

# Étude de l'interopérabilité de deux langages de programmation basée sur la machine virtuelle de Java

Mémoire présenté par Frédéric Minne - Promoteur : Baudouin Le Charlier  
UCL - Année académique 2002-2003

## Résumé

### Motivations

L'interopérabilité entre deux langages de programmation est la capacité d'un programme (ou d'un fragment de programme) écrit dans un langage de programmation  $X$  d'utiliser un programme (ou un fragment de programme) écrit dans un langage de programmation  $Y$  différent de  $X$ .

Le problème de l'interopérabilité entre langages de programmation est un problème majeur en informatique. On s'y trouve confronté dans de nombreux cas, entre autres lors de :

- l'interconnexion de systèmes d'information hétérogènes lors de la fusion de deux entreprises;
- la réalisation de services web à partir d'applications déjà existantes, réalisées dans des langages de programmation ne permettant pas une intégration aisée avec les technologies de l'internet;
- l'utilisation de plusieurs langages de programmation pour la résolution des différents aspects d'un problème, ce qui permet de tirer parti des spécificités de chaque langage.

### Objectifs

L'objectif de ce mémoire est d'étudier l'interopérabilité entre deux langages aux concepts très différents, Java et Prolog, et de voir comment la réaliser concrètement en se basant sur des mécanismes de la machine virtuelle de Java.

Cette interopérabilité permettra d'intégrer des fonctionnalités de programmation logique et d'intelligence artificielle dans un programme Java (systèmes experts, jeux, smartbots...), de créer des interfaces graphiques pour des programmes Prolog existants, ou, encore, d'ajouter des fonctionnalités au langage Prolog sous forme de prédicats prédéfinis (accès à des bases de données, networking...). De plus, de nombreux logiciels pourraient être simplifiés grâce à l'adjonction d'un programme Prolog. C'est le cas, par exemple, du logiciel de configuration des interfaces réseau du système d'exploitation Windows NT, qui intègre un interpréteur Prolog pour lier les interfaces réseau aux protocoles de transport et aux pilotes correspondants.

### Démarche

Pour étudier cette interopérabilité, ce mémoire développe un mécanisme d'interopérabilité basé sur la machine virtuelle de Java (JVM), et utilisant comme interpréteur Prolog un compilateur transformant les programmes Prolog en bytecode exécutable par la JVM. Ce mécanisme est toutefois réalisé de manière à être implémenté avec n'importe quel interpréteur Prolog.

Ce mécanisme d'interopérabilité est composé de trois parties : un mécanisme de conversion des données entre Java et Prolog, un mécanisme permettant d'invoquer un programme Prolog depuis un programme Java, et un mécanisme permettant d'invoquer, depuis un programme Prolog, des méthodes écrites en Java.

Le mécanisme de conversion consiste à fournir des classes Java représentant les termes du langage Prolog, des méthodes permettant la conversion entre ces deux types d'objets, et des méthodes permettant de récupérer le contenu de ces classes pour les manipuler dans un programme Java.

Le mécanisme d'interopérabilité de Java vers Prolog est basé sur l'idée de fournir un interpréteur Prolog abstrait qui pourra être instancié et invoqué depuis un programme Java, et cachera les détails d'implémentation de l'interface.

Enfin, le mécanisme d'interopérabilité de Prolog vers Java sera basé sur un ensemble de règles de conception pour les méthodes Java invocables, sur l'extension d'une classe bibliothèque de fonctions, sur l'emploi d'une directive d'importation et sur l'utilisation d'un précompilateur.

Outre la description et l'implémentation du mécanisme d'interopérabilité proprement dit, ce mémoire se penche également sur les différences qu'il existe entre la machine virtuelle de Java et un interpréteur Prolog, du point de vue de la gestion de la mémoire, et sur la manière dont ces deux machines pourront cohabiter.

Il présente, de plus, plusieurs exemples et scénarios d'application du mécanisme d'interopérabilité entre Java et Prolog et s'intéresse aux conditions dans lesquelles l'un des sens du mécanisme (de Java vers Prolog ou de Prolog vers Java) doit être préféré.

## **Perspectives**

Le mécanisme décrit dans ce mémoire devrait assez facilement être adapté à l'interopérabilité entre Java et d'autres langages interprétés, tels que Lisp, Scheme... Il pourrait même être utilisé pour créer un langage de programmation hybride entre programmation logique et programmation orientée-objets.

## **Pour en savoir plus**

Une description plus précise de ce travail, ainsi que le code source du mécanisme d'interopérabilité et des articles complémentaires, peuvent être trouvés sur le site internet consacré à ce projet : <http://blc-interop.be.tf>.