POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title: Etudes des aspects cognitifs applicables au géie logiciel

Auteurs: Authors: Pierre N. Robillard

Date: 1996

Type: Rapport / Report

Référence: Citation: Robillard, P. N. (1996). Etudes des aspects cognitifs applicables au géie logiciel. (Rapport technique n° EPM-RT-96-18). https://publications.polymtl.ca/9863/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL: https://publications.polymtl.ca/9863/

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Terms of Use: Tous droits réservés / All rights reserved

Document publié chez l'éditeur officiel
Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: Report number: EPM-RT-96-18

URL officiel: Official URL:

Mention légale: Legal notice:

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

https://publications.polymtl.ca

1 3 FEV. 1997

Étude des aspects cognitifs applicables au génie logiciel

gratuit

Pierre N. Robillard, Ph.D., Ing.

Professeur titulaire

Département de génie électrique et de génie informatique

École Polytechnique de Montréal.

Septembre 1996.

Tous droits réservés. On ne peut reproduire ni diffuser aucune partie du présent ouvrage, sous quelque forme pour par quelque procédé que ce soit, sans avoir obtenu au préalable l'autorisation écrite des auteurs.

Dépôt légal, 09, 96
Bibliothèque nationale du Québec
Bibliothèque nationale du Canada

Études des aspects cognitifs applicables au génie logiciel (EPM/RT-96/18)

Pierre N. Robillard (génie électrique et génie informatique)

Pour se procurer une copie de ce document, s'adresser au:

Service des Éditions
École Polytechnique de Montréal
Case postale 6079, Succursale Centre-ville
Montréal (Québec) H3C 3A7
Téléphone: (514) 340-4473
Télécopie: (514) 340-3734

Compter 0,10$ par page et ajouter 3,00$ pour la couverture, les frais de poste et la manutention. Régler en dollars canadiens par chèque ou mandat-poste au nom de l'École Polytechnique de Montréal

Nous n'honorerons que les commandes accompagnées d'un paiement, sauf s'il y a eu entente préalable dans le cas d'établissements d'enseignement, de sociétés ou d'organismes canadiens.

1. INTRODUCTION **3**

2. DÉMARCHE INTELLECTUELLE **5**

2.1 PSYCHOLOGIE COGNITIVE 5
2.2 RÉSOLUTION DE PROBLÈME 6
2.3 MODÈLE MENTAL 9

3. REPRÉSENTATION NATURELLE DE LA CONNAISSANCE **13**

3.1 CONCEPTS ET CATÉGORIES 14
3.2 CONNAISSANCE PROCÉDURALE ET DÉCLARATIVE 16
3.3 CHUNKS 17
3.4 SCHÉMA 18
3.5 PROPOSITION 21

4. REPRÉSENTATION ARTIFICIELLE DE LA CONNAISSANCE **23**

4.1 LOGIQUE DE PREMIER ORDRE 24
4.2 RÈGLES DE PRODUCTION 24
4.3 PROGRAMME INFORMATIQUE 25

5. TRAITEMENT DE LA CONNAISSANCE **26**

5.1 CONCEPTION 27
5.2 PLANS 30
5.3 ACTIVITÉ DE PLANIFICATION 32
5.3.1 Systématique *33*
5.3.2 Opportuniste. *33*
5.3.3 Expérience *34*

6. CONCLUSION **36**

7. REMERCIEMENT **39**

8. BIBLIOGRAPHIE **40**

1. Introduction

Le génie logiciel est caractérisé par une grande variété d'activité mentale. Depuis le début des années 1980 on tente de supporter ces activités par toute une famille de méthodes, d'approches et d'outils logiciels.

Les différentes phases du cycle de vie du logiciel illustrent bien cette diversité. La définition des requis consiste à bien comprendre les besoins du client. C'est une activité ou la connaissance du domaine d'application est importante. La capacité de décrire les besoins dans un langage approprié pour le client est primordiale.

L'analyse peut résulter de deux démarches distinctes. L'une consiste à choisir une solution appropriée parmi un ensemble de solutions existantes. L'autre s'apparente à une approche de solution de problème dans l'espace de connaissance du domaine d'application. Il s'agit ici de concevoir des alternatives et de les valider. Cette approche est souvent exploratoire.

Le design ou la conception lient le résultat de l'analyse à l'environnement informatique. La conception implique une activité de créativité dans un cadre contraignant. Les exigences du support informatique et du langage de programmation bornent l'espace de création.

La réalisation ou la codification est une activité mentale quasi routinière qui consiste à traduire les spécifications du design dans le langage de programmation approprié en respectant le guide de programmation ou les règles de l'art.

La documentation est une activité de synthèse qui consiste à décrire de façon concise les caractéristiques du système et de sa réalisation.

Chacune de ces activités est réalisée différemment en fonction de l'expérience des individus. Le niveau d'expérience est primordial mais son rôle comme paramètre dans la

démarche mentale est mal connu. Peu de méthodes en génie logiciel tiennent compte du facteur expérience dans leur utilisation.

Ce bref survol des différentes phases du cycle de vie du logiciel illustre la variété des processus mentaux impliqués. Pour chacune de ces phases il existe une démarche mentale appropriée et probablement distincte des autres. Les démarches actuelles sont parfois le résultat d'observations empiriques ou de succès individuels qui ont été diffusés dans la communauté et qui servent de modèle.

Ces méthodes ou approches sont rarement le résultat d'études basées sur le comportement cognitif de l'individu. Il est possible qu'une connaissance des processus mentaux propres à une activité spécifique puisse aider à concevoir des approches qui lui soit approprié et efficace. Il n'existe peut-être pas de démarche universelle applicable à toutes les situations mais quelques approches pertinentes bien adaptées seraient déjà un gain significatif.

Depuis de nombreuses années les psychologues et les cognitiens étudient les processus mentaux. Le but de ce document est de présenter un aperçu des connaissances acquises dans le domaine de la psychologie cognitive. Le domaine est très vaste et l'étude doit se restreindre aux aspects susceptibles de contribuer à améliorer notre connaissance des processus mentaux utilisés en génie logiciel.

Ce document exclu les aspects reliés à l'intelligence artificielle. L'objectif de l'intelligence artificielle est de comprendre les processus cognitifs dans le but de les modéliser et de les reproduire. Cette démarche est à l'opposé de celle poursuivi par cette recherche qui consiste à utiliser au profit de l'humain notre meilleure comprehension des processus cognitifs.

Ce document est divisé en cinq parties. Suite à cette introduction, les aspects globaux de la démarche intellectuelle sont présentés. Par la suite les bases des représentations naturelles et artificielles des connaissances sont résumées. Le traitement

de la connaissance fait l'objet de la cinquième partie. La conclusion résume la démarche et identifie le point de départ de nouvelles recherches.

2. Démarche intellectuelle

2.1 Psychologie cognitive

La psychologie cognitive est l'étude des capacités de traitement du cerveau en fonction de différentes tâches. L'hypothèse de base stipule que le cerveau est un système physique de traitement de symboles. Cette hypothèse est le préalable pour un comportement intelligent. Un tel système est capable d'insérer, de retirer, de stocker et de modifier des structures de symboles et de réaliser des actions en réponse aux symboles eux-mêmes.

Plusieurs études ont permis de découvrir des invariants propres au mécanismes intellectuels qui ont des conséquences fondamentales sur son fonctionnement. Ces invariants sont présentés ici dans un ordre quelconque:

- la compréhension se fait principalement sur une base propositionnelle;
- la mémoire à court terme ne peut contenir simultanément qu'une à quatre propositions;
- un acte de reconnaissance demande près d'une seconde pour se réaliser;
- la réaction humaine la plus simple se mesure en centaines de milisecondes;
- nous considérons principalement les objets à leur niveau de base.

Ces constantes physiologiques de base déterminent les types de traitement qui sont applicables dans certaines situations et à quelle vitesse ils seront réalisés. Ces invariants constituent la plus importante découverte de la psychologie cognitive et expliquent la plupart des comportements observés lors du processus de réflexion et d'apprentissage.

Toute méthode basée sur le comportement humain doit tenir compte de l'aspect fondamental des limites de la mémoire à court terme. La découverte de Miller concernant les chunks est primordiale. Un chunk n'est pas une mesure fixe de la capacité de stockage de la mémoire à court terme, mais plutôt tout stimulus qui nous est familier, donc reconnaissable par l'expérience.

Il va de soi que ces découvertes ne constituent en rien les bases de fonctionnement de l'esprit. En effet, comment un système aussi limité peut-il nous permettre de comprendre le monde qui nous entoure? Nous n'avons à ce jour identifié que quelques éléments limitatifs du fonctionnement du cerveau. Les caractéristiques fondamentales qui assurent sa capacité exceptionnelle de traitement de l'information demeurent encore inconnues.

Il est raisonnable de penser que toute méthode intellectuelle doit au moins tenir compte des connaissances actuelles en psychologie cognitive. *121, 123*

2.2 Résolution de problème

Toutes les tâches intellectuelles ne sont pas également difficiles à réaliser par un même individu et la même tâche peut paraître de difficulté variée à différents individus. Ceci nous amène à définir ce qui constitue la difficulté d'une tâche.

Certaines tâches peuvent se résumer qu'à un travail routinier dans le sens que toutes les connaissances et les mécanismes intellectuels nécessaires pour la réaliser correctement sont disponibles. D'autres tâches, par contre, peuvent demander une charge cognitive importante et sont alors associées à une activité de solution de problème.

On définit un problème comme suit:

> un problème est une représentation d'un système cognitif construit à partir d'une tâche sans disposer immédiatement d'une procédure admissible pour atteindre le but

Suivant Reitman, un problème est bien défini si l'état initial, le but et un ensemble d'opérations possibles pour atteindre le but à partir de l'état initial sont spécifiés. *142, 132-167, 132-168*

Les problèmes académiques que l'on pose pour vérifier la maîtrise d'une connaissance sont des problèmes bien définis. Par exemple, calculer les forces dans un treillis, décrire la trajectoire d'une particule chargée dans un champ électromagnétique, obtenir le résultat d'une réaction chimique, prédire l'évolution démographique à partir d'un modèle de population, etc. La solution de ce type de problème est souvent caractérisée par la recherche en mémoire d'un algorithme existant permettant d'obtenir la réponse recherchée. La maîtrise de ce type de problème s'acquiert par l'étude, donc par l'acquisition de connaissances. Nous qualifions ce type de problème d'exercice intellectuel et nous réservons le mot problème pour les problèmes mal définis.

Un problème mal défini n'a pas de but très clairement spécifié parce qu'il peut exister plusieurs buts acceptables. Il en découle que les opérations cognitives nécessaires pour atteindre le but ne peuvent être clairement spécifiées parce qu'il peut exister plusieurs façons d'atteindre le but. *101*

Par définition ce type de problème est mal structuré et mal défini, en ce sens que la représentation initiale du problème ne permet pas de déclencher une procédure d'exécution. L'espace-problème peut être trop large. Dans la résolution de problème le sujet dispose des buts et il doit trouver les actions pour les réaliser. *142-*

La formulation moderne d'une théorie générale de la résolution de problèmes comme sujet de recherche est due à Newel et Simon. Dans leur formulation de l'espace-problème, la représentation de ce dernier comprend quatre types d'éléments:

1. une description de l'état initial ou la résolution de problème commence;

2. une description de l'état-but à atteindre;

3. un ensemble d'opérateurs ou d'actions susceptibles d'être entreprises qui servent à modifier l'état actuel du problème;

4. les contraintes de cheminement qui imposent des conditions supplémentaires pour que le chemin vers la solution soit correct au-delà de la simple atteinte du but. *139-572*

On considère habituellement les problèmes de design comme des problèmes mal structurés. Ils n'existent donc pas de critères définitifs pour tester la solution proposée. Ainsi, les solutions de design sont en général plus ou moins acceptables ou satisfaisantes et sont ni correctes ni incorrectes. *101*

La tâche de design implique une activité intensive de structuration du problème qui est le processus de découvrir l'information manquante et de l'utiliser pour définir l'espace de problème. Le processus de design implique la découverte de nouvelles connaissances et en particulier la découverte de buts non identifiés et de critères d'évaluation. *30*

Un modèle de design devrait décrire le processus de création de solution, spécifier la connaissance créée pendant le design et démontrer les interactions complexes entre le retrait de la connaissance et la création de connaissance qui survient lors du design. *41*

Un examen des dialogues durant le design a révélé que la démarche consiste en une succession de cycles de design. De nouvelles spécifications sont souvent découvertes dans le processus d'examens des solutions et le système de solution de problème peut alors démarrer un autre cycle de design. Les experts savent comment déterminer les sous-buts pour atteindre une solution acceptable. *126*

En résumé, le design est composé de recherche d'informations existantes, de création de nouvelles informations et d'organisation originale des informations existantes ou nouvelles. La création d'informations est relative à l'individu et est fonction de l'expérience de l'individu.

L'habileté à résoudre un problème est alors basée sur l'expérience acquise. Il est important ici de ne pas confondre exercice intellectuel qui est basé sur l'acquisition de connaissance et souvent facile pour les gens sans expérience et la solution de problème qui est la caractéristique des experts qui ont pris le temps de développer l'expérience nécessaire.

Trois facteurs ont été identifiés comme pouvant contribuer à la structure d'un problème: *48*

1. le degré de nouveauté du problème;
2. le degré de complétude de ces spécifications;
3. le nombre de domaines de connaissance nécessaires pour le résoudre.

Le principal critère de sélection des mécanismes intellectuels est l'économie cognitive. C'est à dire qu'on veut minimiser le niveau de difficulté.

Dans ce qui suit nous allons décrire les deux éléments nécessaires pour la solution d'un problème: la connaissance et les stratégies intellectuelles.

Étudier la résolution de problèmes oblige à un examen attentif des situations elles-mêmes, des capacités cognitives et des connaissances antérieures des individus en situation ainsi qu'à une analyse de la tâche dans laquelle ils sont impliqués. *139-554*

2.3 Modèle mental

Un problème est en général trop vaste intellectuellement pour être solutionné directement. On doit fractionner l'espace des solutions et utiliser différentes méthodes pour permettre de converger vers une solution acceptable. En un mot, on doit effectuer une certaine réflexion. L'activité de réflexion est une suite d'activités sur les symboles internes qui conduisent à de nouvelles idées ou à des conclusions.

Plusieurs études supportent l'hypothèse que l'activation de connaissances est limitée et très spécifique et que le processus de pensée est, en fait, un cheminement à travers un espace de connaissances. Les modèles mentaux définissent le contexte dans lequel s'effectue la réflexion. *110.*

Il semble que le processus de pensée de base soit le même quelle que soit la complexité de la connaissance ou le niveau d'expertise. Ces approches sont basées sur l'existence d'une représentation mentale du problème ou d'une partie de celui-ci. Cette section introduit la notion de modèle mental et illustre l'utilisation de ce concept.

Un modèle mental est toute représentation mentale qui s'élabore au moment de l'interaction de l'individu avec le monde, la situation ou les choses qui l'entourent. Il s'agit en fait d'une structure mentale transitoire et provisoire qui se construit dans la tête de l'individu dès qu'il agit ou traite de l'information et qui dure le temps de l'exploration de la situation. Le modèle mental est utilisé dans la manipulation mentale qui permet de voir en pensée des transformations possibles de cette même représentation. Cette structure mentale est donc dynamique et évolue pendant le traitement de la situation.

Les constituants de cette structure mentale provisoire en mémoire de travail proviennent à la fois des informations en cours de traitement et d'autres matériaux stockés en mémoire à long terme et constitués par les conceptions et les connaissances antérieures. *438, 132-135, 584*

Les modèles mentaux ont pour contraintes les connaissances techniques de l'utilisateur, ses expériences antérieures avec des systèmes semblables ainsi que la structure du système humain de traitement de l'information. Plusieurs études basées sur l'hypothèse de l'existence des modèles mentaux ont permis les observations suivantes:

- les modèles mentaux sont incomplets;
- les capacités des gens à *faire tourner leurs modèles* sont sérieusement limitées;

- les modèles mentaux sont instables: les gens oublient les détails du système qu'ils sont en train d'utiliser;
- les modèles mentaux n'ont pas de frontières fermes: des dispositifs et des opérations semblables peuvent être confondus les uns avec les autres;
- les modèles mentaux sont non-scientifiques;
- les modèles mentaux sont économiques (économie cognitive).

Les modèles mentaux sont, par nature, à cause de leurs aspects dynamiques, des modèles en élaboration; c'est dans l'interaction avec le système cible ou le monde réel que les gens forment des modèle mentaux de ce système.

La logique et les mathématiques ont permis de concevoir des modèles mentaux formels. L'utilisation d'une notation formelle a pour but de limiter les interactions entre les éléments du modèle mental et de définir une sémantique restreinte basée sur un vocabulaire limité. Le modèle mental ainsi formé peut être sémantiquement correct et complet. Par contre, il n'y a pas de garantie qu'il est conforme à la réalité.

Un modèle mental n'a d'utilité pratique que si l'on fait l'hypothèse de l'existence d'un isomorphisme entre le modèle et le monde (d'une correspondance terme à terme entre les éléments du monde et les entités du modèle). Toutefois la notion d'homomorphisme, qui est moins forte, est peut-être préférable, puisqu'elle décrit une correspondance entre le monde et le modèle dans le cadre d'une relation plus globale d'ensembles d'éléments à éléments.

L'établissement de l'homomorphisme est une des difficultés reliées au modèle mental. En effet, comment mesurer le degré d'homomorphisme? Il semble que l'expérience de l'individu y joue un rôle prépondérant.

Une autre difficulté reliée au modèle mental est la démarche d'enrichissement du modèle. Comment s'y prend-on pour préciser et ajouter de l'information au modèle mental?

Il existe une prédiction empirique sur le nombre de modèles mentaux dont l'humain a besoin pour résoudre un problème. Plus le nombre de modèles requis est grand, plus grande est la demande sur la mémoire de travail et plus grand est le nombre d'erreurs possibles. *113, 117*

Dans la solution de problèmes complexes, on doit trouver des façons de réduire le nombre de modèles mentaux utilisés en mémoire de travail. On doit donc choisir un nombre optimal d'alternatives. En augmentant inutilement nos alternatives, on ne fait qu'augmenter notre probabilité de faire un mauvais choix. Comme il n'y a pas de solution unique, mais plutôt une famille de solutions, vouloir trouver la meilleure solution conduit à charger inutilement la mémoire de travail.

Le modèle mental définit l'espace de réflexion. C'est une espèce de filtre qui s'applique sur tous les liens que l'esprit peut effectuer à partir de mots ou d'images. Ne sont retenus ou favorisés que les liens qui sont en accord avec le modèle mental. Le modèle mental est un outil de convergence intellectuelle.

Par exemple, lorsqu'on utilise un système d'exploitation de type Windows95, on n'a pas le même modèle mental que lorsqu'on utilise un système Window3.1 ou un sytème DOS. On peut conclure que le cheminement intellectuel est basé sur le modèle mental qu'on se fait de l'environnement.

Dans ce sens, une méthodologie peut être considérée comme un modèle mental. La méthodologie détermine dans quel sens se fera la démarche intellectuelle. Une approche orientée-objet n'est pas la même démarche de réflexion qu'une approche procédurale.

Le modèle mental peut aussi être considéré comme une préparation sémantique. Pour optimiser la communication entre deux personnes, il serait normal de s'assurer qu'elles possèdent le même modèle mental. Par exemple, les quiproquos peuvent parvenir de communications effectuées dans deux modèles mentaux différents. Un bon modèle mental définit des démarches qui produisent des résultats conformes aux attentes.

Un modèle mental est un environnement contextuel qui définit un cadre de traitement de l'information. Les modèles mentaux s'imbriquent les uns aux autres. Chaque modèle mental est unique, il représente notre perception de la situation au moment ou nous la vivons.

3. Représentation naturelle de la connaissance

Une représentation, c'est la possibilité de disposer des caractéristiques d'un objet en son absence. Une représentation peut être naturelle, c'est-à-dire mentale et immatérielle, ou artificielle. La représentation naturelle ou mentale est transitoire et n'existe que dans la conscience de l'individu. La représentation artificielle est physique, elle peut être décrite et montrée. Elle est souvent formelle et peut se représenter, par exemple, sous forme d'énoncés de logique de premier ordre, de règles de production ou de programmes informatiques.

Il est difficile de définir ce qui compose la connaissance puisqu'il s'agit de tout ce que nous connaissons sur quelque chose. Une difficulté supplémentaire provient du fait que la connaissance peut être définie en fonction de différents points de vue. En fait, les modèles de la connaissance sont élaborés en fonction du contexte dans lequel on a effectué les études.

Pour modéliser cette connaissance on utilise différentes notions. Ces notions sont des concepts, des types de connaissances procédurales ou déclaratives, des chunks, des schémas, et des propositions entre autres. Il faut comprendre que ces notions ne sont pas des objets tangibles ou mesurables, mais plutôt des concepts créés par les cogniticiens et les psychologues pour caractériser une démarche intellectuelle. Ces notions ont, par définition, des frontières floues. Elles sont utiles pour caractériser un comportement ou une façon de faire. Ces notions sont elles-mêmes des modèles mentaux d'un système cible mal connu, soient les mécanismes de la pensée humaine. La prudence nous recommande d'être critique vis-à-vis ces notions et de les considérer comme une représentation de notre connaissance actuelle de certains mécanismes mentaux.

Le modèle mental est dynamique. Notre pensée n'est habituellement pas fixe. Nous voyageons constamment d'un concept à un autre. Bien qu'à un temps fixe nous n'avons en mémoire de travail ou en pensée qu'un concept, nous conservons une trace des concepts passés.

Nous avons deux types de changement de représentation. Ceux qui réduisent la quantité d'information manipulée, sans changer sa forme et ceux qui modifient la forme de présentation de l'information.

Plusieurs études relient la connaissance et la compréhension. On retrouve alors les notions de connaissance syntaxique et sémantique et de concepts. Le psycho-physiologiste associe la connaissance au type de mémoire ou structure cérébrale dont elles dépendent. On retrouve alors la notion de connaissance procédurale et déclarative qui est liée aux types de mémoire biologique respective. Le psychologue est conscient que l'humain ne peut traiter que certains éléments de connaissance à la fois qu'il nomme des chunks. L'informaticien qui s'intéresse à l'intelligence artificielle formalise certains éléments de connaissance sous forme de schéma, scénario, cadre, plan etc. Finalement, ceux qui s'intéressent à l'acquisition de connaissance la décomposent sous une forme atomique en terme de propositions et de prédicats.

Notre but est d'utiliser ces modèles pour mieux comprendre comment l'ingénieur logiciel s'y prend pour passer d'une représentation naturelle à une représentation artificielle, qui est ultimement le programme informatique.

3.1 Concepts et catégories

Toutes les notions que nous connaissons sont reliées entre elles, sinon, il serait impossible de les retrouver. Une nouvelle notion est toujours créée dans le cadre d'un modèle mental. Lorsque différentes notions se regroupent dans le contexte de différents modèles mentaux, on peut estimer que ces notions forment une catégorie. La conceptualisation consiste à rechercher les aspects communs de différentes notions. Un concept permet d'accéder à ces notions.

Un concept est une représentation mentale d'une classe qui inclut ce que nous connaissons au sujet de quelque chose. Une catégorie est un ensemble d'exemples appartenant à un concept. La principale fonction d'un concept est la classification qui donne accès à toute la connaissance qu'on a sur une entité. La vue classique d'un concept suppose que chaque concept a un ensemble de propriétés qui sont individuellement nécessaires et globalement suffisantes pour catégoriser une instance.

Un aspect important de la relation entre les concepts est relié au niveau d'abstraction dans lequel une entité particulière peut être classée. On note, premièrement, que la compréhension humaine est hiérarchisée et basée sur des classes de niveaux de base. Ceci signifie que les concepts que nous manipulons font partie d'une hiérarchie dont peu d'attributs sont hérités des niveaux supérieurs et seulement quelques attributs peuvent être hérités des niveaux inférieurs.

Par exemple, considérons la hiérarchie de concepts suivante: meuble/chaise/chaise de plage. Il semble que l'humain préfère systématiquement utiliser le concept de chaise qui est le concept de base. Donc, de façon instinctive, l'humain utilise le concept de base qui n'est ni le plus abstrait ni le plus concret. *103*

Le niveau de base est psychologiquement privilégié. Ce niveau n'est ni le plus abstrait ni le plus spécifique, mais plutôt un niveau intermédiaire d'abstraction. Le niveau de base est un compromis entre deux buts opposés des catégories; le désir d'avoir la catégorisation la plus informative possible et le désir d'avoir des catégories le plus discriminatives possibles.

Ce dernier point nous conduit à la notion d'économie cognitive. Il semble que l'humain utilise de façon naturelle la démarche cognitive qui est la plus économique, celle qui demande le moins de charges à la mémoire de travail.

3.2 Connaissance procédurale et déclarative

Les études qui associent le type de connaissance à la façon dont elle sont stockées en mémoire reconnaissent deux types de mémoires ou de connaissances, l'une procédurale et l'autre déclarative.

Connaissances procédurales (ou savoir faire): elles sont dynamiques.

La mémoire procédurale regroupe toutes les informations relatives aux habiletés psychomotrices que nous avons développées pour agir sur notre environnement: marcher, nager, patiner, s'exprimer, gesticuler, etc. La mémorisation et l'acquisition de connaissances reliées à ces stratégies cognitives reposent le plus souvent sur une longue pratique et leur exécution ne réclame aucun soutien verbal et ne peut être décrite. De plus, il s'agit d'apprentissages qui ne s'oublient jamais. *98, 140-484*

Connaissances déclaratives (ou savoir): concernent des faits. Elles sont statiques et portent sur les propriétés des objets et de leurs relations. *142*

La mémoire déclarative contient toutes les informations directement accessibles à une récupération consciente. Elle se définit par la connaissance que nous avons des objets, des êtres ou des événements. Elles peuvent être décrites et sont faciles à communiquer.

La distinction entre connaissances déclarative et procédurale semble se confirmer par des études expérimentales. Des dommages à l'hippocampe, par exemple, peuvent causer une inhabileté totale à apprendre de nouveaux faits, mais préservent l'habileté à apprendre de nouvelles aptitudes. *136-87*

La mémoire déclarative englobe deux grands systèmes: les connaissances topiques ou sémantiques et les connaissances épisodiques.

La connaissance topique réfère à l'aspect essentiel du sens des mots, comme par exemple celui qu'on peut trouver dans un dictionnaire ou un article encyclopédique sur le sujet. La mémoire sémantique est constituée de toutes les structures propres à une culture donnée, qui permettent d'organiser la connaissance du monde. *140-484*

La connaissance épisodique (ce terme a été inventé par Tulving en 1972) constitue notre propre expérience de la connaissance. La mémoire épisodique traite toute l'information relative aux différents événements de notre vie.

Le mot oiseau illustre ces types de connaissances:

la définition du dictionnaire est une connaissance topique (petit ROBERT):

> animal appartenant à la classe des vertébrés tétrapodes à sang chaud, au corps recouvert de plumes, dont les membres antérieurs sont des ailes, les membres postérieurs des pattes, dont la tête est munie d'un bec corné dépourvu de dents et qui est en général adapté au vol.

notre expérience des oiseaux est, par contre, une connaissance épisodique:

> les hirondelles qui annoncent le printemps, les pigeons qui envahissent les parcs et les monuments, les impressionantes migrations des oies blanches, le crie perçant du huard sur les lacs, etc.

Par exemple, la crainte d'un animal ou d'un insecte peut être la conséquence d'une connaissance épisodique reliée à la connaissance topique du sujet.

3.3 Chunks

L'approche cognitive utilise le terme de chunk pour définir globalement la connaissance en fonction de sont utilisation. La connaissance que nous pouvons traiter à tout instant donné est limitée et sa nature est par contre très variée. Ce concept a été introduit en 1956 suite aux travaux de Miller sur la capacité de la mémoire à court terme ou de travail.

Un chunk est une unité de représentation qui peut être composé des éléments d'une représentation et groupés ensemble par leur relation fonctionnelle. L'unité de chunk est une notion qui permet de définir de façon relativement constante la capacité de

mémorisation. L'idée de Miller est que la trace mnémonique ne peut s'étendre que sur environ les 7 ± 2 dernières unités de représentation. *102, 129, 142*

Par exemple, mémoriser VGT WMB est plus difficile que TGV BMW. Cette dernière séquence n'est pas vraiment plus difficile à mémoriser que TRAIN-RAPIDE AUTO-DE-LUXE. Dans la première séquence chaque lettre forme un chunk alors que dans la deuxième séquence chaque groupe de lettres forme un chunk. Dans la troisième séquence chaque groupe de mots forme un chunk.

Cette notion a suscité beaucoup d'intérêt à l'époque, mais elle apparaît aujourd'hui trop générale et peut-être un peu simpliste. Elle est toutefois toujours utilisée pour identifier globalement l'ensemble des connaissances que nous pouvons traiter à un moment donné.

3.4 Schéma

La notion de schéma a fait son chemin dans la psychologie cognitive contemporaine à partir des travaux de Bartlett (1932) et de Piaget(1952). Nous esquissons les idées de base du schéma en particulier tel qu'elles sont développées dans les articles de Rumelhart et Ortonoy (1977), Bobrow et Norman (1978) et Rumelhart (1981).

Ce modèle suppose que la connaissance est entreposée en mémoire sous la forme de schémas qui composent les modèles mentaux qui nous permettent d'avoir une vision organisée du monde réel. Nous possédons ainsi un schéma pour chaque classe d'objets ou de personnes. Un schéma représente une information déjà structurée en mémoire. Un schéma décrit un lien particulier entre une variété d'éléments d'information. Les schémas font partie de l'organisation de la mémoire déclarative. Ces schémas peuvent avoir un caractère topique (sémantique) ou épisodique. Les éléments d'un schéma topique représentent des objets ou des concepts. Les éléments d'un schéma épisodique représentent un système d'événements reliés temporellement et causalement.

Par exemple, le schéma topique d'un restaurant représente l'organisation en mémoire de toutes sortes d'informations au sujet d'un restaurant:

son but, son organisation physique, son personnel, etc.

D'autre part un schéma épisodique est associé à un scénario qui décrit les séquences typiques d'événements qui surviennent dans un restaurant:

choisir une table, lire le menu, commander, manger, payer, quitter.

Il va de soi qu'en réalité, lorsqu'on invoque un schéma, il est inconsciemment composé dans un rapport varié d'aspects topiques et épisodiques. *122*

En fait, un schéma est une structure de connaissance qui capture les régularités des objets et des événements. Dans la plupart des théories, les schémas possèdent des structures de cases (slots) dans lesquelles les propriétés des objets, des séquences typiques, les agents particuliers peuvent être spécifiés.

Ces cases sont considérées comme des variables qui peuvent avoir des valeurs par défaut ou encore être spécifiées directement. Autrement dit, le schéma contient des informations sur des valeurs qu'il faut supposer aux variables lorsque l'information d'entrée ne comporte pas de spécifications. Un schéma n'est pas entièrement spécifié: il représente des relations entre des variables et des contraintes sur leurs valeurs ce qui suppose que nous connaissions le domaine plausible dans lequel peuvent se situer les valeurs que peut prendre une variable donnée. Ces variables seront particularisées au cours de la mise en oeuvre du schéma. Les schémas peuvent avoir des attachements procéduraux. Ces variables peuvent avoir des valeurs par défaut. *111, 142*

L'exemple suivant illustre l'importance de la structure de case et de leur valeur par défaut.

Julie va au restaurant, elle demande au garçon ce qu'il y a au menu, elle commande alors une glace vanille et va rejoindre ses amies dans la rue.

Dans le schéma que nous avons d'un restaurant on assume qu'une personne paie pour ce qu'elle commande. Cette structure par défaut fait en sorte qu'il ne vient pas à

l'idée que Julie puisse avoir volé la glace vanille, bien que la case qui définit l'action de payer ne soit pas spécifiée.

Les schémas sont donc des structures de connaissances abstraites qui peuvent inclure de l'information au sujet de multiples objets et de leurs relations. On distingue un schéma d'une catégorie de la manière suivante: une catégorie consiste en un seul type d'objets qui ont un lien taxonomique commun comme une chaise, fauteuil, etc., tandis qu'au contraire, un schéma spécifie des relations entre plusieurs catégories discrètes et souvent sans tenir compte de la taxonomie. *132-134.*

La notion de schéma est utilisée par les psychologues dans l'étude de la compréhension des textes (Bower et al 1979, Rumelhart, 1981). L'étude de la programmation informatique basée sur les schémas a débuté avec les travaux de Rich en 1981 et se poursuit actuellement avec, entre autre, l'équipe de Détienne. Plusieurs études sur la compréhension des programmes ont tenté de faire le lien entre les schémas et les programmes. *41*

En intelligence artificielle, les informaticiens ont besoin d'élaborer la théorie des schémas. Un schéma est alors une structure de donnée qui représente un concept générique stocké en mémoire (Abelson 1981, Minsky 1975, Schank & Abelson, 1977). Cette théorie est complétée par les notions de cadre (frame), scénario (script) et plan.

Il semble que les schémas présentent un certain nombre de caractéristiques nécessaires ou, du moins, utiles pour développer un système qui se comporte de cette manière. (Rumelhart, 1981 et Rumelhart et Ortony, 1977) ont établi les caractéristiques les plus importantes des schémas. Les voici:

- les schémas comportent des variables;
- les schémas peuvent s'enchâsser les uns dans les autres;
- les schémas représentent des connaissances à tous les niveaux d'abstraction;

- les schémas représentent des connaissances plutôt que des définitions;
- les schémas sont des dispositifs actifs de reconnaissance dont la mise en oeuvre a pour but l'évaluation de leur capacité à correspondre aux données en cours de traitement.

Il existe deux sources de données pour construire un schéma:

- l'information est fournie par des sous-schémas (information ascendante);
- l'information provient de schémas dont le schéma courant est un constituant (information descendante). *41, 139-312*

Les schémas s'enchâssant les uns dans les autres, un schéma consiste donc généralement en une configuration de sous schémas. Chaque sous schéma consiste également en une configuration de sous schémas, etc. On devine qu'on atteint éventuellement des schémas plus primitifs ou encore indécomposables. Cette démarche nous conduit naturellement vers la notion de proposition.

3.5 Proposition

La structure propositionnelle de la représentation mentale a été découverte initialement par des philosophes-logiciens et on est redevable de sa conception moderne à l'analyse donnée par Frege (1892)

Une proposition est la plus petite quantité de connaissance qui peut constituer une affirmation: la plus petite unité qui peut être vraie ou fausse. Les propositions sont des représentations discrètes d'éléments de connaissance et semblent interconnectées en mémoire en fonction de leurs arguments partagés. Une proposition est une idée exprimée en mots. Suivant la théorie de Kintsch (1988), les modèles mentaux sont des représentations propositionnelles complexes. *136-98*

L'hypothèse théorique concernant la structure cognitive de l'information dans le système de traitement de l'humain consiste à dire qu'à une certaine étape du traitement,

l'information est organisée sous une forme particulière qu'on nomme PROPOSITIONNELLE. Selon cette hypothèse, les premières opérations de traitement transforment l'information de telle manière que, lors de la sortie de ces opérations, l'information se trouve, pour l'essentiel, sous forme propositionnelle; les opérations de traitements ultérieures maintiennent cette forme et s'effectuent à partir d'elle. *139-383*

Les propositions ont trois fonctions importantes:

1. il est généralement accepté que les propositions puissent représenter toute information bien spécifiée, ce qui implique que la représentation propositionnelle est un mécanisme général pour représenter la connaissance humaine;

2. la représentation propositionnelle préserve la signification, mais non la forme;

3. les propositions supportent de façon naturelle le raisonnement et les inférences. *136-103*

L'idée importante est que la forme propositionnelle, dans l'ensemble, gouverne les activités sémantiques de traitement central qui se déroulent au cours de la compréhension, de la mémorisation, de la conservation et de la partie reproduction du contenu de textes.

Une formulation générale de la même idée est la suivante: dans les représentations humaines, la forme propositionnelle est la forme dominante. Cette hypothèse peut être considérée comme la meilleure de celles qui sont actuellement disponibles.

L'hypothèse propositionnelle est basée sur le prédicat psychologique.

Le prédicat est utilisé en logique par référence aux valeurs de vérité vraie et fausse. C'est une fonction au sens mathématique du terme qui fait correspondre à tout cas particulier, comme par exemple BLOND(x) tel que Pierre est blond, l'une des deux valeurs VRAI ou FAUX.

Une proposition est formée de prédicats et d'arguments. Les propositions qui comportent un prédicat accompagné par un seul argument sont appelées prédicat à une place, monadiques ou unitaires.

Dans plusieurs modèles cognitifs un tampon de mémoire à court terme à capacité limitée joue un rôle crucial dans la compréhension. La taille de ce tampon a été estimée comme située entre une et quatre propositions. Cette estimation est en accord avec les travaux de Miller concernant les tests d'empan mnésique ou toutes les ressources sont utilisées pour le stockage de 7 à 9 chunks. *139-399*

4. Représentation artificielle de la connaissance

On appelle une représentation artificielle une représentation que l'on peut fabriquer, reproduire, montrer ou exhiber. La représentation artificielle existe de façon physique. C'est une description d'une structure empirique dans un langage scientifiquement approprié.

Il existe plusieurs raisons pour justifier l'utilisation de représentations artificielles particulièrement en science. Parmi celles-ci on trouve:

- réduire l'information qui consiste à caractériser un ensemble de concepts théoriques en termes d'un autre;
- améliorer la compréhension qui consiste à résumer une représentation par un petit nombre de principes formalisés par des axiomes;
- représenter la connaissance qui consiste à simplifier efficacement le modèle mental. Toutefois, ceci peut quelquefois conduire à une sur-simplification;
- représenter la complexité qui consiste à capturer les détails des relations entre différents concepts.

Le problème de la représentation artificielle de la connaissance a occupé autant les psychologues, les informaticiens que les experts en intelligence artificielle. On a déterminé

plusieurs modèles pour représenter la connaissance. Ces principaux modèles sont construits dans le but d'en arriver à concevoir des machines qui puissent raisonner ou encore penser.

La représentation de systèmes qui puissent gérer des niveaux supérieurs de connaissance se manifesta en 1975 par la publication de quatre articles

1) *A framework for representing knowledge*	Minsky;
2) *Notes on a schema for stories*	Rumelhart;
3) *The structure of episodes in memory*	Schank;
4) *Concepts for representing mundane reality in plans*	Abelson.

4.1 Logique de premier ordre

Les faits, en logique de premier ordre (FOL First Order Logic), sont représentés par des prédicats. Premier ordre réfère au fait qu'il y a seulement un qualificatif pour les arguments des prédicats et non sur les prédicats eux-mêmes. Les prédicats sont des fonctions qui affectent leurs arguments dans l'une des deux valeurs suivantes; vrai ou faux.

La logique de premier ordre peut en principe effectuer toutes les fonctions de la représentation de la connaissance. Par contre, elle a plusieurs désavantages. L'aspect le plus critique c'est que le raisonnement (preuve de théorème) est intraitable puisqu'il requiert un nombre considérable et non pratique de calculs. En général, le nombre de calculs augmente exponentiellement en fonction du nombre de prédicats. Cette complexité provient du fait que le système doit pratiquement examiner toutes les combinaisons possibles.

4.2 Règles de production

Le système de production est un formalisme général et puissant qui permet de modéliser les connaissances qui supportent les aptitudes cognitives telles que le

raisonnement, la solution de problème et la compréhension du langage. La flexibilité est une caractéristique des règles de production et de la connaissance procédurale qui les distingue de la connaissance déclarative. Les conditions de production sont spécifiques à l'action qui en dérive.

Le but des règles de production est de résoudre des problèmes ou de modéliser le comportement de solution de problème. Son habileté à représenter la connaissance est l'une de ces composantes les plus critiques. Dans un système de règles de production, la connaissance est représentée en termes de condition-action. Si la condition est satisfaite, l'action est effectuée.

L'implémentation de la connaissance en termes de règles de production a plusieurs inconvénients. L'aspect le plus critique est lorsqu'un système de production a plus de quelques centaines de règles. Son comportement est alors difficile à prédire. En se basant sur des études expérimentales, on a estimé que de dix à cent mille règles de production sont requises par un programmeur expérimenté pour réaliser un programme. *129*

4.3 Programme informatique

Un programme informatique est la représentation d'un système dans un langage qui peut être interprété par une machine. Cette représentation peut se faire à l'aide de différents langages qui sont regroupés en trois familles: fonctionnel, déclaratif et procédural.

La représentation informatique a plusieurs particularités. Premièrement, elle regroupe sous une même forme les aspects déclaratifs et opérationnels de la connaissance. Elle permet de décrire simultanément un système et son fonctionnement. Deuxièmement, cette description peut se faire également à différents niveaux d'abstraction:

- opérations ou affectations de base ou élémentaires, additionner, soustraire, déplacer un bit etc;
- les relations ou les propriétés des structures de données;

- les opérations abstraites comme les itérations;
- les représentations globales comme les structures de fichiers.

L'inconvénient de cette représentation c'est qu'on ne peut pas prouver son exactitude excepté dans les cas simplistes et que l'utilité de la représentation ne peut s'obtenir que par feedback de son exécution par la machine désignée.

L'art de bâtir ce type de représentation est l'objet du génie logiciel.

5. Traitement de la connaissance

Maintenant que nous connaissons les principales notions utilisées en psychologie pour représenter la connaissance, nous tentons de comprendre comment l'esprit humain passe d'une perception globale et floue d'une situation à une description précise et complète. En d'autres mots: comment traite-t-on la connaissance?

Pour solutionner un problème un sujet dispose de certaines connaissances, qui peuvent être représentées sous forme de modèle mental, schéma, proposition etc. Il doit naviguer et transformer ces connaissances pour obtenir une solution acceptable, idéalement sous forme d'une représentation artificielle.

Une activité mentale importante est le contrôle du processus de transformation de la connaissance. Il est raisonnable de penser que le contrôle n'est pas indépendant de la base de connaissance du sujet: c'est la théorie du domaine disponible qui permet d'orienter les processus de résolution vers la construction de connaissances utiles, autrement dit de faire des choix parmi toutes les généralisations possibles. Sinon, il y aurait trop de connaissances inutiles et inefficaces qui seraient construites. Il faut supposer un biais de construction vers les connaissances utiles.

La théorie du domaine détermine le langage de description utilisé pour coder les problèmes et leurs solutions. Elle conditionne ainsi la nature des généralisations inductives qui pourront être faites et elle détermine les différents niveaux d'abstraction qui peuvent

être produits compte tenu des hiérarchies conceptuelles disponibles. Elle conditionne le degré de validité des déductions faites puisque la théorie du domaine peut être incomplète, partiellement fausse ou contradictoire. *139-596*

A ce stade nous devons faire l'hypothèse qu'il existe un navigateur mental qui permet de naviguer dans notre univers de connaissances. Ce navigateur fait évoluer le modèle mental en construisant ou assemblant des schémas à partir de différents concepts. Les mécanismes du navigateur sont variés et incluent entre autres, différentes formes de raisonnement:

- l'induction,
- la déduction,
- l'analogie,
- la généralisation,
- l'abstraction,
- la planification.

Chacun de ces mécanismes pris individuellement fait l'objet d'études en psychologie cognitive et en intelligence artificielle. La démarche de planification a été particulièrement étudiée dans le contexte de la programmation des systèmes informatiques.

Il dépasse le contexte de ce travail de s'intéresser à toutes les formes du traitement de l'information. La conception (design) et la planification sont les deux aspects qui sont étudiés parce qu'ils ont un rôle important dans la spécification de représentation informatique.

5.1 Conception

La conception est un problème dans lequel le sujet se représente la tâche comme la construction d'une représentation détaillée du but. Lorsqu'elle est obtenue, le problème

est résolu. La résolution des problèmes de conception se traduit par des changements de représentations et notamment par le passage de représentations schématiques à des représentations détaillées qui peuvent être plus ou moins formelles.

La solution d'un problème de nature scientifique se manifeste surtout par le passage d'une représentation naturelle à une représentation artificielle, bien que certains problèmes consistent à modifier une représentation artificielle existante.

Le sujet dispose initialement d'un certain nombre de propriétés d'une situation, ie d'une représentation très imprécise du but à atteindre et son problème consiste à préciser entièrement cette représentation. Les mécanismes de planification jouent un rôle important dans ces situations de conception, où la situation à concevoir est souvent complexe et doit généralement satisfaire des contraintes qui peuvent être conflictuelles. Ainsi dans un projet informatique, le passage à une représentation détaillée doit être très progressif pour éviter les conflits entre les multiples contraintes.

Le modèle suivant décompose le processus de conception en trois processus de base ayant chacun un état initial et un état final. *126.*

1. L'élaboration des buts consiste en l'énoncé et la discussion des buts du design. Le processus de décomposition du but et de sélection de sous-buts est poursuivi jusqu'à ce que les sous-buts soient suffisamment spécifiques pour être considérés comme des spécifications fonctionnelles qui peuvent s'associer à des propriétés du design. Ceci est l'état final du processus d'élaboration des buts.

2. Le second processus, qui est la génération du design, commence avec les spécifications fonctionnelles et tente d'obtenir l'organisation du design et les éléments du design qui peuvent s'inter-relier dans l'organisation du design de manière à ce que leurs propriétés combinées rencontrent les spécifications fonctionnelles.

3. Lorsqu'un design partiel est généré, le troisième processus, qui est l'évaluation du design, peut commencer. Il débute avec l'introduction d'un nouveau design

partiel et se termine avec son acceptation ou son rejet. L'aspect le plus important de ce processus c'est qu'il peut découvrir de nouvelles spécifications, spécialement celles qui sont floues ou encore difficiles à formaliser.

On peut résumer cette démarche pour le design d'un système informatique à trois niveaux d'activités:

1. établir l'information du domaine de la connaissance (requis);
2. structurer l'information dans le cadre d'une structure de données et d'un algorithme;
3. traduire l'algorithme et la structure de données en langage de programmation.

Les requis servent à capturer la connaissance du domaine tandis que l'analyse et le design servent à trouver les structures conceptuelles profondes. Les requis peuvent être en langage naturel, mais le design doit être plus formel.

Une autre hypothèse est que plus le design progresse, plus la complexité augmente, jusqu'au point où le designer ne peut conserver qu'une petite portion du design en mémoire à court terme ou devant lui sous forme de schéma ou dessin. En conséquence, un agenda est formé par ce qui peut être vu et remémoré. Une planification est nécessaire.

De plus en plus on tente, du moins en informatique, de concevoir des outils (CASE: Computer Aided Software Design) susceptibles d'aider à la phase de design. En se basant sur notre compréhension actuelle des processus cognitifs, voici les principales caractéristiques de ces outils:

- assister à l'organisation des activités;
- ne pas imposer une structure de planification hiérarchique, mais plutôt permettre aux designers, lorsqu'ils suivent un plan, de dévier et même de l'abandonner;

- supporter le retour au plan, mais ne pas le présupposer et encore moins l'imposer;
- assister les activités de représentation;
- permettre au designer de travailler à différents niveaux de détails et d'abstraction;
- produire plusieurs représentations visuelles, iconiques, ou symboliques adaptées au designer suivant son niveau d'expertise et adaptées aux différents points de vue;
- permettre un changement facile du modèle (de la représentation problème vs solution);
- supporter le passage entre les niveaux de représentation et les points de vue;
- assister à la gestion des limites de la mémoire humaine en offrant des affichages et des facilités pour la présentation en parallèle d'information intra et inter niveaux;
- présenter les contraintes de la solution;
- maintenir des traces des tâches abandonnées ou interrompues sur lesquelles on peut vouloir retourner;
- assister le designer dans la réutilisation du design.

5.2 Plans

En général, les plans sont des structures de connaissances qui servent à organiser la connaissance en fonction de différents critères. Les concepts et les formalismes sont variés et il n'existe pas réellement de consensus sur la meilleure façon d'exprimer les plans. *127, 142*

Un plan est une représentation susceptible de guider l'activité du sujet. Les propriétés des plans sont l'anticipation et la simplification. Un plan peut être sommaire, non linéaire et non séquentiel. Un plan n'est pas nécessairement procédural, il peut être constitué d'une succession de sous-buts marquant quelques grandes étapes à parcourir avant d'atteindre le but final, sans que chaque étape ne corresponde nécessairement à un traitement bien défini. Le caractère anticipateur des plans vise à rendre compte des phénomènes d'attentes, devant une situation, grâce au stockage et à la récupération en mémoire d'expérience passée.

On doit distinguer entre le plan d'un produit final qui est hiérarchique et les plans de l'activité qui sont de nature opportuniste. L'activité de solution de problème est effectuée mentalement.

Le plan est composé de trois principaux aspects:

1. *nature heuristique*: oriente rapidement l'activité vers les voies les plus prometteuses, sans nécessiter une analyse fine et fastidieuse de la situation;

2. *utilisation optimale de la mémoire:* ne conserve que les propriétés cruciales d'un objet, par abstraction des détails non pertinents à un moment donné du déroulement de l'activité;

3. *élévation du niveau de contrôle de l'activité*: permet d'exploiter des situations, lors de la mise en oeuvre des propriétés, qu'un niveau de description trop fin ne permet pas d'expliciter. *143*

Un plan déclaratif représente des structures relationnelles statiques. Un plan opérationnel représente des structures d'opérations. Le principe de base de la formalisation des plans opérationnels est la définition d'une hiérarchie d'espaces d'abstraction par opposition à un espace de base qui est celui de l'exécution.

La notion de plan est également utilisée dans les études sur la compréhension des programmes.

5.3 Activité de planification

La planification peut être prospective, en construisant les étapes à parcourir à partir de l'état initial, ou rétrospective en procédant à partir du but ou l'état final.

Le mécanisme de la planification comporte deux fonctions complémentaires:

- l'élaboration de plans qui correspond à la fonction ascendante de l'activité,
- l'utilisation de plans qui relève de la fonction descendante de l'activité.

L'esprit humain peut fonctionner dans un environnement de connaissance imprécise. Il peut facilement extrapoler à partir d'éléments de connaissance. Cette aptitude est nécessaire dans un contexte d'économie cognitive. On n'a pas besoin de tout comprendre et de tout connaître pour être fonctionnel. Contrairement à l'ordinateur actuel où aucune extrapolation n'est permise, tous les éléments de connaissance doivent être définis et encodés.

Dans plusieurs domaines, la connaissance est incertaine (elle est basée sur des hypothèses qui sont plus ou moins possibles). Donc, une représentation réaliste de la connaissance doit avoir une partie incertaine comme base de sa structure. La connaissance doit se valider par une suite de rétroactions successives.

On a besoin d'un mécanisme pour naviguer dans un espace de connaissance incertaine. On ne doit formaliser la connaissance que lorsqu'elle est certaine sinon, cela cause plus de problèmes que de solutions.

Un processus de design renferme une composante importante de planification. Une planification structurée hiérarchiquement va, au minimum, contraindre le concepteur et va probablement le handicaper.

Lorsqu'un humain pense à un système, il existe un environnement global défini par le modèle mental, mais l'humain ne peut être, par ailleurs, que dans un état local caractérisé par les une à quatre propositions qu'il peut considérer. Il doit donc se déplacer

d'un état local à un autre. Il peut faire ceci de façon heuristique et arbitraire ou encore en suivant un plan. Ce plan peut être rigoureux ou opportuniste. *136*

Les plans sont supposés exprimer la structure représentative des connaissances possédées par le concepteur. Une déviation du plan survient lorsque le concepteur choisit, volontairement ou non, une alternative à l'action planifiée et proposée. On note que, premièrement, le concepteur progresse d'un comportement systématique à un comportement opportuniste à mesure que le design évolue et que, deuxièmement, il ne conserve pas toujours son design balancé. *120*

5.3.1 Systématique

Une approche de planification systématique est observée lorsque le concepteur dispose de toute la connaissance requise pour effectuer le design. La conclusion tirée de plusieurs études est que les programmes sont conçus de façon descendante et par expansion avant. Dans ce cadre, les experts présentent d'abord un développement en largeur (breadth-first) tandis que les novices peuvent concevoir seulement aux niveaux des détails donc, présentent en premier un comportement en profondeur (depth-first).

Si la connaissance ne peut être trouvée, une solution doit être créée par expansion. On observe alors une approche ascendante et par expansion arrière. *43.*

5.3.2 Opportuniste.

Souvent, le concepteur décrit son activité comme suivant une planification structurée et hiérarchisée, mais son activité actuelle, c'est-à-dire celle qui est observée, est organisée de façon opportuniste. Le concepteur suit son plan aussi longtemps qu'il ne perçoit pas une action plus opportune. L'analyse qualitative des observations confirme l'hypothèse que l'activité de spécification est organisée de façon très opportune. Le critère le plus important dans le choix d'une action est son coût cognitif relatif. Dans l'évaluation du coût cognitif d'une action, le navigateur considère les facteurs interreliés suivants si possible: *120, 30*

1. la disponibilité d'un schéma pour exécuter l'action;

2. la disponibilité de l'information;

3. la difficulté relative de l'action.

La démarche est opportuniste parce qu'à chaque point dans le processus, la décision courante du concepteur et ses observations suggèrent plusieurs opportunités pour le développement de plans. Les décisions subséquentes découlent des opportunités choisies. Chaque décision est motivée par une ou deux décisions précédentes immédiates plutôt que par un programme exécutif de haut niveau. *48*

Une forme particulière de planification opportuniste survient lorsque le concepteur utilise un élément de donnée du problème pour créer un élément de planification. Lorsque plusieurs éléments ont été identifiés, une tentative est effectuée pour les intégrer dans une planification cohérente. Cette approche est appelée planification coïncidente (serendipitous). Des solutions partielles sont reconnues à différents niveaux de détail ou d'abstraction et sont combinées. *42*

5.3.3 Expérience

Il a été suggéré que la variabilité dans le design décroît avec l'expertise. Certains auteurs avancent que les plans sont inhérents à la nature du problème et non dépendants des caractéristiques du langage. Alors, le développement de tels plans dépend de deux choses: premièrement de l'expérience que le concepteur a du problème et deuxièmement de l'habileté du concepteur d'associer une solution qu'il possède déjà à ce problème.

Toutefois la notion de plan à elle seule semblerait être une structure trop simple pour expliquer directement la différence entre les performances des concepteurs novices et expérimentés.

Des études ont démontré que les représentations des experts possèdent quatre caractéristiques abstraites:

1. elles sont hiérarchiques et à plusieurs niveaux;

2. elles ont une relation explicite entre les différents niveaux;

3. elles sont fondées sur la reconnaissance de patrons de base;

4. elles sont internement très bien connectées.

Ce qui caractérise un modèle mental c'est le domaine de connaissance utilisé.

Chez l'expert on note le comportement suivant pour solutionner un problème. Premièrement, il se forme un modèle mental. Il tente, par la suite, de simuler mentalement ce modèle. S'il n'a pas suffisament d'expérience avec l'objet cible, il développera des contraintes à propos du modèle devenant ainsi suffisament spécifique pour pouvoir effectuer une simulation. *56*

On note les étapes suivantes:

1) formation d'un modèle mental;

2) simulation du modèle mental;

3) expansion systématique du modèle mental;

4) représentation des contraintes;

5) retrait d'étiquettes, ou d'identificateurs, pour les plans;

6) prise de notes.

Les experts organisent leur connaissance de façon plus abstraite et plus profonde. Cette connaissance tend à être plus procédurale que déclarative. Cette connaissance est le résultat d'expériences vécues et non d'informations obtenues à partir de livres ou d'autres individus. La démarche de l'expert est plus souvent le résultat de situations qu'il reconnaît plutôt que le résultat d'une analyse. Ce type de connaissance procédurale implicite est très difficile à communiquer. *103*

6. Conclusion

Cette étude illustre les principaux courants d'idée qui gouvernent notre compréhension de la connaissance et de son traitement. Ces sujets ont été traités aussi bien en psychologie cognitive qu'en intelligence artificielle. Notre intérêt se situe surtout du côté des modèles de la psychologie cognitive. Notre objectif est de mieux comprendre le processus de développement en génie logiciel et de mieux évaluer la pertinence des méthodologies de design. Le principal agent est l'humain. Il ne s'agit pas de concevoir des machines qui puissent traiter de l'information, mais de mieux comprendre comment l'humain peut améliorer son rôle dans le développement du logiciel.

Le développement du logiciel est essentiellement un processus cognitif résultant dans la formalisation d'une représentation artificielle d'une situation. On voit que ce traitement de l'information doit se faire progressivement par un mécanisme d'essais et d'erreurs ou opportuniste.

Lorsque l'information à traiter est considérable, la démarche peut facilement devenir chaotique. L'esprit se perd parce qu'il est incapable de circonscrire correctement le domaine de connaissance. Mesurer le niveau de chaos dans la démarche serait un indicateur de la qualité du design ou du niveau d'expertise.

On constate également qu'une approche de spécifications formelles n'est possible que lorsque le modèle mental est bien défini. Une formalisation prématurée peut nuire au processus de convergence du modèle mental.

Le génie logiciel prend toute sa dimension dans le travail d'équipe. Le travail d'équipe est nécessaire lorsque le problème est trop complexe pour être solutionné par une seule personne dans un temps raisonnable.

Le principal problème du développement en équipe est la communication et la coordination, c'est-à-dire acquérir et partager les connaissances du domaine et trouver un schéma effectif organisationnel.

Il a été trouvé qu'environ un tier du temps total est pris pour clarifier ce qu'on voulait dire dans le cas d'une question spécifique. L'effort d'échange diminue au fur et à mesure que le projet se spécifie. *103*

Les techniques d'inspection et de revue sont des mécanismes privilégiés pour assurer l'uniformité des représentations schématiques (en effet, puisqu'une des difficultés du travail d'équipe est de vérifier que tous les intervenants ont les mêmes valeurs de variables des schémas.) Dans un groupe qui se partage les tâches, cet aspect de la conformité des variables des schémas n'est pas apparent. Il faut s'assurer que les schémas dont on discute lors d'une rencontre sont du même type (topique ou épisodique) pour tous les intervenants.

La difficulté, en génie logiciel, est le passage d'un système de niveaux supérieurs de connaissance à une représentation phrastique, le passage d'une représentation schématique à une représentation propositionnelle.

Pour informatiser une tâche, on doit, premièrement, préciser le modèle mental et, deuxièmement, traduire ce modèle mental en un ensemble de représentations propositionnelles. Souvent, les deux tâches se font simultanément puisqu'on utilise la démarche propositionnelle pour préciser le modèle mental. Dans quelles circonstances cette approche est-elle souhaitable?

L'approche propositionnelle présente peut-être une avenue intéressante pour le développement du logiciel (cette approche étant la plus naturelle chez l'humain.)

Le processus d'informatisation peut être représenté comme la description (requis, exigences) d'une tâche en un ensemble de propositions (compter les propositions plutôt que les points de fonction) que l'on doit traduire dans un langage de programmation. Il existe plusieurs difficultés:

- s'assurer que l'ensemble des propositions (exigences) est complet;
- s'assurer qu'il est cohérent;

- identifier les interactions entre les propositions;
- obtenir un modèle propositionnel des requis;
- traduire ces propositions en un langage de programmation;
- avoir une approche qui permet de traiter seulement de trois à quatre propositions.

Pour conclure ces travaux nous pouvons formuler l'hypothèse suivante et son corollaire:

HYPOTHÈSE:

> dans tous les cas, la démarche cognitive est la même parce que l'humain n'a qu'un seul type de démarche cognitive.

COROLLAIRE:

> toutes les méthodes de design ne sont que le reflet d'une composante d'une démarche cognitive qui est unique et identique pour tous.

7. Remerciement

Ce travail a été réalisé dans le cadre d'une période sabbatique. Plusieurs des concepts et notions ont pu être approfondis grâce à des discussions stimulantes. Je remercie ceux qui ont généreusement et patiemment consacrés leur temps à m'aider dans ce cheminement aux frontières de la psychologie et de l'informatique. Je tiens a souligner la contribution particulière de Thomas Green (Cambridge, U.K.), de Françoise Détienne (INRIA, Paris) et Robert Rist (Australie).

Ce travail a impliqué plusieurs déplacements pour participer à des sessions de travail et surtout pour rencontrer les gens intéressés par ce domaine, approfondir leurs idées et comprendre leurs démarches. Je suis particulièrement reconnaissant au CGLA-CRIM (Centre de génie logiciel appliqué) pour leur aide financière.

8. Bibliographie

Cette bibliographie est gérée automatiquement. Les numéros sont des étiquettes séquentielles qui n'ont aucune signification autre que l'ordre séquentiel d'entrée dans la base de données. Ils ne sont pas liés à l'ordre chronologique de consultation de la référence, ni à l'ordre alphabétique des auteurs, ni à l'importance ou au contenu des articles. Ces numéros sont utilisés pour référer les articles correspondants dans le texte. Le regroupement des textes d'un auteur donné est accidentel. Par exemple, les textes de Adelson *et al* sont regroupés de 53 à 57, par contre certains textes de Détienne *et al* se retrouvent à 16, 22, 67, 100, 109, etc. Une liste des références par ordre alphabétique est disponible.

Certaines références, comme les livres renferment plusieurs sujets. Dans ce cas, nous référons à une page en particulier en faisant suivre le numéro de la référence du numéro de la page. Par exemple, 139-364 signifie que nous référons à la page 364 de la référence 139.

1 Anderson, R.J., Heath, C.C., Luff, P., Moran, T.P. The social and the cognitive in human-computer interaction. International Journal of Man-Machine Studies, Vol. 38 Issue: 6 p. 999-1016, June 1993.

2 Card, S.K., Moran, T.P., Newell, A. The psychology of human-computer interaction. Human-Computer interaction, 1983

3 Curtis, B., Forman, I., Brooks, R.E., Soloway, E.M., Ehrlich, D. Psychological perspectives for software science. Information Processing and Management, Vol.: 20 Issue: 1 p. 81-96, 1984

4 Curtis, B., Soloway, E.M., Brooks, R.E., Black, J.B., Ehrlich, K., Ramsey, H.R. Software psychology: the need for an interdisciplinary program. Human-Computer interaction, 1987

5 Davies, S.P. Knowledge restructuring and the acquisition of programming expertise. International Journal of Human-Computer Studies, Vol.: 40 Issue: 4 p. 703-26 April 1994.

6 Davies, S.P. The structure and content of programming knowledge: disentangling training and language effects in theories of skill development. International Journal of Human-Computer Interaction, Vol.: 5 Issue: 4 p. 325-46 Oct-Dec 1993.

7 Davies, S.P. Models and theories of programming strategy. International Journal of Man-Machine Studies, Vol.: 39 Issue: 2 p. 237-67 Aug. 1993.

8 Davies, S.P. Knowledge restructuring and the development of expertise in computer programming. Knowledge restructuring and the development of expertise in computer programming, 1992

9 Davies, S.P. The role of notation and knowledge representation in the determination of programming strategy: a framework for integrating models of programming behavior. Cognitive Science, Vol.: 15 Issue: 4 p. 547-72 Oct-Dec 1991.

10 Davies, S.P. Skill levels and strategic differences in plan comprehension and implementation in programming. Proceedings of the Fifth Conference of the British Computer Society Human-Computer Interaction Specialist Group, p. 487-502. 1989

11 Davies, S.P., Findlay, J.M., Lambert, A.J. The perception and tracking of state changing in complex systems. Proceedings of the Third International Conference on Human Computer Interaction., Vol. II, p. 510-17. 1989

12 Davies, S.P. Characterizing the program design activity: neither strictly top-down nor globally opportunistic. Behavior and Information Technology, Vol.: 10 Issue: 3 p. 173-90 May-June 1991.

13 Davies, S.P. The nature and development of programming plans. International Journal of Man-Machine Studies, Vol.: 32 Issue: 4 p. 461-81 April 1990.

14 Davies, S.P. Plans, goals and selection rules in a comprehension of computer programs. Behavior and Information Technology, Vol.: 9 Issue: 3 p. 210-14 May-June 1990.

15 Detienne, F. Reasoning from a schema and from an analog in software code reuse. Proceedings of the 4th Workshop on Empirical Studies of Programmers, p. 5-22. 1991

16 Detienne, F. Constraints on design: language, environment and code representation. Proceedings of the NATO Advanced Research Workshop, p. 69-79. 1994

17 Detienne, F. Acquiring experience in object-oriented programming: effect on design strategies. Proceedings of the NATO Advanced Research Workshop, p. 49-58. 1993

18 Detienne, F., Soloway, E.M. Program understanding as an expectation-driven activity. Proceedings of the Third International Conference on Human Computer Interaction., Vol. II p. 28-34. 1989

19 Detienne, F., Soloway, E.M. An empirically-derived control structure for the process of program understanding. International Journal of Man-Machine Studies, Vol.: 33 Issue: 3 p. 323-42 Sept. 1990.

20 Detienne, F. The cognitive ergonomics approach in computer programming. Toulouse '89. Second International Workshop. Proceedings, p. 167-79. 1989

21 Detienne, F., Soloway, E.M. Empirically-derived control structure for the process of program understanding. Empirically-derived control structure for the process of program understanding, 1988

22 Detienne, F. A review of psychological studies on the understanding of computer programs. Technique et Science Informatiques, Vol.: 8 Issue 1 p. 5-20 1989.

23 Detienne, F. Program understanding and knowledge organization the influence of acquired schemata. Cognitive ergonomics understanding, learning and designing human-computer interaction, p. 245-256, 1990

24 Ehrlich, K., Soloway, E.M. An empirical investigation of the tacit plan knowledge in programming. Human factors in computer systems, p. 113-133. 1984

25 Green, T.R.G. The cognitive dimension of viscosity: a sticky problem for HCI. Proceedings of the IFIP TC 13 Third International Conference, p. 79-86. 1990

26 Green, T.R.G. Cognitive dimensions of notations. Proceedings of the Fifth Conference of the British Computer Society Human-Computer Interaction Specialist Group, p. 443-60. 1989

27 Guindon R. The process of knowledge discovery in system design. Proceedings of the Third International Conference on Human Computer Interaction., Vol. II p. 727-34. 1989

28 Guindon R. Users request help from advisory systems with simple and restricted language: effects of real-time constraints and limited shared context. Human-Computer Interactions, Vol.: 6 Issue: 1 p. 47-75 1991.

29 Guindon R. Designing the design process: exploiting opportunistic thoughts. Human-Computer Interactions, Vol.: 5 Issue: 2-3 p.305-44, 1990.

30 Guindon R. Knowledge exploited by experts during software system design. International Journal of Man-Machine Studies, Vol.: 33 Issue: 3 p. 279-304 Sept. 1990.

31 Guindon R., Krasner, H., Curtis, B. Cognitive processes in software design: activities in early, upstream design. Proceedings of the Second IFIP Conference, p. 383-7. 1987

32 Guindon R. A cognitive study of high-level system design: implications of opportunistic behaviors for a theory of design. NSF Engineering Design Research Conference, p. 145-59. 1989

33 Guindon R. Control of cognitive processes during software design: what tools are needed ?. Human Factors in Computing Systems conference Proceedings, 263-268. 1988

34 Lee, A., Pennington, N. The effects of paradigm on cognitive activities in design (programming). International Journal of Human-Computer Studies, Vol.: 40 Issue: 4 p. 577-601 April 1994.

35 Letovsky, S.I. Cognitive processes in program comprehension (1). Proceedings of the 1st Workshop on Empirical Studies of Programmers, p. 58-79. 1986

36 Letovsky, S.I. Cognitive processes in program comprehension (2). Journal of Systems and Software, Vol.: 7 Issue: 4 p. 325-339 Dec 1987.

37 Letovsky, S.I. A cognitive analysis of a code inspection. Proceedings of the 2nd Workshop on Empirical Studies of Programmers, p. 231-247. 1987

38 Pennington, N., Nicolich, R. Transfer of training between programming subtask: is knowledge really use specific ?. Proceedings of the 4th Workshop on Empirical Studies of Programmers, p. 156-76. 1991

39 Pennington, N. Comprehension strategies in programming. Proceedings of the 2nd Workshop on Empirical Studies of Programmers, p. 100-113. 1987

40 Rist, R.S. Search through multiple representations (programming theory). Proceedings of the NATO Advanced Research Workshop, p. 165-76. 1994

41 Rist, R.S. Knowledge creation and retrieval in program design: a comparison of novice and intermediate student programmers. Human-Computer Interactions, Vol.: 6 Issue: 1 p. 1-46 1991.

42 Rist, R.S. Variability in program design: the interaction of process with knowledge. International Journal of Man-Machine Studies, Vol.: 33 Issue: 3 p. 305-22 Sept. 1990.

43 Rist, R.S. Schema creation in programming. Cognitive Science, Vol.: 13 Issue: 3 p. 389-414 July-Sept. 1989.

44 Rist, R.S. Plans in programming: definition, demonstration, and development. Proceedings of the 1st Workshop on Empirical Studies of Programmers, p. 28-47. 1986

45 Soloway, E.M., Ehrlich, K. Empirical studies in programming knowledge. Readings in artificial intelligence and software engineering, p. 507-521. 1986

46 Soloway, E.M., Ehrlich, K. Empirical studies of programming knowledge. IEEE Transactions on Software Engineering, Vol.: SE-10 Issue: 595-609. 1984

47 Soloway, E.M., Ehrlich, K. Empirical studies of programming knowledge (2). Software Reusability: Vol. II, application and experience, p. 235-267. 1989

48 Visser, W. Organization of design activities: opportunistic, with hierarchical episodes. Interacting with Computers, Vol.: 6 Issue: 3 p. 239-74 Sept 1994.

49 Visser, W. Planning and organization in expert design activities. Proceedings of the NATO Advanced Research Workshop, p. 25-39. 1994

50 Visser, W. Design organization: there is more to expert knowledge than is dreamed of in the planner's philosophy. Design organization: there is more to expert knowledge than is dreamed of in the planner's philosophy, . 1992

51 Visser, W. Designer's activities examined at three levels: organization, strategies and problem-solving processes. Knowledge-Based Systems, Vol.: 5 Issue: 1 p. 92-104 March 1992.

52 Visser, W. Evocation and elaboration of solutions: different types of problem-solving actions. an empirical study on the design of an aerospace artifact. Proceedings of the Third COGNITIVA Symposium, p. 161-8. 1991

53 Adelson, B. Problem solving and the development of abstract categories in programming languages. Memory and Cognition, Vol.: 9 Issue: 4, 422-433. 1981

54 Adelson, B. When novices surpass experts: The difficulty of a task may increase with expertise. Journal of Experimental Psychology: Learning, Memory and Cognition, Vol.: 10 Issue: 3, 483-495, 1984

55 Adelson, B., Soloway, E.M. A model of software design. Human-Computer Interactions, 342, Yale University. 1985

56 Adelson, B., Soloway, E.M. The role of domain experience in software design. IEEE Transactions on Software Engineering, Vol.: 11 Issue: 11, 1351-1860. 1985

57 Adelson, B. Comparing natural and abstract categories: a case study from computer science. Cognitive Science, Vol.: 9, 417-430. 1985

58 Alba, J.W., Hasher, L. Is memory schematic ?. Psychological Bulletin, Vol. 93, Issue: 2, 203-231. 1983

59 Arblaster, A.T., Sime, M.E., Green, T.R.G. Jumping to some purpose. The Computer Journal, Vol.: 22 Issue: 2, 105-109. 1978

60 Arblaster, A.T. Human factors in the design and use of computing languages. International Journal of Man-Machine Studies, Vol.: 17, 211-224, 1982

61 Atwood, M.E., Ramsey, H.R. Cognitive structures in the comprehension and memory of computer programs: an investigation of computer program debugging. Technical Report, TR-78-A21. 1978

62 Black, J.B., Kay, D.S., Soloway, E.M. Goal and plan knowledge representation : From stories to text editors and programs. Book, . 1986

63 Brooks, R.E. A theoretical analysis of the role of documentation in the comprehension of computer programs. Human Factors in Computing Systems conference Proceedings, 15-17, , 125-129, 1982

64 Brooks, R.E. Towards a theory of the comprehension of computer programs. International Journal of Man-Machine Studies, 18, 543-554. 1983

65 Curtis, B. A review of human factors research on programming languages and specifications. Human factors in computer systems, 15-17, 212-218. 1982

66 Curtis, B. Fifteen years of psychology in software engineering: individual differences and cognitive science. Proceedings of the Seventh Conference on Software Engineering, 97-106. 1984

67 Detienne, F. Analyse exploratoire de l'activité de compréhension de programmes informatiques. Proceedings AFCET, 161-172. 1984

68 Detienne, F. La compréhension de programmes informatiques par l'expert : un modèle en termes de schémas. Thèse de doctorat, . 1986

69 Detienne, F. Une application de la théorie des schémas à la compréhension de programmes. Le Travail Humain, 1988.

70 Gilmore, D.J., Green, T.R.G. The comprehensibility of programming notations. Proceedings of the congress "Interact", 1984.

71 Gilmore, D.J., Green, T.R.G. Comprehension and recall of miniature programs. International Journal of Man-Machine Studies, 21, 31-48. 1984

72 Green, T.R.G. Conditional program statements and their comprehensibility to professional programmers. Journal of Occupational Psychology, 50, 93-109. 1977

73 Hoc, J.M. L'Articulation entre la description de la tâche et la caractérisation de la conduite dans l'analyse du travail. Bulletin de Psychologie, XXXIII (334), 207-212. 1979

74 Hoc, J.M. L'Étude psychologique de l'activité de programmation : une revue de question. Technique et Science Informatiques, 1, (5), 383-392. 1982

75 Hoc, J.M. Psychologie cognitive et génie logiciel. Intellectica, 1, (1), 1982, 21-36. 1982

76 Hoc, J.M. Une méthode de classification préalable des problèmes d'un domaine pour l'analyse des stratégies de résolution : La programmation informatique chez les professionnels. Le Travail Humain, 46 (3), 205-217. 1983

77 Hoc, J.M., Samurçay, R. La psychologie de la programmation informatique. Monographie, AFCET. 1986

78 Hoc, J.M. L'Apprentissage de l'utilisation des dispositifs informatiques par analogie à des situations familières. Psychologie Française, Numéro spécial, 1988.

79 Johnson, W.L., Soloway, E.M. PROUST: knowledge-based program understanding. IEEE Transactions on Software Engineering, SE-11 (3), 267-275. 1985

80 Laughery, K.R., Laughery, K.R. Human factors in software engineering : A review of the literature. The Journal of Systems and Software, 5, 3-14. 1985

81 Letovsky, S.I., Soloway, E.M. Delocalized plans and program comprehension. IEEE Software, 3, (3), 41-49. 1986

82 Littman, D.C., Pinto, J., Letovsky, S.I., Soloway, E.M. Mental models and software maintenance. Proceedings of the 1st Workshop on Empirical Studies of Programmers, 80-89. 1986

83 Mynatt, B.T. The effect of semantic complexity on the comprehension of program modules. International Journal of Man-Machine Studies, 21, 91-103. 1984

84 Norcio, A.F. Indentation, documentation and program comprehension. Human Factors in Computing Systems, 15-17, 118-120. 1982

85 Pennington, N. Stimulus structures and mental representations in expert comprehension of computer programs. Cognitive Psychology, 19, 295-341. 1987

86 Rumelhart, D.E. Schemata : The building blocks of cognition. Technical Report, 1978.

87 Samurçay, R. Plans et schémas de programmation. Psychologie Française, 1988.

88 Sheil, B.A. The psychological study of programming. Computing Surveys, 13, (1), 1981, 101-120. 1981

89 Shneiderman, B. Software psychology. Human factors in computer and information systems, 1980.

90 Soloway, E.M., Ehrlich, K., Bonar, J. Tapping into tacit programming knowledge. Human factors in computer systems, 15-17, 52-57. 1982

91 Soloway, E.M., Bonar, J., Ehrlich, K. Cognitive strategies and looping constructs : an empirical study. Communication of the ACM, 26, (11), 853-860. 1983

92 Weiser, M., Shertz, J. Programming problem representation in novice and expert programmers. International Journal of Man-Machine Studies, 19, 391-398. 1983

93 Widowski, D., Eyferth, K. Representation of computer programs in memory. Proceedings of the Third International Conference on Human Computer Interaction., 15-19. 1986

94 Brooks, R.E. Towards a theory of the cognitive processes in computer programming. International Journal of Man-Machine Studies, 9, 737-751. 1977

95 Hoc, J.M., Green, T.R.G., Samurçay, R., Gilmore, D.J. Psychology of Programming. Book, Academic Press. 1990

96 Walker, W.H., Kintsch, W. Automatic and strategic aspects of knowledge retrieval. Cognitive Science, 9, 261-283. 1985

97 Rosson, M.B., Alpert, S.R. The cognitive consequences of object-oriented design. Technical Report, RC 14191 New York, IBM. 1988

98 Anderson, R.J. Psychological status of the script concept. American Psychologist, 36 (7), 715-729, 1981

99 Detienne, F. Expert Programming knowledge: A Schema-based Approach., Psychology of Programming. 1990

100 Detienne, F. Difficulties in designing with an object-oriented language: an empirical study. Proceedings, Interact '90 3rd IFIP conf. on HCI, Cambridge, UK,. 1990

101 Visser, W., Hoc, J.M. Expert Software Design Strategies., Psychology of Programming. 1990

102 Von Mayrhauser A., Vans A. Marie Program Comprehension During Software Maintenance and Evolution. IEEE Computer, pp. 44-55. August 1995

103 Herbsleb J.D., Klein H., Olson G. M., Brunner H., Olson J., Harding J. Object-Oriented Analysis and Design in Software Project Teams. Human-Computer Interactions, Vol. 10, pp. 249-292. 1995

104 Davies, S.P., Gilmore, D.J., Green, T.R.G. Are Objects that Important? Effects of expertise and familiarity on classification of OO Code. Human-Computer Interactions, Vol. 10, pp. 227-248. 1995

105 Pennington, N., Lee, A., Rehder B.N/A Cognitive Activities and levels of Abstraction in procedural and OO Design.. Human-Computer Interactions, Vol. 10, pp. 171-226. 1995

106 Bürkle U., Gryczan G., Züllighoven H. Object-Oriented System development in a Banking project: Methodology, Experience, and Conclusions. Human-Computer Interactions, Vol. 10, pp293-336. 1995

107 Detienne, F., Rist, R.S. Introduction to this special issue on empirical Studies of Object-Oriented design.. Human-Computer Interactions, Vol. 10, pp. 121-128. 1995

108 Curtis, B. Objects of our desire: empirical research on Object-Oriented development.. Human-Computer Interactions, Vol. 10, pp337-344. 1995

109 Detienne, F. Design Strategies and knowledge in Object-Oriented programming: effects of Experience. Human-Computer Interactions, Vol. 10, pp. 129-169. 1995

110 Ericsson A., Hastie R. Contemporary approaches to the study of Thinking and Problem Solving., Thinking and Problem Solving, Sternberg. 1994

111 McNamara T. P. Knowledge representations, Thinking and Problem Solving, Sternberg. 1994

112 Dominowski R.L., Bourne L.E. History of Research on thinking and Problem Solving, Thinking and Problem Solving Sternberg. 1994

113 Rips Lance J. Deduction and its cognitive Basis, Thinking and problem solving, Sternberg. 1994

114 Sternberg Robert J. Intelligence, Thinking and Problem Solving , Sternberg. 1994

115 Ross Brian H., Spalding Thomas L.N/A Concepts and categories, Thinking and Problem Solving, Sternberg. 1994

116 Lubart Todd Creativity, Thinking and Problem solving, Sternberg. 1994

117 Hunt Earl Problem Solving, Thinking and Problem Solving, Sternberg. 1994

118 Gerrig Richard J., Banaji M. R. Language and Thought, Thinking and Problem Solving, Sternberg. 1994

119 Bisanz J., Bisanz G. L., Korplan C.A. Inductive reasoning, Thinking and Problem solving, Sternberg. 1994

120 Visser, W. More or less following a plan during desing: opportunistic deviations in specification. International Journal of Man-Machine Studies, 33, 247-278. 1990

121 Simon Herbert A. Invariants of Human Behavior. Annual revue Psychology, 41 1-19. 1990

122 Simon Herbert A. Information Processing Models of Cognition. Annual revue Psychology, 30, 363-396. 1979

123 Wiedenbeck Susan, Fix Vikki Characteristics of the mental representations of novice and expert programmers: an empirical study. International Journal of Man-Machine Studies, 39, 793-812. 1993

124 Wiedenbeck Susan The initial stage of program comprehension. International Journal of Man-Machine Studies, 35, 517-540. 1991.

125 Khali Omar E., Clark Jon D. The influence of programmers'cognitive complexity on program comprehension and modification. International Journal of Man-Machine Studies, 31, 219-236. 1989

126 Malhotra Ashok, Thomas John C., Carroll John, Miller Lance Cognitive processes in design. International Journal of Man-Machine Studies, 12, 119-140. 1980

127 Robertson Scott P., Yu Chiung-Chen Common cognitive representations of program code across tasks and languages. International Journal of Man-Machine Studies, 33, 343-360. 1990

129 Koubek Richard J., Salvendy Gavriel, Dunsmore Hubert, Lebold William Cognitive issues in the process of software development: review and reappraisal. International Journal of Man-Machine Studies, 30, 171-191. 1989

130 Sutcliffe A.G., Maiden N.A.M Analysing the novice analyst: cognitive models in software engineering. International Journal of Man-Machine Studies, 36, 719-740. 1992

131 Shneiderman, B. Measuring computer program quality and comprehension. International Journal of Man-Machine Studies, 9, 465-478. 1977

132 Finke R. A., Ward T. B., Smith S. Creative Cognition, MIT Press. 1992

133 Smith S., Ward T.B., Finke R. A. The creative Cognition Approach, MIT Press. 1995

134 Cleeremans A. Mechanisms of Implicit Learning, MIT Press. 1993

135 Peacocke C. A study of Concepts, MIT Press. 1992

136 Fagin R., Halpern J.Y., Moses Y., Vardi M. Y. Reasoning About Knowledge, MIT Press. 1995

137 Boden M. A. Dimensions of Creativity, MIT Press. 1994

138 Sternberg R. J., Davidson J.E. The Nature of Insight., MIT Press. 1995

139 Le Ny J-F., Gineste M-D. La Psychologie., Larousse. 1995

140 Suppes P., Pavel, M., Falmagne J.-Cl Representations and models in Psychology. Annual revue Psychology, 45: 517-544. 1994

141 Burkhardt J.M., Detienne, F. La réutilisation de solutions en conception de programmes informatiques. Monographie, INRIA. 1996

142 Holt R., W., Boehm-Davies D., Schultz A.C.N/A Mental representations of Programs for Student and Professional Programmers., 1996

143 Hoc, J.M. Psychologie cognitive de la planification., Presse Universitaire de Grenoble. 1987

ÉCOLE POLYTECHNIQUE DE MONTRÉAL
3 9334 00170938 3

École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec)
H3C 3A7