Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Logiciel de télécommunication, conception et développement d'une interface homme-machine centrés sur l'utilisateur

Colmant, Patrick

Award date:
1992

Awarding institution:
Universite de Namur

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 16. Apr. 2024

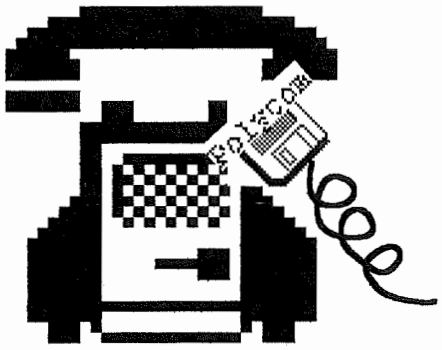
FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX

NAMUR

INSTITUT D'INFORMATIQUE

Logiciel de télécommunication, conception et développement d'une interface homme-machine centrés sur l'utilisateur

par Patrick COLMANT



Promoteur :

Professeur F. Bodart

Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en informatique

Année académique 1991-1992

à Matthieu,
mon p'tit fiou

RESUME

En l'absence de démarche systématique pour développer une interface homme-machine, nous en approchons une, centrée sur l'utilisateur. Nous travaillons par expérimentation, avec pour exemple, un logiciel de télécommunication. Les éléments de la démarche adoptée sont basés sur la modélisation de l'interface homme-machine comme un langage. Nous décomposons ce langage de l'interface en différents niveaux d'abstraction, à savoir les quatre niveaux pragmatique, sémantique, syntaxique et lexical.

L'utilisateur et l'ordinateur vont devoir communiquer ensemble pour accomplir certaines tâches. Or, en matière de communication, ils sont sur des planètes différentes. Alors que les personnes s'expriment sur base de leurs raisonnements, l'ordinateur présente des données sur base de l'exécution de programmes. L'interface devra d'une part amener la personne à s'exprimer le plus naturellement possible en des termes reconnus par la machine et, d'autre part, elle devra amener la machine à présenter des données en des termes reconnus le plus naturellement possible par l'utilisateur.

ABSTRACT

In the absence of a systematic approach to develop a user interface, we draw near to one focussed on the user. We work by experimentation, with a telecommunication software as an example. Our approach is characterized by the representation of the user interface as a language. We split this interface's language into several abstract's levels, namely the levels pragmatics, semantics, syntactics and lexical.

The user and the computer shall have to communicate to each other to accomplish several tasks. About communications, they live in opposite planets. When the human use his faculty of reasoning to express himself, the computer perform programs to show data. The interface will bring the human to express himself the most naturally it can be by means of terms recognised by the computer. The interface will also bring the computer to show datas in the way of terms recognised the most naturally it can be by the human.

SOMMAIRE

	Résumé, abstract	ii
	Sommaire	iii
	Remerciements	iv
1.	Introduction	1
2.	Système de télécommunication par modem	4
3.	Les acteurs	6
4.	Méthodologie	15
5.	Niveau pragmatique	21
6.	Niveau sémantique	51
7.	Niveau syntaxique	58
8.	Niveau lexical	65
9.	Architecture et implémentation	80
10.	Evaluation par prototype	88
11.	Synthèse, apports et perspectives	90
	Utilisateurs interrogés	a
	Bibliographie	c
	Table des matières	f

REMERCIEMENTS

Je tiens à remercier tout particulièrement celles et ceux parmi vous qui m'ont aidé dans l'élaboration de ce mémoire. Il s'inscrit dans le cadre de l'obtention du titre de Licencié et Maître en Informatique à l'Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix à Namur.

Ce mémoire est le fruit des utilisateurs rencontrés tout au long de mon chemin. Il fait suite à un stage presté chez Devlonics N.V. à Kortrijk sous la direction d'Alain Raulo.

Malgré les défaites, j'ai toujours pu compter sur l'avis constructif de François Bodart, mon promoteur.

Bienvenue à vous qui participerez à ma défense de mémoire. Vous y obtiendrez un complément d'informations sur tout point vous intriguant.

1. INTRODUCTION

Le stage a porté sur le ***logiciel fictif "Polycom"*** qui est un logiciel de télécommunication qui permet à l'utilisateur d'accéder à des services prestés par des centres serveurs télématiques. Nous avions apporté des observations en termes de développement d'interface homme-machine dans le cadre d'une mise à jour d'"Eurocom". ***Bien qu'Eurocom soit un logiciel de télécommunication, ce présent mémoire ne sera pas consacré aux télécommunications mais bien à la conception et au développement d'une interface homme-machine, avec pour exemple un logiciel de télécommunication.*** Eurocom se sert des modems commercialisés par la firme Devlonics pour offrir les fonctionnalités visées par Polycom.

Nous avons remarqué pendant le stage, que l'interface homme-machine des produits de la firme se construisait au fil du temps, au gré des améliorations techniques réalisées. L'interface traduit en quelque sorte la performance du produit. S'il doit y avoir une remise en cause de l'interface, c'est au niveau de son esthétique. Le produit doit être plus simple à utiliser et l'interface doit toujours valoriser les performances du produit. Nous laissons cette approche à ceux qui la préfèrent.

Nous adoptons une autre démarche, à la rencontre de divers acteurs. Comme Bruce Tognazzini (Tog) [Tog92], "Human Interface Evangelist" chez Apple, nous parlerons volontiers d'"interface visible" plutôt que d'"interface graphique". Pour Tog, une interface visible est un environnement complet dans lequel les utilisateurs peuvent travailler confortablement, sachant toujours où ils en sont, où ils vont aller, et quels objets leur sont disponibles sur leur chemin.

L'interface graphique est un élément de l'interface visible. Pour porter le nom d'interface graphique, une interface n'a besoin que de faire usage d'objets qui ont une représentation graphique distincte les uns des autres. Le graphisme est un canal parmi d'autres qui sont mis à la disposition du programmeur pour communiquer avec l'utilisateur. Le multimédia est la prolifération de ces canaux. Quant à l'interface visible, elle est la partie perçue de l'iceberg que constitue l'application. L'utilisateur interagira à sa façon avec elle.

Une application peut présenter de magnifiques fenêtres, toutes sortes de boîtes à cocher et de boutons, tout en laissant la clé de son fonctionnement cachée ou invisible. Nous voilà retombés sur les propositions de Donald A. Norman [Nor83]. Alors que le concepteur se base sur son modèle de la tâche pour donner une image à son système, l'utilisateur se base sur "son" modèle de la tâche pour manipuler l'image qu'il perçoit du système. Chacun d'entre eux se base sur "son" propre modèle. Un modèle d'une tâche est basé sur deux approches complémentaires. La première est l'approche descriptive, répondant à la question suivante : quels sont les objets mis à notre disposition? La seconde est l'approche prescriptive, répondant

à une autre question : quelles sont les opérations que l'on peut effectuer sur ces objets? Le lien entre ces deux approches est que si l'utilisateur saisit de quels objets le système est composé, alors il peut agencer les opérations sur ces objets pour produire des marches à suivre et ainsi contrôler le système.

Afin de délimiter notre terrain d'action, nous nous réserverons à l'étude de quatre activités les plus fréquentes de l'utilisateur ciblé, qui répondent aux attentes des acteurs rencontrés. Elles sont :

- Régler son système de télécommunication par modem pour qu'il puisse communiquer avec un centre serveur. Pour des raisons historiques, l'utilisateur précisera toutes sortes de paramètres pour pouvoir établir une communication avec le centre serveur télématique choisi.
- Appeler le centre serveur. L'utilisateur entre en communication avec un centre serveur.
- Archiver des écrans reçus. L'utilisateur conserve ce qui a été affiché à l'écran de son terminal.
- Echanger des fichiers. L'utilisateur reçoit des fichiers en provenance du centre serveur, ou lui en envoie.

Au cœur de ce mémoire, en l'absence de démarche systématique pour développer une interface homme-machine, nous en approchons une, centrée sur l'utilisateur. Nous travaillons par expérimentation, avec pour exemple, un logiciel de télécommunication. Les éléments de la démarche adoptée sont basés sur la modélisation de l'interface homme-machine comme un langage. Nous décomposons ce langage de l'interface en différents niveaux d'abstraction, à savoir les quatre niveaux pragmatique, sémantique, syntaxique et lexical. Ces quatre niveaux peuvent être décrits par ce que l'on appelle la sémiotique, c'est-à-dire "the study of signs and symbols, especially the relations between written or spoken signs and their referents in the physical world or the world of ideas"[1]:

- ***Le niveau pragmatique*** porte sur une réalité humaine, c'est-à-dire "the study of those aspects of language that cannot be considered in isoloation from its use". L'utilisateur exprime ses buts dans des termes et sur des objets qui ont un sens pour lui. Pensons à l'exemple des objets qui sont disposés dans une classe d'école, quel objet permet d'écrire au tableau?

- ***Le niveau sémantique*** porte sur la représentativité pour la personne du comportement d'un objet, c'est-à-dire "the study of the relationships between signs and symbols and what they represent". Le comportement des objets ne doit pas être dénaturé par rapport aux expectatives de l'utilisateur. Dans l'exemple de la classe, les chaises se placent-elles sur les tables, à côté des tables ou les quatre pieds en l'air?

---

[1] Ces définitions en anglais sont extraites du *COLLINS ENGLISH DICTIONARY*, 2nd ed., 1986.

•***Le niveau syntaxique*** porte sur la formalisation du comportement des objets, c'est-à-dire "the branch of semiotics that deals with the formal properties of symbol systems". L'utilisateur se pose la question : "Comment cela se passe-t-il avec les moyens interactifs pour faire ceci, faire cela?" Dans l'exemple de la classe, comment faire avec les moyens du bord pour marquer une faute sur une copie d'élève, comment caricaturer le comportement de tel ou tel professeur?

•***Le niveau lexical*** porte sur la symbolisation d'une réalité humaine, c'est-à-dire "the meaning of a word in relation to the physical world or to abstract concepts, without reference to any sentence in which the word may occur". L'utilisateur se demande : "De quoi est-il question avec ce symbole, avec ce mot?" Prenons les exemples du mot "pion", ou de la couleur rouge. Ce niveau lexical permet d'établir une connexion avec le niveau pragmatique. Cette connexion est supportée par la question suivante, que l'utilisateur nous a posée aux commandes de prototypes : "Qu'entend-on par?".

Les prototypes de l'application ont été progressivement développés au moyen d'une architecture et d'une implémentation orientées objets.

2. SYSTEME DE TELECOMMUNICATION PAR MODEM

Polycom concerne un système de télécommunication par modem (fig. 2.1). Un tel système permet à l'utilisateur d'accéder à des services prestés par des centres serveurs télématiques. Ce système comporte : un ordinateur, un logiciel, un modem, une prise téléphonique pour accéder au réseau téléphonique de la R.T.T., de quoi raccorder l'ordinateur au modem, de quoi raccorder le modem à la prise téléphonique et, un centre serveur.

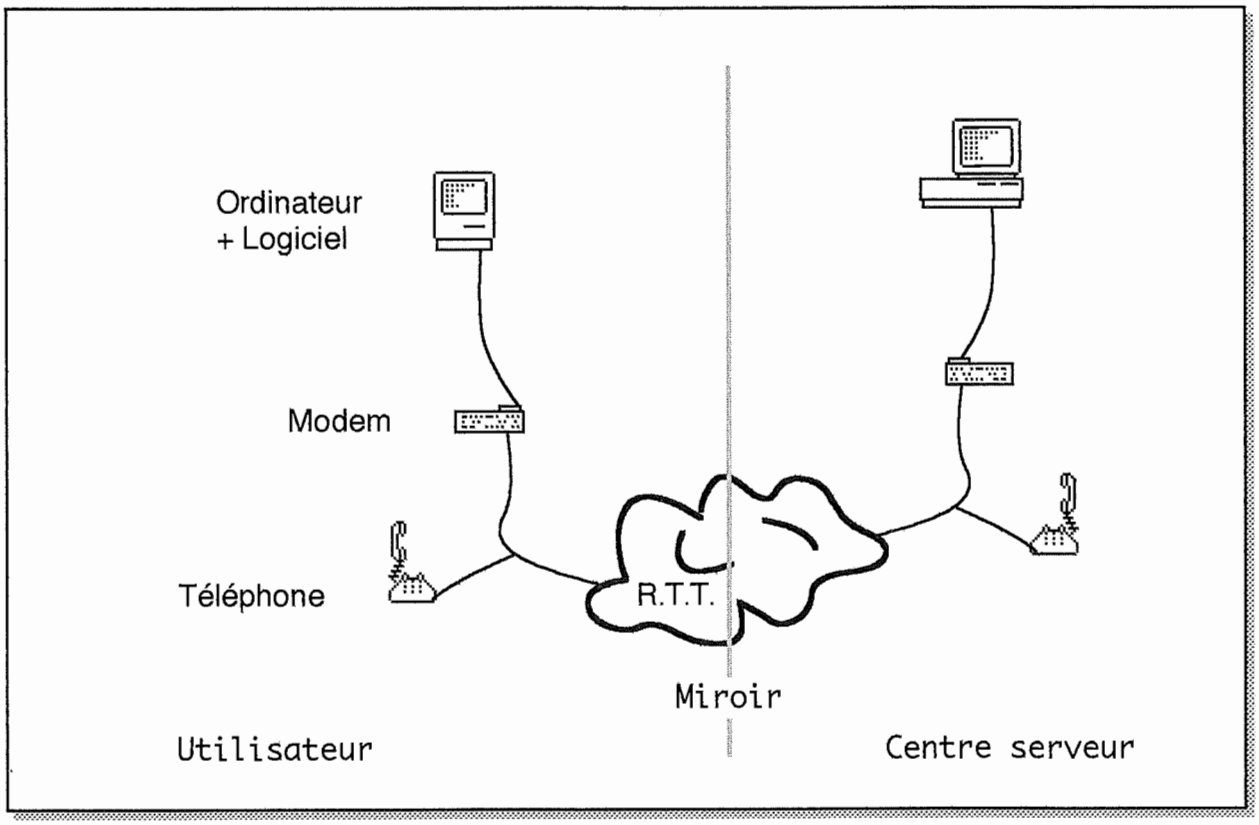


Fig. 2.1 : Le système de télécommunication par modem

Deux familles de modems sont commercialisés par Devlonics. Les modems externes à l'ordinateur font partie de la première. Dans ce cas, pour relier l'ordinateur au modem, un câble sera nécessaire. Les modems internes à l'ordinateur font partie de la seconde famille. Quelle qu'en soit la famille, tout modem doit être relié à la ligne téléphonique (fig. 2.2, page suivante). D'habitude, cela se fait au moyen d'une prise gigogne qui vient se placer entre la prise R.T.T. et la fiche du téléphone, bien que la présence d'un téléphone soit facultative.

Techniquement, un modem interne est un modem externe auquel on a ajouté une interface et supprimé l'alimentation, pour pouvoir le brancher à l'intérieur de l'ordinateur. Un électronicien n'a guère de difficulté pour transformer un modem interne en un modem externe. Cependant, les utilisateurs non averti estiment qu'un modem interne n'est pas équivalent à un

modem externe. C'est un peu comme s'il ne s'agissait pas de la même chose. Celui qui est à l'intérieur de l'ordinateur leur semble parfaitement intégré à l'environnement de travail proposé par la machine.

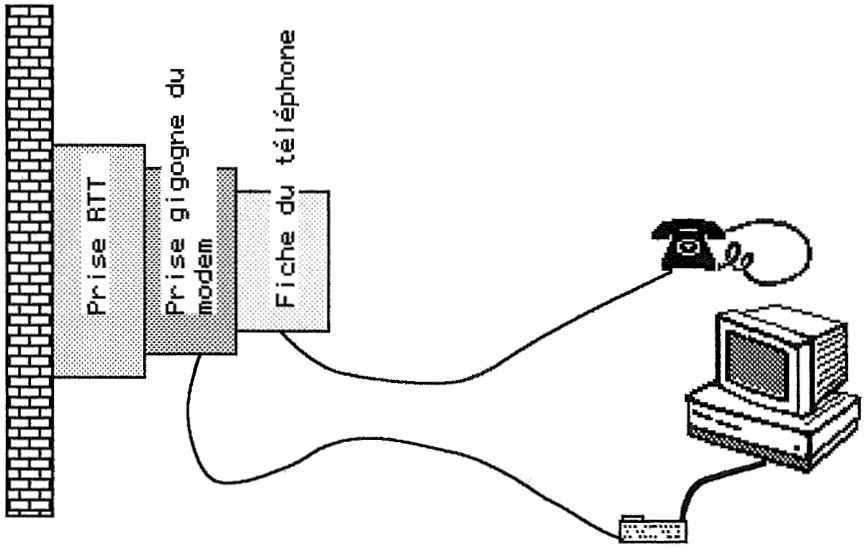


Fig. 2.2 : Raccordement du modem à la ligne téléphonique

L'accès à un serveur prend la forme d'un appel téléphonique. Une description précise des tâches étudiées est reprise en tête du chapitre suivant.

3. LES ACTEURS

Plusieurs acteurs sont concernés par les tâches étudiées. Nous analysons ces tâches afin de voir quels rôles ils y prennent. La vue de chacun d'entre eux précisera notre champ d'action.

3.1. ANALYSE DES TACHES

Nous limiterons cette étude à des utilisateurs qui procèdent à des tâches bien précises, ou qui exploitent ces tâches. Nous avons décrit au chapitre précédent, ce qu'est un système de télécommunication par modem. Au moyen d'un tel système, l'utilisateur établit une communication avec un centre serveur télématique. Une fois celle-ci établie, nous ne savons pas ce qu'il fait. Finalement, il la terminera. Nous prendrons en charge quatre tâches qui incombent à l'utilisateur : Régler son système pour qu'il puisse communiquer avec un centre serveur. Appeler le centre serveur. Archiver des écrans reçus pour pouvoir traiter l'information qu'ils contiennent par après. Echanger des fichiers.

•Régler son système pour qu'il puisse communiquer avec un centre serveur

Pour des raisons historiques, il faut préciser des paramètres pour pouvoir établir une communication avec un centre serveur. Par exemple, la vitesse de transmission, la manière de gérer l'écran, etc... Or seul des utilisateurs avertis en la matière, sont à même de les préciser.

Cette tâche fait appel à des notions très techniques. Elle demande d'être très méthodique pour ne pas se tromper dans les réglages. Sa complexité est grande parce qu'il n'existe pas de description standardisée des réglages et que ceux-ci peuvent varier fortement d'un centre serveur à l'autre, d'un logiciel de télécommunication à l'autre. Elle est effectuée en prenant le temps, chaque fois que l'utilisateur souhaite accéder à un nouveau centre serveur ou profiter de nouvelles particularités de celui-ci.

•Appeler le centre serveur

L'utilisateur entre en communication avec un centre serveur.

Cette tâche est effectuée au plus vite, chaque fois que l'utilisateur accède à un centre serveur. Les réglages du système devront êté correctement positionnés. Le serveur propose souvent une procédure d'identification stricte de l'utilisateur (et du terminal).

•Archiver des écrans reçus

L'utilisateur conserve une copie de ce qui a été affiché à l'écran de son terminal, afin de revoir certains écrans reçus, ainsi que pour effectuer des «copier-coller» avec d'autres applications.

Cette tâche est effectuée au plus vite, chaque fois que l'utilisateur veut se rappeler ce qui s'est passé au cours d'une communication. Elle est très fréquente quand il s'agit de revoir des extraits d'une communication. Elle est peu fréquente quand il s'agit de revoir le déroulement complet d'une communication. Nous verrons qu'elle permettra d'établir un pont vers la composition et l'organisation d'un serveur comme le demande Devlonics.

•Echanger des fichiers

L'utilisateur envoie des fichiers vers un centre serveur, ou reçoit des fichiers en provenance d'un centre serveur.

Cette tâche est effectuée au plus vite, chaque fois que l'utilisateur veut échanger un fichier avec le centre serveur. Elle fait appel à des notions techniques si le serveur s'en sert dans un dialogue avec l'utilisateur.

•D'autres tâches

A l'avenir, une panoplie d'autres tâches ayant trait à la composition et à la gestion d'un petit serveur télématique sera prise en compte.

3.2. L'ORGANISATION DEVLONICS[2]

Devlonics vient de *développements électroniques*. Devlonics groupe des experts en électronique (ingénieurs civils, ingénieurs industriels, gradués en électronique) sous le couvert de différentes sociétés. Une de ces sociétés est spécialisée dans la conception et le développement de modems (nommés par exemple : PC-mod12, PC-mod24,...). Le modem n'est pas un produit dont Devlonics a l'exclusivité. Nombreux sont les autres fabriquants de modems de par le monde. Ainsi, Banksys conçoit ses modems à Evere en Belgique et les fait fabriquer à Taïwan. Ce qui est propre à Devlonics, c'est de commercialiser ces modems sous plusieurs formes spécifiques comme par exemple en faisant le lien entre des caméras vidéo qui filment un ring et un centre de contrôle routier ou encore, entre des bornes télématiques situées dans des lieux de passages touristiques et un centre de téléchargement de ces bornes. Dans le cas qui nous intéresse, l'objectif de Devlonics est de placer ces modems chez des utilisateurs de services Vidéotex ou non, disposant d'un ordinateur personnel. Pour cela Eurocom, un logiciel de télécommunication approprié, a été mis en chantier. Ce logiciel est partiellement

[2] Avant sa faillite en février 1992

identique à ceux développés dans le cadre d'autres projets comme le contrôle routier. La réutilisation du code prend une importance particulière.

Devlonics cherche des niches de marché dans le domaine des nouvelles technologies de l'information. En 1987, Eurocom s'intégrait dans cet objectif car il répondait à une demande dans un secteur bien spécifique et sans concurrence. Il s'agissait de pouvoir se connecter à un centre serveur télématique qu'il soit ou non de type Vidéotex. Aujourd'hui, nombreux sont les logiciels qui en font autant (ClarisWorks, LCE-Com, ...) et de ce fait Eurocom requiert une nouvelle jeunesse aux yeux de Devlonics parce qu'il réprésente de moins en moins une niche de marché. Dans ce mémoire, le rôle de Devlonics a été de trouver comment rendre cette nouvelle jeunesse à Eurocom.

Pour qu'Eurocom procure à nouveau une niche de marché, une fonctionnalité de gestion de petit serveur Prestel va lui être ajoutée. Si cette ouverture s'avère intéressante, cette voie sera poursuivie. Sur cette lancée, Devlonics va faire une refonte d'Eurocom pour l'adapter à un environnement de type Windows et reévaluer l'interface homme-machine du produit.

3.3. L'UTILISATEUR

Notre souhait est de faire participer les utilisateurs dans la phase de conception de l'interface homme-machine. Dans ce mémoire, son rôle consistera à valider des propositions par l'intermédiaire de questionnaires ou de tests sur prototypes. Pour mieux cibler les utilisateurs, nous construirons une grille de leurs profils cognitifs. Nous montrerons au moyen d'une autre grille, celle des interfaces, que différentes interfaces homme-machine peuvent être envisagées par tâche et par profil cognitif. Pour tester l'interface, nous spécifierons les performances d'usage.

Profils cognitifs

Nous nous adressons à des utilisateurs valides qui se servent déjà d'un PC avec Windows ou d'un Macintosh pour d'autres applications. Ils n'auront pas nécessairement déjà utilisé un système de télécommunication par modem. Ils auront peut-être besoin d'un minimum de formation quand à l'installation du modem et à la notion de serveur. Ils savent se servir du téléphone, voire du fax.

Seul un ***utilisateur averti*** peut procéder à la tâche de réglage parce qu'il s'y retrouve dans les paramètres à régler pour communiquer avec un centre serveur. Cette personne connaît les concepts et le jargon qui rend cette tâche hermétique aux autres utilisateurs. Elle l'accomplit pour rentrer une première fois en communication avec tout centre serveur. L'***utilisateur non averti*** se limitera à l'exploitation des réglages effectués par des utilisateurs avertis, jusqu'au jour où il décidera de se former en la matière pour devenir un utilisateur averti.

Certains d'entre les utilisateurs (avertis ou non) se focalisent sur l'outil de communication en temps que tel, sur l'outil comme une fin en soi. Ils aiment entendre parler de paramètres, options, réglages typiques de tel ou tel modem, de tel ou tel serveur, dire que tel ou tel logiciel ne vaut rien ou est super. D'autres par contre se focalisent sur les services du serveur, sur

l'outil de communication comme un moyen en soi. Ils aiment pouvoir se connecter à un serveur sans avoir à dire comment, ils ne veulent pas être ennuyés avec le logiciel. Interrogés à propos des réglages, ces utilisateurs, même s'ils sont avertis, disent n'en avoir cure, ce n'est pas leur rayon. Les premiers que nous appellerons les ***utilisateurs passionnés***, seraient des fanatiques de la communication par modem. Les autres, que nous appellerons les ***utilisateurs non passionnés***, préfèrent un outil le plus transparent possible, surtout pas une béquille qui serait une solution qui ne leur facilite pas la tâche.

Des utilisateurs ont exprimé le désir intense d'un mécanisme similaire au fax avec lequel il n'y aurait plus aucun paramètre à préciser. Des initiatives dans ce sens existent à divers échelons comme le Minitel ou comme la négociation automatique de la vitesse de transmission entre modems. Quand aux autres utilisateurs, ils souhaitent que des réglages fins restent possible. Nous avons aussi entendu parler du coût élevé des communications et, de ce que le modem fait partie du monde de l'informaticien et pas de Monsieur tout le monde.

L'utilisateur averti et passionné va naturellement entrer en contact avec un grand nombre de centres serveurs. Pour lui, la fréquence de la tâche de réglage est assez élevée. Son comportement assez caractéristique a été observé avec l'aide de Philippe Lanssens qui est responsable du centre serveur du club d'informatique Micho à Tournai. Il accède par curiosité à un nouveau serveur; il en fait le tour, c'est-à-dire qu'il s'essaie aux services proposés et échange des fichiers. Après quelques tentatives ou quelques mois, selon ses accointances, il le délaisse.

L'utilisateur averti qui n'est pas passionné, va entrer en contact avec un groupe assez stable de centres serveurs, par obligation (professionnelle) ou pour d'autres motifs. Il effectuera les réglages occasionnellement. C'est-à-dire, une première fois pour rentrer en contact avec un serveur et ensuite, à chaque évolution qu'il trouve intéressante et qui concerne l'accès au serveur. Par exemple, un nouveau numéro de téléphonne rend l'accès moins onéreux. Un autre exemple : le serveur supporte un nouveau mode de communication aux caractéristiques plus attrayantes.

Quand à l'utilisateurs non-averti, cette tâche devrait lui être épargnée, à moins qu'il ne veule absolument s'y essayer. Ce fut le cas pour un utilisateur passionné par ailleurs par les radiocommunications. Pour eux, la fréquence de cette tâche ne sera pas très loin d'un multiple de zéro. Un utilisateur averti devra pouvoir pré-régler Polycom pour les besoins spécifiques de celui qui ne l'est pas.

Ci-dessous (fig. 3.1), nous reprenons un tableau récapitulatif des différents profils cognitifs des utilisateurs. De cette taxonomie, nous dégageons le besoin de différents types d'interface pour ces différents types d'utilisateur.

	Utilisateur passionné	Utilisateur non-passionné
Utilisateur averti	Il maîtrise les notions techniques du domaine des télécommunications par modem. Il s'implique dans le logiciel.	Il maîtrise les notions techniques du domaine des télécommunications par modem. Il ne s'implique pas dans le logiciel.
Utilisateur non-averti	Il ne maîtrise pas les notions techniques du domaine des télécommunications par modem. Il s'implique dans le logiciel.	Il ne maîtrise pas les notions techniques du domaine des télécommunications par modem. Il ne s'implique pas dans le logiciel.

Fig. 3.1 : Classification des profils cognitifs

Grille d'interfaces

Une grille nous permet de montrer que différentes interfaces doivent être envisagées par tâche et par profil d'utilisateur.

	Utilisateur passionné et averti	Utilisateur passionné et non averti	Utilisateur non passionné et averti	Utilisateur non passionné et non averti
Tâche de réglage. Requiert des notions techniques.	Rendre un maximum de réglage disponnible, p.e. au moyen d'un langage de programmation.	Expliquer le rôle des réglages pour qu'il puisse devenir un utilisateur averti le plus rapidement possible. Nous leur proposerons une formation.	Rendre un minimum de réglages indispensables. Un dialogue interactif au travers de boîtes de dialogue sera développé à leur intention.	Leur laisser exploiter les réglages effectués par une autre personne, p.e. au moyen d'une configuration pré-établie contenue dans un fichier.

Fig. 3.2a : Classification des interfaces (début)

	Utilisateur passionné et averti	Utilisateur passionné et non averti	Utilisateur non passionné et averti	Utilisateur non passionné et non averti
Tâche d'appel. Ne requiert pas de notion technique.	Un mécanisme lui permettra de commander le modem 'manuellement'. Utiliser la métaphore de l'appel téléphonique.	Utiliser la métaphore de l'appel téléphonique.	Utiliser la métaphore de l'appel téléphonique.	Utiliser la métaphore de l'appel téléphonique.
Tâche d'archivage. N'est pas sensé requérir de notions techniques.	Lui laisser revoir les derniers écrans reçus. Lui laisser grouper des recopies électroniques d'écrans dans un «album». Effectuer des copier-coller entre applications.	Idem. Utiliser des métaphores.	Idem.	Idem. Utiliser des métaphores.
Tâche d'échange. Requiert quelques notions techniques.	Utiliser les termes techniques repris par les serveurs.	Former l'utilisateur au moyen de métaphores à propos des termes techniques parce que ceux-ci sont repris par les serveurs dans les dialogues d'échange de fichiers.	Minimiser les manipulations pour effectuer cette tâche. Amoindrir l'aspect technique de la tâche.	Former l'utilisateur au moyen de métaphores à propos des termes techniques parce que ceux-ci sont repris par les serveurs dans les dialogues d'échange de fichiers. Utiliser des protocoles de transfert de fichiers qui minimisent l'intervention de l'utilisateur.

Fig. 3.2b : Classification des interfaces (fin)

Spécification des performances d'usage

La spécification des performances d'usage sert d'unité étalon pour évaluer l'adéquation de l'interface homme-machine. Elle a été déterminée avec l'aide de techniciens. Une fois l'équipement installé, les membres de clubs d'informatique, avertis en matière de

télécommunication et qui utilisent un environnement de type Windows, devront être capables, sans formation préalable, d'effectuer un appel sur le serveur BBL Home Banking en maximum 15 minutes, en n'utilisant que les supports d'apprentissage fournis. Si les réglages du système ont été correctement effectués à leur intention, les membres non avertis devront être capables d'effectuer ce premier appel en moins de 10 minutes. 90% de ceux qui auront réussi ce premier test devront être capables d'archiver et de revoir un écran, en moins de 10 minutes. Les durées ne tiennent pas compte de deux facteurs sur lesquels nous n'avons pas de prise : le temps d'établissement de la communication et le temps que l'utilisateur met pour être à même de naviguer au sein du serveur.

Les termes et les symboles utilisés devront être significatifs pour les utilisateurs concernés. Les réactions de Polycom devront être prévisibles.

3.4. LE TECHNICIEN

Dans ce mémoire, le technicien a un rôle de support. Il nous apporte des conseils, surtout en matière de télécommunication. L'objectif du technicien, chez Devlonics, est de développer un produit conforme aux visées de la direction, sous contrainte de respecter un cahier des charges qu'il aura négocié auparavant. Avec la figure 3.3 de la page suivante, nous trouvons l'exemple des prévisions de tâches et leurs durées respectives du cahier des charges d'Eurocom sous Windows. Ce cahier des charges est réalisé avant le début du développement du logiciel. Nous observons que "Interface utilisateur et fonctions d'aide" forment une seule tâche et que "Manuel de directives de mise en œuvre" en forme une autre. Ces deux tâches totalisent 7 homme/mois, soit près de 17% du travail estimé.

Le technicien ne travaille pas seul, il travaille en groupe. Les groupes sont formés au gré du développement des projets. Un nouveau projet peut réutiliser de l'expertise acquise au cours du développement d'un autre projet. C'est ainsi que des fonctionnalités semblables entre produits, peuvent être envisagées par différentes personnes dans des environnements de développement et des contextes d'utilisations variés.

Une petite histoire pour voir de quoi il est question : Au début Devlonics achetait des terminaux Vidéotex Philips. Ces terminaux ont l'aspect et les fonctionnalités du Minitel, au standard français. Devlonics changeait les ROM de ces machines, ainsi que la carte mère supportant l'électronique. Les ROM contenaient Maxitel, un logiciel de la maison aux possibilités plus étendues que celui d'origine. En plus d'offrir les fonctionnalités du Minitel, il offrait celles du Bistel, et plus tard celles d'un terminal télétype et VT100 avec un mini répertoire téléphonique. Fin 1991, une nouvelle version du boîtier Philips est sortie. Il avait changé de morphologie. Par exemple, le clavier était redessiné et ne disposait plus exactement des mêmes touches qu'auparavant. Devlonics en avait profité pour développer une autre carte mère en utilisant de nouveaux composants, économiquement plus intéressants.

En 1985, le marché des ordinateurs personnels tournant sous MS-DOS et des utilisateurs de systèmes Vidéotex semblait porteur. Le logiciel Maxicom de gestion de modems est né de cette demande. Développé sur base de Maxitel, Maxicom était ouvert à plusieurs protocoles de communication employés en Europe (le Télétel du Minitel français, le Prestel britannique repris

par la système Bistel belge) ainsi qu'au standard VT100 des terminaux d'ordinateurs centraux tel le VAX. Maxicom pouvait communiquer soit avec ces protocoles de gestion de l'écran, soit sans ces protocoles sous le mode télétype dit TTY. Au fil du temps Maxicom est devenu Eurocom. Aux protocoles déjà existants est venu s'ajouter l'allemand (le Bildschirmtext ou en bref : BTX). En mode VT100 ou TTY les correspondants pouvaient dorénavant s'échanger des fichiers avec le protocole XModem. Aujourd'hui, il s'agit de s'adapter à Windows.

Projet: VIDEOTEX PROCESSABLE DATA TERMINAL	
Durée: 17 mois	
Environnement: Microsoft Windows v3.0	
Tâche	Homme/Mois
Décodeur des protocoles T et D	5
Décodeurs Vidéotex	4
Editeurs Vidéotex	4
Imprimante virtuelle Vidéotex	3
BulkUpdate Prestel et Viditel	4
Support d'un réseau local	2
PageUpdate Télétel	3
Conversion Ecran Vidéotex -> BitMap	0,5
Conversion BitMap -> Ecran Vidéotex	3
Interface utilisateur et fonctions d'aide	5
Téléchargement du Terminal -> Serveur	2
Tests	4
Manuel de directives de mise en œuvre	2

Fig. 3.3 : Cahier des charges d'Eurocom

Quand à Maxitel, il n'utilisait plus que le châssis du terminal de Philips. Devlonics toujours à la recherche de niches de marché, a décidé de produire elle-même des chassis sous forme de bornes télématiques. Ce sont ces bornes vertes en forme de champignon à chapeau hexagonal que l'on voit dans des centres touristiques ou de passage, plutôt en Flandres.

Au cours du temps, un logiciel de base sera repris à toutes les sauces. Les techniciens articuleront le code et l'interface homme-machine selon leurs exigences du moment.

3.5. LE VENDEUR

Dans ce mémoire, le vendeur a un rôle de support. Il nous apporte des conseils, surtout en matière de relations à l'utilisateur. Son action est basée sur un réseau de revendeurs. En contact avec le marché, ses investigations sont précieuses pour observer des niches de marché, établir un pont entre les clients et les techniciens, déterminer des opportunités ponctuelles. En bref, il sert de lien et de filtre entre le monde intérieur, c'est-à-dire ce qui se passe chez Devlonics, et le monde extérieur, c'est-à-dire ce qui se passe sur les marchés.

Ainsi, il négociera des journées de formation/présentation de produits. Il trouvera sur le marché des produits complémentaires à ceux déjà proposés par Devlonics, comme Instant/Com qui permet de partager des modems sur un réseau local. Il fera des enquêtes sur l'état global du marché avec l'aide de son réseau de revendeurs et vendra.

4. METHODOLOGIE

L'utilisateur et l'ordinateur communiquerons ensemble pour accomplir les tâches que nous étudions. Or, en matière de communication, ils sont sur des planètes différentes. Alors que les personnes s'expriment sur base de leurs raisonnements, l'ordinateur présente des données sur base de l'exécution de programmes. L'interface homme-machine que nous allons développer devra d'une part amener la personne à s'exprimer le plus naturellement possible en des termes reconnus par la machine et, d'autre part, elle devra amener la machine à présenter des données en des termes reconnus le plus naturellement possible par l'utilisateur. Ce chapitre nous permettra de présenter brièvement le modèle de l'activité de l'utilisateur qui a été développé par D. A. Norman. Ensuite nous aborderons la notion du "plus naturellement possible". Finalement, nous verrons comment prendre en compte les possibilités de l'utilisateur et de l'ordinateur au sein du développement de l'interface homme-machine de Polycom.

4.1. MODELISATION DE L'ACTIVITE DE L'UTILISATEUR

Dans le cadre d'applications interactives, D. A. Norman a développé un modèle de l'activité de l'utilisateur pour établir des ponts entre la façon de communiquer de celui-ci et celle de l'ordinateur. Observons le schéma de la page suivante. On y distingue une personne et un ordinateur placés sur deux berges séparées par un golfe (qui paraît bien petit mais le dessin n'est pas à l'échelle). Une flèche va de la personne vers l'ordinateur. Elle passe sur un premier pont. Une autre flèche fait le trajet inverse. Elle passe sur un second pont.

Le premier pont (fig. 4.1, page suivante) va de la personne vers la machine, il enjambe le golfe d'exécution. L'utilisateur a l'***intention (1)*** de réaliser un tout indiscernable qui, d'après lui, est directement dans les cordes du logiciel. Pour communiquer avec l'ordinateur, l'utilisateur devra transformer l'intention qu'il a en une ***séquence d'opérations (2)*** qu'il lui demandera d'***exécuter l'une après l'autre (3)***. Le second pont va de la machine vers la personne, il enjambe le golfe d'évaluation. Le logiciel présente des données qui devront symboliser de l'information pour l'utilisateur. Le programmeur devra s'arranger pour que chaque ***élément perçu (1')*** par l'utilisateur fasse partie d'une séquence que l'utilisateur ***interprète (2')*** dans son ensemble pour finalement ***évaluer (3')*** si elle correspond bien à ce à quoi il s'attend. Pour la personne, l'accent est placé sur les capacités cognitives acquises et, pour la machine, l'accent est placé sur les capacités à présenter des données.

Passons en revue les étapes du modèle de D. A. Norman avec l'exemple d'une personne qui souhaite connaître l'état de son compte bancaire au moyen du BBL Home Banking. L'ordinateur est un Macintosh. La tâche de l'utilisateur consiste à ouvrir un document reprenant les réglages, puis à donner l'ordre à Polycom d'établir la communication. Le numéro de chaque étape fait référence au schéma du modèle de Norman (fig. 4.1, page suivante):

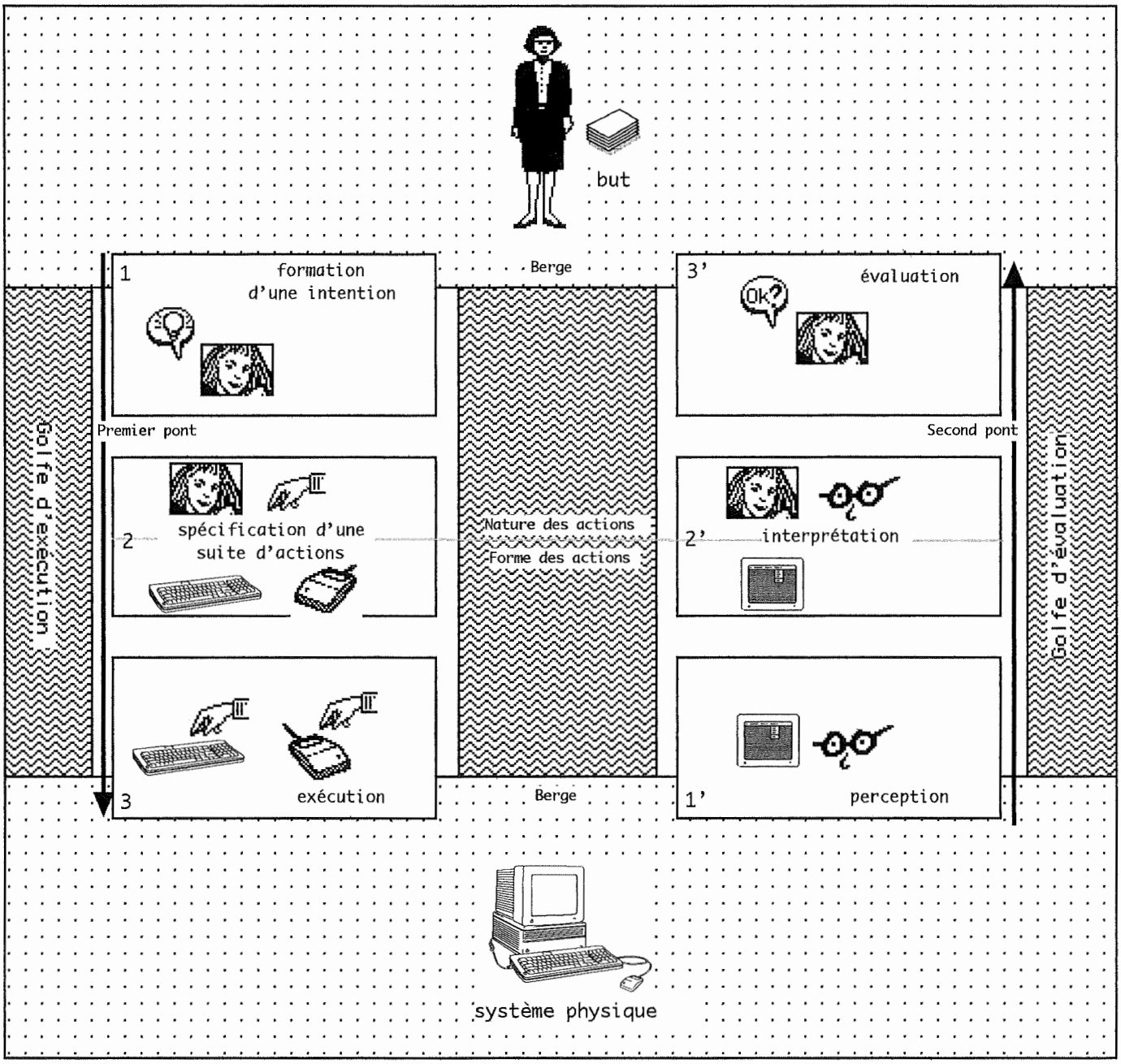


Fig. 4.1 : Schéma du modèle de Norman

·But de la personne : Connaître l'état de mon compte

·1) [Sur la fig. 4.1, on descend le golfe d'exécution; on est à la formation d'une intention]. Je dois entrer en contact avec l'ordinateur de la banque. C'est possible en utilisant un document tout préparé qui permet d'établir la liaison.

·2) Côté utilisateur : [Sur la fig. 4.1, on est à la spécification d'une suite d'actions] Sélectionner le document de Polycom qui me permet d'établir la liaison. Ouvrir ce document.

·2) Côté machine, pour 'Ouvrir' le document : [Sur la fig. 4.1, on est à la spécification d'une suite d'actions] Cliquer une première fois sur l'icone ou le nom qui tout deux symbolisent ce document, pour l'activer. Ensuite soit cliquer une seconde fois peu de temps après, sans avoir déplacé la souris hors de l'icone ou du nom; soit sélectionner l'item 'Ouvrir' du menu 'Fichier' en suivant la procédure ad hoc.

·3) [Sur la fig. 4.1, on est à l'exécution] Quand le pointeur de la souris se trouve sur l'icone qui symbolise le document : L'utilisateur se sert de son index pour enfoncer deux fois de suite le bouton de la souris.

```
·1') [Sur la fig. 4.1, on remonte le golfe d'évaluation; on est à la perception].
     Au premier clic, l'icone et le nom du document sont dessinés en blanc sur
     fond noir. Au second clic, des traits pointillés qui forment un rectangle
     s'agrandissent par pas de 20 points depuis la position de l'icone jusqu'à
     une taille de 2/3 de l'écran. Des mots sont affichés sur la partie
     supérieure de l'écran, etc…
·2') Côté machine : [Sur la fig. 4.1, on est à l'interprétation] Au premier clic,
     l'icone et le nom du document sélectionné sont mis en relief. Au second
     clic, l'image du pourtour d'une fenêtre qui s'ouvre est renvoyée. La barre
     du menu principal de la nouvelle application est affichée, etc…
·2') Côté utilisateur : [Sur la fig. 4.1, on est à l'interprétation] J'ai
     sélectionné le document. Il s'ouvre. Polycom affiche son menu, etc…
·3') [Sur la fig. 4.1, on est à l'évaluation] Ok, le document est ouvert.
·1) Maintenant je dois établir la liaison téléphonique avec l'ordinateur de la
     banque. C'est quelque chose que Polycom me permet de faire, etc…
```

Les étapes pivots entre l'homme et la machine sont celles où l'on passe plus spécialement de l'environnement mental de la personne à son environnement physique. C'est-à-dire quand l'utilisateur traduit la nature de la séquence d'opérations à réaliser en une séquence d'opérations formellement réalisables avec l'équipement informatique, et inversément. C'est ainsi que nous avons abordé les étapes 2 et 2' sous les angles de l'utilisateur et de la machine. Nous l'avons marqué par un trait pointillé sur le schéma du modèle de D. A. Norman. Au dessus de ce trait, du côté de la personne, nous parlons de la nature de l'action. En dessous de ce trait, du côté de la machine, nous parlons de la forme de l'action. Cette marque met en relief la présence d'un langage de l'interface, défini dans [Bod89 p10] comme : "le langage utilisé pour indiquer aux dispositifs de l'interface la nature des actions que l'on veut exécuter".

4.2. NOTION DU "PLUS NATURELLEMENT POSSIBLE"

L'expression "le plus naturellement possible" recouvre l'utilisation de procédés comme la technique des indices de rappel ou, de la reconnaissance [Lie80], ainsi que l'application de critères et règles d'ergonomie de l'interface [Bod89], [Tog92], [Van92] et [Mei91] . Le procédé des indices sert à améliorer le rappel de la mémoire naturelle. Pour montrer l'intérêt d'un procédé de rappel, on le compare avec le rappel libre défini comme "une épreuve qui consiste à évoquer sans aide des éléments ayant été mémorisés ou appris" [Lie80 p189].

La technique des indices est définie par A. Lieury [Lie80 p119] : "Un ***indice de rappel*** est un groupe d'informations de nature variée qui permet de récupérer de la mémoire [naturelle], un ensemble structuré déjà stocké (mots, idées, souvenirs)". C'est par exemple un cliché qui rappelle toute une série de souvenirs, mais cela peut être aussi une association sémantique (comme : caméra -> film, filmer, visionner, bobine, cassette), une odeur, une musique, etc... L'auteur cite [Lie80 p120] deux conditions pour que les indices soient efficaces : l'association préétablie et l'intégration des éléments.

> "•L'association préétablie: Dans les expériences de laboratoire où des mots étaient à apprendre, une condition nécessaire à l'efficacité des indices a été mise en évidence, c'est le caractère préétabli de la liaison entre l'indice et le mot à mémoriser. Il faut par exemple, qu'il existe une liaison sémantique entre les deux mots, abeille et miel par exemple, «abeille» servant ultérieurement d'indice pour aider à rappeler «miel». Une liaison arbitraire ne sera pas efficace : ainsi, si je fais apprendre une liste de mots, parmi lesquels se trouve «miel» et que je donne «train» comme indice de rappel pour retrouver «miel», cet indice ne sera pas efficace (sauf si je remplis la seconde condition)[...].
>
> •L'intégration des éléments: Lorsqu'il n'y a pas de liaison préétablie entre les éléments, par exemple, entre le mot qui sert d'indice et le mot qui est à rappeler, il est nécessaire qu'il y ait intégration de ces deux mots au moment de leur mémorisation [...] L'intégration de deux mots peut se faire au moyen de la médiation d'une

> image ou d'une médiation verbale, mot ou phrase. Ainsi, si je dois apprendre «train-miel» en sachant que l'on me fournira «train» comme indice, j'essaierai d'intégrer ces deux mots en formant une image «un train transportant du miel» et/ou une phrase «je prends le train pour ma lune de miel».»

La technique de la reconnaissance est décrite par A. Lieury [Lie80 p26] : "Il suffit de présenter l'information originale (le mot ou la photo, etc.) parmi des informations pièges qui n'ont pas été présentées; le sujet doit reconnaître l'information parmi les pièges". L'auteur présente une expérience de Harry et Phyllis Bahrick et Roy Wittlinger (1975) pour comparer la reconnaissance au rappel libre.

> "Ces auteurs ont eu l'idée de retrouver des personnes plusieurs années après leur départ du collège et de sonder leur mémoire à propos des noms et des photographies des camarades de leur collège. La reconnaissance des photographies est très bonne et stable puisqu'elle est de 90% pour des gens ayant quitté le collège depuis 35 ans. La reconnaissance des noms est également bonne et stable, quoique légèrement plus faible, puisqu'elle passe de 90% après 3 mois à 80% après 35 ans. Ceci confirme d'ailleurs les résultats de laboratoire qui montrent que les images sont mieux retenues que les mots. En revanche, le rappel libre de ces noms de camarades est très faible (l'épreuve ayant été faite avant la reconnaissance, bien entendu): 15% de noms sont rappelés après 3 mois et 12% après 35 ans. La capacité d'attribuer des noms aux photographies (rappel indicé) a également été testée. Elle est de 70% après 3 mois et diminue régulièrement pour atteindre 33% au bout de 35 ans."

Ci-dessous nous reprenons un tableau extrait de [Lie80 p125]. Il rend compte de l'efficacité du type de rappel (pourcentage de rappels corrects) en fonction du délai, pour des noms de camarades de collège (d'après H. et P. Bahrick et R. Wittlinger, 1975):

	Délai de rappel		
	3 mois	25 ans	50 ans
Rappel libre	15%	13%	7%
Rappel indicé	70%	40%	18%
Reconnaissance	90%	80%	70%

Ce tableau montre que le rappel indicé est supérieur au rappel libre et la reconnaissance est supérieure au rappel indicé, que le délai de rappel soit de 3 mois, de 25 ans ou de 50 ans. ***Nous retiendrons de cette expérience que*** la tâche de l'utilisateur d'un programme sera facilitée si l'utilisateur est l'impliqué dans un processus de reconnaissance plutôt que dans un processus de rappel libre. Par exemple, il lui sera plus facile de retrouver un item dans un menu que de l'encoder par rappel libre. Nous retiendrons aussi que les associations sémantiques préétablies peuvent servir efficacement d'indices de rappel si elles sont effectivement préétablies. Cela sera généralement le cas pour «miel» et «abeille», mais qu'en est-il de «terminal», «VT102» et «Bildschirmtext» ou encore «Télétel 80 colonnes», «ISO 6420» et «normalisation VT100»? Particulièrement en matière de télécommunication, des spécialistes émettent des concepts. Des médiateurs scientifiques en font un touillage à destination de l'homme de la rue qui ne sait plus quoi. Cette technique de l'association sémantique qui dépend des acquis de l'utilisateur, nous demandera de savoir quels éléments il associe à quel indice. Nous pensons que de bons indices lui donneront l'impression de prédire ce qu'il peut faire ou ce qu'il va se passer. Nous avons par exemple observé que les utilisateurs s'inquiétaient particulièrement du numéro de téléphone du centre serveur quand il s'agissait de l'appeler. En parlant avec eux, nous avons constaté l'association «correspondant», «numéro de téléphone» et «appeler». Dans le menu où se trouvait l'item «appeler», nous avons placé l'item «correspondant» juste au dessus de «appeler». Ensuite,

nous avons demandé à quatre autres utilisateurs de changer le numéro de téléphone du centre serveur. Ils ont directement sélectionné «correspondant».

4.3. METHODOLOGIE

Comme nous l'avons présenté au chapitre introductif, la méthodologie adoptée est calquée sur la modélisation de l'interface homme-machine comme un langage. ***Comment prendre en compte les possibilités de l'utilisateur au sein du développement du langage de l'interface?*** L'accent sera mis sur les possibilités cognitives de la personne dans le but de rencontrer la manière dont la machine se comporte. Il s'agit de réduire la ***distance sémantique*** entre les objectifs de l'utilisateur et la signification d'une expression dans le langage de l'interface. Par exemple, la distance sémantique entre "connaître l'état de son compte" et entrer en communication avec l'ordinateur central de la banque, s'identifier, sélectionner son compte, donner son mot de passe, en demander l'état. L'utilisateur peut aussi donner un coup de fil à son banquier ou jetter un coup d'œil à sa liasse d'extraits. La notion de distance sémantique est intuitive. Nous n'avons pas d'outil pour la mesurer. Pour la cerner, nous allons procéder comme suit :

- ***Au niveau pragmatique***, nous nous placerons au niveau des objets sous-jacents à Polycom. Notre but sera de dégager un modèle pragmatique qui soit reconnu des utilisateurs pour réaliser les tâches étudiées. Ce modèle, porteur d'une réalité pour l'utilisateur, se composera d'opérations qui s'articulent sur des objets de l'application. Par exemple l'objet «compte bancaire» dont on peut connaître l'«état». Pour l'un, son compte est géré dans un ordinateur de la banque. Pour l'autre, il est géré par son banquier. Le choix des objets de l'application s'est fait suivant leur efficacité comme indices de rappel.

- ***Au niveau sémantique***, nous nous placerons sur le versant sémantique de l'étape pivot. Notre but sera de décomposer les opérations de l'utilisateur en une séquence de comportements, selon les références qu'il établit envers les éléments du modèle pragmatique. Par exemple, quel cheminement d'opérations effectuer pour connaître l'état de son compte? Pour l'un, on y accède par un logiciel de télécommunication. Il s'agira ainsi de recourir au Home Banking en suivant une procédure très stricte. Pour l'autre, il suffit de donner un coup de fil à son banquier. Dans ce mémoire nous harmoniserons autant que possible l'ensemble des comportements des utilisateurs pour en dégager une seule formalisation. Nous nous servirons de critères d'ergonomie tels que repris dans [Bod89a], [Bod89b pp57-58], [Cou90a pp57-88], [Mei90 pp101-103], [Sca pp13-14], [Tog92].

Comment prendre en compte les possibilités de l'ordinateur au cours du développement du langage de l'interface? L'accent sera mis sur les possibilités matérielles de la machine dans le but de rencontrer la manière dont l'utilisateur se comporte. Il s'agit de réduire la ***distance articulatoire*** entre la signification d'une expression dans le langage de l'interface et sa forme au sein des moyens d'interactions. Par exemple, on parlera de la distance articulatoire entre "connaître l'état de son compte via le Home Banking" et

cliquer deux fois consécutivement sur le symbole qui représente le compte, taper son mot de passe, cliquer sur un item de menu. Même cas si l'utilisateur encode le nom du logiciel de télécommunication suivi de son identification, du numéro du compte, et de l'option «/V». La notion de distance articulatoire est intuitive. Nous n'avons pas d'outil pour la mesurer. Pour la cerner, nous allons procéder comme suit :

•***Au niveau syntaxique***, nous nous placerons sur le versant syntaxique de l'étape pivot. Notre but sera de faire correspondre le mieux possible chaque séquence de comportement du niveau sémantique en un groupement d'opérations formelles sur des objets interactifs abstraits. Prenons l'exemple cité au paragraphe précédent où on a présenté plusieurs cheminements d'opérations à effectuer pour effectivement connaître l'état de son compte via le Home Banking. Nous nous servirons de recommandations sur l'ergonomie telles que reprises dans [Van92], [Sca], [Tog92], [Cou90a pp57-88], [Mei90 pp103-110].

•***Au niveau lexical***, notre but sera de déterminer la forme de chaque échange entre l'homme et la machine au moyen d'objets interactifs concrets qui symbolisent au mieux une réalité pour l'utilisateur. Par exemple, comment représenter graphiquement un compte sur l'écran de l'ordinateur de l'utilisateur pour qu'il se rende compte que cela symbolise un compte et qu'il lui soit aisé de cliquer dessus. Ou encore, l'option «/V» est-elle préférable à une autre orthographe? Nous nous servirons de recommandations sur l'ergonomie telles que reprises dans [Van92], [Tog92] ou dans la littérature des différents constructeurs selon l'environnement ciblé par l'implémentation [Apple, IBM, Microsoft].

5. NIVEAU PRAGMATIQUE

Pour rappel, le niveau pragmatique porte sur une réalité humaine, c'est-à-dire "the study of those aspects of language that cannot be considered in isoloation from its use".

Aux deux niveaux pragmatique et sémantique, nous abordons l'analyse de la distance sémantique entre les objectifs de l'utilisateur et la signification d'une expression dans le langage de l'interface. Chaque fois que l'utilisateur aura un objectif à atteindre, il devra l'exprimer en des termes reconnus par le logiciel. Or, il a une certaine appréhension de son environnement. Il exprime ses buts dans des termes et sur des objets qui ont un sens pour lui. Prenons l'exemple suivant : un membre du club d'Aviron de Tournai parlera de "sortir les pelles du Polychinel". Le Polychinel est un bateau (un skiff) conçu pour un rameur unique et les pelles sont des rames (des avirons). Les membres des clubs d'Aviron savent que deux avirons sont nécessaires pour manœuvrer un skiff. Sortir quelque chose signifie porter l'objet depuis le hangar où tout est stocké jusqu'à la zone d'embarquement. On sort un bateau pour le mettre à l'eau et on sort des pelles pour les mettre sur la rambarde du ponton. Pouvons-nous demander de remettre les pelles en place? ou, que le Polychinel bêche avec? ou, qu'on essuie le Polychinel avant de le rentrer? A quoi l'indice «aviron» nous fait-il penser?

Ainsi, l'utilisateur devra se faire comprendre par la machine. Lorsqu'il aura un but à atteindre, il établira une méthode pour l'atteindre sur base de la représentation mentale qu'il se fait du système. La spécification pragmatique a pour but de mettre en évidence la description des objets importants de l'application et, de les faire correspondre à des connaissances acquises de l'utilisateur parce que celui-ci donnera un sens à ces objets par rapport à ses acquis. Il se basera dessus pour savoir "Que puis-je faire faire par la machine, qui me soit utile?". D. A. Norman affirme [Nor83]:

> "My observations on a variety of tasks, with a wide variety of people, lead me to a few general observations about mental models:
> • Mental models are incomplete.
> • People's abilities to "run" their mental models are severely limited.
> • Mental models are unstable : People forget the details of the system...especially when (the system has) not been used for some period.
> • Mental models do not have firm boundaries : Similar devices and operations get confused for one another.
> • Mental models are "unscientific" : People maintain "superstitious" behavior patterns even when they know they are unneeded because they cost little in physical effort and save mental effort.
> • Mental models are parsimonious : ...People...are willing to trade off extra physical actions for reduced mental complexity."

L'objectif de la démarche au niveau pragmatique est de fixer des objets significatifs sous-jacents aux expressions du langage de l'interface. Nous nous plaçons aux étapes de la formation d'une intention (1) et de l'évaluation (3') sur le schéma du modèle de Norman (fig. 5.1, page suivante). Nous réduirons la distance entre la signification que l'utilisateur donne aux objets mis en œuvre par Polycom et, celle que nous leur attribuons au sein de l'interface. A partir de ses acquis, l'utilisateur devrait pouvoir en inférer plus facilement le comportement

du logiciel eu égard aux instructions données. Ainsi, il sera à même de décomposer un objectif qu'il a en une suite de tâches que les objets mis en œuvre par Polycom semblent lui permettre d'appréhender. Pour réduire cette distance, nous nous servirons de mots indices et de métaphores qui symbolisent des objets du monde réel reconnus par l'utilisateur et qui s'articulent pour permettre d'effectuer les tâches étudiées. Par exemple, l'ensemble des serveurs accessibles formera un annuaire. Les propriétés qu'on a l'habitude d'associer à un annuaire seront appliquables : un annuaire téléphonique s'ouvre à une lettre de l'alphabet.

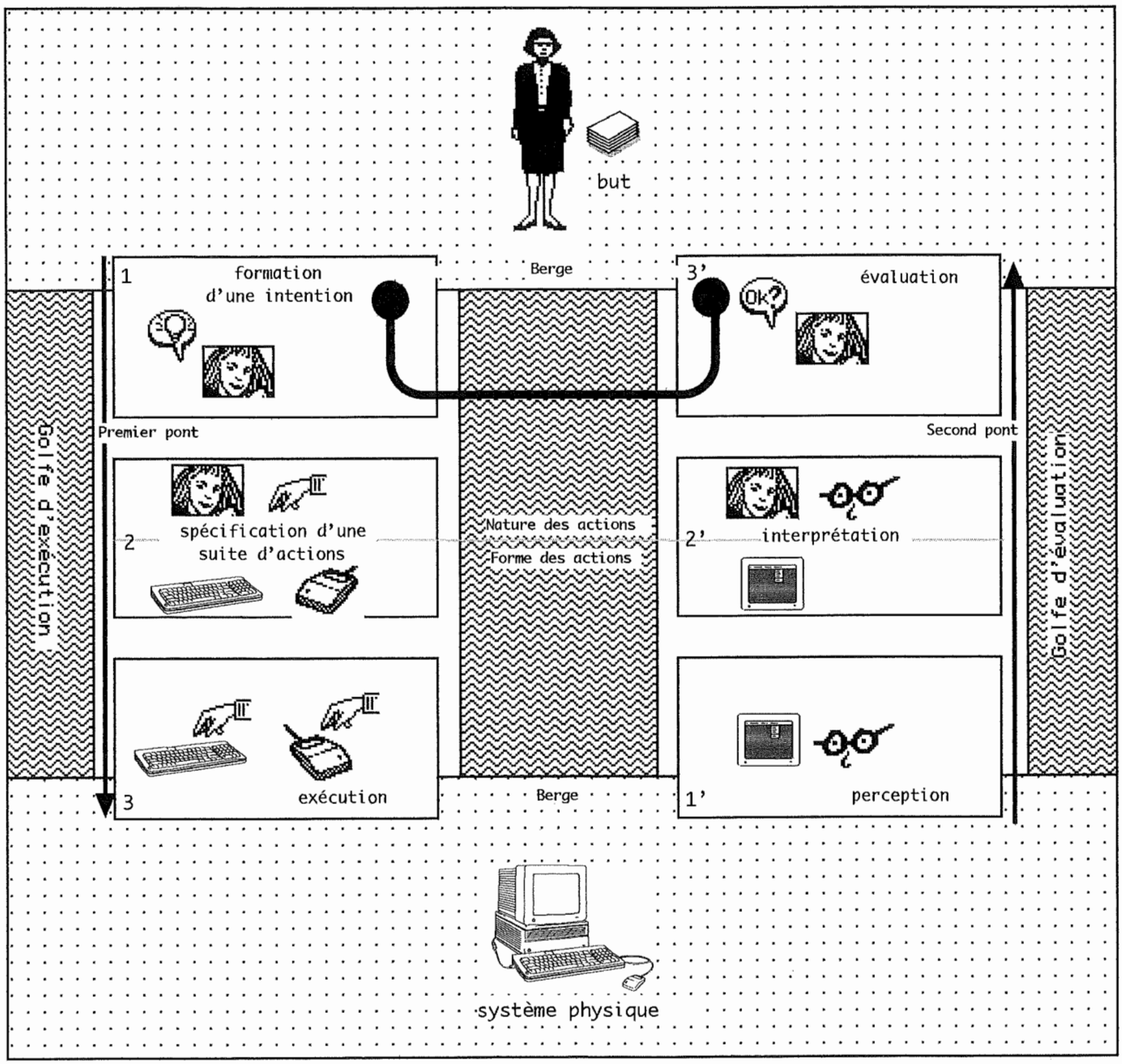


Fig. 5.1 : Le trait épais terminé par deux boules marque le niveau pragmatique sur le schéma du modèle de Norman

Pour chaque tâche de l'utilisateur, nous décrivons quels pourraient être les objets mis à sa disposition, leurs attributs et, les opérations que l'on peut effectuer dessus. ***Nous avons adopté une démarche d'approximations successives***. Les premiers objets proposés aux utilisateurs concernaient la tâche de réglage du système et ils furent directement calqués sur la classification en différents modules de l'application. ***Ces objets proposés a priori étaient des composants d'un modèle personnel*** qui fut invalidé, comme nous allons le découvrir par après au moyen d'un questionnaire. Le choix des objets de l'application se fera suivant leur efficacité comme indices de rappel.

5.1. MODULES DE L'APPLICATION

A priori, nous avons établi une classification de l'application en différents modules. Cette classification n'a pas pour but de s'intégrer en quoi que ce soit avec le modèle en couche d'OSI. Dans ce mémoire, elle permettra de refléter l'écart entre une classification possible des modules de Polycom et une autre classification d'objets de l'application afin de développer un langage de l'interface. Ces modules qui collent à la configuration technique du système, permettent déchanger des données entre l'ordinateur des correspondants. Il s'agit : du lien de l'ordinateur au modem, du lien du modem au réseau téléphonique, du lien de modem à modem et du lien d'ordinateur à ordinateur.

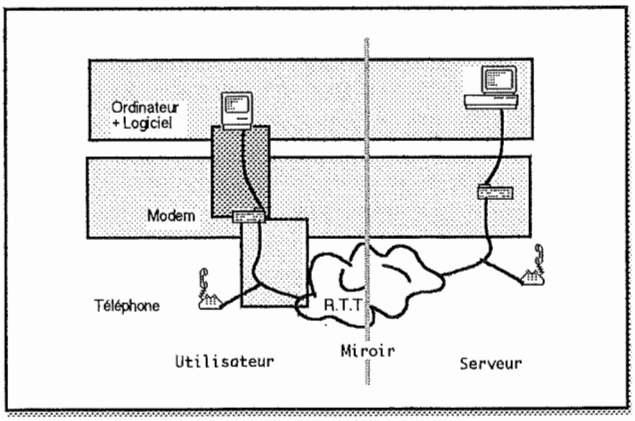


Fig. 5.2 : Lien de l'ordinateur au modem

Le lien de l'ordinateur au modem (fig. 5.2) : concerne ce qui est nécessaire pour connecter l'ordinateur au modem. Au niveau matériel, on peut connecter physiquement le modem à l'ordinateur soit à l'intérieur de l'ordinateur, soit à l'extérieur de l'ordinateur au moyen d'un câble spécial. Dans ce dernier cas, il faudra prévoir de quoi alimenter le modem en électricité indépendamment de l'ordinateur. Dans l'architecture du logiciel, ce lien est le module d'accès au modem. Il échange les données entre l'ordinateur et le modem. Il tient compte des caractéristiques du modem adressé.

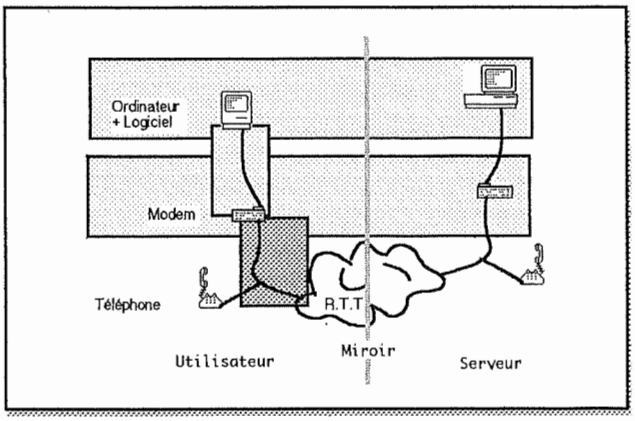


Fig. 5.3 : Lien du modem au réseau téléphonique

Le lien du modem au réseau téléphonique (fig. 5.3) : ce lien concerne tout ce qui est nécessaire pour relier le modem de l'utilisateur à la prise RTT du réseau téléphonique. Au niveau matériel, on peut intercaler la prise téléphonique gigogne du modem entre celle du téléphone et celle de la RTT, ou déconnecter le téléphone et connecter le modem à la place. Dans l'architecture du logiciel, ce lien est le module de gestion du raccordement au réseau téléphonique. Il informe le modem des caractéristiques de la ligne téléphonique adressée.

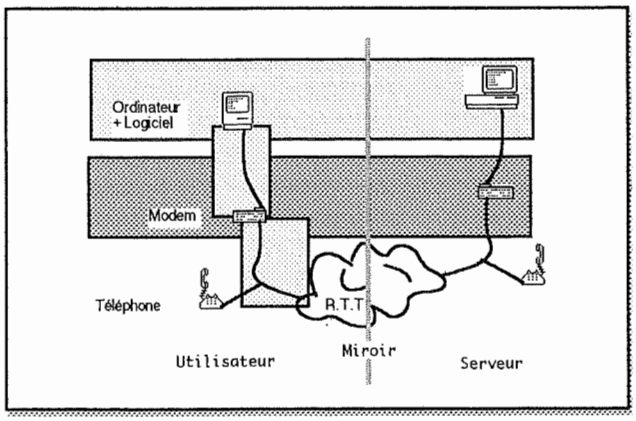


Fig. 5.4 : Le lien de modem à modem

Le lien de modem à modem (fig. 5.4) : ce lien concerne le flux de données entre les modems des correspondants. Au niveau matériel, le modem de l'utilisateur doit se conformer aux exigences de celui du centre serveur pour pouvoir communiquer avec. On parle de compatibilité entre les modems. Dans l'architecture du logiciel, ce lien est module de gestion de la liaison téléphonique. Il gère la liaison avec le modem du centre serveur (par exemple : la ligne est occupée, elle est coupée).

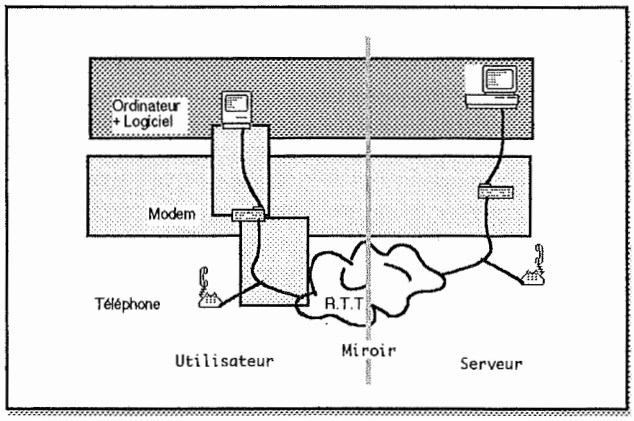


Fig. 5.5 : Le lien d'ordinateur à ordinateur

Le lien d'ordinateur à ordinateur (fig. 5.5) : ce lien concerne le flux de données entre les ordinateurs des correspondants. Au niveau matériel, l'ordinateur de l'utilisateur doit se conformer aux exigences de celui du centre serveur pour pouvoir communiquer aisément avec lui (par exemple : il faut disposer d'un clavier et d'un écran adéquats). Contrairement au lien de modem à modem qui dépend des caractéristiques des modems, ce lien dépend directement des fonctionnalités offertes par Polycom. En effet, ce dernier fera la correspondance entre le matériel disponnible et les exigences du serveur. Dans l'architecture du logiciel, ce lien est le module d'émulation de terminal. Il gère la liaison avec l'ordinateur du centre serveur en faisant abstraction de la liaison téléphonique.

Les fonctions qui nous intéressent concernant ces modules sont celles qui demandent une intervention de la part de l'utilisateur pour régler le logiciel (par exemple, les deux modems seront réglés sur la même vitesse) et le matériel (par exemple, les bons câbles seront raccordés aux bonnes prises) pour pouvoir communiquer avec un centre serveur. Si les réglages ne sont pas correctement effectués, alors la communication posera problème et l'ordinateur de l'utilisateur ne rentrera peut-être pas en contact avec celui du centre serveur.

Force est de constater qu'il n'existe pas de liste standardisée des réglages. On est dans un épais brouillard historique où on essaie souvent de réinventer des paramètres à régler. Chaque logiciel et chaque serveur proposent une manière originale d'aborder ce problème. Dans ce domaine, les logiciels sont souvent à la traîne sur les nouveautés techniques dont les serveurs disposent.

Avec l'aide d'un sysop (c'est-à-dire un responsable d'un serveur, cela vient de system operator) , nous avons établi une liste minimale des réglages à prendre en compte pour les quatre liens cités. Nous nous sommes basé sur ceux d'Eurocom et sur ceux que les serveurs communiquent habituellement, mais pas toujours, à l'utilisateur pour qu'il puisse (tenter de) régler son logiciel. Cette liste est destinée aux utilisateurs avertis qui ne sont pas passionnés. Au niveau pragmatique, dans lequel nous sommes, nous avons alors traduit cette liste en objets significatifs pour les utilisateurs avertis:

- **Vitesse du modem** : La vitesse de transmission sur la ligne téléphonique. Ou bien elle est exprimée en bits par seconde, ou bien elle est implicitement cachée sous

le couvert des normes du CCITT ou de Bell. Avec le tableau de correspondance suivant:

Norme du CCITT	Vitesse (en bits par seconde) dans le meilleur des cas	Tombée automatique en vitesse si nécessaire
V21	300	
V22	1.200	
V22 bis	2.400	1.200
V23	1.200, aller/75, retour	
V32	9.600	4.800
V32 bis	14.400	12.000, 9.600, 7.200, 4.800

La tombée automatique en vitesse permet d'adapter la vitesse du modem avec celle du correspondant comme le font les fax.

Norme de Bell	Vitesse (en bits par seconde)
Bell 103	300
Bell 212A	1.200

- **Correction des erreurs et compression des données** : L'utilisation ou non des mécanismes de correction/compression de donnée V42/V42bis ou MNP pour assurer la qualité de la transmission et réduire le nombre de bits transmis sur la ligne téléphonique. De façon assez étrange, tout le monde est d'accord sur les termes V42/V42bis et MNP.
- **Bits de donnée** : Le nombre de bits de la donnée (7 bits ou 8 bits) dans les mots échangés.
- **Bit de parité** : La parité de la donnée (aucune, impaire, paire) dans les mots échangés.
- **Bits de stop** : Le nombre de bits de stop marquant la fin d'une donnée (1 bit ou 2 bits) dans les mots échangés.
- **Emulation du terminal** : Le jeu de commandes reconnu par l'ordinateur de l'utilisateur pour émuler le terminal. Ce jeu de commandes peut gérer bien plus que les affichages à l'écran. Par exemple, au début du stage, un projet de Devlonics portait sur l'extension du Prestel pour qu'il puisse gérer l'imprimante de l'utilisateur. Nombreux sont les termes techniques regroupés par assimilations dans le domaine de l'émulation.

Termes regroupés:
Emulation, Profil, émulation du terminal, terminal, standard à émuler, terminal de départ, terminal type
Ansi, Ansi X3.64, Ansi-BBS, ANS
Bildschirmtext, Profil 1 (les profils sont definis par la CEPT : Conférence Européenne des Postes et Télécommunications), Vidéotex
Bistel, extension au Profil 3, Vidéotex
Prestel, Profil 3, Vidéotex
Télétel, Profil 2, mode Vidéotex 40 colonnes Télétel 40 colonnes, Vidéotex, VTX, Minitel 40, Minitel 80, Minitel, mode Vidéotex, STUM 1B (Spécification Techniques d'Utilisation du Minitel), STUM 2, STUM 12, langage Protocole
TTY, Télétype, ASCII
VT102, normalisation VT102, ISO 6420, terminal DEC, mode terminal 80 colonnes, mode téléinformatique, Télétel 80 colonnes

Dans ce travail, nous retenons les termes suivants utilisés chez Devlonics : Emulation, Ansi, Bildschirmtext, Bistel, Prestel, Télétel, TTY, VT102.

•**Caractères** : Le jeu des caractères affichés. Comme pour l'émulation, des termes techniques sont regroupés par assimilation dans ce domaine:

Termes regroupés:
Jeu de caractères, NRC (?), caractères
Ansi, IBM ou compatible
Ascii américain, Ascii, US-ASCII
Ascii français, F, Ascii français étendu, French
Macintosh
Vidéotex allemand, German
Vidéotex anglais, English
Vidéotex français, French
Vidéotex belge, Belgian

Dans ce travail, nous retenons les termes suivants utilisés chez Devlonics : Ansi, Ascii américain, Ascii français, Macintosh, Vidéotex allemand, Vidéotex anglais, Vidéotex français, Vidéotex belge.

Si l'émulation du terminal est:	Alors les caractères disponibles sont:
Ansi	Ansi, Ascii américain, Ascii français, Macintosh
Bildschirmtext	Vidéotex allemand
Bistel	Vidéotex belge
Prestel	Vidéotex anglais, Vidéotex belge
Télétel	Vidéotex français
TTY	Ansi, Ascii américain, Ascii français, Macintosh
VT102	Ascii américain, Ascii français

- **Echo local** : Le serveur renvoit ou non l'écho des caractères reçus à l'utilisateur ou si on préfère respectivement, les caractères tapés par l'utilisateur ne sont pas ou sont reproduits à l'écran en plus d'être envoyés vers le serveur.
- **«Saut de ligne» lors d'un «retour chariot»** : Le serveur s'attend ou non à ce que un saut de ligne soit automatiquement reproduit à l'écran de l'utilisateur à l'émission d'un retour chariot. Il se fait que pour le réseau belge DCS le serveur et l'utilisateur utilisent cette option de façon assymétrique.
- **Identification** : Certains serveurs Bistel, Prestel et Télétel permettent une identifiaction automatique des utilisateurs qui s'y connectent. L'identification consiste en un code d'au plus 20 caractères que l'utilisateur a préalablement encodé et que le terminal envoie automatiquement au centre serveur, à la demande de celui-ci.

A ces réglages viendront s'en ajouter d'autres propres à l'équipement de l'utilisateur:

- **Modem** : Le jeu de commandes reconnu par son modem. Souvent il s'agira du jeu de commandes Hayes, parfois du V25bis.
- **Port** : La porte série de l'ordinateur à laquelle le modem est raccordé, pour permettre au logiciel d'adresser le modem.
- **Numérotation** : La numérotation se fait par tonalités (téléphone à touches) ou par impulsions (téléphone à cadran). Les impulsions sont employées par les anciens centraux téléphoniques, ceux qui sont électro-mécaniques. Les nouveaux centraux électroniques reconnaissent les deux types de numérotations.
- **Signal sur la ligne** : Le modem émet des sons sur la ligne téléphonique. L'utilisateur souhaite choisir d'entendre ou de ne pas entendre la communication sur la ligne téléphonique entre les deux modems.
- **Touches du clavier** : L'utilisation du clavier. Il existe de nombreux claviers différents de par le nombre, la disposition et les fonctionnalités de leurs touches. L'utilisateur doit pouvoir gérer au mieux son clavier pour produire les séquences de caractères attendues par le serveur.
- **Format de l'écran** : L'utilisation de l'écran. L'utilisateur doit pouvoir gérer au mieux la présentation de l'écran du terminal qui lui est montré.

Nous disposons d'une part des modules de l'application et, d'autre part des réglages à effectuer. Cela nous permet d'introduire les fonctions de réglages avec ces modules.

Module: Lien de l'ordinateur au modem

Description:
Ce lien concerne tout ce qui est nécessaire pour connecter l'ordinateur de l'utilisateur au modem.

Variables:
- Modem.
- Port.
- Bits de donnée.
- Bit de parité.
- Bits de stop.

Fonction:
- Régler le lien.

Module: Lien du modem au réseau téléphonique

Description:
Ce lien concerne tout ce qui est nécessaire pour relier le modem de l'utilisateur à la prise RTT du réseau téléphonique.

Variables:
- Numérotation.
- Signal sur la ligne.

Fonction:
- Régler le lien.

Module: Lien de modem à modem

Description:
Ce lien concerne le flux de données entre les deux modems des correspondants.

Variables:
- Vitesse du modem.
- La correction des erreurs et la compression de données.

Fonction:
- Régler le lien.

Module: Lien d'ordinateur à ordinateur

Description:
Ce lien concerne le flux de données d'ordinateur à ordinateur des deux correspondants.

Variables:
- Emulation du terminal.
- Caractères.
- «Saut de ligne» lors d'un «retour chariot».
- Echo local.
- Identification.

Fonctions:
- Régler le lien.
- Configurer les touches du clavier. (Cette opération devrait donner naissance au module *clavier*).
- Configurer le format de l'écran. (Cette opération devrait donner naissance au module *écran*).

Les fonctions réalisables avec ces quatre modules se résument aux réglages logiciels qui les concernent. Avant de continuer plus en avant nous avons voulu confronter cette classification avec le point de vue d'utilisateurs. La procédure adoptée fut la suivante : (1) Par un questionnaire, nous leur avons demandé de décrire un environnement de communication par modem (2) Sur base de cette description, pouvions-nous replacer les quatre liens dans l'environnement décrit pour montrer le rôle des réglages? (3) Cette procédure a permis d'invalider la classification proposée et d'en proposer une nouvelle sur base de la représentation intuive que les utilisateurs se font du système de télécommunication.

Description de l'environnement par l'utilisateur

Nous avons interviewé des membres de deux clubs d'informatique, l'un "Micho" à Tournai et l'autre "Micom-80" à Kortrijk, des vendeurs à l'Apple Education Center de Namur, un technicien responsable d'Eurocom chez Devlonics et un autre sysop (c'est-à-dire un responsable d'un serveur) au club Micho. Au gré de ces interviews, nous avons consolidé un questionnaire que nous reprenons ci-dessous. Indépendamment d'un logiciel de communication précis, les questions ont pour dessein de vérifier si le schéma du système de télécommunication par modem proposé plus haut au chapitre 2 (fig. 2.1) est bien reconnu par les utilisateurs comme représentatif. Ce questionnaire comporte cinq questions posées au fur et à meusure de l'entretien:

`1°) Que peut-on connecter d'utile sur une ligne téléphonique?`

Notre objectif est de connaître qu'y a-t-il d'utile qui soit rendu possible grâce à la ligne téléphonique? A quoi peut bien servir une ligne téléphonique? D'où la question: "Que peut-on connecter d'utile sur une ligne téléphonique?" Nombreuses sont les personnes interrogées, non averities en matière de télécommunication, qui disent : "On peut connecter ceci pour faire cela", alors que si on leur demande à quoi peut bien servir une ligne téléphonique, elles ne savent pas que répondre. Par exemple : "*Le fax pour envoyer des fax, le téléphone pour téléphoner.*"

`2°) Que faut-il pour pouvoir utiliser un modem?`

Notre objectif est de percevoir ce qu'il faut pour pouvoir utiliser un modem. Quels sont les différents ingrédients (si possible matériel) d'un environnement de télécommunication par modem? Les personnes interrogées savaient ce qu'était un modem.

`3°) Quel est le déroulement par étapes d'une communication par modem?`

Notre objectif est de percevoir ce qu'il se passe lors de l'emploi d'un modem. Quelles sont les différentes étapes d'une télécommunication?

Pour les personnes inhabituées au maniement des modems, un problème de réglages, de paramètres se dégage de leurs réponses à cette troisième question. Pour pouvoir communiquer il faut préciser beaucoup de choses, on ne sait pas toujours lesquelles et c'est complexe.

4°) Quels sont les outils qu'un logiciel de communication (par modem) se doit de proposer?

Nous souhaitons connaître quels sont les outils qu'un logiciel de télécommunication par modem se devrait d'offrir.

Ensuite nous dessinions sur une feuille vierge tous les éléments cités en réponse aux deux premières questions par la personne interrogée. Bien souvent il s'agissait d'un ordinateur, d'un modem, d'un téléphone, d'un logiciel, d'une ligne téléphonique. Deux d'entre ces éléments étaient déjà cités dans ces deux questions, à savoir : la ligne téléphonique et le modem.

Une fois ces divers éléments dessinés, nous proposions qu'elle (la personne interrogée) les relie entre eux selon ce qui lui semblait logique, cohérent. Toutes les personnes dessinaient à peu près le dessin suivant que nous leur présentions par après en guise de conclusion:

5°) Ce schéma correspond-t-il à ce que vous pensiez?

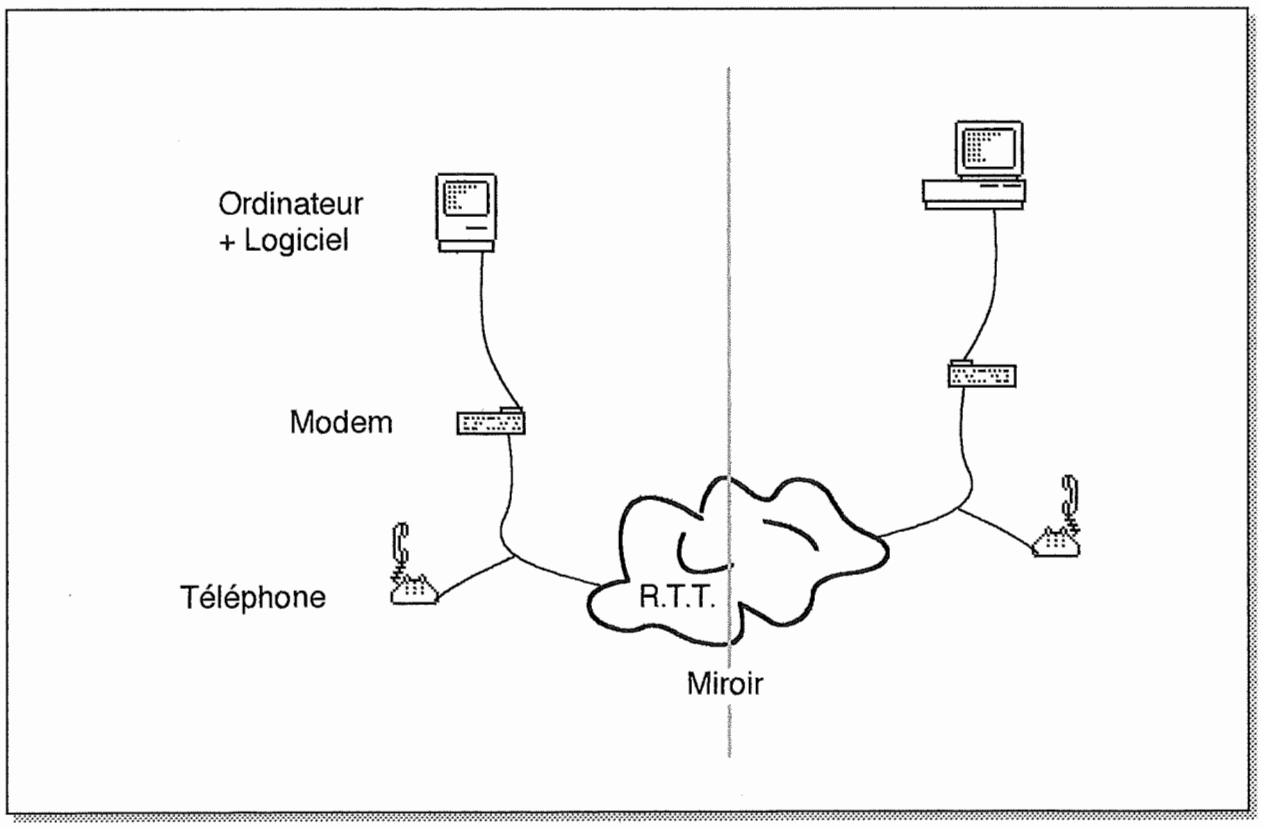


Fig. 5.6 : Le schéma final

Ce schéma nous semble si évident qu'il semble étrange d'être passé par un questionnaire pour le proposer. L'avantage, c'est que maintenant, nous osons affirmer que ce schéma est représentatif d'un système de communication par modem. Mais le plus passionnant dans ce schéma, c'est la façon dont l'utilisateur le construit et l'exploite pour montrer comment une communication par modem se déroule.

Montrer le rôle des réglages

Est-ce que les quatre modules que nous proposions, permettent aux utilisateurs de mieux comprendre le comment et le pourquoi des réglages? Nous avons d'abord laissé l'utilisateur expliquer à sa façon comment se déroulait une communication par modem, même s'il reconnaissait ne pas être très porté sur la chose. Ensuite, pour valider les quatre modules proposés, nous avons adopté deux comportements:

- Quand l'interlocuteur nous disait ne pas connaître grand chose en modem, nous lui expliquions comment cela se passait (il y a un flux de données entre le modem des correspondants et grâce à cela il y a aussi un flux de données entre leur ordinateur) et lui montrions où les réglages intervenaient. Il comprenait.

- Quand l'interlocuteur nous disait être habitué au jargon des modems, nous lui montrions notre proposition. Il l'appréciait.

Un doute reste, le schéma proposé n'est-il pas l'issue fatale des questions posées? Il ne faut pas se le cacher, notre objectif est de trouver si ce schéma est reconnu par le plus grand nombre d'utilisateurs interrogés et les questions sont dirigées dans ce sens.

L'invalidation de la classification

Si nous sommes passés par un questionnaire c'est parce qu'à l'épreuve au moyen d'un premier prototype testé avec des utilisateurs expérimentés, cette classification basée sur la compréhension de l'aspect technique du système de communication s'est avérée être imbuvable. ***Même si les utilisateurs reconnaissent bien le rôle des composants techniques du système, cela ne représente rien.*** Pour réaliser le prototype, nous avons utilisé le Quick C pour Windows et QuickCase:W (un outil permettant de mettre en forme rapidement de petits prototypes). Nous avions créé plusieurs menus dont un, intitulé "Réglage", qui nous intéressait.

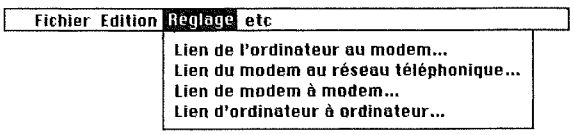


Fig. 5.7 : Premier prototype

Chaque item de ce menu (fig. 5.7) était relié à une boîte de dialogue indiquant textuellement ce que l'utilisateur pouvait faire. Le test consistait à demander à la personne interrogée de "faire" tous les réglages pour se connecter à un serveur qu'elle connaît. Ce fut un échec et petit à petit, par affinements successifs à l'épreuve d'autres prototypes, nous avons consolidé un autre modèle composé d'autres objets, significatifs pour les personnes interrogées. Les quatre modules de l'architecture de Polycom ne pouvaient pas être repris tel quel pour servir d'objets de l'application de base au langage de l'interface. Les réactions des utilisateurs nous ont souvent surpris. Les choix de dénomination des objets ont eu des impacts importants.

Ces objets sont intéressants pour une découpe en module de Polycom ou dans une démarche de formation mais l'utilisateur ne semble pas concevoir le système sous cet angle. Pour montrer comment une communication par modem se déroule, les personnes interrogées présentent naturellement le système comme un cheminement d'informations depuis leur ordinateur vers le centre serveur et vice versa (5.8), plutôt que comme des liens qui s'établissent entre les composants clés de leur système et ceux du centre serveur (fig 5.9). ***Les utilisateurs développent une vision opératoire du système, c'est-à-dire sous la forme d'un assemblage d'opérations. Tandis que nous, nous avons développé une vision fonctionnelle, abstraite, décrivant les fonctionnalités que le système doit remplir.*** Remarquons que cette dernière approche leur permet de mieux comprendre les mécanismes mis en œuvre par le système de télécommunication par modem.

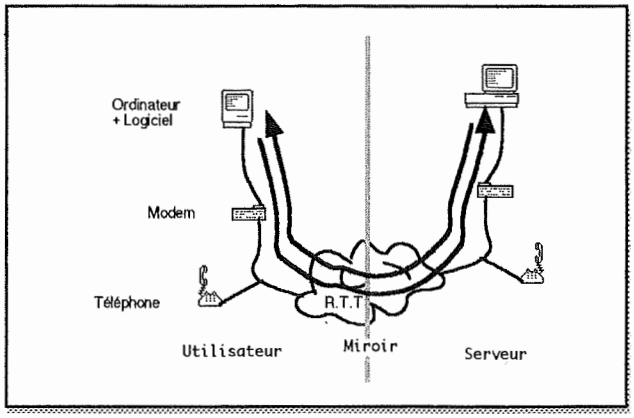


Fig 5.8 : La vision opératoire. Le système vu intuitivement sous l'angle d'un cheminement d'informations.

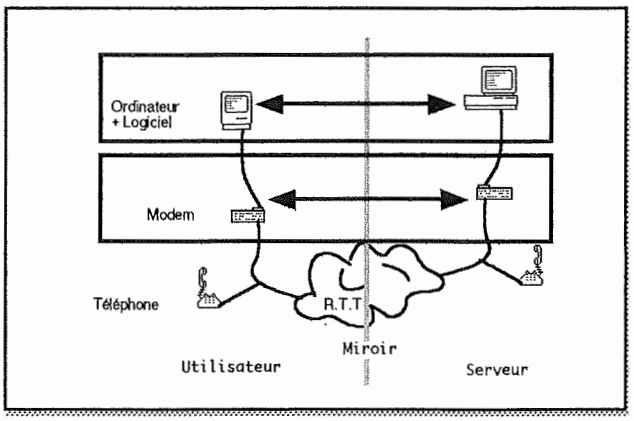


Fig 5.9 : La vision fonctionnelle. Le système vu pédagogiquement sous l'angle de liens entre les composants.

Pour distinguer la vision fonctionnelle (souvent de l'informaticien) de la vision opératoire (souvent de l'utilisateur) prenons l'exemple d'un organe : le cœur. Du point de vue fonctionnel, un cœur permet de faire circuler le sang. Du point de vue opératoire, il y a du sang qui arrive d'un côté et qui ressort de l'autre. Avant d'assimiler des fonctionnalités de l'application, l'utilisateur développera un comportement opératoire sur ce qu'il perçoit de

l'application. Pour s'en rendre compte, observons notre premier comportement aux commandes d'une nouvelle application. Nous nous posons : "Que se passe-t-il si je clique par-ci ou par-là?".

5.2. OBJETS DE L'APPLICATION

Le questionnaire nous a amené à proposer des objets de l'application qui devront être significatifs pour l'utilisateur dans le cadre des tâches abordées. Il s'agit d'une deuxième proposition, a posteriori cette fois-ci. ***Cette proposition tient compte de la vision opératoire développée intuitivement par les utilisateurs interrogés***. Ces objets de l'application seront regroupés pour former un modèle pragmatique de l'application. Ce modèle est élaboré au fil du temps en collaboration étroite avec des utilisateurs. Il exprime la sémantique des composantes du système. Nous avons élaboré une démarche par ajustements progressifs. A partir d'un point de départ que nous fournissons, l'état d'avancement du développement sera rythmé par un va et vient entre les interviews des utilisateurs et leur prise en compte. Nous avons essentiellement eu recours à deux types d'interviews. L'un consiste en un test de prototypes. L'autre est basé sur l'existence d'indices de rappels.

Test de prototypes

Prenons l'exemple de ce qui s'est passé au cours d'un test de prototypes. Nous avons observé que la dénomination des items de menu a un impact sur le comportement de l'utilisateur. Ainsi, alors que le menu [Réglage] reprenait les items [Terminal, Ligne, Modem, Téléchargement, Equipement] (fig. 5.10), nous avons demandé à un utilisateur averti[3] de régler la vitesse de transmission. Il a sélectionné dans l'ordre [Modem], [Terminal], [Modem], [Modem]. A ce stade, nous lui avons demandé s'il avait bien sélectionné tous les items du menu [Réglage]. Il a dit que oui (!), en regardant le menu en question, avant d'affirmer qu'il n'y avait pas moyen de régler la vitesse de transmission. En fait, il fallait sélectionner l'item [Ligne]. Et si nous avions choisi de mauvais indices de rappel? Le menu a progressivement évolué vers [Terminal, Raccordement, Installation] (fig. 5.11).

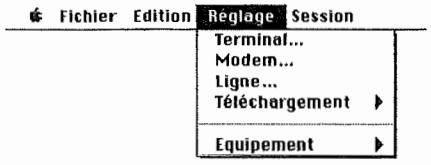


Fig. 5.10 : Une version abandonnée du menu «réglage»

Fig. 5.11 : La version conservée du menu «réglage»

[3] Un vendeur.

Les indices de rappels

La technique des indices de rappels est définie par A. Lieury [Lie80 p119] : "Un ***indice de rappel*** est un groupe d'informations de nature variée qui permet de récupérer de la mémoire [naturelle], un ensemble structuré déjà stocké (mots, idées, souvenirs)". C'est par exemple un cliché qui rappelle toute une série de souvenirs, mais cela peut être aussi une association sémantique (comme : caméra -> film, filmer, visionner, bobine, cassette), une odeur, une musique, etc...

Pour chaque tâche étudiée, nous procédons comme suit pour mettre en évidence des indices de rappels (l'exemple concerne la tâche de réglage):

- 1°) Nous consolidons (petit à petit) une liste de termes destinés à être reconnus par les utilisateurs ciblés. Nous nous attendons à ce que chaque terme dénomme un objet de l'application, sur lequel l'utilisateur associe naturellement des opérations qu'il peut effectuer. Le nom de l'objet sera l'indice de rappels des opérations permises.

  Pour la tâche de réglage, la liste de départ fut : [Terminal, Modem, Ligne]. Elle est devenue : [Terminal, Installation, Raccordement, Correspondant], l'objet *Correspondant* étant rajouté au modèle.

- 2°) En construisant cette liste, nous interrogeons des personnes ayant déjà utilisé un système de communication entre ordinateurs, qu'il soit tel que nous l'avons décrit ou non. Nous leur demandons à quoi elles associent les termes de la liste, en leur donnant un contexte pour qu'elles puissent situer la question.

  Exemple, avec comme contexte qu'il s'agit de communication entre ordinateurs:
  Terminal: Pour avoir les données de l'ordinateur central à plusieurs points. Tout le monde peut travailler avec l'ordinateur central.
  Raccordement: Raccorder ensemble les différents terminaux sur l'ordinateur.
  Installation: C'est tout l'ensemble. Installer l'ordinateur et ses terminaux.
  Correspondant: Tout le monde peut correspondre ensemble. Correspondre avec d'autres firmes, d'autres pays.

  Autre exemple, même contexte:
  Terminal: Station de travail, service, serveur, pas complètement indépendant du serveur.
  Raccordement: Circulation, protection des données, inter-opérabilité.
  Installation: Ce qui permet le raccordement.
  Correspondant: Piratage, échange avec un autre.

- 3°) A ce stade, nous avons fait le bilan pour voir si il y avait concordance entre les diverses propositions. Pour l'exemple :

  Le terminal échange des données avec un serveur.
  Le raccordement permet de faire circuler des données entre le terminal et le serveur.
  L'installation est l'équipement dans son ensemble.
  Le correspondant est l'interlocuteur à qui l'on s'adresse.

•4°) Nous demandions ensuite, aux utilisateurs concernés, de faire correspondre chaque opération à effectuer avec un terme de la liste et, au besoin d'adapter la liste ou les opérations. Pour la tâche de réglage, les utilisateurs étaient avertis.

•5°) Nous en avons dégagé les objets de l'application, à savoir pour la tâche de réglage: *Terminal, Installation, Raccordement et Correspondant.* Ces objets s'assemblent pour constituer le modèle pragmatique de l'interface de la figure 5.12. Ce modèle reflète l'état du système pendant une communication. L'association ternaire «raccordement» est décomposable en deux associations binaires terminal-installation et terminal-correspondant. Nous reprenons néanmoins l'association ternaire qui nous permet de montrer que pour qu'il y ait un raccordement à un moment donné, les trois entités terminal, installation et correspondant doivent être associées :

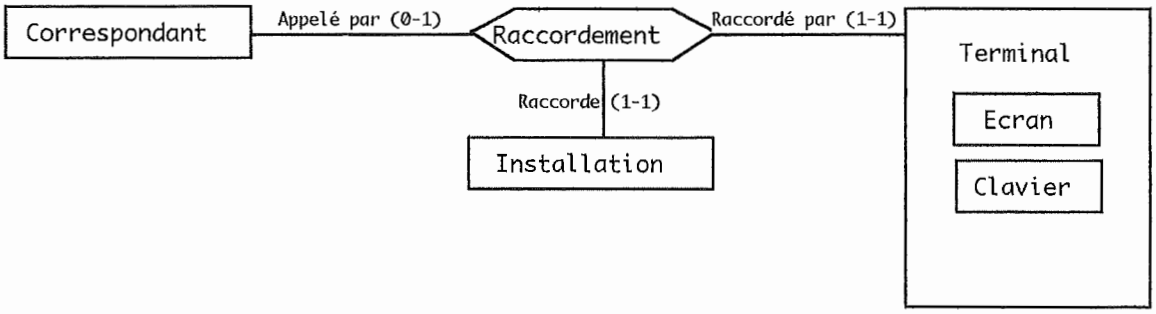


Fig. 5.12 : Pendant une télécommunication, du point de vue de l'utilisateur, le terminal est raccordé par une installation à un correspondant dans le cadre d'un raccordement. Un terminal comporte un écran et un clavier.

5.3. PREMIERE TACHE: REGLER SON SYSTEME POUR COMMUNIQUER AVEC UN CENTRE SERVEUR

Le modèle pragmatique de l'application obtenu pour cette tâche a été schématisé ci-dessus. Les objets de l'application destinés, pour cette tâche, aux utilisateurs avertis sont repris ci-après.

Objet de l'application: Terminal

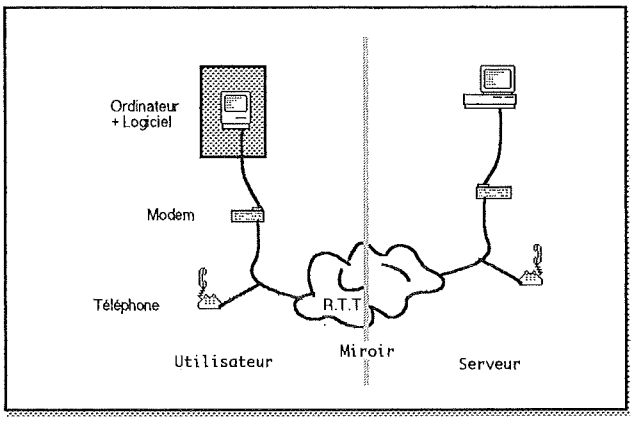


Fig. 5.13 : Terminal

Description:

Le terminal échange des données avec un correspondant. On parle de terminal Télétel, de terminal Bistel, de terminal VT102,...

Attributs:

- Emulation du terminal.
- Caractères.
- «Saut de ligne» lors d'un «retour chariot».
- Echo local.
- Identification.

Opérations:

- Régler le terminal, c'est-à-dire : ajuster les attributs du terminal.
- Configurer les touches du clavier. (Cette opération devrait donner naissance à l'objet *clavier* inclus dans l'objet *terminal*).
- Configurer le format de l'écran. (Cette opération devrait donner naissance à l'objet *écran* inclus dans l'objet *terminal*).

Objet de l'application: Installation

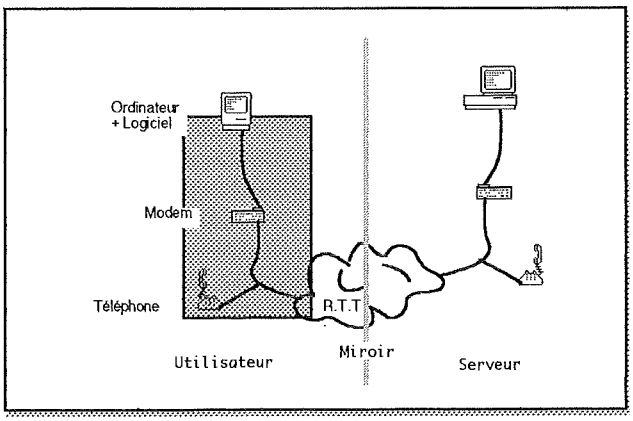


Fig. 5.14 : Installation

Description:

L'installation est l'équipement dans son ensemble, à savoir principalement, l'ordinateur, le raccord à la ligne téléphonique et le modem de l'utilisateur.

Attributs:

- Modem.
- Port.

·Numérotation.
·Signal sur la ligne.

Opérations:
·Régler l'installation, c'est-à-dire : ajuster les attributs de l'installation.
·Tester l'installation.

Objet de l'application: Raccordement

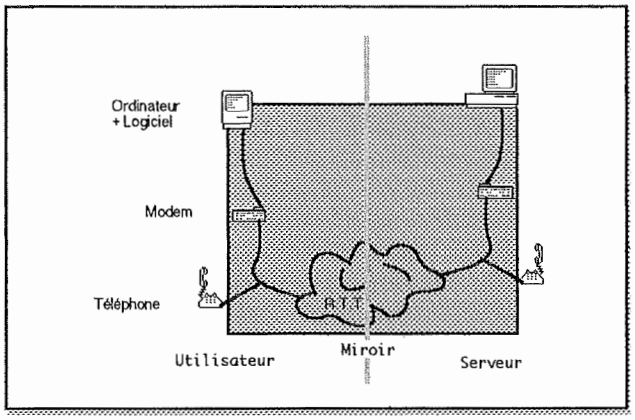


Fig. 5.15 : Raccordement

Description:
Le raccordement permet de faire circuler des données entre le terminal et le correspondant. L'utilisation d'un modem n'est qu'un moyen comme un autre de raccorder deux ordinateurs entre eux. On n'est pas contraint de passer par un modem pour communiquer avec un serveur. Il est par exemple possible de connecter les deux ordinateurs entre eux au moyen d'un câble spécial dit "null modem".

Attributs:
·Vitesse du modem.
·Correction des erreurs et compression des données.
·Bits de donnée.
·Bit de parité.
·Bits de stop.

Opération:
·Régler le raccordement, c'est-à-dire : ajuster les attributs du raccordement.

Objet de l'application: Correspondant (Serveur)

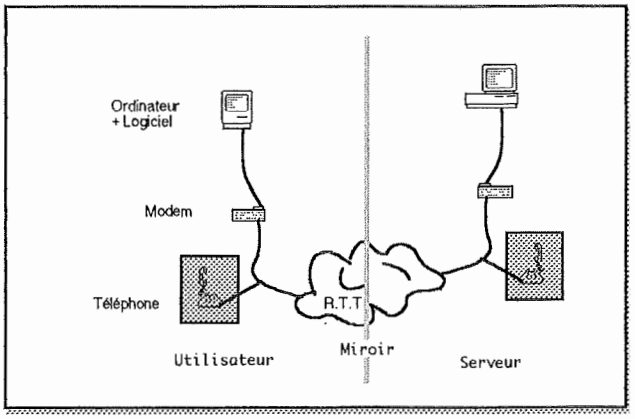


Fig. 5.16 : Correspondant

```
Description:
        Le correspondant est l'interlocuteur à qui on s'adresse. Cette notion met
        en évidence qu'utiliser le réseau téléphonique signifie communiquer avec un
        correspondant. Nous allons mettre en avant cette notion dans notre modèle
        lorsque nous nous attaquerons à la tâche "Appeler le centre serveur" parce
        que lors des tests, nous avons observé que les utilisateurs avaient la
        nette tendance à associer "Correspondant" avec "Appeler" plutôt que
        "Correspondant" avec "Réglage".

Attributs:
        ·Nom.
        ·Remarques le concernant.

Opération:
        ·Indiquer le correspondant, c'est-à-dire : garnir les attributs du
                correspondant.
```

5.4. DEUXIEME TACHE: APPELER LE CENTRE SERVEUR

Alors que la première tâche était exclusivement réservée à des utilisateurs avertis, celle-ci est destinée aux utilisateurs avertis ou non. Pour celui qui n'est pas passionné, cette tâche concerne un ensemble relativement stable de serveur, contrairement au passionné qui appellera un maximum de serveurs. Après avoir correctement réglé (fait régler) son système, l'utilisateur appelle un centre serveur de façon similaire à un appel téléphonique entre personne. Il s'agit de composer un numéro de téléphone et quand le correspondant (serveur) répond, Polycom gère la communication. Une personne qui téléphonne tient compte des signaux qu'elle entend sur la ligne pour gérer la communication. Le modem aussi (par exemple : la tonalité d'invitation à composer le numéro du correspondant, le signal "occupé"). Le modem renseigne ces signaux à Polycom. En voici quelque uns pour un modem donné:

- •Composition du numéro de téléphone.
  - •Le modem a "décroché" la ligne, reçu une tonalité d'invitation à l'appel et, composé le numéro de téléphone.
  - •Le modem a "décroché" la ligne et n'a pas reçu de tonalité d'invitation à l'appel.
- •Cela sonne de l'autre côté de la ligne.
  - •Cela sonne occupé, la ligne est occupée.
  - •Cela ne répond pas.
- •La communication avec le modem du centre serveur est établie.
- •La liaison téléphonique est interrompue.
- •On appelle sur la ligne téléphonique de l'utilisateur.

Les objets *Serveur* et *Annuaire* sont ajoutés à la structure du modèle (fig. 5.17). L'annuaire regroupe des correspondants et un serveur est le correspondant avec lequel l'utilisateur désire entrer en contact. Cet annuaire téléphonique est plus utile à l'utilisateur non passionné qui, comme on l'a observé, communique avec quelques mêmes serveurs:

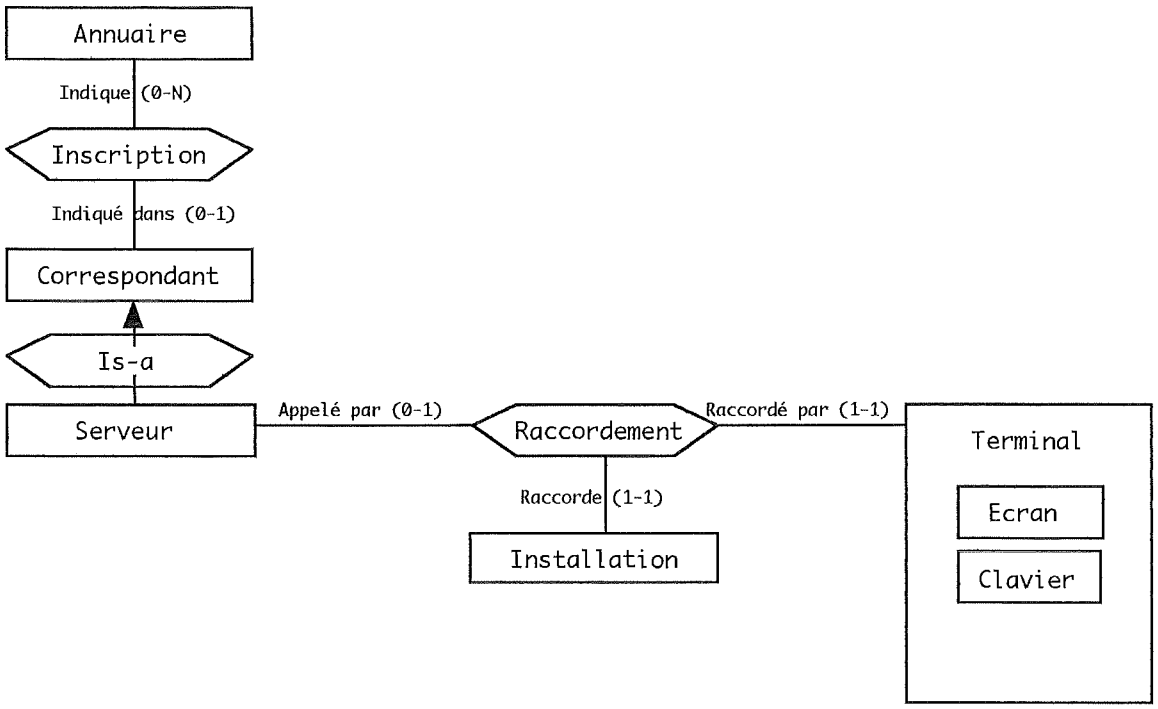


Fig. 5.17 : Un correspondant est indiqué dans un annuaire dans le cadre d'une incription. Le serveur est un correspondant.

Objet de l'application: Correspondant

Description:
Le correspondant est l'interlocuteur à qui on s'adresse.

Attributs:
- Nom.
- Remarques le concernant.
- Numéro de téléphone.

Opération:
- Indiquer le correspondant.

Objet de l'application: Serveur

Description:
Le serveur est le destinataire avec lequel l'utilisateur souhaite établir une communication.

Opérations:
- Appeler, càd se connecter à un serveur.
- Raccrocher, càd se déconnecter du serveur.
- Attendre un appel.

Objet de l'application: Annuaire

Description:
L'annuaire regroupe des correspondants.

Opérations:
- Rechercher un correspondant.
- Mettre l'annuaire à jour.
- Ouvrir l'annuaire à une lettre de l'alphabet.

5.5. TROISIEME TACHE: ARCHIVER DES ECRANS REÇUS

Cette tâche doit pouvoir être accomplie par tous les utilisateurs, avertis ou non. Il s'agit de sauvegarder une copie électronique d'écran(s) reçu(s) pour la manipuler à un autre moment ou, la passer à une autre application. L'analyse de cette tâche est basée sur l'observation et les commentaires d'utilisateurs aux commandes de logiciels existants, en tenant compte de l'objectif de Devlonics quant à envisager la composition d'un petit serveur télématique Prestel:

- 1°) L'utilisateur conserve une copie électronique tel quel de l'écran du terminal. L'utilisateur réunit plusieurs copies d'écrans selon un ordre qui lui est propre, dans deux buts. L'un est de disposer d'une collection des (plus beaux) écrans vus. Cette activité est fréquente pour tous les utilisateurs. L'autre est de classer des écrans sous forme de documents pour par exemple constituer et éditer un (petit) serveur (Prestel) ou encore pour disposer de l'évolution d'une certaine information au fil du temps comme les cours de la bourse. Cette activité concerne les utilisateurs qui ont besoin de manipuler hors communication les écrans issus d'un serveur.
- 2°) L'utilisateur conserve une copie électronique telle quelle d'une séquence dynamique d'écrans reçus pendant le déroulement d'une communication, tel un film dans deux buts. L'un, peu fréquent, est de revoir ou présenter les plus belles séquences reçues. L'autre, suggéré, est de revoir tout le déroulement d'une communication quand celle-ci est terminée, dans le cas où le coût du temps d'accès à un serveur est élevé.
- 3°) L'utilisateur recherche une information contenue dans un des derniers écrans reçus en provenance du serveur, plutôt que de revoir ces écrans tels quels. Cette tâche est très fréquente dès que l'utilisateur navigue dans un serveur ou, veut échanger des fichiers dont le nom se trouvait dans un écran transmis précédemment par le serveur. Sur ce dernier point, nous constatons une habitude de plusieurs serveurs. Soit ils présentent la liste des fichiers qu'ils mettent à disposition, soit ils présentent un menu de commandes pour le téléchargement. Remarquons que l'utilisateur cherche ici plus à retrouver du texte que de s'attacher à l'esthétique d'un écran, contrairement aux deux premiers points.
- 4°) L'utilisateur effectue une opération de copier-coller entre applications. Deux copier-coller fondamentalement différents sont possibles. L'un constiste à mettre dans le presse-papier une recopie électronique de l'écran tel quel. L'autre consiste à placer dans le presse-papier une recopie électronique du contenu textuel de l'écran.

Les deux premières approches qui sont implémentées sous forme d'archivage électronique, sont abordées avec des métaphores. Pour le point (1°), une photographie s'apparente au contenu figé d'un écran, tandis que pour le point (2°) le film vidéo fait allusion à une séquence dynamique d'écrans. Cela nous laisse introduire ***les métaphores suivantes*** : l'utilisateur prend une photographie instantanée de l'écran du terminal au moyen d'un appareil photo (Polaroid) et insère la photo dans un album. Il enregistre un film de l'écran du terminal au moyen d'un magnétoscope sur une bande vidéo et le projette sur un écran de projection. Cependant, nous n'allons pas nous préoccuper plus en avant de la métaphore du film vidéo tant que l'étude de la métaphore de la photographie ne sera

pas dans un état d'avancement suffisant. Ce choix reflète l'objectif de Devlonics qui concerne la composition d'un petit serveur Prestel.

La troisième approche donne naissance à l'objet *Tampon* à défaut d'un objet plus commun. Quand à la dernière approche, il s'agit du presse papier dont le comportement est à déterminer pour prendre en compte le besoin d'échanger du texte ou de l'image.

Les objets *Appareil photo, Photographie et Album photos* sont ajoutés à la structure du modèle (fig. 5.18, page suivante). A nouveau, nous introduisons une association ternaire qui est décomposable en deux associations binaires Photographie-Terminal et Photographie-Appareil photo. Ce choix montre qu'une prise de vue permet à l'appareil photo de reproduire, de dupliquer l'écran sur la photographie:

Objet de l'application: Album photos

Description:
L'album rassemble des photos.

Attribut:
·Nom.

Opérations:
·Entamer un album vierge.
·Ranger un album sur une étagère.
·Aller chercher un album qui est rangé.
·Mettre un album à la poubelle.
·Parcourir l'album, de photo en photo.

Objet de l'application: Photographie

Description:
La photo est un cliché instantané de l'image de l'écran du terminal.

Attributs:
·La photo en elle-même.
·Référence de la photo.

Opérations:
·Ajouter une photo dans l'album.
·Revoir une photo de l'album.
·Supprimer une photo de l'album.

Objet de l'application: Appareil photo

Description:
L'appareil photo permet de prendre des photos.

Opérations:
·Photographier.

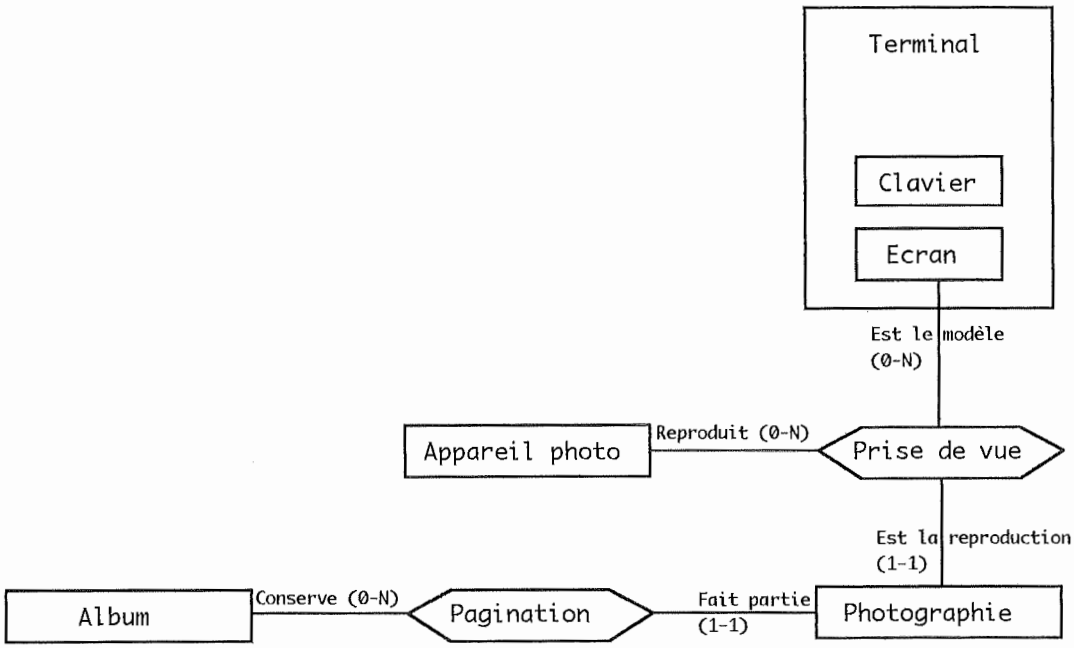


Fig. 5.18 : A un moment douné l'utilisateur peut photographier l'écran du terminal. L'écran du terminal est le modèle que l'appareil photo reproduit sous la forme d'une photographie instantanée au sein d'une prise de vue. Une photo fait partie d'un album photos au gré d'une pagination.

Nous ajoutons l'objet *Tampon* à la structure du modèle (fig. 5.19), à défaut d'un objet plus représentatif:

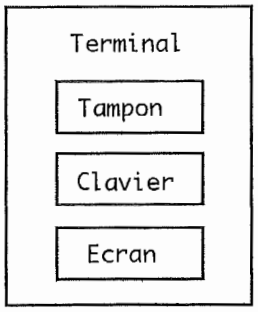


Fig. 5.19 : Le tampon fait partie du terminal

Objet de l'application: Tampon

Description:
Le tampon mémorise les caractères textuels affichés à l'écran au fil du temps. La taille de sa mémoire sera limitée à un certain nombre de caractères. En cours de communication, quand l'utilisateur demande de voir le tampon, celui-ci continue son activité de mémorisation.

Attributs:
·Taille.
·Contenu.

Opération:
·Parcourir le tampon.

Plus haut nous avons dit que le choix de l'album photos s'intègre dans l'objectif de Devlonics qui concerne la composition d'un (petit) serveur Prestel. En effet, nous avons montré pendant le stage que l'utilisateur peut organiser les photos au sein d'un album de façon à reproduire naturellement la structure du serveur Prestel. Un tel serveur se compose de pages qui sont directement accessibles au moyen d'un numéro qui les identifie. Chaque page se compose d'au plus 26 frames numérotés de A à Z. Ces frames sont accessibles dans l'ordre séquentiel. L'utilisateur naviguera à l'intérieur de cette structure au moyen des possibilités suivantes (fig. 5.20) : soit il tape *# ou *09 ou # et, il passe respectivement au frame précédent, courant ou suivant d'une page; soit il indique un numéro de page entre les symboles * et #.

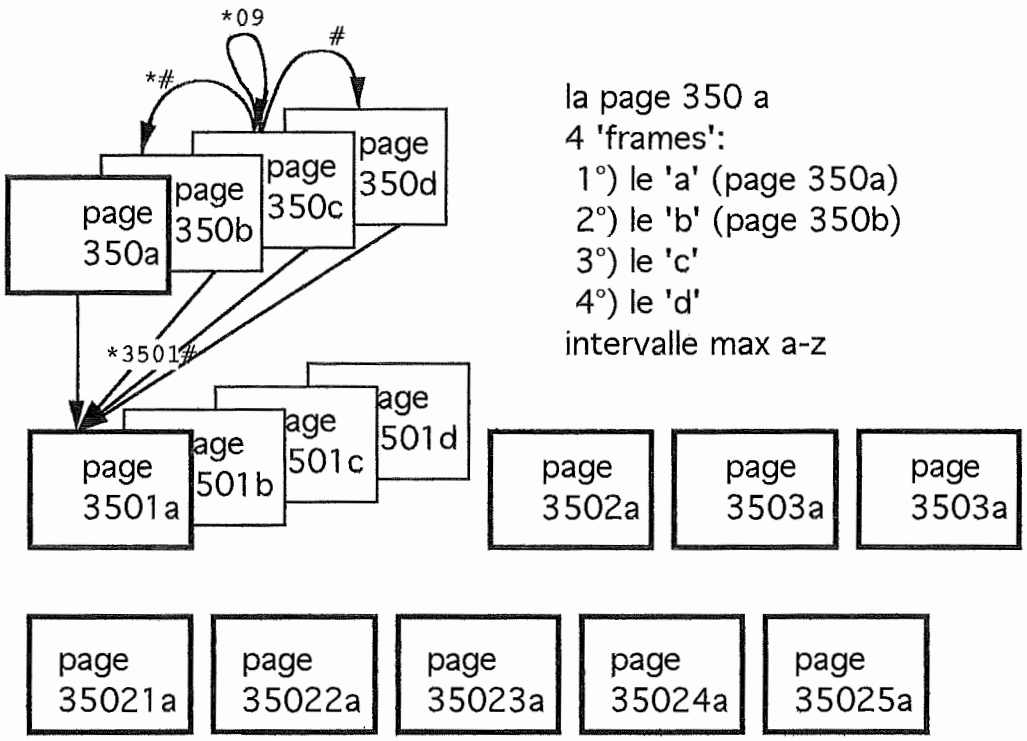


Fig. 5.20 : Schéma d'un extrait significatif d'une structure d'un serveur Prestel

L'organisation d'un tel serveur qui est vue comme une structure arborescente, pose plusieurs problèmes : (1) L'utilisateur qui interroge un tel serveur n'en connait pas a priori la structure; (2) Le responsable du serveur oublie lui-même les méandres de cette structure au fil du temps; (3) Plusieurs personnes sont amenées à mettre à jour des pages du serveur. L'album photos permettra de manipuler ces pages en les regroupant par niveaux de plus en plus fin pour former une structure arborescente telle que le plan d'un livre. A ce jour, diverses réalisations interactives de classements par niveaux existent. Prenons l'exemple du mode plan de Word ou, de la représentation par nom des fichiers dans une fenêtre du Système 7 d'Apple.

5.6. QUATRIEME TACHE: ECHANGER DES FICHIERS

Il s'agit d'envoyer un fichier de l'ordinateur de l'utilisateur vers le centre serveur ou vice-versa. Avant de continuer, nous définissons un objet technique clé qui permet de décoder nos propos:

- **Protocole de transfert** : Méthode pour transférer un fichier entre ordinateurs. Les protocoles les plus répandus dans les clubs sont souvent développés par des amateurs.

Voici une séquence typique d'opérations pour effectuer cette tâche:

- Choisir un fichier à échanger.
- Préciser la manière avec laquelle le fichier sera échangé, c'est-à-dire préciser un protocole de transfert de fichier.
- Donner l'ordre de recevoir le fichier en provenance du serveur ou de le lui envoyer.

Pour traiter le mécanisme de l'échange des fichiers, nous envisageons la métaphore du courrier. On peut envoyer ou recevoir une lettre dans une enveloppe par la poste. Le protocole de transfert de fichier est une enveloppe et la lettre est un fichier. Cependant, cette métaphore doit être très rapidement délaissée au profit des termes techniques pour deux raisons essentielles. La première est que ces termes sont repris par les serveurs dans les dialogues concernant le transfert de fichier. La seconde est que ces termes font aussi partie du dialogue mis en œuvre par de nombreux protocoles de transfert de fichier. Mais alors, il est fait appel à des acquis que l'utilisateur non averti ne possède pas. On constate qu'au fil du temps, des termes techniques qui portent sur le transfert de fichier, sont regroupés par assimilation par les utilisateurs avertis:

Termes regroupés:
Transfert, téléchargement, file transfer
Protocole de transfert, protocol name, protocol(e), type de transfert, file protocol
Kermit, Kermit 7-bit, Kermit 8-bit, Kermit image, SuperKermit
Xmodem, Xmodem Checksum, Xmodem CRC, Xmodem/1K, Xmodem/1K-G, Xmodem-R, Xmodem CRC-16, Christensen protocol, Xmodem batch, Xmodem CRC with fallback, XModem CRC with checksum fallback, Modem7, CPMUG (CP/M User Group)
Ymodem, Ymodem Batch, Ymodem-G, Ymodem-G Batch, Ymodem-B, YMDM-G
Zmodem
CIS A (Compuserve Information System), CIS B, CIS Quick B, Compuserve A, Compuserve B
Bi-Modem, B-Modem
Recevoir un fichier, download, dowload file, autodownload, receive file, logoff after download, Pg dn, réception fichier
Envoyer un fichier, upload, upload file, send file, Pg up, envoi fichier

Un remède palliatif à la multiplication des prérequis nécessaires pour exécuter cette tâche de transfert consiste à laisser à disposition de l'utilisateur des protocoles de transfert de fichier prévus pour limiter tout dialogue, au lieu de lui expliquer en long et en large comment choisir un protocole en fonction des conditions de transfert sur la ligne téléphonique. Cela a donné naissance à des protocoles de transfert propres à certains logiciels à usage (très) limité. En effet, l'utilisateur *et* le serveur doivent faire usage d'un même protocole. Un autre remède, plus suivi, consiste à placer une expertise dans le protocole de transfert de fichier afin, qu'en outre il adapte le transfert aux conditions de la ligne et à la gamme de protocoles connus des deux participants à l'échange. Les protocoles de transfert de fichier les plus récents s'y appliquent en reprenant des idées souvent anciennes. A nouveau, on se retrouve dans l'épaisseur historique des développements de moins en moins amateuristes et simplistes, de plus en plus professionnels et complexes [For87b].

L'objet *Fichier* est ajouté à la structure du modèle (fig. 5.21). A nouveau, nous faisons appel à une association ternaire «téléchargement» décomposable en deux associations binaires Fichier-Terminal et Fichier-Serveur. Ce choix montre qu'au moment du téléchargement un fichier est transféré du terminal vers le serveur ou du serveur vers le terminal:

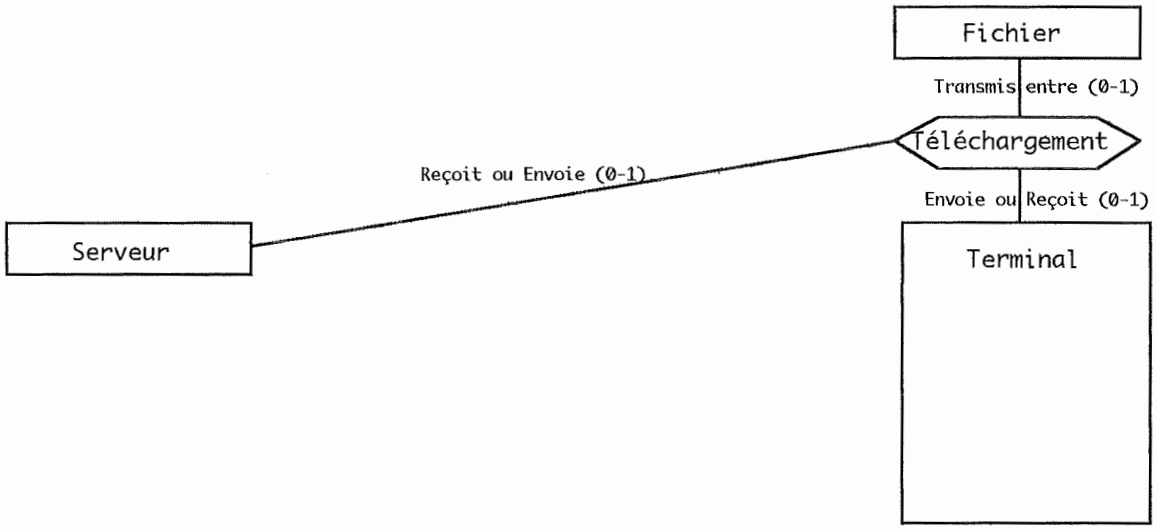


Fig. 5.21 : Au moment du téléchargement, le fichier est transmis entre le terminal et le serveur.

Objet l'application: Fichier

Description:
Le fichier est pris selon le sens présenté par le système d'exploitation.

Attributs:
·Nom.
·Taille.
·Durée de transmission.
·Date de la dernière mise à jour.
·Protocole de transfert.
·Contenu.

Opérations:
·Sélectionner un fichier.
·Envoyer un fichier vers le serveur.
·Sélectionner un protocole de transfert.
·Recevoir un fichier du serveur.

5.7. CONSOLIDATION DE LA STRUCTURE DU MODELE

A la figure 5.22 (page suivante), nous reprenons la structure consolidée du modèle. Pendant une télécommunication, du point de vue de l'utilisateur, le terminal est raccordé par une installation à un serveur dans le cadre d'un raccordement. Le serveur est un correspondant. Un correspondant est indiqué dans un annuaire dans le cadre d'une inscription. Un terminal comporte un écran, un clavier et un tampon. A un moment donné, un fichier est transmis entre le terminal et le serveur. A un moment donné, l'utilisateur peut photographier l'écran du terminal. L'écran du terminal est le modèle que l'appareil photo reproduit sous la forme d'une photographie instantanée au sein d'une prise de vue. Une photo fait partie d'un album photos au gré d'une pagination.

Un autre modèle est-il préférable?

Et si on avait parlé de Rollodex® ou d'annuaire électronique plutôt que d'annuaire papier? Chaque modèle reflète l'optique dans laquelle il a été conçu. Pour le concepteur, le modèle retenu permet d'étendre plus facilement le nombre de réglages disponibles (en associant les notions de clavier, écran, imprimante, souris,... à celle de terminal), ce dont les utilisateurs passionnés sont friands. Quand à la notion d'annuaire, elle permet de se servir du système en reprenant la métaphore de la communication téléphonique, ce que tous les utilisateurs envisagés connaissent. D'autres modèles sont possibles ainsi, ne peut-on pas dire que : ***le point de vue que le concepteur adoptera pour modéliser les objets sous-jacents à l'application se reflètera dans son système, avant même tout prototype***?

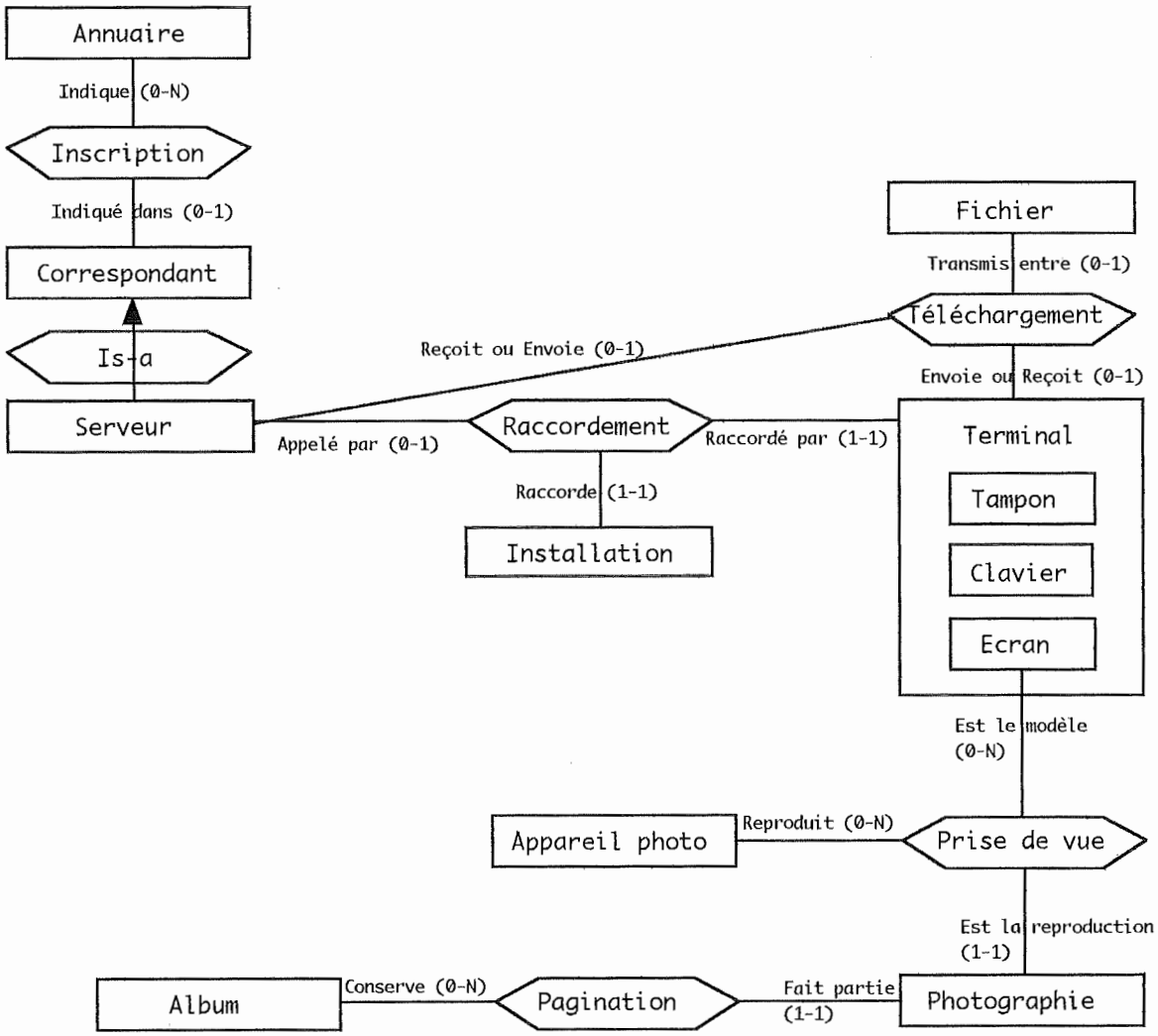


Fig. 5.22 : Structure consolidée des objets de l'application

Prenons l'exemple du Système 7 d'Apple et de la tâche de réglage. Le Système 7 est le dernier système d'exploitation d'Apple. Il est distribué avec tout Macintosh. Ce Système 7 contient une "Communications Toolbox" qui est idéale pour concevoir rapidement un programme tel que Polycom. Ces outils sont destinés à des utilisateurs avertis. Apple place une grande quantité de réglages sous la responsabilité de l'utilisateur. Ce n'est pas le cas de Devlonics qui préfère réduire le nombre de réglages disponibles. Les réglages sont regroupés fonctionnellement selon trois axes : le terminal, la connexion et le transfert de fichier. Le terminal reprend de manière très complète les objets *Terminal, Clavier et Ecran* (fig. 5.23, page suivante). Tandis que avec la connexion, les objets *Installation, Raccordement, Serveur* et *Correspondant*[4] sont regroupés. Le transfert de fichiers reprend des réglages qui n'intéressent pas Devlonics.

[4] Une remarque déjà citée à propos des essais sur prototypes : les utilisateurs associent le numéro de téléphone du serveur avec "correspondant" plutôt que "raccordement" et que "raccordement" est préféré à "connexion".

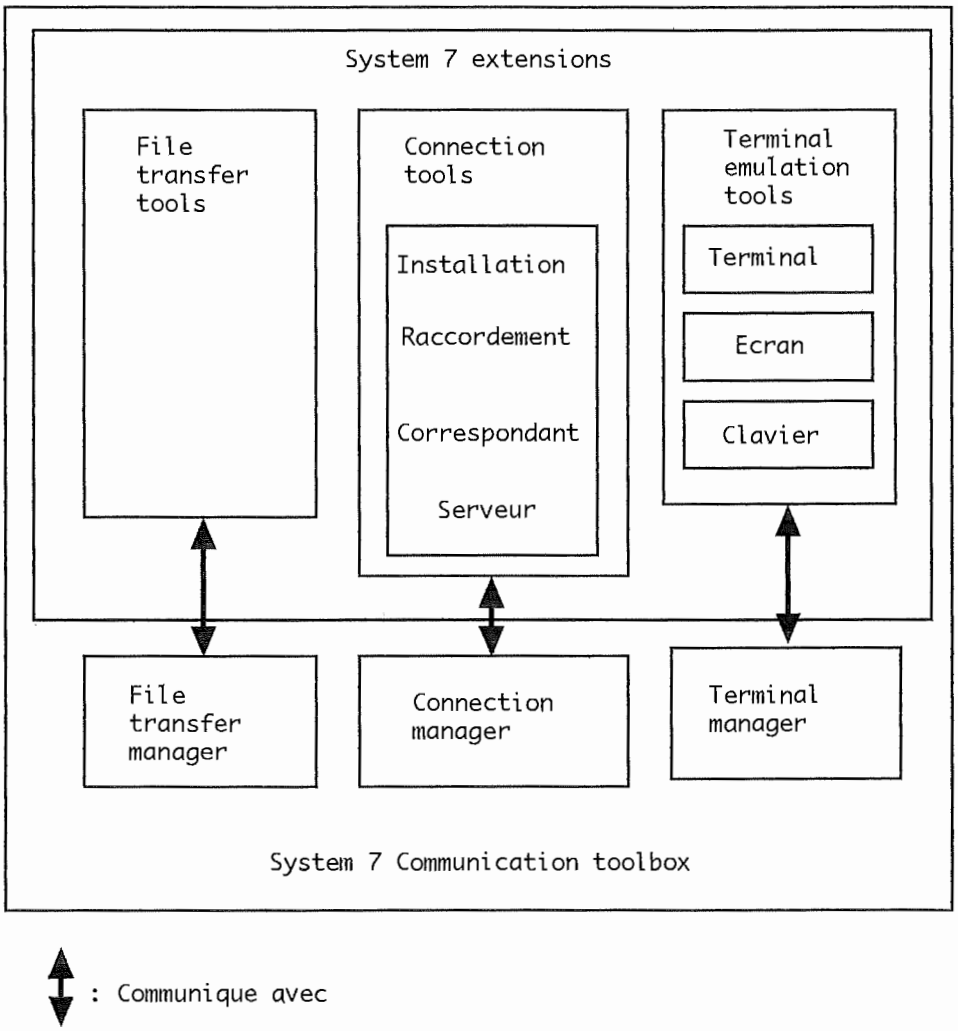


Fig. 5.23 : La structure des objets de l'application proposée par Apple avec le Système 7

Par respect de la cohérence, nous aurions dû reprendre les outils fournis par Apple et donc proposer un modèle reprenant tel quel le "terminal" et la "connexion". Nous ne l'avons pas fait parce que d'une part, bien que les prototypes soient réalisés sur Macintosh, Devlonics les destine aussi aux utilisateurs de PC pour lesquels Microsoft ne propose pas (encore?) de tels outils. D'autre part, ces objets peuvent toujours être replacés dans notre modèle sans le remettre entièrement en cause, bien que nous ayons montré le risque d'utiliser une vue fonctionnelle.

Quant à la tâche de réglage avec Eurocom, elle est aussi abordée d'un point de vue fonctionnel. La procédure d'installation fait appel à un objet similaire à *Installation* (fig. 5.24, page suivante). Elle a été développée pour des utilisateurs avertis. La procédure de mise à jour de l'annuaire des services télématiques fait appel à l'objet *Annuaire* qui regroupe d'une part, sans distinction, les deux objets *Correspondant* et *Serveur* et, d'autre part, toujours sans distinction, les deux objets *Terminal* et *Raccordement*. Des utilisateurs avertis peuvent ainsi fournir une solution toute faite sous la forme d'une entrée dans un annuaire pour ceux qui ne sont pas avertis. Les objets *Clavier* et *Ecran* n'ont pas d'équivalent dans Eurocom. Pour la tâche d'archivage des écrans reçus, on parlera des Pages, Chapters et Libraries (qui elles-

mêmes sont contenues dans une Library). Voici l'explication tirée de la documentation d'Eurocom:

"The library consists of 4 libraries :
- Prestel library for Prestel and Bistel Pages
- Télétel library
- BTX library
- TTY library for TTY and VT100 pages.
Each library is organised in chapters, containing several 'pages'."

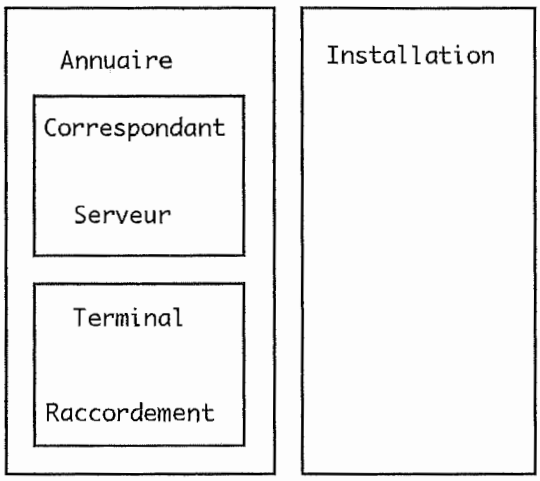


Fig. 5.24 : La structure des objets de l'application proposée par Devlonics avec Eurocom

6. NIVEAU SEMANTIQUE

Pour rappel, le niveau sémantique porte sur la représentativité pour la personne du comportement d'un objet, c'est-à-dire "the study of the relationships between signs and symbols and what they represent".

Au niveau sémantique, nous décrivons le comportement et les attentes naturelles de l'utilisateur qui effectue des opérations sur les objets de l'application. D'une part, l'utilisateur s'attend à devoir procéder d'une certaine façon pour exécuter telle opération sur tel objet. D'autre part, il s'attend à un comportement significatif des objets pour évaluer si telle action est bien réalisée comme il s'y attendait. Par exemple, que se passe-t-il si l'utilisateur ouvre l'annuaire téléphonique à une lettre de l'alphabet? Que se passe-t-il si on demande au Polychinel de boire un pot? Comment sortir les pelles du Polychinel?

Nous nous attachons à la signification des expressions dans le langage de l'interface. Nous nous plaçons aux étapes de la formation d'une intention (1), de la spécification d'une suite d'action (2), de l'interprétation (2') et de l'évaluation (3') sur le versant concerné par la nature des actions sur le schéma du modèle de Norman (fig. 6.1, page suivante). Nous allons nous attaquer à l'écart entre les attentes que l'utilisateur place dans les actions mises en œuvre avec Polycom et, celles que nous leur attribuons. Le comportement des objets ne doit pas être dénaturé par rapport aux expectatives de l'utilisateur. Si au point de départ de l'étude, nous prenons nous-même la place de l'utilisateur, par après il y aura ajustement progressivement au moyen des prototypes. Nous demanderons à l'utilisateur de raconter ce qu'il fait quand il manipule un prototype. Nous nous servirons de critères d'ergonomie tels que repris dans [Bod89a], [Bod89b pp57-58], [Cou90a pp57-88], [Mei90 pp101-103], [Sca pp13-14], [Tog92].

Par tâche, pour chaque opération que l'utilisateur effectue, nous appliquons la grille *opération - conduite - attente* qui est construite comme suit (sur base de [Mei91 pp100-103]) : J'ai l'intention d'effectuer une ***opération.*** Pour cela, je m'attends à agir d'une certaine façon sur l'image que je perçois des objets de l'application. J'adopterai une ***conduite*** pour manipuler ces objets. En retour, je m'***attends*** au résultat que j'escompte. J'émets parfois des suggestions pour me faciliter la tâche. Pour garder la cohérence inter-applications, les opérations propres à l'environnement de travail proposé par le système d'exploitation de l'utilisateur devront aussi être prises en compte. Par exemple : fermer une fenêtre, enregistrer un fichier, copier une photo dans le presse papier... Nous citerons aussi les ***objets*** de l'application et leurs ***attributs*** nécessaires à la conduite des opérations ainsi que les ***erreurs*** qui peuvent être générées.

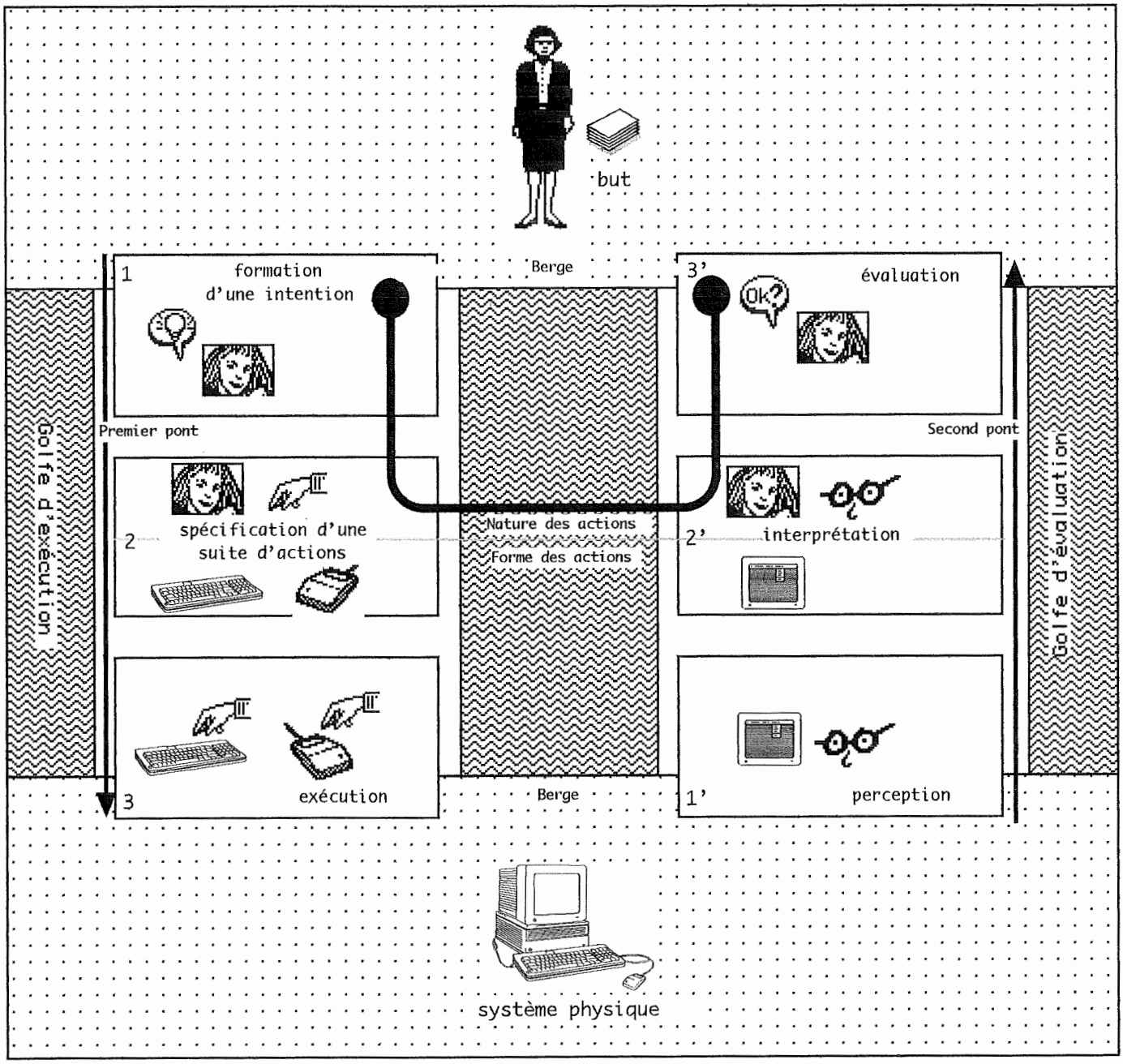


Fig. 6.1 : Le trait épais terminé par deux boules marque le niveau sémantique sur le schéma du modèle de Norman

Nous proposons de regrouper les opérations en phases (ce terme de phase est emprunté à [Bod89b p56]). La phase marquera pour l'utilisateur un ensemble d'opérations auquel il attribue une unité sémantique quand il les exécute. Une untité sémantique s'observe lorsque l'utilisateur est mentalement préoccupé, à un moment donné, par les diverses opérations qui constituent une phase. Pour mettre une phase en évidence, nous avons travaillé par ajustements progressifs. Par exemple, pour régler son système, l'utilisateur règle le terminal, le raccordement et indique le correspondant d'une seule traite, sans configurer l'installation.

De temps en temps il réalise ces réglages séparément ou ensemble. Prenons l'exemple des étapes parcourues par l'utilisateur averti pour se connecter à un nouveau centre serveur:

- Il relie son modem à son ordinateur et au réseau téléphonique.
- Il configure le logiciel pour qu'il reconnaisse son équipement.
- Il se renseigne sur l'existence d'un serveur (bouche à oreille, publicité, magazine).
- Le serveur lui signale la configuration requise pour rentrer en contact avec lui. Ci-joint trois cas:

1°) [In *émission télévisée sur TF1*]

Vous pouvez nous appeler sur le 3615 du Minitel (en France), code de service TVHA.

2°) [In *Manuel d'utilisation Homebanking BBL* - BBL, octobre 1991, page 2]

Vous pouvez appeler le service Homebanking de la BBL:

- par liaison téléphonique directe:

Numéro de téléphone	Vitesse	Modem
02/7353081	2400/2400	V22bis
02/7350168	1200/1200	V22
02/7368020	1200/75	V23

- via le réseau DCS: NUA 299001869
- via le réseau Vidéotex: numéro de téléphone 1310

Les paramètres de votre logiciel de communication doivent être les suivants:
- mode full duplex, donc sans écho local;
- les caractères sont transmis sous la forme de 7 bits de données, 1 bit de parité et 1 bit d'arrêt. La parité paire est maintenue lors de la transmission; à la réception, la parité n'a pas d'importance;

- Homebanking utilise l'émulation d'écran VT102.

3°) [In *Mactivity Magazine* - Mactivity International Ltd., mars 1992, page 12]

Il existe plusieurs n° de tél. aux Pays-Bas et en Belgique permettant aux membres du macclub de s'exercer gratuitement en matière de télécommunication.

- Bulletin Board Service à Bullange: 080-647363

- BBS à Maastricht: 043-619204

Ces BBS sont accessibles sur 1200-2400-9600 bauds et les réglages du modem sont 8-N-1.

- Il configure le logiciel pour qu'il reconnaisse le serveur.
- Il appelle le serveur.
- Il reconfigure son clavier pour l'adapter aux possibilités du serveur (ou si on préfére, il tape au hasard sur les touches de son clavier pour voir lesquelles servent à quoi).

Dans cet exemple, les phases retenues sont:

- •Configurer l'installation : L'utilisateur configure le logiciel pour qu'il reconnaisse son équipement. Il configure l'installation indépendamment des autres réglages.
- •Pouvoir appeler le serveur : L'utilisateur configure le logiciel pour qu'il reconnaisse le serveur. Il règle le terminal, le raccordement et indique le correspondant d'une seule traite.
- •Etablir la communication : L'utilisateur appelle le serveur.
- •Adapter le clavier : L'utilisateur reconfigure son clavier pour l'adapter aux possibilités du serveur.

Dans ce qui suit, nous détaillons quelques opérations qui nous ont paru significatives dans le cadre de notre étude. Ces opérations font parties de trois tâches parmi celles qui ont été abordées au niveau conceptuel. Il s'agit des tâches : régler son système, appeler le centre serveur et archiver des écrans reçus. Une quatrième tâche qui concerne l'environnement de travail de l'utilisateur, est citée afin de prendre en compte le critère d'ergonomie d'universalité (émis par [Mei90 p101]). Ce critère d'universalité propose la "prise en compte de commandes à caractère universel qui se retrouvent sur tous les systèmes, telles que celles de gestion de fichier (...) ou celles d'édition (...); elles sont un élément de consistance de l'interface à travers les applications, voire à travers les différents systèmes."

6.1. TACHE : REGLER SON SYSTEME

Phase : Configurer l'installation

L'utilisateur averti configure le logiciel pour qu'il reconnaisse son équipement. Il configure l'installation indépendamment des autres réglages.

Opération: Configurer l'installation

```
Objet(s): Installation.
Attribut(s): Modem, Port, Signal sur la ligne, Numérotation.
Conduite: Il indique les réglages propres à son installation. Il le fait le moins
       souvent possible, c'est-à-dire seulement quand il change la configuration
       de son équipement. Les réglages ne pourraient-ils pas s'effectuer
       automatiquement? Il s'attend à ce que Polycom se serve de cette
       configuration à l'avenir. Les réglages concernent : le modem (il en indique
       la référence fournie par le constructeur du modem), le port (il en indique
       la référence selon les symboles qui représentent ce port sur son PC ou son
       Macintosh) et, la numérotation (il indique si ce sont des tonalités ou des
       impulsions) et, le signal sur la ligne (il indique si il souhaite
       l'entendre ou non).
Attente(s): Polycom indique les réglages effectués. Polycom ne peut-il pas
       reconnaître l'équipement de lui-même?
Erreur(s): Il n'a pas indiqué le modem qui correspond à celui dont il se sert. Le
       modem n'est pas relié au port indiqué. La numérotation par tonalité a été
       demandée alors qu'il est raccordé avec un ancien central téléhponique
       électro-mécanique.
```

Opération: Vérifier si l'installation est correctement configurée

Objet(s): Installation.
Attribut(s): Modem, Port.
Conduite: Il demande à Polycom de vérifier si son installation est correctement configurée.
Attente(s): Polycom montre si un modem répondant aux commandes de celui qui est renseigné, est bien relié au port série renseigné et, si un signal est présent sur la ligne quand le modem "la décroche".
Erreur(s): Il y a un problème avec le port ou le modem (parce qu'il est mal raccordé ou que le modem renseigné n'est pas le bon). Il n'y a pas de signal présent sur la ligne.

Phase : Pouvoir appeler le serveur

L'utilisateur averti configure le logiciel pour qu'il reconnaisse le serveur. Il régle le terminal, le raccordement et indique le correspondant d'une seule traite.

Opération: Régler le terminal

Objet(s): Le terminal concerné par ces réglages.
Attribut(s): Emulation du terminal, Caractères, «saut de ligne» lors d'un «Retour chariot», Echo local et Identification.
Conduite: Les utilisateurs dénomment l'émulation du terminal de différentes façons. Voici des associations de dénominations:
(1) «Ansi», «BBS (Bulletin Board System)»;
(2) «Bistel» «Terminal du gouvernement belge, de Ma(e)rtens»;
(3) «Bildschirmtext», «Vidéotex»;
(4) «Prestel», «Vidéotex»;
(5) «Télétel», «Minitel», «Vidéotex (français)»;
(6) «TTY», «Télétype»;
(7) «VT102» «Terminal d'un ordinateur».
Les paramètres retenus sont: l'émulation du terminal (l'utilisateur indique le type de terminal à émuler), les caractères (il indique le jeu de caractère du terminal), le «saut de ligne» lors d'un «retour chariot» (il indique si il faut le faire ou pas), l'identification (il renseigne un code), et l'écho local (il indique si il y a ou non écho local). Certains de ces paramètres sont sans objets avec certains terminaux:

Si l'émulation du terminal est:	Alors les choix possibles pour les caractères sont:	Alors les autres réglages possibles sont:
Ansi	Ansi, Ascii américain, Ascii français, Macintosh	L'échol local
Bildschirmtext	Vidéotex allemand	-
Bistel	Vidéotex belge	L'identification
Prestel	Vidéotex anglais, Vidéotex belge	L'identification
Télétel	Vidéotex français	L'identification
TTY	Ansi, Ascii américain, Ascii français, Macintosh	L'écho local et le «saut de ligne» lors d'un «retour chariot»
VT102	Ascii américain, Ascii français	L'écho local

Le recours à des paramètres encore plus fin n'est pas impossible. Il faut tenir compte de la manière avec laquelle le serveur exprime ces réglages.
Attente(s): Polycom montre le terminal choisi et les réglages effectués.
Erreur(s): Les réglages du terminal ne correspondent pas aux attentes du serveur. C'est seulement avec les premières données échangées avec le serveur que l'utilisateur le voit parce que l'écran se remplit anormalement (de caractères bizarres).

Opération: Régler le raccordement

Objet(s): Le raccordement et le correspondant/serveur concernés par ces réglages.
Attribut(s): Numéro de téléphone du correspondant, Vitesse du modem, Bits de donnée, Bit de parité, Bits de stop, Correction des erreurs et compression des données.
Conduite: Habituellement, il indique la vitesse du modem la plus rapide possible entre son modem et celui du centre serveur. Si la ligne est trop mauvaise, il ne le fait pas. Il règle le raccordement en fonction du point d'accès au serveur, c'est-à-dire le numéro de téléphone du correspondant. Les réglages sont : le numéro de téléphone (il doit respecter une syntaxe décodée par le modem), la vitesse du modem (il indique la vitesse tantôt en bits par seconde, tantôt en fonction des normes du CCITT) , les bits de donnée (il indique le nombre de bits), le bit de parité (il indique si elle est paire, impaire au inexistante), les bits de stop (il indique le nombre de bits) et, la correction des erreurs et la compression des données (il indique si le mécanisme est MNP ou V42/V42bis ou inexistant). Il faut tenir compte de la manière avec laquelle le serveur exprime ces réglages.
Attente(s): Polycom me montre les réglages effectués. Pour pouvoir communiquer avec un centre serveur, il n'y a pas que le raccordement à régler. Quels sont les autres réglages qu'il devra effectuer?
Erreur(s): Les réglages du raccordement ne correspondent pas aux attentes du serveur. C'est seulement avec les premières données échangées avec le serveur que l'utilisateur le voit à cause des erreurs de transmission.

Opération: Indiquer le correspondant

Objet(s): Le correspondant/serveur concerné par ces réglages.
Attribut(s): Numéro de téléphone du correspondant, Nom, Remarques.
Conduite: L'essentiel, c'est de donner le numéro de téléphone du correspondant qui est le serveur. L'utilisateur a des difficultés à encoder un numéro de téléphone dans une syntaxe comprise par le modem, surtout qu'il y a beaucoup de manières de représenter un numéro de téléphone. Après, il donne un nom au correspondant. Il trouve parfois pratique de pouvoir aussi ajouter des commentaires comme des mots de passe ou les heures d'ouverture du serveur ou le coût d'accès à ce serveur.
Attente(s): Polycom lui montre les choix effectués. L'utilisateur se base sur le numéro de téléphone pour régler le raccordement. Il n'est pas indispensable d'indiquer le nom et de renseigner des remarques pour pouvoir communiquer avec un centre serveur.
Erreur(s): Le numéro de téléphone est erroné parce que l'utilisateur a, par exemple, oublié qu'il fallait passer par un central téléphonique pour sortir du bâtiment ou qu'il téléphone depuis l'étranger. Cet erreur est plus commune aux utilisateurs d'ordinateur portable.

6.2. TACHE : APPELER LE CENTRE SERVEUR

Phase : Etablir la communication

L'utilisateur appelle le serveur.

Opération: Appeler

Objet(s): Le correspondant/serveur auquel il va se connecter. Le terminal, le raccordement et l'installation concernés par cet appel.
Conduite: Il attend de voir si la communication s'établit correctement. Si c'est le cas, le serveur va prendre en charge la communication et l'écran du terminal se garnit.
Attente(s): Disposer d'une estimation du coût de la communication, tout comme de l'état d'évolution de la liaison: l'utilisateur n'est pas connecté, Polycom initialise le modem; il compose le numéro de téléphone; il ne sait pas composer le numéro de téléphone parce qu'il n'a pas de tonalité sur la ligne; cela sonne de l'autre côté de la ligne; cela sonne occupé; on ne décroche pas; la communication est inopinément interrompue; l'utilisateur est connecté. Cette liste n'est qu'indicative.

Erreur(s): Il y a un problème à l'établissement de la connection. L'utilisateur a oublié de donner un numéro de téléphone correct au correspondant. L'installation n'est pas correctement configurée. Les réglages du terminal ou du raccordement ne sont pas corrects.

Phase : Terminer la communication

L'utilisateur coupe la communication.

Opération: Raccrocher

Objet(s): Le correspondant/serveur duquel il va se déconnecter. Le terminal, le raccordement et l'installation concernés par cet appel.
Conduite: Avant de raccrocher, il a suivi la procédure indiquée par le serveur pour terminer la communication. Ensuite il demande à Polycom de raccrocher et attend de voir si la ligne téléphonique est libérée.
Attente(s): Polycom indique que la ligne téléphonique est bien libérée.

6.3. TACHE : ARCHIVER DES ECRANS REÇUS

Phase : Photographier et insérer la photo dans un album

L'utilisateur prend une photo supplémentaire de l'écran du terminal.

Opération: Photographier

Objet(s): L'écran du terminal concerné. L'appareil photo et une photographie.
Conduite: Il attend que la photo soit prise.
Attente(s): Un éclair de flash marque la prise de vue sans interrompre la communication.

Opération: Placer une photo dans l'album

Objet(s): La dernière photographie prise. Un album photos.
Attribut(s): La page/endroit où intercaller la photo dans l'album.
Conduite: Après qu'une photo soit prise, il indique un album et à quel endroit il souhaite l'y placer.
Attente(s): Quand la photo est collée dans l'album, l'utilisateur reprend la communication. Quand il donne l'ordre de photographier, la photographie s'insère d'elle-même à la suite des autres photos dans un album qu'il a renseigné auparavant.

6.4. TACHE LIEE A L'ENVIRONNEMENT DE TRAVAIL DE L'UTILISATEUR

Phase : Sauvegarder les réglages propres à un serveur

L'utilisateur souhaite conserver les réglages propres à un serveur.

Opération: Enregistrer la configuration d'un terminal pour un serveur

Objet(s): Le terminal, le raccordement et le correspondant/serveur concernés.
Attribut(s): à déterminer.
Conduite: Il attend la fin de l'enregistrement.
Attente(s): Il voit qu'il doit attendre, la progression de l'opération et une indication de ce qui est enregistré (comme 'Les réglages concernant tel serveur sont enregistrés').
Erreur(s): Problèmes d'accès au fichier liés à l'environnement de travail propre à l'ordinateur.

7. NIVEAU SYNTAXIQUE

Pour rappel, le niveau syntaxique porte sur porte sur la formalisation du comportement des objets, c'est-à-dire "the branch of semiotics that deals with the formal properties of symbol systems".

L'utilisateur se pose la question "Comment faire avec les moyens interactifs pour?". Par exemple, comment faire réellement pour ouvrir l'*annuaire* téléphonique à la lettre 'A'? Comment faire effectivement pour sortir les pelles du Polychinel?

Nous laissons l'analyse de la distance sémantique entre les objectifs de l'utilisateur et la signification d'une expression dans le langage de l'interface. Cette analyse portait sur les deux niveaux pragmatique et sémantique de l'interface. Maintenant, nous abordons l'analyse de la distance articulatoire entre la signification d'une expression dans le langage de l'interface et sa forme au sein des moyens d'interactions. Cette analyse portera sur les deux niveaux syntaxique et lexical de l'interface. La terminologie adoptée est reprise de [Pro90].

Nous nous intéressons à la forme des expressions dans le langage de l'interface. Nous nous plaçons aux étapes de la spécification d'une suite d'action (2), de l'exécution (3), de l'interprétation (2') et de la perception (1') sur le versant concerné par la forme des actions sur le schéma du modèle de Norman (fig. 7.1, page suivante). Notre but est de réduire la distance articulatoire en agissant sur la forme des séquences d'actions élémentaires de base, ainsi que sur la forme des retours.

Le passage du niveau sémantique au niveau syntaxique demandera d'associer des comportements de la personne à des comportements de la machine. Au cours de ce passage, nous fixerons une succession d'actions à exécuter pour accomplir chaque opération du niveau sémantique. Pour cela nous verrons quels sont les moyens d'interactions mis à la disposition de l'utilisateur et quels styles d'interactions nous adoptons. De là, nous indentifierons les objets interactifs abstraits de Polycom, en prenant comme exemple le réglage du terminal. Nous nous servirons de recommandations sur l'ergonomie telles que reprises dans [Van92], [Sca], [Tog92], [Cou90a pp57-88], [Mei90 pp103-110].

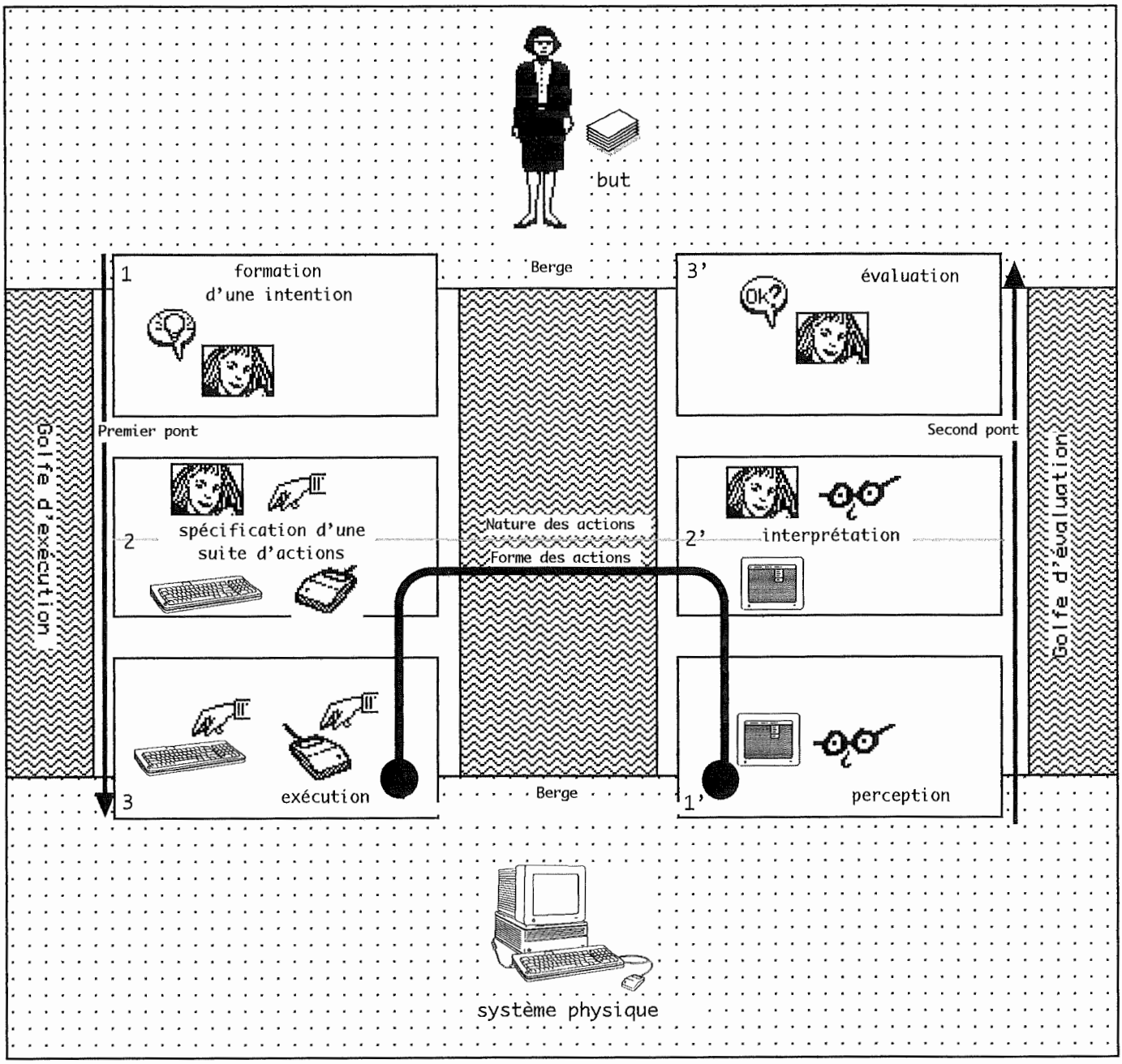


Fig. 7.1 : Le trait épais terminé par deux boules marque le niveau syntaxique sur le schéma du modèle de Norman

7.1. LES MOYENS D'INTERACTIONS

Les moyens d'interactions sont le Macintosh (ou le PC), son clavier, sa souris et l'environnement de fenêtrage du Système 7 (ou Windows). A ceux là, nous devrions ajouter le modem qui contient un dispositif sonore et (si le modem est externe) lumineux pour reporter son état à l'utilisateur.

7.2. LES STYLES D'INTERACTIONS

Le style d'interaction dépendra "du niveau d'entraînement de l'utilisateur et d'un temps de réponse système tolérable" [Van92 p175], ainsi que de l'opération et de l'objet envisagés. Avec les premiers prototypes, nous allons construire l'interface sur la métaphore de la conversation. L'utilisateur et le système auront une conversation au sujet d'un monde supposé mais non explicitement représenté. Progressivement, nous allons passer à une interface basée sur la métaphore du mini-monde qui met en relief les capacités de reconnaissances de l'utilisateur. Passons en revue les styles d'interactions repris dans Polycom:

•Les fenêtres (fig. 7.2, page suivante). Des objets de l'application seront assemblés pour former les fenêtres-documents *Terminal, Tampon, Album photos* et la fenêtre *Clavier*. La fenêtre-document *Terminal* reprendra des informations sur le serveur (qui est un correspondant), sur le raccordement, ainsi que la réprésentation de l'écran du terminal. La fenêtre *Clavier* reprendra la représentation des touches de fonctions propres au terminal, qui ne se retrouvent pas sur le clavier de l'ordinateur de l'utilisateur. La fenêtre-document *Tampon* reprendra le tampon. La fenêtre-document *Album photos* reprendra la représentation d'un album et de ses photos.

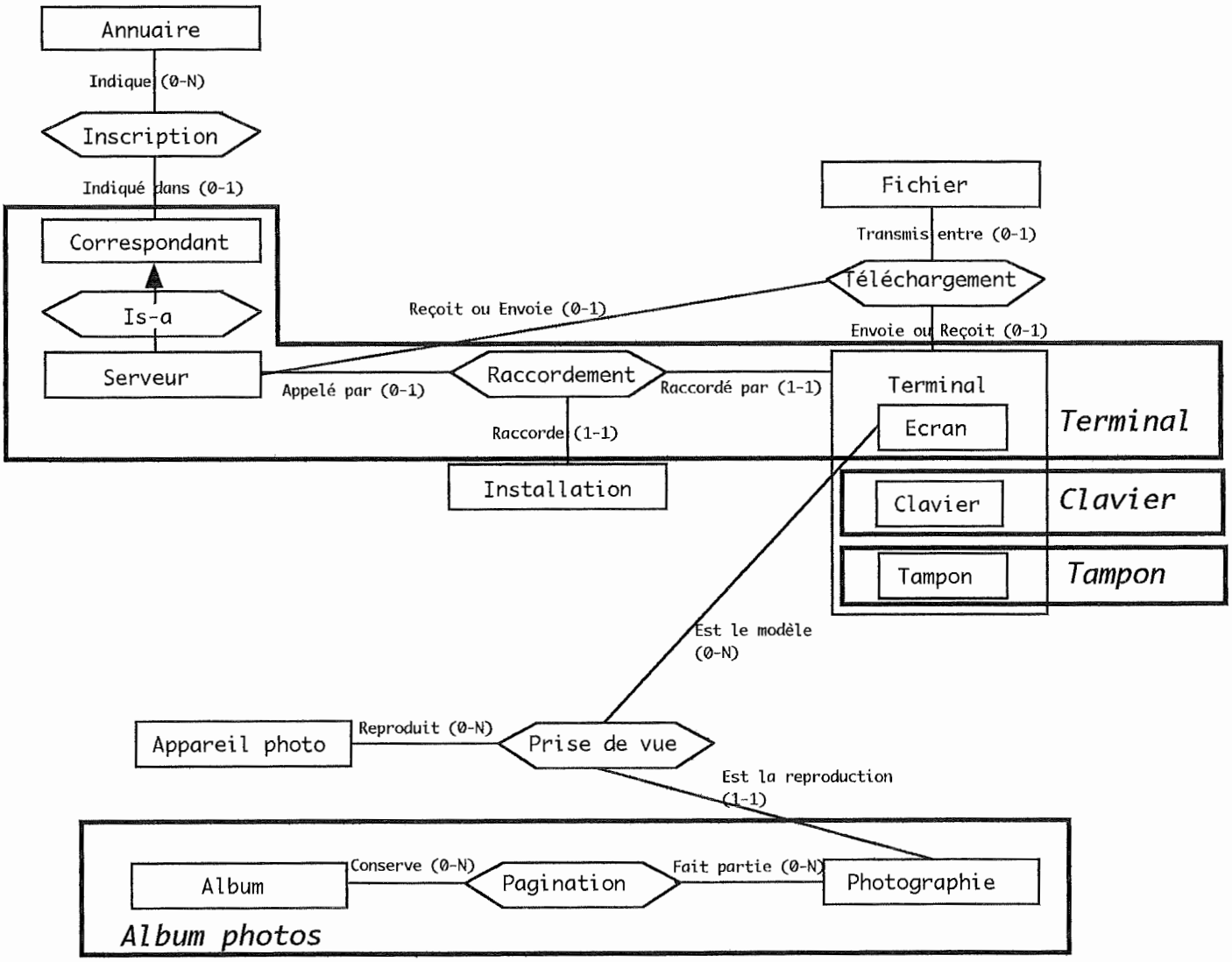
•L'objet *Annuaire* pourrait être représenté par une fenêtre-document et refléter le comportement attendu d'un annuaire papier. Mais avant de décider cela, nous proposons de voir si nous ne pouvons pas l'intégrer aux notions déjà supportées par le système d'exploitation de l'ordinateur de l'utilisateur. L'*Annuaire* serait composé d'un ou plusieurs 'dossiers' et les *Serveurs* y seraient regroupés et stockés sous forme de 'documents', ainsi l'utilisateur se comporte comme il en a l'habitude pour les documents issus d'autres applications. Nous observerons particulièrement l'attitude et les attentes de l'utilisateur pour attribuer un nom (de fichier) à un serveur, ainsi que pour retrouver un serveur afin de rentrer en contact avec lui. Sous Windows, 'BBLHB.POL' est-il aussi explicite que 'Home Banking de la BBL' sous Système 7, est-il aussi facile d'accès?

•La sélection de menu : "lorsque l'utilisateur a peu ou aucun entraînement et, que le temps de réponse système doit être très rapide" [Van92 p175]. Nous établissons un dialogue objet-opération où l'utilisateur sélectionnera d'abord l'objet avant d'effectuer une opération dessus, excepté pour le transfert de fichier. Par exemple il sélectionnera une fenêtre *Terminal* avant de demander d'effectuer l'opération 'Indiquer le correspondant' par sélection de menu. Les items de la barre de menu sont de préférence regroupés par ordre fonctionnel [Van92 p183]:

-«Fichier» que l'utilisateur a l'habitude de retrouver. Les fichiers font référence aux terminaux, aux albums photos et aux tampons (bien que dans le cas des tampons, il n'y aura moyen que d'en avoir un d'ouvert à la fois).
-«Edition» le menu habituel pour accéder au presse-papier.
-«Réglage». Derrière cet item se retrouve tout ce qui concerne la tâche de réglage du terminal, de son clavier et de son écran, du raccordement et, de l'installation.
-«Session». Derrière cet item se retrouve tout ce qui est lié à la tâche d'appel d'un centre serveur et d'archivage de ce qui est transmis.

-«Téléchargement». Derrière cet item se retrouve tout ce qui est lié à la tâche d'échange de fichier. Nous discutons plus bas du choix de l'emplacement pour ce menu.

-«Ecran» que l'utilisateur a l'habitude de retrouver et qui permet d'accéder aux différentes fenêtres dont l'unique tampon, le(s) album(s) photos et, le(s) terminal(aux).



: fenêtre-document contenant des objets de l'application.

Fig. 7.2 : Regroupement des objets de l'application en fenêtres-documents

•Le remplissage de forme : "lorsque l'utilisateur a peu d'entraînement ou un entraînement modéré et que le temps de réponse système peut être lent" [Van92 p175]. Le dialogue se fera par sélection de menu suivi d'un remplissage de forme quand il s'agira de préciser les paramètres d'une commande. Voici l'exemple du style de l'interaction du réglage du terminal:

L'utilisateur sélectionne un item de menu parmi «Ansi» , «Bildschirmtext», «Bistel», «Prestel», «Télétel, Minitel 1B», «TTY», «VT102» (fig. 7.3, 1). Ensuite le dialogue se poursuivra au moyen d'un remplissage de forme pour régler les paramètres propres au terminal choisi (fig. 7.3, 2).

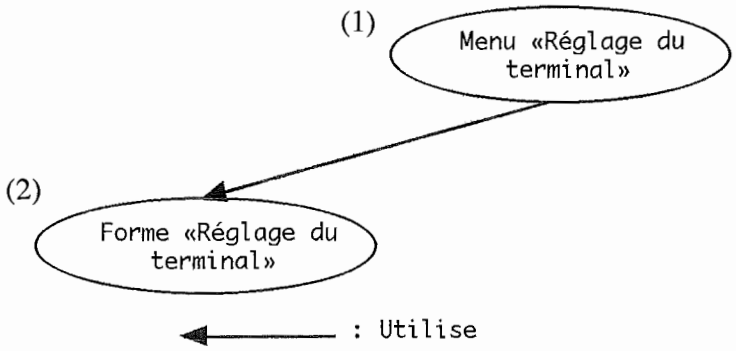


Fig. 7.3 : Structure schématisée du style de l'interaction «réglage du terminal»

•L'interaction iconique : "lorsque l'utilisateur a un entraînement élevé à modéré et que le temps de réponse système doit être très rapide" [Van92 p175]. L'interaction graphique sera utilisée pour associer des actions à des icones (pour un exemple, voir le choix du «port», dans le réglage de l'installation, au niveau lexical). Des icones permettront à l'utilisateur d'actionner les commandes les plus fréquentes (comme Appeler, Raccrocher, Photographier), de parcourir l'album ou le tampon. Elles serviront aussi à représenter graphiquement des touches du clavier du terminal à émuler, qui ne sont pas présentes sur le clavier de l'ordinateur de l'utilisateur. L'interaction iconique s'est aussi avérée fort utile pour représenter un objet afin que l'utilisateur voit mieux de quoi est-il question.

•Le langage de programmation : "lorsque l'utilisateur a un entraînement élevé et que le temps de réponse système peut être modéré à lent" [Van92 p175]. Pour la commande 'Appeller', nous pourrions avoir un dialogue sous la forme d'un langage de programmation. Un texte reprendrait une série de commandes à effectuer pour réaliser la tâche. Ce style de dialogue destiné aux utilisateurs ayant un entraînement élévé à l'utilisation de Polycom, leur permettrait, par exemple, de fournir une solution sur mesure à des utilisateurs ayant peu à aucun entraînement avec Polycom. Cette forme de dialogue est souvent reprise sous le nom de script dans les logiciels de télécommunication. Prenons l'exemple de *Crosstalk Communicator V2.0* pour MS-DOS. C'est une version simplifiée de *Crosstalk Mk. 4*. *Crosstalk Communicator V2.0* coûte 99$ au lieu de 245$ pour *Mk. 4*. Alors que *Mk. 4* permet de développer des scripts avec le langage CASL (Crosstalk Application Script Language) de plus de 300 verbes, *Communicator* permet de développer des scripts avec seulement 17 verbes tout en pouvant exécuter les scripts développés avec *Mk. 4* et ses 300 verbes. Un autre exemple? le langage SUPERSCRIPT de VICOM Pro, pour Macintosh, propose plus de 200 verbes pour développer des scripts qui tourneront avec la version simplifiée du logiciel, à savoir VICOM Connect qui ne dispose pourtant que d'un langage de 35 verbes. Nous n'avons pas envisagé plus en avant le langage de programmation, ni une alternative avec un mécanisme de macro-commande.

7.3. IDENTIFICATION DES OBJETS INTERACTIFS ABSTRAITS

Les interactions porteront le plus souvent sur des objets interactifs abstraits tels que décrits dans [Pro90]. L'identification des objets interactifs abstraits sur base des opérations issues du niveau sémantique, est réalisée avec l'aide de l'utilisateur et de [Van92]. Ce document est une synthèse de règles sur l'ergonomie présentée par J. Vanderdonckt, sous la forme d'un *Guide ergonomique de la présentation des applications hautement interactives*. Par exemple, il s'agit de trouver sous quelle forme l'utilisateur indiquera le jeu de caractères du terminal.

Cette classification a aussi été élaborée avec l'aide de l'utilisateur alors qu'il manipulait des prototypes. Nous avons proposé à l'utilisateur différentes réalisations de base qui ont ensuite évoluées avec sa collaboration. Pour programmer ces prototypes en Think C(++) sous le Système 7 du Macintosh, nous avons d'abord eu recours à une librairie d'objets livrée avec le compilateur. Ensuite, pour disposer de la puissance des objets interactifs abstraits, nous avons programmé des classes d'objets interactifs concrets équivalents en C++. Par exemple, en déclarant l'objet de dialogue «vitesse» comme descendant de la classe «Liste déroulante», il se comporte comme une «Liste déroulante». Cette approche a fortement facilité l'élaboration du comportement et de l'apparence d'un objet interactif, la création de nouveaux objets interactifs et, la résolution de difficultés de programmation liées à l'utilisation de la boîte à outils du Macintosh. Ci-dessous nous présentons l'exemple des objets interactifs abstraits identifiés pour le réglage du terminal.

Dans le cas du réglage du terminal, nous avons identifié trois objets interactifs qui s'inscrivent dans la structure schématisée de l'interaction «réglage du terminal» (fig. 7.3). Il s'agit d'un menu, d'une boîte de dialogue et d'une fenêtre qui sont respectivement dénommés «mr_réglage_terminal», «dm_réglage_terminal» et «fe_terminal».

Objet interactif abstrait: mr_réglage_terminal

```
Définition: Groupement d'items radio du menu «réglage du terminal» qui reprend les
       objets interactifs abstraits avec lesquels l'utilisateur indique
       l'émulation du terminal à prendre en compte.
Nature: Dialogue.
Type: Composé d'items radio d'un menu.
Niveau de l'utilisateur: Averti.
Ordonnancement des objets interactifs: Selon l'ordre alphabétique [Van92 p183].
Agrégation des objets interactifs abstraits qui représentent l'émulation de
       terminal:
       ir_ansi : Item radio du menu, de libellé «Ansi».
       ir_bildschirmtext : Item radio du menu, de libellé «Bildschirmtext».
       ir_bistel : Item radio du menu, de libellé «Bistel».
       ir_prestel : Item radio du menu, de libellé «Prestel».
       ir_teletel : Item radio du menu, de libellé «Télétel, Minitel 1B».
       ir_tty : Item radio du menu, de libellé «TTY».
       ir_vt102 : Item radio du menu, de libellé «VT102».
Evénement(s): Hérités de la classe objet interactif abstrait «groupement d'items
       radio d'un menu».
Primitive(s): Idem que pour événements.
```

Objet interactif abstrait: dm_réglage_terminal

Définition: Boîte de dialogue modale intitulée «réglages du terminal» qui regroupe les objets interactifs abstraits avec lesquels l'utilisateur règle le terminal.
Nature: Dialogue.
Type: Composé des items de la boîte de dialogue.
Niveau de l'utilisateur: Averti.
Ordonnancement des objets interactifs: Selon l'ordre naturel pour l'utilisateur (il n'existe pas de document source) [Van92 p50].
Agrégation des objets interactifs abstraits:
- lb_type : Représente l'émulation de terminal. Texte non éditable pour indiquer le nom de l'émulation de terminal : «Ansi», «Bildschirmtext», «Bistel», «Prestel», «Télétel, Minitel 1B», «VT102», «TTY».
- br_caractère : Représente les caractères. Groupe de boutons radio de libellé «jeu de caractères» avec boîte de regroupement et au plus 7 valeurs actives à un moment donné [Van92 p20] parmi : «Ansi», «Ascii américain», «Ascii français», «Macintosh», «Vidéotex allemand», «Vidéotex anglais», «Vidéotex belge», «Vidéotex français».
- bc_écho : Représente l'écho local. Boîte à cocher [Van92 p20] de libellé «écho local».
- bc_saut_ligne : Boîte à cocher [Van92 p20] de libellé «saut de ligne».
- ed_identification : Représente l'identification. Champ d'édition uni-linéaire [Van92 p24] de libellé «Code identifiant l'utilisateur».
- bd_ok : Représente l'acceptation des réglages. Bouton par défaut de libellé «ok» [Van92 p86].
- bt_annuler : Représente l'annulation de toute modification des réglages. Bouton de libellé «annuler» [Van92 p86].
- bt_aide : Représente une demande d'aide pour effectuer les réglages du terminal. Bouton de libellé «aide» [Van92 p86].

Evénement(s): Hérités des objets interactifs abstraits agrégés.
Primitive(s): Idem que pour les événement(s).

Objet interactif abstrait: fe_terminal

Définition: Fenêtre-document fournissant une vue sur le groupement des objets de l'application *Serveur (Correspondant), Raccordement et Ecran (du terminal)*. Fe-terminal contient les objets interactifs abstraits avec lesquels l'utilisateur percevra le terminal.
Nature: Dialogue.
Type: Composé.
Niveau de l'utilisateur: Averti et non-averti.
Ordonnancement des objets interactifs: Selon l'ordre naturel pour l'utilisateur (il n'existe pas de document source) [Van92 p50].
Agrégation des objets interactifs abstraits: A déterminer.
Evénement(s): A déterminer.
Primitive(s): A déterminer.

8. NIVEAU LEXICAL

Pour rappel, le niveau lexical porte sur la symbolisation d'une réalité humaine, c'est-à-dire "the meaning of a word in relation to the physical world or to abstract concepts, without reference to any sentence in which the word may occur".

Au niveau lexical, les choix porteront sur la représentation des objets interactifs abstraits sous la forme d'objets interactifs concrets. Ils seront manipulés et perçus par l'utilisateur. Un objet interactif concret est la réalisation portée sur une machine d'un objet interactif abstrait. L'utilisateur se demande "De quoi est-il question avec ce symbole, avec ce mot, avec cette couleur?". Par exemple, que signifie l'icone ? Que veut-on représenter avec le bandeau de couleur vert ou rouge autour des pelles du Polychinel[5]?

Le centre d'intérêt est la représentation des expressions dans le langage de l'interface. Nous nous plaçons aux étapes de l'exécution (3) et de la perception (1') sur le schéma du modèle de Norman (fig. 8.1, page suivante). Pour cela, nous agissons sur la présentation physique des entrées et des retours. Notre but est de réduire la distance articulatoire entre le comportement de l'utilisateur et ce que nous lui demandons de faire. Nous prendrons l'exemple de l'objet interactif concret «vitesse», ainsi que l'exemple des boîtes de dialogue du réglage dans son ensemble dont le réglage du terminal qui a été détaillé au niveau syntaxique.

Nous nous servirons de recommandations sur l'ergonomie telles que reprises dans [Van92], [Tog92] ou dans la littérature des différents constructeurs selon l'environnement ciblé par l'implémentation [Apple, IBM, Microsoft]. ***Au-delà de toute justification portant sur l'ergonomie, une image de marque est véhiculée au travers des règles des constructeurs.*** Par exemple, une règle extraite de [IBM89 pp.80-81] reprise dans [Van92] précise que "La position du bouton de commande par défaut (dans une boîte de dialogue) devrait être [...] la plus à gauche si les boutons sont disposés horizontalement. L'ordre de positionnement des boutons de commandes d'une boîte de dialogue doit être le plus proche possible de l'ordre suivant, le cas échéant : (Ok), (Appliquer), (Réinitialiser), (Annuler), (Options), (Aide), (A propos de...) [...].". Cette règle peut être mise en balance avec une autre approche extraite de [App91a p2-29] qui justifie l'emplacement du bouton par défaut en bas et à droite de la boîte de dialogue. De plus Apple ne parle que de deux boutons. L'un (Annuler) est placé à gauche de l'autre (Action). ***Dans l'ensemble, il s'agit plus de valider des critères d'ergonomie qui impliqueront, au cas par cas, le respect de certaines règles (ou norme) d'ergonomie selon l'environnement de travail envisagé.***

---

[5] Cela indique l'emplacement de la pelle respectivement à main gauche, tribord, ou droite, babord, du rameur.

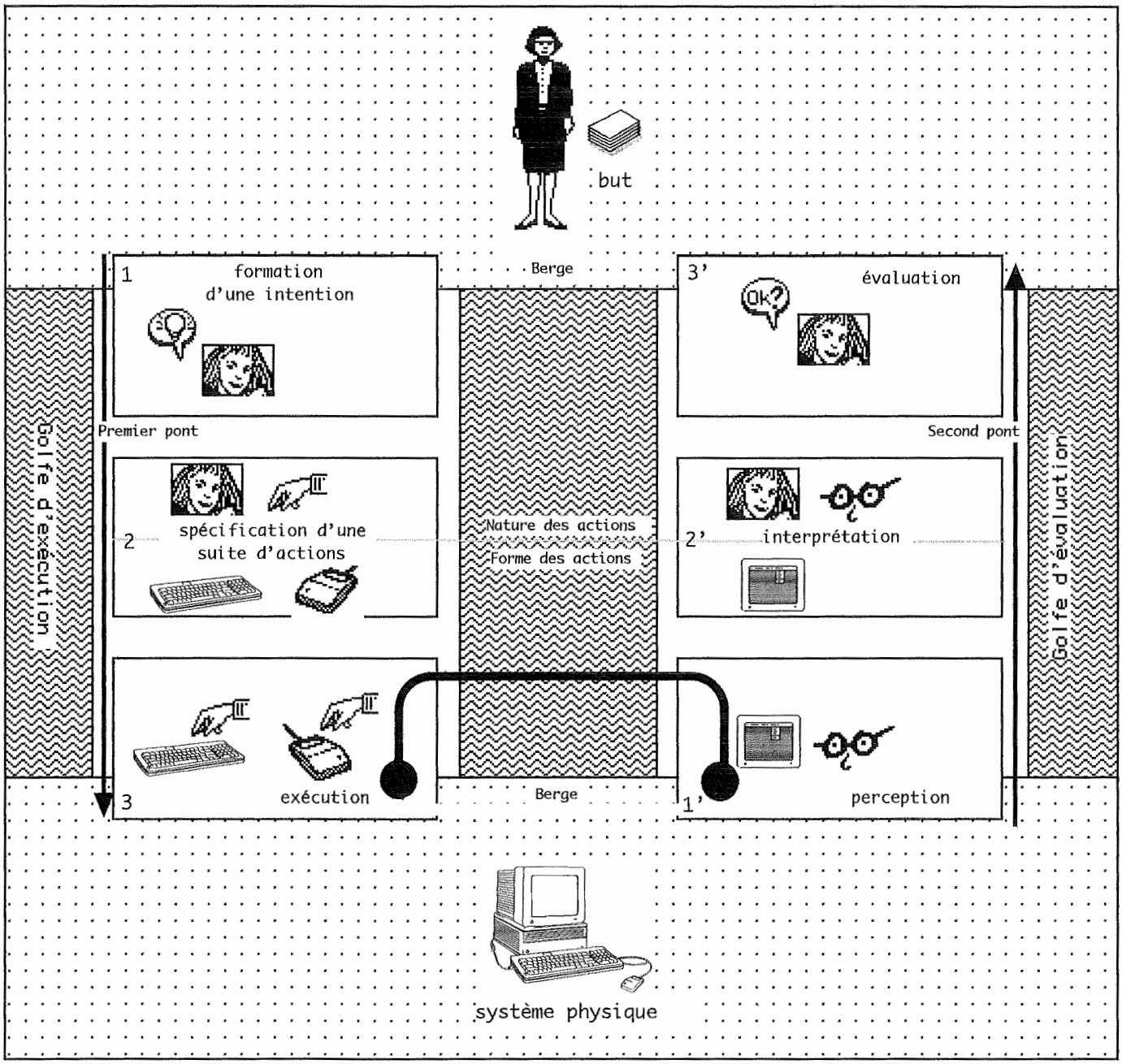


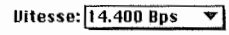
Fig. 8.1 : Le trait épais terminé par deux boules marque le niveau lexical sur le schéma du modèle de Norman

Alors que nous programmions les objets de dialogue du réglage dans son ensemble, nous avons créé un nouvel objet interactif pour que l'utilisateur puisse régler la vitesse du raccordement parce que celle-ci s'exprime tantôt en bits par seconde, tantôt sous le couvert d'une norme du CCITT telle que V21, V22, etc (cfr. p24, en bas). Il s'agit de l'objet interactif concret : vitesse.

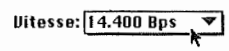
8.1. L'OBJET INTERACTIF CONCRET: VITESSE

Comme nous l'avons présenté plus haut (cfr p24 en bas), la vitesse est tantôt dénommée en bits par seconde (Bps), tantôt elle est couverte par une norme du CCITT (ou de Bell). Par exemple dans son *Manuel d'utilisation du Homebanking BBL*, la BBL parle de «vitesse» (pour les Bps) et de «modem» (pour la norme du CCITT), alors que dans une lettre de mise à jour, elle ne parle plus que de «type de modem» (pour la norme du CCITT). Dans un premier temps, pour régler la vitesse, nous proposions à l'utilisateur (averti) une liste déroulante reprenant la vitesse de différentes façons (en Bps, avec norme, ou les deux). Mais : (1) il ne sait pas si les Bps indiquent la vitesse entre modems ou entre son ordinateur et son modem; (2) bien qu'il est plus fréquent de parler en Bps, on ne le fait pas dans tous les cas; (3) plus rarement, il ne souhaite pas toujours se servir de la vitesse en Bps la plus rapide pour une norme donnée, à condition que le modem le permette aussi (par exemple, choisir le V32 mais à 4.800 Bps au lieu de 9.600 Bps). Prenons l'exemple où la vitesse est reprise par un objet interactif abstrait «Liste déroulante», il saisit la valeur de 1.200 Bps, au lieu de 14.400 Bps par interaction avec l'implémentation concrète de l'objet. Le comportement de l'objet relève du niveau syntaxique et son esthétique du niveau lexical :

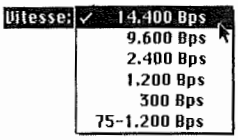
- Affichage : représentation graphique de l'objet «Liste déroulante», spécialisée pour la «vitesse».



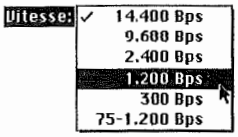
- Entrée de l'utilisateur : il pointe un point quelconque à l'intérieur du cadre avec la souris.
- Retour du système, affichage : le curseur pour pointer avec la souris est visible à l'intérieur du cadre de l'objet.



- Entrée de l'utilisateur : il maintient le bouton de la souris enfoncé.
- Retour du système, affichage : apparition du menu en sur-affichage (encore appelé menu surgissant dans [Mei91]) correspondant à l'objet. L'item sélectionné est marqué par un √. Il est mis en évidence comme l'est le label de l'objet.

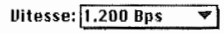


- Entrée de l'utilisateur : il déplace la souris pour pointer un item du menu.
- Retour du système, affichage : l'item pointé est mis en évidence.

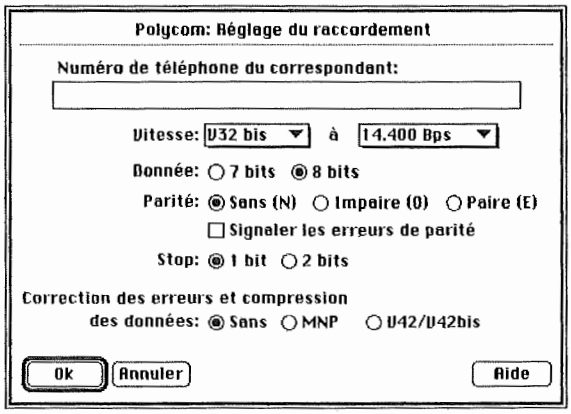


- Entrée de l'utilisateur: il relache la pression sur le bouton de la souris. Dans l'exemple, le choix de 1.200 Bps est éffectué.

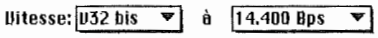
•Retour du système, affichage: Le nouvel item sélectionné prend la place de l'ancien. Le label de l'objet n'est plus mis en évidence.



Dans un deuxième temps, pour répondre aux remarques (2) et (3) de l'utilisateur, nous avons élaboré un nouvel objet interactif abstrait et sa traduction concrète. Cet objet accouple deux «Listes déroulantes», l'une reprenant les normes et l'autre la vitesse en Bps. Il s'agit de laisser l'utilisateur choisir librement une vitesse pour une norme en indiquant d'abord la norme puis la vitesse. Les deux listes sont interdépendantes, c'est à dire que pour une norme donnée, seule une gamme limitée de vitesse est disponible. Ce objet est disposé dans la boîte de dialogue suivante:

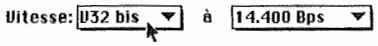


•Affichage : représentation graphique de l'objet «Listes déroulantes couplées» spécialisé pour la «vitesse».



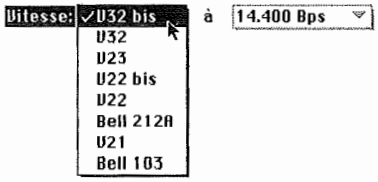
•Entrée de l'utilisateur : il pointe un point quelconque à l'intérieur du premier cadre avec la souris.

•Retour du système, affichage : le curseur pour pointer avec la souris est visible à l'intérieur de ce cadre de l'objet.

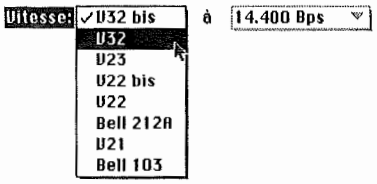


•Entrée de l'utilisateur : il maintient le bouton de la souris enfoncé.

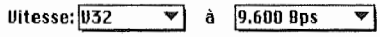
•Retour du système, affichage : apparition du menu des normes en sur-affichage, le cadre du menu des vitesses en Bps est mis en arrière plan (cette notion est du niveau sémantique), c'est-à-dire en grisé (cette notion est du niveau syntaxique), c'est-à-dire un point allumé sur deux (cette notion est du niveau lexical) pour marquer que toute modification de norme entraîne l'ajustement de la vitesse en Bps. L'item sélectionné est marqué par un √. Le label de l'objet et l'item courant sont mis en évidence.



•Entrée de l'utilisateur : il déplace la souris pour pointer l'item «V32» du menu.
•Retour du système, affichage : l'item pointé est mis en évidence.

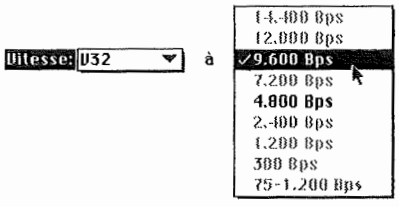


•Entrée de l'utilisateur: il relache la pression sur le bouton de la souris. Dans l'exemple, le choix de «V32» est éffectué.
•Retour du système, affichage: Le nouvel item sélectionné prend la place de l'ancien. Le label de l'objet n'est plus mis en évidence. La vitesse en Bps est ajustée pour refléter la vitesse la plus rapide possible en «V32».

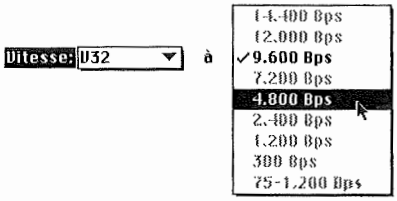


•Entrée de l'utilisateur : il pointe un point quelconque à l'intérieur du second cadre avec la souris.
•Retour du système, affichage : le curseur pour pointer avec la souris est visible à l'intérieur du cadre de l'objet.

•Entrée de l'utilisateur : il maintient le bouton de la souris enfoncé.
•Retour du système, affichage : apparition du menu des vitesse en sur-affichage, le cadre du menu des normes ne change pas parce que une modification de la vitesse en Bps n'entraîne pas d'ajustement de la norme. L'item sélectionné est marqué par un √. Les vitesses que l'utilisateur ne peut pas sélectionner sont mises en grisé. Le label de l'objet et l'item courant sont mis en évidence.

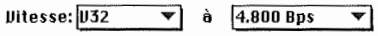


•Entrée de l'utilisateur : il déplace la souris pour pointer l'item «4.800 Bps» du menu.
•Retour du système, affichage : l'item pointé est mis en évidence.



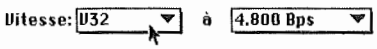
•Entrée de l'utilisateur: il relache la pression sur le bouton de la souris. Dans l'exemple, le choix de «4.800 Bps» est éffectué.

•Retour du système, affichage: Le nouvel item sélectionné prend la place de l'ancien. Le label de l'objet n'est plus mis en évidence.



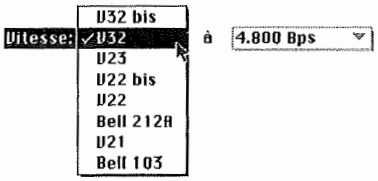
•Entrée de l'utilisateur : il pointe un point quelconque à l'intérieur du premier cadre avec la souris.

•Retour du système, affichage : le curseur pour pointer avec la souris est visible à l'intérieur du cadre de l'objet.



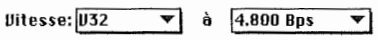
•Entrée de l'utilisateur : il maintient le bouton de la souris enfoncé.

•Retour du système, affichage : apparition du menu des normes en sur-affichage, le cadre du menu des vitesses en Bps est mis en arrière plan . L'item sélectionné est marqué par un √. Le label de l'objet et l'item courant sont mis en évidence.



•Entrée de l'utilisateur: il relache la pression sur le bouton de la souris. Dans l'exemple, le choix de «V32» est conservé.

•Retour du système, affichage: L'item sélectionné est conservé. Le label de l'objet n'est plus mis en évidence. La vitesse en Bps n'est pas ajustée.



Au contact avec les utilisateurs, le comportement de cet objet leur sembla évident tout en leur permettant de régler la norme et la vitesse en Bps.

8.2. LES BOITES DE DIALOGUE DU REGLAGE DANS SON ENSEMBLE

Nous reprenons ci-dessous les différents menus et boîtes de dialogue servant à régler le système, en les accompagnant de remarques, principalement sur le plan lexical. Plus particulièrement, nous citerons deux versions de l'interface que nous rapprocherons avec celle d'Apple. La première de ces deux versions a fait l'objet de tests avec des utilisateurs. C'est ainsi que quatre mois ont été nécessaires pour en arriver au stade où elle en est. Elle a vérifié le point de la spécification des performances d'usage (cfr. p. 11 en bas) suivant : Une fois

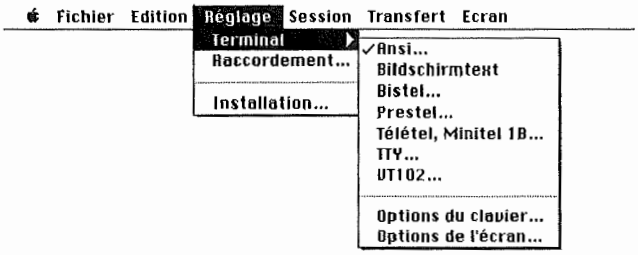
l'équipement installé, les membres de clubs d'informatique, avertis en matière de télécommunication et qui utilisent un environnement de type Windows, devront être capables, sans formation préalable, d'effectuer un appel sur le serveur BBL Home Banking en maximum 15 minutes, en n'utilisant que les supports d'apprentissage fournis. La seconde des deux versions n'a pas encore été confrontée à l'utilisateur. Par rapport à la première version de l'interface, cette seconde version nous permettra de comparer l'attitude de l'utilisateur face à une autre interface encore plus basée sur le graphisme. Tout comme cela a été le cas pour la première version, nous estimons que la seconde version évoluera au contact de l'utilisateur avant donner des résultats probants. L'architecture et l'implémentation retenues nous permettent de passer facilement d'une version à l'autre de l'interface sans remise en cause des acquis.

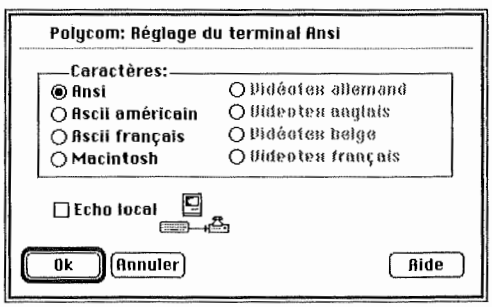
Le résultat de la première version

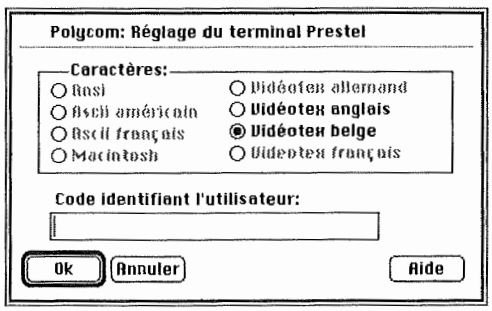
Comme les exemples le montrent, la première version établi un dialogue quasi exclusivement textuel avec l'utilisateur. Son style d'interaction est basé sur la métaphore du dialogue.

•Régler le terminal

Cette version du dialogue correspond à la description du «réglage du terminal» présentée au niveau syntaxique. Ce dialogue est destiné à régler Polycom pour lui faire gérer correctement l'ordinateur en temps que terminal. L'utilisateur sélectionne l'item «terminal» du menu «réglage», avant de sélectionner l'item correspondant au terminal à émuler. Le dialogue se poursuit au moyen de la boîte de dialogue «réglage du terminal ».







Un icone symbolise le type d'écho pour montrer de quoi il s'agit. Elle reprend un ordinateur, un clavier, un modem et, une flèche qui va du clavier vers le modem. Si l'écho local est sélectionné, une autre flèche va aussi du clavier vers l'ordinateur. L'objet interactif abstrait

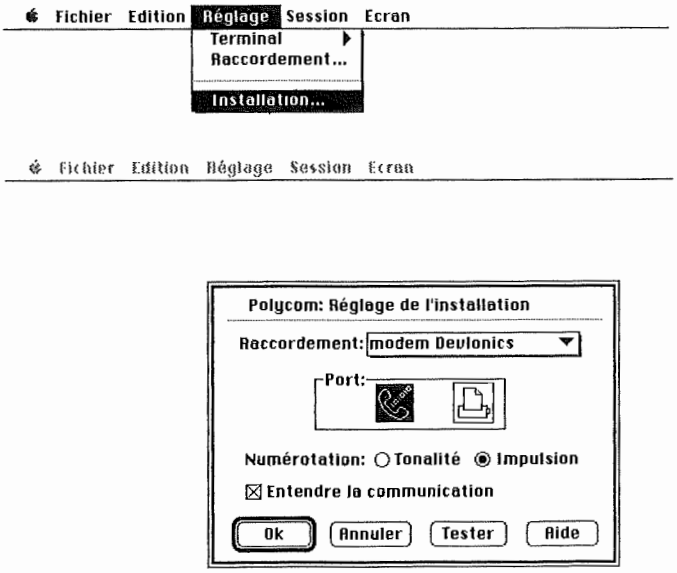
«dm_réglage_terminal» est éclaté en plusieurs boîtes de dialogue afin que l'utilisateur soit uniquement confronté aux paramètres nécessaires à une émulation donnée. Avant cet éclatement, les utilisateurs passaient au dessus de certains réglages actifs mélés à des réglages inactifs.

Un icone devrait symboliser un écran comprenant des lignes avec et sans saut automatique à la ligne, toujours pour montrer de quoi il s'agit.

Voici un exemple de quelques règles extraites de [Van92], dont nous avons tenu compte. "Les menus sont garnis de séparateurs (traits discontinus chez Apple et continus chez IBM) pour marquer des groupes d'items sémantiquement différents (ex.: «terminal» et «raccordement» sont séparés de «installation»). Les items de menu sont combinés pour accomplir entièrement une tâche (ex.: les items du sous-menu «réglage/terminal»). Les dénominations d'items ne doivent pas être des mots techniques appartenant au jargon informatique, elles doivent être des mots courants de la langue maternelle (ex.: les items du menu «réglage»). Les dénominations des items peuvent être empruntées au jargon de l'utilisateur pour autent que cela se justifie et que ces dénominations soient connues de tous les utilisateurs (ciblés) (ex.: le premier groupe d'items du sous-menu «réglage/terminal»). Les dénominations doivent être ortographiées complètement. Les dénominations des items doivent être écrites avec une construction grammaticale homogène, concise et cohérente (...) Les items de menus verticaux doivent être alignés à gauche. Seule l'initiale des items de menus peuvent être écrits en majuscules, excepté les mnémoniques (ex.: «Ansi»; «TTY» pour télétype) etc..."

•*Configurer l'installation*

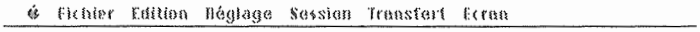
Ce dialogue permet de régler Polycom pour qu'il puisse correctement adresser l'équipement de l'utilisateur. Celui-ci sélectionne l'item «installation» du menu «réglage». Le dialogue se poursuit au moyen de la boîte de dialogue «réglage de l'installation».

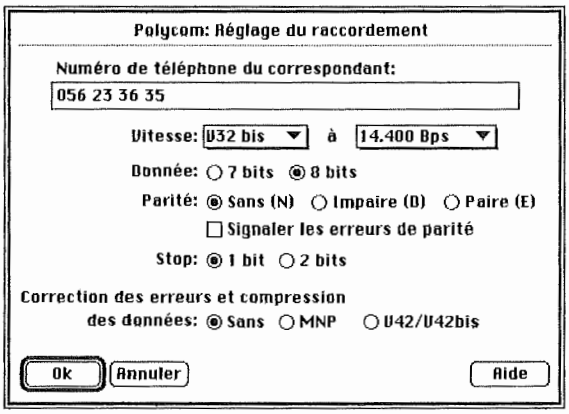


Un icone représentant soit un téléphone avec cadran, soit un téléphone à touche devrait symboliser le type de numérotation pour montrer de quoi il est question.

•*Régler le raccordement*

Ce dialogue permet de régler Polycom pour que le modem soit correctement configuré. L'utilisateur sélectionne l'item «raccordement» du menu «réglage». Le dialogue se poursuit au travers de la boîte de dialogue «réglage du raccordement».

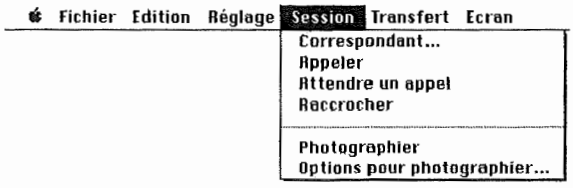


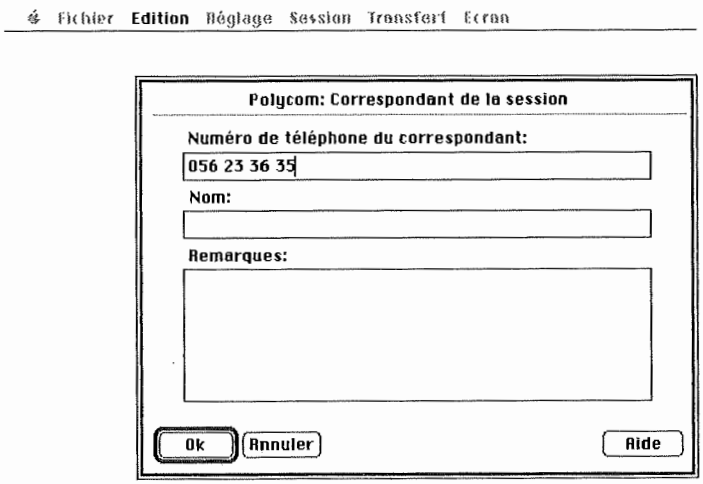


Un icone devrait représenter deux connexions. L'une entre l'ordinateur de l'utilisateur et son modem. L'autre entre les deux modems des correspondants. La vitesse des connexions serait indiquée, pour montrer de quelle vitesse il est question.

•*Indiquer le correspondant*

Avec ce dialogue, l'utilisateur indique le correspondant de la session. Quand l'utilisateur sélectionne l'item «correspondant» du menu «session», le dialogue s'effectue au travers de la boîte de dialogue «correspondant de la session». Les essais sur prototypes nous ont montré que les utilisateurs associent le numéro de téléphone du serveur avec «correspondant» plutôt que «raccordement» et, «raccordement» est préféré à «connexion». En réponse à la question que nous nous posions à la page 60, à propos de la nécessité d'un annuaire, l'observation de l'utilsateur non passioné aux commandes d'autres logiciels de télécommunication nous montre qu'il est plus naturel que les correspondants soient regroupés au sein d'un annuaire, ce qui n'est pas le cas avec cette version-ci. A terme, les correspondants devront être regroupés au sein d'un annuaire téléphonique, même si nous n'avons pas encore de proposition concrète sur ce sujet.





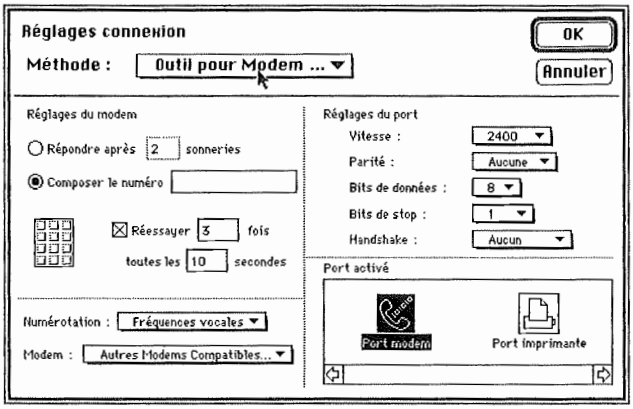
Le numéro de téléphone du correspondant est repris deux fois. Une fois dans ce dialogue-ci et une autre fois dans le dialogue du réglage du raccordement. La raison en est que l'utilisateur associe le numéro de téléphone avec le correspondant et qu'il a besoin de ce numéro pour régler le raccordement. L'emplacement à l'écran de cet objet interactif reste identique pour ces deux dialogues et son label a été rallongé pour bien marquer sa nature. Cela vérifie la règle suivante [Van92 p45] : "Les libellés identificatifs doivent être cohérents : le même libellé doit toujours être employé pour identifier la même donnée."

La version d'Apple

Nous présentons ci-dessous les boîtes de dialogue de la «communication toolbox» réalisée par Apple, à l'exception de celle du réglage du transfert de fichier qui n'intéresse pas Devlonics. La raison de ce désintérêt est que Devlonics souhaite proposer un minimum de réglages à l'utilisateur.

•*Réglage de la connexion*

Comme nous l'avons mentionné plus haut (cfr. p.48), Apple offre une vue fonctionnelle du système de télécommunication par modem au travers de l'interface qui concerne le réglage dans son ensemble. Sous le couvert des réglages de la connexion, Apple a réuni sous un même toit trois dialogues que, dans nos prototypes, nous avons été ammené à traiter séparément. Ces trois dialogues sont celui de la configuration de l'installation, du réglage du raccordement et de l'indication du correspondant.

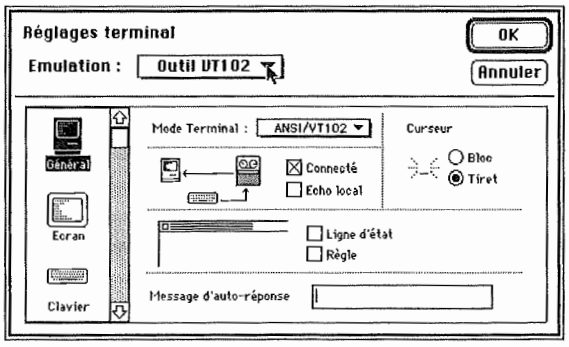


Remarquons qu'un dilemme assez caractéristique du monopole de Belgacom (RTT) apparaît dans la boîte de dialogue "Réglages connexion". En effet, "Réessayer 3 fois toutes les 10 secondes" est divergent avec l'obligation d'utiliser des modems agréés par la RTT sur le réseau téléphonique national. L'utilisateur sera induit en erreur alors qu'il donne cet ordre parce qu'une des limitations imposées pour qu'un modem soit agréé en Belgique est la suivante:

> Si à l'appel, la ligne est occupée alors un rappel est autorisé après 2 secondes. Ensuite, le second rappel et les suivants sont autorisés avec un intervalle de 1 minute, si la ligne est chaque fois occupée. Dans l'heure, il ne peut y avoir plus de 5 appels du même numéro occupé. Ces conditions sont implémentées logiquement dans la ROM du modem qui sera agréé et il n'est pas possible de les contourner. "Il s'agit donc d'une caractéristique assez énervante (parce que trop stricte) imposée aux possesseurs de tels modems, en particulier lorsqu'ils essayent de se connecter à un serveur fréquenté." [Lec91].

•Réglage du terminal

On observe que le réglage du terminal de Polycom est une version limitée par rapport à Apple. A nouveau, cela marque l'objectif de Devlonics qui consiste à minimiser le nombre de réglages disponnibles.



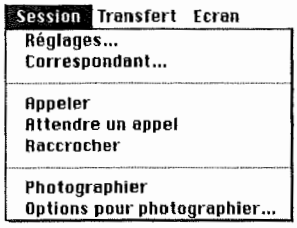
La deuxième version de l'interface

Cette deuxième version de l'interface n'est pas encore passée dans les mains des utilisateurs. Tout ce que nous proposons ci-dessous n'est là qu'à titre de point de départ. Nous procédons par ajustements progressifs. Nous souhaitons trouver un moyen pour baser le style de l'interaction sur la métaphore du mini-monde plutôt que sur la métaphore du dialogue, qui correspond à la première version de l'interface. L'idée est qu'une boîte de dialogue non modale reprenne le schéma reconnu des utilisateurs comme réprésentatif d'un système de télécommunication par modem et que l'utilisateur interagisse avec les éléments qu'il voit pour les régler. Dans la première version l'utilisateur était impliqué dans un processus de rappel indicé. C'est-à dire que pour retrouver un réglage bien précis, il devait activer le bon item (textuel) du menu «réglage» qui sert d'indice. Avec la seconde version, nous tentons de le placer dans un processus de reconnaissance. C'est-à dire que pour retrouver un réglage bien précis, il devra reconnaître l'élément du schéma dans lequel il intervient. Est-ce que ce schéma devra symboliser la vision opératoire ou, est-ce que nous pourrons y introduire une vision fonctionnelle?

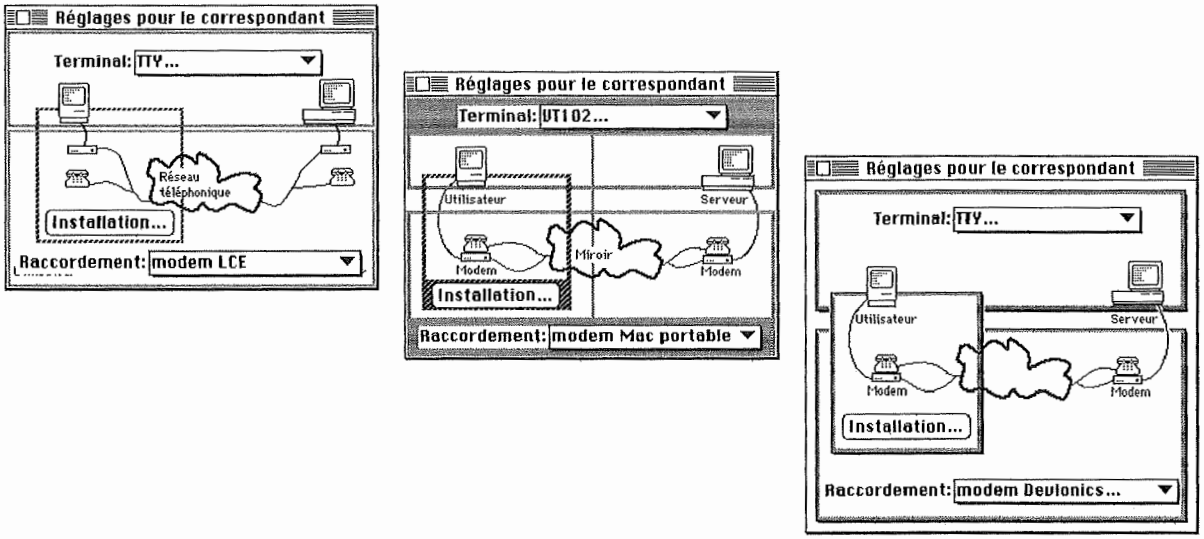
Comme dit Tog [Tog92 p.126] : "People don't want the most abstract interface. They want multiple channels of information. They want neither just words nor just pictures. They want both. The more visual, verbal, vocal, tactile the interface is, the more natural it feels, the more feedback and response it provides, the more confident the user becomes."

•*Les réglages*

Les réglages se font à partir d'une boîte de dialogue non modale à l'exception de l'indication du correspondant que nous laissons de côté. Le menu des régalges disparaît au profit d'un item «réglage» dans le menu «session» que l'utilisateur a l'habitude de regarder en premier lieu lorsqu'il appelle un serveur.



L'image reprise dans la boîte de dialogue non modale représente le système de télécommunication par modem tel qu'il est reconnu par les utilisateurs. Il s'agit de représenter sur cette image, trois éléments qui sont à régler : le terminal, l'installation et le raccordement. La version en bas à droite est celle que nous avons progressivement obtenue. On y distingue plus aisément la présence des trois éléments à régler par rapport aux essais précédents.



De façon assez surprenante, ce dessin est une schématisation du modèle fonctionnel! Nous avons montré que la vision intuitive des utilisateurs est opérationnelle et non pas fonctionnelle. Mais nous avons aussi montré que si on explique de façon fonctionnelle le pourquoi des réglages à un utilisateur, celui-ci comprend mieux le rôle des réglages.

Auparavant, avec les items «terminal», «raccordement» et «installation», nous étions dans un processus de rappel indicé, c'est-à-dire qu'à partir de mots-indices l'utilisateur devait se rappeler où situer quel réglage. Nous pensons qu'en présentant ce schéma, l'utilisateur se placera dans un processus de reconnaissance, c'est-à-dire qu'il n'a plus qu'à situer les réglages sur le schéma qu'il reconnaît.

Il ne nous reste plus qu'à vérifier cette proposition de dialogue, au moyen de tests.

- ***Le terminal***

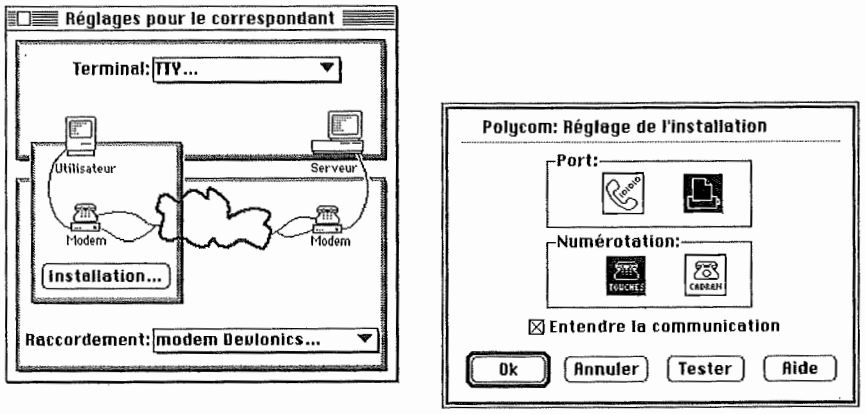
Avec le réglage du terminal, nous présentons une liste déroulante dont les items se terminent par «...». Qu'est que cela veut dire [Tog92 pp46-47]?

Nous avons placé ces trois petits points pour indiquer que la sélection d'un item de cette liste donnera lieu à la poursuite du dialogue au travers d'une boîte de dialogue. Nous conservons

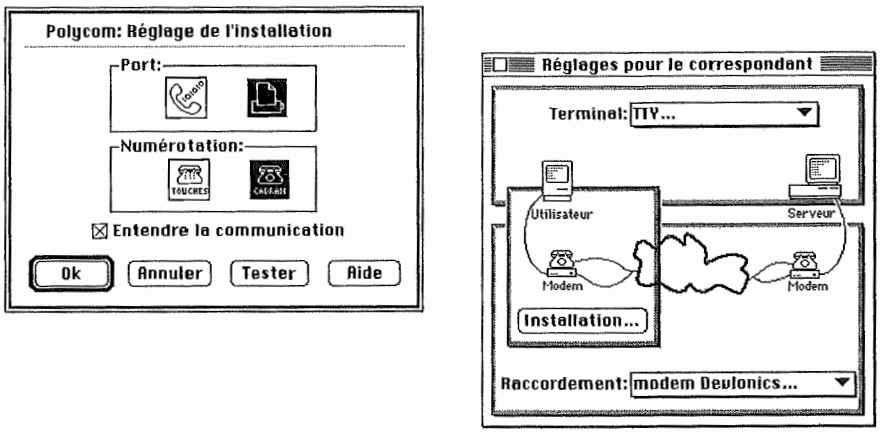
les boîtes de dialogue du réglage du terminal telles quelles ont été consolidées avec la première version de l'interface. Pour en avoir un aperçu, nous nous référons à la description de cette version-là.

•*L'installation*

Pour modifier l'installation, l'utilisateur clique sur le bouton «installation» de la boîte de dialogue des réglages pour le correspondant. Sur l'image qui représente le système de télécommunication par modem, on observe des téléphones à touches parce que l'installation est réglée pour prendre en compte un téléphone à touche (c'est-à-dire que la numérotation se fait par tonalité).



Si l'utilisateur demande de régler l'installation sur un téléphone à cadran (c'est-à-dire que la numérotation se fait par impulsion), alors sont choix se réflète aussi dans la boîte de dialogue des réglages du correspondant.

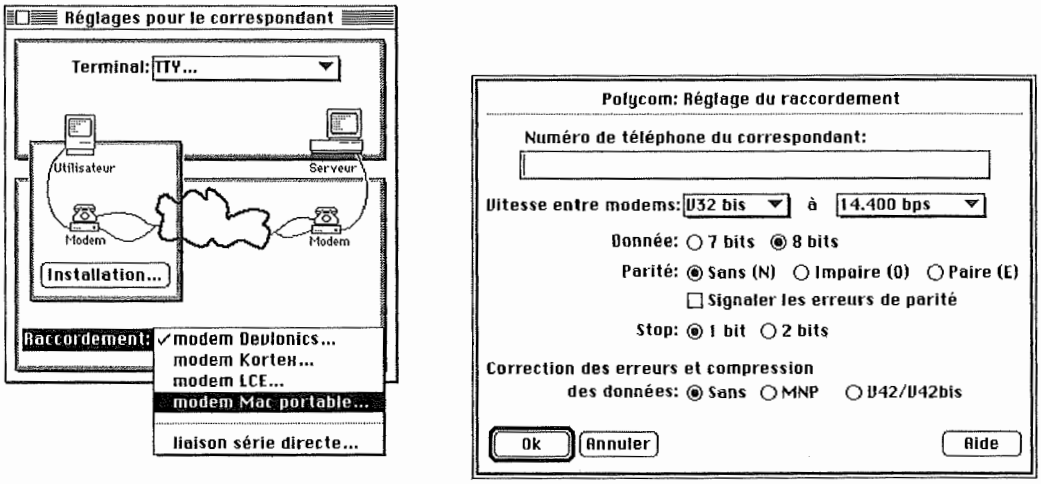


Le choix du type de modem ne se fait plus dans le cadre du réglage de l'installation mais directement au moyen de la liste déroulante intitulée «raccordement», dans la boîte de dialogue des réglages pour le correspondant.

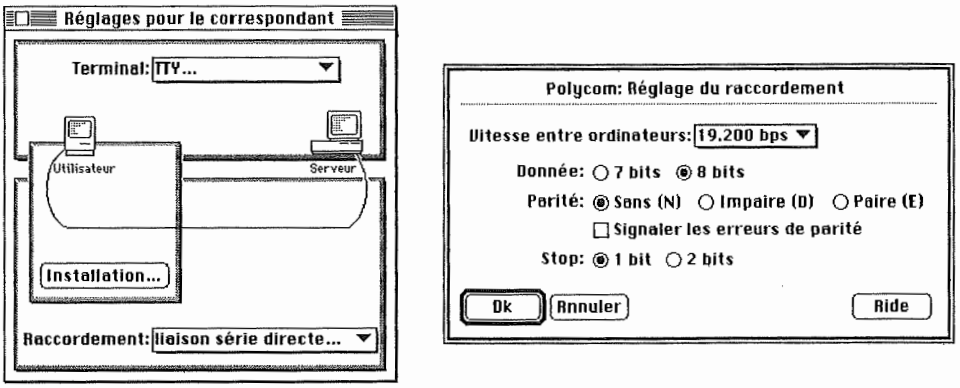
•*Le raccordement*

Dans la première version, le choix du raccordement se faisait au travers de l'installation. A présent, l'utilisateur choisit le raccordement au moment de son réglage. Cette modification a

des répercussions, particulièrement au niveau sémantique où nous devons reconsidérer les phases «configurer l'installation» et «pouvoir appeler un serveur». Une fois de plus, nous procédons par ajustements successifs. Le comportement de l'utilisateur nous permettra de valider ou d'invalider cette approche.



Nous en profitons pour introduire un raccordement que nous appelons la liaison série directe (cfr. p39 en haut). Comme on le voit sur l'image reprise dans la boîte de dialogue des réglages pour le correspondant, il s'agit d'une liaison qui se passe du modem. Dans le jargon technique, le câble qui relie les deux ordinateurs est appelé un câble «null modem». Les réglages du raccordement sont quelque peu différents que lors de l'emploi d'un modem. En ce qui concerne la vitesse, le label précise bien qu'il s'agit de la vitesse entre ordinateurs plutôt que de la vitesse entre modems.



Ces derniers dialogues constituent le point de départ d'une seconde version. Si nous avons introduit cette seconde version, c'est pour aborder le problème du réglage du système sous un autre angle, sous un autre regard, sous une autre forme, toujours dans la but de rencontrer l'utilisateur. Par rapport à la première version de l'interface, notre but est de comparer l'attitude de l'utilisateur face à une autre interface encore plus basée sur le graphisme. Tout comme cela a été le cas pour la première version, nous estimons que la seconde version devra évoluer au contact de l'utilisateur avant donner des résultats probants.

9. ARCHITECTURE ET IMPLEMENTATION

L'architecture développée pour Polycom est une architecture orientée objet (OO) basée sur [Sac 9x] qui structure les dialogues au sein d'objets de dialogue. Les objets de dialogue lient les objets de l'application à des objets interactifs. Ces termes sont tirés de [Sac 9x]. Nous décrirons l'implémentation de cette architecture en C++. Nous terminerons ce chapitre par une remarque à propos des règles sur l'ergonomie.

9.1. ARCHITECTURE OO

L'architecture orientée objet (OO) utilisée définit le comportement des objets interactifs abstraits au sein de dialogues avec l'utilisateur, sous la forme d'objets de dialogue. Ce comportement relève du niveau syntaxique.

Comme pour les objets interactifs abstraits, la classification des objets de dialogue a été élaborée en étroite collaboration avec l'utilisateur. Nous lui avons proposé différentes réalisations de base qui ont ensuite évoluées avec son aide. Pour programmer ces prototypes en Think C(++) sous le Système 7 du Macintosh, nous nous sommes servis de fonctions dites «virtual», selon la syntaxe du C++. Brièvement, parce que nous l'expliquerons plus en détail dans peu de temps, ce type de fonction permet au noyau de base de l'application d'informer les seuls objets de dialogue concernés lors de chaque événement reporté par le système d'exploitation.

Ci-dessous, nous développons l'exemple du choix des objets de dialogue pour le réglage du terminal.

Identification des objets de dialogue pour le réglage du terminal

Pour le cas du réglage du terminal, les objets de dialogue identifiés sont (fig. 9.1, page suivante) : l'*application*, la *fenêtre-document terminal*, la *saisie des réglages du terminal* et le menu des *réglages du terminal*. L'*application* est le noyau de base de l'interface. Elle dirige les événements générés par l'utilisateur, vers les autres objets de dialogue. La *fenêtre-document terminal* est représentée par l'objet interactif *fe_terminal*. Le *menu des réglages du terminal* est représenté par l'objet interactif *mr_réglage_terminal*. La *saisie des réglages du terminal*, sert à coordonner l'activité de l'utilisateur dans son interaction avec la boîte de dialogue représentée par *dm_réglage_terminal*.

Schéma de l'architecture pour le réglage du terminal

Le schéma ci-dessous représente l'architecture retenue pour le réglage du terminal.

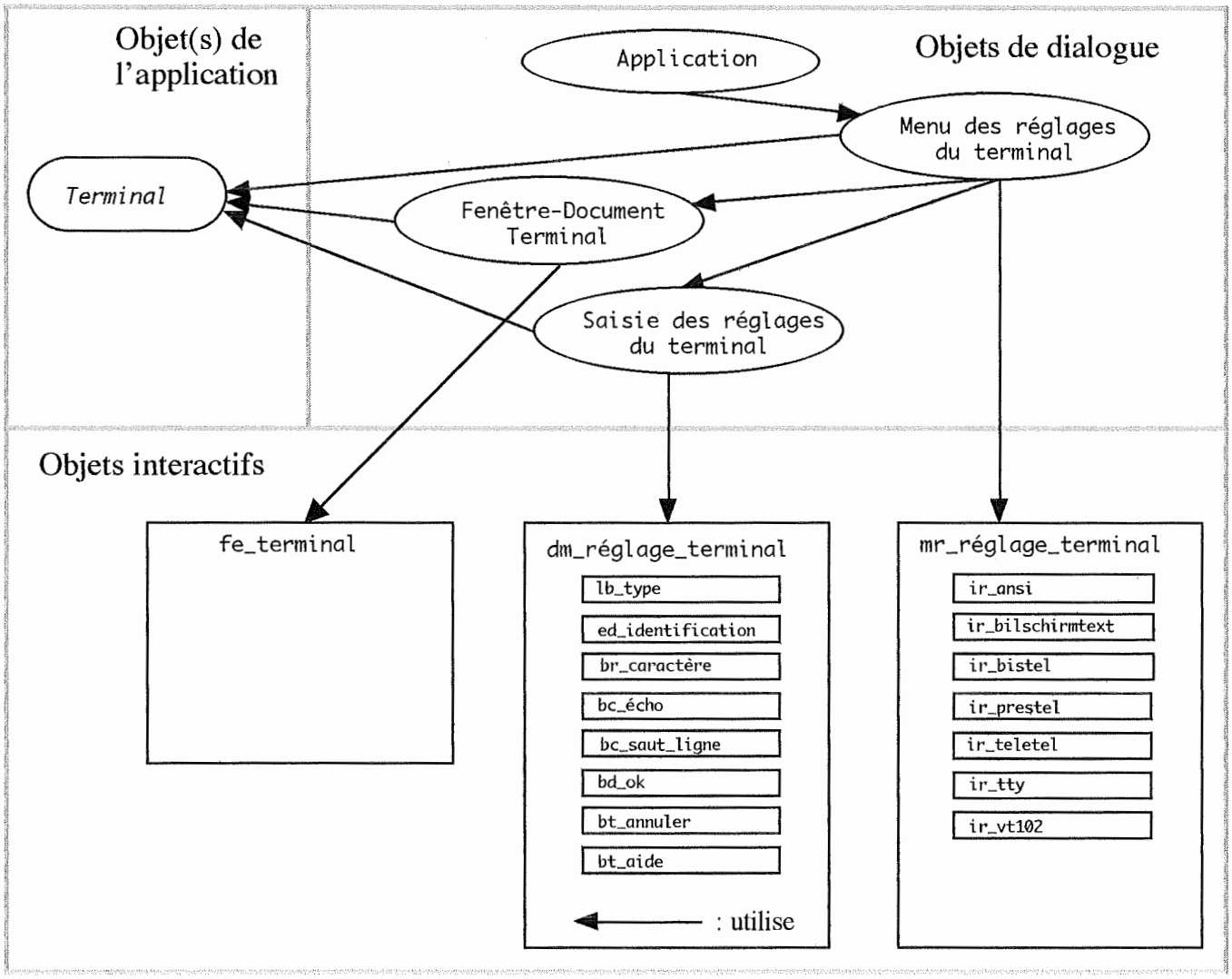


Fig. 9.1 : Schéma de l'architecture concernant le dialogue «réglage du terminal»

Nous supposons que l'utilisateur dialogue avec l'*application* au travers de la *fenêtre-document terminal* qui utilise l'objet de l'application *terminal* et l'objet interactif *fe_terminal*. A la demande de l'utilisateur, l'*application* sollicite le *menu des réglages du terminal* pour lui laisser choisir l'émulation de terminal qu'il souhaite au travers du menu *mr_réglage_terminal*. L'item de *mr_réglage_terminal* correspondant au choix de l'utilisateur sera marqué d'un √. *Fe_terminal* sera adapté en conséquence par la *fenêtre-document terminal*. Dès qu'un item du menu *mr_réglage_terminal* est renseigné, le *menu des réglages du terminal* utilise la *saisie des*

réglages du terminal. Celle-ci utilisera l'objet de l'application *terminal*. Elle entamera un dialogue avec l'utilisateur si un complément d'informations est nécessaire pour affiner l'émulation retenue selon le tableau suivant:

Si le terminal choisi est:	Alors les choix actifs de br_caractères sont (le premier de chaque ligne est sélectionné par défaut):	Alors les autres réglages possibles portent sur:
Ansi	Ansi, Ascii américain, Ascii français, Macintosh	bc_echo
Bildschirmtext	Vidéotex allemand	-
Bistel	Vidéotex belge	ed_identification
Prestel	Vidéotex belge, Vidéotex anglais	ed_identification
Télétel	Vidéotex français	ed_identification
TTY	Ansi, Ascii américain, Ascii français, Macintosh	bc_echo, bc_saut_ligne
VT102	Ascii américain, Ascii français	bc_echo

Ci-dessous, nous détaillons les objets de dialogue mis en évidence pour le réglage du terminal, ainsi que leur comportement.

Objet de dialogue: Menu des réglages du terminal

```
Description: L'utilisateur effectue les différents réglages du Terminal. Pour
       cela, il sélectionne un item de mr_réglage_terminal. Le dialogue se
       poursuivra au moyen de l'objet de dialogue «saisie des réglages du
       terminal» à condition qu'un réglage plus fin soit possible. Finalement la
       «fenêtre-document terminal» sera avertie du choix de l'utilisateur.
Utilise le(s) objet(s) interactif(s) abstrait(s):
       mr_réglage_terminal : Groupement d'items radio du menu «réglage du
              terminal».
Utilise le(s) objet(s) de l'application:
       Terminal : Le terminal est l'ordinateur de l'utilisateur "métamorphosé" en
              terminal Télétel,  terminal Bistel, terminal VT102,…
Utilise le(s) objet(s) de dialogue:
       «Saisie des réglages du terminal» : permet la saisie des réglages du
              terminal.
       «Fenêtre-document terminal» : gère la présentation du terminal à
              l'utilisateur.
Comportement(s):
       L'item de mr_réglage_terminal correspondant au choix de l'utilisateur est
              marqué d'un √.
       Après chaque changement de réglage, générer l'évenement
              «ModifierRéglageTerminal» à destination de la «fenêtre-document
              terminal».
Primitive(s):
       Héritage des primitives de la classe groupe d'item radio d'un menu.
Evénement(s) généré(s):
       vers «Fenêtre-document terminal»:
              ModifierRéglageTerminal : Signale à l'objet de dialogue «fenêtre-
                     document terminal» que les réglages qui la concerne sont
                     modifiés.
```

vers «Saisie des réglages du terminal»:
Ansi : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «Ansi».
Bildschirmtext : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «Bildschirmtext».
Bistel : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «Bistel».
Prestel : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «Prestel».
Télétel : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «Télétel».
TTY : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «TTY».
VT102 : Signale à l'objet de dialogue «saisie des réglages du terminal» que l'utilisateur demande de régler le terminal pour l'émulation «VT102».

Objet de dialogue: Saisie des réglages du terminal

Description: Saisie des réglages du terminal. Les modifications de réglages seront signalées à l'objet de l'application *Terminal*.
Utilise le(s) objet(s) interactif(s) abstrait(s):
dm_réglage_terminal : Boîte de dialogue modale intitulée «réglages du terminal».
Utilise le(s) objet(s) de l'application:
Terminal : Le terminal est l'ordinateur de l'utilisateur "métamorphosé" en Minitel, terminal Bistel, terminal VT102,...
Utilise le(s) objet(s) de dialogue: -.
Comportement(s):
Un changement de réglage porte sur un changement de l'élement sélectionné courant d'un des objets interactifs parmi l'émulation de terminal demandée (au moyen d'une primitive offerte par ce objet-ci : Ansi, Bilschirmtext, Bistel, Prestel, Télétel, TTY, VT102), br_caractère, ed_identification, bc_écho et, bc_saut_ligne.
A chaque changement de réglage, lb_type de dm_réglage_terminal est garni du nom de l'émulation de terminal choisie.
Le tableau suivant est vérifié:

Si le terminal choisi est:	Alors les choix actifs de br_caractères sont (le premier de chaque ligne est sélectionné par défaut):	Alors les autres réglages possibles portent sur:
Ansi	Ansi, Ascii-américain, Ascii-français, Macintosh	bc_echo
Bildschirmtext	Vidéotex allemand	-
Bistel	Vidéotex belge	ed_identification
Prestel	Vidéotex belge, Vidéotex anglais	ed_identification
Télétel	Vidéotex français	ed_identification
TTY	Ascii-américain, Ascii-français, Macintosh	bc_echo, bc_saut_ligne
VT102	Ascii-américain, Ascii-français	bc_echo

Primitive(s):
Héritage des primitives de la classe boîte de dialogue.
Ansi : Effectuer les réglages pour un terminal Ansi, en s'adressant directement aux objets br_caractère, ed_identification, bc_écho et bc_saut_ligne concernés.

```
        Bildschirmtext : Effectuer les réglages pour un terminal Bildschirmtext, en
                s'adressant directement aux objets br_caractère, ed_identification,
                bc_écho et bc_saut_ligne concernés.
        Bistel : Effectuer les réglages pour un terminal Bistel, en s'adressant
                directement aux objets br_caractère, ed_identification, bc_écho et
                bc_saut_ligne concernés.
        Prestel : Effectuer les réglages pour un terminal Prestel, en s'adressant
                directement aux objets br_caractère, ed_identification, bc_écho et
                bc_saut_ligne concernés.
        Télétel : Effectuer les réglages pour un terminal Télétel, en s'adressant
                directement aux objets br_caractère, ed_identification, bc_écho et
                bc_saut_ligne concernés.
        TTY : Effectuer les réglages pour un terminal TTY, en s'adressant
                directement aux objets br_caractère, ed_identification, bc_écho et
                bc_saut_ligne concernés.
        VT102 : Effectuer les réglages pour un terminal VT102, en s'adressant
                directement aux objets br_caractère, ed_identification, bc_écho et
                bc_saut_ligne concernés.
Evénement(s) généré(s): -.
```

Objet de dialogue: Fenêtre-document terminal

```
Description: La fenêtre-document terminal gère la présentation du terminal à
        l'utilisateur.
Utilise le(s) objet(s) interactif(s) abstrait(s):
        fe_terminal : Fenêtre-document fournissant une vue sur le groupement des
                objets Serveur (Correspondant), Raccordement et Ecran (du terminal)
                de l'application.
Utilise le(s) objet(s) de l'application:
        Terminal : Le terminal est l'ordinateur de l'utilisateur "métamorphosé" en
                terminal Télétel,  terminal Bistel, terminal VT102,…
Utilise le(s) objet(s) de dialogue: -.
Comportement(s):
        A chaque modification de réglage du terminal, l'apparence du terminal est
                adaptée pour refléter le nouveaux réglages.

Primitive(s):
        Héritage des primitives de la classe fenêtre.
        ModifierRéglageTerminal : Refléter à l'égard de l'utilisateur, l'état
                courant des réglages du terminal.
Evénement(s) généré(s): -.
```

9.2. IMPLEMENTATION EN C++

L'implémentation des réglages du terminal en Think C(++) sur Macintosh reprend l'architecture orientée objet proposée, à l'exception des menus qui le seront sous peu. Les objets de dialogue font le pont entre les objets de l'application et les objets interactifs. Les classes représentant les objets actuellement implémentés sont repris à la figure 9.2 (page suivante):

- Les (classes d')objets de l'application commencent par les lettres «COa» .
- Les (classes d')objets interactifs commencent par les lettres «COi».
- Les (classes d')objets de dialogue commencent par les lettres «COd», à l'exception de la classe CWindow qui joue un rôle particulier comme nous le montrons par la suite.

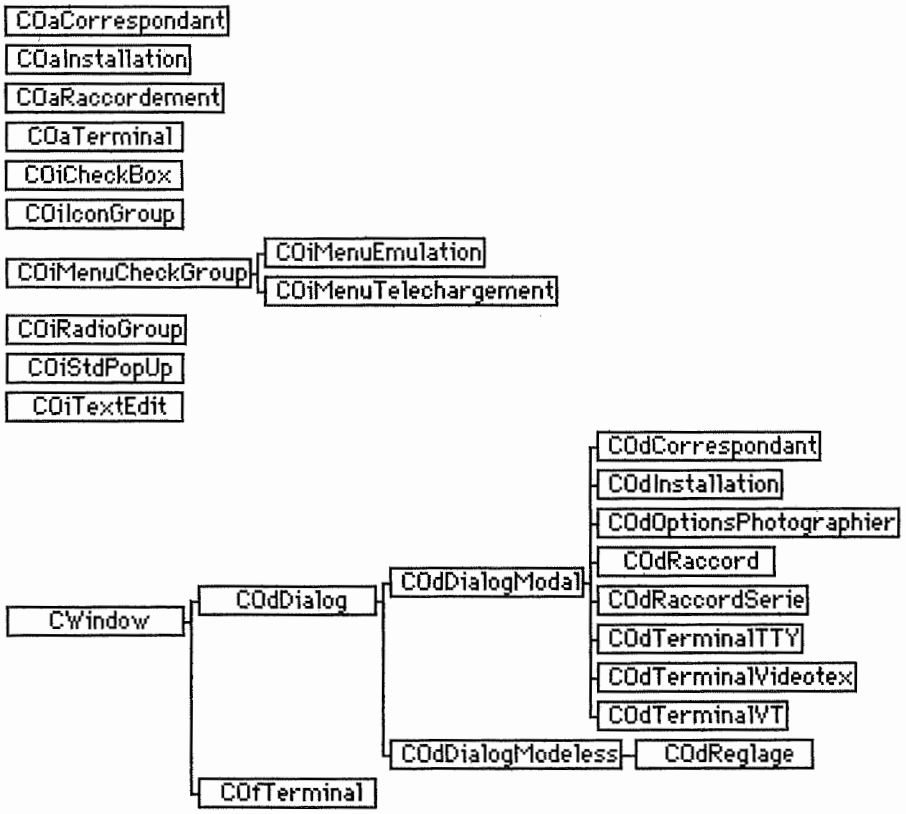
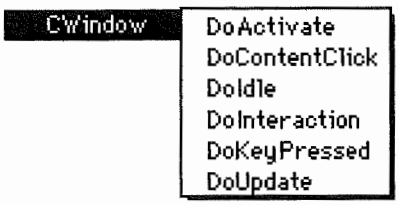


Fig. 9.2 : Les classes d'objets du prototype représentées par le «Class browser» du Think C(++), c'est-à-dire l'outil qui permet d'examiner les classes d'objets

Cette structure s'est révélée être bien adaptée pour implémenter l'interface homme-machine grâce aux opérations/fonctions déclarées «virtual» en C++ dans la classe CWindow. Avec ce type de fonctions, le noyau de base du code source peut distribuer les événements générés par le système d'exploitation aux objets de dialogues concernés, sans que ce code source ne doive être modifié lorsque nous ajoutons un nouvel objet de dialogue. La classe CWindow est une classe racine, qui n'a que pour seul but d'être la racine pour les fonctions «virtual»:




```
class CWindow : indirect {
       public:
              virtual char DoKeyPressed(char Key, short Modifiers) {};
              virtual void DoContentClick(Point Where) {};
              virtual void DoUpdate(void) {};
              virtual void DoActivate(short Activate) {};
              virtual void DoInteraction(short Item) {};
              virtual void DoIdle(void) {};
};
```

Techniquement cela se passe comme suit : Dans le noyau de base du code source, pour chaque événement généré par le système d'exploitation qui nous intéresse, nous faisons appeler la

fonction de l'objet de dialogue qui y répondra en passant par un pointeur sur un objet de type CWindow. Au moyen d'un extrait du code du noyau de base, observons l'exemple de l'événement «mouseDown inContent», qui signale que l'utilisateur a cliqué avec la souris à l'intérieur d'une fenêtre:

```
void MainEventLoop(void)
{ …

 …
 while(!gQuit) {
  gotEvent = WaitNextEvent(everyEvent,&theEvent,0L,cursorRgn);
  if (gotEvent) {
   switch(theEvent.what) {
    case mouseDown:
     theWindowPart = FindWindow(theEvent.where, &theWindow);
     switch(theWindowPart) {
      case inMenuBar:       …
      case inSysWindow: …

      case inContent:
       if(theWindow!=FrontWindow() )
        SelectWindow(theWindow);
       else
        if( IsDialogEvent(&theEvent) ) {
         if ( DialogSelect(&theEvent,&theDialog,&item) )
           DoInteraction(theWindow,item);
        }
        else
          DoContentClick(theWindow,theEvent.where);
       break;

      case inDrag: …
      case inGrow: …
      case inGoAway: …
     }
     break;
   }
  }
 }
}


void DoContentClick(WindowPtr theWindow, Point Where)
{       CWindow *OWindow;

        if(theWindow!=NIL) {
               OWindow = (CWindow *) GetWRefCon(theWindow);
               if(OWindow) OWindow->DoContentClick(Where);
        }
}
```

Au moment de l'exécution de `OWindow->DoContentClick(Where)`, l'appel sera dirigé vers la fonction définie *non pas dans* la classe CWindow *mais bien dans* la classe de l'arborescence menée par CWindow (fig. 9.2), à laquelle l'objet pointé par le pointeur appartient. En l'occurence, ces classes de l'arborescence représentent les objets de dialogue. Ainsi, ce sera la fonction de l'objet de dialogue concerné par l'événement qui sera appelée. Par contre, si la fonction appelée dans de telles conditions n'était pas déclarée «virtual», l'appel serait dirigé d'office vers la fonction telle que définie dans la classe CWindow, ce qui ne répondrait pas à nos attentes.

Pour garnir le pointeur OWindow, nous utilisons une zone qui est contenue dans la structure qui décrit chaque fenêtre et, qui est réservée par Apple à l'intention du programmeur. Cela implique qu'un objet de dialogue est toujours associé à une et une seule fenêtre. Cette zone de quatre octets est garnie lors de la création d'une fenêtre :

```
leDialogue=GetNewDialog( IdDialogue, NULL, (WindowPtr)-1 );
SetWRefCon(leDialogue,(long) this);
```

Avec Windows sur PC, nous n'avons pas pu mettre au point un tel procédé car contrairement à Apple, Microsoft n'a pas réservé de zone à l'intention du programmeur dans la structure qui décrit l'occurence de chaque fenêtre. Pour résoudre ce problème avec leurs compilateurs C++ respectifs, Microsoft et Borland ont établi deux stratégies différentes[6] puisqu'il s'agit ici, d'associer l'identifiant donné par Windows à chaque fenêtre de l'application avec l'adresse d'un objet de dialogue.

9.3. A PROPOS DES REGLES SUR L'ERGONOMIE

Un grand nombre de règles d'ergonomie a un impact particulier dans la mise au point des objets de dialogue. Prenons par exemple, deux règles tirées de [Tog92] : (1) "Do not attempt a 3-D look in one-bit graphics" et (2) "Human-computer interactivity should not be tied to the speed of the system clock". Quelle est la règle qui sera prise en compte au niveau lexical du langage de l'interface et quelle est la règle qui sera prise en compte au moment de la description des objets de dialogue? C'est pour cela, que nous avons décris les objets de dialogues en termes de comportement plutôt qu'en d'autres termes. Un autre exemple tiré de [Van92] : (1) "La position initiale du curseur (en anglais, home position) doit être globalement cohérente" et (2) "Si des curseurs multiples sont utilisés, ils doivent être rendus visuellement distincts les uns des autres."

---

[6] A ce sujet, on lira la rubrique "Questions & Answers, C/C++" de Bob Chiverton dans *Microsoft Systems Journal*, Vol. 7 n°5.

10. EVALUATION PAR PROTOTYPE

L'évaluation par prototype est un indicateur des attentes de l'utilisateur. Tog [Tog92 chapter 14] insiste sur le caractère essentiel et bon marché que peut prendre l'évaluation par prototypes. Il distingue les prototypes horizontaux des prototypes verticaux. Les prototypes horizontaux s'étalent superficiellement sur l'ensemble de l'application. Ils permettent d'avoir une vue d'ensemble. Les prototypes verticaux ciblent en profondeur un élément de l'application. Ils permettent d'avoir une vue détaillée.

Ensuite, Tog propose une méthode en dix étapes qui mettent à profit l'observation de l'utilisateur alors qu'il pense à haute voix :

> "1. Introduce yourself.
> 2. Describe the purpose of the observation (in general terms).
> 3. Tell the participant that it's OK to quit at any time.
> 4. Talk about the equipment in the room.
> 5. Explain how to "think aloud".
> 6. Explain that you will not provide help.
> 7. Describe the tasks and introduce the product.
> 8. Ask if there are any questions before you start; then begin the observation.
> 9. Conclude the observation.
> 10. Use the results."

Dès que nous avons pris connaissance de cette méthode, nous l'avons appliquée. La prise en compte des résultats ainsi obtenus s'est avérée positive. Les deux versions des prototypes présentées au chapitre précédent ont été obtenues progressivement. Nous reprenons ci-dessous les grandes étapes de l'évolution de ces prototypes:

1°) Dans un premier temps, nous nous sommes servi de la feuille de papier pour représenter l'interface. Nous construisions, dessinions des projets horizontaux. Nous les présentions progressivement à un groupe restrain d'utilisateurs.

2°) Nous avons programmé des menus que nous avons ensuite complété par des boîtes de dialogue. Chaque fois nous retournions vers l'utilisateur. L'implémentation fut réalisée en Quick C sous MS-Windows.

3°) Nous sommes passé au Macintosh et à la programmation objet. Le compilateur utilisé est le Think C. Nous nous sommes provisoirement servi de la librairie d'objets livrée avec ce compilateur. Sa mise en œuvre nous a fait perdre beaucoup de temps.

4°) Finalement nous avons utilisé une architecture proche de l'architecture OO proposée dans [Sac9x]. Avec le temps nous maîtrisons progressivement la programmation de l'interface utilisateur sur Macintosh. Les contacts avec les utilisateurs ont été de plus en plus enrichissant. Nous avons appliqué la méthode proposée par Tog sur des prototypes verticaux.

Nous avons été fréquement surpris par l'attitude de l'utilisateur aux commandes de prototypes. Cela a été un enseignement enrichissant. Particulièrement si nous ne reproduisions pas deux scéances d'essais avec un même utilisateur, parce que dans ce cas, à la seconde séance il avait tendance à se focaliser sur la prise en compte de ses remarques à propos des aspects négatifs de la première version.

11. SYNTHESE, APPORTS ET PERSPECTIVES

En l'absence de démarche systématique pour développer une interface homme-machine, nous en avons approchée une, centrée sur l'utilisateur. Nous avons travaillé par expérimentation, avec pour exemple, un logiciel de télécommunication. Les éléments de la démarche adoptée ont été basés sur la modélisation de l'interface homme-machine comme un langage. Nous avons décomposé ce langage de l'interface en différents niveaux d'abstraction, à savoir les quatre niveaux pragmatique, sémantique, syntaxique et lexical.

An niveau pragmatique, nous nous sommes particulièrement basés sur la technique des indices de rappels afin d'introduire des éléments porteurs d'une réalité pour l'utilisateur dans les objets de l'application. Ces objets sont assemblés pour former un modèle pragmatique de l'application. Avec leurs réponses à un questionnaire, les utilisateurs ont présenté une vue opératoire plutôt que fonctionnelle du système de télécommunication.

Au niveau sémantique, nous nous sommes aidés de la grille *opération - conduite - attente* et de la notion d'unité sémantique pour organiser les opérations de l'utilisateur en une séquence de comportements.

Au niveau syntaxique, notre but était de faire correspondre le mieux possible chaque séquence de comportement du niveau sémantique en un groupement d'opérations formelles sur des objets interactifs abstraits.

Au niveau lexical, notre but était de déterminer la forme de chaque échange entre l'homme et la machine au moyen d'objets interactifs concrets qui symbolisent au mieux une réalité pour l'utilisateur.

Les prototypes de l'application ont constitué un outil inestimable. Ils ont été développés au moyen d'une architecture et d'une implémentation orientées objets. L'architecture retenue met en œuvre des objets de dialogue dont le comportement s'exprime par la mise en relation des objets de l'application et des objets interactifs. L'implémentation met en relief le rôle des fonctions déclarées «virtual» en C++.

La démarche adoptée devra être complétée. Plusieurs aspects importants de l'interface n'ont pas été détaillés. Nous ne profiterons pas de ces lacunes pour oublier que l'utilisateur est au centre de nos préoccupations. Nous citons essentiellement :

- La validation approfondie des réalisations par rapport aux règles d'ergonomie, même si nous ne les avons pas négligées.

- La réalisation d'un système d'aide; la possiblité d'annuler et de refaire une opération.

•L'adaptation de l'interface à d'autres utilisateurs au delà de la classification des profils cognitifs proposés. L'interface étudiée est, par exemple, limitée à un profil d'utilisateur valide. L'adaptation de l'interface à différents niveaux d'utilisateurs, comme par exemple ceux proposés par Microsoft [Mic90] : débutant en informatique, débutant dans le logiciel, habitué au logiciel, expert dans le logiciel.

•L'adaptation de l'interface eu égard à une évolution des techniques mises en œuvre dans les systèmes de télécommunication étudiés.

Au-delà des qualités et des manques de notre approche, l'attitude de l'utilisateur nous a fréquemment surpris tout au long de ce travail. Nous retiendrons ceci:

•Sa vue opératoire plutôt que fonctionnelle du système nous a posé bien des questions. Nous en avons néanmoins tenu compte parce que même s'il semble mieux comprendre le système sous un angle fonctionnel, cet angle est une abstraction de la vue intuitive opératoire qu'il conçoit.

•Il positionne la partie visible de l'interface par rapport à lui, sans toujours se préoccuper d'une justification.

•Il se rend rarement compte qu'une image de marque est véhiculée par les environnements de travail des constructeurs de systèmes informatiques.

A l'avenir, nous souhaitons plus particulièrement :

•Mieux situer le lieu du développement de l'architecture et de l'implémentation du logiciel dans la démarche.

•Situer chaque règle et critère d'ergonomie au sein de chaque niveau de la démarche. Peut-on classer l'ensemble des règles et critères d'ergonomies proprement dans chaque niveau du langage de l'interaction?

•Où, comment et à quel moment situer la réalisation d'un système d'aide et la possiblité d'annuler ou de refaire une opération?

•Comment alléger effectivement les difficultés de programmation de l'interface d'un même logiciel sur les divers environnements de travail, toujours en mouvance, des constructeurs de systèmes informatiques?

UTILISATEURS INTERROGES

Nous nous permettons de rappeler ci-dessous la grille de classification des profils cognitifs des utilisateurs, suivie de la liste des utilisateurs interrogés au 23 mai 1992.

	Utilisateur passionné	Utilisateur non-passionné
Utilisateur averti	Il maîtrise les notions techniques du domaine des télécommunications par modem. Il s'implique dans le logiciel. (1)	Il maîtrise les notions techniques du domaine des télécommunications par modem. Il ne s'implique pas dans le logiciel. (2)
Utilisateur non-averti	Il ne maîtrise pas les notions techniques du domaine des télécommunications par modem. Il s'implique dans le logiciel. (4)	Il ne maîtrise pas les notions techniques du domaine des télécommunications par modem. Il ne s'implique pas dans le logiciel. (3)

Classification des profils cognitifs

Nom	Sujet	Lieu	Date	Profil
Philippe Lansens	Fonctionnalités de Polycom	Chez lui	11/91	(1)
Philippe Vandenbulcke	Fonctionnalités de Polycom	Devlonics	11/91	(2)
Alain Raulo	Fonctionnalités de Polycom	Devlonics	9/91 à 1/92	(1)
Yves Stouten	Gestion d'un serveur Prestel	Devlonics	12/91	(1)
Alain Raulo	Prototype	Devlonics	1/92	(1)
Michel Devos	Prototype	Micho	14/3/92	(3)
Damien Mathé	Prototype	Micho	14/3/92	(3)

Liste des utilisateurs interrogés au 23 mai 1992 (début)

Nom	Sujet	Lieu	Date	Profil
Carl Callens	Prototype	Micho	14/3/92	(3)
Philippe Vandenbulcke	Questions et prototype	Micom-80	22/3/92	(2)
?	Questions et prototype	Micom-80	22/3/92	(3)
Nicolas Blockeel	Questions et prototype	Micom-80	22/3/92	(3)
Gilles Duponchelle	Questions	Micho	28/3/92	(3)
?	Questions	Micho	28/3/92	(3)
Carl Callens	Questions	Micho	28/3/92	(3)
Serge Carpentier	Questions	Apple	1/4/82	(3)
Laurent Gerardy	Questions	Apple	1/4/82	(3)
Yves Stouten	Questions et prototype	Micom-80	26/4/92	(1)
Philippe Lansens	Questions et prototype	Chez lui	5/92	(1)
Alain Raulo	Questions	Chez moi	23/5/92	(1)

Liste des utilisateurs interrogés au 23 mai 1992 (fin)

BIBLIOGRAPHIE

A PROPOS DE L'INTERFACE HOMME-MACHINE

[App91a] Apple Computer Inc. (ed.), *Inside Macintosh volume IV*, Addison Wesley, Etats-Unis, 1991.

[App91b] Apple H., "De Macintosh-filosofie", *Praktijkboek : Macintosh Classic*, Kluwer Technische Boeken, Pays-Bas, 1991.

[Bea91] BEAUDOUIN-LAFON M., "Interfaces homme-machine : Vue d'ensemble et perspectives", in *Journ'Almin*, n°21, pp 9-21, FUNDP, Namur, 1991.

[Bor89] BORENSTEIN Ph. (ed), *Think C object-oriented programming manual*, Symantec corporation, Cupertino (U.S.A.), 1989-1991.

[Bod89a] BODART F., "Notes de cours provisoires du Chapitre 1", in *Cours introductif aux interfaces homme-machine*, FUNDP, Namur, 1989.

[Bod89b] BODART F., PIGNEUR Y., *Conception assistée des systèmes d'information*, Masson, Paris, 1989, 2nd éd.

[Cou90a] COUTAZ J., *Interfaces homme-ordinateur (conception et réalisation)*, Bordas (Dunod informatique), (F.), 1990.

[Cou90b] COUTAZ J., "Architecture models for interactive software : failures and trends", *Engineering for human-computer interaction*, G. COCKTON (ed.), Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1990.

[IBM89] IBM (ed.), *Systems application architecture, common user acces : advanced interface design guide*, document SC26-4582-0, International Business Machines, Boca Raton (Etats-Unis), juin 1989. Des documents semblables sont édités par d'autres firmes (Apple, Microsoft) qui veulent développer une image de marque au sein des interfaces qui sont développées à partir de leurs produits.

[Kna92] KNASTER S., ROLLIN K., "Concepts and ideas", *Macintosh programming secrets*, Addison Wesley, (U.S.A.), 2nd ed., 1992.

[Lie80] LIEURY A., *Les procédés mnémotechniques (science ou charlatanisme?)*, Pierre Mardaga, Bruxelles (B.), 1980.

[Lit91] LITTLE G., SWIHART T., "Ballon help", *Programming for System 7*, Addison-Wesley, (U.S.A.), 1991.

[Mat91] MATTHIES K., HOGAN T., "Isolating user interface code", *Macintosh C programming by example*, Microsoft press, Redmond-Washington (U.S.A.), 268-269, 1991.

[Mei91] MEINADIER J. P., *L'interface utilisateur (pour une informatique plus conviviale)*, Dunod (Informatique & stratégie), Paris (F.), 1991.

[Mic90] MICROSOFT (ed.), "Help tools", *Microsoft Windows programming tools for Version 3*, Microsoft press, Redmond-Washington (U.S.A.), Part 4 (Chapters 15-19), 1990.

[Nor83] NORMAN D. A., "Some observations on mental models", *Readings in human-computer interaction, a multidisciplinary approach (1987)*, 1983.

[Pre90] PREECE J., KELLER L., *Human-computer interaction*, Prentice Hall, (U.K.), 1990.

[Pro90] PROVOT I., VANDERDONCKT J., *Les objets interactifs : classification et typologie*, FUNDP, Namur, 1990.

[Sac9x] SACRE B., PROVOT I., *Architectures des applications interactives*, FUNDP, Namur, 199x.

[Sca] SCAPIN L., *Guide ergonomique de conception des interfaces homme-ordinateur*, INRIA (Institut National de Recherche en Informatique et en Automation), France.

[Tog92] TOGNAZZINI B., *Tog on interface*, Addison-Wesley (Apple computer, Inc.), (U.S.A.), 1992.

[Van92] VANDERDONCKT J., *Guide ergonomique de la présentation des applications hautement interactives*, FUNDP, Namur, 1992.

A PROPOS DES TELECOMMUNICATIONS

[Ber86] BERAUDO DE PRALORMO J.-L., TOUZALIN J.-C., *Minitel 1B : Specifications techniques d'utilisation*, Centre commun d'études de télédiffusion et télécommunications, France, 1986.

[Bos86] BOSWELL P., "Xmodem, CRC Xmodem, Wxmodem File Transfer Protocols", in file WXMODEM.DOC, FidoNet (B.B.S.), 1986.

[Dac84] DA CRUZ F., CATCHINGS B., "Kermit : a file transfer protocol for universities, part 1", *Byte*, june, 255-278, 1984.

[Dac84] DA CRUZ F., CATCHINGS B., "Kermit : a file transfer protocol for universities, part 2", *Byte*, july, 143-403, 1984.

[For87a] FORSBERG C., "The XMODEM-YMODEM File Transfer Protocol", in file YMODEM.DOC, Omen Technology Incorporated, Portland Oregon, FidoNet (B.B.S.), 1987.

[For87b] FORSBERG C., "The ZMODEM Inter Application File Transfer Protocol", in file ZMODEM.DOC, Omen Technology Incorporated, Portland Oregon, FidoNet (B.B.S.), 1987.

[Hor91] HORIGHAUSEN J., "EMSI/IEMSI Protocol definitions", in file EMSC-001.001, may, Fidonet (B.B.S.), 1991.

[Lec91] LECLERCQ J., "Télécommunications", in file TELECOM.DOC, décembre, ULTRA BYTE (067) 56 17 39 (B.B.S.), 1991.

[Mer92] MERY D., "Les protocoles X, Y et ZMODEM", *PC Expert*, mars, 279-284, 1992.

[Rin90] RINCE J.-Y., *Le Minitel*, Presses universitaires de France (Que sais-je?), Paris (F.), 1990.

[Shu84] SHUFORD R., "ANSI Standard (X3.64) Control Sequences for Video Terminals and Peripherals in alphabetic order by mnemonic", in file ANSIX364.TXT, Fidonet (B.B.S.), 1984?.

A PROPOS DE LOGICIELS DE TELECOMMUNICATION

[Byr91] BYRD M., "Asynchronous communications software : Improving information access", *PC Magazine*, april 30, 101-198, 1991.

[Dan90] DANULOFF C., McCLELLAND D., "Conecting to a bulletin board & Downloading a file", *Encyclopedia Macintosh*, SYBEX, 1990.

[Dar88] DARGERY Y., "L'ordinateur au téléphone", in *Science & vie micro*, 54, 88-96, 1988.

[Dem91] DEMOLY J., "Mettez un Minitel dans votre micro", in *Science & vie micro*, mai, 106-116, 1991.

[Mee87] MEEKS B., "Communications : a dozen top performers with a variety of features", in *Byte bonus edition*, summer 1987, 91-99, 1987.

[Ric91] RICHELMUNN B. et G. LIST et J. RICHTER, *Communication Modems Télématique*, Micro application, Paris (F.), 1991.

[RouX] ROUX I., "ZZ-COM et EMULCOM pour Atari ST", in *Science & vie micro*, 59, 115-116.

[San91] SANZ D., "Comment faire pour gérer son portefeuille boursier", in *Science & vie micro*, mai, 156-159, 1991.

[Sto86] STONE D., "Asynchronous communications : Shopping for software", *PC Magazine*, october 28, 126-201, 1986.

[Stu91] STUUR A., "Nieuwe Procomm Plus kent vele verfijningen", in *Software magazine*, 6, 47-48, 1991.

TABLE DES MATIERES

	Résumé, abstract	ii
	Sommaire	iii
	Remerciements	iv
1.	Introduction	1
2.	Système de télécommunication par modem	4
3.	Les acteurs	6
	3.1. Analyse des tâches	6
	3.2. L'organisation Devlonics	7
	3.3. L'utilisateur	8
	Profils cognitifs	8
	Grille d'interfaces	10
	Spécification des performances d'usage	11
	3.4. Le technicien	12
	3.5. Le vendeur	13
4.	Méthodologie	15
	4.1. Modélisation de l'activité de l'utilisateur	15
	4.2. Notion du "plus naturellement possible"	17
	4.3. Méthodologie	19
5.	Niveau pragmatique	21
	5.1. Modules de l'application	23
	Description de l'environnement par l'utilisateur	30
	Montrer le rôle des réglages	32
	L'invalidation de la classification	32
	5.2. Objets de l'application	34
	Test de prototypes	34
	Les indices de rappels	35
	5.3. Première tâche: Régler son système pour communiquer avec un centre serveur	36
	5.4. Deuxième tâche: Appeler le centre serveur	39
	5.5. Troisième tâche: Archiver des écrans reçus	41
	5.6. Quatrième tâche: Echanger des fichiers	45
	5.7. Consolidation de la structure du modèle	47
	Un autre modèle est-il préférable?	47

6. Niveau sémantique 51
6.1. Tâche : Régler son système 54
Phase : Configurer l'installation 54
Phase : Pouvoir appeler le serveur 55
6.2. Tâche : Appeler le centre serveur 56
Phase : Etablir la communication 56
Phase : Terminer la communication 57
6.3. Tâche : Archiver des écrans reçus 57
Phase : Photographier et insérer la photo dans un album 57
6.4. Tâche liée à l'environnement de travail de l'utilisateur 57
Phase : Sauvegarder les réglages propres à un serveur 57
7. Niveau syntaxique 58
7.1. Les moyens d'interactions 59
7.2. Les styles d'interactions 60
7.3. Identification des objets interactifs abstraits 63
8. Niveau lexical 65
8.1. L'objet interactif concret: vitesse 67
8.2. Les boîtes de dialogue du réglage dans son ensemble 70
Le résultat de la première version 71
La version d'Apple 74
La deuxième version de l'interface 76
9. Architecture et implémentation 80
9.1. Architecture OO 80
Identification des objets de dialogue pour le réglage du terminal 80
Schéma de l'architecture pour le réglage du terminal 81
9.2. Implémentation en C++ 84
9.3. A propos des règles sur l'ergonomie 87
10. Evaluation par prototype 88
11. Synthèse, apports et perspectives 90

Utilisateurs interrogés a
Bibliographie c
A propos de l'interface homme-machine c
A propos des télécommunications d
A propos de logiciels de télécommunication e
Table des matières f