Du recueil d'expertises à la production de logiciels fiables Application aux systèmes de neurostimulation médicale

Clément Duffau, Mireille Blay-Fornarino

Du recueil d'expertises à la production de logiciels fiables

Application aux systèmes de neurostimulation médicale

Clément Duffau

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis
2000, route des Lucioles
06903 Sophia Antipolis Cedex

duffau@i3s.unice.fr

MOTS-CLÉS : génie logiciel, ligne de produits logiciels, connaissance, expert, medical

KEYWORDS: software engineering, software product line, knowlegde, expert, medical

ENCADREMENT : Mireille Blay-Fornarino (PR)

1. Contexte

La problématique portée par la société AXONIC est la production de systèmes de neurostimulation médicale (SNSM) dédiés à différentes pathologies (e.g., obésité, douleur fantôme). Un SNSM se présente comme un neurostimulateur (partie hardware), ses électrodes de stimulation et un logiciel de pilotage. Ce logiciel est un système d'information utilisé par différents corps de métiers (chirurgien, clinicien, patient). Les systèmes ainsi développés doivent répondre à des normes strictes qui reposent entre autres sur un cycle de développement qui inclut de nombreuses phases de tests et en particulier, une fois la Vérification & Validation passée, sur des campagnes d'expérimentations mettant en jeu différents sujets (de l'animal à l'homme avec différentes caractéristiques). En fonction des phases de tests, des propriétés différentes sont recherchées (*e.g.* seuils de douleur, efficacité, identification des effets indésirables). On parle d'expérimentation, lorsque l'on mène sur un même sujet un ensemble de stimulations. Chaque stimulation donne lieu à des résultats physiques "bruts" (*e.g.* ressenti du patient, imagerie médicale). Les résultats obtenus dans le cadre d'une expérimentation sont

ensuite interprétés par des experts métiers (*e.g.* cliniciens, chirurgiens) qui en déduisent alors la validation ou non des propriétés recherchées ; l'ensemble stimulation, résultat, et interprétation devient alors une *connaissance*.

Sur la base de ces connaissances, il s'agit alors de contraindre les SNSM pour, par exemple, interdire les stimulations qui conduisent à de la douleur. Aujourd'hui, cette étape repose principalement sur le développeur qui doit configurer le logiciel pour traduire ces différentes "connaissances" en contraintes et recommandations. Or chaque SNSM présente ses propres caractéristiques (*e.g.,* type de stimulateur, implantation ou non, forme de l'onde électrique), ce qui conduit à une grande variabilité des SNSM. Les systèmes eux-mêmes présentent des plages de paramétrages différentes en fonction des sujets, des pathologies, des propriétés recherchées.

Dans cette thèse nous nous intéressons à faciliter la production de SNSM adaptés aux sujets et aux pathologies en intégrant la prise en compte des expérimentations dans le cycle de production de ces SNSM. Bien qu'appliquée dans le contexte de cette thèse au domaine médical porté par la société AXONIC, cette problématique est généralisable à la mise en œuvre de grands systèmes, dès que la connaissance sur ces systèmes dépend d'expérimentations.

2. État de l'art

Les lignes de produits logiciels (LPL) sont des solutions bien adaptées pour produire des logiciels fiables, automatiquement (Pohl *et al.*, 2005). Dans notre cas d'étude, la ligne doit évoluer pour intégrer non seulement de nouvelles fonctionnalités (*e.g.*, nouveau stimulateur, nouvelle pathologie) mais également et surtout, s'enrichir des connaissances acquises. En cela, nous nous approchons d'une problématique d'écosystèmes (Bosch, 2009). Toute la difficulté est alors de faire évoluer les assets associés (au sens de la LPL, artefacts nécessaires à la production de codes)(Seidl *et al.*, 2012).

L'évolution du logiciel a largement été étudiée (Mens *et al.*, 2005). Plus spécifiquement, nous nous intéressons aux évolutions au cours du cycle de vie, à quel niveau ont-elles un impact et comment les mettre en œuvre (Buckley *et al.*, 2005). Nous axons notre travail sur l'enrichissement d'une LPL à partir d'expérimentations qui est un point non abordé dans la littérature à notre connaissance.

Il s'agit donc d'être capable de capitaliser les connaissances recueillies lors des expérimentations. Ainsi Polacsek (Polacsek, 2016) propose d'organiser les connaissances sous la forme de graphes d'argumentation. A partir d'évidences et conformément à des stratégies bien fondées, des conclusions sont établies (logique, responsabilité humaine). Dans le cadre de cette thèse, il s'agit d'assurer la cohérence de l'ensemble du graphe et de permettre son enrichissement dans ce processus semi-automatique. Sur la base des conclusions, le système doit automatiquement respecter les contraintes et situations apprises des expérimentations. Toute la difficulté pour nous est alors de n'autoriser que des assertions et de détecter les incohérences. Ainsi contrairement à des approches

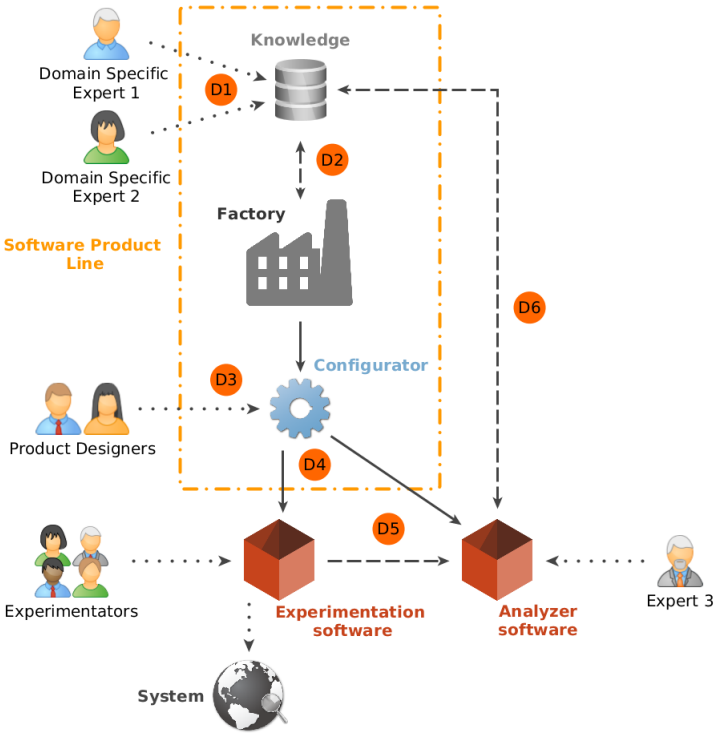

Figure 1. *Vision globale de la problématique*

de data mining (Donoghue *et al.*, 2015), nous ne cherchons ni à nous substituer aux cliniciens ni à optimiser les paramètres de sélection des stimulations.

Ces expérimentations suivent un processus prédéfini 1) établissement d'un protocole clinique (*e.g.* nombre de sujets, taux de succès du traitement attendu, profils de sujets), 2) planification des essais (*e.g.*, cibler des sujets disponibles, se coordonner avec la structure hospitalière), 3) traiter (*e.g.*, pose du dispositif médical, calibrage, stimuler), 4) analyser les résultats (*e.g.*, réactions du patient, imagerie médicale, effets secondaires), 5) présenter (*e.g.*, écriture d'article scientifique, certification). Dans la lignée des travaux de (Wohlin *et al.*, 2000), nous nous intéressons plus spécifiquement aux étapes 4 et 5, analyse/interprétation et présentation, avec l'objectif de proposer des moyens numériques pour aider et capitaliser sur ces étapes essentielles.

3. Problématique

La figure 1 présente notre point de vue actuel sur la problématique abordée dans cette thèse : construire automatiquement un Système d'Information (SI) qui respecte les

contraintes apprises à partir d'expérimentations. On distingue 3 catégories d'acteurs : (1) les experts apportent leur expertise technique en amont du projet pour définir les contraintes globales du système en s'appuyant sur l'état de l'art ; (2) les "Product Designers" sont en charge de configurer la ligne pour générer le SI dédié ; (3) les expérimentateurs sont les utilisateurs finaux. Nous identifions les défis suivants.

D1, D6 : Alimenter la base de connaissances Aujourd'hui les expérimentations sont mémorisées à travers des fiches techniques, des documents ou tableurs synthétisant les résultats d'une expérimentation. Ces données sont hétérogènes, non liées entre elles et constituent un ensemble documentaire volumineux. Les informations sont donc diffuses et peu maîtrisables par la totalité des experts. Le défi est de maintenir l'expression de ces expertises séparée et incrémentale, tout en garantissant la cohérence des connaissances qui en résultent. Or une expérimentation peut être contredite ultérieurement. Il faut donc être capable de gérer ses incohérences et évolutions. La problématique du passage à l'échelle se pose aussi, notamment sur la remontée de connaissances issues d'expérimentations.

D2 : Relation entre la connaissance et la LPL La maîtrise de la « ligne de produits logiciels » repose aujourd'hui sur l'ingénieur logiciel, à l'intersection de plusieurs corps de métiers. Pour libérer le développeur et en même temps garantir la prise en compte des connaissances acquises, le défi principal est d'être capable d'exploiter automatiquement les connaissances au niveau des différents types d'assets qui constituent la ligne de produits et de garantir différentes propriétés sur le système résultant comme par exemple que l'ensemble des configurations de stimulation proposées pour un traitement chez l'Homme ont été "testées" auparavant, avec toutes les ambiguïtés à lever derrière ce terme.

D3 : Utiliser la LPL pour configurer des SIs dédiés Contrairement aux approches classiques, les produits issus de la ligne, suivant leur degré de maturité, peuvent être utilisés pour des besoins différents : R&D, telle qu'une recherche sur une nouvelle pathologie (*e.g.* état de l'art, affiner un traitement) ou production quand l'étape de R&D est jugée suffisamment concluante. Le SI dédié se trouve fortement impacté par cette distinction. Ainsi, suivant les besoins, le Product Designer qui sera amené à le configurer sera différent. Ceci impacte les choix dans la configuration du produit notamment sur les rôles utilisateurs. Il semble qu'il y ait un niveau intermédiaire de configuration (Hubaux *et al.*, 2009).

D4, D5 : Générer un produit utilisable issu de la configuration Le produit généré sera composé de deux modules qui doivent interagir : le logiciel pour réaliser les expérimentations et celui pour effectuer le recueil de résultats d'expérimentations. Le premier pilote les expérimentations tandis que le second permet de capitaliser sur les résultats de ces expérimentations. La problématique est donc de générer ces modules en tenant compte du modèle métier et de l'usage qui en sera fait.

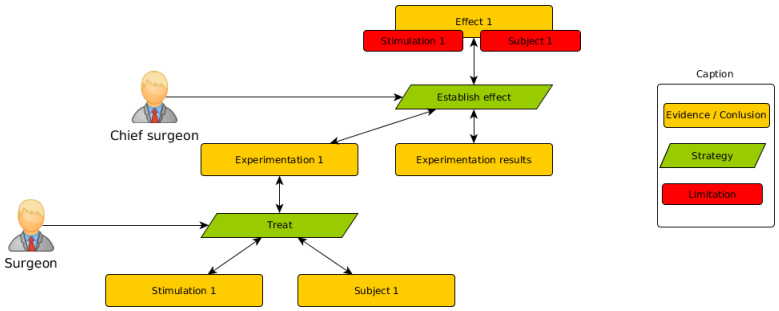

Figure 2. *Exemple de diagramme d'argumentation dans le contexte d'application du biomedical*

4. Actions réalisées

***Domain Specific Languages* (DSL) dédiés** Un DSL, pour le clinicien, a été défini permettant de collecter des contraintes (D1). Nous avons également traité la transformation de ces contraintes dans un modèle exploitable par la LPL abordant ainsi le défi D2.

LPL La caractérisation du domaine a été réalisée pour les produits existants au sein de la société via une représentation sous forme de Feature Models (Kang *et al.*, 1990). Nous avons aussi travaillé autour du module de recueil d'analyse expérimentations pour définir une maquette de la GUI répondant aux besoins des expérimentateurs, qui sont les cliniciens et plus généralement les professions médicales dans le cadre de l'entreprise (D3, D4).

Représentation de la connaissance Le travail sur cette GUI nous a permis de réaliser un premier modèle abstrait des informations à afficher et à recueillir (D5).

5. Actions futures

***Domain Specific Languages* (DSL) dédiés** Nous travaillons actuellement à définir un DSL, métamodèle et *Graphical User Interface* (GUI), pour un autre expert dans le but de se confronter à l'hétérogénéité des données (D1) et à la collecte des expérimentations (D5, D6). La gestion des incohérences qui peuvent apparaître dans la base de connaissance (D1, D6) fait partie des actions futures.

LPL La correspondance entre les contraintes issues de la base de connaissances et les assets reste un défi pour lequel les pistes envisagées portent sur des techniques de méta-modélisation (D2)(Favre *et al.*, 2006). Le développement du premier prototype du GUI d'expérimentation doit nous permettre d'établir les bases pour la production automatique idoine des GUI (D4, D5).

Représentation de la connaissance Il nous reste à valider ce modèle sur des expérimentations réelles puis à travailler sur leur transformation en données exploitables. Une piste nous vient de la théorie de l'argumentation (Polacsek, 2016)(D6). Les diagrammes d'argumentation permettent de structurer l'évolution de la connaissance par des pas d'argumentation. Chaque pas représente une analyse humaine ou automatique basée sur des évidences et amenant donc à un conclusion réutilisable comme une évidence pour les pas suivants (Figure 2).

6. Bibliographie

Bosch J., « From Software Product Lines to Software Ecosystems », , vol. 1, p. 111–119, 2009.

Buckley J., Mens T., Zenger M., Rashid A., Kniesel G., « Towards a taxonomy of software change », *Journal of Software Maintenance and Evolution*, vol. 17, n° 5, p. 309–332, 2005.

Clarke D., Proença J., « Towards a Theory of Views for Feature Models », *Proceedings of FMSPLE'10*, 2010.

Donoghue J. O., Roantree M., Boxtel M. V., « A Configurable Deep Network for high-dimensional clinical trial data », *Neural Networks (IJCNN), 2015 International Joint Conference on*, p. 1-8, July, 2015.

Favre J.-M., Establier J., Blay-Fornarino M. (éd.), *L'ingénierie dirigée par les modèles : au-delà du MDA*, Hermes-Lavoisier, Cachan, France, 2006.

Hubaux A., Classen A., Heymans P., « Formal Modelling of Feature Configuration Workflows », *SPLC'09*, IEEE, p. 221–230, 2009.

Kang K. C., Cohen S. G., Hess J. A., Novak W. A., Spencer Peterson A., Feature-oriented domain analysis (FODA) feasibility study, Technical Report n° November, The Software Engineering Institute, 1990.

Mens T., Demeyer S., Wermelinger M., Hirschfeld R., Ducasse S., Jazayeri M., « Challenges in software evolution », *International Workshop on Principles of Software Evolution (IWPSE)*, vol. 2005, p. 13–22, 2005.

Pohl K., Böckle G., Linden F. J. v. d., *Software Product Line Engineering : Foundations, Principles and Techniques*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

Polacsek T., « Argumentation Tree : A New Player in your Diagrams. », p. 223–224, 2016.

Seidl C., Heidenreich F., Assmann U., « Co-evolution of Models and Feature Mapping in Software Product Lines », *Proceedings of the 16th International Software Product Line Conference (SPLC)*, vol. 1, p. 76–85, 2012.

Wohlin C., Runeson P., Höst M., Ohlsson M. C., Regnell B., Wesslén A., *Experimentation in Software Engineering : An Introduction*, Kluwer Academic Publishers, Norwell, MA, USA, 2000.