

# Abandon d'un plan hiérarchique dans une activité de conception

Willemien Visser

► **To cite this version:**

Willemien Visser. Abandon d'un plan hiérarchique dans une activité de conception. Pierre Chavance and François Gros. Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87, 1987, Paris, France. Centre d'études des systèmes et des technologies avancées (CESTA) (Paris), 1, pp.366-371, 1987, <<http://hal.inria.fr/inria-00075737/fr/>>. <hal-00641048>

**HAL Id: hal-00641048**

**<https://hal.inria.fr/hal-00641048>**

Submitted on 21 Nov 2011

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

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

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

## ABANDON D'UN PLAN HIERARCHIQUE DANS UNE ACTIVITE DE CONCEPTION

Willemien VISSER

Projet de Psychologie Ergonomique pour l'Informatique INRIA - Rocquencourt (France)

*Résumé.* Des observations ont été conduites sur la conception d'une machine-outil.

L'étude se distingue de précédentes études empiriques notamment sur les points suivants:

\* Nous avons observé en temps réel des concepteurs pendant leur activité de travail habituel.

\* Nous avons étudié plusieurs étapes dans le processus de conception d'un projet réel complexe, impliquant différents opérateurs.

Les résultats présentés sont centrés sur l'éventuel caractère hiérarchique de l'activité de conception.

1. Chaque opérateur décrit son activité sous la forme d'un plan hiérarchique. Il décompose son activité en composants de niveau différent qui seront traités dans un certain ordre. Ce plan ne traduit cependant pas l'activité réelle de l'opérateur. Celle-ci est plutôt opportuniste. La hiérarchie du plan est abandonnée. Des retours en arrière et des anticipations complètent les composants de conception mais y apportent également des modifications qui vont de la correction d'erreur à la remise en cause de décisions.

Nos résultats confirment ceux d'études antérieures conduites sur des tâches de conception simplifiées. Ils comportent cependant aussi un aspect nouveau: nous montrons que ces traitements qui caractérisent la conception ne s'appliquent pas seulement à l'activité de chaque opérateur dans son étape de conception, mais au processus de conception dans son ensemble. Ainsi, les spécifications établies antérieurement, par exemple, sont complétées et modifiées également.

2. Nous avons analysé les facteurs qui conduisent à ces abandons du plan hiérarchique. Les principaux processus psychologiques qui en ressortent sont présentés. Il s'agit de processus bien connus dans d'autres contextes de résolution de problème ou de diagnostic.

D'une part, ces résultats permettent alors de préciser dans quelle mesure l'activité de conception se rapproche et se différencie de ces autres activités.

D'autre part, la connaissance de ces processus et de leurs conditions spécifiques de déclenchement nous fournit des renseignements précieux pour la modélisation de l'activité de conception.

Ces résultats sont illustrés, à l'aide d'exemples.

## GIVING UP A HIERARCHICAL PLAN IN A DESIGN ACTIVITY

*Summary.* An observational study has been conducted on the design of a machine tool installation. This study is different from preceding empirical research, especially on the following points:

- \* We did real time observations on designers during their actual work activity.
- \* We have been studying several stages of the design process of a real complex project involving three different operators.

The results presented focus on the possible hierarchical nature of the design activity.

1. Each operator describes his activity as a hierarchical plan. He decomposes his activity in components at different levels, which are to be treated in a fixed order. The operator's real activity however, which is rather opportunistic, departs from this plan. The plan's hierarchy is given up: (a) components are abandoned before completion; (b) addenda and modifications are made on past and future components. The modifications go from error correction to challenging design decisions. These processing characteristics are in agreement with the results obtained in previous laboratory studies conducted on simplified design projects. They are however also more general: they apply not only to each operator's activity in his design stage, but also to the entire design process. So, specifications established in previous stages, for example, are completed and challenged by the operators observed.

2. We analysed the factors leading to these deviations from a hierarchical plan. The principal psychological processes among them are presented. They are processes well known to be used in other contexts of problem solving or diagnosis. So, on the one hand, these results permit to specify to which degree the design activity is similar to or different from these other activities. On the other hand, knowledge about these processes and their associated triggering conditions, provides us with precious information for the modeling of the design activity.

Illustrations of the results are presented.

## 1. Modèles théoriques et activité réelle de conception

Lorsque l'on demande à des responsables de bureaux d'étude ou de départements Méthodes de décrire le processus de conception dans leur entreprise, on obtient en général une liste d'étapes consécutives, bien séparées, dans chacune desquelles interviennent un ou plusieurs opérateurs différents. Nous présentons, dans le Tableau 1, la description du processus de conception telle qu'elle nous a été donnée par un responsable d'une entreprise de conception de logiciel.

- spécification
- conception préliminaire
- conception détaillée
- codage
- test
- intégration
- recette

Tableau 1. Les étapes théoriques du processus de conception dans une entreprise de développement de logiciel

Chaque étape doit donner lieu à la production d'un ou plusieurs documents. Ceux produits par les opérateurs dans une certaine étape constituent, pour leurs collègues chargés de l'étape suivante, un état du projet qui est censé ne plus être mis en question.

Le processus de conception qui ressort de ce genre de descriptions ressemble beaucoup à celui qui est préconisé ou prescrit par des méthodologies de conception.

Ainsi, dans MERISE, une méthode française de conception et de développement de systèmes d'information, le cycle de vie (de la conception à la maintenance) de tels systèmes "se caractérise par plusieurs étapes qui se situent nécessairement sur une échelle de temps". (Tardieu, Rochfeld & Colletti, 1986, p. 38) D'autre part, "l'utilisation de niveaux d'abstraction est absolument nécessaire [entre autres] pour ... ne prendre en compte qu'une classe de problème à chaque niveau" (id. p. 41). (pour des revues critiques de ces méthodes, cf. Carroll & Rosson, 1985; Robert, 1979).

L'activité de conception cependant telle qu'elle a été observée dans des études empiriques sur la conception de logiciel est bien différente de celle qui ressort de ces modèles théoriques (normatifs ou prescriptifs). Elle y apparaît comme non-hiérarchique (le traitement se fait ni complètement de haut en bas, ni complètement de bas en haut), procédant de façon cyclique, introduisant au fur et à mesure de nouveaux buts (cf. les revues de question citées ci-dessus).

Les études empiriques sur lesquelles sont basées ces conclusions ont cependant toutes été des études de laboratoire. Même si certaines ont bien eu comme sujets des concepteurs (débutants et/ou experts), ces études portaient, en général, sur des tâches de conception simplifiées et bien délimitées par rapport au travail réel de conception. En plus, ces études portaient sur des sujets travaillant

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

individuellement, tandis qu'un projet réel de conception est caractérisée par la collaboration avec d'autres et une dépendance plus ou moins grande (du travail fait par) des collègues.

Pour modéliser l'activité réelle de conception, s'imposent donc des études empiriques de cette activité sur des concepteurs travaillant sur des projets réels, donc complexes. Il est vrai que ce type d'étude demande, comparée avec des études de laboratoire, un investissement important en temps et que ses résultats peuvent être plus difficiles à traiter du fait des nombreux facteurs que l'on ne peut pas, comme au laboratoire, éliminer pour la durée de l'observation.

Cet article présente une étude longitudinale que nous avons conduite sur un projet de conception d'une installation industrielle automatisée commandée par programme informatique.

Etant donnée la division existante du travail, nous avons pu étudier plusieurs étapes consécutives de cette conception. Elles sont, comme dans toute entreprise de ce type, censées être successives et indépendantes. Or, à la suite des études auxquelles nous avons fait référence ci-dessus, nous nous demandons dans quelle mesure l'individu s'adapte au processus hiérarchique qui lui est imposé par la division du travail et/ou dans quelle mesure il transgresse les frontières des étapes.

L'étude présentée ici se faisant dans le cadre d'une recherche centrée sur la programmation (cf. Visser, 1985), nous avons observé des étapes qui nous paraissent (les plus) pertinentes dans cette perspective. A côté de l'étape de la programmation proprement dite, nous avons observé les deux étapes qui se trouvent, l'une en amont, l'autre en aval de celle-ci. Les trois étapes étudiées sont alors les suivantes:

- 1 - construction, par un mécanicien, d'une représentation fonctionnelle de la partie opérative de l'installation. Le schéma produit constitue le document principal de spécification pour la conception du programme de commande de l'installation
- 2 - construction de ce programme de commande par un électricien-automaticien (le "programmeur")
- 3 - mise au point du programme par un autre électricien-automaticien (le "metteur au point")

Notre étude se distingue alors des précédentes études empiriques notamment sur les points suivants:

- \* Nous avons observé en temps réel des concepteurs pendant leur activité de travail habituel.
- \* Nous avons étudié plusieurs étapes dans le processus de conception d'un projet réel complexe, impliquant différents utilisateurs.

## **2. Méthode utilisée**

Pendant treize semaines au total, nous avons conduit des observations dans une usine produisant des machines-outils commandées par programme informatique. Chacun des trois opérateurs a été observé pendant toute la durée de son travail. Nous leur avons demandé de procéder comme d'habitude, à une exception près: nous les avons priés d'énoncer, le plus possible, à haute voix, leurs pensées concernant leur travail.

Pour obtenir des données sur l'activité des opérateurs, nous avons pris des notes concernant:

- tout ce que l'opérateur écrivait et disait;
- l'ordre de production des différents documents, leur construction progressive et les modifications qui y sont apportées;
- les sources d'information consultées;
- les événements conçus (par nous) comme des indicateurs de la rencontre par l'opérateur de difficultés.

Pour le metteur au point, qui ne produisait pas seulement des documents mais qui intervenait également sur le procédé physique, s'y ajoutent des notes concernant ces interventions.

A côté des notes, nous avons recueilli toutes les traces matérielles de l'activité:

- les différentes versions des documents produits;
- les brouillons de (parties de) documents;
- les diagrammes et schémas que l'opérateur s'est construits.

### **3. Résultats**

Parmi les résultats que nous avons obtenus, nous présentons ici ceux centrés sur un aspect particulier de l'activité de conception observée, à savoir son éventuel caractère hiérarchique.

Nous verrons que l'opérateur prévoit une activité hiérarchique, mais que son activité réelle ne suit pas ce plan. Les principaux processus psychologiques qui conduisent à cet abandon seront présentés.

#### *3.1 Le plan de l'activité tel qu'il est décrit par l'opérateur*

Nous avons demandé aux opérateurs, soit avant (programmeur), soit avant et après (mécanicien et metteur au point), de nous décrire leur activité. Dans les deux conditions, chaque opérateur observé décrit son activité sous la forme d'un plan hiérarchique avec plus ou moins de niveaux.

Nous présentons ici, en exemple, le plan du mécanicien (v. le Tableau 2, page suivante).

La suite de buts et d'actions est rendue dans l'ordre de description par l'opérateur. Il s'agit donc d'une planification "de haut en bas" et "en profondeur d'abord".

Pour des causes que nous présentons ci-dessous, ce plan ne rend pas compte de l'activité réelle du mécanicien. Il en va de même pour les deux autres opérateurs. Les plans qu'ils décrivent pour présenter leur activité ne couvrent pas leur activité réelle. Plus généralement, les opérateurs ne suivent pas un parcours systématique à travers l'arbre représentant le plan de leur activité.

Ce texte est un pre-print de  
 Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
 Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

- + **Construire une représentation fonctionnelle**
- + Définir le Cycle Général
  - + Définir la première fonction
    - + Déterminer sa première fonction
    - + Déterminer ses descripteurs
      - + Déterminer son identificateur (id)  
Inscrire sa valeur dans la colonne id
      - + Déterminer sa durée (d)  
Inscrire sa valeur dans la colonne d
      - + Déterminer son ordre de départ (dép)  
Inscrire sa valeur dans la colonne dép
      - + Déterminer ses conditions de fin (fin)  
Inscrire sa valeur dans la colonne fin
  - + Définir la deuxième fonction
    - + Déterminer sa deuxième fonction
    - + Déterminer ses descripteurs
    - ...
  - ...
  - + Définir la n<sup>ième</sup> fonction
    - ...
- + Définir le Cycle de la première fonction
  - + Définir la première opération
    - + Déterminer sa première opération
    - + Déterminer ses descripteurs
      - + Déterminer son identificateur (id)  
Inscrire sa valeur dans la colonne id
      - + Déterminer sa durée (d)  
Inscrire sa valeur dans la colonne d
      - + Déterminer son ordre de départ (dép)  
Inscrire sa valeur dans la colonne dép
      - + Déterminer ses conditions de fin (fin)  
Inscrire sa valeur dans la colonne fin
      - + Déterminer son actionneur (act)  
Inscrire sa valeur dans la colonne act
  - + Définir la deuxième opération
    - + Déterminer sa deuxième opération
    - + Déterminer ses descripteurs
    - ...
  - ...
  - + Définir la m<sup>ième</sup> opération
    - ...
- + Définir le Cycle de la deuxième fonction
  - ...
  - ...
- + Définir le Cycle de la n<sup>ième</sup> fonction
  - ...
  - ...

Tableau 2. Représentation schématique de la description de son activité par le mécanicien

Légende: + précède un but à réaliser (les lignes qui ne sont pas précédées de + décrivent les actions réalisant le but immédiatement superposé)  
 ... remplace des buts dont la décomposition suit celle donnée plus haut dans le tableau pour un but analogue

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

### 3.2 L'activité réelle telle qu'elle a été observée

#### 3.2.1 Reprises des composants à l'intérieur de chacune des étapes

Le plan énoncé par chacun des opérateurs nous livre sa décomposition du problème qui lui est posé. Nous renverrons, dans ce qui suit, par "composants" aux différents résultats de cette décomposition.

Pour le mécanicien, un composant est donc un cycle ou un élément de cycle (fonction ou opération).

Pour le programmeur, il s'agit d'un module de programme, un sous-module, une instruction ou une branche d'instruction.

Les composants du metteur au point sont des "unités" de l'installation dans différents modes de fonctionnement (manuel, semi-automatique ou automatique). Pour cet opérateur, une "unité" peut couvrir d'une opération jusqu'à la machine dans son ensemble, en passant par les postes de la machine.

Nous constatons que chaque opérateur reprend les différents composants de son projet de conception plusieurs fois avant de les considérer achevés. Ces reprises se font après des intervalles plus ou moins importants, pendant lesquels d'autres composants sont traités.

Un composant n'est donc traité ni dans son ensemble, ni définitivement au moment où est commencé le traitement du composant suivant (quand nous renvoyons à des éléments d'ordre, ceux-ci proviennent du plan décrit par l'opérateur).

Ces deux aspects de déviation du plan hiérarchique nous font distinguer deux types de reprise: celles qui complètent petit à petit un composant en y ajoutant des éléments et celles qui modifient des (éléments de) composants.

Reprises complétant un composant. Le premier traitement d'un composant est, en général, partiel. C'est lors de reprises ultérieures que ce traitement se complète.

Ainsi, le mécanicien revient, en moyenne, cinq fois sur un cycle après qu'il ait commencé la description d'un autre cycle. Ces reprises de cycle servent surtout à compléter les informations spécifiant le démarrage du cycle (le descripteur "Ordre de départ"; cf. le Tableau 2).

Le programmeur construit certaines instructions regroupant beaucoup d'informations en une dizaine de traitements, faisant à chaque fois des ajouts et/ou des modifications il en va ainsi pour les "Auxiliaires" de l'Avance et du Retour des postes d'usinage, c'est-à-dire des instructions réunissant les conditions de l'Avance ou du Retour dans les différents modes de fonctionnement de l'installation).

Reprises modifiant un composant. Les modifications apportées lors de reprises de composants peuvent aller de la correction d'erreur d'un élément à la remise en question d'une décision prise antérieurement par l'opérateur sur un composant (les remises en question d'une décision dans une étape antérieure par un autre opérateur sont discutées ci-dessous, dans le § 3.2.2).



Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

Exemple: dans un premier temps, le programmeur définit une fonction de Bridage par poste d'usinage, comme c'est le cas sur la majorité des installations telles que celle qui est conçue (cf. § 3.3.3 "Une lecture guidée par l'utilisation d'un schéma possédant des valeurs prototypiques pour certaines variables"). Plus tard, en reprenant les spécifications, il découvre qu'il n'y a qu'un seul Bridage pour les deux postes. Il enlève alors les deux sous-modules traduisant la fonction de Bridage dans chacun des modules couvrant un poste d'usinage; à partir de l'un des deux, il crée un module de Bridage.

### 3.2.2 Reprises des composants à travers les étapes

Comme elle a été conduite sur plusieurs étapes du processus de conception, cette étude nous a permis de constater que chaque opérateur reprend des composants non seulement de son étape de la conception, mais également d'étapes antérieures qui sont censées être closes.

Nous retrouvons, à ce niveau, les deux types de reprise que nous avons distingués ci-dessus.

Reprises complétant un composant. Un exemple de ce type de reprise est fourni par le mécanicien qui introduit dans les spécifications qu'il produit des opérations qui n'avaient pas été spécifiées jusque là. Dans tous les cas d'opérations ajoutées par le mécanicien (11 opérations sur un total de 26), il s'agit d'opérations qui constituent des prérequis d'autres opérations.

Leur introduction dans un cycle peut d'ailleurs, à l'intérieur du projet de conception du mécanicien, également faire l'objet de reprises. Le mécanicien souvent ne repère une opération prérequis que lors du traitement, soit de l'opération dont elle constitue le prérequis, soit de son opération contraire, soit d'une opération analogue (cf. ci-dessous § 3.3.2). C'est-à-dire, il ne la traite pas au moment où, selon son plan, il aurait dû le faire.

Reprises modifiant un composant. Un exemple intéressant de remise en cause du projet de conception développé dans les étapes antérieures est le suivant: le programmeur, jugeant que la façon dont le mécanicien a spécifié une certaine fonction n'est pas réalisable avec les moyens élect(ron)iques prévus sur l'installation, modifie les spécifications sur ce point. Il ne les modifie pas seulement en programmant sa solution; il traduit celle-ci également sur la représentation fonctionnelle construite par le mécanicien, en modifiant celle-ci.

### 3.3 *Processus psychologiques conduisant aux abandons du plan hiérarchique*

Nous avons présenté, à l'aide d'exemples, quatre types de déviation du plan hiérarchique que les opérateurs prévoient pour leur activité. Si ces déviations rendent opportuniste le plan de l'activité réelle, dans le sens que l'activité est guidée par les circonstances, ceci ne signifie pas que l'activité est orientée par le hasard.

Nous avons analysé les facteurs d'abandon et de reprise de composants; parmi ceux-ci, nous présentons ci-dessous les principaux processus psychologiques. Tous ne sont pas utilisés par chacun des opérateurs ou dans chacune des activités.

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

### 3.3.1 Processus conduisant à compléter un composant

L'utilisation économique des moyens disponibles. Parmi les sources d'information externes et internes et ses ressources cognitives dont l'opérateur peut disposer en principe, toutes ne sont pas également accessibles. L'utilisation économique des moyens disponibles conduit alors à:

- l'ajournement du traitement (d'éléments) d'un composant

\* quand l'opérateur n'a pas sous la main la source d'information nécessaire pour son traitement (ou quand il a sous la main une source d'information utile pour le traitement d'un autre composant (cf. l'exemple donné ci-dessous à propos du "Traitement de composants guidé par la structure de la source d'information disponible"). Ainsi le programmeur "saute" certains sous-modules pour le traitement desquels il a besoin d'informations du mécanicien et il établit une liste de questions pour celui-ci.

Il lui arrive aussi de laisser un "trou" dans une instruction, estimant qu'il y aura un moment plus propice pour créer l'information qui doit y figurer. Il sait donc qu'il lui faudra encore y remplir une variable. En général, il connaît dans ce cas, certaines des caractéristiques de la variable et parfois il les mentionne au-dessus du trou sous la forme d'un commentaire verbal.

\* quand l'opérateur estime que ce traitement sera plus facile après le traitement d'autres composants (car il disposera alors, par exemple, de plus de connaissances au sujet d'interactions avec le composant en question). Ainsi, le mécanicien saute dans un premier temps le cycle du Chargement, qui lui paraît trop difficile au moment où, selon son plan, il devrait le décrire.

- le traitement (d'éléments) de composants guidé par (la structure de) la source d'information disponible. Exemple: le mécanicien en est au traitement d'une certaine opération. Il cherche un document présentant les éléments permettant de déterminer, pour cette opération, ses deux premiers descripteurs ("identificateur" et "durée"). Ce document présente également les éléments permettant de déterminer, pour deux autres opérations du cycle, ces deux descripteurs "identificateur" et "durée". Il "en profite" alors pour déterminer en effet, pour ces trois opérations, ces deux descripteurs. S'il avait suivi son plan, il ne les aurait déterminés que pour la première des trois opérations, après quoi il aurait déterminé, par d'autres moyens, les descripteurs restants de cette opération, avant de passer à l'opération suivante.

Dans cette catégorie, nous plaçons également les "dérives" du metteur au point qui se font selon les problèmes rencontrés. C'est la structure de ceux-ci qui guide ses vérifications.

Il est impossible que le metteur au point vérifie le comportement de l'installation dans toutes les configurations d'états que ses différentes parties peuvent prendre. Certaines sont examinées systématiquement. Beaucoup cependant ne constituent l'objet d'une vérification qu'à la suite d'un problème qui surgit lors de la vérification d'une autre configuration. Ainsi, l'ordre des vérifications effectuées par l'opérateur est plutôt fonction des problèmes rencontrés que de son plan qui prévoit de vérifier les composants dans un certain ordre.

Exemple: le plan du metteur au point prévoit de vérifier l'Avance bien avant les mouvements (manuels) des outils. A un certain moment, le metteur au point vérifie l'Avance du poste d'usage

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

Ebauche. Elle ne se fait pas. Son hypothèse (formulée après plusieurs examens de différentes instructions) étant qu'un outil est responsable de ce fonctionnement inattendu, il vérifie d'abord cet outil.

Cet exemple est simplifié car la dérive peut conduire à la vérification de quatre ou cinq composants non prévus, chaque vérification étant dictée par les problèmes rencontrés lors de la précédente.

L'anticipation du traitement (d'éléments) d'un composant (nous aurions pu classer ce processus également dans la catégorie précédente; cf. exemple présenté) Exemple: le programmeur "pense à" un composant analogue (qui, selon le plan, serait à traiter ultérieurement) ou un composant dont il estime qu'il risque de l'oublier s'il ne le traite pas immédiatement et il interrompt le traitement en cours pour traiter le composant auquel il a pensé.

Un autre type d'anticipation (mais non pas sur le plan) est l'utilisation par le programmeur de variables regroupant un certain nombre d'informations (des "bits intermédiaires") avant leur définition.

### 3.3.2 Processus conduisant à compléter et/ou à modifier un composant

L'identification d'un certain type de relation entre (un élément du/)le composant sous traitement et (un élément de) celui qui est repris. Il peut s'agir de:

- une relation d'analogie ou de correspondance (certaines modalités de ce processus peuvent être classées dans la catégorie "Utilisation économique des moyens disponibles"). L'installation comportant deux postes d'usinage, les opérateurs, lors du traitement d'un composant d'un poste, reprennent ou anticipent souvent le traitement du composant analogue sur l'autre poste.

Le mécanicien traite le cycle du poste d'usinage ébauche avant celui du poste d'usinage finition et il revient souvent, pour des raisons d'analogie, de la finition vers l'ébauche. Le programmeur procède à l'envers (entre autres, parce qu'il s'inspire d'un programme-"exemple" présentant les modules correspondants dans cet ordre). Ainsi, le programmeur écrit le sous-module de Bridage sur le poste d'usinage ébauche en prenant exemple sur celui du poste d'usinage finition, qu'il a déjà écrit.

- une relation d'interaction. Cette relation d'interaction peut être basée, comme dans l'exemple donné ci-dessus, sur une relation entre deux unités physiques de l'installation (le plateau tournant est en interaction avec chacun des trois autres unités: celles-ci doivent être reculées pour que lui puisse tourner). Le mécanicien décrit, partiellement, le cycle du Plateau avant celui des autres unités. Lors du traitement de certaines de ces unités, il revient en arrière pour introduire des informations sur le Cycle du Plateau.

- une relation de contraires. Le mécanicien repère l'oubli d'une opération lors du traitement de son opération inverse. Le metteur au point vérifie si une opération, dont il a établi qu'elle peut se faire (en mode manuel) dans un sens, peut se faire également dans le sens inverse.

- une relation de prérequis. Le mécanicien repère l'oubli d'une opération (par lui et par ses collègues) lors du traitement de celle dont elle est le prérequis. Nous avons constaté le statut spécifique des prérequis dans le traitement d'information déjà dans une autre étude, faite sur des élèves automaticiens (Morais & Visser, 1985).

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

La vérification de composants déjà traités fait revenir, de temps en temps, en arrière aussi bien le mécanicien que le programmeur (il ne s'agit pas ici de la "vérification du programme" par le metteur au point, qui sera présenté, schématiquement, ci-dessous). Le mécanicien interrompt, par exemple, le traitement d'un cycle d'usinage pour vérifier (à partir de vitesses et de courses) les calculs des durées d'opérations dont il a trouvé les valeurs sur des documents de spécification qu'il a reçus.

Le programmeur ayant introduit, dans des instructions, des bits intermédiaires pas encore définis (cf., ci-dessus, "L'anticipation du traitement (d'éléments) d'un composant"), il lui arrive de vérifier, après la construction des bits en question, leur utilisation dans ces instructions.

Le metteur au point, procède, pour vérifier des fonctions, à des vérifications mentales (à partir de la lecture d'instructions) et à des vérifications par l'observation du fonctionnement (à partir d'essais).

L'essai d'une fonction est matérialisé, en général, par l'appui sur un bouton d'un pupitre de commande et l'observation du comportement de l'installation qui s'ensuit.

La simulation est une stratégie de vérification utilisée fréquemment par le metteur au point, mais le programmeur y fait appel également quand, de temps en temps, il interrompt l'écriture des composants pour vérifier ce qu'il a écrit.

Le metteur au point simule mentalement le fonctionnement de l'installation, par exemple, lors de la première lecture d'un (sous)-module de programme. Cette simulation peut l'aider à repérer des erreurs ou des oublis (cf. Détienne, 1984).

Un exemple intéressant dans le contexte de cet article est celui d'une modification par le metteur au point des spécifications du mode manuel, telles qu'elles ressortent du programme écrit par le programmeur. A partir d'une simulation de l'utilisation future de l'installation, aussi bien mentale (en lisant le programme) que par l'action (en faisant fonctionner la machine), il modifie le programme. La contrainte qu'il introduit, par ce biais, dans les spécifications du fonctionnement (manuel) de l'installation, traduit le principe selon lequel le fonctionnement doit permettre une utilisation manuelle "le plus souple possible" (cf., ci-dessus, "Le changement de critères utilisés").

La génération et la mise à l'épreuve d'hypothèse est une stratégie fondamentale dans la vérification des fonctions par le metteur au point. Nous concevons chaque essai (cf. ci-dessus) comme une mise à l'épreuve d'une hypothèse. Même s'il ne l'explique pas (devant nous ou devant lui-même), le metteur au point a une hypothèse au sujet du comportement de l'installation auquel doit conduire son essai. Si l'essai conduit au comportement de l'installation attendu par le metteur au point, celui-ci passe, en général, au point suivant, c'est-à-dire à la vérification de la fonction ou de l'opération suivante. Sinon, il procède à un diagnostic des problèmes rencontrés.

### 3.3.3 Processus conduisant à modifier un composant

Le changement de critères utilisés conduit les opérateurs à remettre en cause des décisions prises antérieurement par eux-mêmes ou par les collègues.

La modification de la durée d'une opération par le mécanicien en constitue souvent un exemple. Elle peut concerner la conception telle qu'elle a été définie par le mécanicien lui-même ou par un autre opérateur (cf. l'exemple donné ci-dessus dans "La vérification de composants déjà traités"). Dans les

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

deux cas, le mécanicien peut préférer, par exemple, attribuer à l'opération reprise une durée plus "prudente", donc plus longue. Une valeur plus longue est plus "prudente" pour la raison suivante.

Une des principales contraintes imposées par le demandeur du projet de conception est une durée maximale pour les cycles. Un critère important de la longueur des durées attribuées aux opérations est alors, pour le mécanicien, la durée totale du cycle en question. Régulièrement, il vérifie, en faisant l'addition des durées des opérations du cycle sous traitement, qu'elles ne transgressent pas cette limite.

Ainsi des durées plus longues sont plus "prudentes", car si, avec des durées plus longues, les opérations ne transgressent pas la durée limite du cycle, l'opérateur est quasiment sûr qu'elles tiendront également sur la machine quand elle sera réalisée.

L'homogénéisation de la structure (d'éléments) de plusieurs composants est appliquée beaucoup par le programmeur. C'est sur plusieurs plans du programme que celui-ci cherche à créer une structure uniforme:

- ordre des instructions dans les modules;
- ordre des branches dans les instructions;
- ordre des bits dans les branches des instructions;
- numéros de certains bits couvrant des variables que l'on retrouve sur plusieurs postes (s'il accorde B613, B614, B615 à trois variables sur un poste, par exemple, il attribuera B713, B714, B715 aux variables analogues sur un autre poste et B813, B814, B815 sur un troisième).

Dans ce but, il revient régulièrement sur des instructions déjà écrites:

- pour les rendre homogènes avec l'instruction qu'il est en train d'écrire;
- pour rendre homogène avec elles l'instruction qu'il est en train d'écrire.

D'autre part, le programmeur essaie de donner au programme qu'il écrit une structure la plus homogène possible avec celle du programme qu'il prend en exemple. Ceci le conduit parfois à des réorganisations assez importantes de son programme.

Une lecture guidée par l'utilisation d'un schéma possédant des valeurs prototypiques pour certaines variables peut expliquer que le programmeur apporte, par le biais de son programme, certains changements dans les spécifications. Ainsi, sur des machines d'usinage, les opérations d'usinage se font, en général, pendant l'"Avance". Sur un des postes de l'installation étudiée, la dernière opération d'usinage se fait pendant le "Retour". Le programmeur donne à cette opération cependant les mêmes conditions qu'à celles qui se font pendant l'"Avance".

#### **4. Conclusion**

Un premier résultat concerne l'abandon par les opérateurs du plan hiérarchique dont ils estiment qu'il traduit (ou guide) leur activité. Ce résultat est intéressant aussi parce qu'il vient confirmer, dans

Ce texte est un pre-print de  
Visser, W. (1987). *Abandon d'un plan hiérarchique dans une activité de conception*. Actes du Colloque scientifique MARI 87 Machines et Réseaux Intelligents - COGNITIVA 87 (Tome 1), Paris, La Villette.  
Version anglaise : Visser, W. (1988). *Giving up a hierarchical plan in a design activity* (INRIA Research report n° 814). Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique. Also accessible at <http://hal.inria.fr/inria-00075737/en/>

une étude faite sur des tâches complexes, les résultats d'autres études conduites en laboratoire sur des tâches de conception simplifiées. Il renforce alors la conclusion selon laquelle la conception est une activité qui, même si elle est souvent guidée par un plan hiérarchique conduisant à une décomposition du problème ou de sa solution en composants hiérarchisés, comporte de nombreux aspects non hiérarchiques, mais plutôt opportunistes (cf. Hayes-Roth & Hayes-Roth, 1979).

Si ces résultats confirment ceux d'études antérieures, ils en apportent cependant aussi de nouveaux: nous montrons que ce caractère opportuniste ne caractérise pas seulement l'activité de chaque opérateur dans son étape de conception, mais, également le processus de conception dans son ensemble.

Ce qui nous paraît cependant l'apport le plus important de notre étude, c'est l'analyse des facteurs qui conduisent les opérateurs à l'abandon de leur plan hiérarchique. Les processus psychologiques que nous avons présentés sont tous bien connus dans d'autres contextes de résolution de problème ou de diagnostic.

D'une part, ces résultats permettent alors de préciser dans quelle mesure l'activité de conception se rapproche et se différencie de ces autres activités.

D'autre part, la connaissance de ces processus et de leurs conditions spécifiques de déclenchement nous fournit des renseignements précieux pour la modélisation de l'activité de conception.

## **Bibliographie**

- Carroll, J. M. & Rosson, M. B. Usability specifications as a tool in iterative development. In H. Rex Hartson (Ed.), *Advances in human-computer interaction* (Vol. 1). Norwood, N.J.: Ablex, 1985.
- Détienne, F. Analyse exploratoire de l'activité de compréhension de programmes informatiques. *Proceedings AFCET du Séminaire "Approches quantitatives en Génie Logiciel"*, Sophia Antipolis, France, 7-8 juin 1984.
- Hayes-Roth, B. & Hayes-Roth, F. A cognitive model of planning. *Cognitive Science*, 1979, 3, 275-310.
- Morais, A. & Visser, W. Etude exploratoire de la programmation d'automates programmables chez les élèves de l'enseignement technique (Rapport de Recherche INRIA N° 404). Rocquencourt: INRIA, 1985.
- Robert, J.-M. Les résultats des recherches sur le processus de conception, les comportements et les processus cognitifs mis en jeu (Rapport INRIA EC 7912 R01). Rocquencourt: INRIA, 1979.
- Tardieu, H., Rochfeld, A. & Colletti, R. La méthode MERISE. Tome 1: principes et outils (éd. revue et corrigée). Paris: Les Editions d'Organisation, 1986.
- Visser, W. Modélisation de l'activité de programmation de systèmes de commande. Actes du colloque COGNITIVA 85 (tome 2). Paris: Cesta, 1985.