THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR ES SCIENCES
Informatique

par

Michel CAPLAIN

LANGAGE de SPECIFICATIONS

Thèse soutenue le 20 décembre 1978 devant la Commission d'Examen :

Président : B. VAUQUOIS

Examinateurs : J.R. ABRIAL
L. BOLLIET
A. COLMERAUER
P. DESVERGNES
A. GUILLON
P. JORRAND

BIBLIOTHEQUE DU CERIST

RESUME

- *Le problème qui est à l'origine de cette thèse est de fournir aux industriels les moyens d'obtenir des programmes certifiés conformes à des spécifications.*

L'insuffisance des techniques classiques de vérification a posteriori (chapitre 0) me conduit à proposer un système de Programmation Assistée par Ordinateur garantissant la production de programmes corrects par une méthode de transformations de spécifications: il s'agit de passer de la spécification au programme par une suite de transformations dont chacune est prouvée valide. Donc le système oblige et aide le concepteur à documenter et justifier chacun de ses choix d'implémentation.

Le principal ingrédient de ce système est un langage permettant d'exprimer à la fois: la spécification initiale (Intention), les spécifications intermédiaires (Documentation), le programme final et les raisonnements justifiant chaque transformation.

Ce langage de spécifications doit satisfaire à quatre exigences apparemment contradictoires:

- *être clair, pour que le texte de la garantie soit lisible, crédible et convaincant ;*
- *tolérer les ambiguités, puisqu'on ne peut jamais les exclure complètement (chapitre I) ;*
- *être défini de façon suffisamment simple et rigoureuse pour se prêter à des preuves formelles ;*
- *être d'un emploi facile pour l'utilisateur.*

Un tel langage peut évidemment aussi servir à la Conception Assistée de produits autres que des programmes, et à un vaste sous-ensemble de l' "Intelligence Artificielle".

On trouvera ici une présentation aussi intuitive que possible du langage de spécifications que je propose.

BIBLIOTHEQUE DU CERIST

- *Syntaxe du langage:* *Pour accommoder l'infinie diversité des applications et la simplicité d'utilisation, le langage doit être "dynamiquement extensible" afin d'épouser progressivement la forme d'expression choisie par l'utilisateur.*

La grammaire est donc évolutive ; elle comporte à tout instant un ensemble d'opérateurs dits "généralisés" parce que leurs arguments et résultat sont délimités par des "types" qui - au lieu d'être déterminés - peuvent être seulement liés par une relation (et chaque opérateur a une syntaxe propre

L'ensemble de ces types est lui-même évolutif et défini à tout instant par des types de base et des constructeurs de types. Cet ensemble est de plus, muni d'un ordre partiel correspondant à l'inclusion (chapitre I).

Chaque identifieur ou expression apparaissant dans le langage a un type qui, s'il n'est pas exactement connu, peut être approché par des bornes inférieure et supérieure (ambiguité de types).

L'analyse syntaxique (chapitre II) a pour but d'établir pour chaque phrase de l'utilisateur un arbre syntaxique conforme à la syntaxe propre de chaque opérateur. S'il y a zéro ou plusieurs arbres, le système pose des questions dont les réponses permettront de déduire les extensions nécessaires (apprentissage, par le système, du langage de l'utilisateur).

L'utilisateur est sensé ne rien connaître de la grammaire et des types ; son seul rôle est de s'exprimer en respectant quelques conventions, de répondre aux questions, et de changer certains symboles si le système le lui demande.

- *Signification du langage*

L'utilisation du langage suppose qu'on puisse prouver qu'une spécification est conforme à (c'est-à-dire implique) une autre. Pour ce faire, on définit par des "règles d'inférence" une sémantique déductive du langage (chapitre III) qui, à tout ensemble de phrases (même ambigües) associe les phrases qui en sont des conséquences certaines.

BIBLIOTHEQUE DU CERIST

Un démonstrateur de théorèmes est donc nécessaire pour au moins approximer cette sémantique.

Parmi les divers critères de qualité applicables à tout démonstrateur, on montre que le plus important est l' "autonomie" (difficulté des théorèmes qu'il peut prouver en un temps donné) ; en effet, en améliorant l'autonomie, on diminue d'autant l'intervention requise de l'utilisateur. Malheureusement, l'autonomie ne sera jamais suffisante pour que le démonstrateur puisse être entièrement automatique: il sera nécessairement interactif.

Le démonstrateur proposé est fondé pour l'essentiel sur une technique de "pattern matching" très efficace qui pourrait être appliquée aussi à d'autres domaines: l'information est codée sous forme de "spectres", et la déduction est dirigée par la ressemblance (résonance) des spectres des diverses hypothèses.

Avenir du langage

Seule une expérimentation en vraie grandeur permettra de décider du bien-fondé de cette approche, même s'il existe déjà des indices favorables, notamment:

- *simulation à la main ;*
- *fondements semblables à ceux du langage LCF de Milner, dont le succès est indiscutable ;*
- *contournement de difficultés classiques de l'Intelligence Artificielle.*

La principale difficulté d'implémentation sera de déterminer l'état initial du langage (types, opérateurs, axiomes et théorèmes initiaux) permettant aux utilisateurs de s'en servir facilement.

Enfin, l'intérêt et l'étendue des applications du langage justifient largement le coût estimé (relativement élevé) de son implémentation.

BIBLIOTHEQUE DU CERIST

SOMMAIRE

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST