

# EVALUATION D'UN OUTIL D'AIDE A LA SPECIFICATION FONCTIONNELLE DES SYSTEMES COMPLEXES

**Olga GOUBALI**

LIAS/ENSMA & Segula  
1 avenue Clément Ader,  
86961 Chasseneuil,  
France  
[olga.goubali@ensma.fr](mailto:olga.goubali@ensma.fr)

**Patrick Girard,  
Laurent Guittet**

LIAS/ENSMA  
1 avenue Clément Ader,  
86961 Chasseneuil, France  
[girard@ensma.fr](mailto:girard@ensma.fr)  
[laurent.guittet@ensma.fr](mailto:laurent.guittet@ensma.fr)

**Alain Bignon**

**Djamal Kesraoui  
Laurianne Boulhic**  
SEGULA Technologies  
BP 50256, 56602  
LANESTER Cedex, France  
[alain.bignon@segula.fr](mailto:alain.bignon@segula.fr)  
[djamal.kesraoui@segula.fr](mailto:djamal.kesraoui@segula.fr)  
[laurianne.boulhic@segula.fr](mailto:laurianne.boulhic@segula.fr)

**Pascal Berruet**

Université de Bretagne Sud  
Lab-STICC  
BP 92116  
56321, Lorient Cedex,  
France  
[pascal.berruet@univ-ubs.fr](mailto:pascal.berruet@univ-ubs.fr)

**Jean-Frédéric Bouillon**

ENSM  
38 Rue Gabriel Péri  
44103 Nantes, France  
Jean-  
[frédéric.bouillon@supmaritime.fr](mailto:frédéric.bouillon@supmaritime.fr)

## RÉSUMÉ

Pour la conception de la commande globale d'un système sociotechnique<sup>1</sup> complexe, les spécifications fonctionnelles sont à la charge des concepteurs du système à piloter car ce sont eux qui en maîtrisent le fonctionnement. Ces experts n'ont pourtant généralement pas les connaissances en programmation de ceux qui conçoivent le système de pilotage. Pour capturer la connaissance de l'expert sur le système et avoir plus facilement des spécifications fonctionnelles vérifiées et validées, sans le former aux méthodes formelles, nous proposons un démonstrateur d'outils de spécification de commande inspiré des techniques de la Programmation sur Exemple (PsE). Cet outil est composé d'une IHM de spécification connectée à deux modules : un module de généralisation et un module de génération d'interface de fonction. Dans le but de rendre l'IHM plus utilisable (ISO 9241-11 et ISO-13407), nous avons confronté cette interface à des utilisateurs expérimentés. Pour compléter notre analyse, un audit ergonomique portant sur l'utilisation de l'IHM est présenté. Cet article présente notre approche et démontre son utilisabilité au travers du cas d'étude d'un système auxiliaire de navire.

## MOTS CLES

IHM ; spécification fonctionnelle ; programmation sur exemple ; entretien semi-directif ; audit ergonomique.

## ABSTRACT

To design a sociotechnical and complex system, system designers are in charge of the functional specification because they have an expert knowledge of its operation. However, these experts do not usually have the programming knowledge of those who design the supervision system. To capture the knowledge of the expert on the system and to have more easily

verified and validated functional specifications, without training in formal methods, we propose a tool that is inspired by Programming by Example techniques (EBP). This tool is composed of a User Interface (UI) for specification connected to two modules: a generalization module and a function interface generation module. To validate the proposed interface, we have evaluated the usability (ISO 9241-11 and ISO-13407) of the system with real users. To complete our analysis, ergonomic audit on the use of the UI is shown. This paper presents our approach and demonstrates its usability.

## ACM Classification Keywords

Usability testing; Empirical studies in interaction design.

## INTRODUCTION

Nos travaux se placent dans le cadre l'obtention de la commande globale d'un système sociotechnique. Idéalement, les spécifications fonctionnelles décrivent la façon dont un système permet d'atteindre les buts de l'utilisateur. Elles permettent de traduire les exigences liées à la réalisation des tâches de l'utilisateur. L'expression claire des besoins et des exigences, et leur traduction en spécifications fonctionnelles, n'est cependant pas chose aisée, malgré l'importance de ces dernières pour la phase de conception [7].

Suivant la complexité du système, la description des spécifications fonctionnelles peut être fastidieuse. Ces spécifications sont les suites d'actions de l'opérateur sur le système, nécessaires à la réalisation d'une fonction et tenant compte de toutes les possibilités (configurations). Une bonne spécification fonctionnelle doit être correcte, sans ambiguïté, complète et cohérente. Certaines méthodes formelles ou semi-formelles [10] ont été proposées pour aider au respect de ces propriétés.

Les méthodes formelles ou semi-formelles sont connues pour analyser et valider le cahier des charges, limitant ainsi les erreurs d'interprétation. Cependant, face à la complexité des systèmes, l'écriture, la compréhension et l'utilisation des spécifications formelles restent encore un réel défi [3]. Dans certains secteurs de l'industrie, comme la construction navale, le langage naturel [15] reste privilégié car les spécificateurs n'ont pas la culture technique nécessaire

<sup>1</sup> Système en forte interaction avec l'humain (par exemple, un navire est un grand système sociotechnique). Le système sociotechnique complexe est composé d'interfaces de supervision et de programmes de commande associés.

à l'expression des spécifications dans d'autres langages. Le document de spécification reste donc sujet à interprétation et source d'incompréhension entre celui qui rédige la spécification (le concepteur du système à piloter) et celui qui l'exploite (le concepteur du système de pilotage).

Les erreurs qui découlent de l'interprétation des spécifications affectent le coût de développement et d'évolution des applications. En effet, 55% des erreurs détectées dans le cycle de vie d'un logiciel proviennent de la phase d'analyse [12]. Bien qu'il existe des techniques de détection de défauts dans les spécifications, elles nécessitent beaucoup d'efforts et ne permettent pas forcément de tout détecter [14].

La démarche que nous proposons consiste à offrir la possibilité à l'expert d'exprimer partiellement ses spécifications, puis de vérifier et valider les spécifications fonctionnelles générées, le tout à travers une interface intégrant des techniques de la Programmation sur Exemple (PsE) [9].

Depuis plusieurs années, les techniques de PsE ont démontré leur utilité dans de nombreux domaines. Extension du concept des enregistreurs de macros, ce concept permet à l'utilisateur d'enregistrer et de rejouer ses actions. Cependant, le rejeu des macros se réduit uniquement aux actions enregistrées, tandis que dans le cas de la PsE, le système produit une généralisation des actions enregistrées. La généralisation s'appuie sur les exemples enregistrés pour générer un programme. Les systèmes intégrant les techniques de la PsE enregistrent, généralisent et rejouent les actions de l'utilisateur à travers des interfaces particulières qui gardent l'objectif fonctionnel de l'application. Ainsi, un système de PsE permet à des utilisateurs non-informaticiens de programmer [9].

Nous avons mis en œuvre ces techniques au travers d'une preuve de concept qui consiste en un démonstrateur d'outils de spécification de commande (Figure 1) composé d'une part d'une interface de spécification et, d'autre part d'un généralisateur appliquant la théorie des Graphes. L'interface de spécification permet d'offrir la possibilité à l'expert d'exprimer de façon interactive, sur une maquette de l'interface de supervision du système, le comportement attendu par le système. Afin de vérifier nos choix de conception, une étude exploratoire a été réalisée auprès de plusieurs utilisateurs expérimentés par moyen d'entretiens semi-directifs. Cette étude a pour but de valider notre démarche de spécification et de vérifier l'utilisabilité de l'interface proposée car une absence de compatibilité entre, d'une part l'outil, et, d'autre part la logique des utilisateurs lors de la spécification peut entraîner des difficultés d'utilisation de l'interface. Nous complétons notre étude par un audit ergonomique. Contrairement aux tests utilisateurs, l'audit ergonomique ne nécessite pas de participants et se base sur des méthodes d'inspection des interfaces.

L'analyse des résultats de ces deux évaluations permettent de dégager quelques pistes d'amélioration pour l'outil proposé.

## OUTIL ET DEMARCHE DE SPECIFICATION

La Figure 1 présente, en détail, la démarche de spécification mise en œuvre dans notre démonstrateur. Cette démarche suit plusieurs phases. Pendant la première phase, l'expert utilise l'interface de spécification pour montrer des exemples de séquences d'exécution pour toutes les fonctions (exemples de spécifications fonctionnelles). Deux catégories de généralisation sont faites au cours de la deuxième phase : la généralisation du programme et la généralisation des configurations. La troisième phase permet d'utiliser les configurations généralisées et les modèles de tâches pour proposer les modèles d'interface pour le lancement des fonctions du système. Les configurations possibles générées ainsi que les modèles d'interface de fonctions doivent être vérifiées et validées par l'expert, lors de la dernière phase (4. Rejeu : vérification et Validation).

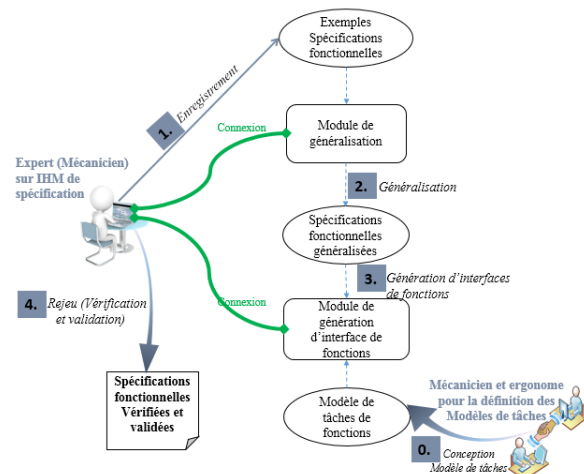


Figure 1. Outils de spécification

L'application de ces techniques à la conception des systèmes sociotechniques complexes dans le domaine naval nous a permis de valider la faisabilité de l'approche.

Le cas d'application considéré est un système de production, de distribution et de stockage de l'eau douce embarquée sur un navire, noté EdS<sup>1</sup>. Ce système dispose de 7 fonctions de haut niveau. À l'issue de travaux précédents, nous avons généré automatiquement et conjointement un programme de commande et une interface de supervision pour ce système [2], à partir d'un schéma PID<sup>2</sup>. Le PID est schéma technique unique qui définit tous les éléments d'un procédé industriel. L'interface de supervision générée ne permet cependant que des commandes

<sup>1</sup> EdS : Eau douce Sanitaire

<sup>2</sup> PID : Piping and instrumentation diagram

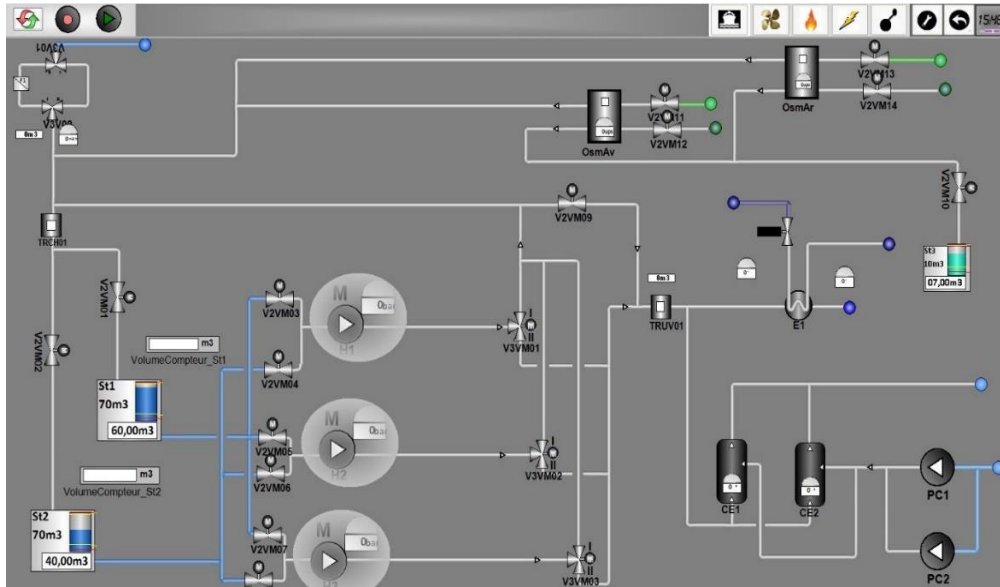


Figure 2. IHM de spécification conçue pour le système EdS

élémentaires du type « ouvrir » ou « fermer » une vanne. La réalisation des fonctions de haut niveau à partir de cette interface conduit à des interactions élément par élément entre le système et l'utilisateur.

Ces interactions sont toujours des séquences d'actions de l'utilisateur (par exemple l'ouverture de vannes puis le démarrage de pompes) et de réaction du système (par exemple la visualisation de l'écoulement du fluide) appartenant à des tâches de plus haut niveau, conformes aux buts de l'utilisateur. L'exécution de commandes globales permet mieux coller à ces buts, et ainsi de diminuer le nombre d'interactions nécessaires à la réalisation d'une fonction.

L'implémentation de ces commandes nécessite cependant la définition des spécifications fonctionnelles. Ces spécifications sont les suites d'actions de l'opérateur sur le système, nécessaires à la réalisation d'une fonction avec prise en compte de toutes les possibilités (configurations).

Pour notre cas d'application, 7 fonctions sont à spécifier (transfert, brassage, distribution embarquée, distribution depuis le quai, production, embarquement et débarquement). Compte-tenu de l'architecture du système, chaque fonction peut être réalisée suivant plusieurs configurations pour un total de 73 configurations unitaires à définir, nombre qui ne prend pas en compte la possibilité d'effectuer plusieurs fonctions simultanément. L'expert doit ainsi définir ces 73 configurations ainsi que tous les cas d'exécutions simultanées. Les spécifications fonctionnelles sont ensuite communiquées aux concepteurs de l'interface de supervision et du programme de commande qui sont en charge de les implémenter et de les intégrer au système.

Afin de vérifier la faisabilité de notre approche nous l'appliquons à la définition de la spécification des

fonctions du système EdS. L'IHM de spécification (Figure 2) est conçue sous le logiciel de type SCADA Panorama E2 [6]. Sa conception se base sur un modèle d'enregistreur/rejoueur générique adapté à partir des informations issues de modèles de tâches, et sur les vues graphiques des éléments du système à concevoir, stockées dans une bibliothèque d'éléments standard. Les widgets enregistreur/rejoueur se trouvant en haut à gauche de l'IHM (Figure 2), permettent la mise en œuvre des techniques de la PsE ; les widgets se trouvant en haut à droite sont décrits dans [13]. Le modèle de tâches de chaque fonction décrit les actions que l'opérateur en supervision effectue pour lancer, contrôler, surveiller les événements du système (les alertes, les messages de bon fonctionnement), évaluer et arrêter le transfert (s'il y a besoin).

L'utilisateur décrit les spécifications fonctionnelles à travers l'interface proposée suivant les trois étapes de la PsE : l'enregistrement, le rejeu et la généralisation.

```

Transfert1.vbs
1 PnSetValue"../SouteDepart.Value", St1
2 PnSetValue"../SouteArrivee.Value", St2
3 PnSetValue"../Volume.Value", 5
4 PnSetValue"../ValidationTransfert.Value", 1
5 Res = PnWaitProperty("../VolumeCompteur_St2", "=",
6 "../VolumeTransfert", 0)
7 PnSetValue"../ValidationTransfert.Value", 0
  
```

Figure 3. Exemple1 de programme enregistré pour la fonction de transfert

```

Transfert2.vbs
1 PnSetValue"../SouteDepart.Value", St2
2 PnSetValue"../SouteArrivee.Value", St1
3 PnSetValue"../Volume.Value", 10
4 PnSetValue"../ValidationTransfert.Value", 1
5 Res = PnWaitProperty("../VolumeCompteur_St1",
6 "=", "../VolumeTransfert", 0)
7 PnSetValue"../ValidationTransfert.Value", 0
  
```

Figure 4. Exemple2 de programme enregistré pour la fonction de transfert

```

Transfert.vbs
1 PnSetValue "../SouteDepart.Value", SouteDepart.Value
2 PnSetValue "../SouteArrivee.Value", SouteArrivee.Value
3 PnSetValue "../Volume.Value", Volume.Value
4 PnSetValue "../ValidationTransfert.Value", 1
5 Res = PnWaitProperty("../VolumeCompteur_" & SouteArrivee.Value,
6 "=", "../VolumeTransfert", 0)
7 PnSetValue "../ValidationTransfert.Value", 0

```

**Figure 5. Résultat de la généralisation des deux exemples de scripts enregistrés**

Lors de l'enregistrement, le système enregistre toute la séquence d'actions faite par l'expert ainsi que l'état des éléments manipulés pour la réalisation des fonctions. Pour être généralisée, deux exemples de chaque fonction doivent être enregistrés (Figure 3 et Figure 4).

Un système de *généralisation* utilise ces deux exemples pour en déduire un programme générique (Figure 5) prenant en compte les autres configurations possibles d'une fonction.

La généralisation des configurations suit plusieurs étapes. Tout d'abord l'utilisation d'un algorithme de combinaison permet de proposer à l'utilisateur tous les chemins possibles qu'un système peut utiliser pour atteindre l'objectif fonctionnel demandé. Ces chemins possibles sont validés par l'expert métier puis combinés avec l'utilisation d'un solveur et d'un algorithme de configuration. Le but est de fournir l'exhaustivité des possibilités de configuration permettant l'exécution des fonctions spécifiées. Une dernière étape consiste à utiliser les configurations généralisées et les modèles de tâches pour proposer les maquettes d'interface pour le lancement des fonctions du système. Les configurations possibles générées, ainsi que les modèles d'interface de fonctions doivent être vérifiées et validées par l'expert lors de la phase de jeu. Les configurations validées, ainsi que le programme générique seront utilisés pour la génération de l'interface de supervision et du programme de commande contenant les fonctions de haut niveau du système.

L'implémentation de l'interface de spécification nous a permis de valider notre approche d'utilisation de la PsE pour la description des spécifications fonctionnelles d'un système. Dans le but de rendre l'IHM plus utilisable, nous l'avons confrontée à des utilisateurs expérimentés.

## EVALUATION DE L'INTERFACE DE SPECIFICATION

### Objectif et contexte

L'objectif de notre étude est de vérifier d'une part les bénéfices de l'utilisation de la PsE dans une démarche de spécification fonctionnelle et, d'autre part, d'évaluer l'utilisabilité de l'interface de spécification proposée. L'intérêt sera de vérifier à posteriori, les choix de conception de l'interface, d'impliquer l'utilisateur (concepteur ou expert mécanicien) dans la conception de l'interface de spécification et d'identifier les points à modifier afin de rendre cette dernière plus utilisable. Pour cela nous avons évalué la spécification

fonctionnelle de la fonction *Transfert* auprès d'utilisateurs expérimentés, puis nous avons soumis l'interface à un audit ergonomique.

### Evaluation 1 : Entretiens semi-directifs

La méthode d'évaluation utilisée est l'entretien semi-directif qui vise à recueillir des informations spécifiques et de nature qualitative [5]. Cette technique est souvent employée pour la réalisation d'études exploratoires afin d'améliorer la connaissance d'un champ d'étude dont les thèmes essentiels sont familiers aux acteurs, mais qui présentent des aspects qui méritent un approfondissement.

L'entretien semi-directif était centré sur le thème de la spécification fonctionnelle des systèmes complexes. Le guide d'entretien utilisé pour notre expérimentation était articulé autour de douze questions, se rapportant à trois thèmes principaux. Ils ont été définis, d'une part, par rapport à la connaissance que nous avons et voulions acquérir du sujet et d'autre part, par rapport à l'utilisation et l'amélioration de notre outil.

#### Participants

Nous avons confronté l'interface à 5 participants (masculins) âgés entre 43 et 60 ans (49 ans en moyenne), ayant une expérience variant entre 1 et 23 ans (9 ans 1 mois en moyenne) et disposant tous d'une très bonne connaissance du système qui leur était présenté. Ils en connaissaient le fonctionnement et l'utilité. Ils ont déjà navigué sur des navires marchands ou de transport de passagers de toutes tailles. Certains de ces navires disposaient d'un système similaire au système EdS. Les participants étaient donc familiarisés avec ce type de système. Cependant, aucun d'entre eux n'a contribué à la conception de ce type de système.

#### Protocole

L'expérimentation s'est déroulée dans une salle réservée à cet effet, à l'Ecole Navale Supérieure Maritime (ENSM) de Nantes. Le dispositif expérimental était composé d'un écran de 23 pouces permettant d'afficher l'IHM de spécification, d'une souris et d'un clavier. Le participant s'asseyait en face de l'IHM et l'expérimentateur à côté de lui. Tout d'abord, le schéma PID construit par le mécanicien lui était présenté. Pour vérifier qu'il avait une bonne connaissance de ce type de système, il lui était demandé de décrire oralement le schéma (à quoi il sert, comment le système fonctionne...). Il lui était également demandé de décrire les étapes à suivre pour réaliser le transfert d'une soute à une autre, en se servant du même schéma. Ensuite, une simulation de modèles de tâches (les deux modèles de tâches décrivent comment réaliser le transfert avec et sans commande de haut niveau) lui était présentée à travers l'outil Prototask [8]. L'objectif de cette partie était de vérifier si la méthode utilisée pour définir (spécifier) la fonction de transfert, puis pour la lancer (en utilisation) est correcte. Ensuite, les participants étaient formés pendant environ 15 min à l'utilisation de l'interface de

spécification (la partie de l'interface qu'ils ne connaissent pas); puis il leur était demandé de l'utiliser pour spécifier la fonction *Transfert*. Enfin, ils répondaient à un questionnaire permettant de récolter leurs impressions et leurs critiques pour améliorer l'outil.

À des fins d'analyse, l'expérimentation a été enregistrée puis retranscrite.

#### Analyse des résultats

Les entretiens semi-directifs ont permis aux participants de donner leur perception de la démarche de spécification fonctionnelle. L'analyse des propos tenus par les participants tend à démontrer la généralité de notre approche car ils se sont souvent référés à d'autres systèmes auxiliaires de navire pour expliquer le système EdS. Tous les participants ont fait une bonne description du système. Ils connaissent l'architecture du système et son utilisation. Ils connaissent chacun des éléments et leurs rôles dans le système. Ils ont pu décrire le fonctionnement global, quelques fonctions et les éléments qui servent à réaliser les fonctions. Cette étape nous a permis de confirmer que même si les participants n'ont pas un profil « concepteur », ils ont suffisamment de connaissance du système pour en décrire les spécifications.

Les tâches décrites pour la réalisation d'un transfert en utilisant une commande ont toutes été validées à l'unanimité. Nous avons pu ainsi vérifier la conformité du modèle de tâches de fonctions ayant servi à la conception de l'interface de spécification. Quant aux tâches décrites pour la réalisation d'un transfert manuel, les participants projettent la réalisation de la fonction sur d'autres systèmes. De l'analyse de cette partie résultent deux scénarios possibles :

*Scénario 1 : Transfert par pompe sans sécurisation.*  
Pour ce type de transfert, on choisit le circuit, puis on ouvre les vannes concernées et on démarre la ou les pompes.

*Scénario 2 : Transfert par pompe avec sécurisation.*

Dans ce cas, on choisit la ou les pompes à utiliser, on ouvre les vannes autour dans le respect des règles de conduite. Ensuite on ouvre la ou les pompes choisies, puis on sécurise le circuit en vérifiant que les éléments qui peuvent altérer la fonction sont fermés.

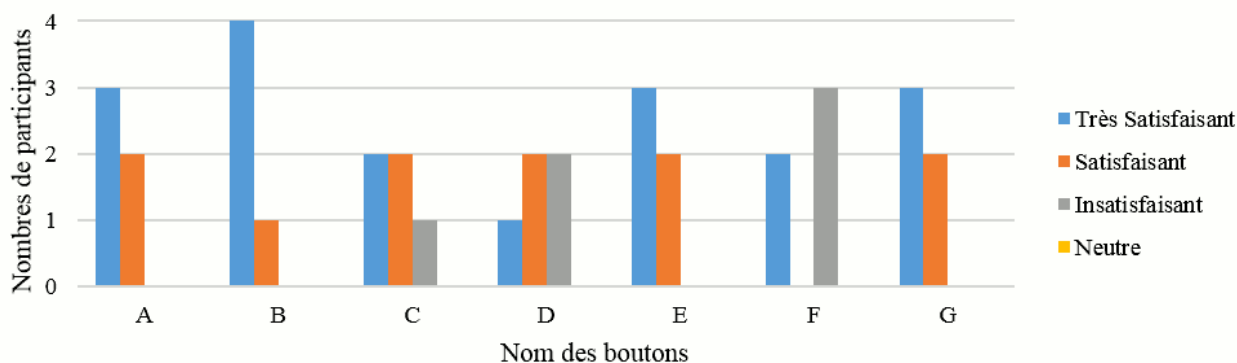
Nous avons constaté que le scénario 1 est le plus utilisé des deux par les participants. Cette analyse nous a permis de remarquer que les séquences d'actions élémentaires pour la réalisation d'une même fonction peuvent varier en fonction du type de système. Ainsi, nous avons validé l'utilité de notre démarche qui est de capturer pour chaque système, les connaissances de l'expert afin d'avoir les bonnes spécifications.

Les différentes étapes proposées sur l'interface pour atteindre la spécification d'une fonction ont toutes été validées. L'intuitivité des actions par rapport aux thèmes employés pour la réalisation des tâches de chaque phase est récapitulée sur la Figure 6.

Les participants préconisent de remplacer le terme « réseau » (dans la légende de la Figure 6) par « circuit ». Ceci explique par exemple le taux d'insatisfaits élevé sur la barre F (Figure 6).

Sur une échelle de 1 à 10, avec 1 « très facile » et 10 « très compliqué », le niveau de difficulté d'utilisation de l'IHM de spécification est placé à 4. Les participants ont tous signalé le fait que l'utilisation devient beaucoup plus facile après la spécification de la première fonction. Cependant, plusieurs points d'amélioration ont été identifiés.

Concernant la mise en œuvre de la spécification fonctionnelle à travers l'interface proposée, les participants ont relevé le manque de spécification de la « sécurisation » du système. Selon les participants, certaines erreurs de spécification apparaissent dans la définition de toutes les sécurités liées à chacune des fonctions, pour la robustesse du système. Il est donc essentiel de compléter les phases de spécification par une définition de contraintes permettant de rendre le système sûr suivant les différentes configurations.



A : Enregistrement des spécifications; B : Sélection départ/arrivé; C : Désignation des points de passage obligés; D : Configuration du réseau; E : Renseignement des conditions de fin; F : Arrêt des éléments du réseau; G : Finir enregistrement des spécifications

Figure 6. Évaluation des boutons

Parmi les moyens d'interaction présents sur l'interface se trouve le « glisser/déposer ». Nous avons remarqué que ce dernier perturbe les utilisateurs car il complique leur tâche.

D'un point de vue ergonomique, le manque d'information au niveau de la phase de rejeu a été remarqué par les participants. Pendant cette phase le concepteur doit lancer les fonctions afin de vérifier et de valider les spécifications produites par le système. Cependant, à partir du moment où il lance une fonction, il n'a aucune information sur cette dernière.

Les participants trouvent important que l'utilisateur puisse à tout moment connaître les fonctions en cours sur son système. Pour le passage d'une étape de spécification à une autre, il faut que les participants cliquent sur le bouton correspondant à l'étape à réaliser mais ceci n'est pas du tout évident pour eux malgré les messages pour les guider.

L'analyse des résultats de l'entretien semi-directif, nous a permis de confirmer que notre approche est une solution originale bien accueillie par tous les participants. Deux participants ont trouvé le démonstrateur facile d'utilisation (note de facilité d'utilisation inférieure à 3/10), trois participants ont éprouvé d'avantage de difficulté (notes comprises entre 5 et 6/10). Pour compléter notre analyse, un audit ergonomique de l'IHM basé sur les dix principes heuristiques de Jacob Nielsen [11], sur les critères de Bastien et Scapin [1] et sur les 12 règles de l'ergonomie web [4] a été réalisé.

- ✓ Pour la fonction transfert, seules les soutes sont cliquables. Le reste des éléments est grisé, indiquant à l'utilisateur qu'ils ne sont pas disponibles.
- ✓ Une fois que l'enregistrement est lancé, l'icône clignote.
- ✓ Le surlignage vert s'affiche au fur et à mesure de la progression dans le processus de spécification fonctionnelle.
- ✗ Le message de la fenêtre pop-up demande à l'utilisateur d'effectuer une action à la suite de celui-ci, sans lui expliquer comment. L'utilisateur s'attend alors à devoir effectuer une action lorsque cette fenêtre disparaîtra. Or, après avoir cliqué, il peut d'abord avoir l'impression que rien ne se passe, puis remarquer le message apparu en haut de l'écran lui demandant de cliquer sur le bouton « sélection départ/arrivée ». Lorsqu'il voit ce message, l'utilisateur s'attend à trouver un bouton, à un endroit approprié. Or, ce bouton passe complètement inaperçu car il est intégré à l'ensemble des boutons qui constituent les étapes de la spécification fonctionnelle. Ces deux messages viennent alourdir et ralentir inutilement le processus de spécification fonctionnelle.

**Suggestion :** Lorsque l'utilisateur démarre cette étape, une fenêtre pop-up apparaît, lui demandant s'il souhaite lancer l'enregistrement en lui laissant la possibilité d'annuler. Dans le cas où l'utilisateur clique sur OK, il passe directement à l'étape suivante, sans avoir à effectuer d'action supplémentaire.

**Référence :**

**Guidage (Bastien & Scapin) :** ensemble des moyens mis à disposition pour conseiller, informer orienter l'utilisateur (implicites ou explicites).

→ Guider l'utilisateur par des indices pour qu'il comprenne ses actions possibles et la façon de les réaliser (incitation).

→ Répondre à l'utilisateur pour l'informer sur l'action accomplie et sur son résultat (feed-back immédiat).

**Charge de travail (Bastien & Scapin) :** ensemble des éléments qui jouent un rôle dans la réduction de la charge perceptive, mnésique et l'augmentation de l'efficacité du dialogue.

→ Limiter les actions demandées à l'utilisateur pour atteindre son but (Actions minimales).

**Rapidité (Boucher) :** Éviter les actions inutiles. Pour y arriver, vous ne devez pas contraindre les internautes à des actions que vous pourriez leur éviter. Ne demandez pas à vos internautes un effort qui pourrait être pris en charge par la machine !

**Fenêtre pop-up →**

**Message dans barre des tâches ↓**



Figure 7. Méthode d'audit ergonomique

## Evaluation 2 : Audit ergonomique

### Méthode

L'audit ergonomique est une méthode dite « experte » qui consiste à analyser tout ou partie d'une interface en s'appuyant sur ses connaissances, expériences et convictions [4]. Il fait appel à la capacité d'analyse de l'ergonome ainsi qu'à son expérience et est soutenu par des critères ergonomiques [1]. Cette méthode dont un extrait est présenté sur la Figure 7, nous permettra d'aller plus loin que les problèmes mis en lumière par le test utilisateur.

### Résultats

Le problème majeur mis en lumière par l'audit ergonomique de l'interface est un guidage inapproprié. En effet, le système utilise des formes de guidage explicite pour orienter et aider l'opérateur dans la réalisation de ses tâches. Cependant, ces formes de guidage nécessitent des actions superflues de la part de l'opérateur et augmentent son temps de réalisation des tâches.

Les fenêtres contextuelles (ou fenêtres pop-up) sont utilisées tout au long du processus de spécification fonctionnelle pour communiquer des instructions au concepteur et le guider. Cette forme de guidage explicite n'est cependant pas appropriée car elle demande du temps et de l'attention supplémentaire à l'opérateur qui doit prendre connaissance de ces messages puis effectuer les actions nécessaires à la tâche. Les fenêtres augmentent considérablement le nombre de clics que l'opérateur doit effectuer au cours

du processus ainsi que sa charge mentale. De plus, il y a un risque que l'opérateur valide le message sans l'avoir lu, et soit donc perdu sur l'interface. En effet, les fenêtres contextuelles sont intrusives car elles interrompent brutalement l'opérateur dans sa tâche et peuvent être perçues comme indésirables. Il faut réserver ce mode de communication aux informations de haute importance pour la tâche et pour lesquelles il est nécessaire de s'adresser directement à l'opérateur.

*Les boutons de confirmation de début et de fin de tâche* nécessitent des actions qui devraient être prises en charge par le système. Là encore, on utilise une forme de guidage explicite qui nécessite des clics superflus. En effet, le passage d'une étape à l'autre devrait se faire via le bouton « valider » ou « enregistrer » qui est déjà présent et sur lequel l'opérateur doit cliquer à chaque fois. De plus, ces boutons sont mal placés car ils se situent dans la barre de progression et remplacent temporairement le nom de l'étape en cours.

*L'emplacement et l'apparence des éléments* sont sujets à caution. En effet, le système n'indique pas toujours clairement quelles sont les actions à effectuer pour réaliser la tâche, comment les effectuer et dans quel ordre. À certaines étapes du processus par exemple, l'opérateur doit effectuer une séquence d'actions dans un ordre précis, alors que l'interface lui fournit l'ensemble des champs à remplir en même temps, ce qui peut entraîner des difficultés. L'utilisation du glisser/déposer constitue un autre problème de guidage car aucun indice ne laisse penser à l'opérateur que ce type d'interaction est possible avec le système. De plus, il n'existe pas de tutoriel pour guider et informer l'opérateur lors de la première utilisation.

L'ensemble de ces éléments rend le processus très long à effectuer, très coûteux d'un point de vue cognitif et déplaisant pour l'opérateur. Les préconisations ont été de repenser l'interaction homme-machine en passant d'un guidage explicite à un guidage implicite en agissant sur l'agencement des champs à l'écran, leur ordre d'apparition ainsi que leurs représentations graphiques pour rendre le processus plus rapide, intuitif et autonome.

### **Discussion**

Des résultats différents mais complémentaires découlent de ces deux évaluations.

D'une part, les entretiens semi-directifs nous ont permis de démontrer que notre approche permet de faciliter la spécification fonctionnelle des systèmes complexes. D'autre part, notre approche a été bien reçue par les participants, ce qui démontre son acceptabilité. Tous les participants ont apprécié, la phase de rejeu pendant laquelle le système leur propose une maquette interactive de la commande de haut niveau qu'ils spécifient, telle qu'elle sera présentée sur l'IHM finale, et leur offre la possibilité à travers cette maquette de vérifier et de valider les configurations

générées par l'outil. En effet, dans ce type de système, il est essentiel que la validation des solutions proposée soit explicite.

Outre les problèmes d'interaction qui alourdissent le processus, d'importantes informations sur le processus de spécification nous permettront d'améliorer les fonctionnalités de notre démonstrateur d'outil de spécification. En effet, pendant le processus de spécification tous les participants relevaient le nombre important de clics à faire sur l'interface pour la réalisation de la tâche de spécification.

L'audit ergonomique a, quant à lui, relevé des problèmes liés à la *longueur du processus, au guidage explicite gênant* et au *manque de tutoriel*. L'un de ces problèmes (la longueur du processus) ressort également des résultats de l'entretien semi-directif. Cependant, quelques points positifs ont été également relevés. Ces points sont jugés en s'appuyant sur les critères définis par les travaux de [1 ; 11].

En effet, la barre d'outils de l'interface est composée à 100% d'icônes en accord avec la situation de référence, ce qui facilite la compréhension de la tâche de l'utilisateur. Pendant le processus de spécification, l'utilisateur a toujours la main pour contrôler les traitements en cours ce qui permet d'éviter des erreurs. Le système permet d'annuler et de refaire une action et fourni un retour approprié, en temps réel, à l'utilisateur.

L'interface dispose également d'une icône d'aide qui permet de fournir une aide à l'utilisateur s'il est perdu. En effet, bien qu'il soit préférable que le système puisse être utilisé sans le recours à une documentation, il peut cependant être nécessaire de fournir de l'aide et de la documentation. Les informations de ce type devraient être faciles à trouver, centrées sur la tâche de l'utilisateur, indiquer concrètement les étapes à suivre et ne pas être trop longues [11].

L'un des objectifs de l'audit ergonomique est de raccourcir le processus de spécification fonctionnelle en retravaillant le guidage au sein de l'interface. Les propositions d'amélioration qui ont été présentées après cette expertise répondent à cet objectif puisqu'elles permettent de réduire de plus de 40% le nombre d'actions que l'opérateur doit effectuer. Elles devraient également permettre de diminuer le nombre d'erreurs et d'augmenter l'utilisabilité perçue de l'interface au cours des prochaines sessions de test utilisateurs.

### **CONCLUSION ET PERSPECTIVES**

Le démonstrateur d'outils de spécification de commande présenté apporte une aide supplémentaire dans le processus de spécification fonctionnelle d'un système sociotechnique. Notre objectif est de réduire l'effort de spécification tout en obtenant des spécifications fonctionnelles vérifiées et validées à travers une interface familière au concepteur. L'introduction des techniques de la PsE dans cet outil

de spécification donne le pouvoir à l'expert d'exprimer naturellement par des clics sur une interface les spécifications fonctionnelles du système qu'il conçoit. L'utilisation de cette interface par l'expert non-informaticien permet de résoudre les problèmes de communication et d'interprétation, ce qui peut réduire considérablement le temps de spécification fonctionnelle d'un système, donc de la réalisation du projet. Cependant la conception de l'interface peut être fastidieuse en fonction de la complexité du système. Les travaux actuels visent à réduire cet effort de conception en offrant des méthodes et outils permettant de générer automatiquement l'IHM de spécification.

Bien que l'interface de spécification n'ait pas encore été générée, son implémentation nous a permis de valider notre approche d'utilisation de la PsE pour la description des spécifications fonctionnelles d'un système. L'utilisabilité (ISO 9241-11 et ISO-13407) de l'outil de spécification a été testée lors des entretiens semi-directifs avec les utilisateurs. Un audit ergonomique basé sur des méthodes d'expertises nous a permis d'apporter des solutions aux problèmes détectés lors des tests utilisateurs et de dégager d'autres points d'amélioration qui n'ont pas été détectés lors des tests. Les informations issues de ces analyses seront utilisées pour rendre l'IHM plus intuitive à l'utilisateur. L'IHM sera ensuite soumise à des tests utilisateurs plus stricts basés sur un questionnaire SUS (*System Usability Scale*).

Dans la suite de nos travaux, l'IHM de spécification sera générée automatiquement. La génération de l'IHM de spécification suivra les mêmes étapes que celle de l'IHM de bas niveau [2 ; 6]. Nous avons défini un flot de conception basé sur les principes de l'IDM, qui nous permettra de mettre en œuvre toute notre approche pour générer les modèles opérationnels. L'IHM de spécification améliorée à partir des résultats des tests utilisateurs et l'audit ergonomique, ainsi que l'IHM de haut-niveau générée à partir de ces spécifications, feront l'objet d'une évaluation expérimentale.

#### REMERCIEMENTS

Les auteurs tiennent à remercier Eric Le bris (Ingénieur conception) pour l'intérêt qu'il a accordé à notre projet et le temps passé à évaluer notre outil. Ils remercient également l'ENSM (Ecole Naval Supérieure maritime) de Nantes pour leur collaboration, ainsi que ses enseignants pour avoir participé à notre évaluation.

#### BIBLIOGRAPHIE

1. Bastien, J.M.C., Scapin, D. (1993). *Ergonomic Criteria for the Evaluation of Human-Computer interfaces*. Institut National de recherche en informatique et en automatique, France.
2. Bignon, A., A. Rossi et P. Berruet (2013). An integrated design flow for the joint generation of control and interfaces from a business model. *Computers in industry* 64. p. 634-649.
3. Bollin, A., & Rauner-Reithmayer, D. (2014, June). Formal specification comprehension: the art of reading and writing z. In *Proceedings of the 2nd FME Workshop on Formal Methods in Software Engineering* (pp. 3-9). ACM.
4. Boucher, A. (2009). *Ergonomie web : Pour des sites web efficaces* (2nde Ed.). Eyrolles.
5. Caumont, D. (2010). *Les études de marché*. Dunod.
6. Goubali, O., Bignon, A., Berruet, P., Girard, P., & Guittet, L. (2014.). Anaxagore, un exemple d'ingénierie dirigée par les modèles pour la supervision industrielle.
7. Kolski C. et Ezzedine H. Conception et évaluation des IHM de supervision : éléments méthodologiques. *Revue Génie Logiciel* (2003), 65, pp.2-11.
8. Lachaume, T., Girard, P., Guittet, L., & Fousse, A. (2012). ProtoTask, new task model simulator. In *Human-Centered Software Engineering* (pp. 323-330). Springer Berlin Heidelberg.
9. Lieberman, H., editor (2001). *Your wish is my command: Programming by example*. Morgan Kaufman.
10. MOUSSA, F., Kolski, C., & Riahi, M. (2006). Analyse des dysfonctionnements des systèmes complexes en amont de la conception des IHM: apports, difficultés, et étude de cas. *Revue d'Interaction Homme-Machine Vol, 7(2)*.
11. Nielsen, J. (1994). Heuristic evaluation. *Usability Inspection Methods*, 17 (1), 25-62.
12. Pham H. *System Software Reality* (Spring Series in Reality Engineering), Springer-Verlag, New York (2005).
13. Rechard, J., 2015. *Introduction des critères ergonomiques dans un système de génération automatique d'interfaces de supervision*. Université de Bretagne Sud.
14. Trudel, S. (2012). *Using the COSMIC functional size measurement method (ISO 19761) as a software requirements improvement mechanism*. Université du Québec.
15. Weigers, Karl E. 2003. *Software requirements*, 2nd. Redmond, Wash.: Microsoft Press, xii, 350 p.