POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Conceptualisation de l'interface d'un logiciel d'assistance à l'analyse ergonomique de tâche avec intégration de plusieurs méthodes
Title:

Auteur: Robert Morin
Author:

Date: 2014

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Morin, R. (2014). Conceptualisation de l'interface d'un logiciel d'assistance à l'analyse ergonomique de tâche avec intégration de plusieurs méthodes [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1526/
Citation:

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: https://publications.polymtl.ca/1526/
PolyPublie URL:

Directeurs de recherche: Jean-Marc Robert
Advisors:

Programme: Génie industriel
Program:

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

CONCEPTUALISATION DE L'INTERFACE D'UN LOGICIEL D'ASSISTANCE À L'ANALYSE ERGONOMIQUE DE TÂCHE AVEC INTÉGRATION DE PLUSIEURS MÉTHODES

ROBERT MORIN

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INDUSTRIEL)

AOÛT 2014

© Robert Morin, 2014.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CONCEPTUALISATION DE L'INTERFACE D'UN LOGICIEL D'ASSISTANCE À L'ANALYSE ERGONOMIQUE DE TÂCHE AVEC INTÉGRATION DE PLUSIEURS MÉTHODES

présenté par : MORIN Robert

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. DESMARAIS Michel, Ph.D., président

M. ROBERT Jean-Marc, Doctorat, membre et directeur de recherche

M. ROBILLARD Pierre-N., Ph.D., membre

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche, M. Jean-Marc Robert, pour sa patience et son soutien tout au long de ce qui fut un parcours un peu chaotique par moments, mais ayant finalement produit des résultats qui, je l'espère, auront justifié le tout.

Aussi et surtout, je veux remercier ma femme pour son amour et son appui indéfectibles durant toute l'aventure.

RÉSUMÉ

Il arrive souvent, dans le cours d'une analyse ergonomique de tâche, qu'il soit nécessaire d'utiliser plus d'une méthode d'analyse pour tirer un maximum d'enseignements des diverses sources d'informations concernant la tâche effectuée par un humain (p. ex. : documents, interviews, observations, prises de mesures, etc.). Malheureusement, dans le contexte actuel, les logiciels qui sont mis à la disposition des analystes ne supportent qu'une seule méthode d'analyse, ce qui force ces derniers à soit n'utiliser qu'une seule méthode soit, s'ils veulent vraiment utiliser plusieurs méthodes, à dupliquer certaines informations pour utiliser d'autres logiciels supportant chacun une autre méthode. Il y a donc tout lieu de croire qu'un logiciel supportant plusieurs méthodes d'analyse peut être utile. L'objectif de cette recherche est donc d'effectuer une proposition détaillée de ce que pourraient être les fonctionnalités et l'interface utilisateur d'un tel logiciel.

Après une revue de la littérature sur les méthodes d'analyse ergonomique de tâche, il a été décidé que le logiciel allait supporter un éventail de méthodes permettant l'analyse de tâches individuelles, incluant l'analyse hiérarchique de tâche (AHT), l'ordinogramme de traitement humain de l'information (OTHI) qui est une forme d'analyse procédurale, la méthode analytique de description de tâche (MAD), l'analyse temporelle, KLM-GOMS et CMN-GOMS. En cours de route, CMN-GOMS s'est révélé être une méthode d'analyse hybride, puisqu'elle est, en fait, une combinaison de deux méthodes : AHT et KLM-GOMS. La méthode CMN-GOMS ne sera donc supportée par @Esperanto que sous la forme d'une analyse multiméthode, soit l'une des nombreuses possibilités du logiciel.

Parce que les méthodes d'analyse de tâches susmentionnées doivent être supportées par un seul et même logiciel, elles ont été soigneusement étudiées pour déterminer leurs points communs et leurs différences. Constatant de nettes différences entre les méthodes sur le plan de leur expressivité, on a voulu, sans affecter leur essence, réduire ces différences pour faciliter les échanges d'informations entre elles. Les méthodes retenues ont donc été rangées dans l'une de deux grandes classes, selon qu'elles servent à analyser une tâche globalement ou à en préciser tous les détails, puis leur expressivité a été examinée et, si possible, augmentée pour obtenir dans chaque classe la plus large base commune possible.

Par la suite, tous les efforts ont porté sur la conceptualisation du logiciel et sur le prototypage très détaillé de plusieurs fenêtres de son interface utilisateur. Le résultat donne une bonne vue d'ensemble des principales fonctionnalités du logiciel et de l'organisation de son interface utilisateur et pourra servir, espère-t-on, comme outil de promotion pour la création d'un logiciel d'analyse ergonomique de tâche multiméthode.

ABSTRACT

Oftentimes, in the course of an ergonomic task analysis, it is necessary to use more than one task-analysis method in order to get the most out of the information gathered about a task that humans must carry out (e.g., documents, interviews, observations, measurement taking, etc.). Unfortunately, the current situation is that all widely available task-analysis software applications support only one task analysis method, thus forcing analysts to either use only one method or to duplicate some information in order to use many applications, each supporting a different method. Therefore, there is every reason to believe that an application supporting multiple task-analysis methods would be useful. The current research goal is to create a detailed proposal for the envisioned functionalities and user interface of such application.

After a literature review on ergonomic task analysis methods, it was decided that the proposed application would support an array of methods for the analysis of individual tasks, including Hierarchical Task Analysis (HTA), a form of procedural analysis, MAD (the French acronym for Analytical [Task] Description Method), a timeline analysis method, KLM-GOMS and CMN-GOMS. Along the way, CMN-GOMS was found to be a hybrid method, as it is a combination of two methods: HTA and KLM-GOMS. CMN-GOMS will therefore not be directly supported; rather, it will be supported as one of the many possible combinations for multi-method analysis.

As the aforementioned task-analysis methods must be supported by a single application, they have been carefully studied in order to highlight their commonalities and their differences. Because considerable differences in regard to expressiveness were exposed — while adamant in our desire to not change the essence of each method — we set out to minimize these differences. For this, we assigned every supported method to one of two classes, depending on whether it is meant to analyze the various components of a task or to delve into its minutiae. Then, the expressiveness of each was examined and, as feasible, extended in order to increase commonalities among methods of the same class.

Once this was done, all efforts have focused on conceptualizing the application's functionalities as well as prototyping most of its user interface. The resulting thesis provides a good overview of the application's planned functionalities and proposed user interface which, it is hoped, can serve as a promotion tool to create excitement leading to the creation of this multi-method ergonomic task analysis application.

TABLE DES MATIÈRES

REMERCIEMENTS III

RÉSUMÉ IV

ABSTRACT VI

TABLE DES MATIÈRES VII

LISTE DES TABLEAUX XI

LISTE DES FIGURES XII

LISTE DES SIGLES ET ABRÉVIATIONS XV

LISTE DES ANNEXES XVIII

INTRODUCTION 1

CHAPITRE 1 PRÉSENTATION DES MÉTHODES D'ANALYSE DE TÂCHE 8

1.1 Quelques points communs 8

1.2 Réseaux 11

1.3 Analyse hiérarchique de tâche (AHT) 12

1.4 Analyse procédurale / Ordinogramme de traitement humain de l'information (OTHI) 17

1.5 Méthode analytique de description de tâche (MAD) 21

1.6 Analyse temporelle (AT) 26

1.6.1 Représentation textuelle 27

1.6.2 Représentations graphiques 28

1.7 Goals, Operators, Methods, and Selection Rules (GOMS) 32

1.8 KLM-GOMS 34

1.9 CMN-GOMS 38

1.10 Synthèse 40

CHAPITRE 2 NORMALISATION DES MÉTHODES D'ANALYSE DE TÂCHE 41

2.1 Récapitulatif des différences entre les méthodes d'AT 41

2.2 Méthode native et méthode courante 41

2.3 À propos de la normalisation des méthodes d'analyse de tâche 43

2.4 Avant de continuer : un petit secret… Ce sont toutes des AHT! 45

2.5 Normalisation et notation des méthodes d'analyse supportées 47

2.6 Normalisation de l'AHT 48

2.6.1 Normalisation et notation des tâches en AHT 48

2.6.2 Normalisation et notation des plans en AHT 51

2.6.3 Représentations d'une AHT 56

2.7 Normalisation de l'analyse procédurale / OTHI 57

2.7.1 Représentations d'un OTHI 59

2.7.2 Sous-routines et paramètres 65

2.7.3 Création de procédures pour l'aide à la performance humaine 67

2.8 Normalisation de MAD 67

2.8.1 Représentations d'une analyse selon MAD 68

2.9 Normalisation de l'analyse temporelle 69

2.9.1 Identification des chemins d'exécution d'intérêt 71

2.10 Normalisation de KLM-GOMS 72

2.11 Normalisation de CMN-GOMS 72

2.12 Synthèse 73

CHAPITRE 3 CONCEPTUALISATION ET PROTOTYPAGE DE L'IU 74

3.1 Fonctionnalités d'utilité générale 74

3.1.1 Recherche de tâches à partir de la fenêtre principale 74

3.1.2 Navigation dans une analyse de tâche 75
3.2 AHT 78
3.2.1 Processus de construction d'une AHT 78
3.2.2 Modification d'une AHT à partir de la fenêtre principale 85
3.2.3 Éditeur de plan structuré 88
3.2.4 Éditeur graphique de plan 91
3.3 OTHI 91
3.3.1 Processus de construction d'un OTHI 91
3.3.2 Fenêtre principale 92
3.3.3 Modification d'une OTHI à partir de la fenêtre principale 94
3.3.4 Éditeur de plans AHT en représentation graphique 103
3.4 MAD 104
3.4.1 Processus d'élaboration d'une description de tâche selon MAD 104
3.4.2 Fenêtre principale 106
3.4.3 Fenêtre favorisant l'entrée de masse (ou éditeur de tâche) 106
3.4.4 Modification d'une description de tâche MAD de la fenêtre principale 111
3.4.5 Éditeur de plan structuré 112
3.5 Analyse temporelle 112
3.5.1 Processus de construction d'une analyse temporelle 112
3.5.2 Extraction d'un scénario d'une étude globale (AHT, OTHI ou MAD) 112
3.5.3 Création et modification d'une analyse temporelle 115
3.5.4 Visualisation d'une analyse temporelle 116
3.6 KLM-GOMS 117
3.6.1 Processus de construction d'une analyse KLM-GOMS 117

3.6.2 Création et modification d'une analyse KLM-GOMS 118

3.7 CMN-GOMS 119

3.7.1 Une méthode toute désignée pour @Esperanto 119

3.8 Analyse multiméthode 120

3.8.1 De la complexité de l'ensemble aux menus détails 120

3.8.2 Transitions possibles entre méthodes d'analyse 120

3.8.3 Changement de méthode d'analyse 122

3.8.4 Changement pour une autre méthode d'analyse globale 123

3.8.5 Changement pour une méthode d'analyse locale d'une analyse globale 125

3.8.6 Analyse locale mixte 125

3.8.7 Un cas particulier : AHT + KLM-GOMS == CMN-GOMS 127

3.9 Conversion entre AHT et OTHI 129

3.9.1 Conversion d'une AHT en OTHI 129

3.9.2 Conversion d'un OTHI en AHT 130

3.10 Synthèse 133

CONCLUSION 134

BIBLIOGRAPHIE 136

ANNEXES 138

LISTE DES TABLEAUX

Tableau 1-1 Énoncés disponibles pour les plans d'une AHT. 14

Tableau 1-2 Notation utilisée par TaskArchitect© pour les plans d'une AHT. 15

Tableau 1-3 Opérateurs KLM-GOMS couramment utilisés. 35

Tableau 1-4 Opérateurs KLM-GOMS ajoutés pour l'étude d'IHM sur téléphones mobiles. 35

Tableau 2-1 Caractéristiques des méthodes supportées par @Esperanto. 42

Tableau 2-2 Notation textuelle pour les tâches. 49

Tableau 2-3 Notation graphique pour les tâches. 49

Tableau 2-4 Formes identifiant un aspect important d'une tâche. 52

Tableau 2-5 Énoncés pour les plans d'une AHT (notation basée sur Stanton (2006)). 54

Tableau 2-6 Énoncés pour les plans d'une AHT (notation basée sur TaskArchitect©). 55

Tableau 2-7 Notation graphique des différents enchaînements de tâches. 60

Tableau 2-8 Notation graphique des énoncés inspirés de la programmation structurée. 61

Tableau 2-9 Les divers éléments de la définition d'une sous-routine. 66

Tableau 3-1 Étapes nécessaires pour définir l'AHT du chapitre précédent. 82

LISTE DES FIGURES

Figure 1-1 AHT partielle pour la tâche consistant à envoyer un courriel. 15

Figure 1-2 AHT de la figure 1 1 en représentation textuelle. 16

Figure 1-3 Symboles de base d'un ordinogramme. 18

Figure 1-4 Symboles supplémentaires représentant un aspect important d'une tâche. 18

Figure 1-5 OTHI correspondant approximativement à l'AHT partielle pour la tâche consistant à envoyer un courriel.....19

Figure 1-6 Représentation textuelle de l'OTHI précédent. 20

Figure 1-7 Description habituelle d'une tâche selon MAD. 23

Figure 1-8 Représentation possible de la tâche «Envoyer courriel» du précédent exemple. 25

Figure 1-9 Analyse temporelle d'une méthode réalisant l'envoi d'un courriel. 27

Figure 1-10 AT pour l'envoi d'un courriel (représentation de base). 28

Figure 1-11 AT pour l'envoi d'un courriel (avec cumulatifs). 29

Figure 1-12 AT pour l'envoi d'un courriel (représentation linéaire horizontale simple). 29

Figure 1-13 AT pour l'envoi d'un courriel (avec parallélisme). 30

Figure 1-14 AT pour l'envoi d'un courriel (avec indication relative de la CMT). 31

Figure 1-15 AT pour l'envoi d'un courriel (représentation verticale). 31

Figure 1-16 KLM-GOMS pour une méthode réalisant l'envoi d'un courriel. 37

Figure 1-17 CMN-GOMS pour l'envoi d'un courriel (avec méthodes en KLM-GOMS). 39

Figure 1-18 CMN-GOMS pour l'envoi d'un courriel (comme vu dans la littérature). 40

Figure 2-1 Représentations graphiques optionnelles pour les tâches d'une AHT. 50

Figure 2-2 AHT normalisée pour la tâche d'envoi d'un courriel (repr. graphique).....57

Figure 2-3 AHT normalisée pour la tâche d'envoi d'un courriel (repr. textuelle). 58

Figure 2-4 OTHI normalisé correspondant à l'AHT partielle pour l'envoi d'un courriel. 62

Figure 2-5 OTHI normalisé identique au précédent hormis l'utilisation de sous-routines. 63

Figure 2-6 Représentation textuelle de l'OTHI de la figure 2-4. 64

Figure 2-7 Exemple de sous-routine selon la syntaxe du tableau 2-9. 66

Figure 2-8 Représentation normalisée possible de l'envoi d'un courriel. 68

Figure 2-9 Deux représentations optionnelles pour le corps d'une tâche MAD. 69

Figure 2-10 Deux analyses KLM-GOMS pour l'entrée de l'adresse du destinataire. 73

Figure 3-1 Figuration de la lentille virtuelle de la vue dite hyperbolique, dans laquelle les parties plus pâles montrent où les éléments à l'écran sont le plus rapetissée et où les déformations graphiques sont plus importantes. 76

Figure 3-2 Simulation des différences entre la vue normale et celle dite hyperbolique. Dans ce cas, c'est la tâche « 3.2 Entrer objet » qui est le point stable entre les vues. 77

Figure 3-3 Fenêtre pour la visualisation et l'édition d'AHT. 79

Figure 3-4 Contenu suggéré de la fenêtre d'acquisition d'une grande quantité de données. 80

Figure 3-5 Représentations graphique et textuelle de l'AHT comme entrée jusqu'ici. 84

Figure 3-6 Outils pour l'insertion de tâches par survol. 86

Figure 3-7 Mini dialogue pour l'entrée des informations concernant une nouvelle tâche. 87

Figure 3-8 Éditeur de plan AHT favorisant leur création de manière structurée. 88

Figure 3-9 Fenêtre pour la visualisation et l'édition d'un OTHI. 93

Figure 3-10 Les trois autres palettes pour la construction d'OTHI. 94

Figure 3-11 Dialogue acceptant les informations relatives à tout nouveau symbole « SI ». 95

Figure 3-12 Outils pour l'insertion de symboles par survol. a) Flèches bleues apparaissant lors du survol d'une tâche. b) Mini palette montrant les quatre onglets apparaissant au-dessus des palettes d'outils. c) L'utilisateur ayant choisi l'onglet «Groupes», la mini palette correspondante apparaît. 96

Figure 3-13 Dialogue des propriétés d'un symbole de groupement. 96

Figure 3-14 OTHI pour la tâche consistant à envoyer un courriel (utilisant deux sous-ordinogrammes non montrés).....97

Figure 3-15 Palettes considérées pour l'insertion ou le remplacement de symboles. 99

Figure 3-16 Éditeur d'OTHI en représentation textuelle.....100

Figure 3-17 Plan assez complexe en représentation graphique.....104

Figure 3-18 Plan de la figure précédente en représentation textuelle. 105

Figure 3-19 Plan des deux figures précédentes en représentation hybride. 105

Figure 3-20 Fenêtre pour la visualisation et l'édition d'une tâche selon MAD. 107

Figure 3-21 Contenu suggéré de la fenêtre d'acquisition d'une masse de données.....108

Figure 3-22 Informations entrées à propos de la tâche principale « Envoyer courriel ». 109

Figure 3-23 Représentations graphique et textuelle de l'arbre MAD comme entrée jusqu'ici. ..110

Figure 3-24 Infobulle montrant les détails de la tâche principale en réponse à son survol. 111

Figure 3-25 Éditeur d'analyse temporelle avec une représentation graphique horizontale.....116

Figure 3-26 Éditeur d'analyse temporelle avec une représentation graphique verticale. 117

Figure 3-27 AHT dans laquelle une tâche est analysée par un OTHI.....123

Figure 3-28 AHT avec une petite insertion en OTHI.....124

Figure 3-29 Visualisation d'informations relatives à une tâche analysée de manière mixte. 126

Figure 3-30 Insertions de petites analyses locales dans la représentation d'une AHT.....127

Figure 3-31 Analyse multiméthode simulant l'application de la méthode CMN-GOMS.....128

Figure 3-32 AHT servant de point de départ aux exemples de conversion AHT ↔ OTHI. 129

Figure 3-33 OTHI résultant du traitement de la racine de l'AHT de départ.....130

Figure 3-34 OTHI correspondant à l'AHT de départ (la hiérarchie est cependant perdue).....131

Figure 3-35 AHT résultant de la conversion d'un OTHI (sans égard à la hiérarchie).....131

Figure 3-36 OTHI où quatre portions ont été identifiées et leurs buts indiqués. 132

LISTE DES SIGLES ET ABRÉVIATIONS

@Esperanto — Le nom du logiciel conceptualisé dans ce mémoire. Ce nom comporte deux parties, l'arobas (« @ ») et « Esperanto », le nom du langage créé par L. L. Zamenhof en 1887.

D'une part, l'arobas qui est quelques fois appelé « at » est, dans ce cas, l'acronyme de « analyse de tâche ». D'autre part, le logiciel conceptualisé ici, à l'image du langage, a pour but d'être un environnement neutre dans lequel tous les analystes de tâche trouveront un environnement qui transcende les frontières et facilite les communications entre les membres de la communauté internationale.

Activité — Dans le monde de l'ergonomie francophone, le terme « activité » désigne chacune des étapes effectuées dans le cadre d'une tâche réelle. Toutefois, dans ce mémoire, le terme « tâche » (qui réfère plutôt aux étapes d'une tâche prescrite) est souvent utilisé pour désigner indifféremment les étapes de la tâche à analyser qui peut aussi bien être la tâche prescrite que la réelle.

AHT — Analyse hiérarchique de tâche.

Aïeul — Tout nœud d'un arbre sur le chemin allant de la racine jusqu'au parent du nœud considéré.

Analyse — Le mot « analyse » ainsi que l'expression « analyse de tâche » sont employés pour désigner une analyse ergonomique de tâche. Il n'est jamais question, dans ce document, d'analyse informatique ou de toute autre forme d'analyse.

Analyste — Comme pour le mot « analyse », le mot « analyste » est réservé pour les analystes de tâche (le plus souvent des ergonomes). Il n'est pas question, dans ce document, d'analyste informatique, financier ou autre…

Arbre — Arbre hiérarchique, soit un graphe acyclique dont un et un seul nœud n'a aucun lien entrant (la racine). Les éléments d'un arbre hiérarchique sont reliés par des relations dites « parent-enfant » dont la signification peut varier selon l'utilisation de l'arbre (par exemple, « moyens-finalité » ou « cause-effet »). Dans le cas de l'analyse hiérarchique de tâche, c'est « but-moyens ».

AT	Analyse de tâche ou analyse temporelle. Ce sigle ayant deux significations possibles, il n'est utilisé que lorsqu'il n'y a aucune confusion possible.
CMN-GOMS	Card-Moran-Newell GOMS.
GOMS	Goals, Operators, Methods, and Selection methods. Une famille de méthodes d'analyse ergonomique de tâche.
Enfant	Tout nœud d'un arbre dont le nœud considéré est le parent direct.
Étude	Ce terme est employé de manière assez générale pour désigner indifféremment une analyse de tâche ou un ensemble d'analyses de tâche effectuées pour un même but.
Frère/sœur	Tout nœud d'un arbre ayant le même parent direct que celui considéré.
Infobulle	Fenêtre, affichée en réponse au survol d'un item, dont le but est de fournir des informations à propos dudit item qui sont complémentaires à celles déjà visibles à l'écran.
IHM	Interface humain-machine.
IU	Interface utilisateur. C'est un cas particulier d'IHM référant spécifiquement à l'interface qu'offre un logiciel à ses utilisateurs.
KLM-GOMS	Keystroke-Level Model GOMS.
MAD	Méthode analytique de description de tâche.
Nœud	Éléments constitutifs d'un arbre qui sont reliés par des relations père-enfant.
OTHI	Ordinogramme du traitement humain de l'information, une forme d'analyse ergonomique de tâche procédurale.
Parent	Nœud d'un arbre directement avant celui considéré dans le chemin entre la racine et lui-même.
Procédure	Suite de tâches (ou activités) à effectuer les unes après les autres.
Racine	Le seul nœud de l'arbre qui n'a pas de parent (donc, de lien entrant).
Support	Ce terme, souvent employé en développement informatique, a plusieurs acceptions. Dans ce mémoire, celle utilisée indique qu'un logiciel ayant été

programmé avec certaines connaissances à propos d'une ou plusieurs méthodes ou techniques est capable d'assister ses utilisateurs dans l'utilisation desdites méthodes ou techniques. Cette assistance peut prendre plusieurs formes : accompagnement des utilisateurs au travers les diverses étapes requises pour réaliser leur but, automatisation de certaines étapes, stockage et validation des données fournies, présentation des résultats avec affichages particuliers, etc.

Survol — Action de placer le pointeur de la souris au-dessus d'un item apparaissant à l'écran, puis d'attendre quelques dixièmes de secondes. De cette action peut résulter l'apparition d'une infobulle.

Tâche — Dans le monde de l'ergonomie francophone, le terme « tâche » est réservé pour les étapes suggérées d'une tâche prescrite. Toutefois, dans ce mémoire, ce terme est souvent utilisé pour désigner indifféremment les étapes de la tâche à analyser qui peut aussi bien être la tâche prescrite que la réelle. Lors d'une décomposition hiérarchique, on parlera alors de tâches, de sous-tâches, de sous-sous-tâches, etc.

Utilisateur — Ce terme est employé dans son acception habituelle en développement informatique. Ce mémoire montrant ce que pourrait être un logiciel d'assistance à l'analyse ergonomique de tâche, les utilisateurs principaux de celui-ci seront des analystes de tâche (qui seront probablement aussi des ergonomes cognitifs). Puisque les résultats d'une analyse de tâche doivent être montrés, compris et acceptés pour toutes les parties prenantes d'un projet, les analystes ne sont pas les seuls utilisateurs du logiciel. Cela dit, il est clair que lesdits analystes auront accès à l'ensemble des fonctionnalités du logiciel, alors que les autres utilisateurs ne pourront que consulter les résultats et ajouter des commentaires.

LISTE DES ANNEXES

ANNEXE A SYNTHÈSE DES FONCTIONNALITÉS ET DES FENÊTRES …….. 138

INTRODUCTION

Cette recherche porte sur la conceptualisation d'un logiciel d'assistance à l'analyse ergonomique de tâche supportant plusieurs méthodes d'analyse de tâches, même concurrentes. Conséquemment, dans ce mémoire, les termes « analyse de tâche » ou simplement « analyse » réfèrent à une analyse ergonomique de tâche et non pas à une analyse informatique d'un programme ou à tout autre type d'analyse.

Cela dit et bien qu'il commence par une explication de ce qu'est une analyse ergonomique de tâche, ce mémoire n'a pas la prétention d'être une introduction à l'analyse de tâche ou à ses méthodes. À tous les lecteurs désireux d'approfondir ce sujet, l'auteur recommande la lecture de Crandall, Klein et Hoffman (2006) ou alors, pour un résumé très correct qui montre bien le contexte historique et permet de bien comprendre les forces et les faiblesses de ce genre d'analyse, Hollnagel (2012). Les diverses méthodes par lesquelles on peut réaliser une analyse de tâche pourront ensuite être étudiées selon les besoins et les intérêts de chacun.

Analyse ergonomique de tâche

Au sens qui nous concerne, une tâche est un travail qu'une personne (ou une équipe) doit accomplir pour atteindre un objectif connu, en général, dans un temps prédéterminé et des conditions prédéfinies. Conséquemment, une bonne analyse de tâche se doit de prendre en considération non seulement les objectifs de la tâche, mais aussi le contexte dans lequel elle doit être accomplie.

Il existe plusieurs justifications pour entreprendre une analyse de tâche, mais toutes ont en commun le besoin de mieux comprendre la tâche. Il se peut, par exemple, que l'on veuille concevoir une interface humain-machine, améliorer les façons de faire actuelles, évaluer l'impact d'un nouvel équipement, créer une formation appropriée, connaître les risques dont il faut se prémunir, connaître les exigences d'une tâche dans le but de sélectionner le personnel, automatiser une tâche, connaître les besoins de main-d'œuvre ou évaluer les coûts de réalisation d'une tâche.

Le terme « analyse de tâche » a une signification quelque peu différente selon les articles qu'on lit à ce sujet. Selon Hollnagel (2012) qui s'appuie sur les résultats d'un sondage fait par Kirwan

et Ainsworth au début des années ‘90, les diverses définitions peuvent être rangées dans l’une des cinq catégories suivantes :

- La description et l’analyse d’observations concernant, soit la manière par laquelle une tâche est réalisée, soit les agissements survenus lors d’événements anormaux (incidents, accidents, etc.).
- L’analyse et la description de tâches ou de situations de travail encore inexistantes ou basées sur d’hypothétiques événements.
- La représentation des situations décrites plus haut en matière de notations utilisées pour recueillir les informations d’intérêt.
- Toutes manières d’analyser et de tirer le maximum des informations recueillies à propos des situations susmentionnées.
- Les modes de représentation des résultats et les différentes façons de documenter ce qui est sorti de la description et de l’analyse.

Parce qu’il y a plusieurs raisons pour réaliser une analyse de tâche, différentes méthodes pour le faire ont été développées au fil des ans, chacune avec ses caractéristiques propres et répondant donc à certains besoins mieux que d’autres. Par exemple, certaines méthodes mettent l’accent sur la décomposition hiérarchique d’une tâche complexe en buts, sous-buts, sous-sous-buts, etc. D’autres visent à établir une procédure pour mettre en évidence un mode d’opération permettant à une personne d’accomplir la tâche. D’autres encore sont préoccupées d’abord et avant tout par le temps nécessaire pour accomplir les différentes étapes permettant d’accomplir la tâche.

Face à un nouveau projet d’analyse de tâche, les analystes doivent donc décider de la ou des méthodes à utiliser en fonction de nombreux critères, dont le but de l’étude et l’aspect à étudier en priorité comme, par exemple, la structure de la tâche, les décisions à prendre ou le temps de complétion. Il leur faut aussi considérer d’autres facteurs, tels les outils disponibles et les méthodes qu’ils supportent, la familiarité des gens impliqués dans le projet avec une méthode ou un type de représentation en particulier, la facilité à communiquer les résultats aux personnes impliquées et d’autres encore.

Le point important ici est qu’aucune méthode n’est une panacée pour l’analyse de tâche. Les analystes doivent donc choisir parmi bon nombre de méthodes celle ou celles qu’ils utiliseront pour arriver à leurs fins.

Autre point, au moment de colliger les données et informations qui serviront de base pour une analyse de tâche, il faut porter une attention toute particulière aux sources. Ceci parce qu’il arrive

souvent que la manière de réaliser une tâche pour l'opérateur intervenant sur le terrain dans un contexte réel diffère de la manière prévue et prescrite. C'est la différence entre une *tâche prescrite* et une *tâche effective*. La première correspond à ce qui est demandé par leurs supérieurs à ceux qui doivent réaliser une tâche (en termes de buts et de résultats à atteindre et souvent aussi d'organisation, de méthodes et d'outils de travail à utiliser, de procédures à suivre, etc.), alors que la deuxième fait référence à ce que font réellement les opérateurs pour réaliser cette tâche, ce qui peut être assez éloigné de la tâche prescrite. D'une certaine manière, on peut voir la tâche effective comme le résultat d'une fonction tenant compte, entre autres, des buts et exigences de la tâche à accomplir, des compétences, aptitudes, expériences et état physiologique de l'opérateur, des caractéristiques du contexte de travail et des moyens disponibles.

Cela dit, une analyse de tâche peut aussi bien porter sur l'une ou l'autre de ces tâches. Il suffit de choisir ses sources d'information en fonction de la tâche d'intérêt et de faire attention à ne pas croiser des informations à propos de la tâche prescrite avec d'autres concernant plutôt la tâche effective.

Indépendamment de la méthode retenue pour effectuer une analyse de tâche, il s'avère qu'avec l'ampleur de la tâche à analyser, la représentation des résultats devient vite difficile à comprendre et surtout à gérer… Et, sans surprise, les tâches qui valent la peine d'être analysées comportent le plus souvent un grand nombre d'opérations et présentent donc une certaine complexité, soit dans l'exécution de certaines étapes en particulier, soit dans les interrelations entre celles-ci. En réponse à cette situation, quelques outils logiciels ont été créés pour alléger tout au moins les problèmes liés à la création et surtout à la modification de la représentation desdits résultats puisqu'elle peut amener des modifications en cascade (sur la numérotation des tâches, par exemple).

De ces outils logiciels, la plupart ont été créés pour assister les analystes dans la représentation de leurs résultats d'analyse de tâche selon les préceptes de l'une ou l'autre des principales méthodes connues. Par exemple, `TaskArchitect©` [1] assiste ses utilisateurs dans l'élaboration d'analyse

---

[1] Un logiciel de Task Architect, Inc, qui en détient la marque de commerce. Cette compagnie a annoncé sa fermeture en juillet 2013. En conséquence, TaskArchitect n'est plus vendu et la compagnie a offert de vendre le logiciel à une entité qui voudrait en reprendre le développement.

de tâche selon la méthode nommée « analyse de tâche hiérarchique », alors que KMADe© [2], lui, utilise plutôt la méthode nommée « méthode analytique de description de tâche ».

De tous les outils logiciels trouvés, seul Dolphin[3] semble tenir compte de plus d'une méthode d'analyse, ce que le logiciel @Esperanto, conçu et présenté dans ce mémoire, se propose de faire. Par contre, aucune trace d'utilisation n'a pu être localisée. Il se pourrait donc que la seule mention de ce logiciel apparaisse dans Limbourg et al. (2001), un article dont le point principal concerne la création d'une topologie qui doit permettre d'exprimer un maximum des concepts d'une méthode dans une autre des méthodes que supporte Dolphin.

Malgré la possibilité d'utiliser un logiciel de dessin généraliste comme Visio© [4] pour créer une représentation des résultats d'analyse de tâche, ce genre de logiciels n'est en aucune manière concurrent d'@Esperanto ou de n'importe lequel des logiciels que l'on vient de mentionner, puisque les généralistes n'ont aucune connaissance de l'analyse de tâche et ne peuvent donc pas prétendre supporter une quelconque méthode d'analyse de tâche.

But d'@Esperanto : Supporter l'analyse de tâche utilisant plusieurs méthodes

Il arrive souvent qu'un analyste veuille utiliser une méthode pour la majeure partie de son analyse de tâche et d'une ou de plusieurs autres pour des parties spécifiques. Par exemple, il peut être approprié d'approcher une tâche complexe en utilisant une méthode qui nous amènera à en faire une décomposition hiérarchique tout en pouvant utiliser une deuxième méthode qui permettra l'étude détaillée du temps requis pour chacune des procédures élémentaires qu'elle nécessite. C'est précisément là le but du logiciel @Esperanto.

@Esperanto[5] est donc un logiciel d'analyse ergonomique de tâche qui offre la possibilité d'utiliser plusieurs méthodes même dans le cadre d'une seule analyse. Il est destiné à des ergonomes, des ingénieurs, des informaticiens, des responsables de formation, des professeurs,

---

[2] Un logiciel en licence libre (GNU LGPL) dont la marque de commerce est détenue par Free Software Foundation.

[3] Un logiciel développé pour fins de recherche par Limbourg, Pribeanu et Vanderdonckt (2001).

[4] Un logiciel de Microsoft, Inc, qui en détient la marque de commerce.

[5] Évidemment, ce logiciel est encore un rêve, mais dans le reste de ce mémoire on en parlera comme s'il existait.

des étudiants ou tout autre professionnel devant effectuer une analyse de tâche et désirant bénéficier d'une assistance logicielle dans l'élaboration de son analyse.

@Esperanto offre la possibilité de changer de méthode pour les portions choisies par l'utilisateur, permettant ainsi de convertir (traduire d'une certaine manière) une analyse de tâche réalisée avec une méthode en son équivalent d'une autre méthode, ceci dans le but de permettre aux utilisateurs de changer de méthode en cours de route sans pour autant perdre tout ce qui aurait été fait selon la première méthode.

@Esperanto doit aussi aider à définir la terminologie qui devrait être employée tout au long d'une analyse de tâche par l'utilisation de réseaux de concepts (donc, non dirigés) et assister l'utilisateur dans la création d'une vue d'ensemble du domaine d'application par l'utilisation de réseaux dirigés. Finalement, dans la mesure où les intrants de l'analyse (résultats d'interview, notes d'observations, prises de mesure, etc.) lui sont connus, il doit permettre une certaine traçabilité du contenu de l'analyse de tâche.[6]

Méthodes supportées

Les méthodes d'analyse supportées par @Esperanto ont été choisies avec l'objectif d'offrir un bon éventail de possibilités aux analystes afin de leur permettre de s'attaquer à divers problèmes d'analyse, de diverses manières selon les besoins de chaque projet et les préférences de chacun. Spécifiquement, ces méthodes sont les suivantes :

AHT	Analyse hiérarchique de tâche
MAD	Méthode analytique de description de tâche
OTHI	Ordinogramme de traitement humain de l'information (Une forme d'analyse procédurale utilisant les ordinogrammes.)
AT	Analyse temporelle
KLM-GOMS[a]	Keystroke-Level Model GOMS
CMN-GOMS[a]	Card-Moran-Newell GOMS

[a] Une variante de la méthode GOMS (*Goals, Operators, Methods, and Selection rules*).

[6] Évidemment, le degré auquel @Esperanto peut permettre la traçabilité dépend de ce qui est porté à sa connaissance. Au pire, si l'utilisateur ne fait connaître aucun intrant au logiciel, aucune traçabilité n'est possible.

Parmi ces méthodes, certaines mettent l'accent sur la décomposition hiérarchique de la tâche pour la comprendre de manière à pouvoir l'améliorer (changement des façons de faire, automatisation, etc.), la standardiser (homogénéisation d'un site à l'autre, formation, etc.) ou pour toutes autres raisons. D'autres s'attardent plutôt à la séquence d'opérations et de décisions impliquées dans la réalisation de la tâche pour, par exemple, l'étudier finement de manière à optimiser un processus ou à l'encoder pour permettre son exécution par des techniciens adéquatement formés, mais ne possédant pas une connaissance scientifique ou d'ensemble du domaine d'application. D'autres encore permettent d'étudier ou de prévoir le temps nécessaire pour achever une tâche.

Un point commun aux méthodes d'analyse retenues pour @Esperanto est que ce sont toutes des méthodes d'analyse de tâche individuelle. Bien que certaines puissent être adaptées pour analyser des tâches réalisées par une équipe plutôt que par un individu, ce n'est pas leur force.

Ces méthodes sont présentées au chapitre suivant.

Réseaux

Avant de commencer la ou les analyses de tâche nécessaires dans un domaine d'application encore inconnu d'eux, les analystes doivent établir une taxonomie (ou une simple terminologie) de ce domaine pour communiquer au mieux avec les experts de celui-ci. En effet, non seulement les analystes doivent-ils connaître le vocabulaire et les concepts manipulés par les diverses parties prenantes du domaine, ils doivent aussi produire leurs résultats sous forme aisément compréhensible par eux, ce qui implique d'utiliser leur terminologie.

En plus de la taxonomie (ou terminologie), il est nécessaire que les analystes aient une vue d'ensemble des tâches qui pourraient leur être mentionnées dans le cadre de leur étude. Bien sûr, il ne s'agit pas nécessairement de décrire l'ensemble des tâches reliées au domaine d'application, mais tout au moins celles dont les analystes vont entendre parler lors de leur collecte de données. Encore une fois, le but ici est de faciliter la communication entre les analystes et les experts.

Pour combler ces deux besoins, @Esperanto supporte l'utilisation de réseaux qu'ils soient dirigés ou non.

La structure du mémoire

Trois chapitres constituent le corps de ce mémoire. Au chapitre 1, les différentes méthodes d'analyse supportées par @Esperanto sont présentées comme elles sont généralement connues

par les analystes ergonomiques de tâche. On y présente aussi les réseaux (bien élémentaires) qui sont supportés par @Esperanto.

Au chapitre 2, pour certaines des méthodes supportées, on décrit les modifications apportées pour étendre leur expressivité et on explique les implications de celles-ci sur les possibilités d'analyse.

Au chapitre 3, différentes interfaces utilisateurs sont proposées pour créer et modifier des analyses de tâche selon chaque méthode supportée. C'est à la fin de ce chapitre que l'on aborde vraiment l'aspect multiméthode d'@Esperanto.

Pour terminer, on conclut en plaidant pour l'implémentation d'@Esperanto.

CHAPITRE 1 PRÉSENTATION DES MÉTHODES D'ANALYSE DE TÂCHE

Ce chapitre présente les méthodes d'analyse ergonomique de tâche individuelle qui seront prises en charge par @Esperanto, soit celles identifiées dans l'introduction. Mentionnons que, bien qu'il existe des méthodes comme GroupWare Task Analysis et ConcurTask Trees[7], qui permettent nativement d'analyser des tâches devant être réalisée par plusieurs équipiers, celles-ci ne seront pas considérées dans la première version d'@Esperanto.

On ne cherche ici qu'à présenter les grandes lignes de chacune des méthodes à supporter. Chacune d'elles est donc sommairement décrite dans son état habituel d'utilisation (possibilités, limitations, notations, etc.) dans le but de rappeler ce qu'est chaque méthode et, dans certains cas, de mettre en évidence les caractéristiques importantes pour la suite de ce document.

1.1 Quelques points communs

Avant de décrire les particularités de chacune des méthodes d'analyse de tâche qui nous intéressent, on s'attardera d'abord sur leurs points communs. En effet, étant toutes des méthodes d'analyse de tâche, elles partagent plusieurs choses.

Premièrement, avant de commencer une analyse de tâche, il faut savoir pourquoi elle doit être réalisée, puisque ce n'est qu'en se basant sur les buts que l'analyste pourra adéquatement choisir les aspects de la tâche auxquels il doit porter une attention particulière et, comme on le verra plus loin, limiter la granularité de l'étude sans affecter la pertinence des résultats. Il est donc impératif de connaître les buts à atteindre avant de commencer une analyse de tâche. En fait, de ceux-ci dépendra en partie aussi le choix même de la méthode d'analyse à employer.

Deuxièmement, quelle que soit la méthode retenue, il est essentiel d'assurer la validité des informations relatives à la tâche étudiée. Cette responsabilité incombe aux analystes, puisque les différentes méthodes ne servent qu'à agencer les informations relatives aux tâches d'une manière

---

[7] Voir respectivement van der Veer, Lenting et Bergevoet (1996) et Paterno, Mancini et Meniconi (1997).

qui rendra possible la meilleure analyse.[8] L'adage anglais « garbage in; garbage out! »[9] résume très bien la raison pour laquelle tous doivent se préoccuper de la véracité des informations colligées dont sont nourris les outils mis à leur disposition. Pour maximiser la probabilité de bien réussir, les analystes se doivent d'en apprendre assez à propos de la tâche à analyser et de son domaine d'application pour être capables, non seulement de bien communiquer avec les gens du domaine, mais aussi de détecter des erreurs ou contradictions parmi les intrants qu'ils recueillent.[10]

En corollaire de ce qui précède, on notera qu'il est tout aussi facile d'agencer des informations relatives à une tâche prescrite qu'à une tâche réelle. La seule différence ici est encore une fois sous l'entier contrôle des utilisateurs d'@Esperanto.[11] C'est donc à eux d'adapter leur recueil de données pour obtenir des informations véridiques et concernant exactement ce qu'ils veulent étudier.

Troisièmement, par définition, il est impossible de prévoir l'imprévisible. Bien sûr, il est possible de tenir compte d'autant de circonstances inhabituelles que désiré (erreurs humaines, cas d'exception, incidents, accidents, situations d'urgence ou autres), mais au prix d'une lourdeur disproportionnée dans la représentation résultante. Pour cette raison, la plupart des analyses de tâche ne tiennent pas compte des erreurs humaines. Dans certains cas, GOMS par exemple, on avoue même ne s'intéresser qu'aux tâches réalisées sans erreurs et de manière experte (donc sans hésitation quant à l'enchaînement des étapes devant être réalisées). Cela dit, il est faisable d'obtenir des informations qui permettraient de tenir compte du niveau d'expertise de l'opérateur, mais l'effort de validation qui serait alors requis rendrait l'entreprise par trop longue et

---

[8] Ici, la définition de « meilleure » est bien sûr dépendante des buts de l'analyse et donc du choix de la méthode d'analyse en fonction de son adéquation avec ceux-ci.

[9] En français, ce serait « [des] ordures en entrée [entraînent la production d'] ordures en sortie! ».

[10] Il est bien clair que les analystes n'auront aucune autorité pour affirmer que ce qu'ils ont détecté est une erreur ou une contradiction. Il leur faudra faire confirmer (et corriger) ce genre de choses par un expert.

[11] On suppose que les utilisateurs d'@Esperanto seront des ergonomes devant réaliser une analyse de tâche, mais le logiciel devrait être facile d'utilisation pour toute personne connaissant les méthodes d'analyse supportées.

coûteuse.[12] Dans les domaines où la sécurité est importante, il est courant d'analyser un grand nombre de scénarios possibles impliquant diverses conditions exceptionnelles (erreurs, incidents ou accidents) en vue de former adéquatement les opérateurs et de leur offrir des moyens de travail performants.

En conséquence des points précédents, on s'aperçoit que lors de situations inhabituelles, si on n'a rien prévu au moment de la conception du système ou de la formation des opérateurs, le résultat d'une analyse de tâche peut être en grande partie inutile pour rétablir la situation. Cela parce que les méthodes d'analyse de tâche ne s'attardent qu'à décrire les buts à atteindre et les actions à prendre pour réussir l'entreprise. Il n'est nulle part fait explicitement mention de l'état du système dans lequel les opérateurs doivent évoluer, ce qui est nécessaire à la détection et à la correction d'une situation imprévue. Au prix de complexifier l'analyse de tâche résultante, il est toutefois possible de pallier à cet inconvénient en ajoutant des informations supplémentaires utiles au diagnostic d'anomalie. Ce prix semble souvent prohibitif au vu de la complexité additionnelle de l'analyse de tâche.

Quatrièmement, toutes les méthodes d'analyse procèdent en décomposant la tâche en unités (tâches ou activités) plus petits. Certaines d'entre elles, dont le but est de bien comprendre la structure de réalisation d'une tâche, utilisent une décomposition hiérarchique de la tâche analysée, ce qui correspond à une manière toute naturelle pour les humains d'aborder un problème complexe. Les autres méthodes, dont le but est surtout d'étudier le temps ou l'ordre d'exécution des activités élémentaires, utilisent plutôt une décomposition linéaire (ou procédurale) de la tâche principale.

On notera ici que les méthodes utilisant une décomposition hiérarchique tiennent aussi compte des buts d'une tâche à analyser. Ainsi, comme on le verra dans la description de l'analyse hiérarchique de tâche, on parlera de buts et de sous-buts, en plus des tâches et des activités.

Finalement, dans tous les cas, il est nécessaire de savoir quand arrêter l'analyse d'une tâche. Le but ici est de ne poursuivre l'analyse que si c'est rentable et productif. Il existe donc différents

---

[12] Ceci est d'autant plus vrai lorsque le temps d'exécution est un élément important de l'analyse.

critères dits d'arrêt pour décider lors de l'analyse d'une tâche de l'adéquation du niveau de détail courant par rapport aux buts à atteindre. On a donc les critères d'arrêt suivants :

- Niveau de détail satisfaisant : toutes les parties prenantes s'entendent pour affirmer qu'une tâche ou une activité a été décomposée à un niveau assez fin pour justifier de ne pas poursuivre son analyse.
- Inadéquation avec les buts : rien ne peut justifier la poursuite de l'analyse d'une tâche ou d'une activité qui n'a aucun impact sur l'atteinte des buts de celle-ci.
- Faible probabilité ou impact des erreurs qui pourraient être commises si on ne poursuivait pas l'analyse plus avant : si le risque encouru en n'analysant pas une tâche ou une activité est soit très faible, soit sans conséquence notable, il est inutile de la poursuivre. Ce critère est souvent associé à l'analyse hiérarchique de tâche, qui l'a fait connaître sous la forme de l'équation $p \times C$ (où p est la probabilité estimée de ces possibles erreurs si l'analyse d'une tâche T_i n'est pas faite et où C est leur coût estimé)[13]. Ce critère, plein de gros bon sens, doit véritablement s'appliquer à toutes les méthodes.

1.2 Réseaux

Dans le cadre d'une analyse de tâche, les réseaux servent surtout lors des premières étapes sous-entendues à la section précédente parce qu'ils permettent de montrer les relations entre les divers éléments d'un domaine d'application ou d'un système technique ou collaboratif.

En particulier, les réseaux sont utilisés pour mettre en relation :

- Les termes et les concepts d'un domaine d'application. Dans ce cas, ils servent donc à établir la taxonomie du domaine d'application.
- La place de chaque tâche dans la chorégraphie générale prescrite ou réelle survenant dans le milieu de l'étude.
- Les responsabilités de chaque personne impliquée dans la chorégraphie susmentionnée.

Pour faciliter la compréhension des réseaux produits, @Esperanto permet aux utilisateurs de définir plusieurs types d'arcs en fonction de leur signification. Par exemple, dans un réseau du personnel d'une usine, on pourrait avoir des liens d'autorité hiérarchique (figurés par des flèches en traits pleins) et des liens d'obligations ou dépendances fonctionnelles (figurés par des flèches en pointillé). Il est possible d'aller encore plus loin en apposant une étiquette sur n'importe quel lien du réseau.

---

[13] L'analyse d'une tâche pouvant être arrêtée du moment où le résultat de l'équation est jugé assez faible.

Puisque les réseaux ne sont pas essentiels à l'analyse de tâche et qu'ils ne feront pas faire partie de la première implémentation d'@Esperanto, ils ne seront pas mentionnés plus avant dans ce mémoire.

1.3 Analyse hiérarchique de tâche (AHT)

Depuis son introduction par Annett et Duncan (1967), l'AHT a été utilisée pour analyser des tâches dans virtuellement toutes les sphères de l'activité humaine; une belle preuve de son utilité et de sa relative simplicité d'utilisation.

Par la date de son introduction, bien antérieure à la spectaculaire pénétration de l'informatique personnelle, on comprend aussi que l'AHT est une méthode ne nécessitant pas l'utilisation d'un ordinateur. Peut-être aussi en bonne partie pour cette raison, son formalisme est plutôt désinvolte, d'où la relative simplicité d'utilisation qui en a facilité l'acceptation, mais aussi le manque de rigueur que certains lui reprochent.

Pour analyser une tâche selon cette méthode, il faut commencer par déterminer le but que doit atteindre la tâche à analyser et commencer une décomposition hiérarchique à partir de là. Lors de cette décomposition, il faudra redéfinir toutes tâches suffisamment complexes qu'il est besoin de l'expliquer. Pour chacune de ces dernières, la redéfinition consiste à indiquer la liste des sous-buts à atteindre ou des tâches à effectuer qui contribuent à sa réalisation, ainsi que les instructions nécessaires pour y arriver. Ce processus doit se poursuivre récursivement pour tous les sous-buts et tâches qui viennent d'être spécifiés. La décomposition hiérarchique doit se poursuivre ainsi jusqu'à ce que les conditions d'arrêt mentionnées plus haut permettent d'y mettre fin.

Parce qu'une AHT peut servir dans de multiples contextes, les propriétés à conserver concernant chaque tâche ne sont pas fixées dans la définition de la méthode. Il est plutôt laissé aux analystes de définir eux-mêmes les propriétés qui leurs seront nécessaires. Par exemple, pour une étude A, on pourrait vouloir connaître les habiletés et connaissances requises pour accomplir chaque tâche, alors que pour l'étude B on aura plutôt besoin de connaître les équipements requis, les codes du travail applicables et les documents légaux qu'il faudra produire pour chaque tâche. On verra plus loin comment ces propriétés sont représentées.

En AHT, les instructions que l'on vient de mentionner sont connues sous le nom de « plan ». Ce plan, requis pour tout nœud possédant des enfants (sous-buts ou tâches), montre comment il faut

utiliser lesdits enfants pour réaliser le but ou la tâche du nœud associé. Ces plans, dont le langage n'est qu'en partie formel[14], permet bien de stipuler les instructions nécessaires, mais peut rendre difficile l'utilisation d'outils de validation ou de simulation. La notation de ces plans varie d'un auteur à l'autre[15], mais permet toujours de faire essentiellement les mêmes choses. Dans ce chapitre, les notations adoptées sont basées sur une suggestion de Stanton (2006) et sur les possibilités de TaskArchitect©.[16] La première est expliquée au tableau 1-1, l'autre au tableau 1-2.

À titre comparatif, notons que TaskArchitect© et le logiciel équivalent dans la suite Human Factors Workbench (HFW)[17] utilisent tous deux la notation plus verbeuse (mais de premier abord plus claire) pour la création de leurs plans.

On constate donc que, dans l'élaboration de cette méthode, l'accent a été mis sur la facilité d'utilisation pour permettre d'effectuer l'analyse d'une tâche et d'obtenir un résultat utilisable par des humains et, ce, sans recours à un ordinateur. Que le lecteur n'interprète pas cela comme une critique de la méthode, mais comme une simple observation. Après tout, si cette méthode est devenue et est restée un incontournable en ergonomie, c'est qu'elle remplit très bien sa fonction.

À titre d'exemple, voir la figure 1-1 où l'on notera que les tâches sont numérotées hiérarchiquement. En général, les analystes adoptent les conventions suivantes :

- La racine porte le numéro « 0 ».
- Les enfants de la racine portent un nombre entier (« 1 » pour le plus à gauche et successivement en allant vers la droite).
- Les enfants de tous les autres nœuds portent un numéro qui résulte de la concaténation du numéro du parent, d'un point et d'un nombre entier (« 1 » pour le plus à gauche et successivement en allant vers la droite).

---

[14] Les différents logiciels supportant l'élaboration de modèles AHT proposent aux utilisateurs un langage un tant soit peu formel pour la rédaction des plans. Au besoin et au prix d'un support moindre de la part du logiciel, les analystes peuvent toujours revenir à un langage tout à fait informel pour exprimer les conditions les plus élaborées… Précisions et détails peuvent toujours être au rendez-vous!

[15] Les variations sont telles qu'une légende explicative est le plus souvent utilisée.

[16] TaskArchitect© est le plus connu des logiciels d'assistance à l'analyse de tâche selon la méthode AHT.

[17] HFW est la suite logicielle de *Human Reliability* contenant plusieurs outils d'analyse dans le domaine de l'ergonomie des systèmes critiques, dont un pour l'AHT.

Tableau 1-1 Énoncés disponibles pour les plans d'une AHT.

Concept	**Notation**	**Commentaires**
Sous-tâches d'une tâche	<entier à partir de 1>	Les sous-tâches servant à redéfinir une tâche sont numérotées séquentiellement à partir de « 1 » en commençant par celle de gauche.
Exécution séquentielle	1 > 2 > 3 > 4	Toutes les opérations doivent être exécutées dans l'ordre indiqué.
Exécution non séquentielle	1 / 2 / 3 / 4	Toutes les opérations doivent être exécutées sans égard à l'ordre indiqué.
Exécution libre	1 : 2 : 3 : 4	Un certain nombre d'opérations (mais au moins une) doivent être exécutées, peu importe l'ordre.
Exécution d'un choix	1 | 2 | 3 | 4	Une seule des opérations mentionnées doit être exécutée.
Exécution concurrente	1 + 2 + 3 + 4	Toutes les opérations doivent être exécutées concurremment. En AHT, toutefois, le mot « concurremment » peut aussi bien désigner des opérations à faire en même temps par plus d'une personne (en simultané) que des opérations à faire en temps partagé par une seule personne (en parallèle).
Interdiction	¬ <opération>	Ne pas faire l'opération mentionnée.
Exécution conditionnelle	si X alors 1 sinon 2	Selon que la condition « X » est réalisée ou non, on choisit la sous-tâche 1 ou la sous-tâche 2.
Instructions textuelles	« Facultatif », « Au besoin », « Quand vide», etc.	Des instructions en langage naturel peuvent toujours être ajoutées pour spécifier au mieux les détails du plan.
Regroupement d'instructions pour ciblage précis.	(…)	Lorsque nécessaire, il est possible de regrouper certaines instructions pour y appliquer des conditions particulières ou pour modifier localement la séquence d'exécution.

Tableau 1-2 Notation utilisée par TaskArchitect© pour les plans d'une AHT.

Concept	**Notation originale**	**Notation traduite par l'auteur**
Exécution séquentielle	Do in sequence <tasks>	Faire en séquence <tâches>
Exécution non séquentielle	Do in any order <tasks>	Faire sans égard à l'ordre <tâches>
Exécution libre	Do any of <tasks>	Faire l'une de <tâches>
Exécution d'un choix	Do only one of <tasks>	Faire une seule de <tâches>
Exécution concurrente	Do concurrently <tasks>	Faire en même temps <tâches>
Interdiction	Do not do <tasks>	Ne pas faire <tâches>
Exécution conditionnelle	If <condition> do <tasks>	Si <condition> faire <tâches>

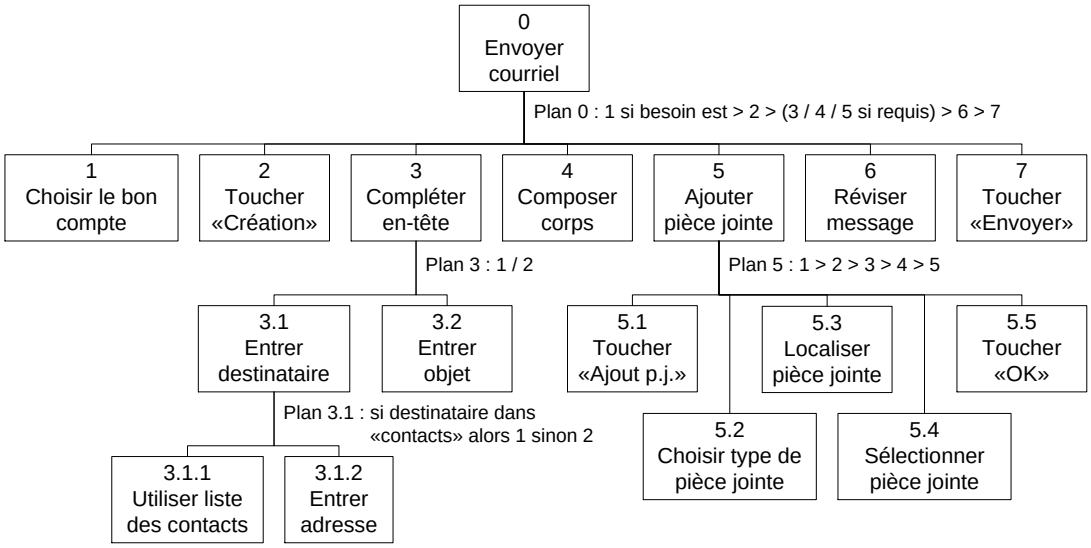


Figure 1-1 AHT partielle pour la tâche consistant à envoyer un courriel.

Le plan associé à un nœud est placé entre celui-ci et ses enfants. Un plan complexe pouvant apparaître au complet (quitte à augmenter l'espacement vertical entre celui-ci et ses enfants) ou être défini plus loin, ce qui doit alors être indiqué par une note là où attendrait ledit plan. On notera aussi que, dans chaque plan, les enfants sont référencés par leur numéro le plus à droite et non en listant tout le numéro hiérarchique (p. ex. : « 2 » au lieu de « 3.1.2 »).

Comme tout arbre, une AHT peut être représentée sous la forme d'un tableau. C'est cette représentation qui est la principale réponse à la question laissée en suspens plus tôt, soit celle concernant les propriétés associées à chaque tâche qui sont spécifiques à chaque analyse de tâche.

Dans l'exemple de la figure 1-2 (qui reprend l'exemple de la figure 1-1), on notera que, par l'ajout de colonnes supplémentaires audit tableau, il est facile de montrer autant de propriétés que souhaité.

Tâches	**Type d'entrée**	**Commentaire**
0 Envoyer courriel		
Plan 0 : 1 si besoin est > 2 > (3 / 4 / 5 si requis) > 6 > 7		
1 Choisir le bon compte	Sélection	Il faudrait mémoriser ce compte pour la prochaine fois.
2 Toucher «Création»	Clic (bouton)	
3 Compléter en-tête		
Plan 3 : 1 / 2		
3.1 Entrer destinataire		
Plan 3.1 : si destinataire dans «contacts» alors 1 sinon 2		
3.1.1 Utiliser liste des contacts	Clic (élément)	
3.1.2 Entrer adresse	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.
3.2 Entrer objet	Texte	
4 Composer corps	Texte	
5 Ajouter pièce jointe		Att : Utilisation occasionnelle!
Plan 5 : 1 > 2 > 3 > 4 > 5		
5.1 Toucher «Ajout p.j.»	Clic (bouton)	
5.2 Choisir type de pièce jointe	Clic (élément)	
5.3 Localiser pièce jointe	Navigation	Fichier local seulement?
5.4 Sélectionner pièce jointe	Clic (élément)	
5.5 Toucher «OK»	Clic (bouton)	
6 Réviser message		
7 Toucher «Envoyer»	Clic (bouton)	

Figure 1-2 AHT de la figure 1 1 en représentation textuelle.

En terminant, notons qu'aucune terminologie pour les plans n'est universellement acceptée, de sorte que la plupart des analyses AHT ont une légende expliquant celle utilisée. Au mieux, observe-t-on une cohérence intra analyste dans l'utilisation de la méthode, mais il existe clairement de nettes différences inter analystes. On remarque aussi que les différents fabricants de logiciels assistant à la création d'AHT proposent l'utilisation d'une notation plutôt formelle

qui se doit cependant de garder des éléments en langage naturel pour permettre la description de toutes les conditions possibles.

1.4 Analyse procédurale / Ordinogramme de traitement humain de l'information (OTHI)

Cette méthode est une forme particulière d'analyse procédurale qui utilise l'ordinogramme (dans sa représentation graphique empruntée au monde de l'informatique) comme outil de représentation principal.

Cela dit, la raison d'être de cette méthode n'est pas seulement d'exposer la décomposition d'une tâche en sous-tâches, mais aussi et surtout de montrer comment atteindre un but par l'exécution ordonnée de certaines tâches, surtout lorsqu'il faut choisir parmi plusieurs chemins d'exécution en fonction de facteurs divers.

L'ordinogramme de l'OTHI sert à décrire graphiquement des tâches par leur enchaînement de processus et de décisions. À la base, les quelques symboles apparaissant à la figure 1-3 suffisent pour représenter un grand nombre de tâches.

Outre ces symboles de base, l'ANSI (*American National Standard Institute*) et différents fabricants de logiciels d'aide à la création d'organigrammes[18] en ont défini plusieurs autres pour éviter de surutiliser le rectangle qui, dans la version de base, sert à toutes les sauces. C'est ainsi que les symboles montrés à la figure 1-4 peuvent être utilisés en lieu et place de rectangles pour communiquer graphiquement un aspect sémantique important d'une tâche.

La figure 1-5 montre une version OTHI de l'exemple d'analyse AHT montrée à la figure 1-1. Comme on le constatera, cet OTHI utilise certains de ces derniers symboles.

---

[18] Dans certains domaines d'application, « organigramme » est le terme usuel pour désigner un ordinogramme.

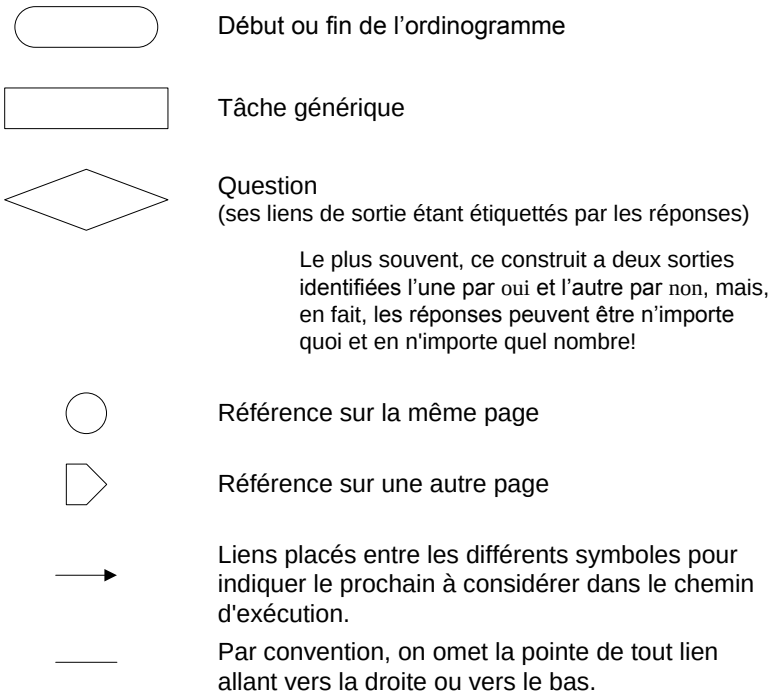


Figure 1-3 Symboles de base d'un ordinogramme.

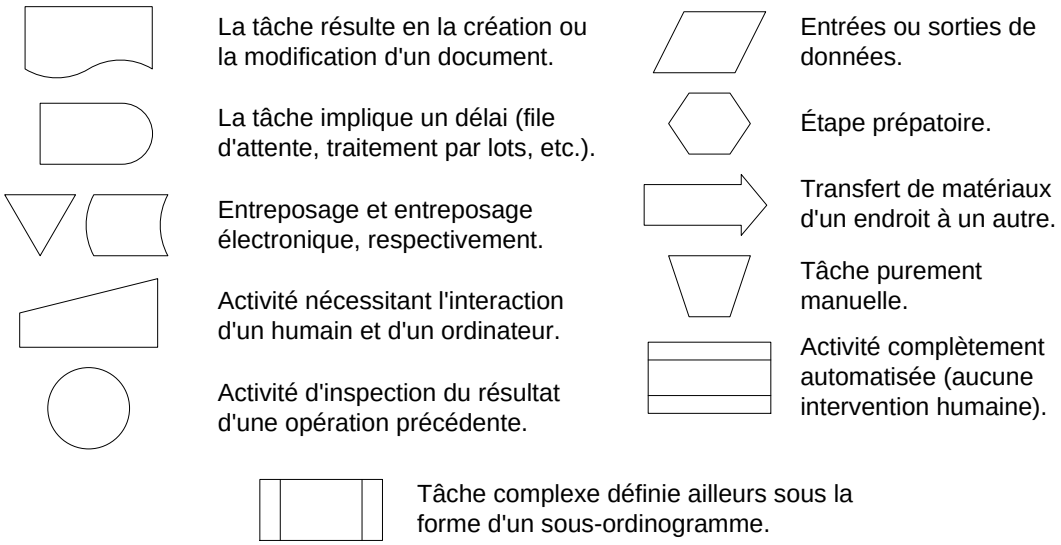


Figure 1-4 Symboles supplémentaires représentant un aspect important d'une tâche.

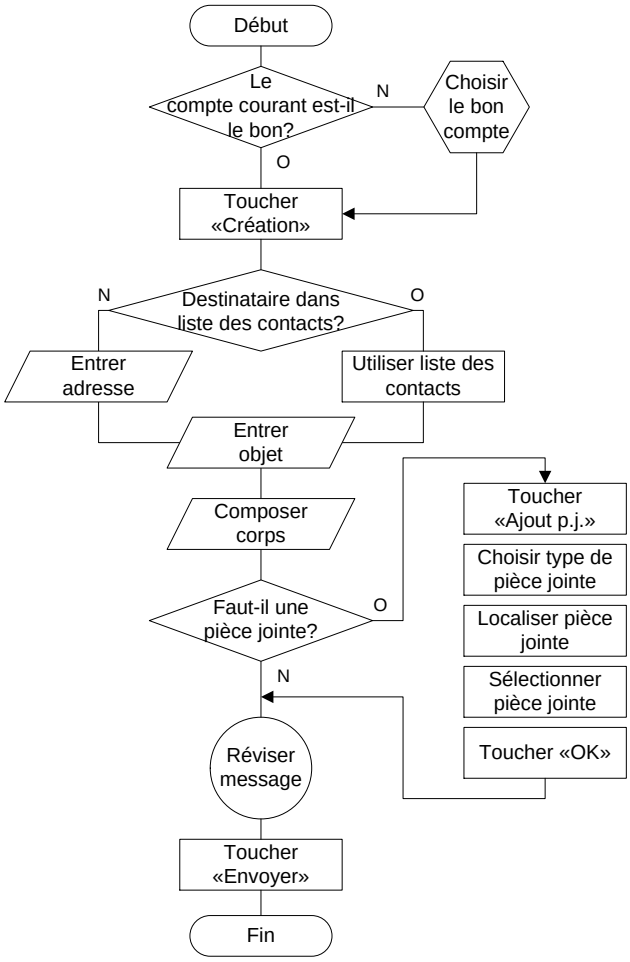


Figure 1-5 OTHI correspondant approximativement à l'AHT partielle pour la tâche consistant à envoyer un courriel.

En regardant la figure 1-5, on pourrait ne pas réaliser que les conditions pour effectuer les tâches dites conditionnelles en AHT sont devenues plus précises en OTHI. Ce résultat est en partie dû à l'attention portée au processus à accomplir et aux décisions qu'il faut prendre.

Bien que la représentation graphique de l'OTHI soit la préférée de la méthode, comme toutes les méthodes d'analyse procédurale, elle peut être montrée sous forme textuelle. Un exemple assez rudimentaire de cette représentation est présenté à la figure 1-6, où les énoncés sont énumérés du haut vers le bas en commençant par l'énoncé qui doit être exécuté en premier. Dans cette version rudimentaire, les énoncés sont simplement numérotés (pour permettre de modifier le flot

d'exécution à un énoncé autre que le suivant) et sont indentés en fonction de leur dépendance à un autre énoncé.

#	Énoncés et tâches	Type d'entrée	Commentaire
1	**SI** le compte n'est pas le bon **ALORS**		
2	Choisir le bon compte	Sélection	Il faudrait mémoriser ce compte pour la prochaine fois.
3	Toucher « Création »	Clic (bouton)	
4	**SI** destinataire dans liste des contacts **ALORS**		
5	Utiliser la liste des contacts	Clic (élément)	
6	**SINON**		
7	Entrer l'adresse courriel	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.
8	Entrer l'objet	Texte	
9	Composer le corps	Texte	
10	**SI** il doit y avoir une pièce jointe **ALORS**		Att : Utilisation occasionnelle!
11	Toucher « Ajout p.j. »	Clic (bouton)	
12	Choisir le type de pièce jointe	Clic (élément)	
13	Localiser la pièce jointe	Navigation	Fichier local seulement?
14	Sélectionner la pièce jointe	Clic (élément)	
15	Toucher « OK »	Clic (bouton)	
16	Réviser le message		
17	Toucher « Envoyer »	Clic (bouton)	

Figure 1-6 Représentation textuelle de l'OTHI précédent.

Lorsque la longueur et la complexité de l'OTHI augmentent, sa représentation textuelle peut devenir difficile à suivre. Pour atténuer ce problème, divers éléments graphiques peuvent être ajoutés au texte. C'est ainsi que l'on peut ajouter des lignes verticales pour mettre en évidence le niveau d'indentation courant même dans un OTHI s'étirant sur plusieurs pages, que l'on peut faire ressortir certains énoncés clés (les SI … ALORS, SINON, TANTQUE et autres) et que l'on peut utiliser un fil d'Ariane (en en-tête de page) pour donner le contexte du premier énoncé de chaque page. Cette représentation, servant de point de départ pour un type de séquence d'instructions très précises connu sous le nom de « procédure », est plus compacte que sa

consœur graphique et peut être très utile pour favoriser l'exécution d'une tâche complexe par du personnel adéquatement formé.

On verra plus loin qu'il ne suffit pas d'ordonner les opérations à faire pour obtenir une procédure adéquate en tant que support à la performance humaine. En effet, leur rédaction est sujette à plusieurs règles d'écriture dont le but est de maximiser leur utilité à cette fin. Cela dit, pour l'instant, on peut continuer en supposant que la représentation textuelle d'un OTHI est assez proche d'une procédure.

Dans certaines circonstances, il est désirable d'introduire une hiérarchie dans ce qui est essentiellement une liste de tâches. Il ne s'agit pas, ici, de relations du type moyens-finalités comme celles que l'on retrouve, entre autres, dans une AHT ou dans MAD. Ce sont plutôt des relations du type existant dans la plupart des documents techniques assez longs entre leurs chapitres, sections et sous-sections. Dans une procédure assez longue, les titres de section pourraient donc servir de repères aux utilisateurs tout en leur donnant une indication de l'aspect important de chaque section. Par exemple, l'appareil ou l'IHM qui en sera le centre d'attention, la phase du processus à laquelle se rapportent les énoncés suivants ou tout autre aspect que l'auteur aura jugé important.

En terminant avec l'OTHI, on notera que les principaux avantages de cette méthode sont sa facilité d'utilisation et de lecture, la présence de son outil de base (l'ordinogramme) dans plusieurs domaines d'application et son applicabilité à la plupart des situations, sans oublier sa notation textuelle très utilisée dans les situations de commandement et de contrôle (en anglais, « Command & Control »).

Par contre, l'OTHI ne permet pas de bien exprimer les concepts de parallélisme ou de simultanéité, non plus que diverses modalités d'exécution des tâches (p. ex., faire les tâches spécifiées dans n'importe quel ordre ou ne faire qu'une seule de ces tâches). On verra plus loin comment l'ajout de symboles appropriés peut corriger ces inconvénients.

1.5 Méthode analytique de description de tâche (MAD)

Introduite par Scapin et Pierret-Goldbreich en 1989, MAD fut par la suite ajustée par Värnild (1993a) pour tenir compte de divers problèmes rencontrés par les analystes. Étonnamment, un document ultérieur, Sebillotte (1994), ne contient pas les rectifications auxquelles on vient de

référer. La version considérée pour @Esperanto est celle de Värnild (1993b) qui incorpore les modifications expliquées dans son premier document.

Dans sa structure, MAD diffère assez peu de l'AHT, puisque les deux méthodes sont basées sur une décomposition hiérarchique de la tâche. Dans sa forme, MAD diffère de l'AHT par l'accent mis sur la description de chaque nœud de l'arbre d'analyse, comme le démontre la quantité d'information à fournir pour chaque but ou tâche identifiés dans la décomposition hiérarchique

Un autre élément qui renforce cet accent mis sur les buts et les tâches est le fait qu'on ne voit que de très rares documents à propos de MAD qui présentent un arbre d'analyse global. On n'y retrouve que la décomposition d'une tâche en sous-tâches.[19]

En revanche, dans tous les documents décrivant MAD, on retrouve une figure ressemblant beaucoup à celle de la figure 1-7 qui indique clairement que, pour chaque nœud (but ou tâche), il faut entrer les informations suivantes :

- Des informations identifiant la tâche, incluant :
    - un nom et un numéro;
    - son but (en langue naturelle pour le bénéfice de toutes les parties prenantes).
- Différentes informations spécifiant les conditions d'exécution de la tâche, incluant :
    - l'état initial, soit l'ensemble des objets représentant le monde dans lequel la tâche va s'exécuter et qu'elle pourra ou non modifier;
    - les préconditions, soit un ensemble de prédicats exprimant des contraintes sur les objets de l'état initial qui doivent être réalisés pour que la tâche puisse commencer;
    - le corps de la tâche, qui est une tâche élémentaire (donc, simple, indécomposable ou dont l'analyse est jugée suffisante) ou alors un ensemble de sous-tâches avec une structure indiquant comment exécuter lesdites sous-tâches. Cette structure devant être l'une des valeurs suivantes :

[19] Bien sûr, on se rend bien compte que l'arbre d'analyse complet peut être produit à partir de la définition de l'ensemble des tâches, mais il est tout de même révélateur qu'on n'en fasse pratiquement jamais mention dans les descriptions de la méthode.

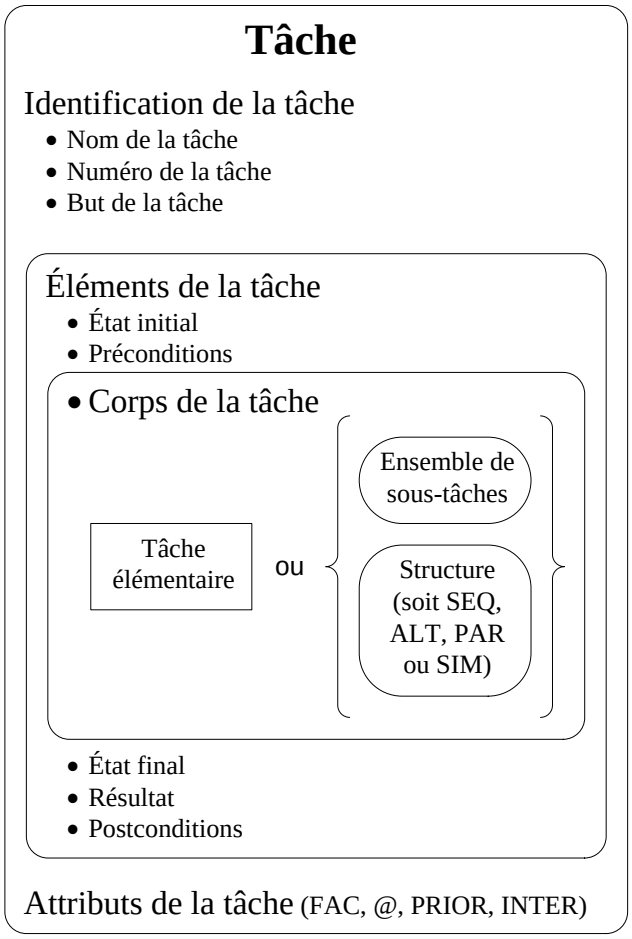


Figure 1-7 Description habituelle d'une tâche selon MAD.

- SEQ : si les sous-tâches doivent être exécutées séquentiellement;
- ALT : s'il faut ne choisir d'exécuter qu'une seule des sous-tâches;
- PAR : si les tâches doivent être exécutées en parallèle par une seule personne;
- SIM : si les tâches doivent être exécutées simultanément par plusieurs personnes;
- ET/OU : dans certains documents, trouvés au hasard de recherches sur Internet, la structure ET/OU est utilisée pour (vraisemblablement) indiquer que l'on peut exécuter autant de tâches que désiré ou nécessaire;

Dans la pratique, toutefois, la structure n'est pas qu'une seule de ces valeurs. Elle prend plutôt la forme d'un graphe dans lequel sont identifiés différents groupes, chacun étiqueté par l'une des valeurs ci-dessus. Un exemple, apparaissant un peu plus bas, rendra tout cela bien clair.

  - l'état final, soit l'ensemble des objets qui pourraient avoir été créés ou modifiés par la tâche, ainsi que ceux qui doivent exister après son exécution;
  - le résultat, soit un sous-ensemble strict de l'état final qui contient tout ce que la tâche a créé ou modifié;
  - les postconditions, soit un ensemble de prédicats exprimant des contraintes sur les objets de l'état final qui sont assurément réalisés lorsque la tâche se termine.
- Divers attributs affectant l'exécution de la tâche, incluant :
  - FAC : lorsque la tâche est facultative;
  - @ : lorsque la tâche peut être répétée;
  - PRIOR et INTER : les paramètres définissant l'interruptibilité d'une tâche (en spécifiant une priorité d'exécution qui sera comparée à celle des autres tâches voulant l'interrompre, ainsi que la manière dont l'exécution devra reprendre, s'il y a lieu, après l'interruption).

Dans la pratique, toutefois, ces attributs sont souvent utilisés dans la structure des sous-tâches où ils sont associés à autant de sous-tâches que nécessaire. Un exemple, apparaissant un peu plus bas, démontrera ce point.

La Figure 1-8 montre comment la tâche « Envoyer courriel » de l'exemple précédent pourrait être documentée selon MAD. On y voit un niveau de détail bien supérieur à ce que l'on a vu avec l'AHT, mais pour une (sous-)tâche à la fois. Chaque tâche peut donc être assimilée assez facilement et complètement au prix d'une plus grande difficulté à voir l'ensemble.[20] En ce qui concerne la décomposition de chaque tâche, on constatera qu'elle ne possède pas la puissance expressive des plans AHT, puisqu'il n'existe aucun énoncé conditionnel.

En terminant, il est intéressant de noter qu'il suffirait d'ajouter les attributs idoines à chaque tâche pour pouvoir utiliser l'AHT en lieu et place de MAD. C'est donc dire que l'AHT peut être adaptée facilement pour que ses capacités deviennent un surensemble strict de celles offertes par MAD.

---

[20] Avouons cependant qu'un logiciel pourrait fort bien assembler un arbre complet en se servant de toutes les décompositions élémentaires réparties dans chacune des tâches documentées.

0 -- Envoyer courriel

But: Envoyer un courriel du compte choisi avec possibilité d'ajouter une pièce jointe.

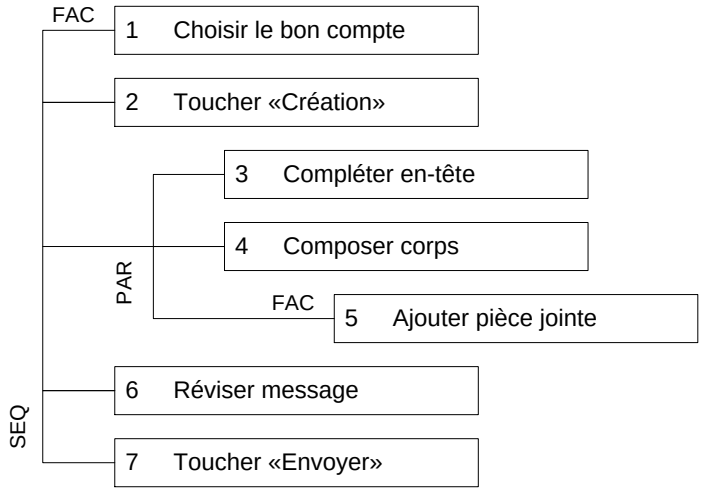
État initial

- Logiciel de gestion des courriels
- Fichier à joindre.

Préconditions

- Logiciel de gestion des courriels ouvert;
- Fichier à joindre disponible au système de gestion des fichiers de l'ordinateur.

Corps de la tâche



État final

- Logiciel de gestion des courriels;
- Courriel;
- Fichier à joindre;

Résultat

- Le courriel et, si désiré, le fichier à joindre sont envoyés.

Postconditions

- Courriel envoyé;
- Fichier à joindre envoyé avec le courriel si la tâche 5 a été effectuée.
- Logiciel de gestion des courriels dans le même état qu'avant la tâche.
- Fichier à joindre non modifié sur le disque dur;

<aucun attribut>

Figure 1-8 Représentation possible de la tâche «Envoyer courriel» du précédent exemple.

1.6 Analyse temporelle (AT)

Le but de l'analyse temporelle est de mettre en évidence une séquence d'opérations permettant d'atteindre un but donné et, le plus souvent, le temps d'exécution nécessaire non seulement pour l'ensemble de la séquence, mais aussi pour chacune des opérations la composant.

La manière de décomposer la tâche principale, donc celle devant réaliser le but précité, n'a pas d'influence sur cette méthode, puisqu'on ne cherche ici qu'à obtenir une liste d'opérations. L'utilisateur peut donc choisir une décomposition procédurale ou hiérarchique. Dans ce dernier cas, pour obtenir la liste recherchée, l'utilisateur devra isoler une ou plusieurs séquences qui lui paraissent d'intérêt.

Lorsque des temps de complétion sont notés, ceux-ci peuvent être exacts (p. ex., 15 minutes) ou être associés à une imprécision (p. ex., 15 ±2 minutes). Il en découle que le temps de complétion spécifié pour les opérations est exact; autrement, une imprécision devra lui être associée.[21] On notera que le mot « exact » ne s'applique pas ici à la sémantique (soit le temps requis par l'une ou l'autre des opérations), mais seulement à la syntaxe (donc, à la valeur entrée elle-même). Ainsi, dans un contexte de production, pour assurer que chaque opération pourra être complétée dans le temps spécifié, il est courant d'assigner à toute opération un temps de complétion sous la forme d'une valeur sans imprécision associée (syntaxiquement exacte) qui est en fait légèrement supérieure à ce qu'il est véritablement nécessaire (donc sémantiquement fausse).

Pour les situations où l'on ne recherche que la séquence d'opérations à réaliser, plusieurs représentations sont possibles, d'une simple liste ordonnée jusqu'à une représentation animée desdites opérations. Le problème de la représentation des résultats d'une AT est plus intéressant lorsque les contraintes temporelles (temps d'exécution, délais et interdépendances) doivent aussi être montrées. Peut-être pour cette raison, il existe beaucoup de représentations graphiques différentes pour une AT. Les plus utilisées sont illustrées un peu plus loin, mais avant on décrit les éléments d'une bonne représentation textuelle, parce qu'elle est très utile malgré qu'elle soit moins accrocheuse.

---

[21] L'imprécision associée au temps de complétion d'une séquence d'opérations complète étant la somme de chacune des imprécisions notée pour chaque opération.

1.6.1 Représentation textuelle

Les éléments que doivent montrer toutes bonnes représentations textuelles d'une analyse temporelle sont ceux énumérés un peu plus haut. On a donc la séquence numérotée d'étapes (sous-tâches) réalisant la tâche à analyser, ainsi qu'un commentaire optionnel et un temps de complétion pour chacune. Dans ce genre de représentation assez simple, on peut tenir compte de la présence de délais de réponse ou de périodes d'attente forcées soit en les incluant dans une étape, soit en créant des étapes artificielles pour les représenter.

Comme pour les autres méthodes, surtout avec une représentation tabulaire, pour chaque étape, il est possible d'ajouter autant d'attributs spécifiques à une étude que désiré. Un exemple d'une représentation textuelle avec des temps estimés est montré à la figure 1-9.

#	Étape	Commentaires	Temps (s)
1	Sélectionner le bon compte dans la liste des comptes connus.	À faire seulement si le bon compte n'est pas déjà choisi.	3
2	Cliquer le bouton «Création».		1
3	Taper l'adresse.	En moyenne, n_1 caractères. (Un claviste moyen tape un caractère à tous les 0.28 sec.)	$1 + (n_1 \times 0.28)$
4	[Déplacer le focus] Cliquer le champ «Objet» et taper l'objet du message.	En moyenne, n_2 caractères.	$1 + (n_2 \times 0.28)$
5	[Déplacer le focus] Cliquer le champ «Corps» et taper le corps du message.	En moyenne, n_3 caractères.	$1 + (n_3 \times 0.28)$
6	Cliquer le bouton «Ajout p.j.».		1
7	Sélectionner le bon type de fichier, dans la liste appropriée.	À faire seulement si le bon type n'est pas déjà choisi.	3
8	Cliquer le bouton «...» à côté du champ «Pièce jointe» et choisir le répertoire «Mes documents/Projet X» (le répertoire initial étant «Mes documents»).		3
9	Sélectionner le fichier à joindre.		2
10	Cliquer le bouton «OK».		1
11	Réviser le message.		$t_{Révision}$
12	Cliquer le bouton «Envoyer».		1
		Temps total (s) :	$18 + ((n_1+n_2+n_3) \times 0.28) + t_{Révision}$

Figure 1-9 Analyse temporelle d'une méthode réalisant l'envoi d'un courriel.

1.6.2 Représentations graphiques

Dans la représentation graphique la plus simple de l'analyse temporelle, toutes les opérations sont symbolisées par des rectangles dont la largeur est fonction du temps d'exécution que chacune requiert. Comme on peut le constater dans l'exemple de la figure 1-10, on suppose ici que toutes les opérations (obtenues de la décomposition hiérarchique de la figure 1-1) s'enchaînent les unes à la suite des autres sans autre interdépendance que la fin d'une opération pour commencer la suivante.

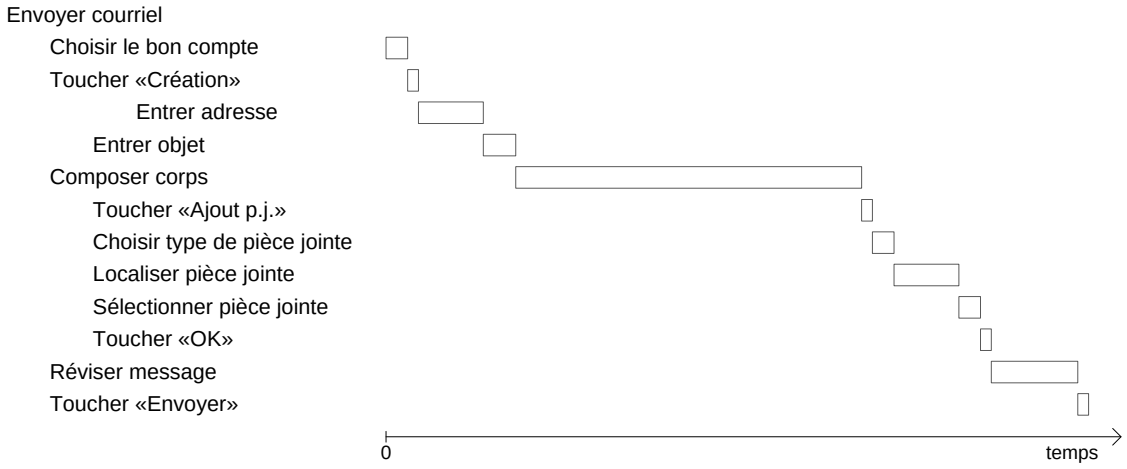


Figure 1-10 AT pour l'envoi d'un courriel (représentation de base).

La figure 1-11 montre la même information que la précédente, mais, ayant conservé les nœuds qui ne sont pas des feuilles dans la décomposition hiérarchique des tâches, cette représentation possède donc des mesures cumulatives sur les étapes collaborant à un même sous-but.

Remarquons qu'une AT est plutôt mal adaptée à la gestion de multiples chemins d'exécution.[22] C'est pourquoi, dans l'exemple précédent, nous avons dû faire le choix d'étudier un seul desdits chemins, notamment celui où il faut choisir le bon compte, où l'adresse doit être tapée (pas choisie parmi les contacts déjà enregistrés) et où il est nécessaire d'ajouter une pièce jointe.

---

[22] En fait, c'est le cas de toutes les méthodes dont le but est d'évaluer le temps nécessaire pour compléter une tâche, puisqu'alors il faudrait donner un temps de complétion pour chaque chemin d'exécution possible.

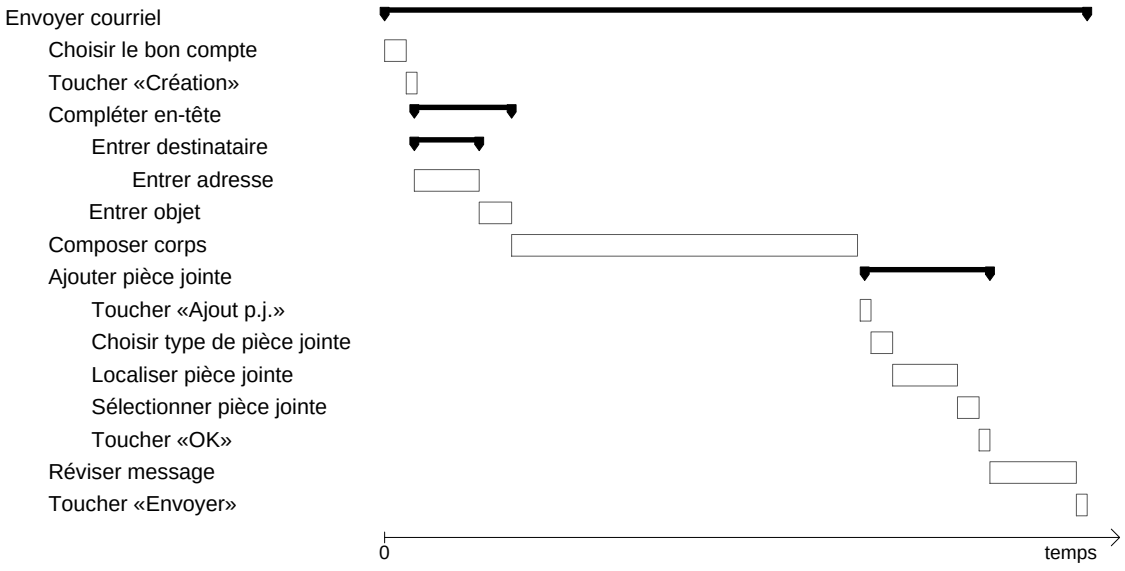


Figure 1-11 AT pour l'envoi d'un courriel (avec cumulatifs).

Pour réaliser une étude complète, il pourrait être nécessaire d'étudier chaque chemin d'exécution. Dans les cas réalistes, le nombre de tels chemins devient tellement grand que l'analyste n'a d'autres choix que d'arrêter son choix sur une sélection représentative et de justifier ses décisions.

Pour en revenir aux représentations, une version linéaire de la représentation précédente est envisageable, mais comme on le constatera en regardant la figure 1-12, cette version se prête mieux aux situations comportant moins de tâches distinctes.

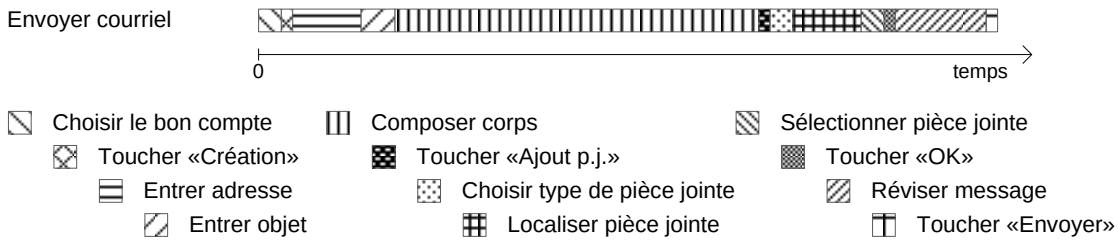


Figure 1-12 AT pour l'envoi d'un courriel (représentation linéaire horizontale simple).

Cette représentation peut être adaptée pour montrer les opérations qui peuvent se faire en parallèle ou simultanément[23] en empilant les blocs correspondant à celles-ci. Il est alors de la responsabilité de l'analyste de déterminer le temps qu'il faut allouer à la réalisation des opérations devant se faire en parallèle. Dans l'exemple ci-dessous, on indique qu'il est possible d'entrer en parallèle l'adresse, l'objet et le corps.

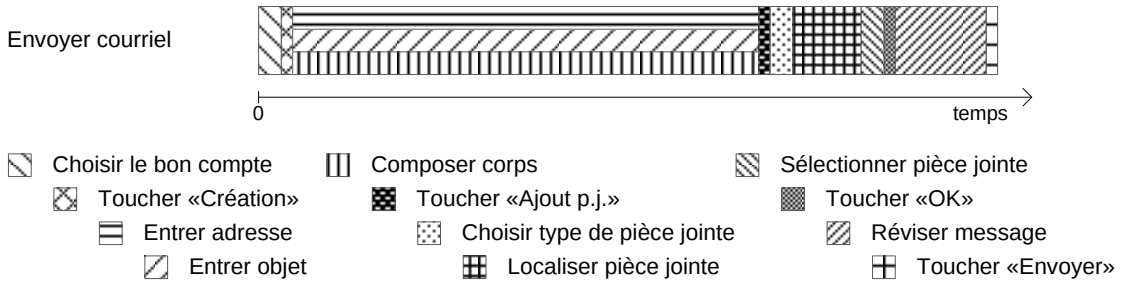


Figure 1-13 AT pour l'envoi d'un courriel (avec parallélisme).

Notons ici qu'il est possible d'utiliser l'AT pour étudier les tâches d'une équipe. Pour cela, il suffit d'utiliser une bande de blocs de tâches par membre de l'équipe, chacune alignée sur la même abscisse pour faciliter la visualisation de la synchronisation entre les tâches de chacun.

L'AT est souvent utilisée dans des études où la charge mentale de travail (CMT) est un paramètre tout aussi important que le temps d'exécution. Dans une telle situation, la bande de blocs de tâche peut être surmontée d'une courbe (ou d'une suite de segments) pour donner une indication de la variation relative de la CMT en fonction des tâches à effectuer ou du temps. Ainsi, on pourrait avoir une représentation comme celle de la Figure 1-14.

Dans les situations où l'on voudrait montrer une mesure de la CMT, plutôt qu'une simple indication relative, il serait nécessaire de remplacer la simple ligne discontinue par un petit graphique. Ce graphique montrerait alors les valeurs possibles de la CMT en ordonnée.

---

[23] Les tâches en parallèle sont réalisées par un opérateur qui va devoir gérer son temps de manière à toutes les accomplir. Le plus souvent, les tâches qui doivent être réalisées simultanément le sont par autant de personnes qu'il y a de tâches.

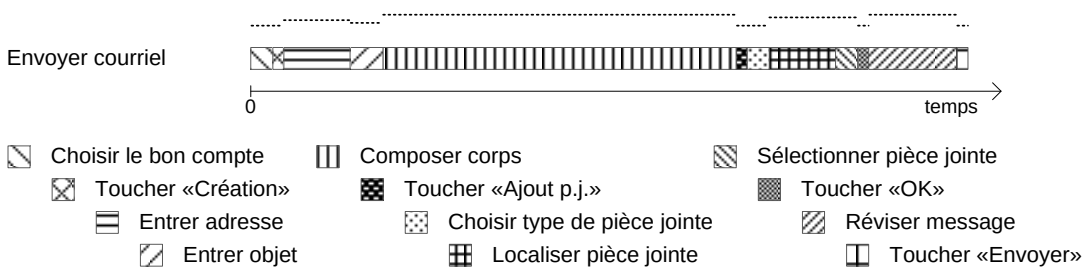


Figure 1-14 AT pour l'envoi d'un courriel (avec indication relative de la CMT).

Finalement, en tournant la bande de blocs de tâches à 90° dans le sens horaire, il est possible de se débarrasser de la légende pour simplement identifier chaque bloc à l'aide d'un texte apparaissant à la droite de ce qui est maintenant la colonne qui représente la séquence des opérations à réaliser. La figure 1-15 démontre aussi la possibilité d'ajouter d'autres informations (p. ex. des commentaires) au schéma résultant.

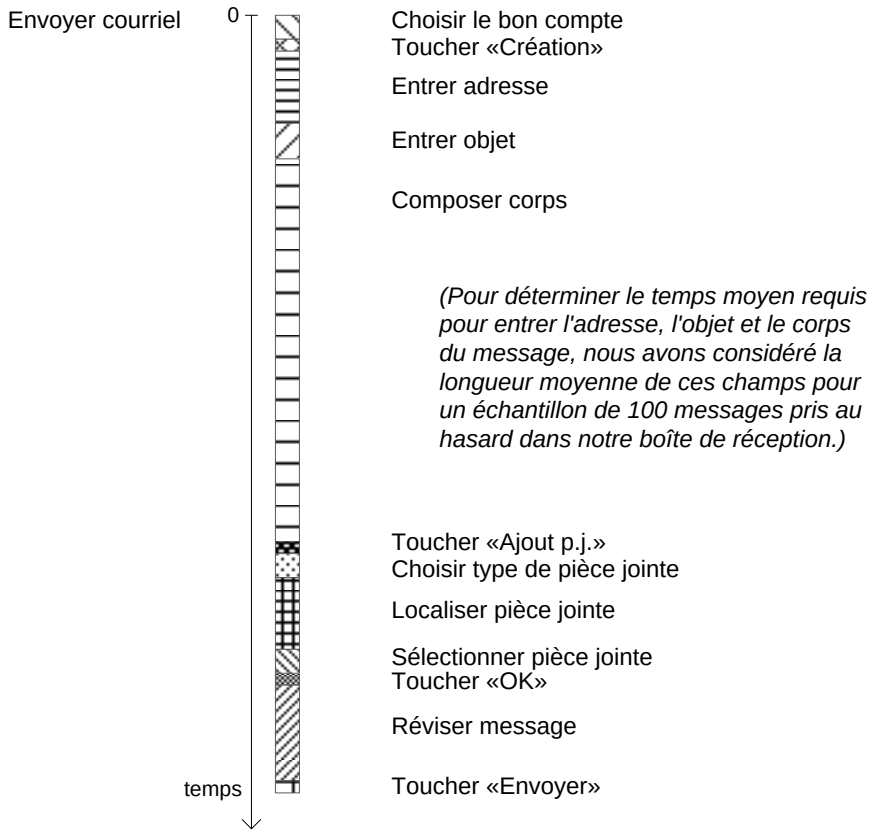


Figure 1-15 AT pour l'envoi d'un courriel (représentation verticale).

Certains lecteurs se demandent probablement quand on en viendra au diagramme de Gantt, mieux connu des gestionnaires de projets. Ce type de diagramme ressemblant assez à la version élémentaire montré à la figure 1-10, ne sera pas abordé parce que trop complexe pour le type d'études dont il est question ici. En effet, en plus de ce qu'on a déjà vu, le diagramme de Gantt permet d'établir différents types de relations entre les tâches (fin-début, début-début, début-fin et fin-fin) et plusieurs types de contraintes sur chaque tâche (commencer dès que possible, finir avant un temps donné, etc.), ce qui peut certes être très utile en gestion de projet avec son besoin de gérer mille et un aléas, mais pas en analyse de tâche où tout doit être connu.

Tout cela étant dit, il existe bon nombre d'autres représentations possibles pour les résultats d'une AT, mais elles ne seront pas considérées dans le cadre de ce projet.

1.7 Goals, Operators, Methods, and Selection Rules (GOMS)

Créée par Card, Moran et Newell, GOMS est une suite de modèles de complexité croissante pour l'analyse de tâches exécutées de manière experte. Ce dernier point implique que les utilisateurs maîtrisent à ce point les tâches étudiées qu'ils les font de manière automatique, sans aucun besoin de réfléchir à ce qui doit être fait ou pourquoi. On peut donc inférer de cela que le but à atteindre est tout aussi expertement connu des utilisateurs qui nous intéressent ici.

Card, Moran et Newell (1983) ont détaillé leur méthode en fonction de la granularité d'analyse jugée nécessaire par l'analyste. Ainsi, ils ont défini les modèles suivants en utilisant comme exemples différentes tâches à effectuer avec un éditeur de texte de l'époque :

- Un modèle de haut niveau baptisé `UT` (pour « *Unit-task level* ») dans lequel on ne fait aucune décomposition des tâches déclarées « unitaires » par l'analyste;
- Deux modèles du niveau fonctionnel baptisés `F1` et `F2` (pour « *Functional level* ») dans lesquels on se contente d'une décomposition minimale servant à mettre en évidence les fonctions à réaliser pour atteindre le but fixé. Le modèle `F2` diffère de l'autre par le fait que l'on tient compte du contexte d'exécution pour spécifier la méthode la plus efficace pour atteindre le but.
- Quatre modèles, d'un niveau que l'on pourrait qualifier de descriptif, baptisés `A1` à `A4` (pour « *Argument level* ») dans lesquels les opérations à exécuter sont spécifiées avec de plus en plus de détails.
- Finalement, deux modèles de niveau opérationnel baptisés K1 et K2 (pour « keystroke level », donc au niveau des actions élémentaires, puisque les outils de travail sont un clavier et une souris). À ce niveau, on décompose la tâche à réaliser jusqu'au niveau des

> actions les plus élémentaires, incluant le déplacement de la main du clavier à la souris ou le fait de presser un bouton de la souris. Le niveau `K2` diffère de l'autre par le fait que l'on indique avec plus de détails les opérations mentales que doivent effectuer les utilisateurs.

Parce que la granularité de l'analyse n'affecte pas fondamentalement une méthode, on ne connaît maintenant de ces modèles que leur essence distillée dans quelques variantes de la méthode, incluant CMN-GOMS qui couvrent les trois premiers niveaux ci-dessus et KLM-GOMS qui correspond au dernier niveau.

John (2003) explique quatre variantes de la méthode GOMS prenant toutes comme fondement le modèle « Human Information Processing (HIP)[24] » qu'introduisirent Card, Moran et Newell au début des années 1980. Les sections suivantes présentent les deux modèles qu'@Esperanto met en œuvre, soit KLM-GOMS et CMN-GOMS.

Pour curiosité, on notera que les deux versions de GOMS retenues pour @Esperanto sont basées sur une version sérielle du modèle que l'on vient de mentionner. On y suppose donc l'existence d'un seul processeur pour l'exécution des différentes étapes de traitement de l'information (perception, décision et exécution), ce qui est suffisant pour les situations simples. Pour des situations plus complexes, CPM-GOMS pourrait être mieux adapté puisque basé sur le « Model Human Processor (MHP)[25] », une version du modèle HIP qui suppose l'existence de plusieurs processeurs plus ou moins indépendants pour l'exécution de diverses étapes du traitement de l'information.

---

[24] En français, ce serait quelque chose comme le modèle de « Traitement humain de l'information ». C'est un modèle dans lequel on compare à celui d'un ordinateur le traitement que font les humains des stimuli d'entrée (tout ce qui peut être perçu par nos sens) et la manière dont ils décident des réponses. L'utilisation de ce modèle a permis de déterminer des moyennes concernant (entre autres) les temps de réaction à divers stimuli, des temps de traitement de l'information selon la nécessité d'accéder à diverses informations en mémoire de travail ou à long terme et des temps de production d'une réponse.

[25] En français, ce serait quelque chose comme « Processeur humain typique ».

1.8 KLM-GOMS

La première chose à faire pour une analyse KLM-GOMS est de déterminer une méthode qui permet de réaliser la tâche étudiée. Cette méthode peut contenir plusieurs étapes suffisamment simples qui, elles, seront décomposées en opérateurs. Ces opérateurs, identifiés un peu plus bas, correspondent à des actions élémentaires du travail à l'ordinateur (des années 80's), telles que la frappe d'une touche du clavier d'un ordinateur, comme l'indique le nom de cette variante, soit « Keystroke-Level Model »[26].

Pour les situations où les experts du domaine ne s'entendent pas quant à la méthode à utiliser pour réaliser une tâche, @Esperanto permet d'en assigner autant qu'il est nécessaire. Lorsque plus d'une méthode est étudiée, l'utilisateur est encouragé à décrire l'utilité ou la provenance de chacune.

Les opérateurs GOMS acceptés par @Esperanto sont essentiellement ceux définis par Card, Moran et Newell pour cette méthode d'analyse. Au départ, six opérateurs ont été définis, mais un autre fut ajouté un peu plus tard pour tenir compte du temps de réponse de l'application permettant ainsi plus de cohérence entre les temps modélisés et ceux mesurés avec les systèmes réels. On a donc les sept opérateurs montrés au tableau 1-3 avec leur temps d'exécution nominal ou calculé en utilisant la loi de Fitts[27]. Remarquons ici qu'il semble que certains analystes utilisent le même opérateur pour coder la pression d'une touche du clavier aussi bien que le clic d'un bouton de la souris. Parce que faire cela rend difficile de tenir compte de la vitesse à laquelle tape la personne étudiée, on préférera l'utilisation d'opérateurs différents pour ces actions.

Par la suite, différentes interfaces ayant été introduites, d'autres chercheurs suggérèrent d'autres opérateurs pour tenir compte des nouvelles interactions possibles. Par exemple, pour leurs travaux concernant les interfaces offertes sur un téléphone mobile, Holleis, Otto, Hussmann et Schmidt (2007) ont proposé les nouveaux opérateurs montrés au tableau 1-4 (pour lesquels aucun temps nominal n'est fourni).

---

[26] En français, ce pourrait être traduit par « Modèle au niveau des frappes [sur le clavier] ».

[27] La loi de Fitts sert à calculer le temps requis pour atteindre une cible en fonction de sa grosseur (dans l'axe du déplacement) et de sa distance.

Tableau 1-3 Opérateurs KLM-GOMS couramment utilisés.

Opérateur	**$t_{Nominal}$ (s)**	**Description**
K	0.28	Pression d'une touche du clavier par une personne d'habileté moyenne qui peut taper 40 mots par minute (des temps ont été déterminés pour d'autres niveaux d'expérience).
C	0.20	Pression (appuyer et relâcher) d'un bouton de la souris.
P	1.10 ou Fitts	Amener le pointeur de la souris sur une cible à l'écran.
H	0.40	Déplacer les mains sur le clavier ou une main sur la souris.
D	Fitts	Dessiner un segment de ligne sur une grille.
M	1.35	Préparation mentale avant l'exécution d'une action ou de plusieurs actions élémentaires sémantiquement reliées. Quelques règles ont été élaborées pour aider les analystes à placer correctement cet opérateur très difficilement observable.
R*(t)*	Au cas par cas.	Temps de réponse du système (pendant lequel l'utilisateur pourrait devoir attendre).

Tableau 1-4 Opérateurs KLM-GOMS ajoutés pour l'étude d'IHM sur téléphones mobiles.

Opérateur	**Description**
S_{Macro}	Déplacement de l'attention du sujet de son téléphone au monde environnant.
S_{Micro}	Déplacement de l'attention du sujet d'une partie de son téléphone à une autre.
X	Distraction due à l'environnement du sujet.
A*(t)*	Action complexe difficilement réductible à une séquence des autres opérateurs, p. ex., effectuer le cadrage pour une photo ou toucher une étiquette IDRF (pour Identification par radiofréquence (en anglais, RFID)).
G	Geste correspondant à une commande pour le téléphone (p. ex., dessin d'un chiffre ou rotation du téléphone dans l'espace).
F	Geste fait avec un doigt sur l'écran tactile du téléphone.
I	Actions initiales, soient celles nécessaires pour amener le téléphone dans une position d'où le sujet pourra l'utiliser.

On remarquera qu'il semble difficile d'assigner un temps d'exécution nominal pour beaucoup de ces nouveaux opérateurs. Il faut plutôt procéder au cas par cas.

Avec les différentes avancées théoriques en ergonomie cognitive, d'autres modifications ont été suggérées, incluant diverses modulations des opérateurs pour des prédictions de temps plus précises, l'à-propos d'utiliser des grammaires (comme les *Task-Action Grammars*) ou des systèmes de production pour la description de tâches, la possibilité de tenir compte de l'apprentissage ou de l'exécution de tâches en parallèle. Pour une introduction à ces avancées, voir Olson et Olson (1990).

Malgré tout cela, il est intéressant de noter que les sept opérateurs initiaux sont encore suffisants pour décrire la plupart des interfaces actuelles. Par exemple, déplacer le doigt sur l'écran correspond plus ou moins à un opérateur de type glisser-déposer, constitué d'une pression sur un bouton de la souris, d'un déplacement du pointeur et enfin d'un relâchement du bouton de la souris, le tout assez semblable aux opérateurs C et D.

Par exemple, le cas traité précédemment par une analyse temporelle pourrait être analysé par KLM-GOMS comme montré à la figure 1-16. Dans cet exemple, en plus d'une grande similarité avec ce qui a été fait par analyse temporelle, on notera deux simplifications qui n'invalident en rien l'analyse KLM-GOMS : l'emploi de temps nominaux plutôt que d'une fonction calculant le temps de déplacement en accord avec la loi de Fitts (ce qui a toutefois tendance à surestimer le temps de complétion) et la supposition que l'on a affaire à un dactylo d'habileté moyenne (d'où le temps nominal associé à l'opérateur K).

Au passage, notons que le temps requis pour taper l'adresse courriel, l'objet du message et son corps ne serait aucunement considéré dans le cadre d'une étude dont le but est de comparer des interfaces nécessitant toutes de taper ces informations. De même, peu importe l'interface, le temps de révision ne peut être étudié par KLM-GOMS et ne serait considéré que si l'on cherchait à estimer le temps requis pour écrire et envoyer un message pour, par exemple, estimer un nombre raisonnable de messages qu'un employé pourrait envoyer durant une journée de travail.

En terminant avec KLM-GOMS, notons qu'elle n'a aucune représentation graphique.

#	Étapes	Opérateurs	Temps (s)	Sommation
1	Sélectionner le bon compte dans la liste des comptes connus.	M H[souris] PC [ouvrir liste] PC [pointer compte; sélect.]	1.35 + 0.40 + 1.10 + 0.20 1.10 + 0.20	4.35
2	Cliquer le bouton «Création».	PC [bouton]	1.10 + 0.20	1.30
3	[Focus déjà sur champ «Adresse»] taper l'adresse.	M H[clavier] K(n_1)	1.35 + 0.40 + n_1 x 0.28	1.75 + (n_1 x 0.28)
4	[Déplacer le focus] Cliquer le champ «Objet» et taper l'objet du message.	M H[souris] PC [champ «Objet»] H[clavier] K(n_2)	1.35 + 0.40 + 1.10 + 0.20 + 0.40 + n_2 x 0.28	3.45 + (n_2 x 0.28)
5	[Déplacer le focus] Cliquer le champ «Corps» et taper le corps du message.	M H[souris] PC [champ «Corps»] H[clavier] K(n_3)	1.35 + 0.40 + 1.10 + 0.20 + 0.40 + n_3 x 0.28	3.45 + (n_3 x 0.28)
6	Cliquer le bouton «Ajout p.j.».	M H[souris] PC [bouton]	1.35 + 0.40 + 1.10 + 0.20	3.05
7	Sélectionner le bon type de fichier, dans la liste appropriée.	PC [ouvrir liste] PC [pointer type de fichier; sélect.]	1.10 + 0.20 + 1.10 + 0.20	2.60
8	Cliquer le bouton «...» à côté du champ «Pièce jointe» et choisir le répertoire «Mes documents/Projet X» (le répertoire initial étant «Mes documents»).	PC [bouton] PC [répertoire «Projet X»] PC [sélect. bon fichier]	1.10 + 0.20 + 1.10 + 0.20 + 1.10 + 0.20	3.90
9	Sélectionner le fichier à joindre.	PC [ouvrir liste] PC [pointer fichier; sélect.]	1.10 + 0.20 + 1.10 + 0.20	2.60
10	Cliquer le bouton «OK».	PC [bouton]	1.10 + 0.20	1.30
11	Réviser le message.	[aucune manipulation]	$t_{Révision}$	$t_{Révision}$
12	Cliquer le bouton «Envoyer».	M PC [bouton]	1.35 + 1.10 + 0.20	2.65
		Temps total (s) :	30.40 + ((n_1 + n_2 + n_3) x 0.28) + $t_{Révision}$	

Figure 1-16 KLM-GOMS pour une méthode réalisant l'envoi d'un courriel.

1.9 CMN-GOMS

Selon John (2003), CMN-GOMS est la première variante de GOMS que présentèrent Card, Moran et Newell au début des années 1980.

Première différence par rapport à KLM-GOMS, cette méthode, dont les trois premières lettres rappellent ses auteurs, permet de spécifier le but à atteindre et de décomposer récursivement celui-ci en sous-buts, jusqu'à ce que chaque sous-but dans la hiérarchie ainsi créée soit suffisamment simple pour pouvoir indiquer une séquence d'opérateurs permettant de le réaliser. Une autre manière d'y penser est d'imaginer un arbre de buts et sous-buts dont la racine est le but à atteindre et où les feuilles sont des séquences d'opérateurs KLM-GOMS.

À l'instar de KLM-GOMS, il est possible de définir plusieurs méthodes pour réaliser un même (sous-)but. Dans ces situations, il est nécessaire de définir aussi une sorte de sélecteur montrant comment choisir la meilleure méthode en fonction du contexte d'exécution.

Dans cette variante, l'acronyme GOMS prend tout son sens puisqu'on y retrouve des buts (***G****oals*), des opérateurs (***O****perators*), des méthodes (***M****ethods*) qui sont des séquences d'opérateurs réalisant un but et des règles de sélection (***S****election rules*) pour indiquer comment choisir la méthode la plus adaptée au contexte courant d'exécution.

Dans l'exemple du figure 1-17, on constatera que les méthodes apparaissent sous la forme de véritables séquences d'opérateurs KLM-GOMS, la plupart ayant été copiées directement de l'exemple en KLM-GOMS. Un examen plus poussé révélera aussi un détail navrant concernant la spécification des endroits où indiquer qu'il doit y avoir déplacement d'une main du clavier à la souris ou vice-versa. En effet, selon la méthode utilisée pour le but « Entrer destinataire », on arrive au but « Entrer objet » soit avec la main sur la souris, soit sur le clavier. Comment savoir alors s'il faut ou non indiquer un déplacement de la main vers la souris?

Peut-être à cause des difficultés de cette nature, les exemples de CMN-GOMS que l'on peut trouver dans la littérature utilisent plutôt des fonctions de base auxquelles on assigne les opérateurs GOMS les réalisant, sans se préoccuper du contexte extérieur de chaque méthode. La figure 1-18 montre un tel exemple.

BUT : Envoyer courriel			
BUT : Ouvrir fenêtre de création			
Sélectionner le bon compte, dans la liste de comptes connus.	M H[souris] PC [ouvrir la liste] PC [pointer compte; sélect.]	1.35 + 0.40 + 1.10 + 0.20 1.10 + 0.20	4.35
Cliquer le bouton «Création».	PC [bouton]	1.10 + 0.20	1.30
BUT : Compléter en-tête			
BUT : Entrer destinataire			
CHOISIR :			
MÉTHODE : Utiliser liste des contacts	...si le destinataire est dans la liste des contacts		
Sélectionner le destinataire dans la liste des contacts et accepter.	M PC [ouvrir la liste] PC [pointer le dest.; sélect.] PC [bouton OK]	1.35 + 1.10 + 0.20 + 1.10 + 0.20 + 1.10 + 0.20	3.95
MÉTHODE : Entrer adresse	…sinon (le destinataire n'est pas dans la liste des contacts)		
Le focus étant déjà dans le champ «Adresse», taper l'adresse du destinataire.	M H[clavier] K(n_1)	1.35 + 0.40 + n_1 x 0.28	1.75 + (n_1 x 0.28)
BUT : Entrer objet			
[Déplacer le focus] Cliquer le champ «Objet» et taper l'objet du message.	M H[souris] PC [champ «Objet»] H[clavier] K(n_2)	1.35 + 0.40 + 1.10 + 0.20 + 0.40 + n_2 x 0.28	3.45 + (n_2 x 0.28)
BUT : Composer corps			
Voir l'étape #5 de l'exemple en KLM-GOMS.			3.45 + (n_3 x 0.28)
BUT : Ajouter pièce jointe			
Voir les étapes #6 à #10 de l'exemple en KLM-GOMS.			13.45
BUT : Envoyer message et fermer fenêtre			
Voir les étapes #11 et #12 de l'exemple en KLM-GOMS.			2.65

Figure 1-17 CMN-GOMS pour l'envoi d'un courriel (avec méthodes en KLM-GOMS).

BUT : Envoyer courriel			
BUT : Ouvrir fenêtre de création			
PRÉPARATION	M	1.35 +	5.35
CLIQUER-LISTE-DES-COMPTES	PK	1.10 + 0.20 +	
CLIQUER-COMPTE-VOULU	PK	1.10 + 0.20 +	
CLIQUER-BOUTON-CRÉATION	PK	1.10 + 0.20	
BUT : Compléter en-tête			
BUT : Entrer destinataire			
CHOISIR :			
MÉTHODE : Utiliser liste des contacts ...si le destinataire est dans la liste des contacts			
PRÉPARATION	M	1.35 +	3.95
CLIQUER-LISTE-DES-DESTINATAIRES	PK	1.10 + 0.20 +	
CLIQUER-DESTINATAIRE-VOULU	PK	1.10 + 0.20 +	
CLIQUER-BOUTON-OK	PK	1.10 + 0.20	
MÉTHODE : Entrer adresse …sinon (le destinataire n'est pas dans la liste des contacts)			
PRÉPARATION	M	1.35 +	1.35 +
TAPER-ADRESSE	K(n_1)	n_1 x 0.28	(n_1 x 0.28)
BUT : Entrer objet			
PRÉPARATION	M	1.35 +	3.05 +
CLIQUER-CHAMP-OBJET	PK	1.10 + 0.20 +	(n_2 x 0.28)
TAPER-OBJET	HK(n_2)	0.40 + n_2 x 0.28	
BUT : Composer corps			
...			
BUT : Ajouter pièce jointe			
...			
BUT : Envoyer message et fermer fenêtre			
...			

Figure 1-18 CMN-GOMS pour l'envoi d'un courriel (comme vu dans la littérature).

1.10 Synthèse

Dans ce chapitre, on a abordé les différentes méthodes d'analyse de tâche qu'@Esperanto doit supporter en commençant par leurs points communs, puis en présentant chacune d'elle avec leurs possibilités respectives et leurs représentations usuelles autant textuelles que graphiques. Un récapitulatif de leurs différences et similitudes apparaît au début du chapitre suivant.

On a aussi présenté les réseaux et leurs utilités en analyse de tâche, simplement pour indiquer qu'une version future d'@Esperanto devra les supporter.

CHAPITRE 2 NORMALISATION DES MÉTHODES D'ANALYSE DE TÂCHE

Dans ce chapitre, après avoir récapitulé les différences entre les méthodes d'analyse de tâche qu'@Esperanto doit supporter, on s'attaque à la normalisation de leur puissance expressive. C'est donc dire que l'on veut réduire les différences d'expressivité entre les méthodes pour éviter le plus possible d'avoir à gérer les particularités de chacune. Par exemple, si le parallélisme entre des tâches ne restait exprimable qu'avec une méthode sur deux, cela nécessiterait qu'@Esperanto ne permette l'utilisation de ce concept qu'avec certaines méthodes.

2.1 Récapitulatif des différences entre les méthodes d'AT

Comme il a été montré au chapitre précédent, les méthodes d'analyse supportées par @Esperanto présentent plusieurs différences, certaines en sont des caractéristiques (marquées par des « ✻ ») d'autres sont plutôt liées à leur puissance expressive (marquées par « oui » ou par « non » selon leurs capacités). Le tableau 2-1 met tout cela en évidence.

2.2 Méthode native et méthode courante

Parce que le but avoué d'@Esperanto est de permettre aux analystes de tâche de changer de méthode à n'importe quel point d'une analyse de tâche, il est nécessaire de distinguer deux concepts.

- Une méthode sera dite **native** pour une tâche si toutes les informations attendues pour cette dernière ont bien été entrées et validées. Conséquemment, toute tâche doit avoir au moins une méthode native, mais pourrait en avoir plusieurs si l'analyste a jugé utile d'entrer les informations requises par plus d'une méthode.[28]
- La méthode dite **courante** désigne la méthode selon laquelle l'analyste veut entrer de nouvelles informations, ce qui peut ou non être l'une des méthodes natives de la tâche à modifier.
  Pour éviter toute ambiguïté, à chaque instant, il ne peut y avoir qu'une seule méthode courante en fonction de laquelle toute modification à une tâche est interprétée. Cette

[28] Comme on l'a vu au chapitre précédent, documenter une même tâche selon les préceptes de plusieurs méthodes peut mener à une meilleure compréhension de celle-ci.

contrainte d'unicité pourra être allégée en changeant automatiquement la méthode courante pour la méthode native de la tâche à modifier pour autant que celle-ci n'en ait qu'une (ce qui sera vraisemblablement le cas pour la grande majorité des tâches).

Pour l'instant, on mentionnera simplement que la notation employée changera localement en fonction de la méthode d'analyse utilisée pour décrire chaque tâche. L'explication détaillée de ces changements et du traitement des tâches ayant plus d'une méthode native est reportée au chapitre suivant.

Tableau 2-1 Caractéristiques des méthodes supportées par @Esperanto.

	AHT	**OTHI**	**MAD**	**Anal. temp.**	**KLM-GOMS**	**CMN-GOMS**
Aspects sur lesquels l'accent est mis						
Structure de la tâche	✲		✲			✲
Décisions à prendre		✲				
Ordre d'exécution		✲		✲	✲	
Temps d'exécution				✲	✲	✲
Décomposition de la tâche						
Procédurale		✲		✲	✲	
Hiérarchique	✲		✲	✲		✲
Enchaînements d'énoncés exprimables						
Séquence	oui	oui	oui	oui	oui	oui
Option unique	oui	non	oui	non	non	non
Options multiples	oui	non	oui	non	non	non
Branchement conditionnel	oui	oui	oui	non	non	non [a]
Boucle (nb itérations quelconque)	oui	oui	oui	non	non	non
Parallélisme	oui [b]	non	oui	oui	non	non
Simultanéité	oui [b]	non	oui	non	non	non
Attributs exprimables d'une entité						
Facultativité	oui	non [c]	oui	oui [d]	non	non [e]
Interruptibilité	non [f]	non	oui	non	non	non
Validation						
Attributs des tâches	✲	✲	✲	✲	✲	✲
Relations entre les attributs			✲ [g]			

[a] Les règles de sélection pourraient être vues comme étant des branchements conditionnels, mais leur but est d'indiquer, parmi les méthodes disponibles pour réaliser une tâche, celle qui est la plus appropriée. Il ne s'agit donc pas d'une réelle modification au chemin d'exécution en fonction du contexte, puisque la prochaine tâche à effectuer n'est pas affectée par une règle de sélection.

[b] En AHT, il est très facile d'indiquer si des tâches doivent être réalisées en parallèle ou en simultané, mais certaines mises en œuvre (celle de TaskArchitect©, par exemple) ne font pas de distinction entre ces deux concepts.

[c] Aucune opération n'est facultative en ce sens qu'il faut impérativement exécuter toutes les opérations sur le chemin d'exécution résultant de l'évaluation des diverses conditions de l'OTHI.

[d] Les chemins d'exécution facultatifs sont facilement gérables lorsqu'on est seulement concerné par l'ordre dans lequel il faut accomplir les différentes opérations. Lorsque le temps d'exécution est important, il est vraisemblable que chaque chemin d'exécution va différer des autres sur cet aspect. Bien que ce ne soit pas ingérable, ce n'est certes pas évident.

[e] Les règles de sélection permettent de rendre optionnelles des séquences d'opérateurs, mais, comme avec KLM-GOMS, dans chaque séquence il ne peut y avoir d'opérations facultatives.

[f] Il serait certainement possible d'indiquer (en langage naturel) qu'une tâche est interruptible, mais une telle description ne saurait être comparée à la description formelle rendue possible par MAD.

[g] Il faut valider les relations entre les attributs suivants : état initial, précondition, état final, postcondition et résultat.

2.3 À propos de la normalisation des méthodes d'analyse de tâche

Toujours en raison de son but de permettre un changement de méthode à n'importe quel point d'une analyse de tâche, @Esperanto se doit d'utiliser au mieux les données entrées par les analystes sans égard à la méthode courante au moment de la saisie. Pour cela, la base commune à toutes les méthodes supportées devrait être la plus grande possible.

Dès le départ, on réalise que toutes les méthodes d'analyse de tâches ont un point en commun : elles servent à décrire des tâches! C'est un bon début, mais en y regardant de plus près, on réalise vite qu'au-delà d'un noyau commun d'information à propos desdites tâches, chaque méthode diffère non seulement dans les caractéristiques documentées, mais aussi dans la manière de documenter certaines caractéristiques essentiellement identiques (les directives d'exécution, notamment).

On réalise aussi que les méthodes que doit supporter @Esperanto peuvent être rangées dans l'une des deux grandes classes de méthodes d'analyse de tâche suivantes. D'une part, il y a celles

capables de gérer la complexité d'une tâche et des diverses situations dans lesquelles on peut devoir la réaliser (AHT, MAD et OTHI), de l'autre, celles portant sur les menus détails d'exécution dans des circonstances bien déterminées (analyse temporelle et KLM-GOMS). On verra plus loin que CMN-GOMS est un cas particulier qui peut être facilement traité par @Esperanto.[29]

La première chose à faire est donc d'établir une liste des caractéristiques d'intérêt pour chaque méthode d'analyse de tâches, puis d'augmenter le langage de chacune de manière à leur permettre d'exprimer un maximum des concepts des autres méthodes. Il est important de noter que cela n'est pas fait pour changer l'esprit de quelque méthode que ce soit, mais simplement pour permettre à chacune de bien représenter un maximum des concepts exprimables par une autre méthode. Cette opération d'élargissement de la base commune à toutes les méthodes supportées est la première étape obligée qui permettra de normaliser le langage de toutes les méthodes.

À nouveau, il est important de clarifier que la normalisation proposée ne vise pas à rendre toutes les méthodes semblables en les privant de leurs spécificités. En particulier, il serait exagéré de modifier les méthodes d'analyse de tâche au point que les deux grandes classes partagent le même langage. Il s'agit seulement d'augmenter le vocabulaire propre à chacune pour leur permettre de formuler autant de concepts que raisonnable[30] à propos des tâches. Cette extension du vocabulaire des méthodes sera donc faite en respectant les spécificités de chacune.

Considérons, par exemple, d'un côté AHT et MAD qui sont des méthodes mettant l'accent sur la structure des tâches nécessaires à la réalisation d'un but, et de l'autre l'analyse temporelle qui est plutôt concernée par l'ordre et le temps d'exécution des opérations d'un processus. Bien qu'il serait fort possible d'ajouter aux tâches des premières méthodes assez de propriétés (attributs) pour aussi noter les informations relatives à l'ordre d'exécution et au temps d'exécution de

---

[29] Pour l'instant, il n'est pas important d'assigner CMN-GOMS à une classe en particulier.

[30] Par conséquent, en tenant compte de leur classe de méthodes d'analyse de tâche.

chacune, aucune tentative en ce sens n'a été effectuée. On rappelle encore une fois qu'il ne s'agit pas de dénaturer les méthodes existantes non plus que de créer une super méthode.[31]

C'est donc pour ne pas dénaturer les méthodes d'analyse supportées que les additions ou modifications proposées pour chacune se limiteront aux seuls éléments du tableau 2-1 qui comportent un « N ». Le but de la normalisation étant d'uniformiser les possibilités des méthodes à l'intérieur de leur classe.

Les caractéristiques marquées par des « ✳ » étant indissociablement liées à l'identité même des méthodes d'analyse de tâche se doivent d'être intégralement préservées.

2.4 Avant de continuer : un petit secret… Ce sont toutes des AHT!

Comme mentionné lors de sa présentation au chapitre précédent, l'AHT permet d'exprimer les agencements de tâches les plus complexes de manière relativement simple, ce qui est en grande partie la raison de son succès durable. De plus, comme on l'a vu au chapitre précédent, il est toujours possible d'associer autant d'attributs que nécessaire aux tâches d'une AHT pour mémoriser toutes informations jugées utiles.[32]

Grâce à ces deux caractéristiques, il est possible d'exprimer en AHT ce qui peut l'être par les autres méthodes d'analyse de tâche supportées par @Esperanto. Démonstration…

MAD : Plusieurs des exemples d'utilisation de MAD trouvés au cours des recherches exploratoires utilisent des variantes plus ou moins compréhensibles pour noter la structure du corps d'une tâche non élémentaire. Ceci peut être expliqué en bonne partie par le fait qu'il n'existe aucune description satisfaisante de cet aspect de MAD, car la description de la méthode laisse penser qu'il faudrait se contenter de n'utiliser qu'une instruction d'enchaînement des sous-tâches parmi « SEQ », « ALT », « PAR » ou « SIM ». Pour cette raison, il a été décidé que la version de MAD

---

[31] Une telle super méthode serait probablement assez complexe à apprendre et à utiliser. De toute manière, on arrive à un résultat semblable en offrant la possibilité de passer d'une méthode à l'autre à n'importe quel point de l'analyse et en permettant, pour une même tâche, l'entrée d'informations selon diverses méthodes.

[32] On verra plus loin que cette possibilité est indépendante de la représentation utilisée pour l'AHT.

qu'@Esperanto va supporter utilisera la structure des plans AHT pour décrire la structure du corps d'une tâche non élémentaire.

Dès lors, il devient clair qu'il suffit d'ajouter un certain nombre d'attributs aux tâches d'une AHT pour conserver ce que MAD associe à chacune (état initial, précondition, etc.).

OTHI : Il est facile de réaliser qu'un ordinogramme (la base même de tout OTHI) est en réalité un plan (au sens de l'AHT) de toutes les tâches qu'il contient. Conséquemment, un OTHI peut être représenté comme une AHT à deux niveaux dans laquelle la racine est le but ou la tâche à réaliser, les enfants sont les sous-tâches mentionnées dans ledit OTHI et où le plan de la racine est l'OTHI lui-même.

Pour garantir la compatibilité, il suffit donc que la puissance expressive des plans AHT soit au moins égale à celle des OTHI, ce qui est déjà le cas sans même commencer la normalisation.

En corollaire de cette équivalence, on voit que l'on pourra utiliser un OTHI comme représentation graphique d'un plan AHT, pour autant que la puissance expressive des OTHI soit au moins égale à celle des plans AHT. Dans ce cas-ci, il faudra attendre de voir les résultats de la normalisation des OTHI.

Analyse temporelle : La différence entre une AHT et une analyse temporelle réside surtout dans ce qui est noté à propos de chaque tâche. Comme on l'a vu plus tôt, cette différence est éminemment facile à éliminer par l'ajout d'attributs du côté de l'AHT.

Quant à la décomposition des tâches qui peut être hiérarchique ou procédurale, soit elle est identique à celle de l'AHT, soit elle correspond à une hiérarchie dégénérée (toutes les tâches étant alors les enfants de la racine).

Cela dit, il faut comprendre que l'analyse temporelle n'est pas habituellement utilisée pour étudier des procédures comportant des choix ou des boucles, surtout parce que les résultats obtenus dans ces circonstances sont loin d'être aussi clairs que dans le cas d'une simple séquence d'opérations. C'est la différence entre obtenir un intervalle de temps possiblement très grand ou une valeur simple.

La structure d'une AHT peut donc certainement représenter les analyses temporelles.

KLM-GOMS : Cette méthode sert spécifiquement à étudier une tâche procédurale routinière faite sans erreur et avec une grande familiarité (comportement d'expert). Sous forme d'AHT, nous aurions donc un arbre à deux niveaux où la racine est la tâche à réaliser, les enfants sont les étapes de la procédure étudiée en KLM-GOMS et où le plan de la racine indique qu'il faut effectuer séquentiellement les étapes de ladite procédure.

CMN-GOMS : Comme l'AHT, la tâche à analyser selon cette méthode est décomposée en une hiérarchie de sous-buts et de sous-tâches. À la différence de l'AHT, aux tâches les plus élémentaires sont associées une ou plusieurs procédures (KLM-GOMS) qui permettent de les réaliser. En présence de multiples procédures pour réaliser une même tâche, une règle de sélection est définie pour permettre un choix éclairé quant à la procédure la plus appropriée à chaque situation prévue par l'analyste.

En termes d'une AHT, la seule différence est la présence de tâches élémentaires pouvant être réalisées de plusieurs manières. Pour chacune de celles-ci, il suffit de créer autant d'enfants qu'il y a de manières possibles de la réaliser et d'avoir un plan qui calque sa règle de sélection, soit « si (situation1) alors option1 sinon si (situation2) alors option2 sinon …etc. ». Un attribut supplémentaire est suffisant pour distinguer les véritables plans de ceux calquant une règle de sélection. Les tâches nommées dans une règle de sélection sont donc traitées comme si l'on était en KLM-GOMS, alors que les autres sont traitées comme si l'on était dans une AHT.

2.5 Normalisation et notation des méthodes d'analyse supportées

On peut maintenant s'attaquer à la normalisation des méthodes d'analyse supportées en commençant, prédominance oblige, par l'AHT. Une fois l'expressivité de celle-ci bien définie, il deviendra plus facile de s'attaquer aux autres méthodes qui devront aspirer, tout en restant fidèles à leur identité propre, à s'approcher le plus possible des possibilités de l'AHT.

Par la même occasion, pour chaque méthode, on détaillera aussi la ou les notations reconnues par @Esperanto, puisque ce sera la première fois que l'entièreté de leurs possibilités est révélée. On constatera que certaines méthodes ont une notation textuelle (le plus souvent, sous forme de tableau), que d'autres ont une notation graphique et que d'autres encore ont ces deux genres de notation.

Pour obtenir une certaine uniformité dans la notation des diverses méthodes, un effort de standardisation a été fait pour qu'un maximum des éléments d'une notation soient identiques ou tout au moins semblables à celle des autres, ce qui ne pourra que plaire à Stammers (1995) qui pense que le manque de cohérence entre les méthodes représente un frein à l'adoption générale de l'analyse de tâche. Les notations textuelle et graphique montrées ici sont le fruit de cet effort.

2.6 Normalisation de l'AHT

En AHT, pour chaque tâche que l'on veut détailler, deux aspects distincts doivent être consignés : la liste des sous-tâches susceptibles de contribuer à sa réalisation, ainsi que le plan indiquant la manière dont celles-ci devront être utilisées.

Pour chacun de ces deux aspects, on définit les additions ou modifications effectuées pour l'expressivité de la méthode, puis on établit les notations textuelle et graphique reconnues par @Esperanto.

2.6.1 Normalisation et notation des tâches en AHT

En AHT, les tâches sont toujours nommées et numérotées hiérarchiquement comme il est courant pour les éléments d'une hiérarchie. Dans le cas d'une AHT, la racine de l'arbre se voit attribuer le chiffre zéro (0) alors que ses enfants sont numérotés séquentiellement à partir de un (1) en commençant par l'enfant de gauche jusqu'à celui le plus à droite. Les enfants des enfants de la racine ont toujours deux niveaux dans leur numérotation (p. ex. : 1.1) alors que les enfants de ceux-ci en ont trois (p. ex. : 1.1.1) et ainsi de suite.

En AHT et (normalisation oblige) dans toutes les autres méthodes où le concept est sensé, une tâche peut être obligatoire, facultative ou conditionnelle. La différence entre une tâche facultative et une tâche conditionnelle est que la première ne comporte pas de condition précise que l'opérateur doit vérifier pour déterminer s'il doit ou non exécuter la tâche.

Dans une notation textuelle, on réfère aux tâches obligatoires en utilisant simplement leur numéro, leur nom ou les deux. Pour les tâches facultatives ou conditionnelles, on met entre crochets la référence à la tâche ainsi qu'une éventuelle condition (celle-ci devant être précédée d'un point d'interrogation). On a donc ce qui suit :

Tableau 2-2 Notation textuelle pour les tâches.

Concept	**Exemples**	**Signification**
Tâche obligatoire	2 1.3.2 2 Vérifier la clé 1.3.2 Nettoyer la serrure	Sous-tâche dont l'exécution est obligatoire.
Tâche facultative	[4] [2.6.4] [4 Vider la trappe] [2.6.4 Nettoyer le plancher]	Sous-tâche dont l'exécution est facultative. Il appartient alors à l'opérateur de décider s'il fera ou non la tâche.
Tâche conditionnelle	[6 ? si température élevée] [3.9.6 ? quand pression basse] [6 Évacuer ? température critique] [3.9.6 Ouvrir soupape ? P > 5 atm]	Sous-tâche dont l'exécution est dépendante d'une condition spécifiée (le plus souvent) en langage naturel.

Dans une notation graphique, les tâches sont toujours représentées par une forme quelconque (le rectangle, pour une tâche banalisée) dans laquelle apparaît une combinaison quelconque du numéro et du nom de ladite tâche. Au lieu de crochets, un rectangle en tireté entoure les tâches facultatives et conditionnelles. On a donc ce qui suit :

Tableau 2-3 Notation graphique pour les tâches.

Concepts	**Exemples**
Tâche obligatoire	2 1.3.2 Vérifier la clé 1.3.2 Nettoyer la serrure
Tâche facultative	4 2.6.4 4 Vider la trappe Nettoyer le plancher
Tâche conditionnelle	6 si température élevée 3.9.6 quand pression basse 6 Évacuer température critique 3.9.6 Ouvrir soupape #9 P > 5 atm

Comme on le verra plus loin, les tâches conditionnelles permettent quelques fois de clarifier les choses en associant certaines conditions directement aux tâches plutôt que de les laisser augmenter la complexité du plan AHT.

Comme on l'a vu plus tôt, il est possible d'ajouter aux tâches autant d'attributs que désiré. En notation textuelle, une AHT est représentée sous la forme d'un tableau où chaque ligne contient les informations relatives à une tâche, ce qui permet de montrer chaque attribut dans une colonne qui lui est dédiée. Bien sûr, en présence d'un grand nombre d'attributs, la largeur des colonnes devient trop faible pour qu'elles soient utiles, ce qui conduit naturellement à l'utilisation d'une bande de quelques lignes pour chaque tâche.

Bien qu'il semble admis que cette possibilité d'ajouter autant d'attributs que désiré aux tâches d'une AHT est une caractéristique de sa représentation textuelle, tel n'est pas le cas puisqu'en notation graphique, les formes utilisées pour représenter les tâches peuvent contenir la valeur desdits attributs en plus de leurs nom et numéro. En prenant exemple sur les diagrammes-objet (bien connus dans le développement informatique), on verrait la valeur desdits attributs sous le numéro et le nom de la tâche, séparés de ceux-ci par une ligne horizontale. Ce pourrait donc être comme l'une ou l'autre des représentations suivantes :

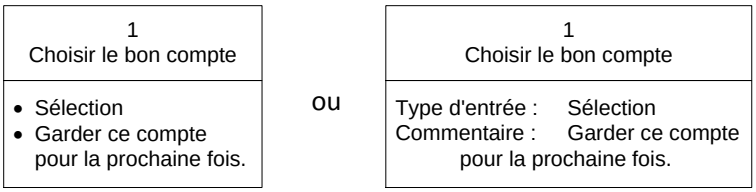


Figure 2-1 Représentations graphiques optionnelles pour les tâches d'une AHT.

Évidemment, peu importe la notation utilisée, il est possible de ne montrer qu'un sous-ensemble des informations disponibles pour chaque tâche ou d'utiliser divers codes pour permettre aux utilisateurs d'avoir au moins une indication des éléments importants. À l'écran, on peut facilement imaginer une commande (comme arrêter le pointeur de la souris au-dessus d'une tâche) qui permettrait de révéler toutes ces informations (dans une infobulle, par exemple).

À l'instar des organigrammes utilisés en informatique (et donc des OTHI), il peut être désirable d'utiliser d'autres formes que le rectangle pour illustrer une tâche selon son aspect jugé le plus important. Ainsi, outre le rectangle utilisé couramment pour les tâches sans aspect prédominant

ou dont cet aspect n'est pas déterminé, l'ANSI reconnaît plusieurs formes incluant celles qui apparaissent au tableau 2-4.

Lorsqu'il utilise la représentation textuelle de l'AHT, l'analyste doit pouvoir assigner une colonne à l'attribut qu'est en fait cet aspect jugé le plus important et indiquer qu'il veut voir celui-ci affiché comme une icône ou textuellement (soit ⬡ ou « Prép »).

2.6.2 Normalisation et notation des plans en AHT

Avant de discuter de la normalisation des plans, notons tout de suite quelques particularités souvent utilisées. D'abord, indépendamment de la représentation adoptée, le plan AHT est toujours en notation textuelle. Cela dit, au risque de gâcher un peu le suspense, @Esperanto peut tout de même montrer graphiquement les plans en profitant de l'équivalence entre AHT et OTHI, mais ne le fera que dans une infobulle et sur commande expresse de l'utilisateur.

Lorsqu'on définit le plan d'un nœud, il est superflu d'utiliser l'entièreté des numéros hiérarchiques de tâche, puisqu'on sait que les seules tâches auxquelles on va référer sont les enfants de ce nœud. En conséquence, on les identifie en n'utilisant que le dernier élément de leur numéro hiérarchique (donc, un numéro entre un (1) et le nombre d'enfants). Les figures du chapitre précédent sont de bons exemples puisqu'ils suivent ce standard.

Il est aussi possible de spécifier des intervalles et des listes de tâches selon la façon de faire habituelle dans plusieurs logiciels. Ainsi, deux numéros de tâches séparés par un tiret représentent toutes les tâches dont le numéro est égal ou entre les deux numéros indiqués, par exemple 2-5 pour les tâches 2, 3, 4 et 5. De même, une liste de tâches ou d'intervalles de tâches représente toutes les tâches énumérées ou faisant partie de l'un des intervalles mentionnés, par exemple 1,3-5,8 pour les tâches 1, 3, 4, 5 et 8.

Pour ce qui est de la normalisation, comme on peut le constater en regardant le tableau 2-1, l'AHT est l'une des méthodes les plus expressives. Par rapport à certaines mises en œuvre (celle de TaskArchitect©, par exemple), il est toutefois possible d'améliorer ce qui peut être exprimé d'une manière relativement formelle.

Pour cela, on introduit d'abord un nouveau terme pour noter différemment les concepts de parallélisme et de simultanéité.

Tableau 2-4 Formes identifiant un aspect important d'une tâche.

Formes	Signification
	Étape préparatoire.
	Tâche purement manuelle.
	Transfert physique de quelque chose.
	Entrées ou sorties de données.
	Activité nécessitant l'interaction d'un humain et d'un ordinateur.
	La tâche implique un délai (file d'attente, traitement par lots, etc.).
	Entreposage.
	Sauvegarde (entreposage électronique).
	Activité complètement automatisée (aucune intervention humaine).
	La tâche résulte en la création ou la modification d'un document.
	Activité d'inspection du résultat d'une opération précédente.
	Sous-ordinogramme (c.-à-d., une tâche complexe définie ailleurs).

Ensuite, inspiré de l'informatique et plus spécifiquement de la programmation structurée qui remplaça l'énoncé de redirection arbitraire « `GO TO` » par des énoncés plus ciblés pour définir des alternatives et des boucles de portée précise, on ajoutera des termes qui permettront de définir des processus itératifs autrement que par l'emploi d'instructions parfois imprécises ou difficiles à interpréter.

Les deux tableaux suivants montrent les divers éléments constituant chacune des deux notations textuelles reconnues par @Esperanto pour la définition des plans avec, pour chaque élément, un exemple apparaissant en italique. La notation présentée au tableau 2-5 est celle basée sur une suggestion de Stanton (2006), alors qu'au tableau 2-6, on a celle basée sur la terminologie de TaskArchitect© qui, bien que plus verbeuse, est au premier abord plus claire.

Pour faciliter l'apprentissage des éléments de la première notation qui concernent les différents enchaînements de tâches, un appui mnémotechnique est crucial. Voici donc une façon de voir qui pourra, espère-t-on, aider les lecteurs :

- Pour l'exécution séquentielle, la tête de flèche ('>') est on ne peut plus claire!
- Pour l'exécution non séquentielle, on déplie la tête de flèche (obtenant donc la '/'), pour indiquer qu'il n'y pas de séquence qui tienne…
- Pour l'exécution d'un choix unique, la barre verticale ('|') est couramment employée. C'est essentiellement un mur dressé entre les options qui interdit d'en choisir plus d'une.
- Pour l'exécution d'un nombre quelconque de tâches (donc, un ou plusieurs choix), on a percé le mur au point qu'il n'en reste qu'un deux-points (':').
- Quant à la différence entre l'exécution parallèle et simultanée, il est évident (!) qu'entre « et » ('&') et « plus » ('+') comme entre parallèle et simultanée, « plus » c'est plus!
- Finalement, le symbole utilisé pour l'interdiction est celui de négation, comme utilisé en logique, par exemple.

Comme le lecteur s'en doute dans la notation pour les plans AHT, les termes <cond>, <bloc> et <obj> représentent respectivement une condition, un bloc d'énoncé et un objet (de l'espace analysé). Quant à <coll>, il représente une collection pouvant regrouper plusieurs objets (par exemple, un moteur a plusieurs valves). Finalement, <liste de tâches> représente une liste de numéros de tâche, définie de manière similaire aux listes de numéros dans plusieurs logiciels courants.

Le seul élément de notation qui pourrait dérouter les personnes n'ayant aucune expérience en programmation est le tout dernier, soit la paire d'accolades. Le problème que cet élément permet de résoudre est celui causé par la présence d'énoncés pouvant eux-mêmes contenir un nombre quelconque d'énoncés. À titre d'exemple, reprenons l'exemple donné dans les deux tableaux précédents, mais sans utiliser les accolades, soit :

si tmp > 90 alors 1 > 3 > tantque tmp > 70 faire 5 > 6

L'interprétation de ce plan devient alors :

si tmp > 90 alors 1 >
3 >
tantque tmp > 70 faire 5 >
6

Tableau 2-5 Énoncés pour les plans d'une AHT (notation basée sur Stanton (2006)).

Concepts	**Notation et *exemples***	**Commentaires**
Exécution séquentielle	(Tâches séparées par ‘>’) *(1 > 2 > 3 > 7)* ou *seq 1-3,7*	Toutes les opérations doivent être exécutées dans l’ordre indiqué.
Exécution non séquentielle	(Tâches séparées par ‘/’) *(1 / 2 / 3 / 7)* ou *ttes 1-3,7*	Toutes les opérations doivent être exécutées sans égard à l’ordre indiqué.
Exécution d’un nombre quelconque de tâches	(Tâches séparées par ‘ :’) *(1 : 2 : 3 : 7)* ou *qcq 1-3,7*	Un nombre quelconque d’opérations (mais au moins une) doivent être exécutées, peu importe l’ordre.
Exécution d’un choix	(Tâches séparées par ‘|’) *(1 | 2 | 3 | 7)* ou *1de 1-3,7*	Une seule des opérations mentionnées doit être exécutée.
Exécution parallèle	(Tâches séparées par ‘&’) *(1 & 2 & 3 & 7)* ou *par 1-3,7*	Toutes les opérations doivent être exécutées en temps partagé (par une seule personne).
Exécution simultanée	(Tâches séparées par ‘+’) *(1 + 2 + 3 + 7)* ou *sim 1-3,7*	Toutes les opérations doivent être exécutées en même temps (par plus d’une personne, probablement).
Interdiction	¬ (Tâches séparées par ‘,’) *¬ (1)*	Ne pas faire les tâches mentionnées.
Exécution conditionnelle (la partie « sinon » étant optionnelle)	si <cond> alors <bloc> [sinon <bloc>] *si tmp > 90 alors 1 sinon 2*	Selon que la condition « X » est réalisée ou non, on choisit la sous-tâche 1 ou la sous-tâche 2.
Boucle « Pour chaque » (la partie « de » étant optionnelle)	@pc <obj> [de <coll>] faire <bloc> *@pc valve de moteur #1 faire 3*	Pour chaque objet de la collection, effectuer les opérations du bloc. La collection peut parfois être omise.
Boucle « Tantque »	@tq <cond> faire <bloc> *@tq tmp > 25 faire 8*	Tant que la condition est vraie, faire les opérations spécifiées.
Boucle « Répéter »	@rp <bloc> jusque <cond> *@rp 8 jusque tmp < 25*	Répéter les opérations spécifiées jusqu’à ce que la condition soit vraie.
Boucle à sortie centrale	@sc <bloc> finir si <cond> sinon <bloc> *@sc 6 finir si tmp < 25 sinon 7*	Exécuter le premier bloc et évaluer la condition de sortie. Si la condition est vraie, terminer. Autrement, exécuter le deuxième bloc et reprendre depuis le début.
Regroupement pour ciblage précis des instructions	{ … } comme dans *si tmp > 90 alors { 1 > 3 > @tq tmp > 70 faire { 5 > 6 } }*	Quelquefois, il faut regrouper des instructions pour leur appliquer des conditions particulières ou pour modifier la séquence d’exécution.

Tableau 2-6 Énoncés pour les plans d'une AHT (notation basée sur TaskArchitect©).

Concepts	**Notation et *exemples***
Exécution séquentielle	Faire en séquence <liste de tâches> *Faire en séquence 1-5,7,9*
Exécution non séquentielle	Toutes sans ordre <liste de tâches> *Toutes sans ordre 1-5,7,9*
Exécution d'un nombre quelconque de tâches	Nombre quelconque <liste de tâches> *Nombre quelconque 1-5,7,9*
Exécution d'un choix	Choisir l'une de <liste de tâches> *Choisir l'une de 1-5,7,9*
Exécution parallèle	Effectuer en parallèle <liste de tâches> *Effectuer en parallèle 1-5,7,9*
Exécution simultanée	Effectuer en simultané <liste de tâches> *Effectuer en simultané 1-5,7,9*
Interdiction	Ne pas faire <liste de tâches> *Ne pas faire 1*
Exécution conditionnelle (la partie « sinon » étant optionnelle)	si <cond> alors <bloc> [sinon <bloc>] *si tmp > 90 alors 1 sinon 2*
Boucle « Pour chaque » (la partie « de » étant optionnelle)	pour chaque <obj> [de <coll>] faire <bloc> *pour chaque valve de moteur #1 faire 3*
Boucle « Tantque »	tantque <cond> faire <bloc> *tantque tmp > 25 faire 8*
Boucle « Répéter »	répéter <bloc> jusque <cond> *répéter 8 jusque tmp < 25*
Boucle à sortie centrale	itérer <bloc> finir si <cond> sinon <bloc> *itérer 6 finir si tmp < 25 sinon 7*
Regroupement pour ciblage précis des instructions	{ … }.. comme dans *si tmp > 90 alors { 1 > 3 > tantque tmp > 70 faire { 5 > 6 } }*

On voit donc comment l'utilisation de la première paire d'accolades a permis d'indiquer que tous les énoncés à la droite du « alors » font partie de ce bloc, alors que la deuxième paire, elle, a permis d'indiquer que les deux derniers énoncés font partie du bloc de la boucle.

Autre point à noter, tout comme une tâche peut être facultative ou conditionnelle, les six enchaînements de tâches possibles le peuvent aussi. On peut donc indiquer que les six premiers énoncés des deux tableaux précédents (ceux commençant par le mot « Exécution ») sont facultatifs ou conditionnels. Cette indication[33] rend simplement possible le fait de ne pas effectuer l'énoncé; il ne change en rien la sémantique de l'énoncé si l'opérateur décidait de l'effectuer. Par exemple, appliquée à un énoncé spécifiant que des tâches doivent toutes être exécutées sans égard à l'ordre, cette indication signifie qu'il est possible de ne rien faire du tout ou alors d'exécuter toutes ces tâches, peu importe l'ordre.

Pour plus de clarté et pour l'euphonie des énoncés, on pourrait ajouter des synonymes pour certains de ces termes, par exemple, dans l'énoncé « pour chaque » on pourrait admettre « du », « de la » et « des » en plus du « de ». On pourrait aussi utiliser des synonymes pour permettre l'utilisation de termes tels « Choisir parmi <tâches> » ou « Ne faire qu'une des <tâches> » en guise d'équivalent à un énoncé du type « Exécution d'un choix ».

Bien sûr, puisqu'il ne faut en rien limiter l'expressivité des plans AHT, ceux-ci pourront toujours être entrés sous la forme de texte libre de toute contrainte. Il faudra alors expliquer aux analystes qu'@Esperanto ne pourra rien faire avec ceux-ci, parce qu'incapable de les comprendre.

Dans une version lointaine d'@Esperanto, des contraintes issues de la gestion de projet pourraient être ajoutées pour préciser encore plus les plans AHT. Par exemple, on pourrait avoir des conditions spéciales comme « Au début », « À la fin », « Avant de terminer » et « À toutes les n fois » pour établir des contraintes d'exécution précises entre les sous-tâches. Lors des premières phases du développement, les instructions en langue naturelle suffiront et permettront d'accumuler des données objectives à propos des véritables besoins des utilisateurs.

2.6.3 Représentations d'une AHT

La figure 2-2 et la figure 2-3 reprennent un exemple du chapitre précédent pour mettre en évidence certaines des modifications apportées lors de la normalisation. On notera l'utilisation de

---

[33] Bien sûr, on parle ici des crochets ('[]') de la notation textuelle et du rectangle en tireté de la notation graphique.

tâches conditionnelles pour simplifier le plan de la racine, ainsi que l'utilisation de formes autres que le rectangle pour identifier l'aspect principal de certaines tâches.

Par la même occasion, on notera que l'on aurait aussi pu modifier le plan 3.1 en utilisant des tâches conditionnelles, l'une devant être effectuée si le destinataire est dans « contacts », l'autre s'il ne l'est pas. L'intention aurait très probablement été obscurcie par l'opération, ce qui tend à prouver qu'il ne faut pas abuser des bonnes choses…

2.7 Normalisation de l'analyse procédurale / OTHI

Comme mentionné lors de sa présentation, l'OTHI est une forme particulière d'analyse procédurale dont l'ordinogramme est le principal outil de structuration des données recueillies lors de l'analyse de tâche.

Les ordinogrammes n'ont pas été prévus pour illustrer facilement tous les concepts pouvant être exprimés de manière (relativement) formelle dans une AHT. Plutôt que de forcer les analystes à n'utiliser que les symboles standards pour enfanter de constructions qui peuvent vite devenir difficiles à interpréter, nous allons étendre le lexique des OTHI par l'utilisation de nouveaux symboles.

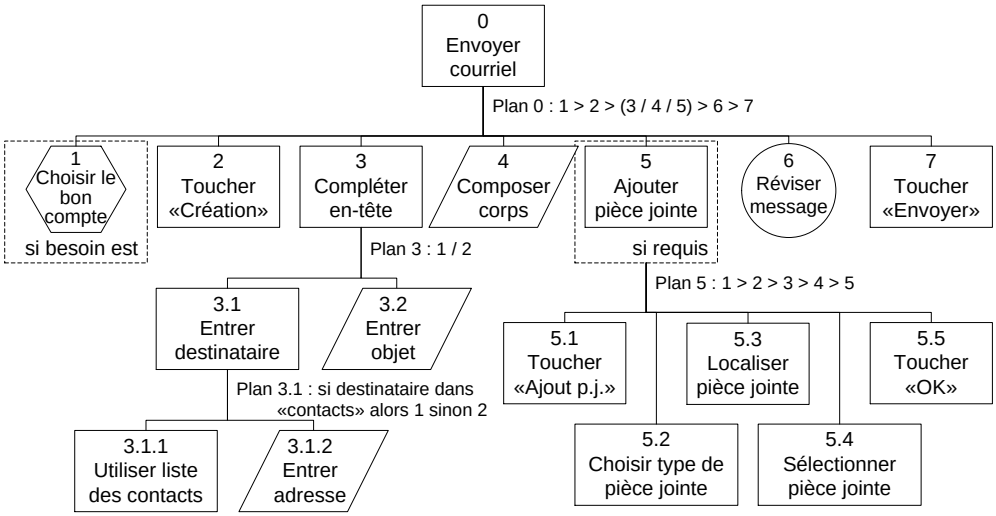


Figure 2-2 AHT normalisée pour la tâche d'envoi d'un courriel (repr. graphique).

Tâches		Type d'entrée	Commentaire
0 Envoyer courriel			
Plan 0 : 1 > 2 > (3 / 4 / 5) > 6 > 7			
[1 Choisir le bon compte ? si besoin est]	⬡	Sélection	Garder ce compte pour la prochaine fois.
2 Toucher «Création»		Clic (bouton)	
3 Compléter en-tête			
Plan 3 : 1 / 2			
3.1 Entrer destinataire			
Plan 3.1 : si destinataire dans «contacts» 1, sinon 2			
3.1.1 Utiliser liste des contacts		Clic (élément)	
3.1.2 Entrer adresse	▱	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.
3.2 Entrer objet	▱	Texte	
4 Composer corps	▱		
[5 Ajouter pièce jointe ? si requis]			Utilisation occasionnelle.
Plan 5 : 1 > 2 > 3 > 4 > 5			
5.1 Toucher «Ajout p.j.»		Clic (bouton)	
5.2 Choisir type de pièce jointe		Clic (élément)	
5.3 Localiser pièce jointe		Navigation	Fichier local seulement?
5.4 Sélectionner pièce jointe		Clic (élément)	
5.5 Toucher «OK»		Clic (bouton)	
6 Réviser message	○		
7 Toucher «Envoyer»		Clic (bouton)	

Figure 2-3 AHT normalisée pour la tâche d'envoi d'un courriel (repr. textuelle).

Pour la normalisation, on commence par introduire dans l'OTHI la notation permettant de spécifier qu'une tâche (ou un groupe de tâches) est facultative ou optionnelle. Pour être cohérent entre les diverses notations, on réutilise ici la notation adoptée pour l'AHT.

On ajoute ensuite six nouveaux symboles pour les six enchaînements de tâches qui ne sont pas déjà exprimables dans un OTHI, puis on termine en définissant de nouveaux symboles correspondant à des énoncés composés inspirés de la programmation structurée (soit les mêmes énoncés qui ont été ajoutés pour les plans d'une AHT).

Tous ces nouveaux éléments de la notation graphique des OTHI sont montrés dans les deux tableaux suivants.

2.7.1 Représentations d'un OTHI

La figure 2-4 reprend un exemple du chapitre précédent pour mettre en évidence certaines des modifications apportées lors de la normalisation. On notera que l'on n'a pas utilisé de tâches conditionnelles (au sens de ce que l'on vient de voir avec l'AHT, par exemple). La raison de cela est que l'on a voulu respecter le style de l'OTHI, une méthode qui, comme on l'a vu plus haut, met l'accent sur le processus et les décisions à y prendre. Pour être précis, comme on le verra plus loin, rien n'interdit l'utilisation de tâches conditionnelles dans un OTHI, mais la nature même de la méthode et son accent sur les décisions à prendre favorisent plutôt l'emploi de conditions explicites.

On remarque que le premier jet, tout aussi expressif que l'AHT de la section précédente, est cependant moins facile à comprendre que la version standard (mais moins expressive) montrée dans le chapitre de présentation. La difficulté vient de la relative complexité des tâches pouvant être réalisées dans n'importe quel ordre. Heureusement, la création de deux sous-routines[34] remédie facilement à ce problème, comme on peut le constater à la figure 2-5. On remarquera aussi que l'appel à la sous-routine « Ajouter pièce jointe » est maintenant conditionnel, plutôt qu'être dans la partie « alors » d'un énoncé conditionnel; ce qui est sémantiquement identique.

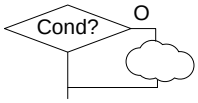
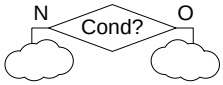
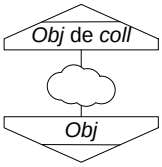
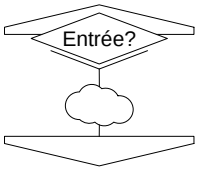
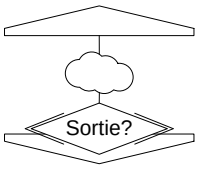
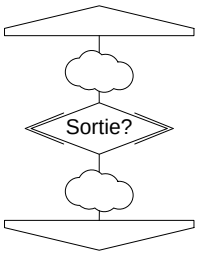
---

[34] Une sous-routine est une construction très utilisée en programmation pour décomposer un gros programme en unités plus petites, donc plus facilement concevables et compréhensibles. Un programme va alors appeler des sous-routines qui elles-mêmes auront la possibilité d'en appeler d'autres. Dans un organigramme, elle a la même utilité.

Tableau 2-7 Notation graphique des différents enchaînements de tâches.

Concept	**Notation graphique**	**Commentaires**
Exécution séquentielle	T_1 — T_2 — T_3 → T_4 — … — T_n	Aucun changement par rapport aux ordinogrammes existants.
Exécution non séquentielle	T_1 … T_n	À noter le nombre de lignes de côté qui diminue avec le nombre de tâches ou d'énoncés à exécuter…
Exécution d'un nombre quelconque de tâches.	T_1 … T_n	De deux pour l'exécution de toutes les tâches, à une pour l'exécution du nombre désiré…
Exécution d'un choix	T_1 … T_n	…à aucune pour l'exécution d'une seule tâche ou d'un seul énoncé.
Exécution parallèle	T_1 … T_n	Ici, les deux lignes <u>parallèles</u> en haut et en bas vendent la mèche!
Exécution simultanée	T_1 … T_n	Trois lignes pour le simultané qui, d'une certaine manière, est plus que du parallélisme.
Interdiction	T	Dans cette notation, il y a deux indications plutôt qu'une qu'il ne faut pas exécuter cette tâche.

Tableau 2-8 Notation graphique des énoncés inspirés de la programmation structurée.

Concept	**Notation graphique**[35]	**Commentaires**
Exécution conditionnelle		Modèle classique pour une action ne devant être réalisée que sous une condition donnée.[36]
Exécution conditionnelle avec option		Modèle classique pour faire face à une situation nécessitant des actions différentes selon une condition donnée.[36]
Boucle « Pour chaque » (la partie « de » étant optionnelle)		Modèle à utiliser lorsqu'il faut faire certaines choses pour tous les objets d'une certaine collection (qui pourrait rester anonyme).[37]
Boucle « Tantque »		Modèle à utiliser lorsqu'il faut exécuter certaines actions tant et aussi longtemps que la condition d'entrée est vraie.
Boucle « Répéter »		Modèle à utiliser lorsqu'il faut exécuter certaines actions au moins une fois, mais aussi longtemps que la condition de sortie reste fausse.
Boucle à sortie centrale		Modèle à utiliser lorsqu'il faut exécuter plusieurs actions de manière itérative, mais où la condition de sortie ne peut être au début ou à la fin de cette série d'actions.

[35] La raison d'être des nuages apparaissant dans les divers symboles sera expliquée en détail dans le chapitre suivant. Pour l'instant, il suffit de savoir qu'ils indiquent l'endroit où devront être placés des tâches ou énoncés.

[36] Ce modèle n'a rien de neuf. Il est offert en guise de commodité pour les analystes.

[37] Lorsqu'il ne peut y avoir confusion quant à la collection dont il est question, il est inutile de la mentionner ici.

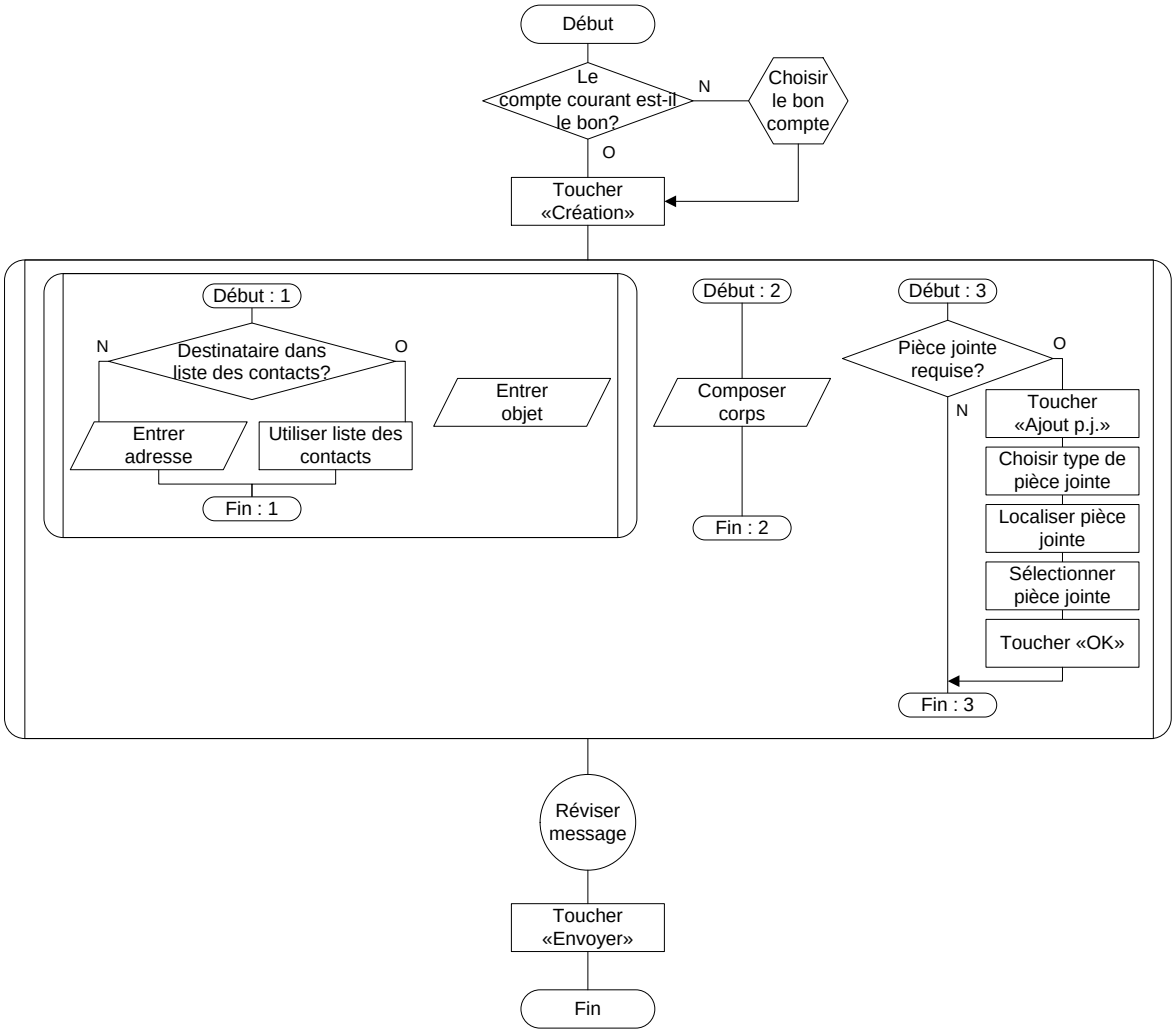


Figure 2-4 OTHI normalisé correspondant à l'AHT partielle pour l'envoi d'un courriel.

Quant à la représentation textuelle d'un OTHI, il va sans dire que la notation retenue est essentiellement celle adoptée pour l'AHT. Le seul ajout nécessaire ici consiste à gérer les sous-routines avec un énoncé pour les définir et un autre pour les appeler.

On traitera plus loin du cas particulier des procédures, ces représentations textuelles d'OTHI construites en respectant des contraintes de rédaction qui les rendent aptes à servir d'aide à la performance humaine.

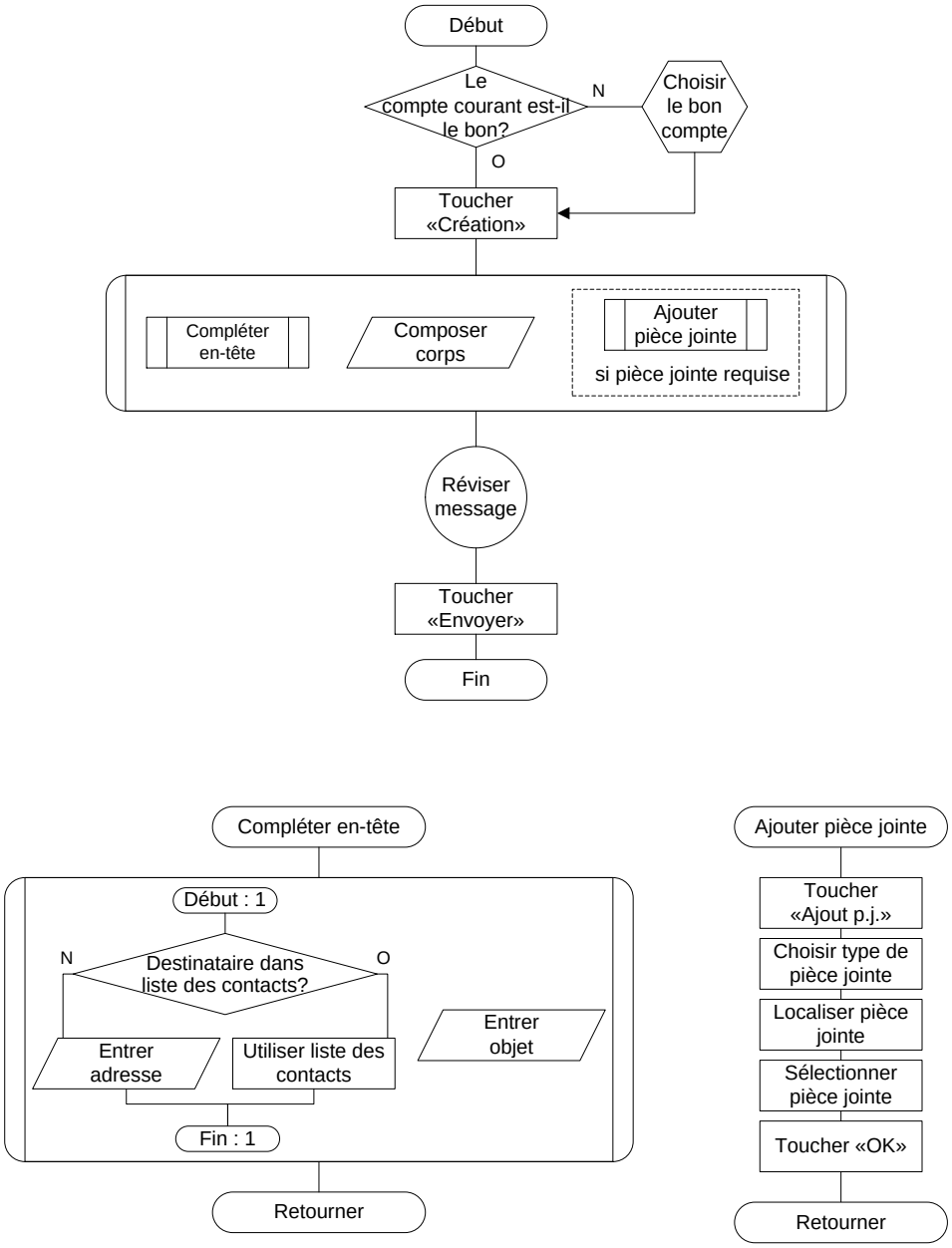


Figure 2-5 OTHI normalisé identique au précédent hormis l'utilisation de sous-routines.

Pour l'instant[38], on dira qu'une sous-routine est appelée, à l'instar d'une fonction dans la plupart des langages de programmation, par son nom suivi de parenthèses. Cette définition comprend trois parties, soient un en-tête, un corps (les instructions) et les mots « FIN OTHI ». Quant à l'en-tête d'une sous-routine, elle doit commencer par le mot composé « SOUS-ROUTINE », suivi du

[38] Une description plus détaillée des sous-routines est présentée un peu plus loin.

	OTHI Envoyer un courriel		
1	**SI** le compte n'est pas le bon **ALORS**		
2	Choisir le bon compte	Sélection	Il faudrait mémoriser ce compte pour la prochaine fois.
3	Toucher « Création »	Clic (bouton)	
4	**Toutes sans ordre**		
	{		
4.1	Compléter en-tête(),		
4.2	Composer corps,	Texte	
4.3	[Ajouter pièce jointe() ? si pièce jointe requise]		
	}		
5	Réviser message		
6	Toucher «Envoyer»	Clic (bouton)	
7	**FIN OTHI**		
	SOUS-ROUTINE Compléter en-tête ()		
1	**Toutes sans ordre**		
	{ **DEBUT** : 1		
1.1	**SI** le destinataire est dans la liste des contacts **ALORS**		
1.1.1	Utiliser la liste des contacts	Clic (élément)	
1.2	**SINON**		
1.2.1	Entrer l'adresse courriel	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.
	FIN : 1		
2.1	Entrer l'objet	Texte	
	}		
2	**RETOURNER**		
	FIN SOUS-ROUTINE		
	SOUS-ROUTINE Compléter en-tête ()		Att : Utilisation occasionnelle!
1	Toucher « Ajout p.j. »	Clic (bouton)	
2	Choisir le type de pièce jointe	Clic (élément)	
3	Localiser la pièce jointe	Navigation	Fichier local seulement?
4	Sélectionner la pièce jointe	Clic (élément)	
5	Toucher « OK »	Clic (bouton)	
6	**RETOURNER**		
	FIN SOUS-ROUTINE		

Figure 2-6 Représentation textuelle de l'OTHI de la figure 2-4.

nom de ladite sous-routine et de parenthèses. Parmi les instructions d'une sous-routine, le mot « RETOURNER » sert à indiquer qu'il est temps d'arrêter l'exécution de cette sous-routine pour plutôt reprendre l'exécution des instructions de l'appelant immédiatement après l'appel de la sous-routine courante.

En guise d'exemple, la figure 2-6 présente l'équivalent textuel de l'OTHI de la figure 2-5. On y reconnaîtra plusieurs éléments de la notation textuelle de l'AHT.

2.7.2 Sous-routines et paramètres

Jusqu'ici, la seule chose que l'on devait connaître à propos des sous-routines est qu'elles contiennent une série d'instructions et que l'on peut les appeler (dans le but d'exécuter ladite série d'instructions) simplement en utilisant leur nom. C'était suffisant puisqu'il ne pouvait y avoir de confusion quant à ce sur quoi agissent les instructions des sous-routines montrées dans les exemples précédents. Ces sous-routines regroupaient simplement des instructions que, dans la représentation graphique, il aurait été possible de mettre sur une autre page en utilisant des marqueurs du type « référence sur une autre page » (▷).

Dans certains cas cependant, une sous-routine contient des instructions qui peuvent être appliquées à l'un de plusieurs composants semblables. Par exemple, alors qu'en général une automobile n'a qu'un moteur, un avion de ligne en possède plusieurs. Il est alors plus logique de définir une seule fois les ensembles d'instructions pouvant s'appliquer à n'importe quel moteur, plutôt que de les définir indépendamment pour chaque moteur. C'est donc dire que l'on veut indiquer une seule fois comment démarrer ou arrêter un moteur (définition de sous-routines) tout en pouvant indiquer de démarrer ou d'arrêter tous les moteurs en utilisant chaque fois les mêmes instructions, mais pour des moteurs différents (plusieurs appels d'une même sous-routine). C'est ici que la notion de paramètres est utile.

La définition complète d'une sous-routine comporte les éléments montrés au tableau 2-9.

Dans cette définition, il reste à spécifier que la définition des paramètres consiste, pour chaque paramètre, à lui donner un nom et à le décrire. En ce qui concerne la syntaxe, ces deux éléments sont séparés par un deux-points et les définitions sont séparées par un point-virgule. Par exemple, on pourrait avoir l'en-tête de la figure 2-7 pour une sous-routine.

Tableau 2-9 Les divers éléments de la définition d'une sous-routine.

Partie	**Syntaxe**	**Commentaire**
En-tête	SOUS-ROUTINE *nom-de-la-sous-routine* (*définition-des-paramètres*)	Introduction et nom Liste des paramètres
Corps	*liste-d'énoncés* RETOURNER	Suite d'énoncés Indique le retour vers l'appelant
Terminaison	FIN SOUS-ROUTINE	Fin de la définition

```
SOUS-ROUTINE  Ajuster une valve moteur      (NoValve : numéro de la valve à ajuster;
                                             NoMoteur : numéro du moteur où est la valve)
Aller au moteur # NoMoteur
Localiser la valve # NoValve
        …
FIN SOUS-ROUTINE
```

Figure 2-7 Exemple de sous-routine selon la syntaxe du tableau 2-9.

L'idée d'avoir des paramètres servira surtout lorsque @Esperanto (dans une version lointaine) permettra l'exécution assistée d'une procédure. Dans une telle situation, le nom des paramètres des sous-routines est automatiquement remplacé par la valeur passée en paramètre dans l'appel courant. Pour l'instant, on dira simplement que les valeurs données à ces paramètres lors des divers appels de sous-routines vont être utilisées pour déterminer l'ensemble des valeurs possibles pour chaque paramètre.

Il est aussi important de noter que l'énoncé « RETOURNER » peut apparaître n'importe où dans le corps d'une sous-routine[39], mais qu'il est optionnel à la fin d'une sous-routine parce que considéré implicite devant un énoncé « FIN SOUS-ROUTINE ».

[39] Il suffit de s'assurer que les énoncés suivants ne sont pas rendus inaccessibles par sa présence. Par exemple, dans la séquence suivante, il y a un énoncé inaccessible : « … / RETOURNER / Réviser le travail / FIN SOUS-ROUTINE ».

2.7.3 Création de procédures pour l'aide à la performance humaine

Les ajouts faits à la méthode lors de cette normalisation sont surtout utiles aux analystes qui doivent d'abord bien connaître les tâches à réaliser pour arriver aux buts souhaités. Cependant, lorsque vient le temps de définir les processus qui devraient être mis en place pour atteindre ces buts, il est entendu qu'il faut être plus directif que ce que permet l'OTHI normalisé.

Pour garantir que les procédures élaborées pour des opérateurs soient dénuées de tous ces ajouts, @Esperanto offre deux possibilités aux analystes. D'abord, il peut mettre en évidence tous les énoncés résultant de la normalisation, ce qui facilite l'inspection d'une procédure pour vérifier que ceux-ci ont bien été laissés intentionnellement. Ensuite, assisté de l'analyste qui identifiera les tâches à réaliser et celles à mettre de côté, @Esperanto va transformer chacun des nouveaux énoncés en une suite d'énoncés plus simples pour produire une nouvelle procédure ne contenant aucun des ajouts introduits lors de la normalisation.

Il ne restera alors au rédacteur de procédures qu'à vérifier les règles de composition particulières aux procédures. Par exemple, assurer que chaque tâche commence par un verbe actif, que les seuls sigles employés sont connus des opérateurs, qu'il n'y ait pas de négation qu'il serait préférable de transformer en énoncé positif, etc. Il serait bon qu'@Esperanto puisse assister le rédacteur dans sa tâche, mais ce pourrait être dans une version assez lointaine.

2.8 Normalisation de MAD

Par rapport à l'AHT, la seule faiblesse de MAD réside dans ses directives d'utilisation des sous-buts et sous-tâches enfants servant à atteindre le but ou réaliser la tâche qui est le parent. Comme mentionné plus haut, @Esperanto annule cette faiblesse en incorporant les plans AHT dans MAD. Cette décision entraîne la disparition de certains éléments de MAD, soit les booléens « Boucle » et « Facultative », puisque leurs fonctions sont aisément remplies par l'utilisation de plans AHT.

Pour passer d'AHT à MAD, l'analyste devra, pour chaque tâche, définir la valeur des attributs emblématiques de MAD (état initial, précondition,), ce qui peut être un peu long, mais ne présente aucun problème méthodologique. Dans le sens inverse, plutôt que d'enlever les attributs

emblématiques de MAD, il suffit de les ajouter à chaque tâche AHT comme si c'était des attributs spécifiques à l'étude.

Il n'y a donc rien de plus à faire pour la normalisation de MAD.

2.8.1 Représentations d'une analyse selon MAD

La figure 2-8 reprend un exemple du chapitre précédent pour montrer que l'utilisation de plans AHT en MAD ne change rien d'important à la méthode.

0 -- Envoyer courriel

But: Envoyer un courriel du compte choisi avec possibilité d'ajouter une pièce jointe.

État initial
- Logiciel de gestion des courriels
- Fichier à joindre.

Préconditions
- Logiciel de gestion des courriels ouverts;
- Fichier à joindre disponible au système de gestion des fichiers de l'ordinateur.

Corps de la tâche

Plan : 1 > 2 > (3 / 4 / 5) > 6 > 7

Sous-tâches

[1 Choisir le bon compte ? si besoin est]
2 Toucher «Création»
3 Compléter en-tête
4 Composer corps
[5 Ajouter pièce jointe ? si requis]
6 Réviser message
7 Toucher «Envoyer»

État final
- Logiciel de gestion des courriels;
- Courriel;
- Fichier à joindre;

Postconditions
- Courriel envoyé;
- Fichier à joindre envoyé avec le courriel si la tâche 5 a été effectuée.
- Logiciel de gestion des courriels dans le même état qu'avant la tâche.
- Fichier à joindre non modifié sur le disque dur;

Résultat

Le courriel et, si désiré, le fichier à joindre sont envoyés.

<aucun attribut>

Figure 2-8 Représentation normalisée possible de l'envoi d'un courriel.

Étant donné que l'AHT a aussi une représentation graphique et qu'un OTHI est équivalent à une AHT, le corps de la tâche pourrait apparaître comme l'une des options montrées à la figure 2-9.

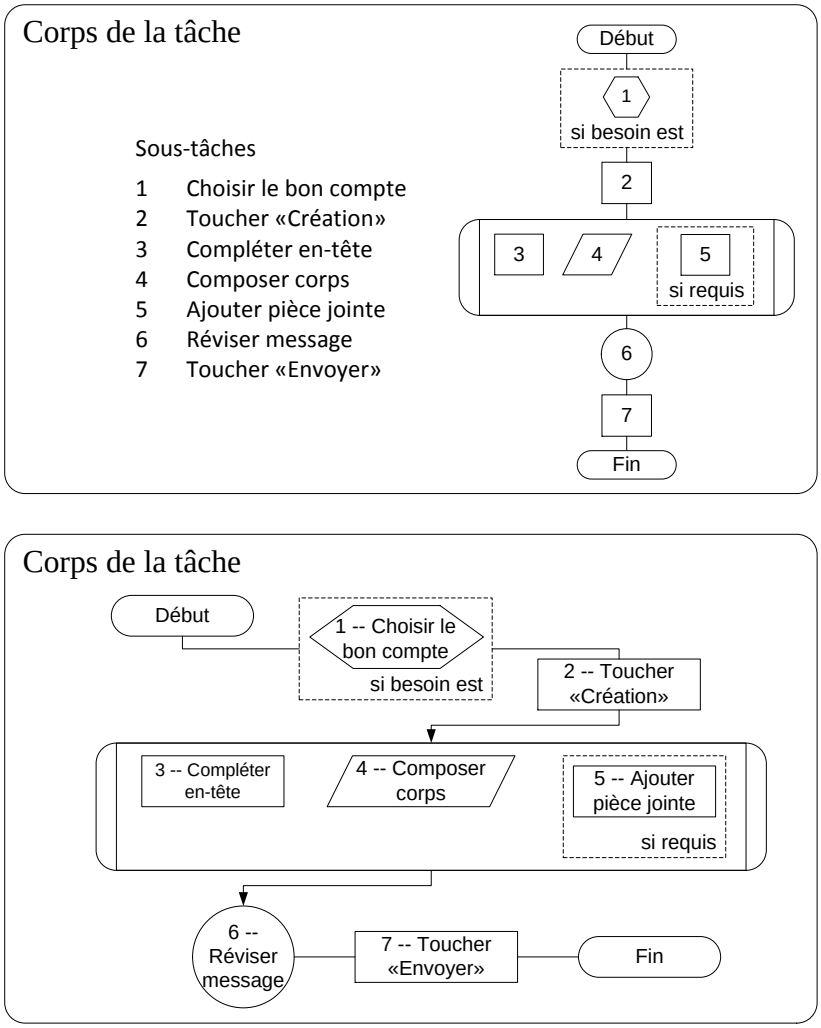


Figure 2-9 Deux représentations optionnelles pour le corps d'une tâche MAD.

2.9 Normalisation de l'analyse temporelle

Dans l'état actuel des choses, l'analyse temporelle ne porte que sur des séquences d'opérations permettant dans certaines circonstances de réaliser une tâche.

Pour satisfaire cette exigence de la méthode, l'analyste détermine un ensemble de situations qu'il désire étudier. Partant de là, il détermine pour chacune de ces situations une séquence d'opérations qui peut réaliser la tâche principale. En présence d'une analyse de tâche plus complète, que ce soit sous la forme d'une AHT, d'un OTHI ou de MAD, l'analyste se servira

plutôt de celle-ci pour dégager la séquence d'opérations appropriée à chacune des situations à étudier.

Le terme *scénario* est souvent employé pour référer à chacune de ces séquences qui réalisent la tâche lors de circonstances précisément définies, ainsi qu'à l'ensemble des circonstances dans lesquelles elles peuvent être utilisées.

Le point important est que les séquences d'opérations obtenues ne doivent comporter aucune condition ou boucle qui pourrait modifier le chemin d'exécution. Ce n'est qu'ainsi que l'ordre des opérations peut être une séquence bien définie et que le temps nécessaire pour les effectuer peut être calculé à partir du temps d'exécution de chaque opération de ladite séquence (plutôt qu'estimé être dans un intervalle de temps plus ou moins grand).

Cet état de fait sert très bien l'analyse temporelle dont le but même est de connaître l'ordre d'exécution et, le plus souvent, le temps nécessaire pour réaliser la tâche principale.[40]

Cela dit, il serait techniquement possible d'ajouter à l'analyse temporelle des énoncés qui lui permettraient d'exprimer des enchaînements autres que la simple séquence de tâches. De même, on pourrait lui ajouter divers énoncés exprimant des conditions ou des boucles.

La résultante de n'importe lequel de ces ajouts serait de créer des situations où il y aurait plus d'un chemin d'exécution possible pour arriver à réaliser la tâche principale… en fonction de diverses circonstances. Comme on peut s'en douter, ces ajouts iraient à l'encontre de ce qui fait la simplicité et la force de cette méthode.

Reconnaissant donc que les buts de l'analyse temporelle n'incluent pas une compréhension globale de la tâche principale au même titre qu'en AHT, OTHI ou MAD, aucune tentative de rapprochement avec ces méthodes n'a été tentée.

Le seul assouplissement que l'on accepte pour cette méthode est effectué dans le but de la rapprocher de KLM-GOMS, qui est souvent employé pour analyser plusieurs manières d'atteindre un même but. Histoire donc de faciliter l'analyse de plusieurs manières d'accomplir une même tâche, que ce soit pour un seul et même scénario ou pour plusieurs, @Esperanto autorise l'analyste à définir autant d'analyses qu'il le juge nécessaire. Il doit cependant nommer

---

[40] Il arrive aussi que la charge mentale de travail soit étudiée en même temps que l'ordre et le temps d'exécution.

et définir les conditions d'utilisation de chacune de manière à pouvoir les identifier plus tard. Par exemple, pour la tâche « Entrer destinataire » de l'exemple récurrent de ce chapitre, l'analyste pourrait effectuer deux analyses (correspondant ici à chacun des scénarios possibles), l'une nommée « Utiliser liste des contacts / si le contact est connu » et l'autre « Entrer adresse / dans le cas d'un nouveau contact ».

2.9.1 Identification des chemins d'exécution d'intérêt

La plus grande assistance que peut fournir @Esperanto à ses utilisateurs est de leur faciliter l'identification des chemins d'exécution d'intérêt à partir d'une analyse globale réalisée avec une autre méthode.[41]

Parce qu'@Esperanto est capable de convertir une AHT ou une analyse MAD en un OTHI, il est capable de représenter toute tâche en termes de ses sous-tâches et de ses opérations élémentaires. Partant de cette représentation, il est plus facile pour l'analyste de choisir le ou les chemins d'exécution qu'il désire analyser sous l'angle de l'analyse temporelle. Cela dit, l'analyste ne doit pas être contraint par un chemin d'exécution et doit pouvoir modifier l'ordre des tâches obtenues de cette façon.

Il est à noter que l'agencement des tâches d'un scénario peut contenir des tâches qu'il faut effectuer en parallèle ou en simultané. Ceci amène le concept de « fils d'exécution » qui ne doit pas être confondu avec celui de « chemin d'exécution » que l'on vient de voir. Un fil d'exécution ne peut comporter qu'une tâche à la fois, conséquemment, si trois tâches doivent être réalisées en parallèle ou en simultané, il y aura trois fils d'exécution tant et aussi longtemps qu'elles seront en cours.

Les détails de fonctionnement sont laissés pour le chapitre suivant.

---

[41] Ce qui ne veut pas dire que l'utilisateur se verra empêché d'analyser une tâche qui n'a pas fait l'objet d'une analyse avec une autre méthode.

2.10 Normalisation de KLM-GOMS

Une analyse KLM-GOMS est une sorte d'analyse temporelle très fine et basée sur des théories de la psychologie moderne concernant les capacités de perception et d'action des êtres humains plutôt que sur des observations du travail ou des estimés à propos des temps d'exécution.

Cela dit, on comprendra que les buts poursuivis lors d'une analyse KLM-GOMS sont essentiellement les mêmes que ceux poursuivis lors d'une analyse temporelle et que, pour les mêmes raisons, aucune modification ne sera faite à la méthode comme telle.

Une analyse KLM-GOMS étant souvent employée pour évaluer les différences entre plusieurs manières de réaliser une même tâche, la version de KLM-GOMS supportée par @Esperanto autorise la présence de plusieurs séquences d'opérations associées à une même tâche. À l'instar de l'assouplissement vu plus tôt concernant l'analyse temporelle, l'analyste doit nommer et définir les conditions d'utilisation de toutes ces analyses. On peut alors voir cette dernière construction comme étant la règle de sélection qui permettra à l'opérateur de choisir la bonne façon de faire. Par exemple, pour la tâche « Utiliser liste des contacts » de l'exemple récurrent de ce chapitre, l'analyste pourrait effectuer deux analyses, l'une nommée « Utilisation de la souris / si l'utilisateur est un novice » et l'autre « Utiliser tous les raccourcis clavier / si l'utilisateur est un expert ». Comme montré à la figure 2-10, il est aussi possible d'analyser différents scénarios; pas seulement diverses façons d'accomplir la même chose dans le même scénario.

2.11 Normalisation de CMN-GOMS

A priori, par sa décomposition hiérarchique d'une tâche (qualifiée ici de « principale »), il semble concevable de penser à une analyse CMN-GOMS comme étant une AHT à laquelle on aurait associé une analyse KLM-GOMS à chaque feuille de l'arbre d'analyse. Ainsi, CMN-GOMS permet en quelque sorte d'établir le contexte pour chaque analyse KLM-GOMS.

Dans le cas où l'on voudrait effectuer des calculs de temps de complétion pour les nœuds de l'arbre AHT qui ne sont pas des feuilles, les plans se doivent d'indiquer que toutes les tâches sont à faire de manière séquentielle. D'une manière générale, ce n'est toutefois pas nécessaire.

Comme pour l'analyse temporelle, ce qui intéresse l'analyste ici n'est pas tant la décomposition hiérarchique en elle-même que les différents scénarios possibles pour réaliser chaque sous-tâche

Entrer destinataire		
CHOISIR:		
Utiliser liste des contacts… si le contact est connu		
1 [Pointer et cliquer le bouton « To > »]	PC	1.30
2 [Taper les trois premiers caractères du nom de famille]	MHKKK	2.59
3 [Cliquer le bon nom dans la liste des contacts]	HPC	1.70
4 [Pointer et cliquer le bouton « OK »]	PC	1.30
	Total	*6.89*
Entrer adresse… dans le cas d'un nouveau contact		
1 [Pointer et cliquer le champ d'édition de l'adresse courriel]	PC	1.30
2 [Taper l'adresse courriel (en moyenne 24 caractères)]	MHK(24)	8.47
	Total	*9.77*

Figure 2-10 Deux analyses KLM-GOMS pour l'entrée de l'adresse du destinataire.

identifiée dans la réalisation de la tâche principale. Ces scénarios étant constitués ici par une séquence d'opérations choisies, à chaque étape, parmi les diverses possibilités. Conséquemment, les scénarios CMN-GOMS peuvent aussi être identifiés par la liste des conditions dans chaque règle de sélection qui correspondent à chacune des méthodes choisies.

On constate que les conditions d'utilisation de la méthode sont essentiellement les mêmes que pour l'analyse temporelle.

Même constatation pour l'analyse KLM-GOMS puisque chaque scénario CMN-GOMS correspond en fait à une analyse KLM-GOMS.

Aucune normalisation ne sera donc entreprise.

2.12 Synthèse

Les méthodes d'analyse supportées par @Esperanto étant normalisées selon leur classe (globale ou locale), le chapitre suivant porte sur les interfaces utilisateur suggérées pour créer des analyses selon chacune des méthodes supportées.

CHAPITRE 3 CONCEPTUALISATION ET PROTOTYPAGE DE L'IU

Principalement, ce chapitre expose de manière assez détaillée les fenêtres importantes de l'interface utilisateur d'@Esperanto. Une section consacrée aux fonctionnalités utiles sans égard à la méthode employée précède celles dévouées à chacune des méthodes d'analyse de tâche supportées.

Par la suite, on aborde le fonctionnement des analyses multiméthode et on termine par une démonstration de conversions entre AHT et OTHI.

3.1 Fonctionnalités d'utilité générale

3.1.1 Recherche de tâches à partir de la fenêtre principale

Comme dans tous les logiciels, il est possible de rechercher un élément en fonction d'une chaîne de caractères. Dans le cas d'@Esperanto, la recherche de base permet de trouver une tâche en fonction de son numéro ou de son nom. Conséquemment, lorsqu'un utilisateur veut trouver une tâche particulière, il doit invoquer la fonction de recherche, puis entrer un texte faisant partie du numéro ou du nom de ladite tâche, en réponse à quoi, @Esperanto lui montre une liste des tâches dont le numéro ou le nom comporte le texte entré.

Pour plus de souplesse, @Esperanto permet aussi de rechercher les tâches par la valeur assignée à chacun de ses attributs (qui sont spécifiques à chaque étude). Ainsi, dans les exemples d'AHT et d'OTHI du chapitre précédent, il serait possible de rechercher toutes les tâches dont l'attribut « Type d'entrée » contient la chaîne de caractères "élément".

Notons que tout ensemble de tâches résultant d'une recherche peut rester accessible, dans une fenêtre non modale, donc parallèlement à la fenêtre principale. Cela permet d'amener successivement et sous le contrôle de l'utilisateur n'importe laquelle des tâches trouvées au centre de la fenêtre principale. Dans le cas des exemples susmentionnés, l'ensemble des résultats contiendrait trois tâches sur lesquelles il suffirait de cliquer pour les amener en vue dans la fenêtre principale.

Pour aider l'utilisateur à choisir la bonne tâche, lorsque plusieurs tâches remplissent la condition soumise, @Esperanto offre à l'utilisateur différentes vues sur les résultats. Il peut afficher les

tâches en ne montrant que leur numéro et leur nom, ou alors montrer tous leurs attributs. Lorsque cela reste insuffisant, l'utilisateur peut survoler chacune des tâches dans les résultats de la recherche pour la voir en contexte, car en réponse à cela @Esperanto va montrer une infobulle dans laquelle la tâche survolée apparaîtra bien identifiée dans une portion raisonnablement affichable de son analyse de tâche (cette portion étant dans la même représentation que celle couramment utilisée dans la fenêtre principale).

3.1.2 Navigation dans une analyse de tâche

Pour l'utilisateur qui veut simplement naviguer à travers de son analyse de tâche, il est important de fournir des outils qui permettent d'en apercevoir la plus grande portion possible. La première chose à laquelle on pense est le zoom, soit la possibilité de modifier la taille des éléments montrés à l'écran de manière à pouvoir afficher une portion plus ou moins grande de l'analyse dans une fenêtre dont les dimensions ne changent pas. La seule limitation à imposer sur cette fonctionnalité est d'arrêter de rapetisser les éléments lorsque toute l'AHT est devenue visible dans la fenêtre ou de les grossir lorsqu'une tâche deviendrait plus grosse que la fenêtre. Quant au reste, ce sera à chaque utilisateur de déterminer les limites dans lesquelles il trouve pratique l'utilisation du zoom.

De manière à permettre d'en voir un peu plus, l'affichage passe de la vue normale à une vue plus ou moins hyperbolique tant et aussi longtemps que la touche « Maj » reste enfoncée. Lors des transitions d'une vue à l'autre, le point stable entre elles est la tâche située le plus au centre de la fenêtre, rendant ainsi prévisible ce qui va se passer lors d'un tel changement. Lors du passage vers la vue hyperbolique, non seulement la tâche qui était la plus proche du centre de la fenêtre est-elle ramenée exactement à son centre, mais le pointeur de la souris l'est aussi. La raison de ce comportement est que, dans ce type de vue, c'est la position dudit pointeur par rapport au centre de la fenêtre qui détermine la direction et la vitesse du mouvement de la lentille virtuelle au travers de laquelle l'analyse est visualisée.[42]

---

[42] En fait, c'est le vecteur orienté partant du centre de la fenêtre jusqu'à la position du pointeur de la souris qui indique la direction et la vitesse du déplacement. La position centrale où le pointeur est placé initialement correspond donc à un déplacement nul.

La forme de cette lentille virtuelle est dépendante de la méthode et de la représentation couramment utilisées. En effet, certaines représentations, telle l'AHT en représentation graphique, ont tendance à s'étirer plus dans une dimension que dans l'autre (ici, sur la largeur).

Reconnaissant cela, la vue dite hyperbolique pourrait ne pas en être une à proprement parler, parce que sa lentille virtuelle pourrait ne pas être ronde, mais plutôt adopter une forme favorisant la lecture des éléments sémantiquement proches de la tâche au centre de l'écran. Dans le cas d'une AHT en représentation graphique, par exemple, les tâches significativement proches d'une tâche donnée sont son parent, ses sœurs et ses enfants, ces derniers étant généralement disposés latéralement. La lentille virtuelle de la vue dite hyperbolique pour l'AHT est donc un ovale plus large que haut, comme illustré à la figure 3-1.

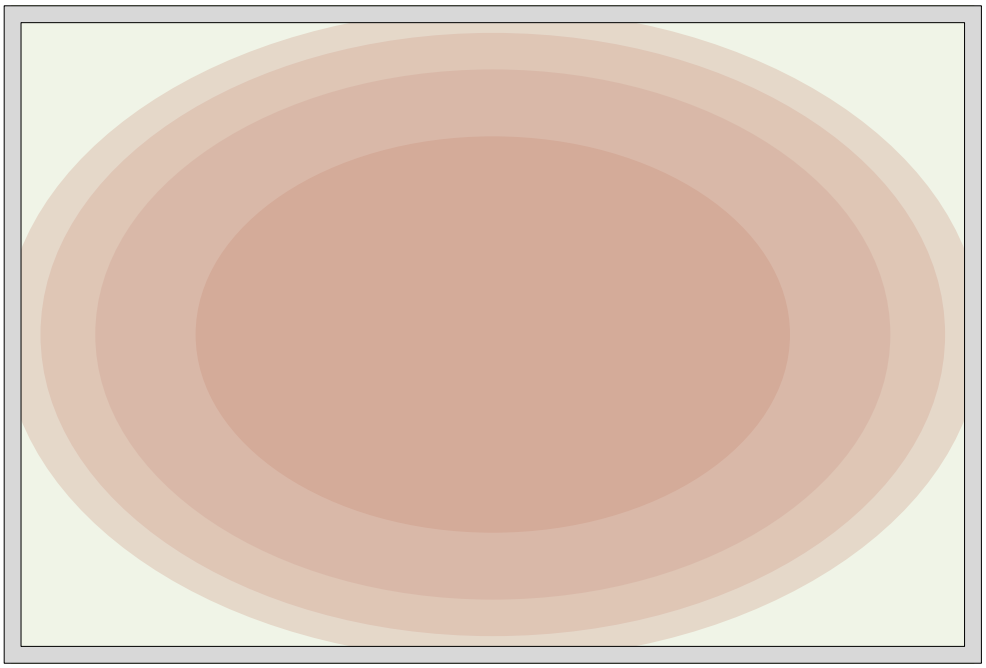


Figure 3-1 Figuration de la lentille virtuelle de la vue dite hyperbolique, dans laquelle les parties plus pâles montrent où les éléments à l'écran sont le plus rapetissée et où les déformations graphiques sont plus importantes.

La figure 3-2 fournit un exemple de cette fonctionnalité. L'auteur demande ici la clémence des lecteurs, puisqu'aucun logiciel n'a été utilisé pour effectuer les calculs pour déterminer les véritables coordonnées et dimensions de chaque élément apparaissant dans la fenêtre. En effet, le tout a été construit de manière approximative « à la main ». De surcroît, la fenêtre est

relativement petite ce qui ne favorise pas ce genre de vue qui profite beaucoup de grands espaces. Cela dit, l'idée de base devrait tout de même être bien compréhensible.

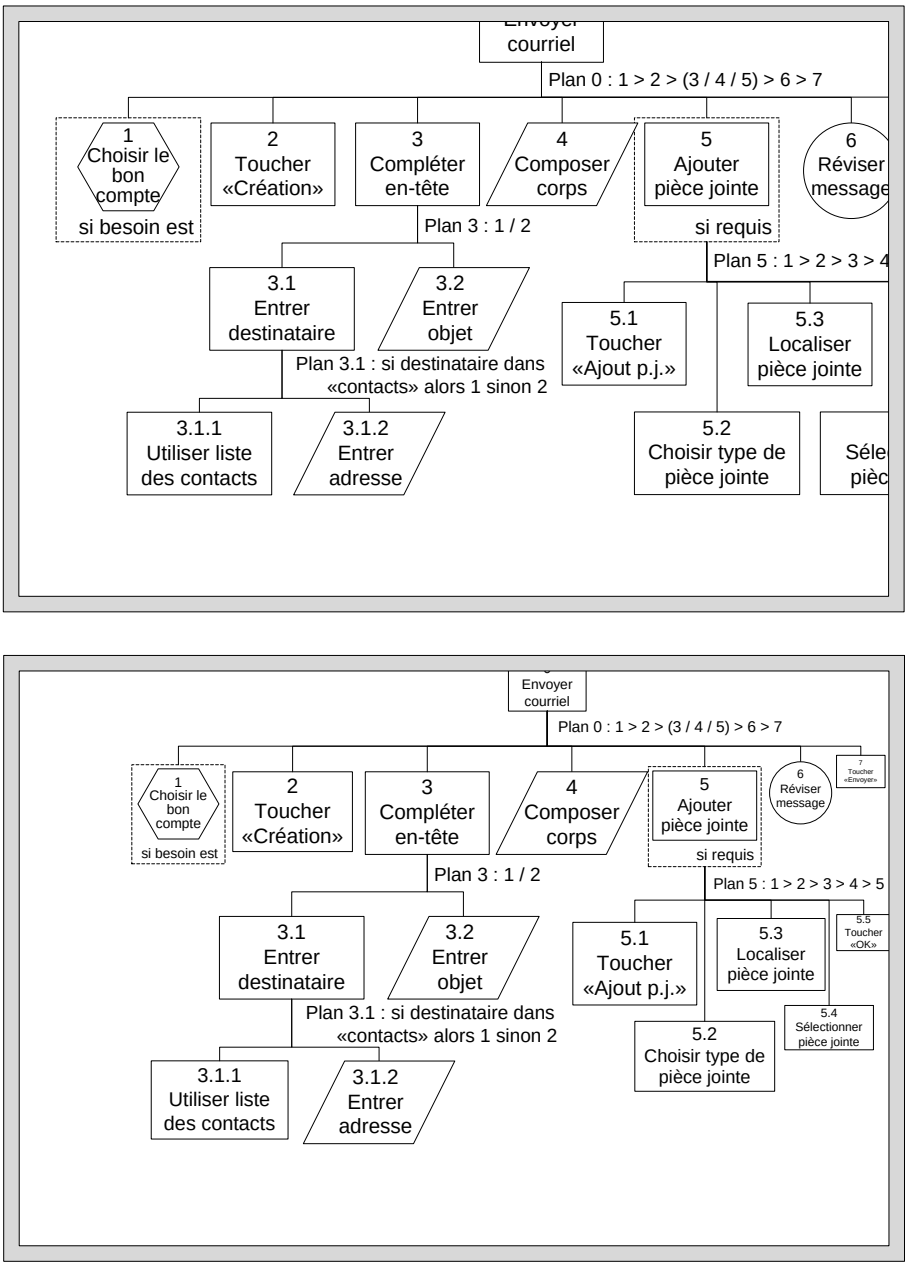


Figure 3-2 Simulation des différences entre la vue normale et celle dite hyperbolique. Dans ce cas, c'est la tâche « 3.2 Entrer objet » qui est le point stable entre les vues.

En terminant avec ce sujet, notons que la vue hyperbolique ne sert qu'à la navigation. Toute modification doit se faire à partir de la vue normale.

3.2 AHT

3.2.1 Processus de construction d'une AHT

La construction d'une AHT est un processus récursif commençant par le but de la tâche à analyser.[43] À partir de là, il faut décomposer chaque but ou tâche jusqu'à la réalisation des critères d'arrêt de l'analyse de tâche qui ont été exposés au chapitre 1.

Notons ici que, lorsque l'utilisateur requiert la création d'une nouvelle AHT, @Esperanto lui demande immédiatement les informations relatives au but de la tâche à analyser. Conséquemment, dès sa création, une AHT contient ce but qui est son point de départ obligé.

Pour chaque but ou tâche à décomposer, il faut faire deux choses. D'abord, il faut indiquer les sous-buts ou les tâches qui pourraient intervenir dans la réalisation de ce que l'on veut décomposer. Ensuite, il faut définir le plan spécifiant la procédure par laquelle les sous-buts ou tâches indiquées plus tôt vont permettre d'accomplir ce que l'on veut.

La création d'une AHT peut donc commencer très tôt dans le processus d'analyse de tâche et se poursuivre aussi longtemps qu'il est nécessaire pour obtenir le niveau de détail correct selon les objectifs de l'analyse à effectuer.

Les deux étapes de la décomposition d'un but ou d'une tâche ne sont liées que par le fait que le résultat de la première constitue les données de base de la deuxième. Cela n'implique pas qu'il faille absolument terminer la première pour commencer la deuxième; seulement que la deuxième ne peut être complétée qu'en dernier. @Esperanto se doit donc de permettre le va-et-vient entre ces deux étapes tout en indiquant de façon claire l'existence de plans incomplets (par l'ajout de sous-tâches) ou invalides (par le retrait de sous-tâches mentionnées dans le plan).

3.2.1.1 Fenêtre principale

La fenêtre principale pour la visualisation et l'édition des AHT doit offrir un maximum d'espace pour les afficher tout en montrant les stencils comportant les symboles qu'elles pourraient

---

[43] Lorsque l'analyse a commencé par une sous-tâche quelconque du véritable but à étudier, @Esperanto permet la création d'une nouvelle racine pour celui-ci et le rattachement de l'arbre d'analyse courant à cette nouvelle racine.

utiliser. Elle pourrait ressembler à celle de la figure 3-3, qui présente l'aspect initial d'une AHT. La raison pour laquelle toute nouvelle AHT comporte une racine dès sa création est qu'@Esperanto requiert de l'utilisateur les renseignements qui la concernent en réponse à la commande de création.

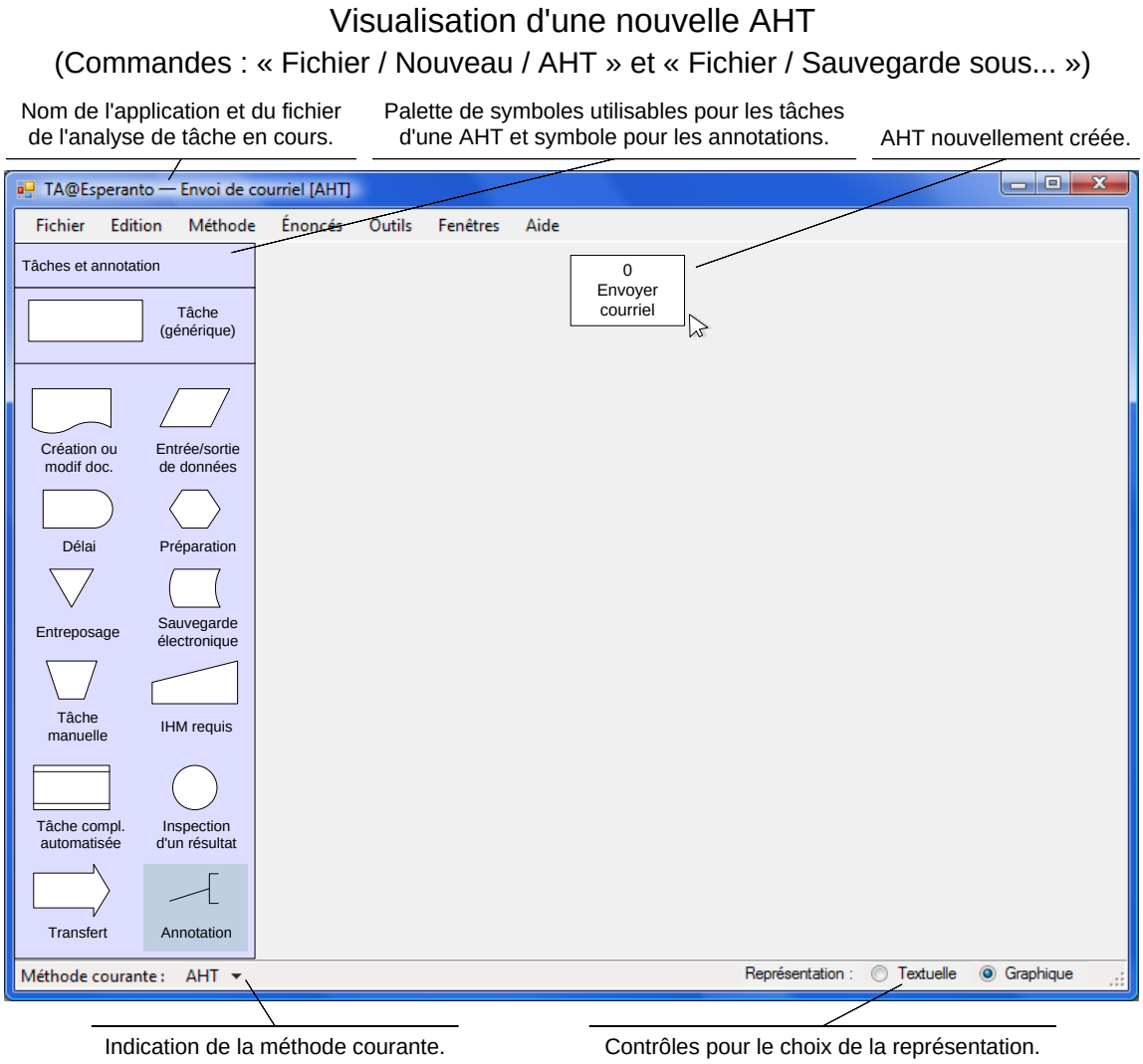


Figure 3-3 Fenêtre pour la visualisation et l'édition d'AHT.

Dans cette fenêtre, on voit que la palette montre tous les symboles pouvant illustrer l'aspect dominant d'une tâche (comme cela se fait couramment avec les ordinogrammes), ainsi que le symbole permettant d'ajouter des annotations dans le but d'expliquer divers points d'une analyse.

Bien qu'il soit possible de créer une AHT à partir de cette fenêtre, ce serait un processus assez long. Conséquemment, une fenêtre spécialisée pour l'entrée d'une grande quantité de données est proposée.

3.2.1.2 Fenêtre favorisant l'entrée de masse

Typiquement, les premières étapes d'une analyse de tâche consistent à déterminer exactement ce qu'il faut faire et comment il faut le faire. Cela fait, une méthodologie expérimentale est mise au point pour l'acquisition des connaissances nécessaires à la réalisation de l'analyse. Ensuite, une grande quantité d'information est accumulée par le biais de multiples techniques, dont l'observation, les interviews (structurées ou non), la prise de mesures et l'étude de comptes-rendus ou de manuels pertinents. C'est généralement après avoir accumulé bon nombre de données que le premier jet de l'arbre d'analyse hiérarchique est prêt à être entré dans un logiciel.

La fenêtre d'@Esperanto prévue pour l'entrée d'un grand volume de données dans une AHT tire profit de la représentation textuelle (tabulaire) parce qu'elle s'y prête mieux que sa représentation graphique. Le contenu suggéré pour cette fenêtre est comme suit :

<numéro et nom des tâches aïeules>

*<nom et numéro de la **tâche courante** ainsi que de ses soeurs>*

<plan associée à la tâche courante> ...

#	Nom	Fac./Cond.	Type	*<attr. étude #1>*	*<attr. étude #2>*
1	*<nom du premier enfant>*				
2	*<nom du deuxième enfant>*				
	...				

Figure 3-4 Contenu suggéré de la fenêtre d'acquisition d'une grande quantité de données.

En regardant la figure 3-4, on constate ce qui suit :

- Le contexte de la tâche courante, soit celle pour laquelle on peut définir et voir les enfants, est toujours disponible sous la forme de deux énumérations : celle de ses parents à partir de la racine et celle de ses sœurs.
    - Parmi toutes ces tâches, seule la tâche courante est mise en évidence.
    - Au besoin, ces bandes peuvent être agrandies pour contenir plus de lignes permettant ainsi de lister tous les parents et toutes les sœurs.
- Le plan associé à la tâche courante est visible (comme d'habitude) entre celle-ci et ses enfants. Pour l'instant, il suffit de savoir que le plan peut être tapé directement dans ce champ (selon l'une ou l'autre des notations décrites au chapitre précédent) et que le bouton à sa droite appelle l'éditeur de plan décrit plus loin.
    - Si le plan venait à prendre trop de place, ce champ peut être agrandi pour contenir jusqu'à trois lignes. Autrement, il faudra faire appel à une infobulle en survolant le plan.
- Comme d'habitude dans la représentation textuelle, l'utilisateur a accès aux divers attributs qui ont été définis spécifiquement pour les besoins de l'étude courante (à raison d'un par colonne).
    - Cette section peut nécessiter l'utilisation de barres de défilement.
- L'utilisateur n'a pas besoin de gérer la numérotation des tâches, puisqu'elle est prise en charge par @Esperanto.
    - Dans le cas où l'utilisateur modifierait l'ordre des enfants (par insertion, destruction ou déplacement d'un enfant vers une autre position), @Esperanto modifiera le plan en conséquence.
- Tous les numéros et noms de tâche peuvent être considérés comme une sorte d'hyperliens, puisqu'un double clic sur l'un d'eux charge la tâche visée et en fait la tâche courante. L'exemple suivant illustre le concept.

Le tableau 3-1 montre un enchaînement d'étapes qui permet de définir l'AHT du chapitre précédent. Après la fermeture de la fenêtre d'entrée de masse, la fenêtre principale est de nouveau visible et, comme montré à la figure 3-5, affiche l'AHT qui a été décrite (en représentation graphique ou textuelle selon les contrôles dans le coin inférieur droit de la fenêtre principale).

Rappelons ici que, comme montré lors de la présentation des méthodes, la représentation graphique peut inclure autant d'attributs que désiré pour les tâches. Dans ce but, @Esperanto permet à ses utilisateurs, par l'utilisation d'une fenêtre de configuration spécialisée, de choisir les attributs à montrer pour chaque tâche dans la représentation graphique.

Tableau 3-1 Étapes nécessaires pour définir l'AHT du chapitre précédent.

État initial (après la commande « Fichier / Nouveau / AHT » et acceptation du dialogue qui s'en suit).

On se souviendra qu'@Esperanto demande les informations concernant le but de l'étude dès la création, d'où le fait que le nom de la racine est déjà placé (en fait, c'est la tâche courante).

0 / Envoyer courriel

#	Nom	Fac./Cond.	Type	Type d'entrée	Commentaire

Après la définition des enfants de la racine, puis de son plan, on arriverait à ceci.

Pour continuer, on pourrait double-cliquer sur l'enfant « Compléter en-tête », ce qui nous amènerait à l'état montré ci-après.

0 / Envoyer courriel

1 > 2 > (3 / 4 / 5) > 6 > 7

#	Nom	Fac./Cond.	Type	Type d'entrée	Commentaire
1	Choisir le bon compte	si besoin est	Prép	Sélection	Garder ce compte pour la prochaine fois.
2	Toucher «Création»			Clic (bouton)	
3	Compléter en-tête				
4	Composer corps		E/S		
5	Ajouter pièce jointe	si requis			Utilisation occasionnelle.
6	Réviser message		Insp		
7	Toucher «Envoyer»			Clic (bouton)	

Après la définition des enfants de « Compléter en-tête », puis de son plan, on arriverait à ceci.

Pour continuer, on pourrait double-cliquer sur l'enfant « Entrer destinataire », ce qui nous amènerait à l'état montré ci-après.

0 / Envoyer courriel

1 / Choisir le bon compte || 2 / Toucher «Création» || **3 / Compléter en-tête** || 4 / Composer corps || 5 / Ajouter pièce jointe || 6 / Réviser message || 7 / Toucher «Envoyer»

1 / 2

#	Nom	Fac./Cond.	Type	Type d'entrée	Commentaire
1	Entrer destinataire				
2	Entrer objet		E/S	Texte	

Tableau 3-1 Étapes nécessaires pour définir l'AHT du chapitre précédent. (suite)

Après la définition des enfants de « Entrer destinataire », puis de son plan, on arriverait à ceci.

Pour aller à la dernière étape, il serait possible de double-cliquer sur l'un ou l'autre des parents, puis de double-cliquer sur « Ajouter pièce jointe » (qui, selon la première tâche double-cliquée, est alors un enfant ou une sœur).

0 / Envoyer courriel – 3 / Compléter en-tête

1 / Entrer destinataire || 2 / Entrer objet

si destinataire dans «contacts» alors 1 sinon 2

#	Nom	Fac./Cond.	Type	Type d'entrée	Commentaire
1	Utiliser liste des contacts			Clic (élément)	
2	Entrer adresse		E/S	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.

Après la définition des enfants de « Ajouter pièce jointe », puis de son plan, on arriverait à ceci.

À ce point, l'entrée de l'analyse AHT est terminée.

0 / Envoyer courriel – 3 / Compléter en-tête

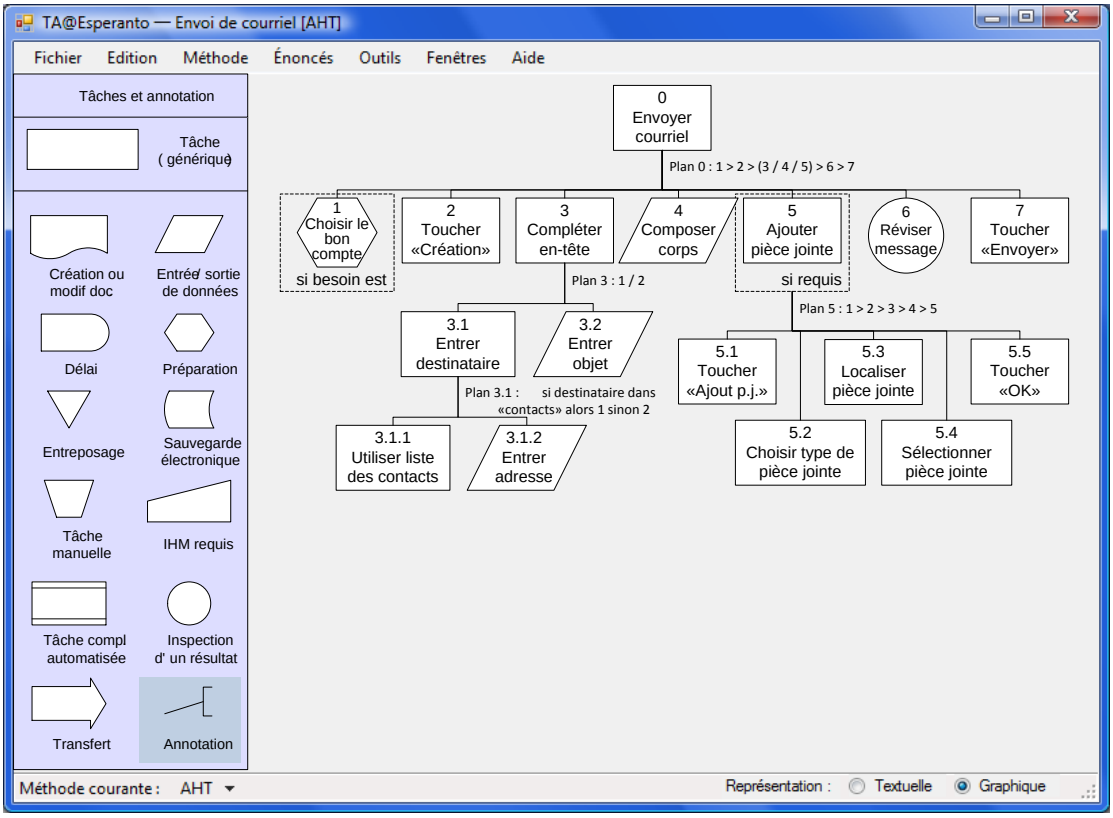
1 / Entrer destinataire || 2 / Entrer objet

si destinataire dans «contacts» alors 1 sinon 2

#	Nom	Fac./Cond.	Type	Type d'entrée	Commentaire
1	Utiliser liste des contacts			Clic (élément)	
2	Entrer adresse		E/S	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.

Cette fenêtre de configuration des attributs spécifiques à une étude permet aussi de choisir l'ordre dans lequel ces attributs seront montrés, ainsi que la forme sous laquelle chaque valeur sera montrée (nom complet, abréviation, ou icône).[44]

[44] Pour cette dernière fonctionnalité, il sera nécessaire de définir les valeurs possibles d'un attribut avec leur nom, leur abréviation et leur icône. Ceci devrait se faire sensiblement au même endroit que celui où l'on définit les attributs spécifiques à une étude. Idéalement, tout ce qui concerne un attribut devrait pouvoir être sauvegardé indépendamment d'une étude de manière à faciliter leur utilisation dans plus d'une étude.



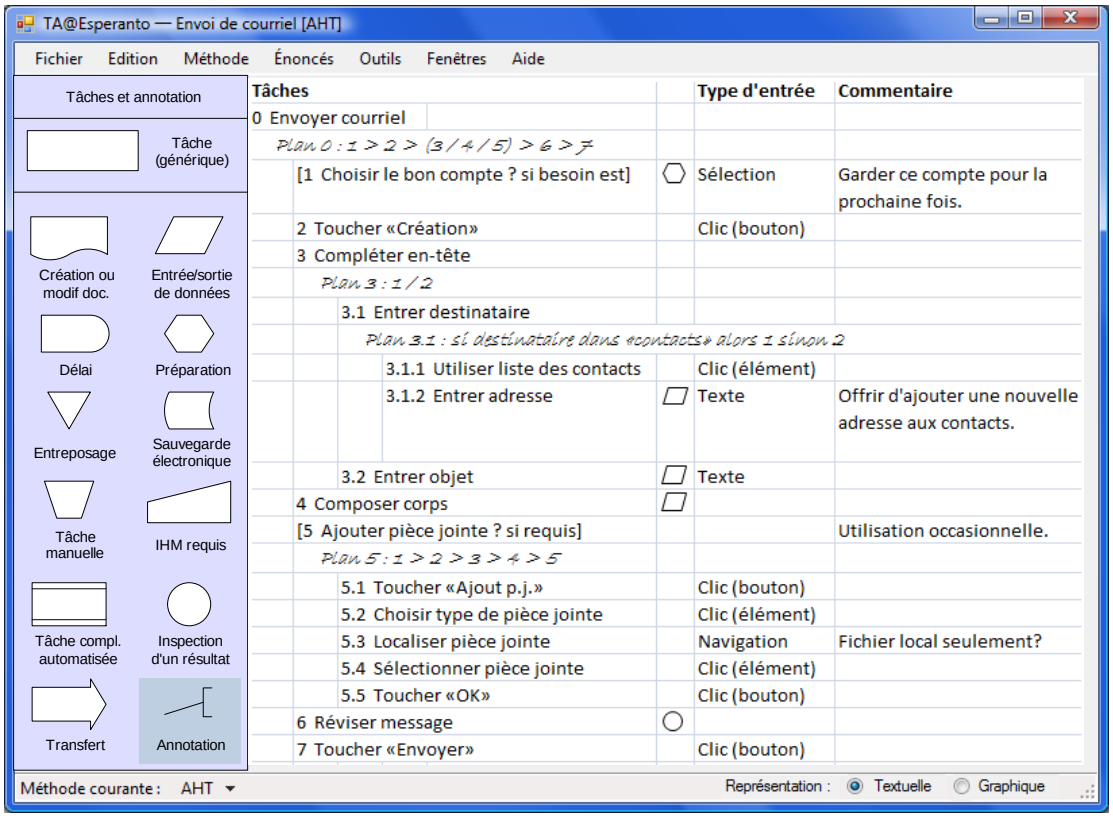


Figure 3-5 Représentations graphique et textuelle de l'AHT comme entrée jusqu'ici.

Notons aussi qu'en réponse au survol d'une tâche en représentation graphique, @Esperanto en affiche tous les attributs dans une infobulle. Si l'utilisateur est satisfait de ce qu'il a vu, il peut fermer l'infobulle d'un clic gauche (ou en touchant « Échapp. »). Pour passer directement à une fenêtre qui lui permettra de modifier la valeur des attributs montrés, il suffit d'un clic droit (ou de toucher « Retour »).

3.2.2 Modification d'une AHT à partir de la fenêtre principale

3.2.2.1 En représentation textuelle…

Les opérations permettant de modifier une AHT sous forme textuelle devraient être assez évidentes, puisqu'elles correspondent essentiellement aux opérations de modifications d'une feuille de calcul comme celles d'Excel©… insertion d'une ligne, retrait d'une ligne, déplacement d'une ligne ou modification de la valeur de l'une quelconque des cellules. La particularité ici est que le contenu étant hiérarchique, les diverses actions qui viennent d'être mentionnées sont soumises aux contraintes habituelles en de telles circonstances. Ainsi :

- L'insertion d'une tâche, avant ou après la tâche courante, ne peut servir que pour ajouter une sœur ou un enfant à la tâche courante.
    - La racine ne peut pas avoir de sœurs puisqu'il s'agit de créer un arbre d'analyse et non une forêt d'analyse.
- La modification des divers attributs d'une tâche ne pose aucun problème sauf en ce qui concerne sa position dans l'arbre ce qui doit être géré de manière à préserver la validité de l'arbre.
- Le retrait d'une tâche entraîne le retrait de tous ses enfants. Selon la complexité du plan du parent, il se pourrait que celui-ci soit invalidé.
- Le déplacement d'une tâche sous un parent différent entraîne le déplacement du sous-arbre dont la racine est la tâche même qui est déplacée par l'utilisateur. Selon la complexité du plan du parent perdant un sous-arbre, il se pourrait que celui-ci soit invalidé. Quant au plan du parent le recevant, il sera certainement invalidé.

Quant à la rédaction d'un plan ou à sa modification, il est possible de l'effectuer sur place en cliquant sur la ligne qui lui est réservée. Un double-clic sur cette même ligne appelle plutôt l'éditeur de plan décrit plus loin.

3.2.2.2 En représentation graphique…

Deux façons de faire sont disponibles pour ajouter des tâches à une AHT. Si la première manière est employée, l'utilisateur sélectionne un symbole dans la palette, puis le fait glisser sur un nœud quelconque où il sera ajouté en tant que dernier enfant de celui-ci. La deuxième manière, qui permet plus de souplesse dans le positionnement de la nouvelle tâche, requiert de l'utilisateur qu'il survole un nœud, ce qui amène @Esperanto à montrer trois petites flèches bleues autour de la tâche ainsi désignée. L'utilisateur peut alors cliquer l'une de ces flèches pour ajouter une tâche générique (rectangle) comme le nouveau dernier enfant (flèche vers le bas) ou comme une sœur avant ou après le nœud désigné (flèche vers la gauche ou la droite). S'il faut insérer un autre symbole, il suffit de survoler la flèche correspondant à la position voulue, d'attendre que la mini palette apparaisse pour enfin cliquer sur le symbole désiré. La figure suivante illustre bien la deuxième façon de faire.

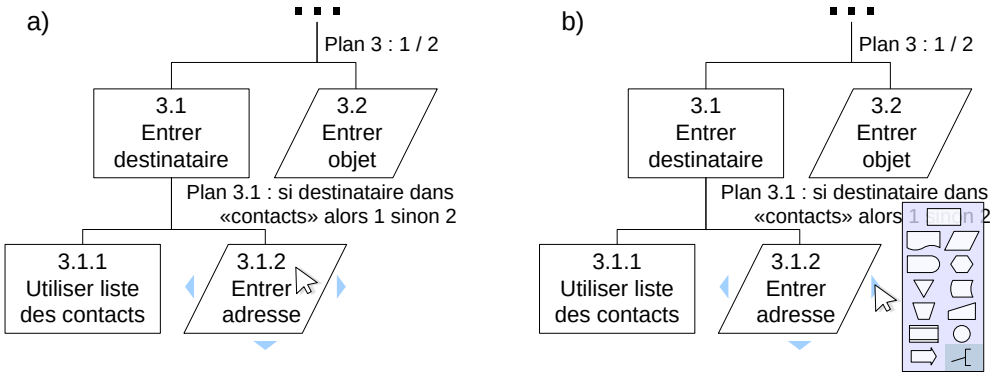


Figure 3-6 Outils pour l'insertion de tâches par survol.
a) Flèches bleues apparaissant lors du survol d'une tâche.
b) Mini palette (comme la grande) apparaissant lors du survol d'une flèche bleue.

Pour terminer et confirmer l'ajout d'une nouvelle tâche, l'utilisateur doit entrer les informations appropriées dont les attributs spécifiques à l'analyse courante, dans le mini dialogue qui apparaît alors (voir la Figure 3-7). Notons que, pour @Esperanto, seul le nom de la tâche est obligatoire.[45]

---

[45] Il serait possible de changer ce comportement en ajoutant la possibilité pour l'utilisateur, qu'un ou plusieurs des attributs spécifiques à l'analyse courante soient considérés comme obligatoires. Tant qu'à y être, on pourrait permettre aussi de spécifier les valeurs possibles, soit par une description lexicale, soit par énumération.

Nom	
Fac./Cond.	
Type	
Type d'entrée	
Commentaire	
Annuler	Créer la nouvelle tâche

Figure 3-7 Mini dialogue pour l'entrée des informations concernant une nouvelle tâche.

À propos de ce mini-dialogue, on notera que le combo « Type » est toujours correctement initialisé, dès l'ouverture du dialogue, mais qu'il peut servir à modifier le symbole choisi (sa liste contenant tous les symboles et un court texte). Aussi, la séquence des « TAB » pour se déplacer d'un champ à l'autre commence au Nom et descend jusqu'au bouton de création, puis (en tout dernier) va à « Annuler ».

La modification du nom d'une tâche peut se faire sur place simplement en cliquant dessus. Pour les autres attributs, il faut faire apparaître le mini dialogue montré ci-dessus[46], soit en double-cliquant sur la tâche appropriée, soit en utilisant le menu contextuel.

Le retrait d'une tâche se fait, comme d'habitude, en sélectionnant une tâche puis en pressant la touche « Suppr » ou à partir du menu contextuel. Comme mentionné plus haut, le retrait d'une tâche entraîne celui de ses enfants, la renumérotation automatique de toutes ses sœurs qui étaient à sa droite et possiblement l'invalidation du plan du parent.

Le déplacement d'une tâche (et de tous ces enfants) peut se faire par glisser-déposer ou par couper-coller. Ici, encore, la numérotation est automatiquement mise à jour aux deux endroits impliqués et les plans sont invalidés pour forcer l'utilisateur à valider les modifications qu'@Esperanto aura réalisées.

Pour rédiger ou modifier un plan, il suffit de faire apparaître le menu contextuel d'un nœud pour appeler l'éditeur de plan décrit ci-après.

---

[46] Bien sûr, dans le cas d'une modification, le libellé du bouton inférieur droit est plutôt « Modifier la tâche ».

3.2.3 Éditeur de plan structuré

Voici finalement l'éditeur de plan qui est appelé lorsque l'utilisateur presse le bouton « … » de la fenêtre d'entrée de masse ou à partir de l'item approprié du menu contextuel des tâches dans la fenêtre principale.

Bien qu'il soit possible de taper directement un plan AHT, @Esperanto privilégie l'emploi de son éditeur de plan parce qu'il offre différents moyens de composer les plans adaptés à la complexité du plan qui doit être établi. Cet éditeur, montré à la figure 3-8, peut paraître complexe de prime abord, mais nous verrons qu'en fait il est très facile d'utilisation.

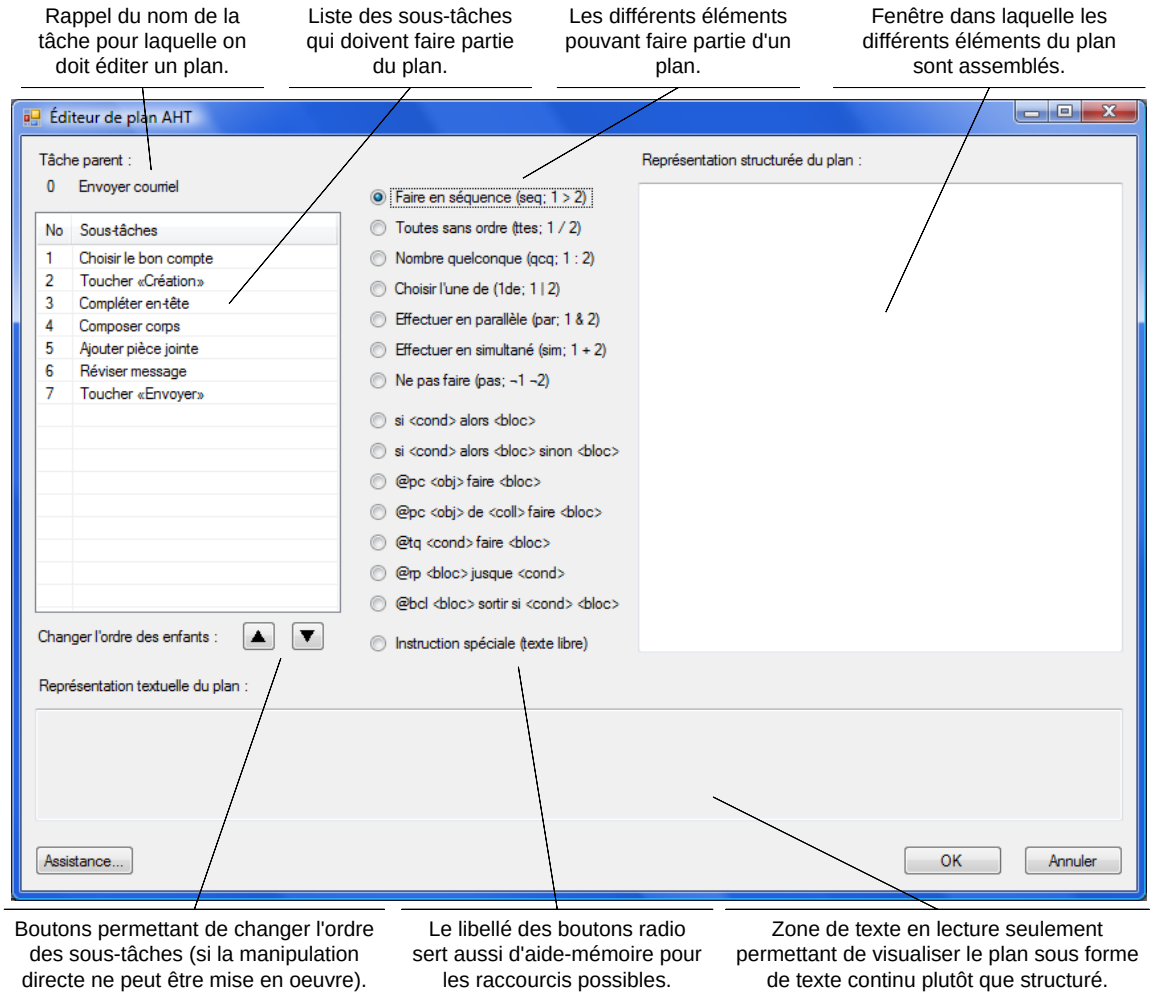


Figure 3-8 Éditeur de plan AHT favorisant leur création de manière structurée.

On note, dans l'éditeur de plan AHT, les composants suivants :

- Les numéros et noms de la tâche pour laquelle le plan est destiné.
- Les numéros (derniers éléments de la hiérarchie, seulement) et noms de chaque sous-tâche devant être utilisée dans le plan. Des boutons sont aussi présents pour permettre de changer l'ordre de ces sous-tâches (ce qui peut occasionner la mise à jour automatique du plan).
- Les divers éléments pouvant être utilisés dans un plan, qui peuvent être répartis en trois catégories :
    - Les enchaînements de tâches (les sept premiers).
    - Les énoncés conditionnels et itératifs (les sept suivants).
    - Les instructions en texte libre.
- Deux représentations du plan en construction, l'une d'elles étant en lecture seulement. La raison d'être de cet apparent dédoublement est que l'on favorise l'élaboration des plans d'une manière structurée, qui (dans le cas d'@Esperanto) peut être assimilé à une construction étape par étape dans laquelle l'utilisation de plusieurs lignes libère le créateur de l'obligation de s'occuper de considérations purement syntaxiques[47] pendant le processus de création. C'est @Esperanto qui va se charger de ces détails en mettant automatiquement à jour la représentation textuelle du plan qui, elle, sera affichée dans les autres fenêtres.

Parce que l'on veut favoriser la construction étape par étape des plans, chaque ligne du plan ne peut contenir plus d'un seul élément (de la colonne centrale de l'éditeur). Les énoncés conditionnels ou itératifs, eux, vont s'étaler sur plusieurs lignes de manière à ce que chacune de leurs portions génériques[48] soit sur sa propre ligne.

Avant de décrire le mode de construction structuré, mentionnons qu'il est toujours possible de composer un plan en le tapant directement dans le champ « Représentation structurée du plan ». Mais, il est possible de faire tellement mieux!

---

[47] Ici, on pense surtout aux accolades indiquant la portée des différentes parties des énoncés de branchement et d'itération.

[48] Par portion générique, on réfère aux parties d'un énoncé que l'utilisateur doit définir (condition, bloc, etc.).

Dans le cas de plans ne comportant que divers enchaînements de tâches (à raison d'un par ligne), il est possible pour chacun de :

- Taper son abréviation suivie d'une liste de tâches.[49]
- Taper l'enchaînement désiré en utilisant le bon séparateur.
- Sélectionner les tâches impliquées dans la liste de gauche, puis choisir l'enchaînement désiré dans la colonne centrale.

Les deux premières manières sont probablement les plus rapides, puisque tout se passe dans le champ d'édition du plan structuré. La dernière méthode est très simple et le fait de sélectionner chaque tâche aide vraisemblablement à réduire les erreurs qui pourraient être commises en tapant les numéros de tâche.

Pour ajouter un énoncé conditionnel ou itératif à un plan, il suffit d'insérer une ligne à l'endroit requis puis de choisir l'énoncé désiré dans la colonne centrale. @Esperanto dispose alors les divers éléments de l'énoncé sur autant de lignes que nécessaire après quoi il reste à l'utilisateur l'obligation de définir chaque portion générique pour compléter son énoncé.[50]

Finalement, pour ajouter une instruction spéciale (forcément un texte en langage naturel), il suffit de l'entrer sur sa propre ligne. En effet, lorsque @Esperanto ne reconnaît pas le texte entré sur une ligne, il sélectionne automatiquement le dernier item de la colonne centrale pour cette ligne, soit « Instruction spéciale (texte libre) ». Si toute l'instruction spéciale était exprimée en langue naturelle, @Esperanto serait incapable d'identifier les références à des sous-tâches qu'elle pourrait contenir, l'empêchant ainsi de vérifier si le plan est bien complété. Pour pallier ce problème, dans une telle instruction, il faut référer aux tâches par leur numéro entouré de « -T » et de « - » comme, par exemple, dans « À tout moment, être prêt à faire -T3- sur ordre du superviseur. »

---

[49] L'abréviation de chaque enchaînement apparaît entre parenthèses dans la colonne centrale et les listes de tâches sont comme définies au chapitre précédent (par exemple, 1-3, 5 pour les tâches 1, 2, 3, et 5).

[50] On pourrait aussi avoir des codes pour ces énoncés. Par exemple, « si1 » pour ajouter un « si … alors », « pcde » pour ajouter un « pour chaque … de », etc. Ainsi, il serait possible d'éviter des allers-retours entre le plan et la colonne centrale.

Dans la fenêtre d'édition, comme d'habitude, le retour de chariot permet d'ajouter des lignes alors que les flèches permettent de déplacer le point d'insertion dans le texte. Pour modifier la position ou l'indentation d'un élément du plan, il faut d'abord y mettre le point d'insertion ou, pour affecter plusieurs éléments contigus, sélectionner au moins une portion de tous ceux-ci. Par la suite, Ctrl-Haut et Ctrl-Bas vont déplacer les éléments visés vers le haut ou le bas, alors que Ctrl-Gauche et Ctrl-Droite vont modifier l'indentation lorsque possible (pour, par exemple, indiquer que des éléments font ou non partie d'un bloc « sinon » ou de l'énoncé suivant).

Bien sûr, @Esperanto va s'assurer que toutes ces modifications préservent la structure des différents éléments. Par exemple, il sera impossible de déplacer un bloc « sinon » avant le bloc « alors » du même « si », mais permis de le déplacer après le bloc « alors » d'un autre énoncé « si » n'ayant pas lui-même de bloc « sinon ».

3.2.4 Éditeur graphique de plan

Pour les plans complexes, il serait préférable d'utiliser un éditeur graphique plutôt qu'une version textuelle, même structurée.

Parce que l'on n'a pas encore toute l'information requise pour voir cela, ce sujet ne peut qu'être abordé plus loin.

3.3 OTHI

3.3.1 Processus de construction d'un OTHI

Un OTHI étant une procédure exprimée de manière préférentielle sous forme d'organigramme, son processus de construction n'en demeure pas moins celui d'une procédure bien ordinaire, quelle qu'en soit la représentation choisie. C'est un processus pas-à-pas dans lequel les éléments à exécuter sont placés les uns à la suite des autres du début de la procédure jusqu'à sa fin.[51]

---

[51] Pour les puristes, un ordinogramme devrait n'avoir qu'un seul symbole de fin. En réalité, la fin d'un fil d'exécution peut survenir à plusieurs endroits de l'ordinogramme. Dans une petite procédure, on peut satisfaire les puristes en traçant un lien entre le (seul) symbole de fin de la procédure et toutes les tâches se trouvant à la fin d'un fil d'exécution. Dans une grosse procédure, qui devrait être imprimée sur plusieurs pages, à la fin de chaque fil

Bien qu'un OTHI soit une procédure et que son processus ne soit pas vraiment récursif, il est fort possible que le premier jet utilise des tâches de haut niveau qui pourraient être raffinées au cours de l'analyse. Le fait reste qu'ici comme pour une AHT, il faut décomposer chaque tâche jusqu'à la réalisation des critères d'arrêt de l'analyse de tâche qui ont été exposés au chapitre 1.

La création d'une OTHI peut donc commencer très tôt dans le processus d'analyse de tâche pour autant que l'on se satisfasse d'énoncés de haut niveau qui devront être raffinés avec l'accroissement de la compréhension de ce qui doit être fait. Ici encore, il faut poursuivre aussi longtemps qu'il est nécessaire pour obtenir le niveau de détail correct selon les objectifs de l'analyse à effectuer.

3.3.2 Fenêtre principale

La fenêtre principale pour la visualisation et l'édition des OTHI doit offrir un maximum d'espace pour les afficher tout en montrant les stencils comportant les symboles qu'elles pourraient utiliser. Elle pourrait ressembler à celle de la figure 3-9, qui présente l'aspect initial d'un OTHI, soit trois symboles : un de début, un nuage (identifiant un endroit où devra apparaître un énoncé) et un de fin. On notera aussi que le nuage est présélectionné, puisqu'il doit être remplacé par un symbole valide pour créer un OTHI valide.

---

d'exécution, il faudrait mettre un symbole de référence vers une autre page pour aller au symbole de fin. Dans ce cas, c'est l'étiquette de ce symbole de référence qui serait interprétée comme étant la fin de la procédure.

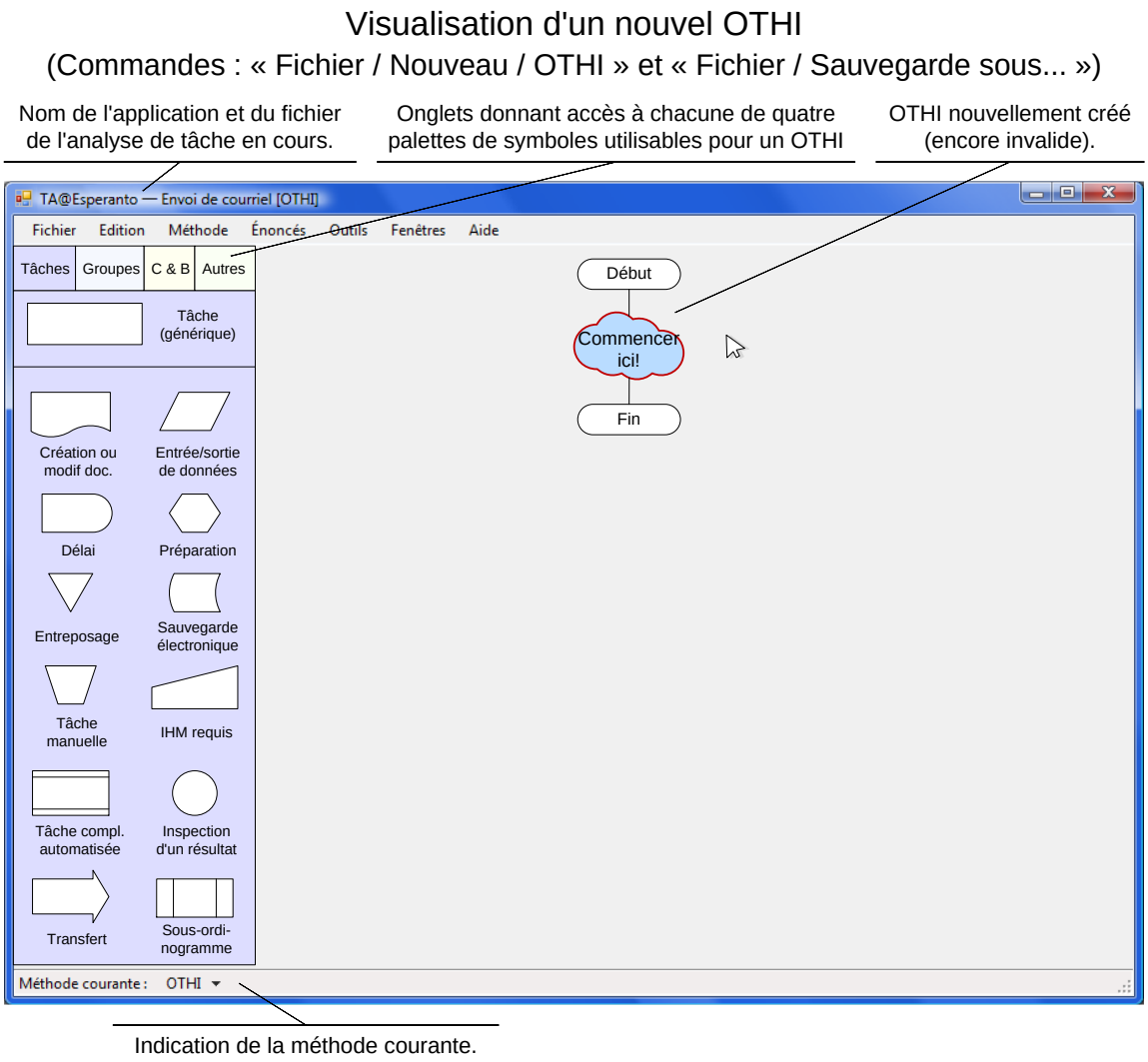


Figure 3-9 Fenêtre pour la visualisation et l'édition d'un OTHI.

Dans cette fenêtre, on voit que la palette montre tous les symboles pouvant illustrer l'aspect dominant d'une tâche, mais la série d'onglets au-dessus de celle-ci nous fait appréhender l'existence de trois autres palettes. Ces palettes, montrées à la figure 3-10, contiennent le reste des symboles qui pourront être placés dans l'OTHI (en accord avec la normalisation effectuée au chapitre précédent).

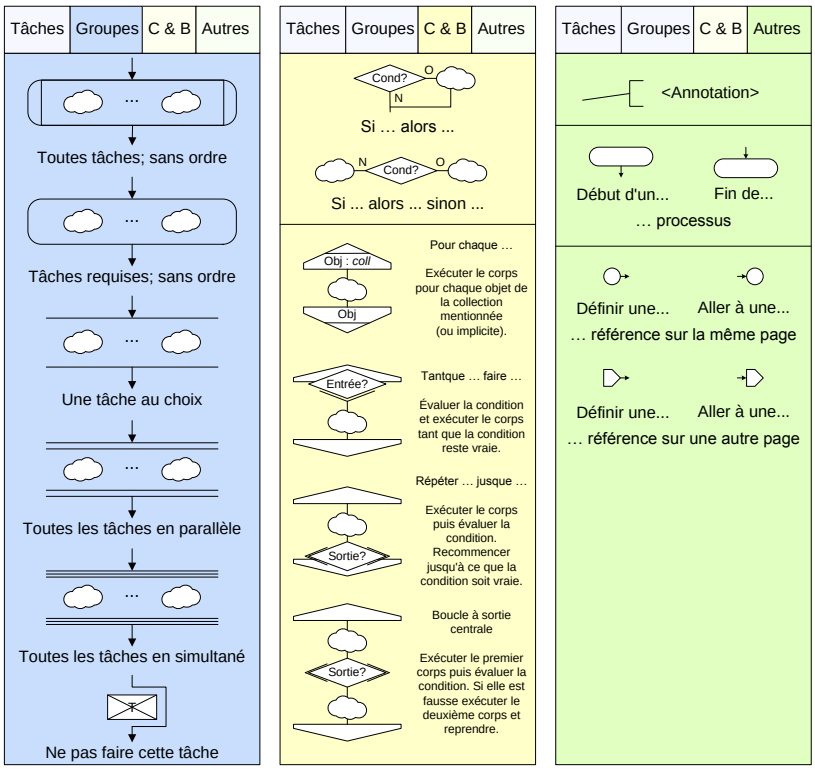


Figure 3-10 Les trois autres palettes pour la construction d'OTHI.

3.3.3 Modification d’une OTHI à partir de la fenêtre principale

3.3.3.1 En représentation graphique (la représentation préférée pour cette méthode)

Trois façons de faire, expliquées ci-dessous, sont disponibles pour ajouter des tâches à un OTHI. Deux d’entre elles ne sont pas très détaillées parce que très semblables à celles vues pour l’AHT.

- Sélection d’un symbole dans l’OTHI, suivi de la sélection d’un autre symbole dans l’une des palettes pour provoquer le remplacement.
  En particulier, avec le nuage de l’OTHI qui a été créé et sélectionné automatiquement, il suffit de sélectionner le symbole du « SI … ALORS » pour causer le remplacement et entraîner l’apparition d’un dialogue pour entrer les informations complémentaires au nouveau symbole (voir la figure 3-11).
- Sélection d’un symbole dans la palette, suivi du glissement de celui-ci sur un lien entre deux blocs déjà en place, causant l’insertion du symbole choisi entre ces blocs.

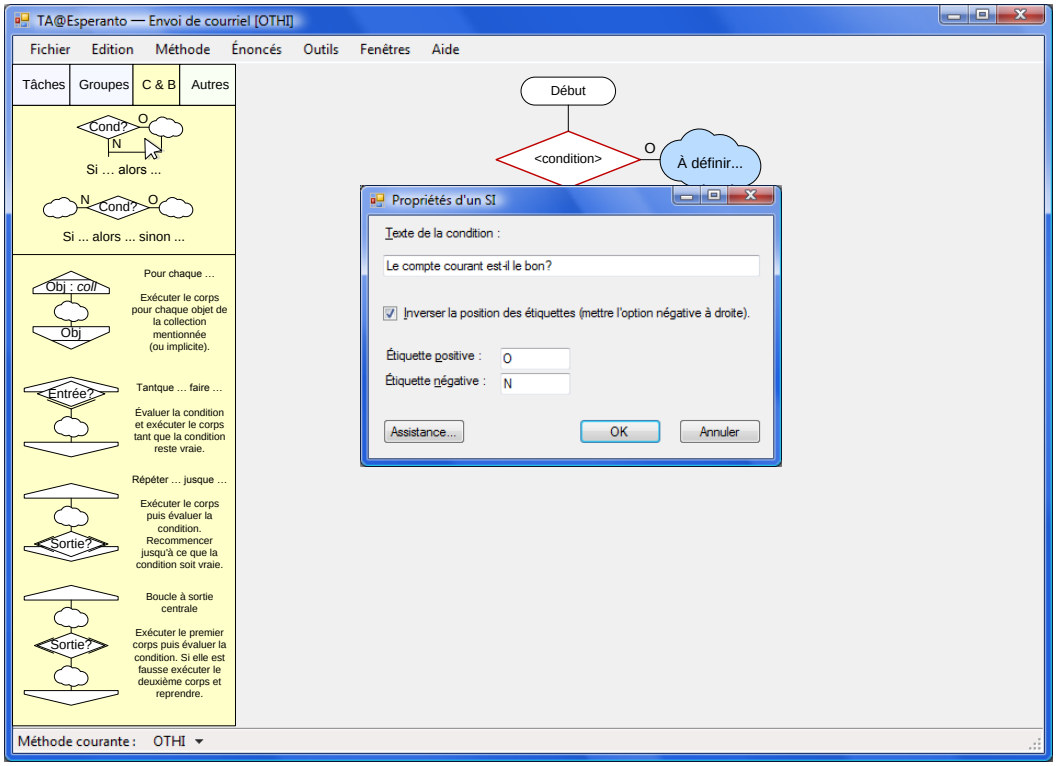


Figure 3-11 Dialogue acceptant les informations relatives à tout nouveau symbole « SI ».

- Survol d'un symbole (@Esperanto montrant alors de petites flèches bleues autour de ce symbole), suivi d'un clic sur l'une de flèches pour ajouter une tâche générique (rectangle) dans le sens de la flèche par rapport au symbole survolé. Une variante de cette façon de faire est de survoler une flèche bleue pour laisser apparaître les mini palettes permettant de choisir le symbole voulu. Ici, la différence avec ce qui se passe en AHT est qu'il y a deux étapes dans lesquelles on choisira d'abord l'onglet désiré (première mini palette), puis le symbole voulu dans une deuxième mini palette dont le contenu dépend de la sélection faite dans la première. La figure 3-12 illustre cette variante.

Pour terminer et confirmer l'ajout ou le remplacement d'un nouveau symbole, l'utilisateur doit entrer les informations spécifiques qui le concernent. Pour cela, il existe un dialogue pour chaque symbole nécessitant des informations autres que le nom qui, lui, peut être tapé directement dans la forme graphique.

Par exemple, on a déjà vu le dialogue qui apparaît quand on insère un symbole commençant par « SI » et on peut voir à la figure 3-13 celui qui apparaît après l'insertion d'un bloc comportant un groupe d'énoncés (comme celui indiquant d'effectuer toutes les tâches indépendamment de l'ordre).

c

Figure 3-12 Outils pour l'insertion de symboles par survol.
a) Flèches bleues apparaissant lors du survol d'une tâche.
b) Mini palette montrant les quatre onglets apparaissant au-dessus des palettes d'outils.
c) L'utilisateur ayant choisi l'onglet «Groupes», la mini palette correspondante apparaît.

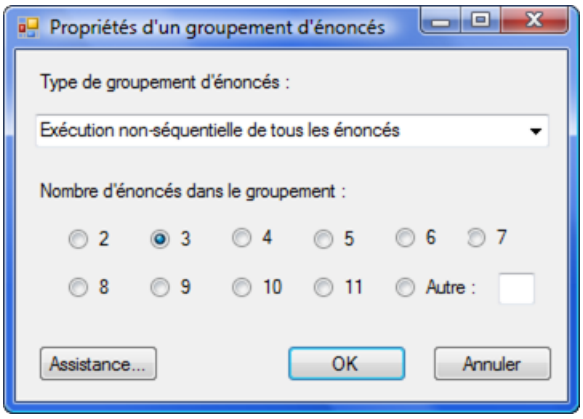


Figure 3-13 Dialogue des propriétés d'un symbole de groupement.

La modification du texte à l'intérieur d'un symbole peut se faire sur place simplement en cliquant dessus. Pour les autres attributs, il faut faire réapparaître le mini dialogue qui lui est spécifique, soit en double-cliquant sur le symbole approprié, soit en utilisant le menu contextuel.

Le retrait d'un symbole se fait, comme d'habitude, en le sélectionnant puis en pressant la touche « Suppr » ou à partir du menu contextuel. Parce que le retrait de certains symboles (énoncés conditionnels (« SI »), itératifs ou de groupe) peut entraîner celui d'énoncés dépendants, une boîte à message demande alors à l'utilisateur ce qu'il faut faire. Par exemple, pour un énoncé « SI … ALORS », @Esperanto doit savoir s'il faut retirer le « SI » et tous les énoncés du bloc « ALORS » ou alors s'il faut n'enlever que la condition de manière à ce que les énoncés du bloc du « ALORS » deviennent inconditionnels.

Le déplacement d'un ou plusieurs symboles (incluant d'éventuelles dépendances) peut se faire, après sélection des symboles voulus, par glisser-déposer ou par couper-coller. Ce peut être utile pour sortir des énoncés d'un énoncé avec dépendances.

Il suffit donc de quelques opérations pour obtenir le résultat montré à la figure 3-14.

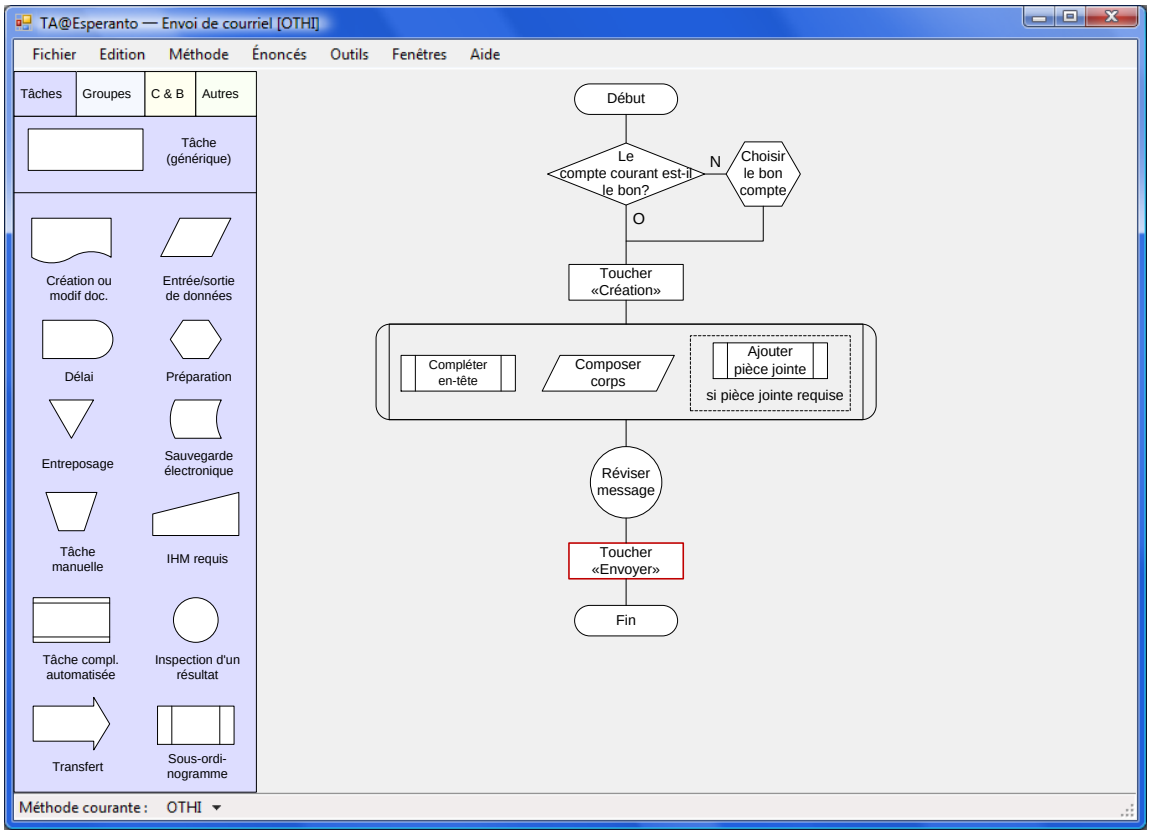


Figure 3-14 OTHI pour la tâche consistant à envoyer un courriel (utilisant deux sous-ordinogrammes non montrés).

Mentionnons en terminant avec la représentation graphique que, comme en AHT, il est possible d'ajouter aux tâches autant d'attributs que désiré, pas seulement un nom. Dans ce but, @Esperanto permet à ses utilisateurs, par l'intermédiaire d'une fenêtre de configuration spécialisée, de choisir les attributs à montrer pour chaque tâche dans la représentation graphique. Cette fenêtre (la même que pour l'AHT) permet aussi de choisir la forme (nom complet, abréviation, ou icône) et l'ordre dans lequel les attributs seront montrés.

Aussi et encore une fois, comme pour l'AHT, en réponse au survol d'une tâche en représentation graphique, @Esperanto affiche tous ses attributs dans une infobulle. Si l'utilisateur est satisfait de ce qu'il a vu, il peut fermer l'infobulle d'un clic gauche (ou en touchant « Échapp. »). Pour passer directement à une fenêtre qui lui permettra de modifier la valeur des attributs montrés, il suffit d'un clic droit (ou de toucher « Retour »).

3.3.3.1.1 Palettes considérées, mais rejetées…

Avant de décider en faveur des mini palettes dans l'IHM d'@Esperanto, d'autres possibilités ont été considérées.

En premier lieu, la possibilité la plus simple, un menu avec un item pour chacun des onglets (au-dessus) des palettes avec, bien sûr, une cascade donnant accès à tous les symboles pouvant apparaître dans un OTHI.

Ensuite, une palette circulaire contenant à elle seule tous les symboles a été considérée. Étant donné la surface requise, un ensemble de palettes circulaires en cascade (à l'image du menu envisagé plus tôt) a aussi été envisagé. Celles-ci étant tout de même assez grosses, il arriverait trop souvent que ces palettes ne puissent être centrées sur la position courante du pointeur de la souris. Il est même possible que ces palettes ne puissent être affichées de manière à ce qu'elles apparaissent sous le pointeur de la souris (lorsqu'elle est dans un coin, par exemple). Même si ce dernier problème peut être résolu par l'utilisation de palettes carrées, les mini palettes (de véritables miniatures des grandes) se sont imposées.

Pour référence, la figure 3-15 montre les palettes circulaires envisagées.

3.3.3.2 En représentation textuelle

La représentation textuelle d'un OTHI a ceci de particulier que l'une de ses utilisations possibles est de servir de support mémoriel ou de guide lors d'actions devant être prises dans diverses situations, soit d'urgence, soit rarement effectuées, soit devant être complétées d'une manière particulière. Utilisée ainsi, on parle de *procédures*. Plusieurs règles concernant leur rédaction et leur présentation limitent assez leur expressivité au profit d'une clarté nécessaire à leur utilisation

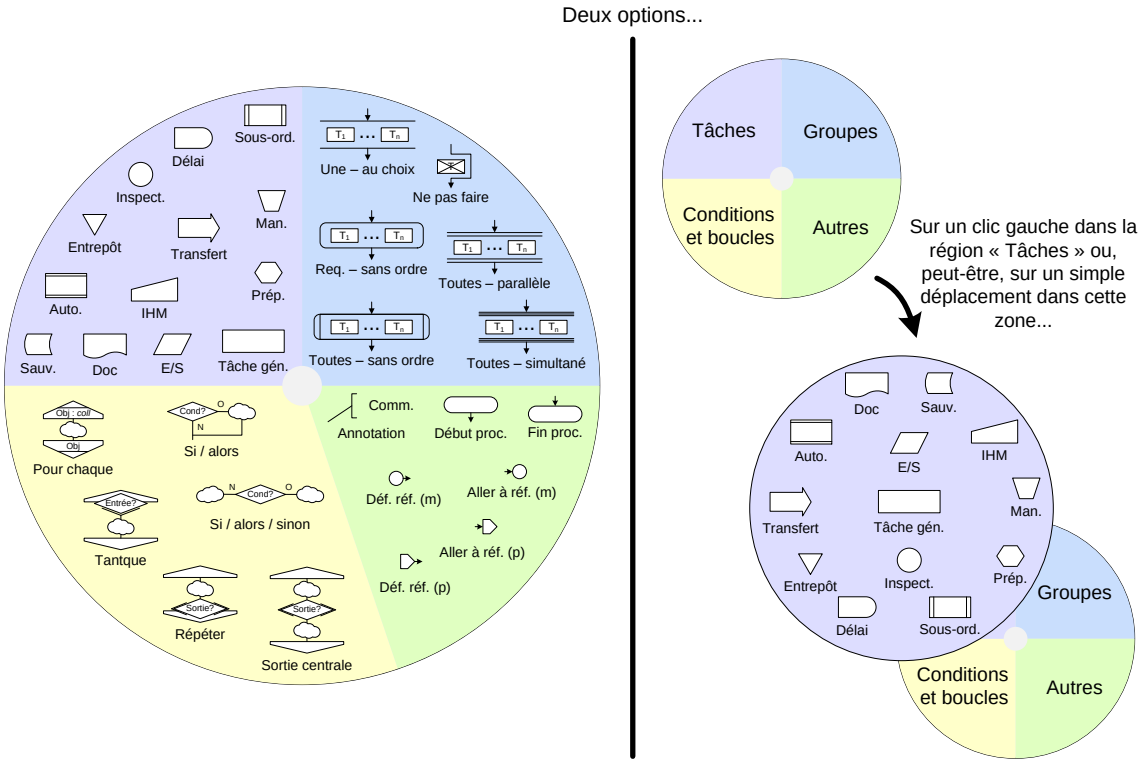


Figure 3-15 Palettes considérées pour l'insertion ou le remplacement de symboles.

dans ce contexte. L'ouvrage de Wieringa, Moore et Barnes (1999) constitue une très bonne référence en ce domaine, bien qu'il n'y soit pas fait mention des ornements graphiques pouvant améliorer sensiblement la lisibilité et le suivi des procédures au moment de l'exécution d'une action avec ce support à la performance humaine.

Cette section est divisée en deux parties. Dans la première, on aborde la représentation textuelle d'un OTHI en ne se préoccupant que de conserver l'expressivité de la méthode. Ce n'est que dans la deuxième partie que l'on aborde la conversion d'un OTHI en une procédure, soit un outil d'aide à la performance humaine.

3.3.3.2.1 Représentation textuelle pour l'analyse de tâche…

Comme montré précédemment, l'OTHI a été normalisé pour permettre d'exprimer tout ce qu'il est possible d'exprimer dans une AHT. Il est donc maintenant possible d'utiliser tous les termes définis pour la représentation textuelle des AHT pour la représentation textuelle d'un OTHI. Pour chaque symbole graphique, il y a donc au moins un équivalent lexical textuel (par exemple, au

losange correspond « Exécution conditionnelle », soit le « SI » avec ou sans une partie « SINON »).

Comme il a aussi été montré précédemment, pour convertir un OTHI sous forme d'AHT, il faudrait créer une racine dont les enfants seraient l'ensemble des tâches contenues dans l'OTHI et dont le plan serait l'OTHI lui-même.

En mettant ces deux informations ensemble, on conçoit facilement que l'édition d'un OTHI en représentation textuelle doit ressembler beaucoup à celle d'un plan AHT (lui aussi en représentation textuelle). L'idée de bâtir un OTHI d'une manière structurée est d'ailleurs tout aussi utile que pour un plan AHT.

L'éditeur d'OTHI en représentation textuelle est montré à la figure 3-16.

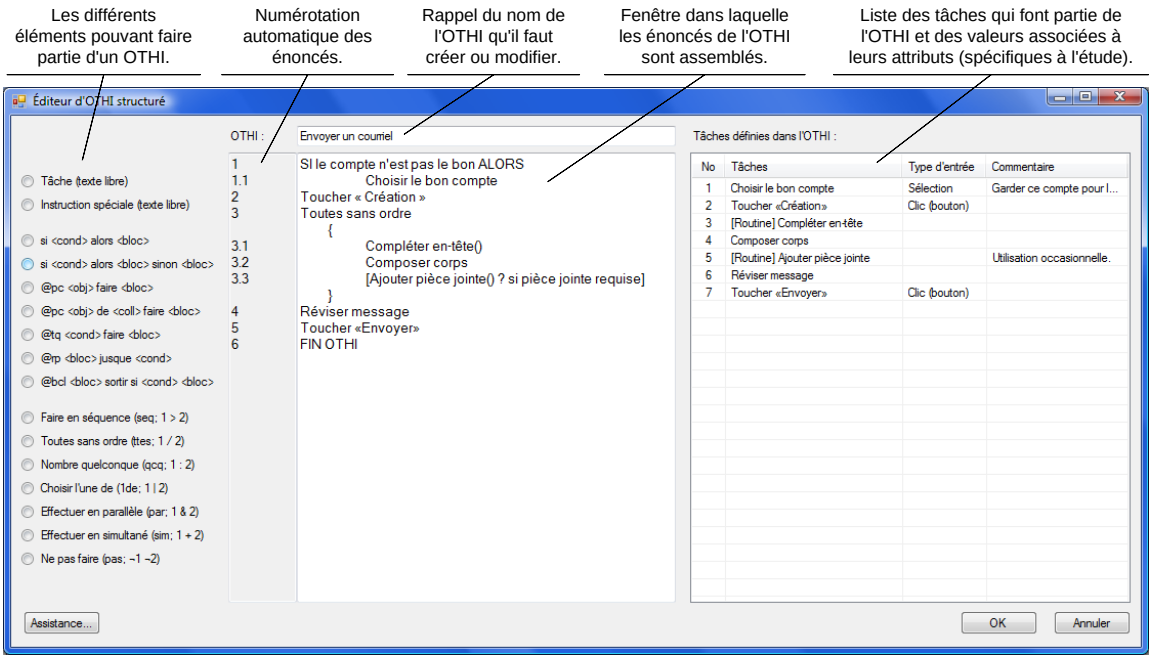


Figure 3-16 Éditeur d'OTHI en représentation textuelle.

À l'image de l'éditeur de plan structuré, chaque ligne de l'OTHI ne peut porter plus d'un énoncé. Un énoncé peut être une tâche, une instruction spéciale ou une consigne (condition, boucle ou enchaînement de tâches).

Selon la même inspiration, cliquer sur une ligne encore vierge de l'OTHI, puis choisir un bouton radio cause l'insertion du modèle correspondant sur cette ligne. Optionnellement, il est aussi possible de taper les énoncés les uns à la suite des autres. Dans ce cas, toute ligne portant une

consigne doit commencer par une séquence de caractères qui l'identifie. Ces séquences correspondent aux divers éléments de la notation textuelle des OTHI (et des AHT) vus au chapitre précédent. Les versions courtes sont d'ailleurs utilisées en guise de rappel dans le libellé des boutons radio. Par exemple, une ligne commençant par « SI » est automatiquement identifiée comme étant un énoncé conditionnel ne possédant pas de « SINON », ce qu'@Esperanto met en évidence en choisissant le bouton radio identifié par « si <cond> alors <bloc> ». Lorsqu'un « SINON » est nécessaire, l'utilisateur peut modifier le choix automatique en choisissant plutôt le bouton radio identifié par « si <cond> alors <bloc> sinon <bloc> ».

Toute ligne ne commençant pas par une séquence connue est automatiquement identifiée comme étant une tâche. Celles-ci sont automatiquement copiées dans la liste apparaissant à droite de l'OTHI, où l'utilisateur pourra assigner une valeur à chacun des attributs de tâche spécifiques à l'analyse en cours. Si un énoncé quelconque n'est pas une tâche et qu'il ne devait donc pas être copié dans la liste de tâches, l'utilisateur doit modifier le choix du bouton radio pour celui indiquant une instruction spéciale.

Les boutons radio peuvent aussi servir à changer un énoncé en un autre du même groupe sémantique.[52] Par exemple, il est possible de changer un « si <cond> alors <bloc> » en un « tantque <cond> faire <bloc> ». Un dialogue apparaît pour demander des précisions et une confirmation lorsque la transformation n'est pas aussi évidente que dans les exemples donnés jusqu'ici (par exemple, pour confirmer le retrait d'une condition pour transformer un « tantque <cond> faire <bloc> » en un « pour chaque <obj> faire <bloc> »).

De manière semblable, sélectionner une tâche ou un groupe d'énoncés puis choisir un bouton radio du groupe *condition et itération* cause l'insertion du modèle correspondant avec la sélection automatiquement placée à la position du premier <bloc> du modèle.

Les modifications peuvent aussi se faire comme il est courant dans plusieurs logiciels. Par exemple, par sélection suivie d'une commande (par le clavier ou le menu), par couper-coller ou par glisser-déplacer.

---

[52] Comme le groupement des boutons radio le suggère, il n'y a que trois groupes sémantiques, soit les tâches et instructions spéciales, les énoncés conditionnels et itératifs, ainsi que les enchaînements de tâches.

La numérotation est entièrement sous le contrôle d'@Esperanto qui assure toute renumérotation lorsque nécessaire. L'utilisation de nom pour les références à un autre endroit de l'OTHI permet aux utilisateurs de s'affranchir de tous ces problèmes.

3.3.3.2.2 Conversion d'un OTHI en une procédure

Après avoir créé un OTHI pour une tâche quelconque, l'utilisateur peut vouloir en déduire une procédure pour favoriser la bonne performance de ceux qui doivent la réaliser.

Étant donné le but de l'OTHI, il est très possible qu'il ne soit pas assez directif pour être efficace en tant qu'aide à la performance humaine. Ce pourrait être dû à la présence de tâches facultatives ou conditionnelles, ou alors d'enchaînements de tâches autres que la simple séquence (p. ex., l'exécution d'un nombre quelconque de tâches ou l'exécution d'un choix).

Notons que le problème des énoncés conditionnels, contrairement à ceux qui sont facultatifs, n'est pas l'ambiguïté puisqu'il est sémantiquement équivalent à un énoncé « SI … ALORS ». C'est plutôt que, dans une procédure, les conditions liées à l'exécution d'une tâche doivent apparaître avant celle-ci. Aussi, dans le but d'offrir une notation consistante, tous les énoncés conditionnels sont convertis en énoncés « SI … ALORS » de signification équivalente.

En l'absence d'énoncé introduisant une quelconque ambiguïté, le processus est évident puisqu'alors l'OTHI est une procédure dont il suffit de revoir le texte pour le rendre clair (pour le public visé) et le plus concis possible.

En présence d'ambiguïtés, @Esperanto doit pointer chacune à l'utilisateur et lui demander comment résoudre chacune. Les choix possibles sont :

- Préserver l'ambiguïté dans la procédure.
    - Un énoncé facultatif est alors précédé de l'étiquette « **[FAC]** ».
    - Un enchaînement autre que séquentiel est mis en évidence par l'utilisation d'une étiquette devant chaque énoncé concerné (p. ex. « **[ttes 1/3]** », « **[ttes 2/3]** », etc.) et de liens graphiques mettant en évidence les blocs indépendants dudit enchaînement.
        - Cette option devrait n'être utilisée que pour de minuscules blocs indépendants, idéalement d'une tâche par bloc.
- Réduire ou résoudre (enlever) l'ambiguïté.

- Un énoncé facultatif est alors remplacé par un énoncé d'exécution conditionnelle par la création d'une condition (p. ex., SI le temps le permet ALORS Nettoyer les moniteurs).
- Un enchaînement autre que séquentiel est transformé en un enchaînement séquentiel selon les indications de l'utilisateur qui décide ainsi de l'ordre dans lequel les opérateurs devraient réaliser la tâche analysée.

Le fait de devoir exécuter des tâches en parallèle ou en simultané n'est pas considéré comme étant ambigu. Pour la présentation de telles tâches (ou séquences de telles tâches), pour la section concernée, la largeur de la page sera divisée en autant de colonnes qu'il y a de fils d'exécution. Une colonne sera alors dédiée à chaque fil. Dans l'éventualité où ces colonnes deviendraient trop étroites, on créerait plutôt diverses sections à la file qui contiennent moins de colonnes. Par exemple, deux sections divisées en deux colonnes plutôt qu'une seule en comptant quatre.

Lorsque l'utilisateur en sent le besoin, @Esperanto lui permet d'insérer des titres correspondant à une portion de la procédure désignée par une sélection. Ainsi, après sélection de certains énoncés, il est possible d'assigner un titre à cette portion de la procédure. Comme mentionné au chapitre de normalisation, le but n'est pas ici de créer une hiérarchie moyens-finalités; il s'agit simplement de diviser la procédure à la manière d'un document technique avec ses chapitres, sections et sous-sections.

Il serait certes pratique qu'@Esperanto puisse assister l'utilisateur dans le processus de révision du texte d'une procédure, mais cela est hors de la portée des premières versions du logiciel.

3.3.4 Éditeur de plans AHT en représentation graphique

Ayant maintenant vu l'éditeur d'OTHI graphique et sachant qu'un OTHI correspond à un plan AHT[53], on voit bien comment on peut obtenir un éditeur de plans AHT graphique. Il suffit en effet de créer une fenêtre comportant l'éditeur d'OTHI graphique et un rappel des sous-tâches devant être mentionnées dans le plan AHT.

---

[53] Voir le « petit secret » du chapitre précédent.

Cela nous amène à nous demander comment représenter un plan particulièrement complexe : sous forme graphique ou textuelle? Histoire de mettre les choses au clair (plus ou moins!), on peut constater qu'un plan assez complexe n'est facilement lisible ni en représentation graphique (figure 3-17) ni en représentation textuelle (figure 3-18).

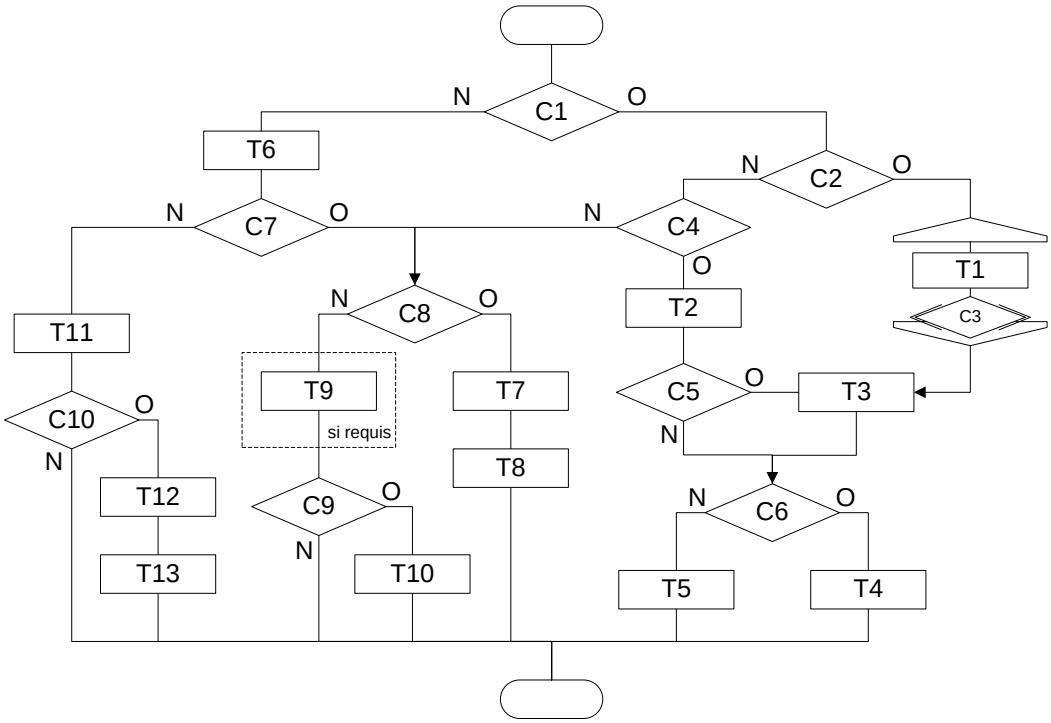


Figure 3-17 Plan assez complexe en représentation graphique.

Pour ce genre de situation, il pourrait être intéressant de considérer offrir la possibilité d'avoir des représentations hybrides, comme celle de la figure 3-19, où le plan est plus facilement compréhensible. Cela n'est toutefois pas nécessaire pour la première implémentation.

3.4 MAD

3.4.1 Processus d'élaboration d'une description de tâche selon MAD

Étant donné le peu de différences entre MAD et AHT, surtout après la normalisation expliquée au chapitre précédent, il va sans dire qu'avec @Esperanto le processus d'élaboration d'une description de tâche selon MAD est assez proche de celui de construction d'une AHT. Les

différences entre ces processus sont dues aux attributs de tâche spécifiques à MAD ainsi qu'à l'accent placé sur la définition de chaque tâche en elle-même.

Notons que, comme pour l'AHT, lorsque l'utilisateur requiert la création d'une description de tâche selon MAD, @Esperanto lui demande immédiatement les informations relatives au but de la tâche à analyser. Conséquemment, dès sa création, la description MAD contient ce but qui est son point de départ obligé (comme en AHT aussi, une nouvelle racine peut être créée plus tard).

```
SI C1 ALORS
 {
   SI C2 ALORS
     {  RÉPÉTER 1 JUSQUE C3 >   3 >   SI C6 ALORS 4 SINON 5  }
  SINON
    {
       SI C4 ALORS
         {  2 >   SI C5 ALORS 3 >   SI C6 ALORS 4 SINON 5  }
       SINON
         {
           SI C8 ALORS { 7 > 8 }
           SINON
               {  [9 ? si requis] >   SI C9 ALORS 10  }
         }
    }
 }
SINON
 {
    6 >
    SI C7 SINON
       {
          SI C8 ALORS { 7 > 8 }
          SINON
            {  [9 ? si requis] >   SI C9 ALORS 10  }
       }
    SINON
       {  11 >   SI C10 ALORS { 12 > 13 }  }
  }
```

Figure 3-18 Plan de la figure précédente en représentation textuelle.

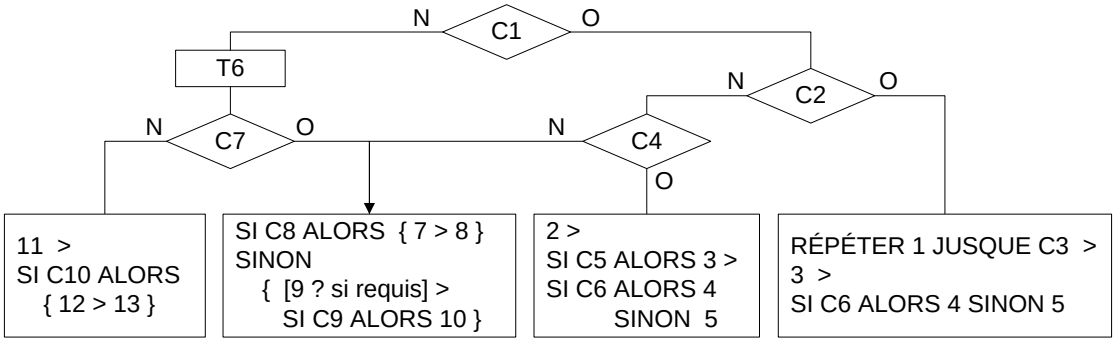


Figure 3-19 Plan des deux figures précédentes en représentation hybride.

Avec MAD, pour chaque but ou tâche à décomposer, il faut d'abord définir son contexte d'exécution sous la forme de ses états initiaux et finaux, de ses pré- et post-conditions et du résultat qui en est attendu. Aussi, pour les tâches pouvant soit en interrompre une autre, soit ayant des restrictions particulières lorsqu'interrompues, il faut définir les paramètres liés à l'interruptibilité. Ensuite, il faut indiquer comment parvenir au résultat attendu en énumérant les sous-buts ou les tâches nécessaires ainsi qu'en définissant le plan qui spécifie la manière par laquelle lesdits sous-buts ou tâches doivent être utilisés.

La description d'une tâche selon MAD peut donc commencer très tôt dans le processus d'analyse de tâche et se poursuivre aussi longtemps qu'il est nécessaire pour obtenir le niveau de détail correct selon les objectifs de l'analyse à effectuer.

3.4.2 Fenêtre principale

La fenêtre principale pour la visualisation de l'arbre de décomposition hiérarchique de la tâche étudiée est évidemment la même que celle des AHT et présente les mêmes possibilités (voir la figure 3-20). Idem quant à ce qui concerne la présence d'une racine dès la création puisque, ici aussi, @Esperanto requiert de l'utilisateur les renseignements à propos de celle-ci dès la commande de création émise.

3.4.3 Fenêtre favorisant l'entrée de masse (ou éditeur de tâche)

Une fenêtre favorisant l'entrée d'une grande quantité d'information existe pour MAD pour les mêmes raisons qu'elle a été jugée nécessaire pour l'AHT. MAD étant très semblable à l'AHT (surtout avec @Esperanto), il est normal que la version MAD de cette fenêtre soit très proche de sa contrepartie AHT.

Sans surprise, la fenêtre d'@Esperanto prévue pour l'entrée d'un grand volume de données dans MAD tire profit de la représentation textuelle (tabulaire) parce qu'elle s'y prête mieux que sa représentation graphique. Les seules différences sont liées au nombre minimal élevé d'attributs qui doivent être documentés pour chaque tâche et l'importance donnée à la décomposition de chacune. Le contenu suggéré pour cette fenêtre est comme montré à la figure 3-21.

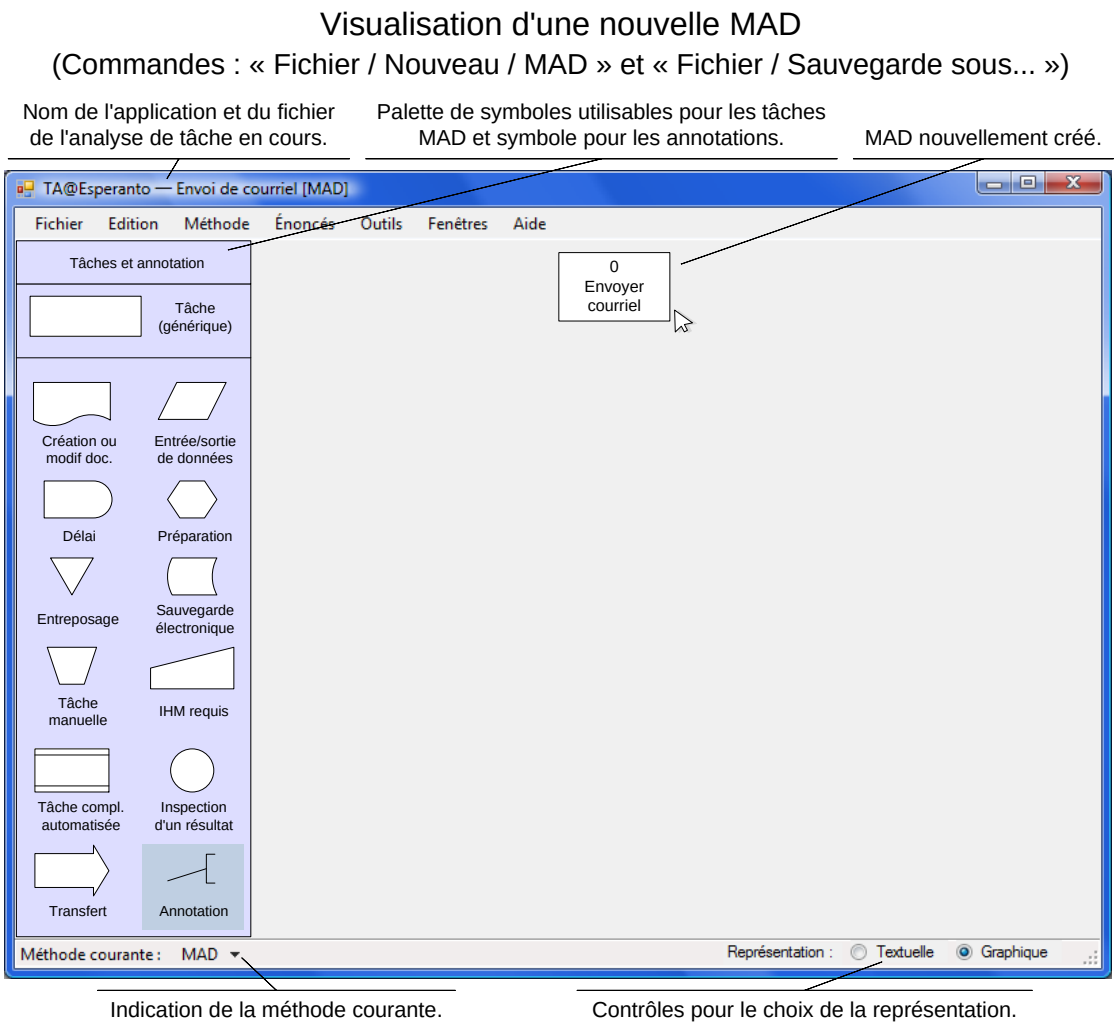


Figure 3-20 Fenêtre pour la visualisation et l'édition d'une tâche selon MAD.

En regardant la figure 3-21, on constate ce qui suit.

- Le contexte de la tâche courante, soit celle pour laquelle on peut définir et voir les enfants, est toujours disponible sous la forme de deux énumérations : celle de ses parents à partir de la racine et celle de ses sœurs.
    - Parmi toutes ces tâches, seule la tâche courante est mise en évidence.
    - Au besoin, ces bandes peuvent être agrandies pour contenir plus de lignes permettant ainsi de lister tous les parents et toutes les sœurs.
- Les attributs MAD de la tâche courante sont tous accessibles directement.
    - D'éventuels attributs spécifiques à une étude peuvent être ajoutés après les attributs MAD, soit après « Résultat ». Selon le nombre d'attributs à ajouter et la taille de la fenêtre disponible, la section des attributs pourrait nécessiter une barre de défilement verticale.

<numéro et nom des tâches aïeules>						
*<nom et numéro de la **tâche courante** ainsi que de ses soeurs>*						
But						
État initial						
Précondition						
État final						
Postcondition						
Résultat						
Type	générique ▼	Priorité	9	Interruptibilité	Interruptible; continuer	▼
<plan associée à la tâche courante>						...

#	Nom	Fac./Cond.	But
1	*<nom du premier enfant>*		
2	*<nom du deuxième enfant>*		
	...		

Figure 3-21 Contenu suggéré de la fenêtre d'acquisition d'une masse de données.

- Le plan associé à la tâche courante est visible (comme d'habitude) entre celle-ci et ses enfants. Comme pour une AHT, ce plan peut être tapé directement ou être composé à l'aide de l'éditeur de plan.
    - Si le plan venait à prendre trop de place, ce champ peut être agrandi pour contenir jusqu'à trois lignes. Autrement, il faudra faire appel à une infobulle en survolant le plan.
    - Pour les sous-tâches, seuls le nom, l'attribut « Facultatif / Conditionnel » et le but sont visibles, mais cela devrait suffire à percevoir le rôle de chacune dans la réalisation de la tâche courante.
    - Cette section peut nécessiter des barres de défilement.
- L'utilisateur n'a pas besoin de gérer la numérotation des tâches, puisqu'elle est prise en charge par @Esperanto.
    - Dans le cas où l'utilisateur modifierait l'ordre des enfants (par insertion, destruction ou déplacement d'un enfant vers une autre position), @Esperanto modifiera le plan en conséquence.
- Tous les numéros et noms de tâche peuvent être considérés comme une sorte d'hyperliens, puisqu'un double clic sur l'un d'eux charge la tâche visée et en fait la tâche courante.

À titre d'exemple, on peut voir ci-dessous la définition de la tâche principale de l'exemple récurrent du chapitre précédent. On y constate que l'appellation « Fenêtre favorisant l'entrée de masse » est moins méritée pour MAD qu'elle ne l'était pour l'AHT, puisque tous les attributs ne sont visibles que pour la tâche courante. Bien que l'appellation « Éditeur de tâches » puisse sembler plus appropriée, son nom a été conservé pour ses capacités de navigation d'une tâche à l'autre qui facilitent l'entrée de plusieurs tâches par rapport à l'ajout de tâches à partir de la fenêtre principale (disponible dans MAD comme pour l'AHT, ainsi qu'on le verra sous peu).

0 -- Envoyer courriel

But	Envoyer un courriel du compte choisi avec possibilité d'ajouter une pièce jointe.
État initial	Logiciel de gestion des courriels Fichier à joindre.
Précondition	Logiciel de gestion des courriels ouverts; Fichier à joindre disponible au système de gestion des fichiers de l'ordinateur.
État final	Logiciel de gestion des courriels; Courriel; Fichier à joindre;
Postcondition	Courriel envoyé; Fichier à joindre envoyé avec le courriel si la tâche 5 a été effectuée. Logiciel de gestion des courriels dans le même état qu'avant la tâche. Fichier à joindre non modifié sur le disque dur;
Résultat	Le courriel et, si désiré, le fichier à joindre sont envoyés.

Type	générique	Priorité	9	Interruptibilité	Interruptible; continuer

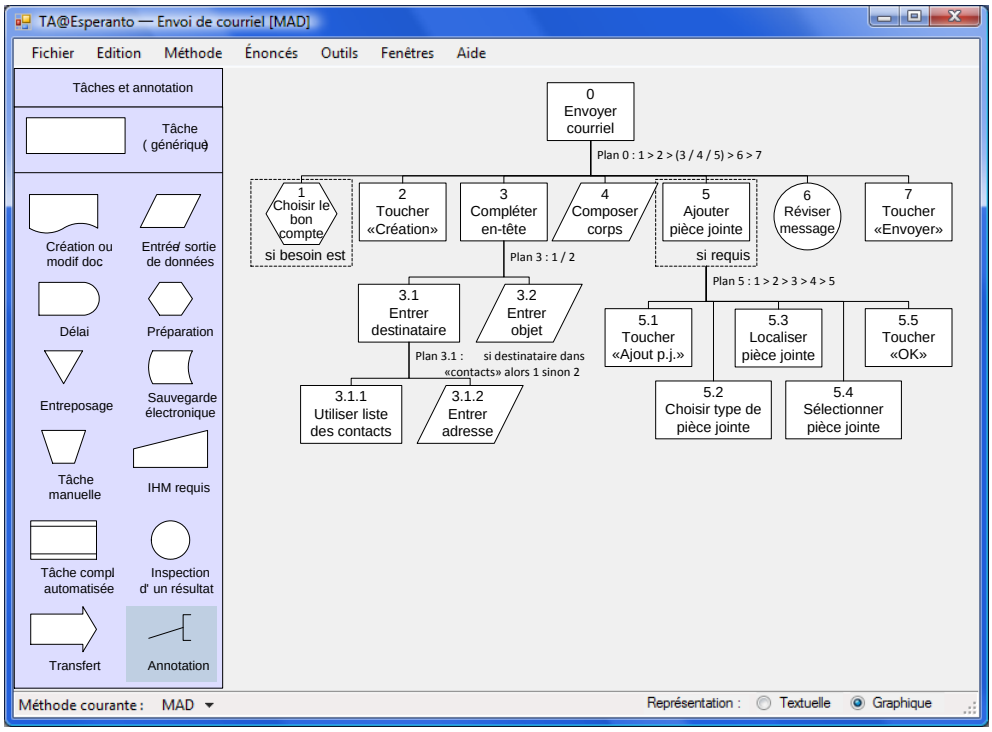
1 > 2 > (3 / 4 / 5) > 6 > 7

#	Nom	Fac./Cond.	But
1	Choisir le bon compte	si besoin est	Choisir le compte pour l'émetteur du mes
2	Toucher «Création»		Ouvrir la fenêtre de création d'un messag
3	Compléter en-tête		Définir les paramètres d'envoi.
4	Composer corps		Composer le corps du message.
5	Ajouter pièce jointe	si requis	Joindre un document au message.

Figure 3-22 Informations entrées à propos de la tâche principale « Envoyer courriel ».

Après l'entrée des informations concernant l'ensemble des tâches faisant partie de l'analyse, l'utilisateur ferme la fenêtre d'entrée de masse et, de retour à la fenêtre principale, il peut maintenant voir l'arbre de décomposition de la tâche principale montré à la figure 3-23. Sans surprise, on constate que cet arbre correspond exactement à celui de l'AHT montrée à la figure 3-5. La seule différence visible dans la fenêtre principale entre AHT et MAD est que, peu importe les attributs spécifiques à une analyse, la représentation textuelle de l'arbre MAD montre

toujours les mêmes attributs, soient le numéro, le nom incluant la notation pour l'attribut « Facultatif / Conditionnel », le type et le but.



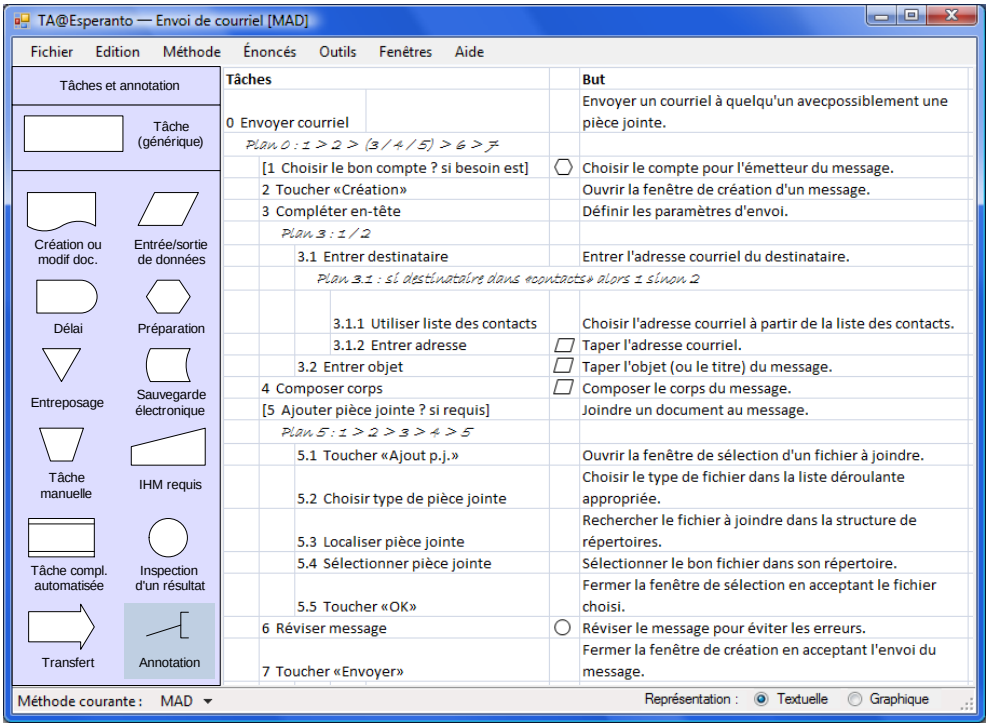


Figure 3-23 Représentations graphique et textuelle de l'arbre MAD comme entrée jusqu'ici.

La différence par rapport à l'AHT est qu'ici, la réponse d'@Esperanto au survol d'une tâche en représentation graphique est de montrer (dans une infobulle) les informations de cette tâche sous la forme traditionnelle de MAD, telle que vue au chapitre précédent. Par exemple, en survolant la tâche principale, la fenêtre principale prendrait l'aspect illustré à la figure 3-24.

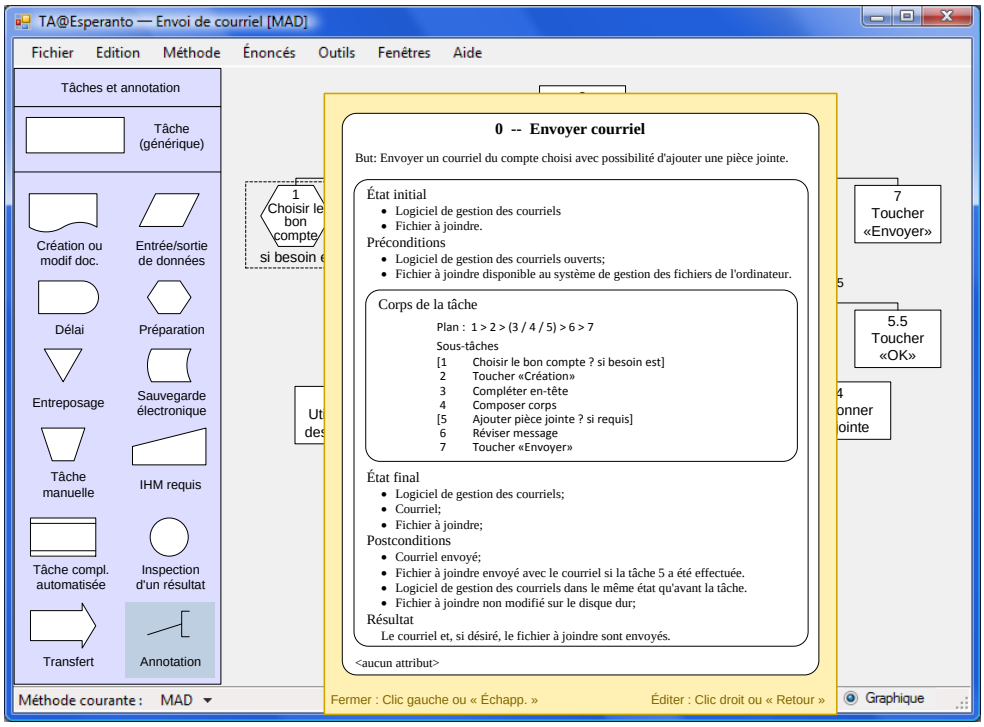


Figure 3-24 Infobulle montrant les détails de la tâche principale en réponse à son survol.

3.4.4 Modification d'une description de tâche MAD de la fenêtre principale

Les modifications que l'on peut apporter à une description MAD directement de la fenêtre principale sont les mêmes que celles que l'on peut apporter à une AHT. La différence est que toutes ces modifications (excepté le retrait d'une tâche) devront être terminées et confirmées dans l'éditeur de tâche (aussi connu comme étant la fenêtre favorisant l'entrée de masse).

Comme on peut le voir au coin inférieur droit de l'infobulle de la figure 3-24, l'éditeur en question est aussi accessible d'un simple clic du bouton droit de la souris ou d'une pression de la touche « Retour ».

La seule différence concerne la modification d'un plan, qui doit se faire à partir de l'éditeur de tâche.

3.4.5 Éditeur de plan structuré

L'éditeur de plan qui est appelé lorsque l'utilisateur presse le bouton « … » de l'éditeur de tâche est le même que celui utilisé pour les AHT. Il devrait n'être utilisé qu'après que tous les enfants de la tâche courante aient été énumérés pour assurer que tous puissent être transférés dans la fenêtre des sous-tâches de l'éditeur de plan, mais il est inutile d'attendre d'avoir aussi défini tous leurs attributs MAD.

3.5 Analyse temporelle

3.5.1 Processus de construction d'une analyse temporelle

Avec @Esperanto, l'élaboration d'une analyse temporelle peut se faire de deux manières différentes. Soit l'utilisateur a déjà analysé la tâche selon une méthode plus globale (AHT, OTHI ou MAD), soit il ne l'a pas fait.

Dans le premier cas, avec l'assistance d'@Esperanto, l'utilisateur peut extraire divers chemins d'exécution qui vont devenir ses scénarios pour lesquels il pourra ensuite peaufiner l'ordre d'exécution des tâches pour arriver à ce qu'il estime être le meilleur agencement pour chacun. Advenant que son analyse, possiblement plus détaillée, l'amène à ajouter ou retirer des tâches, @Esperanto doit signaler à l'utilisateur qu'il doit vraisemblablement mettre à jour l'analyse initiale.

Dans le second cas, l'utilisateur énonce et agence l'ensemble des tâches qui vont réaliser au mieux la tâche à analyser.

Dans les deux cas, si pertinent à son analyse, l'utilisateur détermine les attributs que devront posséder chaque tâche pour ensuite assigner comme nécessaire un temps pour l'exécution et une valeur pour chaque attribut.

3.5.2 Extraction d'un scénario d'une étude globale (AHT, OTHI ou MAD)

La manière la plus simple d'extraire un scénario d'une étude globale d'une tâche est de se laisser guider par @Esperanto à travers l'OTHI (construit ou obtenu par conversion d'une AHT ou d'une description MAD, si nécessaire). Pour une telle extraction, @Esperanto montre l'OTHI à l'utilisateur et se déplace d'un point de décision à l'autre à partir du symbole de début. Pour

chacun, il demande à l'utilisateur quelle situation il veut retenir pour le scénario en construction, note la décision et se déplace au prochain point de décision sur le chemin choisi. À la fin, @Esperanto a produit un scénario identifié par toutes les décisions prises et contenant tous les énoncés rencontrés sur le chemin d'exécution correspondant. L'utilisateur peut alors nommer le scénario pour y référer plus aisément.

Les points de décision possibles sont :

- Les énoncés conditionnels (sans égard à la syntaxe utilisée) pour lesquels l'utilisateur doit indiquer la décision que l'opérateur prendrait dans ce scénario.
- Les énoncés itératifs pour lesquels l'utilisateur doit indiquer le nombre d'itérations à faire.
- Les agencements de tâches laissant une certaine liberté à l'opérateur[54], pour lesquels l'utilisateur doit indiquer la séquence d'énoncés qui serait choisie dans ce scénario.

Par exemple, pour obtenir le scénario montré en exemple au chapitre de présentation de l'analyse temporelle, l'utilisateur serait guidé par @Esperanto à travers l'OTHI de la figure 2-4 et aurait effectué les décisions suivantes :

- Le compte courant est-il le bon? NON!
- Dans quel ordre aborder les tâches de la première série de tâches pouvant être exécutée de manière non séquentielle? Compléter en-tête > Composer corps > Ajouter pièce jointe!
- Dans quel ordre aborder les tâches de la deuxième série de tâches pouvant être exécutée de manière non séquentielle? Entrer destinataire > Entrer objet!
- Destinataire dans liste des contacts? NON!
- Pièce jointe requise? OUI!

Le scénario résultant contient bien la séquence de tâches utilisée dans les exemples vus lors de la présentation de l'analyse temporelle, soit :

Tâche : Envoyer courriel [scénario : cas le plus long]

Choisir le bon compte
Toucher «Création»
Entrer adresse
Entrer objet
Composer corps

---

[54] Ce sont les agencements d'exécution non séquentielle, d'exécution d'un nombre quelconque de tâches et d'exécution d'une tâche parmi un choix de tâches.

Toucher «Ajout p.j.»
Choisir type de pièce jointe
Localiser pièce jointe
Sélectionner pièce jointe
Toucher «OK»
Réviser message
Toucher «Envoyer»

Lorsque l'OTHI a été obtenu par conversion d'une AHT ou d'une description MAD, une recherche de toutes ces tâches dans l'arbre hiérarchique (AHT ou MAD) permet d'identifier les nœuds dont dépendent les tâches contenues dans le scénario. Ainsi, on peut retrouver la hiérarchie correspondant à la séquence retenue, ce qui dans notre exemple est comme suit :

Tâche : Envoyer courriel [scénario : cas le plus long]

Choisir le bon compte
Toucher «Création»
Compléter en-tête
Entrer destinataire
Entrer adresse
Entrer objet
Composer corps
Ajouter pièce jointe
Toucher «Ajout p.j.»
Choisir type de pièce jointe
Localiser pièce jointe
Sélectionner pièce jointe
Toucher «OK»
Réviser message
Toucher «Envoyer»

Pour faciliter la création d'autres scénarios, @Esperanto permet de commencer à partir des réponses données lors de la création d'un précédent scénario où le processus expliqué plus haut reprendra à partir du premier point de décision pour lequel l'utilisateur changera la réponse. Aussi, chaque fois que l'utilisateur fournit une réponse à un point de décision, @Esperanto la conserve pour la présenter dans une liste de réponses précédentes, ce qui sert à la fois d'aide-mémoire concernant les précédents scénarios et accélère l'entrée de réponses que l'utilisateur désire identiques dans l'ensemble des scénarios.

3.5.3 Création et modification d'une analyse temporelle

La fenêtre d'édition d'une analyse temporelle doit utiliser une représentation textuelle pour permettre d'entrer rapidement le nom et, lorsque nécessaire, le temps requis pour compléter chaque tâche. Elle permet aussi d'entrer la valeur de tous les attributs de tâche spécifiques à l'analyse en cours.

Bien qu'il soit plus facile d'agencer les tâches sur des fils d'exécution différents dans une représentation graphique, cette information est présentée et modifiable dans la représentation textuelle. Les portions du scénario qui nécessite l'exécution de tâches en parallèle ou en simultané sont identifiées par une lettre et leurs fils d'exécution par des chiffres. Les fils de la première portion sont donc nommés 'A1', 'A2', etc., ceux de la deuxième portion 'B1', 'B2', etc. et ainsi de suite. En représentation graphique, alors que déplacer une tâche *sur* une autre déplace la première avant ou après la deuxième[55], déplacer une tâche *au-dessus* ou *au-dessous* d'une autre les met en parallèle.

Parce que parmi les actions requises pour l'élaboration d'une analyse temporelle, certaines sont plus faciles en représentation textuelle alors que d'autres sont plus aisées en représentation graphique, l'éditeur d'analyse temporelle les montre toujours simultanément. Bien sûr, c'est l'utilisateur qui doit choisir la représentation graphique courante parmi celles que supporte @Esperanto[56]. Il doit aussi pouvoir modifier son choix à sa convenance. Selon la direction de la représentation graphique, ce pourrait donc être comme montré à la figure 3-25 ou à la figure 3-26.

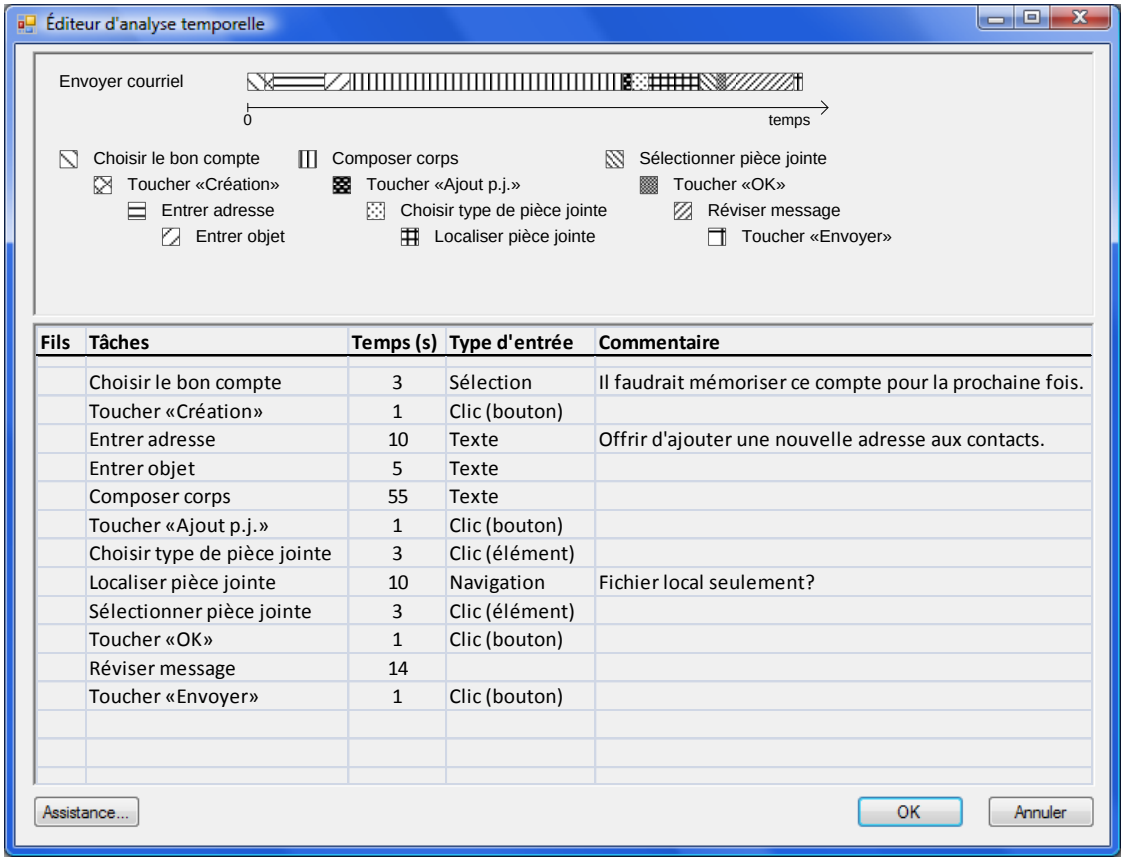
Pour les analyses requérant de tenir compte de la charge mentale de travail (CMT), celle-ci peut être associée à chacune des tâches comme un attribut strictement numérique ou, au besoin, comme une fonction du temps depuis lequel la tâche associée a commencé. Cette dernière option permet à l'utilisateur de tenir compte du fait que certaines tâches n'ont pas un impact constant sur

---

[55] Le déplacement se fait dans le sens du mouvement, donc, si la tâche déplacée est avancée dans le temps elle sera insérée après celle sur laquelle on l'a mise, autrement, elle sera insérée avant.

[56] À ce sujet, voir la section dédiée à l'analyse temporelle du chapitre de présentation des méthodes.

la CMT. Parce qu'@Esperanto connaît les deux versions de cet attribut, il peut le gérer adéquatement et l'afficher dans les représentations graphiques pouvant inclure la CMT.



Fils	Tâches	Temps (s)	Type d'entrée	Commentaire
	Choisir le bon compte	3	Sélection	Il faudrait mémoriser ce compte pour la prochaine fois.
	Toucher «Création»	1	Clic (bouton)	
	Entrer adresse	10	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.
	Entrer objet	5	Texte	
	Composer corps	55	Texte	
	Toucher «Ajout p.j.»	1	Clic (bouton)	
	Choisir type de pièce jointe	3	Clic (élément)	
	Localiser pièce jointe	10	Navigation	Fichier local seulement?
	Sélectionner pièce jointe	3	Clic (élément)	
	Toucher «OK»	1	Clic (bouton)	
	Réviser message	14		
	Toucher «Envoyer»	1	Clic (bouton)	

Figure 3-25 Éditeur d'analyse temporelle avec une représentation graphique horizontale.

3.5.4 Visualisation d'une analyse temporelle

La fenêtre principale pour cette méthode ne présente qu'une représentation graphique de l'analyse courante. Son avantage est de pouvoir répartir une représentation très longue (si horizontale) ou très haute (si verticale) sur plusieurs bandes qui en présentent des sections consécutives.

De cette fenêtre, l'utilisateur peut modifier l'agencement des tâches, ajouter des annotations ou passer à l'éditeur pour des modifications plus fines.

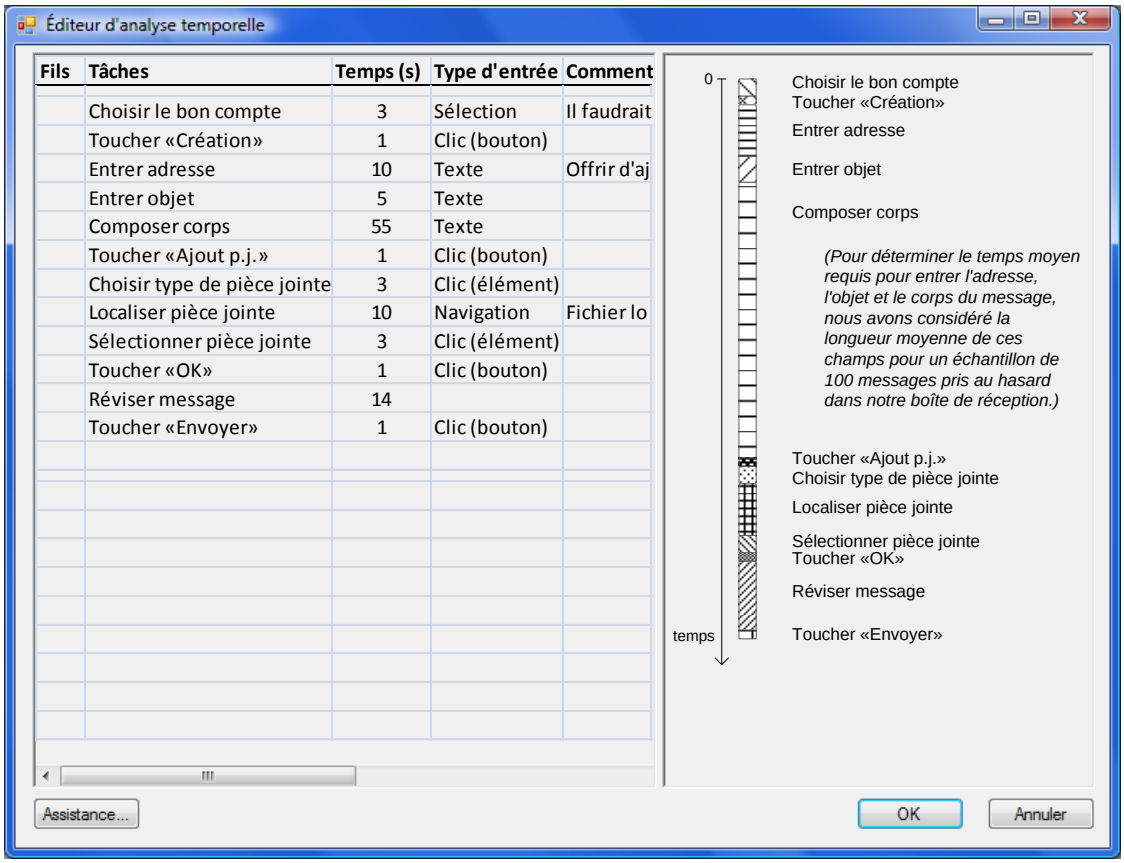


Figure 3-26 Éditeur d'analyse temporelle avec une représentation graphique verticale.

3.6 KLM-GOMS

3.6.1 Processus de construction d'une analyse KLM-GOMS

Comme pour l'analyse temporelle, une analyse KLM-GOMS peut se faire de deux manières différentes. Soit l'utilisateur a déjà analysé la tâche selon une méthode plus globale (AHT, OTHI ou MAD), soit il ne l'a pas fait. Puisque cet aspect a déjà été abordé, il ne sera pas discuté plus à fond ici.

La principale différence entre KLM-GOMS et l'analyse temporelle est la provenance des temps d'exécution pour chaque tâche. Comme on l'a vu au chapitre de présentation des méthodes d'analyse, KLM-GOMS utilise des codes pour référer aux quelques opérations qui vont permettre d'accomplir chacune des étapes de la tâche étudiée. Parce que ces opérations sont des actions aussi élémentaires, le plus souvent, l'exécution de chaque étape n'en nécessite pas qu'une.

Conséquemment, il devient important d'adjoindre quelques commentaires à ces opérations pour les rendre compréhensibles.

Parce qu'@Esperanto autorise la définition de plusieurs séquences d'opérations pour réaliser une tâche, l'utilisateur peut définir plusieurs scénarios pour une tâche donnée. Pour économiser un peu de temps, il doit lui être possible de dupliquer la séquence courante (sans égard à l'achèvement de l'analyse de celle-ci) pour poursuivre l'étude de diverses variantes. Pour chaque variante, @Esperanto demandera à l'utilisateur d'entrer un petit texte servant à l'identifier parmi les autres, créant ainsi une forme de règle de sélection.

On rappelle ici qu'une autre différence avec l'analyse temporelle est l'absence de représentation graphique. En fait, les utilisateurs s'attendent simplement à voir quelque chose ressemblant à l'exemple vu au chapitre de présentation.

3.6.2 Création et modification d'une analyse KLM-GOMS

L'IHM la plus commode pour gérer ce genre d'information est un tableau dont les colonnes contiennent les renseignements attendus et dont les lignes s'adaptent automatiquement en hauteur pour montrer toutes les données entrées. Comme pour l'AHT en représentation textuelle, une analyse KLM-GOMS devrait donc se faire dans une IHM ressemblant beaucoup à une feuille de calcul ressemblant à Excel©. En conséquence, les opérations permettant de créer ou modifier une analyse KLM-GOMS devraient être assez évidentes, puisqu'elles correspondent essentiellement aux opérations de modifications d'une feuille de calcul comme celles d'Excel©… insertion d'une ligne, retrait d'une ligne, déplacement d'une ligne ou modification de la valeur de l'une quelconque des cellules.

En plus des colonnes montrées pour KLM-GOMS dans le chapitre de présentation des méthodes d'analyse (Étape, Tâche, Opérations, Temps et Temps total), à l'instar de l'AHT, il est possible d'ajouter à chaque tâche autant d'attributs spécifiques à une étude (et donc de colonnes) que désiré.

Le plus important pour l'utilisateur est qu'@Esperanto fasse la sommation des temps correspondant aux opérateurs qu'il a entrés pour une tâche et celle correspondant à l'ensemble des tâches comme montré au chapitre de présentation.

En présence d'actions répétitives, @Esperanto devrait permettre à l'utilisateur de définir des codes qui devront référer à toute une séquence d'opérateurs. Par exemple, l'insertion d'une référence à une figure (IRF) dans un document Word© qui se fait par l'intermédiaire d'un dialogue spécialisé pourrait être représentée par les actions suivantes :

M
PC[Positionnement du curseur]
PC[Ouverture du menu]
PC[Sélection de l'item de menu]
MPC[Sélection de la bonne figure]
PC[Cliquer le bouton « Insérer »]
PC[Cliquer le bouton « Fermer »]

Plutôt que d'insérer tous ces opérateurs chaque fois qu'il faut insérer une référence à une figure, il serait intéressant de pouvoir n'entrer qu'un opérateur spécial, disons IRF, pour référer à tout cela. Ainsi, après avoir défini que IRF réfère à l'ensemble des opérateurs montrés ci-dessus, il suffirait d'entrer le code .IRF. (ici, les points de part et d'autre du code servent à empêcher toute confusion avec d'autres opérateurs).

3.7 CMN-GOMS

3.7.1 Une méthode toute désignée pour @Esperanto

Comme on l'a vu lors de la présentation de la méthode, CMN-GOMS tient à la fois de l'AHT par sa décomposition hiérarchique d'une tâche et de KLM-GOMS par l'évaluation que l'on fait des séquences d'opérations pouvant réaliser chaque sous-tâche que l'analyste ne juge pas utile de décomposer plus avant.

Par sa raison d'être, @Esperanto est on ne peut mieux équipé pour supporter CMN-GOMS sous la forme d'une analyse multiméthode. À la section suivante, portant justement sur l'analyse multiméthode, on verra comment un analyste peut réaliser une analyse CMN-GOMS.

La seule particularité qu'offre @Esperanto pour cette méthode est une représentation textuelle qui permet d'exprimer une analyse CMN-GOMS sous l'aspect vu au chapitre de présentation.

3.8 Analyse multiméthode

3.8.1 De la complexité de l'ensemble aux menus détails

Comme on l'a vu au chapitre précédent, exception faite de CMN-GOMS, toutes les méthodes que doit supporter @Esperanto peuvent être rangées soit dans la classe de méthodes capables de gérer la complexité d'une tâche et des diverses situations dans lesquelles on peut devoir la réaliser (AHT, OTHI et MAD), soit dans celles portant sur les menus détails d'exécution dans des circonstances bien déterminées (analyse temporelle et KLM-GOMS). Dans la suite de ce document, on qualifiera ces deux grandes classes de « globale » et « locale », respectivement.

Comme c'est souvent le cas avec des outils offrant une grande liberté d'action à ses utilisateurs, de grandes choses sont possibles, mais il y a des risques. Ici, par exemple, à trop souvent passer d'une méthode globale à une autre, on pourrait transformer une analyse de tâche en un total capharnaüm. Malgré cette éventualité finalement peu probable, @Esperanto considère que ses utilisateurs sont des professionnels responsables et ne leur applique donc aucune espèce de contraintes quant aux nombres de changements de méthodes d'analyse.

3.8.2 Transitions possibles entre méthodes d'analyse

Bien qu'@Esperanto permette de changer de méthode d'analyse à n'importe quel point d'une analyse, toute transition n'est pas admise. En effet, il n'admet pas le passage d'une méthode locale à une globale. Il en découle qu'il est impossible de passer d'une analyse d'un simple scénario faite en utilisant soit l'analyse temporelle soit KLM-GOMS vers une AHT, un OTHI ou une description MAD. Toute autre transition est permise.

D'une manière générale, pour éviter la production de diagrammes difficiles à interpréter, il serait préférable de garder les portions d'analyse séparées à moins que l'insertion[57] soit assez petite, disons de l'ordre d'un demi-écran ou d'une demi-page selon le médium. Cela ne devrait pas nuire à la compréhension des résultats de l'analyse puisqu'il est plus que vraisemblable que les transitions de méthodes coïncideront avec des portions significatives de la tâche qui pourraient

---

[57] Dans ce contexte, une insertion est la portion de l'analyse suivant une transition de méthode d'analyse.

être assimilées à une sous-routine dans le découpage qui est fait de tout ce qui est nécessaire pour réaliser le but principal.

Les transitions possibles sont donc :

AHT à MAD ou vice versa : Après normalisation, la seule différence entre ces deux méthodes réside dans l'obligation d'entrer des informations valides (puisqu'elles seront vérifiées) pour les attributs propres à MAD. Une tâche analysée selon MAD aura donc forcément ces attributs MAD alors qu'une analysée selon AHT pourrait ne pas les avoir.

Cela dit, la transition, même fréquente, entre ces méthodes ne pose aucun problème du point de vue d'@Esperanto. Méthodologiquement, par contre, ce serait une pratique difficilement justifiable.

AHT (ou MAD) à OTHI : Puisqu'il est possible de convertir automatiquement une AHT (ou MAD) en OTHI, on se doute qu'encore ici @Esperanto n'a aucun problème avec cette transition. Si l'utilisateur trouve plus claire la notation associée à l'OTHI pour certaines portions de son analyse, il n'a aucune raison de s'en priver. Dans ce cas, la tâche ciblée pour être analysée en OTHI devra tout de même avoir ses attributs AHT (ou MAD) définis, n'aura pas d'enfants et exhibera une icône identifiant l'OTHI. L'analyse OTHI de cette tâche se présentera sous la forme d'une sous-routine.

La première fois qu'un utilisateur commande une telle transition, @Esperanto doit lui signaler la possibilité de visualiser un plan sous forme graphique en lui signalant que, dans certains cas, cela peut éviter de recourir explicitement à un OTHI.[58]

OTHI à AHT (ou MAD) : @Esperanto modifie toute tâche ciblée pour être analysée en AHT (ou MAD) pour qu'elle apparaisse dans l'OTHI sous la forme d'une sous-routine exhibant une icône identifiant la nouvelle méthode. Ce stratagème et l'obligation de définir les attributs OTHI pour la tâche ciblée permettent de préserver la cohérence de l'ordinogramme et de sa représentation textuelle.

Si jamais l'analyste désirait revenir à une analyse OTHI, @Esperanto pourra l'assister dans cette conversion. Un non-analyste pourrait simplement demander de changer la représentation.

---

[58] La boîte de message doit contenir une case à cocher permettant de ne plus jamais afficher ce message.

Analyse temporelle à KLM-GOMS ou vice versa : Du point de vue d'@Esperanto, ces deux méthodes sont similaires, puisqu'elles acceptent les mêmes informations et que la seule différence réside dans la manière de spécifier un temps de complétion pour chaque tâche. Dans un cas, on attend une valeur numérique (avec possiblement une imprécision) alors que dans l'autre c'est une suite d'opérateurs auxquels correspondent des temps bien connus. Il est donc toujours possible de passer d'une méthode à l'autre.

Méthode globale à méthode locale : Dans ce cas, on s'attend à ce que la tâche ciblée ait tous ces attributs correctement définis, mais ne soit pas autrement décomposée dans la méthode globale. Les informations provenant de l'analyse locale ne représenteront alors qu'un ajout d'information qui pourra apparaître à la suite des résultats de la méthode globale.

Il est possible de montrer ceux-ci en permanence s'ils sont très courts, ou alors dans une infobulle en réponse à un survol de l'icône représentant un chronomètre qui identifie toute tâche portant des informations relatives au temps.[59]

Pour l'impression d'une telle analyse, l'utilisateur peut choisir d'afficher les résultats provenant d'une analyse locale sur d'autres pages que celles où apparaîtront les résultats de l'analyse globale.

3.8.3 Changement de méthode d'analyse

Lorsque l'utilisateur décide qu'il veut changer de méthode d'analyse pour continuer à développer ses connaissances concernant une tâche quelconque, il doit d'abord la sélectionner. Ensuite, il doit choisir l'item de menu « Changer de méthode d'analyse » du menu « Fichier » ou du menu contextuel d'une tâche, ce qui entraîne l'ouverture de la cascade associée à cet item qui contient chacune des méthodes d'analyse qui peuvent être choisies à ce point. À ce moment, l'utilisateur peut choisir une méthode d'analyse globale ou locale. Parce que la réponse d'@Esperanto diffère beaucoup selon la classe de méthode choisie, les deux possibilités sont traitées séparément.

---

[59] Techniquement, il n'y a pas nécessairement d'information concernant un temps de complétion, puisqu'une analyse temporelle pourrait ne conserver qu'un ordre d'exécution… Qu'à cela ne tienne, on garde le chrono!

3.8.4 Changement pour une autre méthode d'analyse globale

Si l'utilisateur a choisi une nouvelle méthode globale, @Esperanto la met en évidence dans l'analyse en cours en lui accolant une icône idoine.[60] Par exemple, à la figure 3-27, on peut voir une analyse AHT dans laquelle l'utilisateur a choisi d'analyser une tâche par un OTHI. Comme mentionné plus tôt, lorsque l'insertion est relativement petite, il est possible de représenter ensemble les résultats des deux analyses, ce qui est démontré à la figure 3-28.

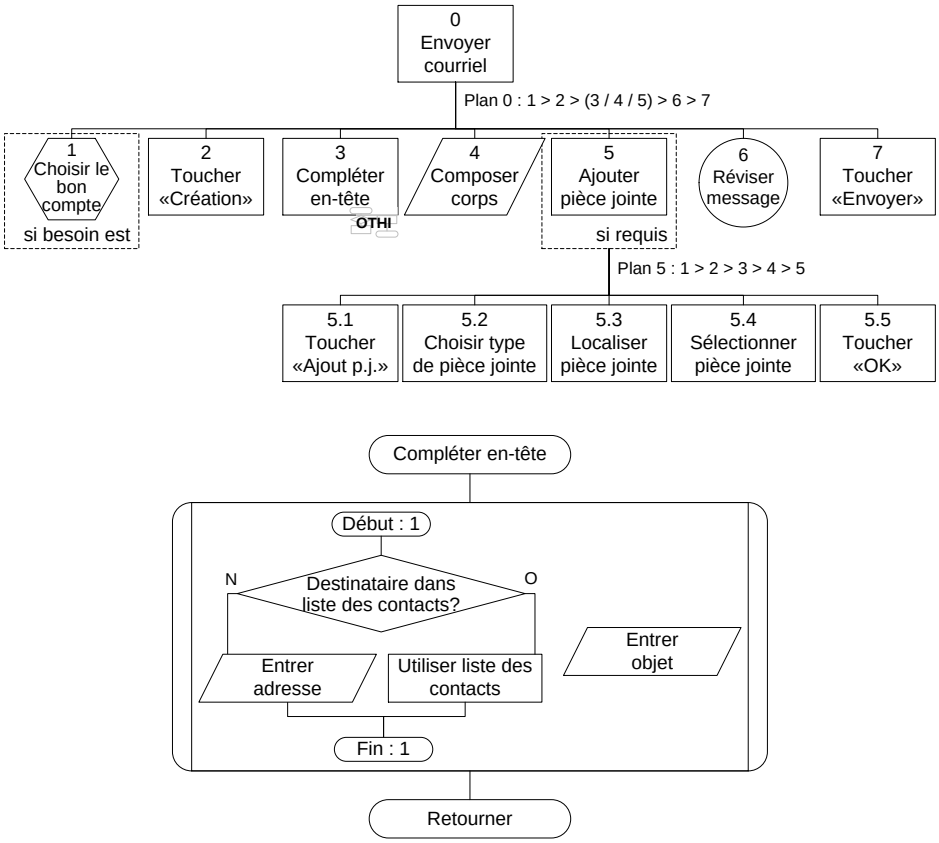


Figure 3-27 AHT dans laquelle une tâche est analysée par un OTHI.

Lorsqu'une tâche est sélectionnée pour être analysée selon une autre méthode que la méthode courante, en plus de l'identifier par une icône représentant cette autre méthode, @Esperanto l'ajoute à une liste de tâches à analyser plus tard. Cette façon de faire permet à l'utilisateur de continuer son travail avec la méthode courante en sachant bien qu'@Esperanto conservera toutes

[60] En bas à droite en représentation graphique et devant le nom de la tâche en représentation textuelle.

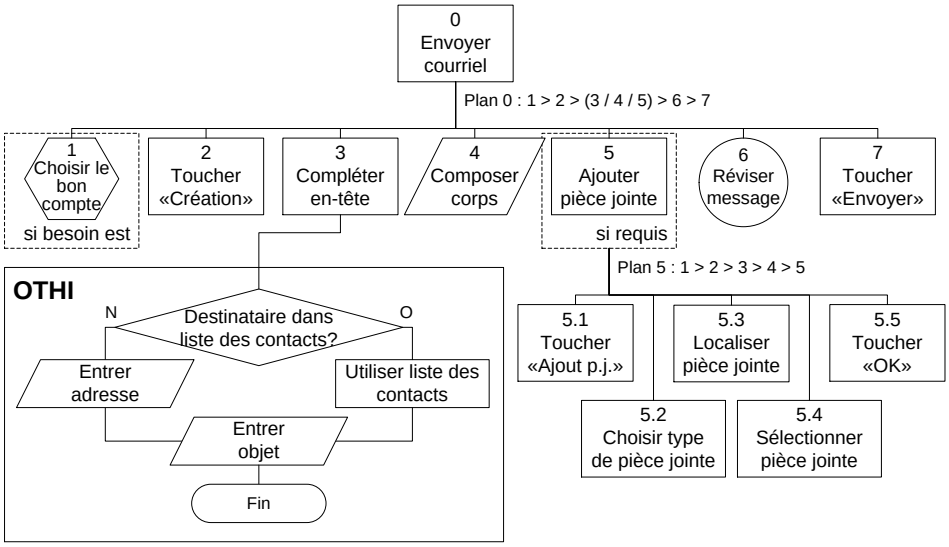


Figure 3-28 AHT avec une petite insertion en OTHI.

les tâches qu'il devra analyser par une autre méthode dans sa liste hiérarchique des tâches principale ou importantes.

Cette liste est en fait un élément essentiel pour l'aptitude d'@Esperanto à réaliser des analyses multiméthode, puisqu'elle représente la carte permettant d'identifier la place de chaque analyse depuis l'analyse de la tâche principale. Ainsi, dans l'exemple de la figure 3-27, cette liste contient deux éléments : d'abord la tâche principale (ici, 0 / « Envoyer courriel ») qui est toujours la racine de la hiérarchie, puis la tâche que l'on veut analyser par une autre méthode (ici, 3 / « Compléter en-tête »).

Notons que cette liste conserve aussi la méthode selon laquelle chacune de ces tâches est ou devra être analysée. Évidemment, l'utilisateur peut sans restriction décider de changer cette méthode pour toutes tâches dont l'analyse n'est pas commencée. Lorsque l'analyse est commencée, l'utilisateur peut encore changer d'une méthode globale à une autre en utilisant les capacités de conversion d'@Esperanto.

Lorsque l'utilisateur veut effectuer ou reprendre l'analyse de l'une des tâches mémorisées par @Esperanto, il n'a qu'à la double-cliquer dans la liste des tâches principale ou importantes, ce qui entraîne l'ouverture de l'éditeur approprié prêt à effectuer l'analyse voulue.

De ce qui vient d'être expliqué, certains auront compris qu'@Esperanto ne travaille vraiment qu'avec une méthode d'analyse à la fois, soit celle stipulée pour la tâche en cours d'analyse, peu

importe que ce soit la tâche principale ou l'une des tâches dites importantes qui sont mémorisées dans la liste hiérarchique. Rappelons qu'@Esperanto doit permettre d'utiliser plus d'une méthode d'analyse de tâches pour une même étude pour permettre d'analyser différentes portions selon diverses méthodes. La promesse est donc bien tenue.

3.8.5 Changement pour une méthode d'analyse locale d'une analyse globale

Si l'utilisateur a choisi une méthode locale, dans la cascade associée à l'item « Changer de méthode d'analyse » d'un menu, la situation diffère nettement de ce qu'on vient de voir. La principale raison étant que, comme mentionné plus haut, ce type d'analyse suppose que la tâche ainsi analysée n'aura pas d'enfants, ce qui veut dire que les résultats d'une analyse locale peuvent être entrés relativement vite une fois qu'ils sont connus. Il n'est donc pas nécessaire de remettre cette analyse à plus tard.

Conséquemment, lorsque l'utilisateur choisit une méthode locale pour la tâche sélectionnée, @Esperanto montre l'éditeur de la méthode désirée dans un dialogue modal permettant ainsi à l'utilisateur d'entrer les résultats de l'analyse locale avant de retourner à son analyse globale. De cette manière, l'utilisateur peut penser qu'@Esperanto permet l'utilisation de plus d'une méthode d'analyse à la fois, mais, en fait, il y a bien un changement net de méthode même si ce n'est que le temps de montrer un éditeur particulier dans un dialogue.

Une fois que des résultats d'analyse locale ont été entrés pour une tâche, @Esperanto la met en évidence dans l'analyse (globale) en cours en lui accolant une icône représentant un chronomètre (figurant le temps de complétion).[61]

3.8.6 Analyse locale mixte

Notons ici la possibilité de remplacer les deux éditeurs spécifiques par un seul éditeur générique. Celui-ci, profitant de la grande ressemblance entre les deux méthodes d'analyse locale, permettrait à l'utilisateur de décider, à chaque étape de son analyse locale, s'il va entrer des résultats correspondant à une analyse temporelle ou à un KLM-GOMS. Essentiellement, cet éditeur dual requerrait de l'utilisateur qu'il entre, pour chaque étape, son nom, tous ses attributs

---

[61] En bas à droite en représentation graphique et devant le nom de la tâche en représentation textuelle.

spécifiques à l'étude et soit un temps (avec ou sans imprécision), soit une suite d'opérateurs KLM-GOMS (avec ou sans commentaires).

Cet éditeur permettrait la création d'analyse locale mixte par sa capacité à utiliser des étapes dont le temps de complétion peut être déterminé soit par estimation (AT), soit par l'utilisation d'opérateurs (KLM-GOMS). Par exemple, à figure 3-29, on voit un cas où l'utilisateur a préféré effectuer une analyse locale mixte plutôt que de poursuivre le développement hiérarchique de la tâche #5 « Ajouter pièce jointe » de l'exemple récurrent du chapitre de normalisation. En fait, cette figure montre que ce qui se passe lorsqu'on survole le chronomètre est en conformité avec ce que fait @Esperanto lors du survol d'une tâche dans l'analyse globale : il montre l'ensemble des informations relatives à l'analyse locale dans une infobulle, d'où l'éditeur est facilement accessible. On notera aussi que les différentes méthodes sont mises en évidence par le fait que ce qui tient de l'analyse temporelle est en vert alors que ce qui tient de KLM-GOMS est en bleu.

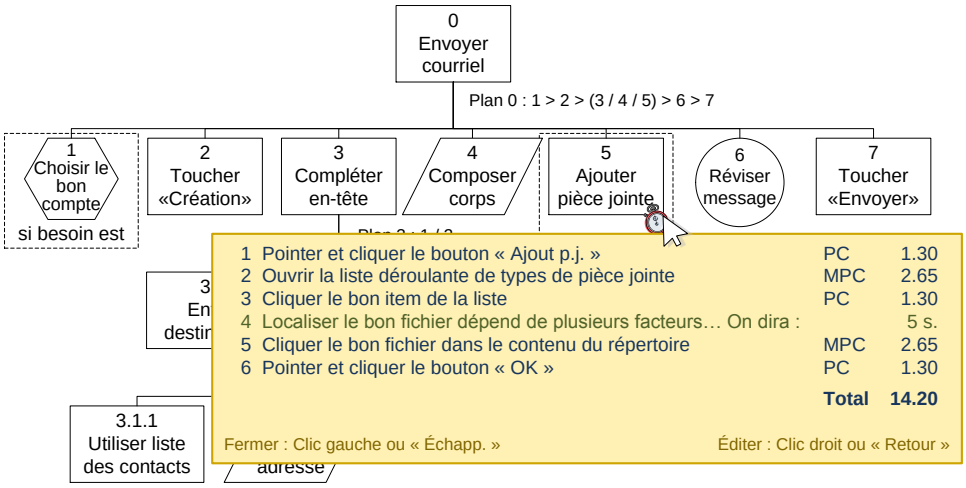


Figure 3-29 Visualisation d'informations relatives à une tâche analysée de manière mixte.

Comme mentionné plus haut, lorsque les insertions sont suffisamment petites, elles peuvent apparaître en permanence parmi les résultats de l'analyse globale. Par exemple, dans le cas de la figure 3-30 où l'utilisateur a choisi de développer hiérarchiquement la tâche #5 « Ajouter pièce jointe » de l'exemple précédent et d'ajouter cinq petites analyses locales pour un résultat équivalent à celui obtenu audit exemple.

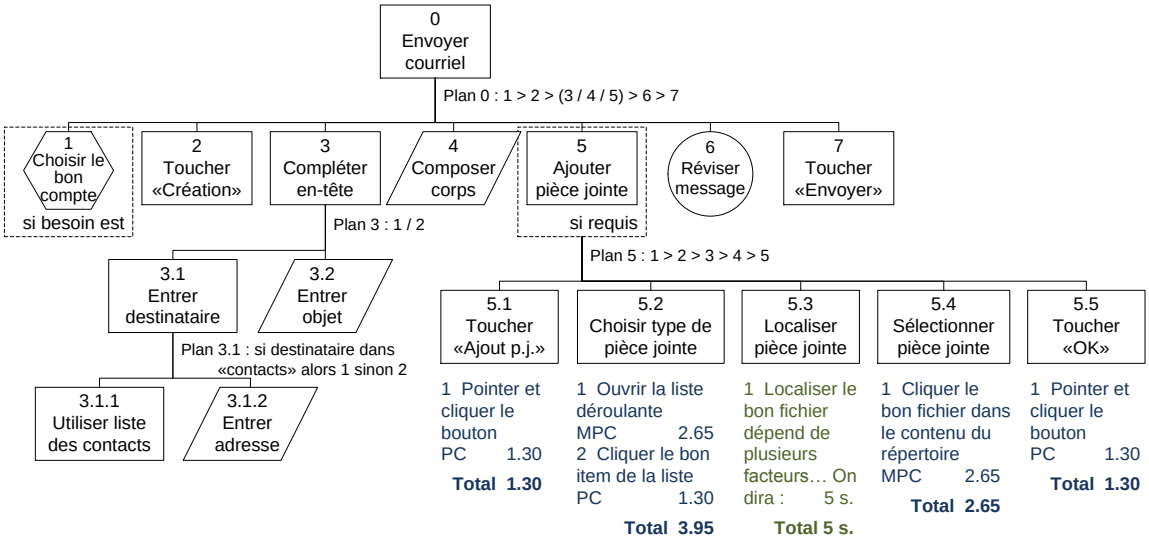


Figure 3-30 Insertions de petites analyses locales dans la représentation d'une AHT.

3.8.7 Un cas particulier : AHT + KLM-GOMS == CMN-GOMS

Après avoir vu les deux derniers exemples, on peut deviner comment @Esperanto supporte CMN-GOMS sans le faire explicitement. Ce n’est qu’une application d’une analyse multiméthode dans laquelle une tâche principale est décomposée hiérarchiquement selon la méthode AHT et dans laquelle chaque feuille est analysée localement selon KLM-GOMS.

Par défaut, @Esperanto offre la possibilité de faire les analyses locales de manière mixte, mais un paramètre des préférences (seulement accessible par le responsable de l’étude) permet de contraindre les utilisateurs à n’utiliser que KLM-GOMS (interdisant donc l’utilisation de l’analyse temporelle).

Pour accentuer la ressemblance avec une version plus classique de CMN-GOMS, on peut représenter l’analyse AHT sous forme textuelle et incorporer les portions KLM-GOMS tout de suite après les feuilles idoines de l’arbre d’analyse. À titre d’exemple, la figure 3-31 montre une telle représentation d’une analyse CMN-GOMS qui n’est pas encore terminée, puisque toutes les feuilles de l’arbre d’analyse n’ont pas encore d’analyse KLM-GOMS.

Tâches	Type d'entrée	Commentaire		
0 Envoyer courriel				
Plan 0 : 1 > 2 > (3 / 4 / 5) > 6 > 7				
[1 Choisir le bon compte ? si besoin est]	Sélection	Garder ce compte pour la prochaine fois.		
2 Toucher «Création»	Clic (bouton)			
3 Compléter en-tête				
Plan 3 : 1 / 2				
3.1 Entrer destinataire				
CHOISIR:				
Utiliser liste des contacts… si le contact est connu				
1 Pointer et cliquer le bouton « To > »			PC	1.30
2 Taper les trois premiers caractères du nom de famille			MHKKK	2.59
3 Cliquer sur le bon nom dans la liste des contacts			HPC	1.70
4 Pointer et cliquer le bouton « OK »			PC	1.30
			Total	**6.89**
Entrer adresse… dans le cas d'un nouveau contact				
1 Cliquer le champ d'édition de l'adresse courriel			PC	1.30
2 Taper l'adresse courriel (en moyenne 24 caractères)			MHK(24)	8.47
			Total	**9.77**
3.2 Entrer objet	Texte			
4 Composer corps				
[5 Ajouter pièce jointe ? si requis]		Utilisation occasionnelle.		
1 Pointer et cliquer le bouton « Ajout p.j. »			PC	1.30
2 Ouvrir la liste déroulante de types de pièce jointe			MPC	2.65
3 Cliquer le bon item de la liste			PC	1.30
4 Localiser le bon fichier dépend de plusieurs facteurs… On dira :				5 s.
5 Cliquer le bon fichier dans le contenu du répertoire			MPC	2.65
6 Pointer et cliquer le bouton « OK »			PC	1.30
			Total	**14.20**
6 Réviser message				
7 Toucher «Envoyer»	Clic (bouton)			

Figure 3-31 Analyse multiméthode simulant l'application de la méthode CMN-GOMS.

3.9 Conversion entre AHT et OTHI

Jusqu'ici, la conversion entre AHT et OTHI a été suggérée plus que démontrée. Nous allons donc voir un exemple de conversion avec un cas plus complexe que celui de l'envoi d'un courriel, mais auquel il est inutile de chercher une signification puisque son seul but est de mettre en scène plusieurs des constructions sémantiques ajoutées dans ce document. L'AHT qui sera la victime de toutes nos conversions est montrée à la figure 3-32.

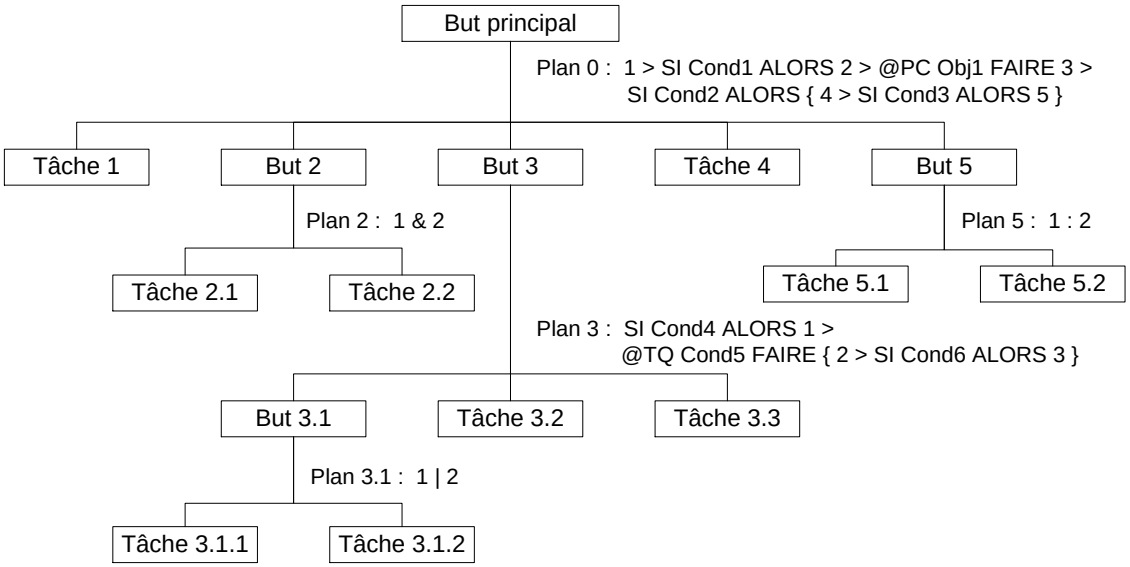


Figure 3-32 AHT servant de point de départ aux exemples de conversion AHT ↔ OTHI.

3.9.1 Conversion d'une AHT en OTHI

La conversion d'une AHT en OTHI ne cause aucune perte sémantique, seule l'information concernant la hiérarchie va disparaître puisque l'OTHI est une méthode procédurale.

Elle se fait en partant de la racine et en ne considérant qu'un niveau à la fois. À chaque niveau, il s'agit d'incorporer les plans de chacun des parents dans l'OTHI existant.

Au départ, l'OTHI étant vide, il suffit de transformer le plan de la racine en son équivalent OTHI… ce qui est automatique, puisqu'ils sont équivalents. Dans notre exemple, le résultat de la première étape est montré à la figure 3-33 où l'on notera la présence de symboles encore inconnus pour représenter les buts qui devront être remplacés dans les étapes subséquentes.

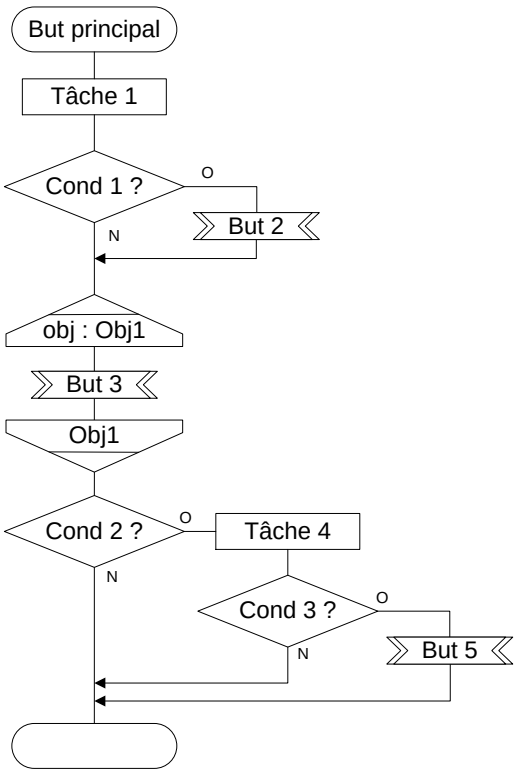


Figure 3-33 OTHI résultant du traitement de la racine de l'AHT de départ.

Lors de la deuxième étape, on remplace les buts 2, 3 et 5 par le plan correspondant dans l'AHT. Lors de la troisième et dernière étape, il ne reste plus que le plan du but 3.1 à incorporer pour obtenir l'OTHI complet montré à la figure 3-34.

3.9.2 Conversion d'un OTHI en AHT

Lors de la conversion d'un OTHI en AHT, deux options sont envisageables : soit on effectue la conversion sans se soucier de hiérarchie, soit on désigne des portions de l'OTHI comme servant chacune à réaliser un but ou une tâche plus complexe.

La première option est évidemment la plus facile, mais c'est aussi celle qui offre les moins intéressants résultats. Comme discuté plus haut, l'AHT obtenue est un arbre dégénéré où le plan de la racine est l'OTHI lui-même. On obtient donc l'AHT montrée à la figure 3-35, où le plan est tout simplement la représentation textuelle de l'OTHI à convertir.

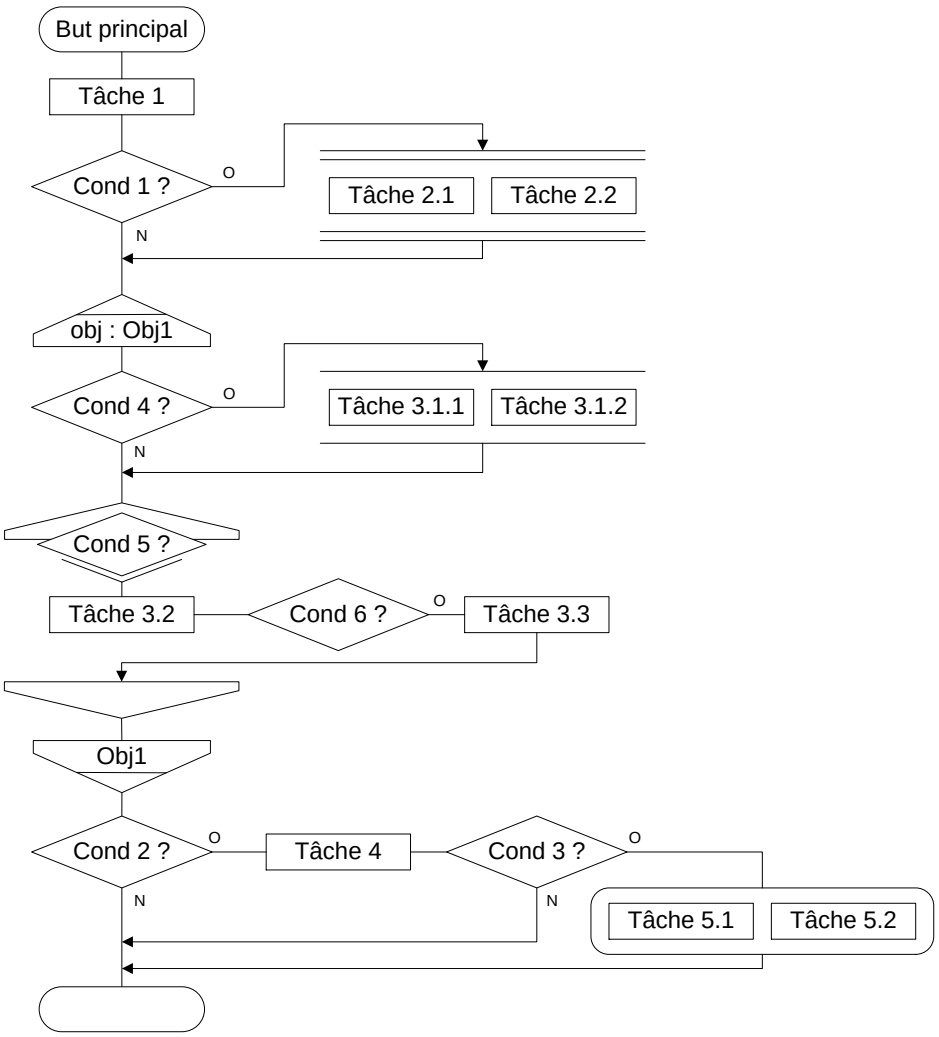


Figure 3-34 OTHI correspondant à l'AHT de départ (la hiérarchie est cependant perdue).

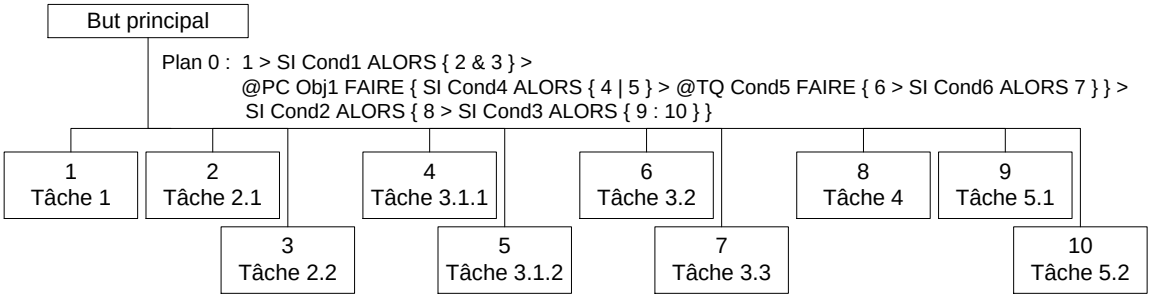


Figure 3-35 AHT résultant de la conversion d'un OTHI (sans égard à la hiérarchie).

La deuxième option requiert de l'utilisateur qu'il délimite autant de portions de l'OTHI que possible et que, pour chacune, il indique son but. C'est ce qui a été fait dans l'OTHI montré à la

figure 3-36, où quatre portions ont été délimitées et, pour chacune, un but a été indiqué. Pour être conforme avec la terminologie employée dans l'AHT de départ, les différents buts portent les mêmes noms que dans celui-ci.

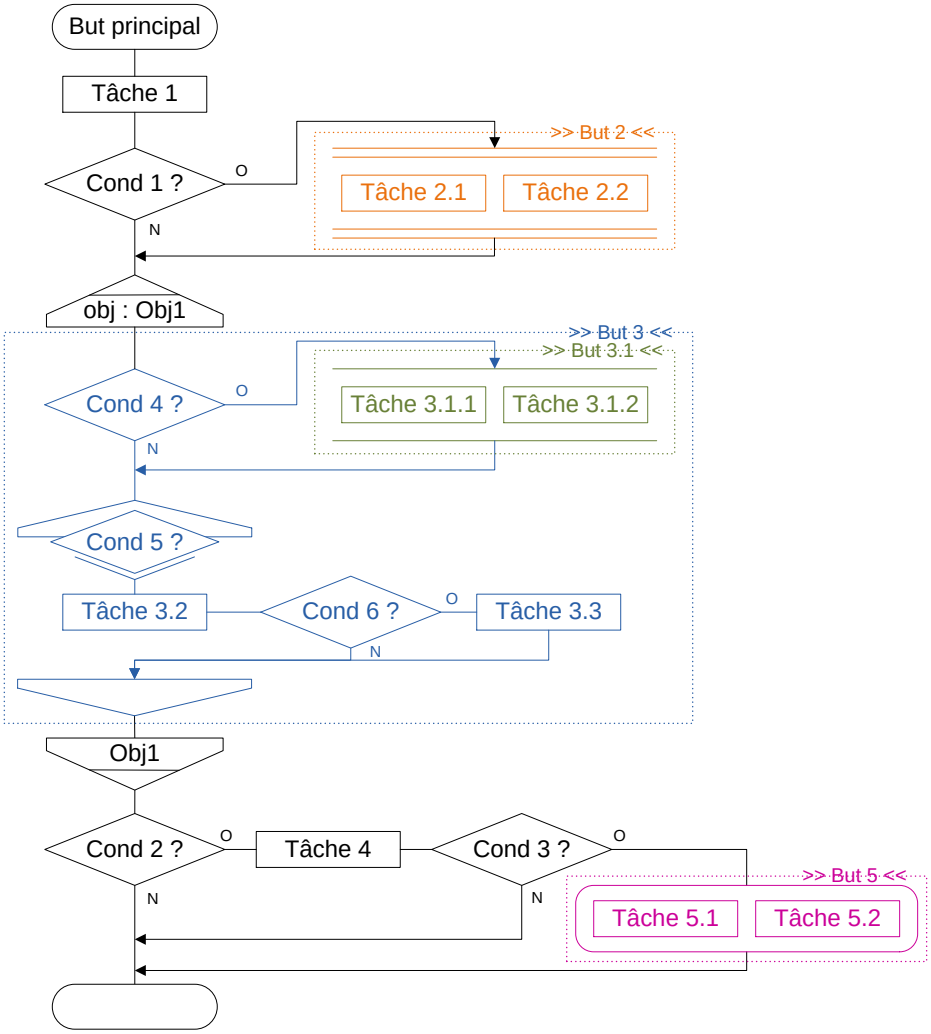


Figure 3-36 OTHI où quatre portions ont été identifiées et leurs buts indiqués.

Ce processus de conversion, comme celui permettant d'obtenir un OTHI à partir d'une AHT, se fait par étapes. Ici, il suffit de ne pas entrer trop vite dans les portions identifiées par l'utilisateur. Lors de la première étape, on n'entre dans aucune de ces portions, de sorte que la racine de l'AHT en construction aura un plan correspondant à l'OTHI dans lequel toutes ces portions sont considérées comme des tâches. Ici, cela correspond à la racine de l'AHT de départ, puisque le nom du but des portions identifiées a été choisi pour être identique à celui de son équivalent dans

l'AHT et que, comme d'habitude, les enfants d'un nœud sont les tâches et buts mentionnés dans le plan.

Dans les étapes subséquentes, on développe successivement chaque portion identifiée en assignant le plan et les enfants appropriés au nœud de l'AHT qui lui correspond. Dans cet exemple, on retrouve exactement l'AHT de départ.

3.10 Synthèse

Après avoir mentionné quelques fonctionnalités d'utilité générale, plusieurs fenêtres et éléments de l'interface utilisateur suggérée pour @Esperanto ont été définis pour chaque méthode d'analyse, à l'exception de CMN-GOMS qui a plutôt servi à illustrer le fonctionnement proposé des analyses multiméthode.

Une preuve de conversion entre AHT et OTHI est ensuite venue clore ce chapitre.

Tout ce qui devait être expliqué l'ayant été, il ne reste plus qu'à clore le sujet et ce mémoire.

CONCLUSION

Comme on l'a vu, @Esperanto supporte certaines méthodes d'analyse globales et d'autres qui sont locales. Parmi les méthodes globales, deux sont hiérarchiques alors qu'une autre est procédurale. @Esperanto offre donc bien à ses utilisateurs une variété de méthodes d'analyse de tâche.

Aussi, @Esperanto comble à merveille son but de permettre l'utilisation de plusieurs méthodes d'analyse même dans le cadre d'une seule étude. D'ailleurs, CMN-GOMS est ici considéré comme étant une combinaison de deux méthodes, soit d'une AHT avec autant d'analyse KLM-GOMS qu'il y a de feuilles dans la hiérarchie identifiée par l'AHT.

On a aussi constaté que les méthodes globales se ressemblent beaucoup plus qu'il n'y paraissait au premier abord. En fait, on les a même rapprochées encore plus en normalisant leur puissance expressive respective, sans pour autant les léser dans leurs points forts. En conséquence, toutes les méthodes globales supportées par @Esperanto sont capables d'exprimer les mêmes choses à propos des tâches (bien que d'une manière un peu différente). Pour refléter cette communauté d'esprit, les notations graphiques et textuelles de chaque méthode ont aussi fait l'objet d'une certaine normalisation, ce qui facilite l'apprentissage et la lecture des informations entrées selon diverses méthodes.

Différentes fenêtres et autres éléments d'interface ont été proposées pour permettre aux utilisateurs de créer et de modifier des analyses de tâches selon chacune des méthodes supportées par @Esperanto. Sans grande surprise, on constate que les IU résultantes pour les diverses méthodes d'une même classe présentent un certain air de famille lorsqu'on les compare entre elles. Ici aussi, un effort de normalisation est venu les façonner. Les différences constatées à ce point de vue résultent principalement du désir de conserver les particularités de chaque méthode.

Qui prendra le relais?

Le travail de conceptualisation effectué ici avait pour but de paver le chemin vers la création d'un logiciel permettant aux analystes d'effectuer leur travail avec la possibilité d'utiliser toujours une méthode adéquate aux circonstances. Comme défini ici, @Esperanto permet l'utilisation facile de plusieurs méthodes d'analyse, sans duplication d'information.

Il ne reste maintenant qu'à requérir les impressions et commentaires d'un maximum experts pour transformer ce mémoire en un document qui, on l'espère, réussira à convaincre un organisme qu'il est justifié et rentable de construire @Esperanto. Peut-être même cet organisme sera-t-il un département de Poly (par l'entremise d'un de ses professeurs) qui planifiera les travaux de différents étudiants pour qu'ils créent les divers composants du logiciel. Dans cette dernière optique, une équipe permanente très réduite pourrait assurer le maintien d'une direction précise durant tout le développement.

S'il en était besoin, c'est avec plaisir que je pourrais prêter assistance pour régler tout problème ou questionnement qui pourrait survenir en cours de développement.

BIBLIOGRAPHIE

Annett, J., & Duncan, K. D. (1967). Task analysis and training design. *Journal of Occupational Psychology, 41* 211-221.

Card, S. K., Moran, T. P., & Newell, A. (1983). The GOMS Model of Manuscript Editing. Dans *The Psychology of Human-Computer Interaction* (p. 139-189). Hillsdale, NJ: Lawrence Erlbaum Associates.

Crandall, B., Klein, G., & Hoffman, R. R. (2006). *Working Minds : A Practitioner's Guide to Cognitive Task Analysis*. Cambridge, MA: MIT Press.

Holleis, P., Otto, F., Hussmann, H., & Schmidt, A. (2007). Keystroke-level model for advanced mobile phone interaction. *Conference on Human Factors in Computing Systems - Proceedings* 1505-1514. doi: 10.1145/1240624.1240851

Hollnagel, E. (2012). Task Analysis: Why, What, and How. Dans G. Salvendy (Édit.), *Handbook of Human Factors and Ergonomics* (4e éd., p. 385-396): Wiley.

John, B. E. (2003). Information Processing and Skilled Behavior. Dans J. M. Carroll (Édit.), *HCI Models, Theories, and Frameworks* (p. 55-101): Morgan Kaufmann.

Limbourg, Q., Pribeanu, C., & Vanderdonckt, J. (2001). Towards Uniformed Task Models in a Model-Based Approach. Dans C. Johnson (Édit.), *Interactive Systems: Design, Specification, and Verification* (vol. 2220, p. 164-182): Springer Berlin Heidelberg.

Olson, J. R., & Olson, G. M. (1990). The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS. *Human-Computer Interaction, 5* 221-265.

Paterno, F., Mancini, C., & Meniconi, S. (1997). *ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models*. Communication présentée à Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction.

Sebillotte, S., et al. (1994). Note de recherche concernant le formalisme MAD. *Projet de Psychologie Ergonomique pour l'Informatique*, 31.

Stammers, R. B. (1995). Factors limiting the development of task analysis. *Ergonomics, 38*(3), 588-594.

Stanton, N. A. (2006). Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics, 37*(1 SPEC. ISS.), 55-79. doi: 10.1016/j.apergo.2005.06.003

van der Veer, G. C., Lenting, B. F., & Bergevoet, B. A. J. (1996). GTA: Groupware task analysis — Modeling complexity. *Acta Psychologica, 91*(3), 297-322. doi: http://dx.doi.org/10.1016/0001-6918(95)00065-8

Värnild, E. (1993a). *Étude sur le formalisme de description de tâches MAD.* (Mémoire, Université de Paris 8). Tiré de http://www.varnild.com/eric/rapportdess.htm

Värnild, E. (1993b). *Nouvelle définition de MAD.* (Mémoire, Université de Paris 8). Tiré de http://www.varnild.com/eric/maddefinition.htm

Wieringa, D., Moore, C., & Barnes, V. (1999). *Procedure Writing: Principles and Practices* (2e éd.). Canada: John Wiley & Sons

ANNEXE A – SYNTHÈSE DES FONCTIONNALITÉS ET DES FENÊTRES

Le contenu de cette annexe est une liste succincte des fonctionnalités et interfaces utilisateur qui ont expliqué en détail au chapitre 3 et qui sont donc suggérées pour @Esperanto.

Les items de cette liste sont regroupés en fonction de la méthode d'analyse à laquelle ils se rapportent et peuvent donc apparaître dans un ordre autre que celui de leur introduction au chapitre 3.

FONCTIONNALITÉS APPLICABLES À PLUSIEURS MÉTHODES

- Recherche de tâches
  - Par numéro, par nom ou par tout autre attribut (incluant ceux spécifiques à une étude).
  - Résultats affichés dans une fenêtre modale (avec plus ou moins de détails).
    - Survol d'une tâche dans les résultats faisant apparaître une infobulle avec la tâche et son voisinage.
- Navigation dans les résultats d'analyse
  - Zoom et vue hyperbolique (si la touche « Maj » est enfoncée).
- Visualisation des détails d'une tâche
  - Survol fait apparaître une infobulle montrant tous les détails de la tâche d'une manière cohérente avec la méthode d'analyse courante.
- Guide d'extraction d'un scénario à partir d'un OTHI.
  - Utile pour commencer une analyse selon une méthode d'analyse « locale ».
  - Si nécessaire, une AHT ou une description MAD est d'abord convertie en un OTHI.

AHT

- Fenêtre principale
  - Une palette contient les symboles utilisables.
  - L'insertion, la modification, le déplacement et le retrait de tâches sont possibles.

- En représentation graphique, l'ajout de tâches peut se faire par l'intermédiaire de mini-palettes.
    - Les informations requises à propos des nouvelles tâches doivent être entrées dans de mini-dialogues spécialisés.
- Un plan peut être créé ou modifié sur place ou en utilisant l'éditeur spécialisé.

➢ Fenêtre favorisant l'entrée de masse

➢ Éditeur de plan (encourageant la création de plans structurés).

➢ Éditeur de plan en représentation graphique : c'est, avec quelques restrictions, l'éditeur d'OTHI en représentation graphique.

OTHI

➢ Fenêtre principale

- Quatre palettes contiennent les symboles utilisables.
- L'insertion, la modification, le déplacement et le retrait de tâches sont possibles.
    - Les informations requises à propos des nouvelles tâches doivent être entrées dans de mini-dialogues spécialisés.
- L'insertion, la modification et le retrait d'énoncés pour le contrôle du flux d'exécution sont possibles.
    - En représentation graphique, les informations requises à propos des énoncés ajoutés ou modifiés doivent être entrées dans de mini-dialogues spécialisés.
    - En représentation graphique, l'ajout de tâches ou d'énoncés peut se faire par l'intermédiaire de mini-palettes.

➢ Éditeur d'OTHI en représentation textuelle inspiré de l'éditeur de plan structuré de l'AHT.

➢ Outil de conversion d'un OTHI en procédure destinée à l'aide à la performance humaine.

- Il permet de trouver les ambiguïtés et assiste l'utilisateur dans leur réduction ou leur élimination.

MAD

- Fonctionnalités et fenêtres presque comme l'AHT.
  - Différences un peu plus importantes pour la fenêtre d'entrée de masse qui doit tenir compte des cinq attributs emblématiques de MAD.

Analyse temporelle

- Éditeur d'analyse temporelle.
- Fenêtre de visualisation d'une analyse temporelle.

KLM-GOMS

- Visualisation et édition d'un KLM-GOMS (sous la forme d'une feuille de calcul comme celles d'EXCEL).

Analyse multiméthode

- Spécification des transitions possibles entre méthodes d'analyse.
- Représentations des transitions entre méthodes.