

Spécification semi-formelle et formelle d'une application de télé-réhabilitation : retour d'expérience

Farid Arfi¹, Anne-Lise Courbis², Thomas Lambolais², François Bughin³, Maurice Hayot³

¹Univ. Montpellier, farid.arfi@umontpellier.fr

²Euromov DHM, Univ. Montpellier, IMT mines Ales, prenom.nom@mines-ales.fr

³PhyMedExp, Univ. Montpellier, INSERM, CNRS, CHRU de Montpellier, Montpellier, France,
f-bughin@chu-montpellier.fr, m-hayot@chu-montpellier.fr

Résumé

Nous nous intéressons aux étapes de recueil des besoins et de spécification d'une application de télé-réhabilitation de patients atteints de maladies chroniques respiratoires. Après avoir réalisé des entretiens avec les experts impliqués dans la définition des processus de réhabilitation, un cahier des charges a été établi sous forme textuelle et sous forme de modèles UML. Nous avons constaté que la modélisation UML est un réel apport pour la maîtrise d'ouvrage mais elle doit être accompagnée d'une étape de modélisation formelle et de vérification pour analyser les processus et vérifier des propriétés. Ici nous nous concentrons sur les propriétés de vivacité et de sûreté. Nous avons choisi UPPAAL comme outil de simulation et de vérification formelle. Nous montrons sur cette étude de cas quels sont les apports de chacune des étapes de modélisation informelle et formelle pour toutes les parties prenantes du projet.

Mots-clés : vérification formelle, UML, automates temporisés, UPPAAL, télé-réhabilitation.

1 Introduction

Le projet m-Rehab¹ concerne la télé-réhabilitation de patients atteints de maladies chroniques respiratoires telles que la bronchopneumopathie chronique obstructive (BPCO) et le syndrome d'apnées-hypopnées obstructives du sommeil (SAHOS) [23, 7]. La réhabilitation qualifie le processus qui vise à amener un patient à l'état de santé, physique et psychique, le plus proche possible de celui qui était le sien avant de tomber malade. La télé-réhabilitation est un domaine de la télésanté, qui utilise les technologies de l'information et de la communication pour fournir des services de réhabilitation clinique à distance [13].

Le projet m-Rehab vise à réaliser un essai clinique sur des patients atteints de maladies chroniques respiratoires dont la réhabilitation est assurée par une équipe interdisciplinaire grâce à une application mobile, nommée également m-Rehab pour *mobile-Rehabilitation*. Cette application est multi-utilisateur. Sont concernés les patients et les *Personnels de Santé et de Soins* (PSS). Nous nous intéressons ici à la phase d'expression des besoins et de spécification à laquelle nous avons été associés et qui a conduit à définir des documents de référence pour le partenaire en charge du développement.

Nous avons conduit les interviews et la transcription des besoins ont été réalisés, jouant ainsi le rôle d'assistance à la maîtrise d'ouvrage (AMOA), afin de constituer un cahier des charges formé de représentations textuelles informelles, mais également de représentations en UML. Les modèles UML ont été établis sous le logiciel Modelio [15]. Parallèlement au développement qui s'est déroulé sur deux années, nous avons réalisé des analyses formelles de processus, dont la criticité était jugée importante, afin de mettre en évidence des *anomalies* de spécification et de les corriger durant le développement de l'application. Cette étape d'analyse formelle, riche d'enseignements, est ci-après présentée sur un extrait de l'application : la gestion d'une alerte déclenchée par le patient.

Au paragraphe 2, nous présentons une synthèse de la phase d'analyse des besoins du projet m-Rehab ainsi qu'un extrait du modèle UML. Nous y détaillerons le processus d'alerte ainsi que sa modélisation et sa vérification formelle. Les principaux travaux de modélisation et de vérification formelle de processus médicaux sont cités au paragraphe 3. Nous ferons un bilan de ce retour d'expérience au paragraphe 4 en identifiant les apports de chacune des étapes pour chacune des parties prenantes du projet.

1. Ce projet est financé par les fonds FEDER Occitanie. Les partenaires sont : PhyMedExp (Univ. Montpellier, Fr), MRM (Univ. Montpellier, Fr), Euromov-DHM (Univ Montpellier, IMT Mines Ales, Fr), et le groupe KORIAN (Fr).

2 Étude de cas extraite du projet m-Rehab

2.1 Présentation de l'application m-Rehab

Le cahier des charges de m-Rehab a été établi par une succession d'entretiens menés avec les experts du domaine. Comme nous l'avons mentionné, la réhabilitation est un processus interdisciplinaire. En plus des processus administratifs (inscription par exemple) et de gestion des données du patient, trois processus sont proposés par m-Rehab, appelés : *parcours Santé*, *parcours Nutrition* et *parcours APA* (Activité Physique Adaptée).

Les experts en réhabilitation sont de formations diverses (médecins pneumologues, nutritionnistes, spécialistes en APA) et ne partagent pas toujours le même vocabulaire. De plus, ils ne connaissent pas précisément les activités de réhabilitation des autres spécialités. Il est donc nécessaire qu'à l'issue du recueil des besoins, on puisse avoir à la fois une vue synthétique et détaillée de ce qui est proposé aux patients. Durant sa téléréhabilitation, le patient sera équipé d'objets connectés (balance, tensiomètre et montre) et suivra des parcours personnalisés à travers l'application m-Rehab. Il aura des contacts en présentiel ou en visio avec les PSS formant son équipe de soin. Il aura un contact privilégié avec une personne, appelée *Care Manager*, qui assure le lien Patient/PSS, par téléphone, visio-conférence ou mail, et qui est en charge de l'orienter s'il a besoin d'information ou si certains marqueurs ne progressent pas. Le métier de *Care Manager* est nouveau et il est co-construit par les experts : les fonctions attendues sont définies par chacun des PSS qui spécifie ce que le *Care Manager* peut apporter à son parcours.

Chaque expert va ainsi définir pour sa spécialité le parcours des patients. Tout parcours s'appuie sur des contenus pédagogiques et des activités qui donneront lieu systématiquement à une évaluation et un feedback. Les processus ont été définis dans un premier temps sous forme textuelle en retranscrivant les entretiens et un diagramme de classes a été élaboré et partagé avec tous les experts de façon à mettre en évidence les concepts. Ce diagramme de classes a été un support de discussion, et a servi également à établir un dictionnaire des termes de façon à ce que toutes les parties prenantes partagent le même vocabulaire sans ambiguïté. On compte environ 300 classes organisées en package représentant les processus. La figure 1 montre une partie du diagramme de classes permettant de mettre en évidence les concepts présentés dans cette introduction.

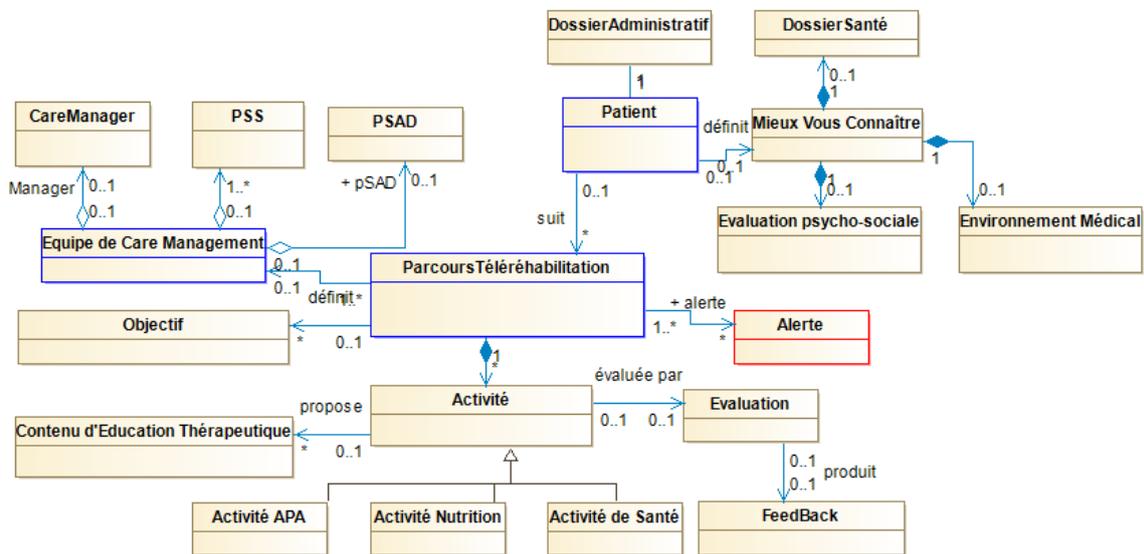


FIGURE 1 – Modèle du domaine : diagramme de classes de m-Rehab (extrait)

Nous allons nous intéresser au processus d'*Alerte* qui apparaît dans le diagramme de classes comme étant associé au Parcours du Patient. Nous en donnons sa spécification textuelle et sa modélisation UML. Nous étudierons ensuite sa modélisation formelle pour effectuer une vérification.

2.2 Extrait du cahier des charges : création et gestion d'une Alerte du Parcours Santé

Nous nous focalisons sur un extrait du cahier des charges dans lequel un médecin a décrit le processus de traitement de l'alerte pouvant être lancée par un patient présentant le syndrome d'apnée du sommeil (SAHOS).

« Le patient peut envoyer une alerte s'il rencontre des problèmes avec son dispositif de Pression Positive Continue. Le Care Manager et le PSAD (Prestataire de Santé à Domicile) sont alors avertis. Le PSAD intervient généralement sous trois jours. Au-delà de ces jours, on demande alors au Patient si son état s'est amélioré. Deux réponses sont possibles : problème résolu ou état non stabilisé. Si le problème est résolu, l'alerte est clôturée. Dans le cas contraire, on questionne le Patient sur son état tous les deux jours pendant 10 jours maximum tant que le problème n'est pas résolu. Quand le problème est résolu, l'alerte est clôturée. Sinon, le Care Manager et le médecin sont avertis : le médecin doit intervenir dans un intervalle d'un jour et l'alerte est alors clôturée. Si le PSAD ne passe pas chez le patient après trois jours du lancement d'alerte, une notification est envoyée au Care Manager et au médecin qui doit intervenir comme dans le cas décrit précédemment. »

Ce processus est représenté dans un premier temps par un diagramme de séquences. Un extrait en est donné en figure 2 correspondant au cas où le PSAD intervient. Deux situations sont mises en évidence : (i) le problème est résolu après le passage du PSAD ou dans un délai de 10 jours (blocs en vert) et (ii) l'état du patient n'est pas stabilisé (bloc en rouge) : une notification est alors envoyée au Care Manager et au médecin afin qu'ils contactent le patient. L'intérêt de ce diagramme est d'avoir une représentation graphique comme support de discussion avec le médecin qui a émis le besoin et pouvoir modifier facilement le processus.

Nous allons formaliser ce processus pour l'analyser et nous allons voir qu'il reste quelques anomalies ne permettant pas de l'implanter sous cette forme.



FIGURE 2 – Diagramme de séquences de la gestion d'une Alerte (extrait)

2.3 Modélisation et vérification d'une alerte du parcours Santé

Les spécifications textuelles souffrent généralement de plusieurs types d'anomalies : (1) incomplètes : informations insuffisantes ou manquantes ; (2) ambiguïtés sur les termes utilisés ou la description des parcours ; (3) incohérences : des résultats différents sont obtenus avec les mêmes données des patients selon

l'interprétation que l'on fait du processus. C'est une anomalie de type incomplétude qui sera traitée dans ce paragraphe.

Bien que la modélisation UML permette de relever un certain nombre d'anomalies lors des échanges avec les parties prenantes, d'autres peuvent persister étant donnée sa nature semi-formelle. Nous avons recours ici à des étapes de modélisation formelle et de vérification pour analyser et corriger la spécification. Nous avons choisi le modèle d'automates temporisés et son outil de vérification UPPAAL [5] pour les raisons suivantes : (1) les méthodes de vérification des propriétés de vivacité et de sûreté basées sur les automates temporisés ont été bien étudiées et développées dans UPPAAL ; (2) le cahier des charges du projet m-Rehab fait référence à des événements temporisés et périodiques : les automates temporisés sont appropriés pour modéliser ce type d'événements.

2.3.1 Modélisation du processus de traitement d'alertes par des automates temporisés

Le processus de traitement d'alertes énoncé précédemment au paragraphe 2.2 est spécifié sous forme d'automates temporisés (cf. figure 3). Nous retrouvons dans cette figure les cinq processus intervenant dans le diagramme de séquences de la figure 2. Les déclarations des horloges, des variables et des canaux de synchronisation des processus sont définies dans la figure 4.

On y voit que lorsque le Patient déclenche une alerte (*Alerte!* dans le processus (b)) celle-ci est synchronisée avec le processus Santé (cf. *Alerte?* dans le processus (a)) qui va de façon immédiate, grâce à l'état engagé *AlerteRecue*, se synchroniser par diffusion avec le Care et le PSAD : cf. *PbPatient!* en sortie de l'état *AlerteRecue* dans (a) et *PbPatient?* en sortie (i) de l'état *Off* dans (c) et (ii) de l'état *On* dans (e). Le PSAD peut répondre à la demande sous trois jours : cf. $x < 3$ invariant de l'état *ResoudrePb* dans processus (c), et sa transition sortante synchronisée sur *InterventionPSAD!* avec le processus (a). Le processus (a) passe alors dans l'état *AttenteReponse1* en attendant la réponse du patient sur son état. Le processus (b) peut fournir la réponse au processus (a) lors de la synchronisation sur le canal *EtatPB*. Cette réponse est choisie de façon aléatoire par le processus (b), où 0 (resp. 1) représente l'état non stabilisé (resp. résolu). Par la suite, le processus (a) analyse la réponse du patient grâce à la variable globale *Etat* servant de moyen de communication entre les processus (a) et (b) lors de la synchronisation sur *EtatPB*.

2.3.2 Spécification et vérification d'exigences

Plusieurs exigences sont énoncées dans le cahier des charges. Nous en avons extrait six. Elles ont émergé suite à des discussions avec le médecin, afin de nous assurer de la bonne compréhension des processus. La première propriété est indépendante du domaine : c'est une propriété attendue de tout processus réactif continu. Ceci correspond aux descriptions informelles ci-dessous. Nous donnons pour chaque propriété sa formalisation sous l'outil UPPAAL, qui utilise un sous-ensemble de la logique TCTL (*Timed Computation Tree Logic*)² :

<p>P_1 : Absence de blocage $A\Box$ not deadlock.</p> <p>P_2 : Le médecin ne recevra pas de notification d'alerte si celle-ci est en cours de traitement par le PSAD. $A\Box$ PSAD.ResoudreProbleme \implies !Medecin.RecevoirNotification.</p> <p>P_3 : Toute alerte déclenchée finira toujours par être clôturée. Patient.AlertOn \longrightarrow !Patient.AlertOn.</p> <p>P_4 : Après avoir lancé une alerte, le patient aura de façon sûre une intervention d'un professionnel de santé (PSAD ou médecin) dans un délai de quatre jours maximum. Sante.AlertRecue \longrightarrow Intervention && $x \leq 4$.</p> <p>P_5 : Il existe des interventions qui permettent de résoudre le problème. $E\Diamond$ Intervention && Etat == Resolu.</p> <p>P_6 : Malgré l'intervention le problème peut ne pas être résolu. $E\Diamond$ Intervention && !Etat == Resolu.</p>

Notons que P_5 et P_6 sont des propriétés du domaine, ou des connaissances métier, et non des exigences sur le système.

2. Pour des formules d'état ϕ et ψ décrites en logique propositionnelle, les notations $A\Box\phi$, $E\Diamond\phi$ et $\phi \longrightarrow \psi$ désignent respectivement le fait que ϕ soit toujours vraie, qu'il existe une branche dans laquelle ϕ deviendra vraie, et qu'en toute circonstance, ϕ conduira à ψ . Voir par exemple [5] pour une présentation plus complète.

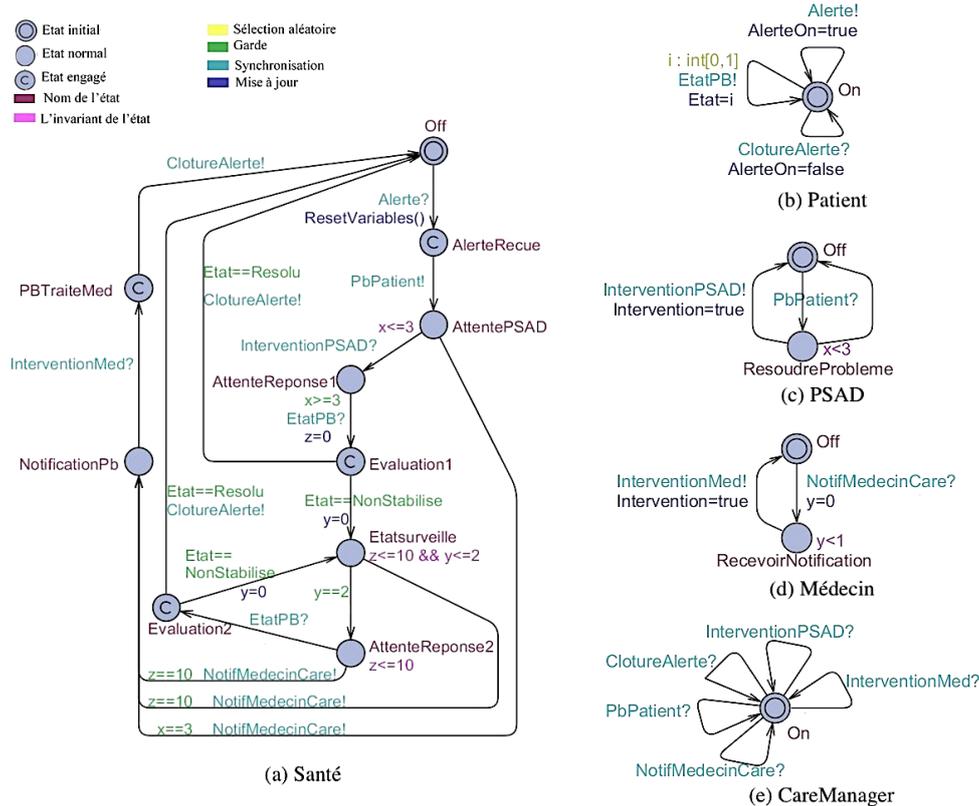


FIGURE 3 – Modélisation par automates temporisés de la gestion d’alerte.

Après vérification, toutes les propriétés sont satisfaites, à l’exception de la propriété P_3 . En voici l’explication : après intervention du PSAD, le patient étant libre de répondre à la question concernant son état de santé à partir du troisième jour du lancement d’alerte, il peut ne pas répondre à cette question ; le processus (a) reste alors en attente de la synchronisation avec le processus (b) sur le canal `EtatPB` dans l’état `AttenteReponse1`. Cette erreur nous a conduits à interroger les médecins de façon à compléter la spécification : ils proposent qu’un délai de deux jours soit laissé au patient pour répondre sur son état de santé. Cette précision engendre une modification de la garde de la transition de synchronisation sur le canal `EtatPB` en sortie de l’état `AttenteReponse1` qui devient : $x \geq 3 \ \&\& \ x < 5$. Le même raisonnement est fait sur l’état `AttenteReponse2` : la garde de sa transition de sortie synchronisée sur le canal `EtatPB` devient : $y < 4$.

```
// global declarations:
broadcast chan InterventionMed, ClotureAlerte, PbPatient, NotifMedecinCare, InterventionPSAD;
chan Alerte, EtatPB;
clock x;
int Etat;
const int Resolu=1, NonStabilise=0;
bool Intervention;
void ResetVariables() { Intervention=false; x=0; Abandon=false; return;}

clock y, z; // local declarations: Sante
bool AlerteOn // local declaration: Patient
clock y; // local declaration: Medecin
```

FIGURE 4 – Déclaration des canaux de synchronisation, des variables et des horloges.

Avec cette modification, la propriété P_1 n’est plus satisfaite : il y a un blocage provenant de la synchronisation avec le patient sur le canal `EtatPB`. En effet, comme le patient n’est pas obligé de se synchroniser exactement pendant l’intervalle préconisé, s’il rate cette synchronisation, il ne peut plus fournir la réponse attendue par le système. La figure 5 montre un contre-exemple généré par UPPAAL où P_1 n’est pas satisfaite, illustré par un diagramme de séquences conduisant au blocage de l’état `AttenteReponse1` du processus Santé (figure 5(a)), et la valeur des variables et des horloges correspondantes (figure 5(b)).

Cette erreur nous a conduits encore une fois à interroger les médecins et compléter le cahier des charges. Ils demandent alors que, dans le cas où le patient ne répond pas dans l’échéance proposée, une notification

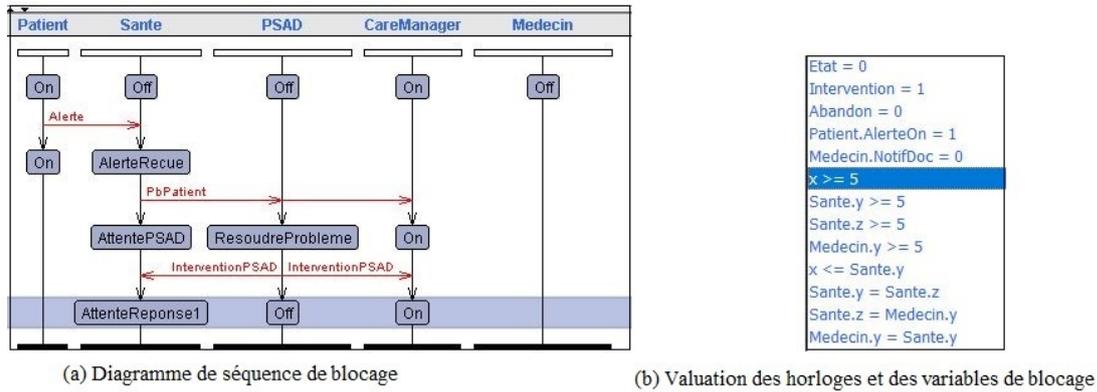


FIGURE 5 – Un contre-exemple montrant que la propriété P_1 n'est pas satisfaite.

soit envoyée au *Care Manager* pour l'informer de la situation. Le *Care Manager* prend alors l'initiative de contacter le patient (non modélisé ici) et l'alerte sera clôturée.

Cette précision nous a permis finalement de corriger le modèle formel et de vérifier l'ensemble des propriétés souhaitées. La figure 6 montre les nouvelles versions des processus Santé et CareManager de la figure 3 permettant de satisfaire toutes les propriétés énoncées. Si le patient ne se synchronise pas au niveau de l'état *AttenteReponse1* pendant l'échéance attendue ($x \geq 3 \ \&\& \ x < 5$), il est nécessaire de provoquer une sortie immédiate de l'état. Pour cela, un invariant est ajouté sur l'état *AttenteReponse1* : $x \leq 5$ et une transition de synchronisation est ajoutée sur le canal *NotifCareAband* vers l'état *PasReponsePatient* avec pour garde $x == 5$. Un raisonnement similaire est appliqué au niveau de l'état *AttenteReponse2* pour être conforme à la nouvelle demande.

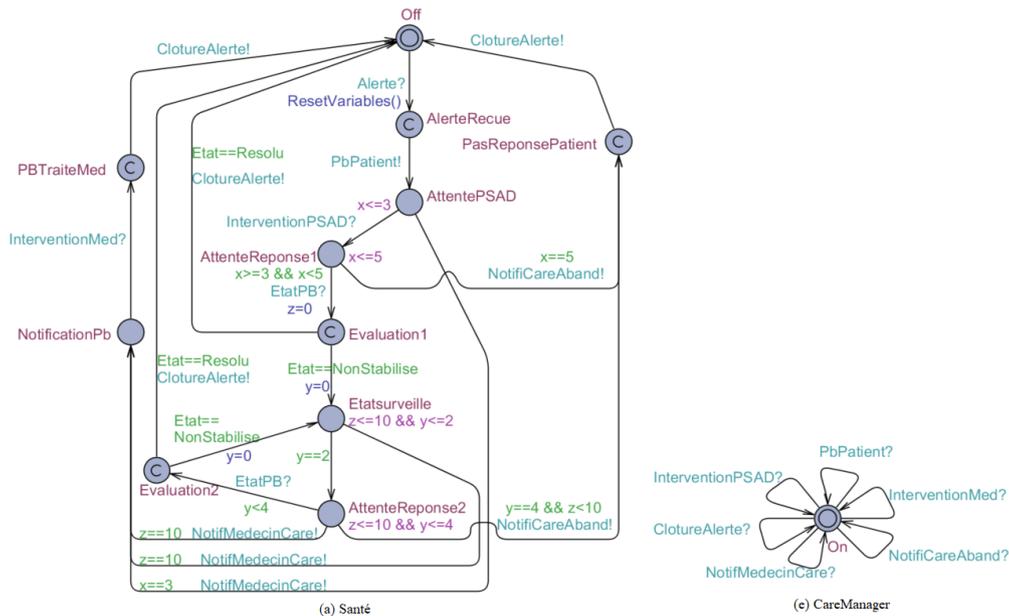


FIGURE 6 – Correction des automates (a) et (e) de la figure 3.

3 Travaux du domaine

Nous présentons ici des travaux concernant les approches de modélisation de processus médicaux puis les approches permettant de faire une vérification formelle des processus soit à partir de modèles formels, soit par transformation de modèles.

Il existe des approches de modélisation spécifiques à l'expression de *protocoles médicaux* telles que Asbru [20] et GLARE [21]. L'état de l'art de [22] et des travaux postérieurs montrent que UML a été largement utilisé pour modéliser des parcours cliniques. On notera l'existence d'un projet de développement

d'un profile UML pour le domaine médical appelé *Healthcare System Specifications* et le *HL7 Development Framework* visant à définir un standard pour l'interopérabilité des modèles de santé [9].

L'utilisation de ces approches contribue à clarifier et faciliter la validation des processus par les experts du domaine médical ou les concepteurs. Néanmoins, elles ne permettent de garantir que les processus modélisés ont le comportement désiré, et une vérification formelle est nécessaire. La revue systématique de [6] qui a été réalisée en 2018 montre que les techniques de vérification formelle sont de plus en plus utilisées dans le domaine médical que ce soit pour la vérification, la validation ou encore la génération de code. Nous avons complété cette bibliographie et nous constatons que la plupart des méthodes formelles sont appliquées sur des dispositifs médicaux critiques plutôt que sur des processus médicaux. Par exemple, nombreuses sont les références qui utilisent les automates temporisés et l'outil UPPAAL (ou la technique de *model-checking*) pour la vérification de dispositifs. Nous citons deux des dispositifs les plus étudiés dans la littérature : le stimulateur cardiaque (Pacemaker) [11, 12] et la pompe à perfusion (Infusion pump) [3, 10]. Concernant les processus médicaux, certains auteurs s'intéressent à l'utilisation des méthodes formelles pour assurer le bon fonctionnement des interactions dans un système médical [14, 18], d'autres par exemple portent leurs travaux sur la confidentialité et la sécurité des données de santé [1, 2].

Quelques travaux portent sur l'utilisation de méthodes formelles pour aider au développement d'applications de santé [4, 19] : les auteurs montrent l'intérêt de l'utilisation du formalisme Z et VDM-SL pour détecter des anomalies de spécification et réduire le coût du processus de développement.

Pour combler l'écart entre les approches de modélisation semi-formelles et les méthodes formelles, des travaux appliqués au domaine médical proposent des transformations de modèles. Citons la transformation des modèles Asbru vers des modèles algébriques dédiés au prouveur de théorèmes KIV [20], ou, dans le cas d'UML, les transformations des machines d'états [17], des diagrammes de classes [16] et diagrammes d'activités [8] vers les automates temporisés de l'outil UPPAAL.

4 Bilan : retour d'expérience

L'application m-Rehab concerne des utilisateurs que sont les patients et experts variés dont aucun n'a la connaissance détaillée de tous les processus, et pour lesquels les services offerts sont différents.

Étant donné le nombre important d'exigences à prendre en compte et la complexité des parcours de réhabilitation proposés aux patients, il était nécessaire d'avoir en premier lieu une approche de modélisation informelle. La représentation des besoins de m-Rehab sous forme textuelle accompagnée de modèles UML a permis de représenter et manipuler tous les concepts de façon unifiée. Le modèle rassemble le lexique (une centaine de termes), les types de données (environ 120), les concepts (300 classes) et leurs relations, la typologie des parcours (Santé, Nutrition, APA et parcours administratif) et pour certains, leur description sous forme de diagrammes de séquences ou de machines à états UML.

Il apparaît que l'utilisation de langages du domaine de l'informatique n'est pas une barrière pour communiquer avec la maîtrise d'ouvrage. L'approche de modélisation UML rend tangible une expertise d'un domaine afin de l'affiner et le corriger par une collaboration entre concepteurs et experts. Le cahier des charges final est ainsi plus clair et précis qu'un document purement textuel. Globalement, UML a permis d'atteindre les objectifs suivants : établir un lexique du domaine et partager les mêmes concepts avec un minimum d'ambiguïté ; avoir un support de discussion pour affiner les besoins des PSS, chacun pouvant voir son processus à haut niveau et les éléments impliqués dans ce processus ; donner aux PSS une vision à la fois détaillée et transverse de la téléréhabilitation ; offrir une co-construction des fonctions d'un rôle nouveau : la vue synthétique des fonctions du *care manager* a permis aux PSS de co-construire ces fonctions. De plus, l'utilisation d'UML permet d'adopter une organisation modulaire pour ensuite mettre en place une approche incrémentale et itérative de développement. Il a ainsi été possible de travailler sur une partie des spécifications sans nécessairement connaître l'ensemble. Ce processus réduit le coût et la complexité de la modélisation.

De façon plus surprenante, la modélisation formelle n'est pas non plus une barrière stricte : les animations et les contre-exemples d'UPPAAL ont rendu compréhensibles des concepts difficiles à appréhender par des non experts de la modélisation par automates temporisés, avec des modèles visuels jugés clairs et précis, ainsi qu'en identifiant des sources d'anomalies qui étaient restées cachées.

En revanche, nous notons que, même informellement, il a été difficile de faire énoncer des propriétés aux experts : même si ces propriétés correspondent à des exigences, les experts ne raisonnent pas en ces termes mais plutôt en décrivant leur savoir-faire. Seule l'AMOA peut les reformuler ainsi.

Cette expérience a mis en évidence la nécessité d’allier modélisation formelle et informelle pour définir des processus validés par les experts du domaine et répondant à des propriétés de sûreté ou de vivacité, et ceci y compris pour des applications de santé jugées *a priori* non critiques.

Remerciements Les auteurs remercient le comité scientifique de *m-Rehab* et ses membres : 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.

Références

- [1] Rima Addas and Ning Zhang. Formal security analysis and performance evaluation of the linkable anonymous access protocol. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8407 LNCS :500–510, 2014.
- [2] Flora Amato and Francesco Moscato. A model driven approach to data privacy verification in e-health systems. *Transactions on Data Privacy*, 8 :273–296, 2015.
- [3] David Arney, Raoul Jetley, Paul Jones, Insup Lee, and Oleg Sokolsky. 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 (HCMDS-MDPnP 2007)*, pages 23–33. IEEE, 2007.
- [4] Muhammad Waqar Azeem, Muhammad Ahsan, Nasir Mehmood Minhas, and Khadija Noreen. Specification of e-health system using z : A motivation to formal methods. In *International Conference for Convergence for Technology-2014*, pages 1–6. IEEE, 2014.
- [5] Gerd Behrmann, Alexandre David, and Kim G Larsen. A tutorial on uppaal 4.0. *Department of computer science, Aalborg university*, 2006.
- [6] Silvia Bonfanti, Angelo Gargantini, and Atif Mashkoor. A systematic literature review of the use of formal methods in medical software systems. *Journal of Software : Evolution and Process*, 30(5), 2018.
- [7] Narelle S Cox, Simone Dal Corso, Henrik Hansen, Christine F McDonald, Catherine J Hill, Paolo Zanaboni, Jennifer A Alison, Paul O’Halloran, Heather Macdonald, and Anne E Holland. Telerehabilitation for chronic respiratory disease. *Cochrane Database of Systematic Reviews*, (1), 2021.
- [8] Zamira Daw, Rance Cleaveland, and Marcus Vetter. Formal verification of software-based medical devices considering medical guidelines. *International Journal of Computer Assisted Radiology and Surgery*, 9 :145–153, 2014.
- [9] HL7-FHIR. <https://www.hl7.org/fhir/>, avril 2022.
- [10] Raoul Jetley, S Purushothaman Iyer, Paul L Jones, and William Spees. A formal approach to pre-market review for medical device software. In *30th Annual International Computer Software and Applications Conference (COMP-SAC’06)*, volume 1, pages 169–177. IEEE, 2006.
- [11] Zhihao Jiang, Miroslav Pajic, Rajeev Alur, and Rahul Mangharam. Closed-loop verification of medical devices with model abstraction and refinement. *International Journal on Software Tools for Technology Transfer*, 16(2) :191–213, 2014.
- [12] Zhihao Jiang, Miroslav Pajic, and Rahul Mangharam. Cyber–physical modeling of implantable cardiac medical devices. *Proceeding of IEEE Special Issue on Cyber-Physical Systems*, 100(1) :122–137, 2011.
- [13] Dahlia Kairy, Pascale Lehoux, Claude Vincent, and Martha Visintin. A systematic review of clinical outcomes, clinical process, healthcare utilization and costs associated with telerehabilitation. *Disability and rehabilitation*, 31(6) :427–447, 2009.
- [14] Marrium Khalid, Hamra Afzaal, Shoaib Hassan, Nazir Ahmad Zafar, Saba Latif, and Aniqah Rehman. Automated uml-based formal model of e-health system. In *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, pages 1–6. IEEE, 2019.
- [15] Modelio. <https://www.modelio.org/>, avril 2022.
- [16] Mehdi Nobakht and Dragos Truscan. An approach for validation, verification, and model-based testing of {UML}-based real-time systems. *ICSEA 2013, The Eighth International Conference on Software Engineering Advances*, pages 79–85, 2013.
- [17] Kasi Periyasamy, Yiwei Zou, and Sachin Padhye. A framework for verification of uml statechart diagrams. *29th International Conference on Computers and Their Applications, CATA 2014*, pages 167–174, 2014.
- [18] Usman Pervez, Osman Hasan, Khalid Latif, Sofiene Tahar, Amjad Gawanmeh, and Mohamed Salah Hamdi. 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)*, pages 43–48. IEEE, 2014.

- [19] Hafiz Muhammad Tahir, Muhammad Nadeem, and Nazir Ahmad Zafar. Specifying electronic health system with vienna development method specification language. In *2015 National Software Engineering Conference (NSEC)*, pages 61–66. IEEE, 2015.
- [20] Annette Ten Teije, Mar Marcos, Michel Balsler, Joyce van Croonenborg, Christoph Duelli, Frank van Harmelen, Peter Lucas, Silvia Miksch, Wolfgang Reif, Kitty Rosenbrand, et al. Improving medical protocols by formal methods. *Artificial intelligence in medicine*, 36(3) :193–209, 2006.
- [21] Paolo Terenziani, Stefania Montani, Alessio Bottrighi, Mauro Torchio, Gianpaolo Molino, and Gianluca Correndo. The glare approach to clinical guidelines : Main features. *Studies in health technology and informatics*, 101 :162–6, 2004.
- [22] Christos Vasilakis, Dorota Leczarowicz, and Chooi Lee. Application of unified modelling language (uml) to the modelling of health care systems : An introduction and literature survey. *Developments in Healthcare Information Systems and Technologies : Models and Methods*, pages 275–287, 2011.
- [23] Danielle SR Vieira, Francois Maltais, and Jean Bourbeau. Home-based pulmonary rehabilitation in chronic obstructive pulmonary disease patients. *Current opinion in pulmonary medicine*, 16(2) :134–143, 2010.