

UIPLML: un outil basé sur les patrons pour l'ingénierie des systèmes d'information multi-plateformes

Thanh-Diane Nguyen¹, Jean Vanderdonckt¹, Ahmed Seffah²

*1-Louvain School of Management, Université catholique de Louvain, Place des Doyens, 1, B-1348 Louvain-la-Neuve, Belgium
{thanh-diane.nguyen, jean.vanderdonckt@uclouvain.be}*

*2-Innovation and Software, School of Business and Management, Lappeenranta University of Technology, P.O. Box 20, FI-5385, Lappeenranta (Finland)
ahmed.seffah@lut.fi*

Article accepté et présenté à la conférence internationale RCIS, Co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS.

RÉSUMÉ. Les patrons de conceptions (pattern en anglais) sont un outil qui a fait ses preuves à la fois dans la communauté IHM et en génie logiciel pour l'ingénierie des systèmes interactifs. Cependant, tel que documenté aujourd'hui, la grande majorité des patrons de conception des interfaces utilisateurs ne supportent pas le développement et l'adaptation des interfaces dites multi-contextes d'utilisation (multiplateformes, caractéristiques et expériences des utilisateurs et multi-environnement logiciel). L'utilisabilité n'est pas aussi prise en compte explicitement. Cette faiblesse rend aussi difficile la tâche des développeurs et gestionnaires des systèmes d'information de choisir le modèle de conception adéquat pour un cas particulier d'utilisation. Cet article introduit le langage UIPLML (User Interface Pattern Language Markup Language) qui exploite un format standardisé basée sur le XML qui permet d'apporter une solution satisfaisante à l'ensemble de ces faiblesses. Ce langage utilise le concept de patron génératif qui va au-delà de la simple documentation des patrons ; il propose d'intégrer dans la description des patrons, la stratégie de génération de code et d'adaptation de l'interface. Pour illustrer ce langage UIPLML, une application est présentée en utilisant le patron de conception : Master/Détails pattern. L'exemple décrit, en plus d'une définition complète, ce pattern incluant l'utilisabilité et la génération d'un fichier XML pour faciliter son implémentation dans le contexte des multiplateformes.

ABSTRACT (As it appears RCIS Version): Information systems are more accessible in the form service (Information systems as a service) by anybody from everywhere, anywhere, and at any time, from almost any device and computing platform. The continuous growth and the heterogeneity of these devices create various user experiences depending on the device and challenge designers and developers to creating methods and tools for engineering of usable, yet accessible, information systems. Instead of repeating a similar development life cycle, design patterns concentrate design solutions with embedded usability and accessibility. However, when an appropriate pattern is selected, the developer is responsible for adequately program the corresponding code, which is a tedious and error-prone task. In order to address these challenges, this paper motivates, presents, and defines UIPLML (User Interface Pattern Language Markup Language), a XML-compliant markup language for defining user

interfaces patterns for multiple contexts of use, e.g., for different users carrying out a task on different devices in different environments. A meta-model with new expressiveness enable multi-facet pattern matching. To validate it, four UIPLML pattern databases have been created: a base of 237 entries for multi-platform systems, a base of 42 entries for context-aware interfaces, a base of 10 entries for culturally-aware interfaces, and a base of 52 entries for accessibility issues. One particular pattern, i.e. the master/detail, is supported by a software for generative pattern-based approach in which application parameters and contextual data govern automated user interface XML creation which, in turns, generates code for multi-context information systems.

MOTS-CLÉS : interface utilisateur multiplateformes; contexte d'utilisation, génération de code,

KEYWORDS: Generative design pattern; multi-platform user interfaces, context of use, code generation, interactive information systems engineering

Résumé étendu

L'émergence d'une multitude de plateformes offre aussi une grande diversité d'accès aux systèmes d'information tout en supportant une très grande richesse en matière d'interaction. Les interfaces utilisateurs (IUs) doivent s'adapter à ces variations de contexte (utilisateurs, plateformes, environnements logiciel, organisation) afin de permettre à l'utilisateur de réaliser sa tâche avec efficacité, efficience et satisfaction.

Cependant, nous voyons émerger de plus en plus des plateformes différentes supportant des règles ergonomiques très variables de conception et d'évaluation d'IUs. Les développeurs et les designers d'IUs ont deux solutions pour résoudre ce problème. La première consiste à développer une interface pour chaque plateforme en respectant les critères d'utilisabilité propre à la plateforme. Cette solution n'est bien sûr pas techniquement viable compte tenu qu'elle nécessite une armée de développeurs. La seconde, celle que nous préconisons consiste à développer et adopter une approche de conception dirigée par les modèles qui utilisent les patrons de conception comme moyen d'encapsuler les règles ergonomiques et comme outil d'instanciation de modèles et de génération de code.

Les principes généraux de cette approche de conception dirigée par les modèles et basée sur les patrons, ont été largement discutés dans la littérature scientifique (Seffah, 2015). La section suivante donne un aperçu sur cet état de l'art. Il est important de noter que cet article discute uniquement de comment les patrons peuvent être documentés en intégrant les règles ergonomiques générales (adaptables sur toutes les plateformes) et spécialisés (accommodées sur une plateforme spécifique).

Les patrons comme : (1) outil de conception des interfaces (Wolff and Forbrig, 2010) (Märting et al., 2013) (Seffah et al, 2007) et (2) comme modèle pour la génération du code dirigé par les patrons (Vanderdonck and Simaro, 2010), ont été

largement décrits dans la littérature. Différentes collections de patrons de conception d'IHM sont ainsi disponibles (van Welie *et al.*, 2003).

Cependant, lorsque l'on examine ces langages et leur applicabilité dans l'approche de conception-dirigée par les modèles et basée sur les patrons, on peut observer les faiblesses suivantes:

- Un manque de consistance entre les différents formats de description de patterns (différents taxonomies, niveau de détail et présence d'homonymes).
- Un manque de structure de documentation en particulier dans l'uniformité dans les attributs de description,
- L'absence de suggestions d'implémentation du pattern pour différentes plateformes.
- Un manque de liens avec les approches de développement logiciel, comme par exemple l'approche par modèles qui est pourtant préconisée pour le développement des interfaces multiplateformes (Molina *et al.*, 2002).
- Un manque d'outillage permettant l'automatisation de l'application des patrons et de la génération de code.

Différentes extensions de ces langages de patrons ont été définies pour répondre à différents contextes d'utilisation (Engel *et al.*, 2015, Seffah, 2015). La première extension a été formulée par le groupe PLML (Pattern Language Markup Language) (Fincher, 2006). Différentes variations de PLML ont émergé afin de répondre à différents problèmes (Wendler *et al.*, 2013). En comparant toutes les extensions disponibles dans la littérature scientifique, nous remarquons que l'aspect multiplateformes et les critères ergonomiques dans ces interfaces ne sont pas suffisamment pris en compte et dans un même lieu.

Le langage UIPLML (User Interface Pattern Markup Language) que nous proposons est fondé sur le langage et le framework de développement des interfaces, UsiXML (Vanderdonckt, 2012). Il tire profit aussi du standard de W3C également utilisé dans le langage étendu PLML. Un aspect fondamental de UIPLM est un méta-modèle qui spécifie les règles ergonomiques, par exemple de (Scapin and Bastien, 1997). Ces règles ergonomiques ont été sélectionnées selon divers critères de l'utilisabilité et proviennent d'une sélection de références diversifiées, populaires et pertinentes. Les patrons de conception intégrés dans le langage UIPLML utilisent les règles ergonomiques décrites dans la base de données DESTINE (Figure 1). La base de données contient 256 enregistrements pour les systèmes multiplateformes, 42 enregistrements sur le contexte d'utilisation, 10 sur la culture et 51 sur l'accessibilité (voir table 1).

Type de pattern	Nombre d'Enregistrement	Objectif
Multi-plateformes patterns	237	Les patterns pour les smartphones, tablettes, ordinateur portables, PC, Pocket PC, PDA sont structurés en 8 catégories (taille de l'écran, page d'accueil, menu, contenu, actions, formulaires, contact)
Context-aware patterns	42	Les patterns sont classés selon la sensibilité du contexte dans les interfaces hommes machines
Culture-aware patterns	10	Les patterns ont classées selon des critères culturels (par exemple, les choix de couleurs, le langage, etc.)
Accessibility patterns	51	Les patterns sont classées selon le degré d'accessibilité pour différentes plateformes

Table 1. UIPLML Patterns dans la base de donnée de Destine Application

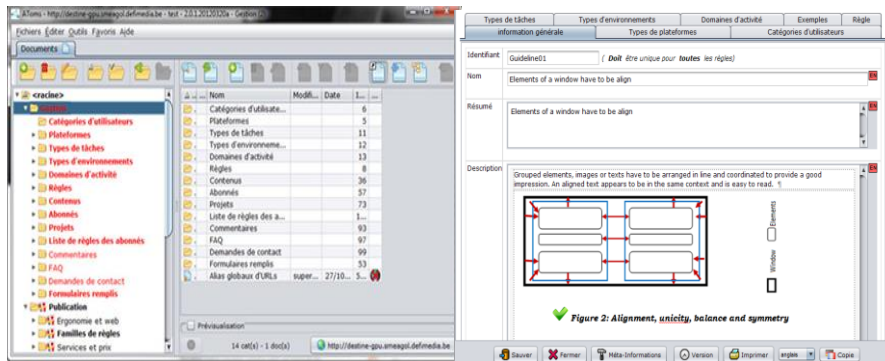


Figure 1. Editeur Correspondant à UIPLML Pattern

Adopter l'approche basée sur les patrons de conception permet d'assurer l'utilité et l'utilisabilité des produits pour les utilisateurs. En effet, les patterns décrivent les problèmes des utilisateurs avec une interface utilisateur au niveau conceptuel. Cela permet de les réutiliser comme briques de construction et de génération de code au niveau de l'implémentation. Leur force se trouve donc dans la description à haut niveau d'abstraction et l'illustration de leur adaptation qui permet rendre la tâche du développeur et designer plus aisée.

La figure 2 présente l'outil proposé de génération de code à partir de patterns. La figure illustre aussi comment les développeurs peuvent concevoir des interfaces graphiques en utilisant des modèles de haut niveau (image centrale de la figure 2) accompagné d'une structure (image à droite¹ de la figure 2) dans un système multiplateforme.

¹ Une version électronique de l'arbre de décision est disponible à l'adresse suivante : <http://usimad.alwaysdata.net/tree>

Pour illustrer le fonctionnement de cet outil, nous avons utilisé le cas le Master/Details Pattern (Molina *et al.*, 2002). Ce patron de conception permet de présenter une liste d'éléments (le Master). La sélection d'un élément permet de détailler celui-ci. La description de ce pattern dans le langage UIPLML inclut la spécification sous forme d'arborescence des différents modèles de représentation possibles avec des « widgets » d'affichage correspondants aux multiplateformes. La spécification est donnée dans un fichier XML. Celui-ci permet surtout de faire un lien entre les patterns et les règles ergonomiques. Il permet aussi de combiner ainsi l'expérience des patrons de conception et l'utilisabilité dans l'implémentation des interfaces utilisateurs multiplateformes.

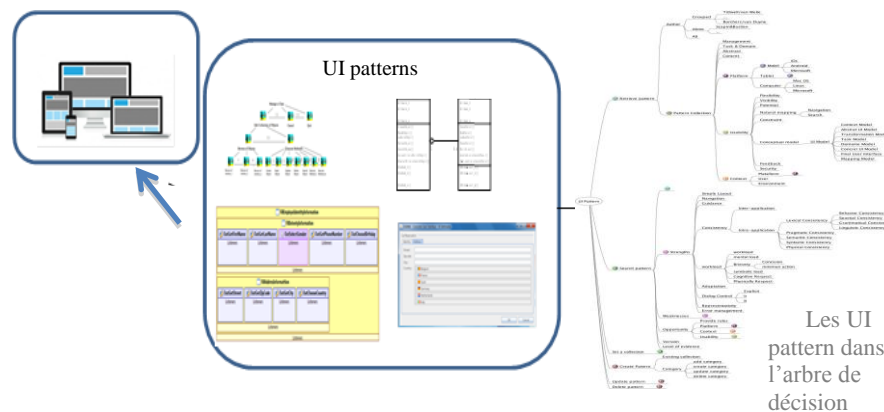


Figure 2. UIPLML pour la conception des multiplateformes

Le langage UIPLML proposé est loin d'être finalisé. Nous avons motivé dans ce papier les raisons et les principes de son développement. Nous avons aussi illustré brièvement l'usage de ce langage en utilisant le cas d'un patron particulier.

References

- Engel J., Martín C. and Forbrig P., (2015). A Concerted Model-driven and Pattern-based Framework for Developing User Interfaces of Interactive Ubiquitous Applications, in *Proc. of 1st Int. Workshop on Large-scale and Model-based Interactive Systems: Approaches and Challenges, LMIS'2015* (Duisburg, June 23, 2015), CEUR Workshop Proceedings, vol. 1380, CEUR-WS.org 2015, pp. 35-41.
- Fincher S., (2006). *PLML: Pattern Language Markup Language*, University of Kent, February 2006, accessible at <https://www.cs.kent.ac.uk/people/staff/saf/patterns/plml.html>
- Märtn C., Herdin C. and Engel J. (2013). Patterns and Models for Automated User Interface Construction - In Search of the Missing Links, in *Proc. of 15th Int. Conf. on Human-Computer Interaction: Human-Centred Design Approaches, Methods, Tools, and Environments HCI International'2013* (Las Vegas, July 21-26, 2013).
- Molina PJ., Santiago M. and Pastor O., (2002). User interface conceptual patterns," in *Proc.*

of the 9th Int. Workshop on Design, Specification, and Verification of Interactive Systems DSV-IS'2002 (Rostock, June 12-14, 2002), Lecture Notes in Comp. Science, vol. 2545, pp. 159-172, 2002.

Scapin D. and Bastien JMC. (1997). Ergonomic criteria for evaluating the ergonomic quality of interactive systems, *Behaviour & Information Technology*, vol. 16, no. 4/5, pp. 220-231, 1997.

Seffah A., (2015). Patterns of HCI design and HCI design of patterns - Bridging HCI design and model-driven software engineering, *Human Computer Interaction Series*, Berlin: Springer.

Van Welie M. Van der Veer G., (2003). Pattern languages in interaction design: structure and organization, in *Proc. of IFIP TC13 Int. Conf. on Human-Computer Interaction INTERACT'2003* (Zurich, September 1-5, 2003), Zurich: IOS Press, pp. 527-534, 2003.

Vanderdonckt J. and Simarro F. (2010). Generative pattern-based design of user interfaces, in *Proc. of the 1st Int. Workshop on Pattern-Driven Engineering of Interactive Computing Systems PEICS'2010* (Berlin, June 20, 2010), New York: ACM Press, pp. 12-19, 2010.

. Wendler S., Ammon D., Philippow I. and Streitferdt D. (2013), A factor model capturing requirements for generative user interface patterns, in *Proc. of 5th Int. Conf. on Pervasive Patterns and Applications PATTERNS'2013* (Valencia, May 27-June 1, 2013), Wilmington: Int. Acad., Research, and Industry Association, pp. 34-43, 2013