Identification de propriétés pour la validation d'un système cyber-physique médical
(Session doctorant)

Yoann Blein
Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
CNRS, LIG, F-38000 Grenoble, France
yoann.blein@imag.fr

Résumé

On s'intéresse ici à la validation de systèmes cyber-physiques médicaux (SCPM). Plus précisément, on cherche à établir la conformité de traces d'exécution par à rapport des propriétés données. Ces propriétés seront exprimées par des ingénieurs dans un langage dédié (DSL) de haut niveau et adapté aux besoins. Il est critique de trouver le juste niveau d'abstraction pour ce langage afin qu'il soit adoptable par nos partenaires industriels.

Cet article rapporte sur les premières étapes de la définition de ce langage dédié. À partir d'une étude réelle de SCPM (pose assistée de prothèse totale du genou), nous identifions les besoins des industriels du domaine, ainsi que les propriétés qu'ils souhaiteraient vérifier sur ce système. Un premier résultat de cette étude est l'identification de trois familles de propriétés de natures radicalement différentes : elles portent sur le comportement du système, sur l'interface homme-machine et les informations affichées par le système, et enfin sur des propriétés de géométrie 3D.

1 Introduction

L'utilisation de systèmes cyber-physiques médicaux (SCPM) est de plus en plus fréquente dans les hôpitaux et les cliniques. Ces appareils ont pour but de faciliter, voire automatiser, le traitement d'un patient. Par exemple, la pompe à infusion administre une solution dans le système sanguin d'un patient, et les assistants de chirurgie permettent d'augmenter la fiabilité des gestes opératoires d'un chirurgien. Même s'ils peuvent avoir des modes de fonctionnement très différents, les SCPM ont pour nature même d'intervenir sur l'organisme d'un patient. De ce fait, leur sûreté de fonctionnement est critique [4].

Dans le cadre du projet ANR MODMED, nous avons l'opportunité d'étudier un SCPM réel permettant d'effectuer la pose assistée d'une prothèse totale du genou. On dispose notamment des traces d'exécution produites par ce système sur un nombre important de chirurgies. Le projet MODMED souhaite concevoir un outil permettant de s'assurer que les exécutions d'un système, reflétées par ses traces, vérifient certaines propriétés.

2 Étude de cas : l'arthroplastie totale du genou assistée par ordinateur

Dans ce travail, nous étudions un système de guidage assisté par ordinateur pour l'arthroplastie totale du genou : ExatechGPS, conçu conjointement par les sociétés Exactech et BlueOrtho. L'arthroplastie totale du genou est une intervention chirurgicale qui consiste à remplacer certaines parties de l'articulation du genou par une prothèse. Le but de cette opération est de soulager la douleur causée par un genou arthritique, tout en maintenant ou améliorant sa fonctionnalité. Afin d'installer la prothèse, il est nécessaire de couper une partie du tibia et du fémur. ExatechGPS aide le chirurgien à réaliser ces

coupes précisément grâce à l'installation pas à pas de guides de coupe dans la *bonne* position. Cette position est déterminée en combinant l'objectif cible donné par le chirurgien et la reconstitution spatiale de la scène établie par le système. Il est actuellement utilisé à l'échelle mondiale.

ExatechGPS est une solution clé en main qui fourni à la fois la prothèse du genou et le système de guidage. Ce dernier comprend plusieurs composants : une machine capable de communiquer avec le chirurgien via un écran tactile, une caméra trois dimensions, un ensemble de traqueurs visibles par cette caméra et un ensemble d'instruments mécaniques permettant de fixer les traqueurs et les guides de coupe sur le tibia et le fémur.

L'installation des guides de coupes est réalisée au travers d'une succession d'étapes que doit réaliser le chirurgien. La nature de ces étapes ainsi que l'ordre dans lequel elles sont réalisées sont, dans une certaine mesure, configurables. Cette configuration, appelée *profil*, est déterminée avec le chirurgien selon ses préférences opératoires. Dans tous les cas, la séquence des étapes prend la forme macroscopique suivante : calibration des capteurs, acquisition de points anatomiques, vérification des acquisitions, ajustement des paramètres cibles, et enfin, réglage des guides de coupe.

ExatechGPS est équipé d'un système d'enregistrement de trace d'exécution. Pour toute chirurgie effectuée, la trace d'exécution correspondante est envoyée à BlueOrtho. L'entreprise dispose ainsi d'un corpus d'environ 4500 traces de chirurgies ayant eu lieu dans des conditions réelles. Chaque trace se compose d'un journal d'événements, d'une description hiérarchique des étapes de la chirurgie contenant les valeurs acquises et calculées, et de l'ensemble des captures d'écran réalisées pour chaque étape. Cet ensemble d'informations permet de comprendre a posteriori le déroulement d'une chirurgie et éventuellement d'identifier des défaillances.

BlueOrtho souhaiterait exploiter le corpus des traces accumulées sur plusieurs années pour attester la robustesse du logiciel et valider son utilisation dans des environnements non contrôlés. En effet, pendant le développement du produit, un certain nombre d'hypothèses ont été faites à la fois sur son environnement et sur son utilisation. Ces hypothèses sont aujourd'hui traduites dans des recommandations d'utilisation. Les développeurs ont choisi que le système soit tolérant au non-respect des recommandations afin d'entraver le moins possible le travail du chirurgien et de le laisser maître de la situation. Concrètement, ces hypothèses peuvent être traduites en propriétés que les traces devraient satisfaire.

Afin de vérifier automatiquement des propriétés sur un ensemble de traces, il est nécessaire de formaliser ces propriétés. Pour cela, on veut déterminer un langage dédié de haut niveau qui est

— adapté au domaine des SCPM ;
— facile à apprendre et manipuler afin d'augmenter ses chances d'adoption dans l'industrie ;
— et qui permet d'exprimer des propriétés vérifiables sur des traces finies.

Un enjeu important est de trouver le juste niveau d'abstraction pour ce langage dédié : il doit être pratique à utiliser et peu sujet aux erreurs, tout en ayant une expressivité suffisante pour décrire des propriétés intéressantes.

3 Travaux connexes sur la formalisation de propriétés

On distingue deux types de formalismes dédiés à la spécification formelle de propriétés. D'une part, il existe les formalismes de bas niveau qui ont une sémantique non ambigüe mais qui sont difficiles à manipuler. On peut citer la logique temporelle linéaire proposée par Pnueli [6] ou des extensions plus récentes de celle-ci [2, 5]. On pense aussi aux travaux modernes de Barringer *et al.* [1] où les propriétés sont spécifiées via des automates expressifs.

D'autre part, nous avons les langages dédiés de haut niveau dont le but est de simplifier la lecture et l'écriture de propriétés. La sémantique de ces langages est souvent définie par rapport à un formalisme de la catégorie précédente. On pense notamment aux travaux initiaux de Dwyer *et al.* sur les patrons de propriétés [3] ou un raffinement de ceux-ci comme proposé par Smith *et al.* [7].

4 Conclusion et perspectives

Le travail que je mène actuellement vise à déterminer un langage dédié pour la formalisation des propriétés attendues. Il consiste à étudier les documents de spécification et le plan de test détaillé pour identifier les propriétés pertinentes et les exprimer dans différents langages (LTL, QEA, Dwyer, ...). À ce jour, trois classes de propriétés distinctes ont été identifiées :

1. Des propriétés portent sur le comportement du système, c'est-à-dire sur la séquence d'événements internes du logiciel. Par exemple « la phase d'acquisition a toujours lieu après la phase de calibration des capteurs » ;
2. Des propriétés portent sur l'interface homme-machine et plus particulièrement sur les informations affichées et les actions disponibles. Par exemple « l'utilisateur doit toujours pouvoir revenir à l'étape précédente » (sur toutes les captures d'écran présentes dans une trace, le bouton « précédent » est présent) ;
3. Des propriétés portent sur la géométrie la scène 3D construite par le système à partir des acquisitions. Ces propriétés permettent notamment de valider l'utilisation du système. Par exemple, « les traqueurs ne sont jamais exploités en dehors du champ recommandé par rapport à la caméra » (la précision de la caméra n'est pas garantie en dehors de cette zone).

Finalement, nous avons également observé des propriétés formées par combinaisons de ces trois classes, telle que « l'utilisateur doit refaire l'acquisition d'un nuage de points si sa qualité ne paraît pas assez bonne ». La première partie correspond à un événement observé par le système (classe 1) et la seconde à un critère géométrique (classe 3). La suite de ce travail comportera une analyse qualitative de la lisibilité et l'utilisabilité de ces langages par les ingénieurs de développement.

Remerciements Ce travail est financé par le projet ANR MODMED (ANR-15-CE25-0010).

Références

[1] H. Barringer, Y. Falcone, K. Havelund, G. Reger, and D. E. Rydeheard. Quantified event automata : Towards expressive and efficient runtime monitors. In *FM 2012 : Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings*, pages 68–84, 2012.

[2] D. A. Basin, F. Klaedtke, S. Müller, and E. Zalinescu. Monitoring metric first-order temporal properties. *J. ACM*, 62(2) :15, 2015.

[3] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett. Patterns in property specifications for finite-state verification. In *Proceedings of the 1999 International Conference on Software Engineering, ICSE' 99, Los Angeles, CA, USA, May 16-22, 1999.*, pages 411–420, 1999.

[4] E. A. Lee. Cyber physical systems : Design challenges. In *11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008), 5-7 May 2008, Orlando, Florida, USA*, pages 363–369, 2008.

[5] R. Medhat, Y. Joshi, B. Bonakdarpour, and S. Fischmeister. Accelerated runtime verification of LTL specifications with counting semantics. *CoRR*, abs/1411.2239, 2014.

[6] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57, 1977.

[7] R. L. Smith, G. S. Avrunin, L. A. Clarke, and L. J. Osterweil. PROPEL : an approach supporting property elucidation. In *Proceedings of the 24th International Conference on Software Engineering, ICSE 2002, 19-25 May 2002, Orlando, Florida, USA*, pages 11–21, 2002.