$!SatietyPathway.Off.
- $Pref_{2.3}$: $A\Box$ GlobalNutritionPathway.MalnutritionAlert `imply` HungerPathway.Off `&&` AbstractSatietyPathway.Off.

The three properties of refinement being verified, the properties P_1, P_2 and P_3 remain valid in the refined model. Only P_4 and P_5 has to be verified:

- P_4: $E\Diamond$!HungerPathway.Off `&&` !AbstractSatietyPathway.Off
- P_5: $A\Box$!AbstractSatietyPathway.Off `imply` HungerPathway.HungerPerformed

The set of target properties being verified, the refinement of the Hunger Pathway is achieved. As shown in Figure 11, the Abstract Sensation Pathway refinement leads to define the Hunger Pathway and the Abstract Satiety Pathway. The new target process to be refined is thus abstract Satiety Pathway. That corresponds in Figure 1 with starting activity *3b.Refinement* on the Satiety process.

7 | STRENGTH AND WEAKNESS OF THE APPROACH

Specifying m-Rehab requirements and properties has been a great experience for applying the proposed approach based on abstraction and refinement both on UML models and timed automata. The large number of requirements, the specificity of the pathways offered to patients, and the interactions of roles belonging to the care management make the application difficult to be handled without a good methodology.

First of all, UML was a very good support to organise and represent the concepts (terms, definitions, classes and their relations) from the beginning of the project and share them among all the stakeholders. Class diagrams give a high-level representation of the application and is a first step to define logical modules and processes, and highlight possible refinements. Defining processes through sequence diagrams or state machine is accessible to non-experts with the help of their graphical representations, and makes abstract notions tangible. It facilitates the communication and the understanding of expert processes and allows them to be validated. It is also a support for sharing knowledge between medical experts themselves and with designers. Thanks to the obtained UML models, the project can be seen according to different points of view: through a given expertise, for example, the Nutrition Pathway studied in the article, or through a transversal view to specialised pathways, such as for example, a taxonomy of all the questionnaires that are offered to patients during their telerehabilitation.

Let's now look at our feedback on the use of formal models. Formal modelling and verification have been applied as processes to improve the specification by pointing out anomalies. We can state that this goal has been achieved for the Nutrition process. It certainly took time to set up formal models and make the expected properties explicit. But we have improved the quality of the specifications which has an impact on time and quality of development. Moreover, despite the complexity of the formal models, we were able to use them as discussion supports with medical experts thanks to the UPPAAL simulator and the generation of counterexamples.

If we go back to Figure 1, we can list weaknesses of some steps. For example, the criteria to select a target process to be analysed and refined is for now a rule of thumb based on the characteristics of the process such as temporal aspects or synchronisation with other processes. Another concern is about the abstraction and refinement processes which require a traceability between abstract and concrete data/processes and the ability to make corrections at different abstraction levels (action 3a in Figure 1). Systematic procedures to make abstraction from data or processes are difficult to be defined. It requires a modelling know-how and a deep understanding of business processes. The more abstract view of a process is its start and stop triggers, but it is difficult to define a general methodology to help designers to do refinements. Moreover, we have used refinement from an intuitive understanding, but it is necessary to handle this concept from its formal definition as we have done in previous work [31]. One way would be to define generic refinement patterns that guarantee the validity of the refinements.

Nevertheless, we can conclude from this experience that models have been very useful to support the specification phase and to improve documents required for the development.

8 | CONCLUSION AND FUTURE CHALLENGES

In this paper, we have applied a refinement-based methodology for designing and analysing a medical application, which exploits timed automata as a formalism for modelling and UPPAAL as a tool for verification. This methodology represents a motivation for the development of complex systems via a step-by-step formal approach, including those which are non-critical. Indeed, it shows how a complex system is progressively modelled and verified from the initial specifications. This approach helps to overcome the inherent complexity of the system, and facilitates the task of development by fixing bugs at an initial phase which is cheaper than adjusting in the deployed system. This will also bridge the gap between research and industry by pointing out that healthcare applications require appropriate methods to achieve a good level of

quality. Furthermore, we have demonstrated that stakeholders were involved throughout the design and the verification phases of the telerehabilitation system through the use of graphical models. It was a way to build up the system by collaborative modelling and to share the knowledge between the team members. Formal models convinced stakeholders that complex system specifications has to be verified before the implementation phase.

As future extensions for this work, we aim at proposing a framework allowing abstraction and refinement processes to be traced to get two benefits: supporting designers to set up formal models from the initial specifications keeping a trace of already developed processes and their corresponding data; make the interpretation of formal verification easier in order to enhance initial specification in a proper way, according to the relevant abstraction level. This framework has to automatically transform UML state or sequence diagram models into UPPAAL models, as it has been partially done in some existing works [44]. Using a language for expressing properties close to the natural language is also a challenge for future work.

AUTHOR CONTRIBUTIONS

Farid Arfi: Formal analysis; software; writing – original draft; writing – review & editing. **Anne-Lise Courbis**: Conceptualization; methodology, software; writing – original draft; writing – review & editing. **Thomas Lambolais**: Writing – review & editing. **François Bughin**: Funding acquisition; project administration; supervision; validation. **Maurice Hayot**: Funding acquisition; project administration; supervision; validation.

ACKNOWLEDGEMENTS

The authors thank Dr. Marlène Richou for her contribution to the definition of the nutrition pathway. Many thanks to the scientific commitee of m-Rehab and its members: Bronia Ayoub, Julie Boiché, Anne-Sophie Cases, Blandine Chapel, Gérard Dray, Nelly Heraud, Nathalie Jean, Pierre Jean, Christophe Latrille, Jordan Michel, Pascal Pomies, Roxana Ologeanu-Taddei.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article.

PERMISSION TO REPRODUCE MATERIALS FROM OTHER SOURCES

Not applicable.

ORCID

Anne-Lise Courbis https://orcid.org/0000-0002-7530-4661

REFERENCES

1. Vieira, D.S., Maltais, F., Bourbeau, J.: Home-based pulmonary rehabilitation in chronic obstructive pulmonary disease patients. Curr. Opin. Pulm. Med. 16(2), 134–143 (2010). https://doi.org/10.1097/mcp.0b013e32833642f2
2. Cox, N.S., et al.: Telerehabilitation for chronic respiratory disease. Cochrane Database Syst. Rev. (1) (2021). https://doi.org/10.1002/14651858.cd013040
3. Bonfanti, S., Gargantini, A., Mashkoor, A.: A systematic literature review of the use of formal methods in medical software systems. J. Softw. 30(5), e1943 (2018). https://doi.org/10.1002/smr.1943
4. Pajic, M., et al.: From verification to implementation: a model translation tool and a pacemaker case study. In: 2012 IEEE 18th Real Time and Embedded Technology and Applications Symposium, pp. 173–184. IEEE (2012)
5. Jee, E., et al.: A safety-assured development approach for real-time software. In: 2010 IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 133–142. IEEE (2010)
6. Jiang, Z., et al.: Closed-loop verification of medical devices with model abstraction and refinement. Int. J. Software Tool. Technol. Tran. 16(2), 191–213 (2014). https://doi.org/10.1007/s10009-013-0289-7
7. Jiang, Z., et al.: Real-time heart model for implantable cardiac device validation and verification. In: 2010 22nd Euromicro Conference on Real-Time Systems, pp. 239–248. IEEE (2010)
8. Jiang, Z., Pajic, M., Mangharam, R.: Cyber–physical modeling of implantable cardiac medical devices. Proc. IEEE 100(1), 122–137 (2011)
9. Arney, D., et al.: Formal methods based development of a PCA infusion pump reference model: generic infusion pump (GIP) project. In: 2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007), pp. 23–33. IEEE (2007)
10. Jetley, R., et al.: A formal approach to pre-market review for medical device software. In: 30th Annual International Computer Software and Applications Conference (COMPSAC'06), vol. 1, pp. 169–177. IEEE (2006)
11. Moudjari, A., Latreche, F., Talbi, H.: Meta-ECATNets for Modelling and Analyzing Clinical Pathways, vol. 64. Springer International Publishing (2019)
12. Latreche, F., Moudjari, A., Talbi, H.: Clinical pathways formal modelling using bigraphical reactive systems. In: International Colloquium on Theoretical Aspects of Computing, pp. 76–90. Springer (2019)
13. Guo, C., et al.: A framework for supporting the development of verifiably safe medical best practice guideline systems. J. Syst. Architect. 104, 101693 (2020). https://doi.org/10.1016/j.sysarc.2019.101693
14. Simalatsar, A., et al.: Representation of medical guidelines with a computer interpretable model. Int. J. Artif. Intell. Tool. 23(03), 1460003 (2014). https://doi.org/10.1142/s0218213014600033
15. Gawanmeh, A.: An axiomatic model for formal specification requirements of ubiquitous healthcare systems. In: 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC), pp. 898–902 (2013)
16. Liu, J., et al.: Modeling and analysis of interactive telemedicine systems. Innovat. Syst. Software Eng. 11(1), 55–69 (2015). https://doi.org/10.1007/s11334-013-0197-8
17. Mtibaa, S., Tagina, M.: An automated petri-net based approach for change management in distributed telemedicine environment. J. Telecommun. 15(1) (2012). ISSN 2042-8839
18. Khalid, M., et al.: Automated UML-based formal model of E-health system. In: 2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), pp. 1–6. IEEE (2019)
19. Pervez, U., et al.: Formal reliability analysis of a typical FHIR standard based e-Health system using prism. In: 2014 IEEE 16th International Conference on E-Health Networking, Applications and Services (Healthcom), pp. 43–48. IEEE (2014)
20. Ding, J., Zhang, D.: An approach for modeling and analyzing the communication protocols in a telemedicine system. In: 2013 6th International Conference on Biomedical Engineering and Informatics, pp. 699–704. IEEE (2013)

21. Addas, R., Zhang, N.: Formal security analysis and performance evaluation of the linkable anonymous access protocol. Lect. Notes Comput. Sci. 8407, 500–510 (2014)
22. Amato, F., Moscato, F.: A model driven approach to data privacy verification in e-Health systems. Trans. Data Priv. 8, 273–296 (2015)
23. Abdmeziem, M.R., Tandjaoui, D.: An end-to-end secure key management protocol for e-Health applications. Comput. Electr. Eng. 44, 184–197 (2015). https://doi.org/10.1016/j.compeleceng.2015.03.030
24. Lamine, E., et al.: Plas'O'Soins: an interactive ICT platform to support care planning and coordination within home-based care. Innovat. Res. BioMed. Eng. 40(1), 25–37 (2019). https://doi.org/10.1016/j.irbm.2018.10.015
25. Barkaoui, K., et al.: Modelling and analyzing home care plans using high-level petri nets. In: 2016 13th International Workshop on Discrete Event Systems (WODES), pp. 284–290. IEEE (2016)
26. Azeem, M.W., et al.: Specification of e-Health system using Z: a motivation to formal methods. In: International Conference for Convergence for Technology-2014, pp. 1–6. IEEE (2014)
27. Tahir, H.M., Nadeem, M., Zafar, N.A.: Specifying electronic health system with Vienna development method specification language. In: 2015 National Software Engineering Conference (NSEC), pp. 61–66. IEEE (2015)
28. Cousot, P., Cousot, R.: Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, pp. 238–252 (1977)
29. Grov, G., Ireland, A., Llano, M.T.: Refinement plans for informed formal design. In: Derrick, J., et al., (eds.) ABZ'2012, vol. 7316, pp. 208–222. Springer Verlag (2012)
30. Lambolais, T., et al.: IDF: a framework for the incremental development and conformance verification of UML active primitive components. J. Syst. Softw. 113, 275–295 (2016). https://doi.org/10.1016/j.jss.2015.11.020
31. Lambolais, T., Courbis, A.L.: Development and verification of UML architectures by refinement and extension techniques. In: European Congress of Real-Time Embedded Systems (ERTS) (2018)
32. Boiten, E.A., Bujorianu, M.C.: Exploring UML refinement through unification. In: Jürgens, J., et al. (eds.) Critical Systems Development with UML—Proceedings of the UML'03 Workshop, pp. 47–62. Technische Universität München (2003)
33. Abrial, J.R., Hoare, A.: The B-Book: Assigning Programs to Meanings, vol. 1. Cambridge University Press, Cambridge (1996)
34. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press, New York (2010)
35. Smith, G.: The Object-Z Specification Language, vol. 1 of Advances in Formal Methods. Kluwer Academic Publishers, Boston (2000)
36. Said, M.Y., Butler, M., Snook, C.: Language and tool support for class and state machine refinement in UML-B. In: FM 2009: Formal Methods: Second World Congress, Eindhoven, The Netherlands, November 2-6, 2009. Proceedings 2, pp. 579–595. Springer (2009)
37. Rasch, H., Wehrheim, H.: Checking consistency in UML diagrams: classes and state machines. In: Formal Methods for Open Object-Based Distributed Systems, pp. 229–243 (2003)
38. Méry, D., Singh, N.K.: Formal specification of medical systems by proof-based refinement. ACM Trans. Embed. Comput. Syst. 12(1), 1–25 (2013). https://doi.org/10.1145/2406336.2406351
39. Fayolle, T., et al.: Modelling a hemodialysis machine using algebraic state-transition diagrams and B-like methods. In: International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z, pp. 394–408. Springer (2016)
40. Arcaini, P., et al.: How to assure correctness and safety of medical software: the hemodialysis machine case study. In: International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z, pp. 344–359. Springer (2016)
41. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. 126(2), 183–235 (1994). https://doi.org/10.1016/0304-3975(94)90010-8
42. Behrmann, G., David, A., Larsen, K.G.: A Tutorial on Uppaal 4.0. Department of Computer Science, Aalborg University (2006)
43. Kossiakoff, A., et al.: Systems Engineering Principles and Practice, vol. 83. John Wiley & Sons (2011)
44. Yang, N., Guo, X., Wang, W.: Formal verification of a UML state chart diagram with Uppaal. Int. J. Hybrid. Inf. Technol. 5, 55–60 (2012)

How to cite this article: Arfi, F., et al.: Formal verification of a telerehabilitation system through an abstraction and refinement approach using UPPAAL. IET Soft. 1–18 (2023). https://doi.org/10.1049/sfw2.12128