

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

***Une bibliothèque pour des moteurs de pilotage
de programmes***

Régis Vincent, Sabine Moisan, Monique Thonnat

N° 3011

Octobre 1996

———— THÈME 3 ————



***Rapport
de recherche***



Une bibliothèque pour des moteurs de pilotage de programmes

Régis Vincent, Sabine Moisan, Monique Thonnat

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Orion

Rapport de recherche n° 3011 — Octobre 1996 — 27 pages

Résumé : Notre objectif est de faciliter la construction de systèmes à base de connaissances (SBC). Pour cela, nous construisons des moteurs *dédiés* à une classe de problèmes. Les travaux exposés dans ce rapport proposent de baser la construction de ces moteurs sur des composants *réutilisables*. Le but est d'améliorer la rapidité de construction et de modification des moteurs ainsi que la sécurité de programmation. Dans un premier temps, nous avons décidé de nous restreindre à une classe de problèmes particulière : les problèmes de *pilotage de programmes*, dont une sous-partie importante est la tâche de planification. À partir de différents modèles de moteurs de pilotage de programmes, nous avons défini une première *bibliothèque de structures de données et d'instructions de base* pour le pilotage de programmes et nous l'avons utilisée pour l'écriture de nouveaux moteurs.

Mots-clé : bibliothèque de composants, pilotage de programmes, planification, système à base de connaissances.

(Abstract: *pto*)

A library for program supervision engines

Abstract: Our aim is to provide easy means to construct knowledge-based systems (*KBS*). For this purpose, we design *engines* dedicated to a class of problems. The work presented in this report proposes to base the construction of such engines on *reusable components*. Thereby we aim at speeding both the engine construction and modification as well as securing the coding. To begin with, we restricted ourselves to a particular class of problems: those of *program supervision*, an important sub-part of which is planning. Based on different program supervision models, we have defined a first *library of components and elementary instructions* for program supervision and we have used it to code new engines.

Key-words: library of components, program supervision, planning, knowledge-based systems

1 Introduction

Un système à base de connaissances (SBC) est composé d'un *moteur* générique de raisonnement et d'une base de connaissances. Dans le but de faciliter la construction de SBC, nous construisons des moteurs qui ne sont pas liés à une expertise particulière, mais qui sont cependant *dédiés* à une classe de problèmes. Les classes de problèmes envisagées sont le pilotage automatique de programmes et l'interprétation d'images. L'approche que nous proposons dans ce rapport est motivée par nos expériences sur les différents moteurs que nous avons développés, dans le cadre de projets de recherche ou industriels. Nous avons déjà construit plusieurs outils dans des domaines différents, par exemple OCAPI [22, 16, 15] dédié au pilotage de programmes en traitement d'images, PLANETE [19] dédié au pilotage de programmes temps-réel et CLASSIC [23, 4] dédié à la classification.

Malgré leurs différences, les moteurs de ces outils reposent sur les mêmes principes :

- ils encapsulent un mécanisme lié à une stratégie de raisonnement particulière. Par exemple pour OCAPI et PLANETE, la stratégie consiste principalement à imbriquer une phase de planification et une phase d'exécution;
- ils s'appuient sur un certain nombre de structures de données permettant de représenter la connaissance spécifique à la classe de problèmes visée.

Pour construire de tels moteurs il est nécessaire de se baser sur une modélisation des connaissances et des méthodes de résolution de problème qui interviennent dans les classes de problèmes étudiées. Le développement d'outils dédiés fournit des systèmes bien adaptés à l'expression des connaissances et largement utilisables dans différents domaines d'applications ([2], [20]). Cependant, à chaque moteur construit correspond une stratégie souvent figée et difficile à modifier si on souhaite élargir la gamme d'applications visées. Toute évolution des spécifications, des besoins ou des contraintes nécessite de re-configurer, voire de re-écrire, les outils construits.

Nous proposons de pallier cet inconvénient en construisant ces moteurs eux-mêmes à partir de composants de base, regroupés au sein d'une bibliothèque. L'utilisation d'une telle approche permet de décrire une stratégie de raisonnement à un niveau abstrait, plus simplement et plus rapidement qu'avec un langage de programmation, et de la faire évoluer plus facilement. En effet, le travail de réécriture de moteurs en différents langages ou l'écriture de variantes d'un même moteur, nécessaires pour des tests ou des domaines d'applications différents, sont très fastidieux et coûteux en temps.