



**HAL**  
open science

## Une démarche structurée pour la conception et l'évaluation d'applications Web par l'exploitation synergique des modèles de tâche et de navigation

Marco Winckler, Christelle Farenc, Philippe Palanque

### ► To cite this version:

Marco Winckler, Christelle Farenc, Philippe Palanque. Une démarche structurée pour la conception et l'évaluation d'applications Web par l'exploitation synergique des modèles de tâche et de navigation. 14ème Conférence Francophone sur l'Interaction Homme-Machine (IHM 2002), Nov 2002, Poitiers, France. pp.153-160, 10.1145/777005.777026 . hal-03667070

**HAL Id: hal-03667070**

**<https://hal.science/hal-03667070>**

Submitted on 13 May 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Une démarche structurée pour la conception et l'évaluation d'applications Web par l'exploitation synergique des modèles de tâche et de navigation

*Marco Winckler, Christelle Farenc, Philippe Palanque*

LIHS – IRIT Université Toulouse 3, 31062 Toulouse Cedex 4, France  
{winckler, farenc, [palanque](mailto:palanque@irit.fr)}@irit.fr

## RESUME

Alors même que le nombre d'applications web est en courante augmentation, le processus de conception de ces applications reste encore basé sur des techniques et méthodes informelles. Actuellement, peu de techniques de description sont disponibles pour modéliser les applications web et il est inhabituel de les modéliser alors que cette démarche est reconnue comme fondamentale pour la conception des systèmes interactifs. Dans cet article, nous présenterons deux notations pour modéliser la navigation dans les applications web et nous nous attacherons à démontrer comment ces modélisations sont nécessaires dans le processus de conception. Ensuite, nous décrirons comment utiliser les modèles de tâches pour évaluer le modèle de la navigation. L'objectif de cette évaluation est de s'assurer avant implémentation que les principales tâches de l'utilisateur peuvent être réalisées avec l'application web.

**MOTS CLES :** Conception d'applications web, modélisation de la navigation et des tâches, utilisabilité, évaluation de l'utilisabilité.

## ABSTRACT

Even though the importance of Web applications is still increasing, design practice in this area is still mainly based on informal techniques and methods. In practice, few description techniques are available to support web modeling and it is unusual to model web applications even though modeling is usually the corner stone of "classical" computer applications. Models not only formalize requirements but also they can help to assess the design in all the stages of the development process. In this paper we present notations dedicated to web navigation modeling and how they can affect the design process. In addition we discuss how task models could be used to assess navigation models according to user's activity. Our aim with this kind of evaluation is to ensure (prior to implementation) that important users tasks can (or cannot) be performed with the system.

**KEYWORDS :** Web design, notations, navigation modeling, task modeling, usability evaluation.

## INTRODUCTION

La grande majorité des applications Web ne prennent que très peu en compte l'utilisabilité et les besoins utilisateurs, et ce manque d'utilisabilité s'explique essentiellement par un processus de conception mal conduit. L'utilisation de certaines techniques de conception centrées utilisateur ont considérablement amélioré l'utilisabilité des sites web. Ainsi, le tri par cartes [19] et les questionnaires [6] sont des techniques courantes utilisées pour détecter les besoins utilisateurs et structurer les informations suivant le point de vue de l'utilisateur. Même si ces techniques sont simples, elles sont aujourd'hui reconnues comme permettant d'améliorer sensiblement l'utilisabilité des applications Web. Cependant, la conception des sites web reste une activité « expérimentale » et non formelle. Par exemple, il existe très peu de notation pour décrire la navigation de manière complète et non ambiguë, alors même que la navigation est considérée comme un élément fondamental et critique dans les applications web. Dans la pratique, les concepteurs utilisent des outils pour l'analyse des besoins comme par exemple les story-board et les sitemaps [13], mais aucun outil ne supporte la mise en application de ces besoins dans la conception. Ce manque de support n'est guère préjudiciable dans le cas d'applications web de « petite » taille pour lesquelles les concepteurs parviennent à prendre en compte intuitivement les quelques besoins utilisateurs, mais cette approche s'avère impossible dans le cas d'applications web de taille importante [7].

Les plus importantes difficultés qu'engendrent les processus de conceptions informels sont :

- détecter et résoudre les ambiguïtés dans les différentes descriptions ;
- fournir une démarche structurée et fiable pour transformer le produit de la conception en implémentation;
- gérer la complexité des modèles qui augmente exponentiellement en fonction de la taille de l'application web;
- maintenir et gérer les modifications.

Actuellement, peu de techniques de description comme les StateWebCharts (SWC) [22] permettent de modéliser

la navigation dans les applications web. La modélisation de la navigation peut être considérée par certains concepteurs comme une tâche non nécessaire, mais modéliser explicitement la navigation dans les applications web est aussi indispensable que modéliser le dialogue dans les systèmes interactifs. Cette modélisation permet :

- de concevoir la navigation à un niveau élevé d'abstraction sans gérer prématurément les détails;
- de séparer la conception de l'implémentation;
- de raisonner sur les modèles pour déterminer les propriétés du futur système, par exemple s'il est possible d'atteindre une page en un clic, ou s'il est toujours possible de revenir à la page principale en moins de trois clics, etc.;
- de fournir aux développeurs une description complète et non ambiguë des choix de conception ;
- de comparer les modèles de navigation avec les autres modèles définis dans le processus de conception comme les modèles de tâches ou les scénarii.

Les modèles de tâches sont non seulement nécessaires pour décrire comment les utilisateurs réalisent leurs tâches mais également quand et pourquoi les tâches sont réalisées. Alors même que la modélisation des tâches a été reconnue comme une étape nécessaire dans le développement des systèmes interactifs, peu d'études traitent de la modélisation de la tâche dans le cadre du développement d'applications web. Tauscher et Greenberg [23] décrivent quelques patterns de navigation, par exemple comment les utilisateurs revisitent les pages d'un site Web, mais les résultats présentés n'expliquent pas quelles tâches sont conduites quand ces patterns sont utilisés. Byrne [3] ont défini une taxonomie des tâches utilisateurs dans le cadre du web basée sur l'analyse des tâches les plus fréquemment rencontrées. Ces études tentent de décrire les tâches à un haut niveau d'abstraction et de généralité. Cependant aucune information n'est donnée ni sur la manière dont les tâches modélisées sont réalisées par l'utilisateur sur un site web spécifique ni sur la manière d'intégrer et d'utiliser les modèles de tâches dans le processus de conception d'une application web.

Nous pensons que la modélisation de la navigation et des tâches utilisateurs peuvent aider les concepteurs à construire des applications web fiables et utilisables. Dans ce papier nous allons décrire comment la notation SWC utilisée pour la modélisation de la navigation peut également être utilisée en correspondance avec les modèles de tâches de manière à identifier des problèmes d'utilisabilité à différentes étapes du processus de développement des applications web. L'objectif principal est de coupler les deux modèles et d'utiliser le modèle de tâche pour évaluer le modèle de navigation en vue de déterminer si le modèle de la navigation supporte effectivement la réalisation des tâches utilisateur.

## Le processus de développement des applications web

Lors de travaux précédents (Scapin et al [18]), nous avons expliqué que le processus de développement d'applications web est un processus itératif où l'étape traditionnelle de recueil des besoins est suivie par la spécification du site (cf. figure 1).

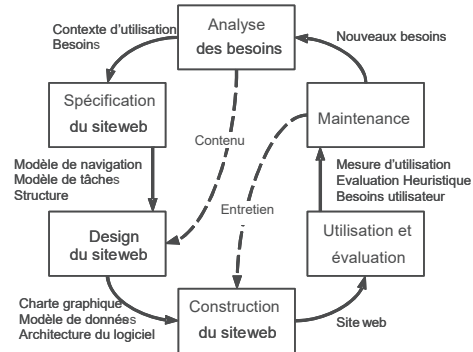


Figure 1 : Cycle de vie en O d'un site web.

Cette étape qui produit une description compréhensible du site (à travers l'expression des besoins, les modèles de tâche et de navigation) aide le développement du site en formalisant les besoins utilisateurs, guide la conception et le développement, et fournit des informations utiles durant les étapes d'évaluation et de maintenance du site. Pour évaluer les applications web une ou plusieurs méthodes d'évaluation peuvent être employées. Une partie de ces méthodes nécessite un prototype du système ou/et un feedback des utilisateurs. De même que pour les systèmes interactifs « classiques », les problèmes d'utilisabilité détectés durant la phase d'évaluation, i.e. à la fin du processus de développement des sites web peuvent provoquer des changements lourds et importants dans les applications web. Ceci justifie la nécessité d'utiliser des méthodes d'évaluation le plus tôt possible dans le processus de développement.

Dans le paragraphe suivant, nous présentons les caractéristiques principales de la notation SWC utilisée pour la description complète et non ambiguë de la navigation dans les applications web. Ensuite, la modélisation de la tâche des applications web est présentée, et avant d'exploiter ces deux modèles de manière à améliorer l'utilisabilité des applications web, nous présenterons l'utilisation de ces deux notations sur un exemple simple.

## LA CONCEPTION DE LA NAVIGATION

Dans le cadre du web, l'activité de navigation est souvent proche de l'activité de recherche d'information et elle représente une grande partie de l'activité de l'utilisateur. Pour les utilisateurs, une mauvaise conception de l'application web se traduit par des difficultés pour trouver les informations et par des difficultés de navigation [14]. En concevant la navigation, il est nécessaire de prendre en compte la représentation mentale que se fait l'utilisateur de l'application web pour garantir une navigation qui supporte les principales tâches des utilisateurs. Les liens rompus, les pages

absentes sont des problèmes fréquemment rencontrés et ils sont symptomatiques de la difficulté de maintenir une navigation efficace dans des applications web de taille importante. Pour traiter ces problèmes, des modèles de navigation ont été proposés comme OOHDM [20], WebML [5], une extension basée sur les réseaux de Petri [21] et les statechart [24]. L'utilisation de ces modèles de navigation permet de gérer la complexité et de définir des stratégies de navigation efficaces. Cependant aucune des techniques de description citées ci-dessus ne permet de représenter tous les aspects suivants :

- La représentation des activités système et utilisateur dans le modèle du dialogue
- Les frontières de la conception, i.e. distinguer clairement quelles parties de l'application web sont à la charge du concepteur et quelles parties ne le sont pas (comme par exemple les liens vers des sites extérieurs) ;
- La conception des tâches Client-Serveur, i.e. quelle partie de l'interface est du côté Client et quelle partie est du côté Serveur ;
- La prise en compte de l'accès direct car des parties de l'interface peuvent être directement accessibles par les utilisateurs, sans que ceux-ci n'aient suivis un chemin d'accès prédéfini.

Dans cet article, nous utilisons la notation SWC pour modéliser la navigation car cette notation permet de couvrir les aspects que nous venons de citer. SWC est une extension du modèle traditionnel statechart [9], basé sur les états et les événements. Chaque page est considérée comme un container d'objet et chaque container est associé à un état. Les liens et les objets interactifs provoquant des transitions sont représentés par des événements. La sémantique d'un état est : quand l'état est courant son contenu est visible par les utilisateurs tandis que les autres états sont cachés.

Les différents éléments des SWC sont représentés sur la figure 2. Les *états de base* (a) représentent des états contenant du code HTML. Les *états serveurs* (b) représentent des informations traitées côté serveur et par conséquent non visibles par les utilisateurs. Les *états dynamiques* (c) représentent des états pour lesquels le contenu est dynamiquement généré par le système. Les *liens externes* (d) font référence à des états hors de l'application traitée. Deux états d'agrégation sont également présents dans la notation, *XOR* (e) et *AND* (f). Dans les états XOR seul un sous-état est visible à un instant donné. Pour les états AND, pour chaque zone séparée par une ligne pointillée, un état peut être visible. Les états XOR sont utilisés pour représenter une hiérarchie d'états alors que les états AND sont utilisés pour représenter différentes visualisations. Sur la figure 2, ces différents états possèdent des identificateurs S1, S2, S3 ou S4 et C1, C2, C3, C4 correspondent au contenu des états. De plus, pour compléter les comportements qui existent dans les applications web, SWC intègre des *faux états*, qui correspondent à des états

qui n'ont pas de container associé. Parmi ces faux états, l'état (g) symbolise les *états initiaux*, l'état (h) représente une « sortie » du système, et les états (i) et (j) symbolisent des historiques respectivement en superficie et en profondeur. Ces quatre *faux états* sont également présents dans la notation statecharts. De plus, les StateWebCharts permettent de représenter à la fois l'activité de l'utilisateur sur l'interface (*transitions utilisateur* (k)) mais également l'activité du système (*transition système ou de finalisation*(l)).

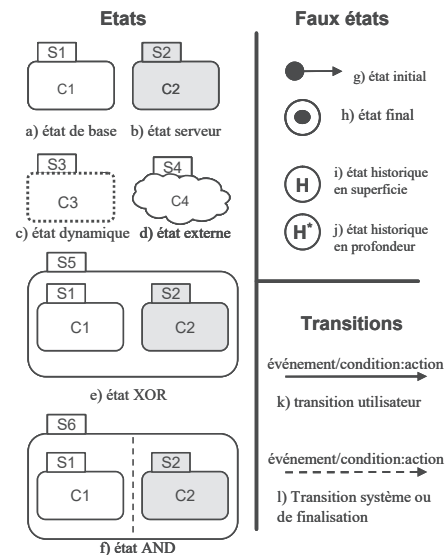


Figure 2 : Représentation graphique des StateWebCharts.

## MODELISATION DE LA TACHE POUR LE WEB

Les différentes études sur les systèmes interactifs classiques ont montrées le rôle prépondérant de l'analyse de la tâche pour la conception de systèmes utiles et utilisables [1, 2, 10]. Cependant, même si la modélisation de la tâche est une activité reconnue comme nécessaire, actuellement l'utilisation de modèles de tâches dans la conception d'applications web est une question encore ouverte. En réalité, les approches traditionnelles pour concevoir des applications web n'indiquent pas comment intégrer les modèles de tâches dans le processus de conception. Le point de vue de l'utilisateur est introduit de manière informelle (parfois simplement à l'aide d'un guide de recommandations [11]), généralement de manière implicite à travers des tests utilisateurs et des interviews. De manière à prendre en compte le point de vue utilisateur de manière plus formelle, nous proposons ci-dessous un processus de conception qui intègre explicitement la modélisation de la tâche dès les premières étapes du cycle de développement.

### Modélisation générique de la tâche pour le web

Ce paragraphe présente différents résultats d'analyses de tâches pour les applications web. Les tâches utilisateurs peuvent être rangées dans deux catégories : les tâches de haut niveau et les tâches primitives.

- Les tâches de haut niveau sont proches des objectifs de l'utilisateur (exemple : réserver un billet d'avion) et peuvent être réalisées de différentes manières. Ce type de tâche permet aux concepteurs de

comprendre les aspects fondamentaux de l'activité de l'utilisateur, elles sont généralement indépendantes du système avec lequel elles vont être réalisées ;

- Les tâches primitives sont plus détaillées et correspondent à une activité de l'utilisateur avec le système. Ces tâches sont génériques pour la majeure partie des applications web comme par exemple la tâche réalisée par l'utilisateur lorsqu'il navigue. Nous utilisons la taxonomie proposée par Byrne [3] comme ensemble de départ de tâches primitives.

Byrne a étudié les tâches des utilisateurs dans les applications web et a créé une taxonomie pour classifier les tâches les plus fréquentes. Cette classification présente les catégories suivantes :

- *L'utilisation d'informations* : décrit comment l'utilisateur traite l'information trouvée ;
- *Se situer dans une page* : décrit les stratégies utilisateur pour identifier les morceaux d'informations sur les pages web, par exemple trouver une image ;
- *Aller sur une page* : décrit toutes les activités qui provoquent l'affichage d'une page particulière, par exemple cliquer sur un lien hypertexte, fournir une adresse URL, utiliser un favori, etc.
- *Fournir de l'information* : décrit les activités de l'utilisateur quand il donne des informations, par exemple remplir un formulaire.
- *Configurer le navigateur* : décrit les tâches pour configurer le navigateur par exemple ajouter un favori, changer la taille du cache, etc.
- *Réagir à l'environnement* : décrit les tâches pour lesquelles le système requiert une action de l'utilisateur, par exemple répondre dans un dialogue, répondre au changement d'affichage, etc.

Les catégories *utilisation d'informations*, *configurer le navigateur* et *réagir à l'environnement* décrivent des activités entièrement indépendantes de la conception du site web. Donc, dans l'ensemble des tâches définies par Byrne, nous utilisons uniquement les tâches primitives concernant la navigation à savoir : *localisation dans une page*, *aller à une page* et *fournir de l'information*.

### La notation ConcurTaskTree

Pour décrire les tâches utilisateurs nous employons la notation ConcurTaskTree (CTT) [16]. Même si cette notation a été définie pour décrire les tâches des utilisateurs dans des systèmes interactifs classiques, elle reste utilisable dans le cadre d'applications web [17].

Dans ce paragraphe, nous allons utiliser CTT pour décrire les tâches primitives *localisation dans une page*, *aller à une page* et *fournir de l'information*. La notation ConcurTaskTree propose de représenter graphiquement les tâches suivant le principe de la planification hiérarchique. CTT permet de représenter les activités interactives et plus précisément i) la concurrence des tâches, qui est très présente dans les applications web où

plusieurs fenêtres/applications peuvent être ouvertes en même temps et ii) les interruptions possibles dans chaque activité de navigation. Les figures 3 et 4 présentent les tâches primitives décrites avec la notation CTT. L'un des objectifs de cette modélisation est de réutiliser ces tâches primitives pour décrire les tâches utilisateurs plus complexes. Par exemple, dans la figure 3, la tâche primitive *Aller sur une page* est réutilisée pour représenter la tâche primitive *Fournir de l'information*.

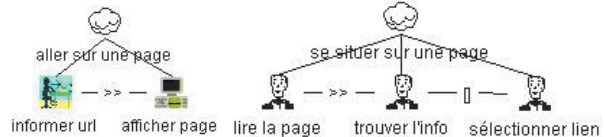


Figure 3 : Les tâches primitives : *Aller sur une page* et *se situer sur une page*

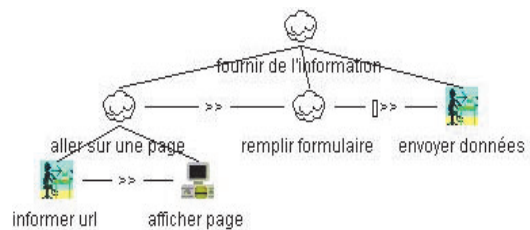


Figure 4 : La tâche primitive : *Fournir de l'information*

CTT intègre quatre types de tâches : *abstraite*, *utilisateur*, *système* et *interactive*. Les tâches abstraites correspondent à des tâches qui se re-décomposent en sous-tâches, par exemple *Aller sur une page* et *se situer sur une page* sont des tâches abstraites (cf. figure 3). Dans la notation CTT, la *tâche utilisateur* représente les tâches manuelles ou cognitives réalisées par l'utilisateur, par exemple *lire une page*, *trouver une information*, ou *sélectionner un lien* (cf. figure 3). Les tâches systèmes sont des tâches réalisées par le système, par exemple *afficher une page* (cf. figure 4). Les tâches interactives permettent de représenter les actions menées à la fois par l'utilisateur et le système, par exemple *informer url* et *envoyer des données*. CTT propose un ensemble d'opérateurs pour représenter les relations temporelles entre les tâches. T1||| T2 indique que les tâches sont réalisées en parallèle, T1 >> T2 signifie que T2 se réalise dès que T1 est terminée et T1 [ ] T2 représente le choix i.e. soit T1, soit T2 est réalisée pour atteindre le but. D'autres opérateurs ont été définis dans la notation CTT et peuvent être trouvés dans [16].

Un ensemble d'outils (CTT Environnement) ont été développés pour éditer, simuler et analyser les modèles de tâches représentés avec la notation CTT. Par exemple, CTTE permet de vérifier automatiquement la syntaxe des modèles de tâches, de comparer des modèles de tâches, de simuler le comportement.

### ETUDE DE CAS

Dans cette section nous présentons une étude de cas portant sur l'application web HIBAM (Hidrografia da Bacia Amazônica) qui est en cours de développement et dont l'objectif est de fournir des informations sur les dix



dernières années de recherche dans le domaine de l'hydrographie dans la région de l'Amazonie (Brésil). L'étude de cas vise à :

- Montrer comment la notation StateWebChart peut être utilisé pour décrire différentes stratégies de navigation ;
- Montrer comment CTT et les tâches primitives peuvent être utilisées pour représenter des tâches de haut niveau ;
- Montrer qu'il est possible de mettre en correspondance et d'exploiter ensemble ces deux notations pour vérifier que le modèle de navigation supporte un modèle de tâche donnée.

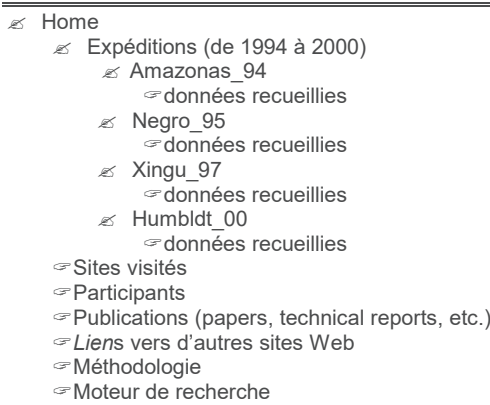


Figure 5 : Modèle hiérarchique pour le site web HIBAM.

Pour des raisons de place, nous ne présentons sur la Figure 5 qu'une version simplifiée de la hiérarchie du site web HIBAM. Cette structure est la même que celle du site réel mais les éléments répétitifs comme les différentes expéditions (plus de 20) ont été enlevés.

**MODELISATION DE LA NAVIGATION**

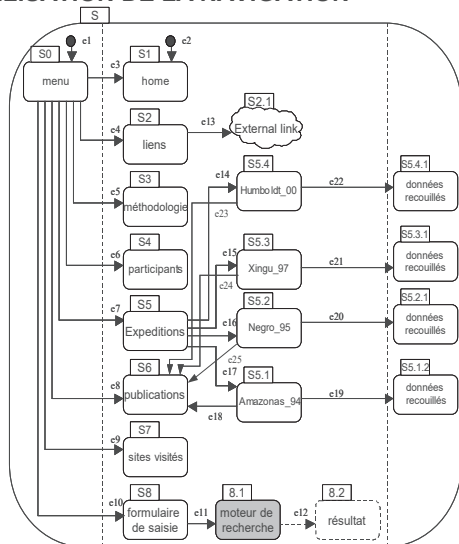
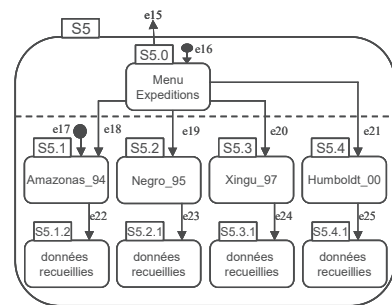


Figure 6 : Option de conception A pour la navigation.

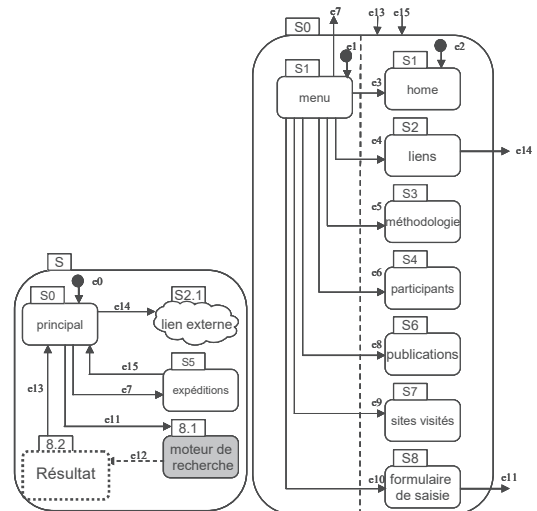
Suivant un processus de conception en parallèle, nous avons construit deux conceptions différentes (A et B) pour la navigation du site web HIBAM. La Figure 6 présente la première alternative de conception, une navigation basée sur deux frames avec un menu

permettant d'accéder aux informations. Dans cette alternative, Menu (état S0) est toujours visible. La configuration initiale est l'état S0 et S1, définis respectivement par e1 et e2. Quand une expédition particulière est sélectionnée, des informations supplémentaires sont présentées de manière concurrente dans une nouvelle fenêtre (cf. événements e19, e20, e21 et e22 respectivement pour les états S5.1.2, S5.2.1, S5.3.1, S5.4.1)

La deuxième alternative de conception (cf. figure 7 (a), (b) et (c)) consiste à cacher le menu principal quand les utilisateurs naviguent en profondeur dans la hiérarchie des pages.



7 (b) Main - Niveau 1



7 (a) HIBAM - Niveau 0

7 (c) Expéditions - Niveau 1

Figure 7 : Option de conception B pour la navigation

StateWebcharts permet de représenter graphiquement les hiérarchies complexes de manière simplifiée en découpant la navigation en plusieurs niveaux. Dans la figure 7.a au plus haut niveau (niveau 0) les utilisateurs commencent dans l'état S0 (menu principal) dont le contenu est détaillé au niveau 1 (figure 7.b). L'état S5 présentant le menu pour les expéditions est également détaillé au niveau 1 (figure 7.c). Quand l'utilisateur sélectionne expédition (déclenchement de l'événement e7), la nouvelle configuration courante contient deux zones visuelles (un menu pour toutes les expéditions et une zone pour une expédition particulière). Il est important de noter que dans l'état S5, seul le contenu relatif aux expéditions est visible.

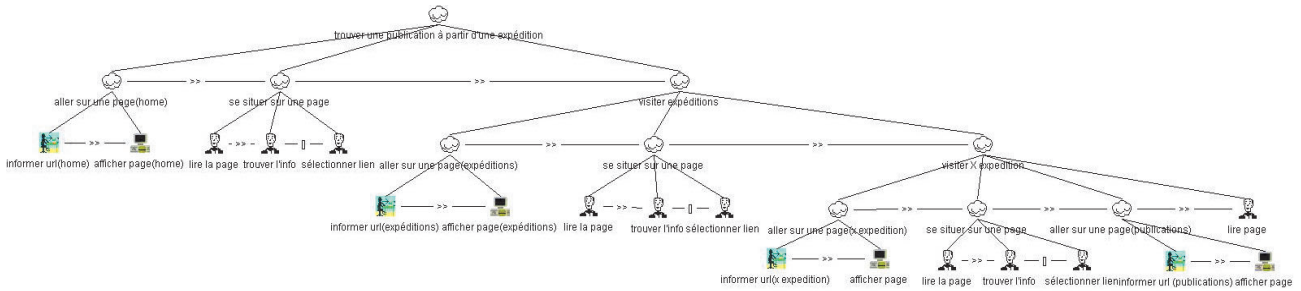


Figure 8. Modèle de tâche pour trouver une publication à partir d'une expédition

### MODELISATION DE LA TACHE

Pour cette étude de cas, nous avons sélectionné deux tâches utilisateur :

- Aller sur un site web externe, i.e. utiliser le site comme point de départ pour trouver d'autres sites;
- Trouver les publications d'une expédition, i.e. récupérer les publications concernant les résultats d'une expédition.

Ces tâches sont décrites avec CTT (cf. figure 8 et 9). Sur la figure 9, dans le modèle de tâche, l'utilisateur passe par la page principale pour trouver un accès vers des sites externes. L'utilisateur doit d'abord visiter la homepage (*aller sur une page (home)*), trouver l'information dans la page (sélectionner un lien externe) et ensuite aller sur une page suivant le lien sélectionné. Le modèle de tâche de la figure 8 est plus complexe, l'utilisateur doit voir toutes les publications (rapports, articles, etc.) qui ont été produites à la suite d'une expédition particulière en sélectionnant une expédition particulière dans la liste des expéditions et puis en visitant les pages concernant cette expédition. Ensuite, l'utilisateur trouve les publications associées à l'expédition.

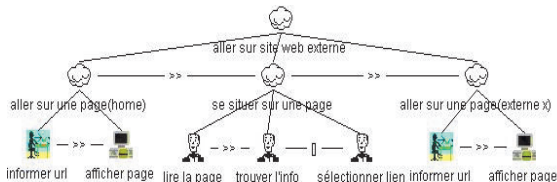


Figure 9 : Modèle de tâche pour aller sur un site web externe.

### VERIFICATION DE LA NAVIGATION EN UTILISANT LES MODELES DE TACHES

Nous avons vu comment décrire la navigation et les tâches utilisateurs avec les notations StateWebChart et CTT. Ces modèles sont particulièrement utiles dans la mesure où ils fournissent aux concepteurs une représentation explicite de la tâche des utilisateurs et de la navigation dans l'application web. De plus, la mise en correspondance de ces deux modélisations peut apporter d'autres intérêts. En effet, la mise en correspondance de ces modèles permet de déterminer si les utilisateurs pourront ou non réaliser une tâche spécifique avec une application web donnée. En utilisant le simulateur CTTE il est possible d'exécuter un modèle de tâche. Le résultat de cette exécution est un scénario (ensemble d'actions réalisées par l'utilisateur). Comme CTT permet la représentation des choix et de la concurrence dans la tâche de l'utilisateur, différents scénarii peuvent résulter

de l'exécution d'un modèle de tâche. La vérification consiste à exécuter un scénario sur le modèle de navigation. Action après action, le scénario est joué sur le modèle de navigation et à chaque action, l'exécution de l'action est mise en correspondance sur le modèle de navigation.

---

**informer url(home)**  
Afficher page(home)  
lire page  
se situer sur une page  
choisir un lien

**informer url(expéditions)**  
Afficher page  
lire page  
se situer sur une page  
choisir un lien

**informer url(x expédition)**  
Afficher page  
lire page  
se situer sur une page  
choisir un lien

**informer url(publications)**  
afficher page  
lire page  
se situer sur une page

---

Figure 10 : Scénario générique extrait de la tâche trouver les publications à partir d'une expédition.

Pour vérifier la mise en correspondance, les étapes suivantes doivent être réalisées :

- Extraire les scénarii du modèle de tâche
- Enlever les actions non interactives des scénarii. La figure 10 présente le scénario extrait de la tâche trouver une publication à partir d'une expédition. Tous les éléments en gras de la figure 10 sont des tâches interactives du scénario. Nous pouvons observer qu'il y a uniquement quatre tâches interactives : *informer url – (home)*, *informer url – (expédition)*, *informer url – (x expédition)*, *informer url – (publications)*. Les autres tâches sont réalisées par le système comme *afficher une page* ou bien elles sont réalisées par l'utilisateur seul comme *choisir un lien* ou *lire une page*. Ces tâches non interactives sont toutes supprimées.
- Lier le scénario et le modèle de navigation. Chaque tâche interactive du scénario est associée à un état (ou différents états s'il y a une concurrence d'états) dans le modèle de navigation. Par exemple, la tâche de haut niveau *informer url (home)* (figure 9) est associée aux états *S0* et *S1* (figure 6). Le résultat de cette étape est un ensemble de scénarii concrets présentés sur la figure 11.
- Vérifier la consistance entre les tâches et le modèle de navigation. Cette validation se fait en exécutant tous les scénarii concrets (figure 11) sur le modèle

de navigation (figure 6). Suivant la structure du modèle de navigation, un scénario peut ou non être exécuté. Par exemple, pour que le scénario de la figure 11 puisse être exécuté, il faut qu'il existe une transition dans le modèle de navigation de la figure 6 entre l'état S5 et S5.1.

Tâches interactives	Etat cible dans le modèle de navigation
Inform url(home):	S0, S1
informer url(expéditions):	S5
informer url(x expédition):	S5.1
Inform url(publications):	S6

Figure 11 : Scénario concret associés au modèle de navigation

## LA VERIFICATION ET LE PROCESSUS DE DÉVELOPPEMENT

Mettre en correspondance les modèles de tâches et les modèles de navigation à différentes étapes du cycle de développement d'une application web présente de nombreux avantages.

**Premières phases :** Ces phases correspondent aux étapes 1, 2 et 3 dans la figure 1. Nous proposons d'utiliser les deux notations (StateWebCharts et ConcurTaskTree) pour décrire et documenter la conception. Avant l'implémentation, l'évaluation à partir des modèles de tâches peut être utilisée pour vérifier si toutes les tâches identifiées peuvent être réalisées avec le modèle de navigation de l'application web.

**Evaluation de l'utilisabilité :** Cette phase correspond à l'étape 5 dans la figure 1. L'évaluation à partir des modèles de tâches peut être utilisée durant l'étape d'évaluation de l'utilisabilité en complément des méthodes d'évaluation traditionnelles. Dans ce cas, les évaluateurs peuvent jouer à la fois les scénarii sur le modèle de la navigation et sur l'implémentation de l'application web. Ceci permet au concepteur de voir les aspects visuels de l'interface qui ne pouvaient être vu lors de l'exécution des scénarii sur le seul modèle de la navigation. Un autre avantage est qu'il est possible d'évaluer dynamiquement la navigation générée. En effet, la notation StateWebChart intègre un type d'état spécial (*état dynamique*) qui représente le fait qu'un contenu dynamique est ajouté au site mais qui ne permet pas de représenter ce contenu lui-même. Enfin, durant l'étape d'évaluation de l'utilisabilité, il est possible de donner le scénario aux utilisateurs et d'analyser comment ils le réalisent avec l'application web. Paterno et Ballard [17] ont déjà proposé ce type d'évaluation à distance.

**Reconception :** Cette phase correspond à l'étape 6 sur la figure 1. Nous pensons que l'approche proposée est particulièrement avantageuse durant cette étape qui est une étape critique du cycle de développement des applications web. En effet, les concepteurs doivent garantir que les modifications réalisées sur l'application permettent de supporter les nouvelles tâches utilisateurs détectées mais également toutes les anciennes tâches identifiées lors de la conception.

Durant les premières phases du cycle de développement et durant la reconception, l'évaluation à partir des modèles de tâches est une évaluation prédictive, i.e. l'évaluation se fait sur le modèle de la navigation et non sur l'application implémentée, ce qui permet d'éviter des lourdes modifications d'implémentations.

## CONCLUSION

Nous avons expliqué en quoi la modélisation des tâches et de la navigation améliore la conception d'une application web. Nous avons également présenté comment ces deux modèles peuvent être mis en correspondance et exploités à différentes étapes du cycle de développement pour évaluer la navigation dans une application web. Pour les systèmes interactifs, des approches basées sur les modèles ont été déjà proposées, mais pour les applications web, seules quelques études commencent à émerger [8]. Récemment, des modèles de description des interfaces web ont été proposés [5, 20, 22] mais ces modèles concernent uniquement la modélisation du système. La conception centrée utilisateur qui nécessite d'intégrer le point de vue de l'utilisateur dans le processus de développement reste une pratique informelle, même si des outils existent pour supporter cette approche [6, 10]. Les études concernant les tâches utilisateur dans les applications web explorent plus l'activité de l'utilisateur que la modélisation de ses tâches et ces études apportent peu de solutions pour aider la conception [3, 13]. Dans la seconde partie de cet article, nous avons présenté une méthode pour évaluer la navigation à partir des modèles de tâche de l'utilisateur. Cette méthode est proche de précédents travaux concernant l'évaluation des systèmes interactifs à partir des scénarii [12] et des travaux de Paterno et Ballard [17] pour la re-ingénierie des applications web. Le principe de vérification est le même : mettre en correspondance les scénarii produits à partir des modèles de tâches avec l'interface. Dans [17], les auteurs proposent d'appliquer cette vérification dans le cas d'une évaluation de l'utilisabilité à distance, et pour cela, ils mettent en correspondance les « log files » avec les modèles de tâches. Cependant cette évaluation concerne uniquement les applications web développées alors que dans notre approche, l'évaluation est réalisée sur le modèle de navigation. Même si leur approche est efficace pour identifier les problèmes d'utilisabilité sur une application web implémentée, elle ne peut être appliquée plus tôt dans le cycle de développement de l'application web.

Navarre et al [12] ont développé un outil pour exécuter les scénarii CTT sur des applications implémentées avec le formalisme des ICO [15]. Cette approche est conceptuellement proche de la notre mais nous pensons qu'elle n'est pas applicable aux applications web. Le formalisme des ICO représente les états de manière implicite, ainsi l'état courant d'une application est représenté par une valeur et par la distribution de jetons dans les modèles. Les états sont centraux dans les applications web et ils sont le cœur de la notation. C'est



une des raisons du choix des StateWebCharts comme notation pour modéliser la navigation dans les applications web. Les applications web sont des applications qui traitent et manipulent principalement des informations. C'est pour cette raison que la notation ICO, adéquate pour modéliser les applications temps réel critiques, ne peut être utilisée pour modéliser les applications web où elle risque de produire des modèles complexes qui rendrait difficile leur utilisation par le concepteur. Enfin, cet article décrit l'utilisation de la technique d'évaluation à différentes étapes du cycle de développement itératif alors que les travaux connexes [12] et [17] ne s'appliquent que sur des étapes spécifiques.

De manière à supporter le processus de développement présenté, un outil est en cours de développement. La première étape consiste à produire un éditeur pour la notation StateWebChart qui s'intègre dans des outils plus génériques supportant la notation UML.

#### REMERCIEMENTS

Nous tenons à remercier ici le CNPq Brésilien qui finance partiellement cette recherche.

#### BIBLIOGRAPHIE

1. Benyon, D. *Task Analysis and System Design: the Discipline of Data*. Interacting with Computers, 4(2), 246-59.
2. Bodart, F. et alli. A Model-Based Approach to Presentation: A Continuum from task Analysis to Prototype. *Proc. of DSVIS'94*, (Bocca di Magra), 8-10 juin 1994.
3. Byrne, M.; John, B. E.; Wehrle, N. S.; Crow, D. C. The Tangled Web We Wove: A Taskonomy of WWW Use. *CHI99* (15-20 May, 1999). p.544- 551.
4. Card, S.; Robertson, G.; York, W. The WebBook and the Web Forager: An Information Workspace for the World-Wide Web. *CHI'96*, (April 13 - 18, 1996), Vancouver, Canada.
5. Ceri, S.; Fraternali, P.; Bongio, A. Language (WebML): a modeling language for designing Web sites. *9th WWW Conference*, Amsterdam, May 15-19, 2000.
6. Dong, J.; Martin, S. Interactive Usage of Customer Satisfaction Surveys to Assess an Evolving Web Site. *Proc. of the 6th Conference on Human Factors and the Web*, Austin, USA, June 19, 2000.
7. Fowler, S. L.; Novack, A. J.; Stillings, M. J. The Evolution of a Manufacturing Web Site. *9th WWW Conference*, Amsterdam, May 15-19, 2000.
8. Fraternali, P.; Paolini, P. Model-Driven Development of Web Applications: the Autoweb System. *ACM Transactions on Office Information Systems* vol. 18 (4), 2000.
9. Harel, D. Statecharts. *A visual formalism for complex systems*. Science of Computer Programming 8, P. 231-274. 1987.
10. Johnson, P. *Human-Computer Interaction: Psychology, Task Analysis and Software Engineering*, Mc-Graw Hill, Maidenhead, 1992.
11. Lynch, P. J; Horton, S. *Web Style Guide : Basic Design Principles for Creating Web Sites*. Yale Univ Press. 1999. p.164.
12. Navarre, D.; Palanque, P.; Bastide, R.; Paternò, F.; Santoro, C. A tool suite for integrating task and system models through scenarios in *8<sup>th</sup> DSV-IS'2001*, June 13-15, 2001, Glasgow, Scotland.
13. Newman, M. W.; Landay, J. A. Sitemaps, storyboards, and specifications: a sketch of Web site design practice. *Proc. of DIS'2000*, ACM Press New York, NY, USA. Pages: 263 – 274.
14. Nielsen, J. Failure of Corporate Websites. Alertbox, 98. <http://www.useit.com/alertbox/981018.html>
15. Palanque, P.; Bastide, R.; Paternò, F. Formal Specification as a Tool for Objective Assessment of Safety-Critical Interactive Systems. *Interact'97*, Sydney, July 1997, Chapman et Hall, pp.323-330.
16. Paterno, Mancinii, Meniconi ConcurTaskTrees: a Diagrammatic Notations for Specifying Task Models, *Interact'97*, pp 362-69, july 97, Sydney, Chapman&Hall.
17. Paterno, F.; Ballardini, G. Model-Aided Remote Usability Evaluation. *Interact'99*, pp.434-442, IOS Press, Edinburgh, September'99.
18. Scapin, D. et al. Transferring Knowledge of User Interfaces Guidelines to the Web. In. *TFWWG'2001*. London: Springer; pp. 293-304.
19. Scholtz, J.; Laskowski, S.; Downey, L. Developing Usability Tools and Techniques For Designing and Testing Web Sites. *4th Conf. on Human Factors & the Web*, June 5, 1998, Basking Ridge, USA (1998).
20. Schwabe, D. et al. *Engineering Web Applications for Reuse*, IEEE Multimedia, Spring 2001, pp. 2-12.
21. Stotts, D. P.; Furuta, R. *Petri-net-based hypertext: document structure with browsing semantics*. *ACM Trans. Inf. Syst.* 7, 1 (Jan. 1989), pp. 3-29.
22. Winckler, M.; Farenc, C.; Palanque, P.; Bastide, R. Designing Navigation for Web Interfaces. In *Proc. of IHM-HCI2001*, Lille, França, 10-14 Sep. 2001.
23. Tauscher, L. ; Greenberg, S. (1997). Revisitation patterns in World Wide Web navigation. *Proc. of CHI97* (pp. 399-406). New York: ACM Press.
24. Zheng, Y.; Pong, M. Using statecharts to model hypertext. In. *Proc. of the ACM Hypertext*. Milan, Italy. 1992.