

Prototypage et développement de cockpit avec djnn

Stéphane Chatty¹, Stéphanie Rey¹, Stéphane Conversy¹, Mathieu Magnaudet¹,
Mathieu Poirier¹, Daniel Prun¹, Matthieu Pujos¹, Jean-Luc Vinot¹
Alexandre Lemort², Vincent Peyruqueou², Stéphane Vales², Yves Rinato³, Philippe Antoine⁴

¹Université de Toulouse - ENAC
7 av. Edouard Belin
31055 Toulouse, France
prénom.nom@enac.fr

²Ingenuity I/O
10, avenue de l'Europe
31520 Ramonville, France
prénom.nom@ingenuity.io

³Intactile Design
20 rue du Carré du Roi
34000 Montpellier, France
prénom.nom@intactile.com

⁴Aquinea Services
30 route de Belberaud
31450 Pompertuzat, France
prénom.nom@aquinea.fr

RÉSUMÉ

Dans cette démonstration, nous proposons de présenter les réalisations effectuées lors de quatre projets avec l'environnement de programmation djnn. Nous illustrons ainsi comment l'architecture de djnn permet d'aborder à la fois les exigences d'agilité liées aux processus de conception d'IHMs innovantes et les exigences du secteur aéronautique en matière de processus maîtrisés de développement et vérification de composants logiciels.

Mots Clés

User interface framework ; design process ; model-based architecture ; interactive component ; software verification.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces; D.2.2 Software Engineering: Design Tools and Techniques

INTRODUCTION

En matière d'ingénierie de l'IHM, on observe souvent deux catégories de travaux. D'une part, les chercheurs focalisés sur l'innovation mettent l'accent sur des processus agiles : la conception participative, le prototypage, les processus itératifs, etc. Ils cherchent des architectures capables d'exprimer des combinaisons toujours nouvelles d'entrées et de sorties. D'autre part, les chercheurs focalisés sur les processus industriels s'intéressent à des méthodes de spécification et de développement rigoureuses, afin de supporter des processus de vérification et de certification. Mais le secteur aéronautique ne peut se satisfaire de cette frontière. Il a besoin d'innover dans les IHM qu'il conçoit tout en respectant les contraintes imposées à ses cycles de développement. Qui plus est, il aspire à le faire en utilisant des outils communs pour les phases amont de conception et les phases aval de développement, car toute hétérogénéité est source d'inefficacité.

À travers quelques exemples, nous montrons comment le "framework" djnn offre des solutions à la fois en matière de conception itérative d'IHM innovante et de développement rigoureux.

LE FRAMEWORK DJNN

djnn met en oeuvre un paradigme de programmation interactive, dans lequel un programme est une collection hiérarchique de composants interactifs. L'exécution se décrit sous forme d'interactions entre les composants. Le

concept de composant interactif s'applique aussi bien à des widgets tels qu'un bouton ou une boîte de dialogue, qu'à des composants élémentaires tels qu'une forme graphique, une variable, une instruction de calcul, ou une structure de contrôle. Il s'applique aussi à des composants représentant des périphériques d'entrées ou de sortie, ce qui permet d'analyser l'interaction homme-machine comme une extension du modèle d'exécution de djnn.

Le framework djnn peut être compris comme un interpréteur de composants interactifs. Les composants sont soit créés avec une API, comme dans un framework traditionnel, soit chargés depuis des fichiers. En particulier la syntaxe XML est supportée, en utilisant les formats XML standards quand ils existent. Par exemple, djnn peut charger les composants graphiques depuis des fichiers SVG (Scalable Vector Graphics) ce qui permet à des designers graphiques de travailler de manière autonome sur les éléments graphiques avec leurs outils habituels, tandis que des programmeurs peuvent implémenter le comportement de ces éléments [1]. Le format XML peut aussi être utilisé pour effectuer des vérifications formelles sur l'arbre des composants.

Nous montrons ici l'utilisation de djnn à travers 4 projets de développement de cockpits aéronautiques auxquelles l'ENAC participe. Ces projets illustrent la diversité des processus de développement et des types d'IHM supportés par djnn.

VOLTA : UN HÉLICOPTÈRE ÉLECTRIQUE

Volta est un hélicoptère électrique conventionnel avec une motorisation 100% électrique (voir Figure 1). L'ENAC a collaboré à la mise en œuvre du tableau de bord de l'appareil. Pour ce projet, la contrainte temporelle était un facteur important, car les essais devaient être réalisés au plus tôt, dans un cadre de concurrence internationale.



Figure 1. L'hélicoptère électrique Volta de la société Aquinea

L'interface du tableau de bord de Volta est constituée d'indicateurs et de témoins (voir Figure 2). Sa fonction est essentiellement d'afficher l'état des systèmes de propulsion, le pilotage étant entièrement mécanique.

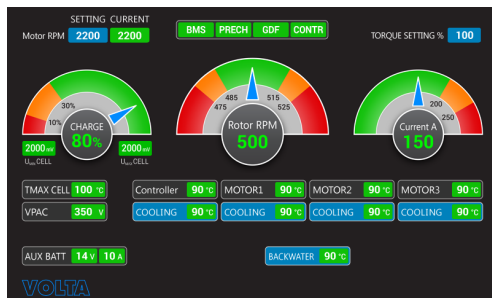


Figure 2. Le tableau de bord de l'hélicoptère électrique Volta

L'architecture de l'application réalisée est de type data-flow. Des composants ont été réalisés pour représenter les données reçues depuis les bus CAN et I2C, ainsi que les ports GPIO du Raspberry Pi. Le reste du programme est une collection d'objets graphiques chargés depuis des fichiers SVG, et de composants qui filtrent les données pour contrôler la couleur et la géométrie des objets graphiques.

COCKPIT POUR UN AVION ÉCOLE

L'ENAC mène une réflexion sur le cockpit des futurs avions école, destiné aux premières heures d'instruction des pilotes. En raison de la nature très particulière des vols effectués lors de ces premières heures, ces avions peuvent faire appel à des modes de propulsion moins bruyants mais à autonomie limitée. D'autre part leur planche de bord peut être conçue en fonction des besoins spécifiques de l'instruction.



Figure 3. Une vue d'ensemble du tableau de bord de l'avion école

Dans ce cadre, nous avons mené un processus de design centré sur les besoins de l'instructeur et avons pu explorer des solutions innovantes d'affichage et d'interaction répondant à la double spécificité de l'avion école : maîtrise de la consommation d'énergie et contexte d'instruction (Figure 3). Le découplage entre graphisme et comportement et la facilité d'intégration de ces deux composants nous ont permis de confier le design graphique à des experts et d'explorer rapidement différentes solutions favorisant ainsi le processus d'innovation.

EXPLORATION DE LA MULTIMODALITÉ

I Permission to make digital or hard copies of part or all of this work
 2 for personal or classroom use is granted without fee provided that
 3 copies are not made or distributed for profit or commercial
 4 advantage and that copies bear this notice and the full citation on
 5 the first page. Copyrights for third-party components of this work
 6 must be honored. For all other uses, contact the Owner/Author.
 7 Copyright is held by the owner/author(s).
 8 IHM '15, Oct 27-30 2015, Toulouse, France

une maquette de cockpit en carton développée par Intac-tile Design, permet de tester très rapidement de nouvelles combinaisons de modalités.



Figure 4. Le cockpit en carton

Le contenu des écrans du cockpit est produit par un vidéo projecteur (Figure 4). Les moyens d'interaction sont fixés sur le carton. Ils incluent la reconnaissance de gestes 3D (Leap Motion), le suivi du regard, la reconnaissance vocale, l'interaction aux doigts et stylet sur tablette, et des interacteurs tangibles (Phidgets). Le contenu des écrans interactifs peut être modifié en temps réel en utilisant un éditeur de type Inkscape ou Illustrator.

PROCESSUS DE DÉVELOPPEMENT INDUSTRIEL

Dans ce dernier exemple, nous nous appuyons sur un scénario concret de conception d'un PFD (Primary Flight Display), issu d'observations en milieu industriel, pour illustrer comment djnn permet de mettre en oeuvre les diverses phases des processus de développement.



Figure 5. Le Primary Flight Display (PFD) développé

Nous illustrons dans la démonstration quatre phases de développement : le travail en parallèle entre graphiste et programmeur, l'adaptation dynamique à un environnement de simulation plus fidèle, le test de plusieurs options de design, et la définition et la vérification formelle de propriétés [2].

BIBLIOGRAPHIE

1. Chatty S. Supporting multidisciplinary software composition for interactive applications. In Proc. LNCS, Springer Verlag (2008), 173–189.
2. Chatty S., Magnaudet M. & Prun D. Verification of properties of interactive components from their executable code. In Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (2015), 276–285.