



Rapport d'Adaptation des interfaces à l'environnement

Etude bibliographique et
comparaison technologique

Alexia Llorens M2 IHM

Cours encadré par Madame Pinna-Dery (Année 2014-2015)

Sommaire

Etude technologique de Bootstrap 3 et Foundation 5.....	3
Présentation rapide des frameworks	3
Bootstrap.....	4
Foundation	4
Responsive Web Design	5
Similitudes et Différences.....	6
Préprocesseurs	7
Interface, fonctionnalités et personnalisation	7
Unité de mesure	8
Navigateurs.....	8
« Grid »	8
« Mobile First ».....	9
Conclusion	9
Synthèse du travail de recherche.....	10
Présentation de l'article	10
Présentation de l'approche	11
Modèle	12
Avantages/Inconvénients et illustration	13
Synthèse	14

Etude technologique de Bootstrap 3 et Foundation 5

« Choisir entre Bootstrap et Foundation c'est comme choisir entre du vin blanc et du vin rouge, les deux ne se marient pas forcément avec les mêmes aliments, et tout dépend aussi du goût de chacun. »

Cette partie du document se concentrera sur une comparaison entre Bootstrap et Foundation. Le but n'est pas de montrer si l'un est « meilleur » que l'autre mais de donner des éléments de réponse à un choix d'utilisation. Néanmoins même si ces frameworks sont similaires nous verrons également que certains éléments rendent l'un et l'autre unique.

Présentation rapide des frameworks

Pour commencer il est important de rappeler ce qu'est un framework ici pour le web: c'est tout simplement un ensemble gratuit d'outils pour créer des sites web et des applications web. Des templates HTML et CSS sont fournis que ce soit sur les éléments d'interface, la navigation, l'apparence générale etc. On pourrait assimiler cela à une caisse à outils, souvent simples d'utilisation afin d'offrir un accès à un public large. Comme les sites web ont des structures similaires il est tout à fait logique de chercher à réutiliser certains éléments. (Nous rejoignons d'ailleurs ici notre sujet d'étude sur l'adaptation des ihm). Les frameworks étudiés ici n'ont pas la prétention de s'adresser aux néophytes car un minimum de connaissances web est nécessaire mais prétendent à faciliter la vie d'un développeur.

Le contexte d'usage de ces deux frameworks est mixte, aussi bien sur les plateformes d'environnement (qu'on parle ici de devices ou de navigateur), et utilisateur puisque le but est de toucher un public large.

A la conception ces frameworks offrent tout ce qui est nécessaire pour un développement sans trop de préoccupation sur la maintenance, au sens où il ne faudra pas maintenir une multitude de bibliothèques à jour. De plus ces frameworks sont fluides, ils ne sont pas là pour ralentir le site. Leur but principal à l'exécution est de fournir une bonne adaptation du site au device sur lequel il est visualisé (voir plus loin le paragraphe sur le Responsive Web Design).

Dans le cadre de l'adaptation ces deux frameworks sont vraiment très proches. Nous pouvons nuancer ce propos par exemple par le fait que l'un ou l'autre avait telle ou telle approche avant l'autre ou parce qu'il s'adresse à un public légèrement différent mais il est impossible de poser, sur des critères d'adaptation, un vrai écart de comportement entre ces frameworks et donc d'autant moins pouvoir conclure à la prévalence de l'un sur l'autre.

Bootstrap

Twitter's Bootstrap a une accroche qui nous fait de suite comprendre à quel point il se veut accessible « Designed for everyone, everywhere. Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes. »

Designed for everyone, everywhere.

Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes.



Preprocessors

Bootstrap ships with vanilla CSS, but its source code utilizes the two most popular CSS preprocessors, **Less** and **Sass**. Quickly get started with precompiled CSS or build on the source.



One framework, every device.

Bootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries.



Full of features

With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.

Bootstrap est né de la collaboration d'un designer et d'un développeur chez Twitter. Créé en 2010 par Mark Otto et Jacob Thornton il a d'abord été connu sous le nom de Twitter Blueprint. Il est par la suite devenu open-source(2011) et a trouvé son nom actuel. Bootstrap a su se faire rapidement reconnaître comme l'un des plus populaire framework front-end et projet open-source. Bootstrap trouve son énergie dans une communauté massivement active sur GitHub. Actuellement Bootstrap en est à la version 3 mais une version 4 est en préparation. Nous y reviendrons plus tard.

Foundation

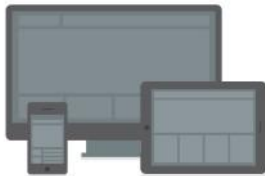
ZURB Foundation a été créé en 2011 et peut se vanter d'avoir été le premier framework responsive et clairement orienté « mobile first ». Les origines de ce framework datent de 2008 quand il a été décidé qu'il était nécessaire de disposer d'un framework pour prototyper rapidement, et de permettre la réutilisation d'éléments CSS, des pratiques, etc. Foundation s'affiche clairement comme ayant vu venir la popularisation de l'interaction mobile et d'avoir cherché dès le départ à solutionner le problème du passage à des écrans de tailles différentes, orientés différemment etc. Le framework en est actuellement à sa version 5.

Foundation a une communauté moins étendue que celle de Bootstrap mais est tout de même également l'un des projets open source les plus populaires.

Foundation's Prime Directive

The goals of Foundation 5

Our goal: to provide a framework that allows others (and ourselves) build better designed future-friendly sites. Being the fastest way to responsively prototype and build for any device is not enough. Foundation also has to make us better designers that thrive in the ever-evolving world of mobile devices. That's why we strive to make every new version of Foundation more advanced so you can **code smarter and faster**. [Explore the Foundation 5 Features →](#)



Faster for Users

Foundation 5 has a faster core that allows you to create beautifully optimized pages for whatever screen you want. The updated core also removes unnecessary wait time for users, especially for animations and transitions. The new updated visual style is simpler, cleaner, sexier and faster to restyle.

[View our core features →](#)



Faster to Code

With a medium grid and custom semantic breakpoints, creating websites and applications for any device is faster than ever. There's a new command line tool that allows you to spin up Foundation projects at super speeds. And our Foundation Business service allows you to receive professional support, consulting, tools and training for all your company projects.

[Ways to code faster →](#)



Faster to Learn

To help you improve your future-friendly website building skills, we've improved our documentation and created a community forum so you can learn more quickly. We've also written a new "Getting Started" guide that gets you up going with Foundation faster than you can make the Kessel Run. There's also our Foundation Forum where you can ask the entire Foundation community for help.

[Ways to learn with Foundation →](#)

Responsive Web Design

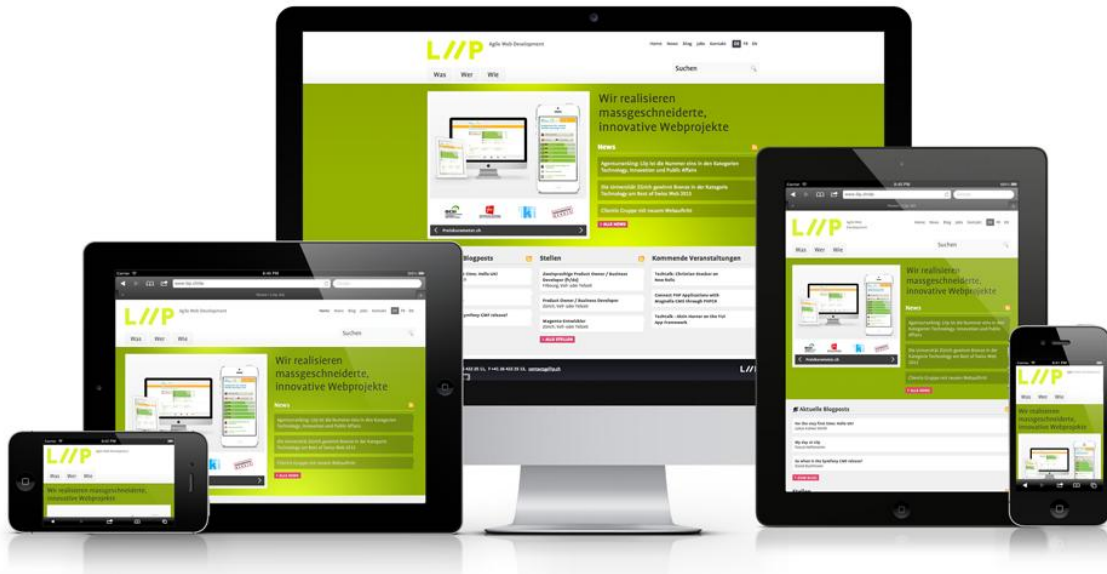
Avant de continuer il est important de comprendre ce qu'est « RWD ». S'il y a quelques années les notions d'adaptation d'une interface étaient peu prises en compte il est désormais aberrant de faire autrement. Cette approche est relativement simple : l'idée est qu'un site designé avec du responsive web design sait adapter son apparence à l'environnement dans lequel il est perçu. Pour parler plus clairement, l'idée est qu'un site observé sur un grand écran d'ordinateur ne doit pas avoir la même interface que le même site observé sur un écran de smartphone.

Il y a plusieurs façons de faire, soit par un principe de grille les éléments sont placés à des positions relatives et leurs tailles sont adaptées en fonction de l'appareil sur lequel on visualise la page, soit on opte pour un design foncièrement différent. C'est-à-dire que l'on va faire le choix de ne pas afficher certains éléments, ou de ne les afficher que si l'utilisateur fait une certaine action. Le but étant ainsi, s'il est sur smartphone, de lui fournir le contenu le plus pertinent et le plus intéressant, présenté de la meilleure façon possible.

Généralement ces deux manières sont couplées ce qui permet une meilleure expérience utilisateur. Ainsi par exemple si sur mon ordinateur de bureau je réduis progressivement la taille de ma page internet certains éléments vont passer, par exemple, en dessous d'autres sous forme d'un tableau à colonnes au lieu d'une longue ligne, et par exemple quand ma page sera vraiment trop réduite je

n'afficherais plus qu'une colonne globale contenant simplement les éléments les plus pertinents pour ma navigation.

Ci-dessous une image d'exemple parmi tant d'autres :



Similitudes et Différences

Il est réellement complexe de faire une différenciation extrêmement nette entre ces deux frameworks puisqu'ils ont tout de même un but similaire. Néanmoins certaines différences peuvent être un réel critère de choix pour le développeur, encore une fois cela dépend de chacun.

Dualité dans l'approche

Ces deux frameworks n'ont pas été imaginé dans la même optique. Et chacun l'affirme d'ailleurs de façon tout à fait claire.

Bootstrap essaye de fournir tout ce qui est nécessaire à la bonne création du projet. Foundation donne, vous l'aurez deviné, les fondations sur lesquelles construire. La philosophie de Bootstrap est du développement rapide de prototype là où la philosophie de Foundation serait plus tôt de faire sa propre approche du design.

Reprenons en pages 4 et 5 les images de présentation des deux frameworks. Nous pouvons voir que celles-ci se distinguent l'une de l'autre.

Foundation semble plus concentré sur l'utilisateur là où Bootstrap semble « jouer sur tous les tableaux ». Foundation se base sur la rapidité dans tous les domaines là où Bootstrap se base sur l'abondance. Ce sont là des différences fondamentales.

Foundation est plus à destination d'un public de développeurs qui a déjà un certain niveau de programmation.

Préprocesseurs

Cette étude ne s'étend pas à la comparaison entre les préprocesseurs Sass et Less. Non pas que la comparaison ne soit pas intéressante, loin de là, mais elle apporterait peu d'éléments très pertinents à l'étude présentée ici. Il y a tout de même quelques petites choses à savoir. Un préprocesseur CSS (Sass, Less et Stylus étant les plus connus) a une façon bien à lui de remplir une tâche donnée, il est une sorte d'abstraction au-dessus du CSS qui permet l'utilisation de fonctionnalités utiles dépassant les limitations normales du CSS (limitations qui font d'ailleurs toute son accessibilité à un large public) tout en gardant une propreté de code et sans avoir à tenir compte des compatibilités de navigateur. Cela permet de gagner du temps sur le développement (utilisation de syntaxe) et d'avoir donc l'accès à des fonctionnalités supplémentaires.

La différence sur ce point entre Bootstrap et Foundation c'est que Bootstrap possède les deux préprocesseurs Sass et Less alors que Foundation ne permet l'utilisation que du préprocesseur Sass. En soit comme dans un même site web il n'est pas possible d'utiliser deux préprocesseurs, est-ce là un réel handicap ?

Interface, fonctionnalités et personnalisation

Dans la continuité d'une approche différente Foundation offre un nombre limité d'éléments d'interface (« UI elements ») là où Bootstrap essaye de proposer presque tout ce qui peut être fait. Le fait que la communauté soit très étendue aide beaucoup dans cette démarche. Mais ce n'est pas cela qui fait que Foundation soit plus limitée sur ce point, c'est justement leur ambition. Pour eux c'est à l'utilisateur d'imaginer le reste, ainsi il peut le customiser autant qu'il le désire. C'est en effet un reproche qui est fait à Bootstrap : un site fait à l'aide de Bootstrap a le « Bootstrap look », autrement dit on le reconnaît assez facilement. Heureusement il existe une grande variété de thèmes et d'options de customisation pour rendre unique l'apparence du site web. Mais cette opération reste plus aisée avec Foundation. L'abondance des thèmes de Bootstrap montre toujours une volonté d'efficacité et de productivité quand Foundation s'attarde plus sur l'unicité du site. Foundation est plus faite pour un designer qui veut un contrôle maximal sur sa personnalisation. Le bilan de ce paragraphe est très mitigé, il s'agit vraiment du goût de chacun, préfère-t-on une interface entre guillemets clé en main un peu plus complexe à modifier ou partir de moins pré-fait et plus libre ? La réponse à cette question peut dépendre de plusieurs choses : le goût du développeur, le temps disponible, les exigences du client, en effet si celui-ci veut quelque chose du même acabit qu'un site et qu'un thème de Bootstrap s'en rapproche légèrement il est tout à fait judicieux de partir sur cette base. Foundation a des fonctionnalités totalement uniques par exemple « Interchange » qui permet le chargement plus pratique du contenu responsive, là où Bootstrap a des fonctionnalités qui donnent une sensation d'être plus complètes, reposant sur des structures solides et dont l'efficacité est prouvée.

Unité de mesure

Les unités de taille sont utiles sur tous les points : la taille des images, les polices d'écriture, les largeurs de tableaux, boutons, éléments, etc. Attention il est important de préciser que malgré un choix d'unité différente les deux frameworks obtiennent des résultats similaires.

Nous avons donc Bootstrap en pixels et Foundation avec les REMs. Il est néanmoins utile de préciser que Foundation fournit avec Sass la possibilité de convertir des pixels en REMs afin de pouvoir réfléchir en pixels encore. Du côté de Bootstrap la version 4 abandonnant le support d'IE8 ils passeront également aux REMs, mais il est fort probable qu'ils utiliseront aussi la possibilité de réfléchir en pixels et de faire ensuite la conversion afin de ne pas déstabiliser les fervents utilisateurs des pixels.

L'utilisation des pixels implique de définir une taille absolue pour le moindre composant sur tous les devices visés. Dans un temps où il est strictement impossible de définir une norme sur ce point l'utilisation des pixels devient de plus en plus obsolète. A l'heure du responsive c'est le relatif qui fait foi. Pourtant les pixels sont un repère pour encore beaucoup de monde, plus familiers, plus accessibles. Si la personne ressent un réel besoin de travailler avec quelque chose où elle se sent plus à l'aise alors Bootstrap est plus adéquat.

Le fait est que l'arrivée de la version 4 devrait contenter tout le monde (encore une fois sans trop s'avancer pour autant il paraît logique qu'ils mettront en place quelque chose de similaire à Foundation).

Navigateurs

Chrome, Safari, Firefox, Opera, pas de mise à l'écart de ce côté-ci de la part d'aucun des deux frameworks. C'est sur le point d'Internet Explorer que se fait une petite nuance : Bootstrap supporte encore la version 8 là où Foundation ne supporte que ce qui est supérieure ou équivalent à la version 9. Sauf que comme dit plus haut la version 4 de Bootstrap prévoit l'abandon de IE8. Aussi ils auront des navigateurs similaires. Sur ce point les deux frameworks ont-ils raison ? Il reste encore un pourcentage, non négligeable de gens possédant des versions inférieures à IE8. Ceci étant dit, comme cela leur instaure de potentielles limitations de développement il est compréhensible qu'ils préfèrent se focaliser sur la majorité.

« Grid »

Dans un premier temps Foundation s'est fait connaître avec son système de grille. Néanmoins Bootstrap l'a vite rattrapé, réduisant cet avantage à presque néant. En effet les deux frameworks offrent un système de grille par défaut parfaitement customisable. Bootstrap utilise des « breakpoints » ce qui fait que certaines tailles sont statiques, sauf qu'à l'aide d'une classe la largeur des colonnes peut dépendre de pourcentage (et donc être aussi fluide et flexible que celle de Foundation). La syntaxe de construction des grilles est différente (on en revient toujours aux préférences de chacun), mais les codes produits ont le même fonctionnement. Foundation a tout de même quelques fonctionnalités sur les grilles qui lui sont propre comme le « block grid » qui permet aux designers de diviser plus aisément le contenu, le présenter plus facilement. Quand nous reprenons la phrase précédente nous ne pouvons nous empêcher de voir les mots responsive et mobile clignoter.

« Mobile First »

Parlons d'ailleurs de cette approche Mobile First chère à Foundation. Bootstrap a plus tendance à se définir « desktop, tablet, mobile » c'est une façon de montrer qu'il ne se cantonne pas à une classe de devices et est toujours accessible (encore aussi cette notion d'abondance).

Mais Foundation pousse à son paroxysme l'approche Mobile First, en effet tout ce qui n'est pas encadré d'un media query est considéré comme étant du mobile. Si on veut une apparence différente sous tablette ou ordinateur il faut un media query pour le préciser. Le développement est donc aussi mobile first. Bootstrap considère plus que l'utilisateur verra ce que voit l'utilisateur sous ordinateur si jamais les media query n'ont pas été mis.

L'approche de Foundation sur ce point est très intéressante car elle fait appel à un esprit de synthèse : en effet elle tend à considérer qu'il est important de se concentrer sur le contenu le plus pertinent pour l'utilisateur et améliorer son expérience si de la place supplémentaire est disponible. Réduire du contenu quand il a d'abord été imaginé pour un pc est plus complexe. Après si l'on considère une situation où la personne a dès le départ maquetter ses interfaces pour les différents devices le problème est nettement réduit.

Conclusion

Cette étude n'avait pas pour but de mener à une conclusion sur la suprématie d'un framework par rapport à l'autre mais plutôt de comprendre les questions qu'il est essentiel de se poser quand vient un jour la question de « Bootstrap ou Foundation ? ». Trouver deux personnes qui auront la même justification sur l'utilisation d'un framework ou de l'autre est peu probable. Il est en plus compliqué de résumer les éléments qui peuvent pousser au choix car comme ces frameworks évoluent beaucoup ils savent corriger les points qui peuvent leur être reprochés. Globalement il faut se demander si l'on désire quelque chose rapide à mettre en place ou quelque chose où plus de réflexion est demandé et qui ne nécessite que des bases ? Désire-t-on des thèmes divers et variés qui ne nécessitent que des ajustements ou au contraire l'établissement complet d'un design ? Veut-on travailler avec des pixels ou des REMs ? Avec une grille plus ou moins robuste selon l'usage ? Est-ce que l'approche mobile first est le but premier du site à mettre en place ? Quelles sont nos capacités de développement ? Quel temps de disponible ? La liste pourrait être encore plus longue. Cette myriade de questions nous fait bien comprendre que chacun doit poser ses propres critères qu'il juge absolument fondamentaux, se baser sur sa propre expérience et ses propres préférences.

Dans tous les cas que ce soit Bootstrap et Foundation le choix sera bon, ce sont deux excellents frameworks, la nuance se fera sur de la simplicité sur certains points.

Synthèse du travail de recherche

Cette partie du document se concentrera sur l'analyse d'un article dans le contexte de l'adaptation.

Article choisi : **A User-centered Approach for the Design and Implementation of KDD-based DSS: A case study in the Healthcare Domain**

Lien: <http://www.univ-valenciennes.fr/LAMIH-intra/site/specifique/publications/204838>

Comparé à d'autres articles celui-ci est assez abstrait sur le point de l'interface, en revanche je l'ai sélectionné par rapport à son rapport avec les méthodologies de l'ihm, et les points sur lesquels de l'adaptation est faisable.

Présentation de l'article

Dans un premier temps il est important de présenter quelques termes utilisés par l'article pour une meilleure compréhension :

DSS : « Decision Support System » Système d'aide à la décision : le principe de ce genre de système consiste en un logiciel interactif qui aide les décideurs à déterminer les informations les plus utiles à partir d'un ensemble de données afin d'identifier des incidents et prendre des décisions.

KDD « Knowledge Discovery from Databases » : Processus de découverte de connaissances utiles depuis une collection de données.

Data mining : phase d'analyse du processus KDD.

Nous allons essayer de simplifier un peu les problèmes posés par l'article car ils sont assez complexes. En gros le processus de décision est basé en trois étapes : la recherche d'informations, avec l'identification et le fait de poser une problématique qu'il faut gérer, le processus de design qui inclut la génération, le développement et l'analyse de différentes séries d'actions (donc une solution) possibles et enfin le choix d'une solution ce qui comprend la recherche, l'évaluation et la sélection de la solution la plus appropriée. Le problème de cette approche est que la personne, quand bien même soit-elle experte du domaine, n'a pas forcément idée de comment identifier le problème de base. La nécessité d'un outil capable de faire des corrélations dans un ensemble de données prend tout son sens. Si l'on peut proposer à une personne un ensemble d'éventuels problèmes reliés à l'ensemble de données son expertise lui permet de trouver lequel est réellement un problème. Ensuite l'expert ne peut que se baser sur des solutions déjà vues ou explorées dans sa recherche de solutions et enfin il n'a pas forcément d'heuristique qui lui permette d'évaluer la sélection effectuée. On peut voir que ce problème est assez volumineux. Hors il y a pourtant besoin des DSS, la complexité des problèmes, l'impact de ceux-ci justifient d'un tel besoin. Où est-il possible d'agir et dans quelle mesure ? Les auteurs nous exposent ainsi leur avis qu'il faudrait une meilleure synergie entre l'humain et les processus automatisés. Que le succès ou non d'un DSS soit dépendant de ses capacités à interagir avec l'humain plutôt que sur des capacités de performance comme de la rapidité ou de solveur, est un fait inquiétant. Une plus grande intervention de l'humain s'avère donc être nécessaire. Celui-ci ne doit pas être un spectateur mais au contraire tenir un rôle actif. Comment faire ? Avec une approche orientée utilisateur. L'alliance des compétences de l'ordinateur et de l'humain, l'interactivité dans toute sa « splendeur ».

L'importance du processus KDD a été montrée plus haut, mais ceci est encore une fois un processus indépendant de l'humain. Les auteurs veulent établir une solution de DSS basé sur du KDD avec l'aide d'une approche centrée utilisateur.

Présentation de l'approche

Entrons plus dans le détail de ladite approche. Prédire le futur en fonction du passé est très ambitieux, mais possible, mais cela nécessite une validation de la part de l'utilisateur. Les auteurs nous expliquent plus en détail les modules nécessaires à la bonne implémentation de leur « KDD-based DSS ». Des activités d'HCI (Human Computer Interaction) seraient adéquates pour guider les utilisateurs au travers de chaque étape du processus KDD. Mais il est intéressant de voir également que les auteurs voudraient que cette approche s'adapte autant que possible au « decision-maker » nous avons donc une volonté d'adaptation à l'utilisateur.

La suite de l'article n'est, ni plus ni moins, qu'une analyse de l'existant, choses bien connues en conception ihm. Le problème des modèles de développement logiciel est leur terrible manque de considération ihm. En effet ces modèles ont plus un but d'évaluation technique du futur système proposé. Pour autant un tel modèle leur est nécessaire dans leur approche. Les méthodologies agiles se rapprochent déjà plus de ce qu'ils désirent de par leur prise en compte de la satisfaction de l'utilisateur. Mais c'est le modèle « Unified Process » (UP) qui va retenir notre attention. Pour avoir déjà travaillé avec ce modèle, en particulier l'implémentation Rational UP, l'un de ses points forts est de fournir une méthode de développement guidée par les besoins utilisateurs, ce qui répond tout à fait au désir des auteurs. De plus cette méthode étant itérative et incrémentale peut venir idéalement s'imbriquer dans le processus de KDD, en effet comme les résultats peuvent être soumis plusieurs fois au processus, une sorte de schéma temporel logique nous vient en tête. Nous avons donc là un modèle côté développement qui se rapproche du but recherché. UP est en quelques sortes l'exception parmi les autres modèles puisqu'il est moins focalisé sur la technique et plus sur le besoin utilisateur. Néanmoins cette approche ne mentionne pas explicitement l'utilisateur et à quel point celui-ci est impliqué dans le processus. Aussi il est nécessaire d'enrichir cela avec des modèles de développement « HCI-enriched ».

Ici c'est le modèle en U qui va prévaloir (pour lequel il semblerait d'ailleurs qu'il soit assez complexe de trouver beaucoup de données le concernant). Le principe de ce modèle est en tout cas de montrer la validation d'un système homme-machine ou de souligner ses déficiences et déterminer plus en détails celles-ci. Le résultat est de généraliser des comportements spécifiques de l'utilisateur et de réutiliser ces éléments dans des situations à configuration similaire. La réutilisation est un point assez important de l'adaptation pour le souligner ici.

On a donc un contexte d'usage utilisateur mais aussi environnement, on veut s'adapter à l'utilisateur mais également pouvoir réappliquer l'approche à d'autres environnements (nous verrons plus loin lesquels dans des cas pratiques).

Pour en revenir à l'article l'approche cumule donc le modèle en U et le modèle UP appliqué au DSS basé sur du KDD.

Modèle

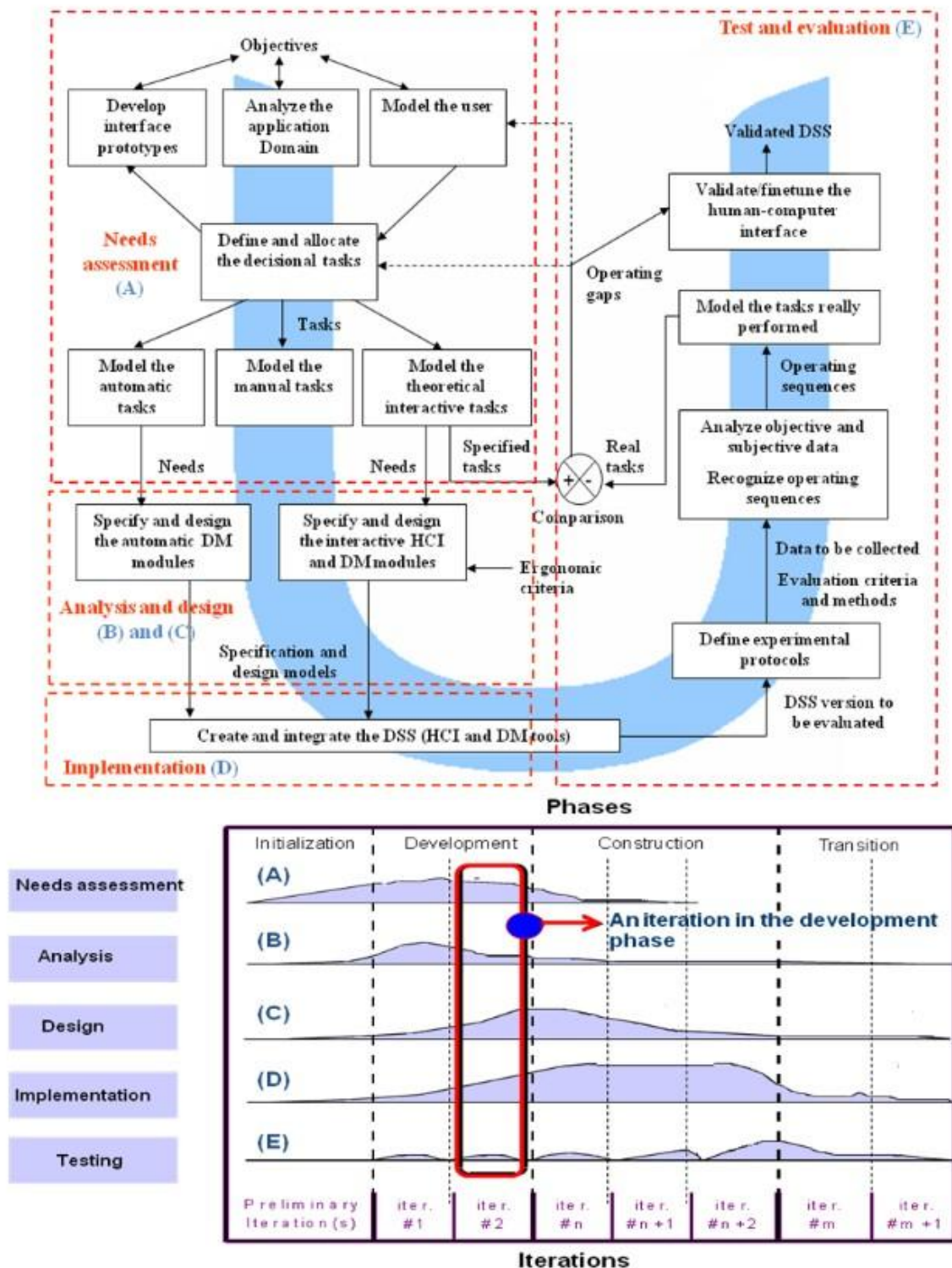


Figure 6: The UP/U approach

Avantages/Inconvénients et illustration

L'avantage d'une telle approche est pour commencer son côté innovant. L'approche présentée semble répondre en bonne logique aux différents problèmes répertoriés dans l'article auxquels l'existant ne trouve pas réponse. Travailler sur la complémentarité de deux modèles de sous-domaines différents afin de les exécuter ensemble et renforcer ainsi la cohérence me laisse une opinion mitigée en revanche. La séparation entre les deux n'est-elle pas en partie nécessaire ? Il me paraît complexe de chercher à tout cumulé, de plus une telle connexion et imbrication ne rend-elle pas l'ensemble ultra-dépendant de lui-même ? De plus comme à l'heure où l'article était écrit leur approche n'avait pas encore été testée, comment être sûrs des bonnes performances de celle-ci ? Il est important de prendre en compte l'utilisateur mais les performances techniques d'un système, surtout pour de l'exploration/extraction de données sont plus qu'essentielles. Néanmoins il est important de tenir compte de la phrase (dans la partie discussion de l'article) « our approach must be a model that guides the developer to build up the system by respecting the interactivity and the iterativity qualities ». Aussi mieux vaut-il voir cette approche comme un ensemble de guidelines. Le problème qui reste toujours posé quand on dépend de l'utilisateur, ici nous nous « attaquons » au modèle en U, est de lui demander de penser à l'ensemble de ses besoins. D'expérience il est rare que ceux-ci soient déterminés et définitifs dès le début, surtout dans ce genre d'approche où l'interactivité et l'échange doivent rebondir sur le changement. Cela impliquerait donc une intégration continue de l'utilisateur. D'un autre côté comme celui-ci sera l'utilisateur final de l'application, sa présence est justifiée, voire même encouragée puisque cela lui permet une meilleure assimilation du système développé.

Dans cet article l'exemple est basé dans le domaine du médical sur la prévision des infections nosocomiales chez un patient. Chacun a déjà entendu parler au moins une fois de ce type d'infection que ce soit par un proche ou lui-même. C'est une inquiétude répandue dans le cadre d'une hospitalisation notamment chez les personnes à risques. L'idée de pouvoir mieux appréhender de telles infections est un véritable gain humain (et disons ce qui est financier puisque cela entraîne un surcoût (traitement, hospitalisation allongée, diagnostic etc). Il faut malheureusement être réaliste sur le point que ce genre de solutions sera sans doute plus soutenu sur ce qu'elle peut apporter comme gain financier et non humain.

Le fait d'avoir pensé leur approche au contact de médecins très impliqués dans le processus ne les a peut-être-t-il pas mené à une méthodologie trop spécifique qui la rendrait peut-être plus complexe à utiliser sur un autre public d'expert ?

Il apparaît évident que l'utilisation d'une telle approche peut être étendue au domaine médical en général pour, bien entendu, diagnostiquer les maladies. En dehors du médical ce genre de système est souvent utilisé pour tout ce qui est trafic routier, ferroviaire, aérien. L'implication de l'utilisateur dans un tel système présente à la fois un intérêt et un danger. Le danger d'une implication plus importante de l'utilisateur est une erreur d'expertise. De l'autre l'humain peut parfois connaître des éléments non informatisés que la machine ne peut percevoir. Parfois un ensemble de contraintes trop fermé fait exploser en vol les solveurs là où l'humain peut choisir de classer les contraintes. A tort ou à raison.

Pour en revenir au côté plus informatique de l'approche, UP se basant sur l'UML (Unified Modeling Language) sa réutilisabilité avec d'autres processus de développement est très aisée. Autrement dit si nous relevons des éléments pertinents dans un autre processus son intégration au modèle UP ne présente pas forcément une contrainte gênante.

Le modèle UP présente de réelles capacités d'adaptation qu'il est important de souligner ainsi que ses capacités d'interaction. Renforcé par le modèle en U qui apporte au designer toute l'importance et la prise de conscience des facteurs humains, la combinaison est optimale côté utilisateur.

Egalement comme cette approche est plus que basée sur les besoins du decision-maker celui-ci bénéficie d'une expérience utilisateur maximale, et est donc parfaitement en phase avec son DSS ce qui augmente les performances puisque nous avons montré plus haut qu'un DSS pâtissant d'une mauvaise interaction avec son utilisateur, potentiellement, s'auto-condamnait.

Le fait, pour moi, de simplifier la tâche, d'autant plus dans un domaine aussi lourd de conséquences que le médical, des decision-maker est un excellent point pour cette approche.

Synthèse

Le but des auteurs à court terme est d'achever leur DSS pour leur cas d'exemple de l'article. A long terme, d'étendre leur approche à d'autres méthodes d'extraction et de traitement des données, toujours dans une visée d'accessibilité pour l'utilisateur. Ils désirent également proposer une méthodologie spécifique sur les « DM-based DSS ». Nous pouvons supposer qu'ils visent également à pouvoir proposer leur approche à d'autres domaines comme nous avons pu en évoquer quelques uns plus haut.

Cet article nous a donc permis d'étudier des éléments qui concernent l'aide à la décision, un domaine fort intéressant de mon opinion personnelle de par ses impacts dans la société. L'approche proposée vise à être réutilisée dans différents domaines, avec différents experts.