

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna
Faculté des Sciences de l'Ingénieur
Département d'Informatique



THÈSE DE MAGISTÈR

Présenté par : *Salim KALLA*

Spécialité : *Informatique*
Option : *Informatique Industrielle*

THÈME

TRACE ET ANALYSE DE TRACE EN CHR
(Constraint Handling Rules)

Composition du jury:

Dr. Abdelmadjid Zidani	U.Batna	Président
Dr. Mohammed Benmohammed	U.Constantine	Rapporteur
Dr. Mohamed Chawki Baatouche	U.Constantine	Examineur
Dr. D. Eddine Saidouni	U.Constantine	Examineur
Dr. Brahim Belattar	U.Batna	Examineur

Table des matières

Introduction	6
1 Dimensions du problème	8
1.1 Introduction	8
1.1.1 Le projet DiSCiPl	8
1.1.2 Le projet OADymPPaC	10
1.2 Le débogage dynamique	10
1.3 Outils pour débogage dynamique	11
1.4 Etat de l'art	12
1.4.1 Cosytec	13
1.4.2 EMN(<i>Ecole des Mines de Nantes</i>)	13
1.4.3 ILOG	13
1.4.4 LANDE	14
1.4.5 PLC	14
1.4.6 Contraintes	14
1.5 Exemple d'outil (Outil d'exploration 3D avec VRML)	14
1.5.1 Expérimentation(le problème des reines)	15
2 Programmation avec contraintes (PaC)	19
2.1 Introduction	19
2.2 La PaC : un nouveau paradigme	20
2.3 Qu'est-ce qu'une contrainte ?	20
2.3.1 Quelques caractéristiques des contraintes	21
2.3.2 Définition d'une contrainte	21
2.3.3 Arité d'une contrainte	21
2.3.4 Différents types de contraintes	22
2.4 Un paradigme déclaratif puissant	24
2.4.1 Déclaratif vs Opérationnel	24
2.4.2 Un exemple "électrique"	25

TABLE DES MATIÈRES	4
2.5 Les solveurs	26
2.6 Rôle du programmeur	27
2.7 Contraintes domaines finis (fd)	27
2.7.1 CLP(FD)	27
2.8 Trace d'exécution	28
2.9 Modèle de trace pour CLP(fd)	29
2.10 Sémantiques formelles de langages de programmation	29
3 Trace générique	31
3.1 Solveur sur domaines finis	31
3.2 Sémantique observationnelle pour les contraintes avec domaines finis	32
3.2.1 Contrôle	34
3.2.2 Propagation	34
3.2.3 Opérateur de réduction	35
3.3 Trace générique	36
3.4 État de l'art	37
3.4.1 VisualCHR : un outil de visualisation CHR	37
3.4.2 SICStus Prolog : le débogueur CHR	37
3.4.3 Opium : un analyseur extensible de trace pour Prolog	38
4 CHR sur domaine fini(fd)	40
4.1 Syntaxe	40
4.2 Sémantique déclarative	41
4.2.1 Solveur CHR(fd)	42
4.2.2 Opérateur de réduction	43
4.2.3 Partie du solveur CHR(fd)	45
4.2.4 Exemple	46
4.3 CHR dans ECLIPSE	47
5 Sémantique opérationnelle de CHR	48
5.1 La forme normale	49
5.2 Sémantique opérationnelle de CHR	49
5.3 Exécution d'un programme CHR	51
5.4 Sémantique observationnelle de CHR(fd)	52
5.4.1 Sémantique opérationnelle raffinée	52
5.4.2 Sémantique observationnelle	55

TABLE DES MATIÈRES	5
6 Implémentation d'un traceur	60
6.1 Environnement de programmation	60
6.1.1 Initiation à la programmation en logique avec contrainte(Eclipse)	60
6.1.2 Utilité partique de Prolog	61
6.2 Traduction des règles CHR(fd)	61
6.3 Traduction des règles sémantiques	63
6.4 Méta interprète	65
6.5 Conclusion	65
Discusion et Conclusion	66
Bibliographie	67
Liste des Tableaux	70
Table des figures	71
Annexe	72

Résumé : Les outils d'observation des exécutions sont utiles pour développer et maintenir des programmes logiques avec contraintes (PLC). Cependant, les outils actuels sont difficilement portables d'une plate forme à l'autre. L'analyse d'une trace d'exécution peut donner des vues intéressantes de l'exécution. Un modèle de trace générique a été mis au point de manière à être valable pour les solveurs sur les domaines finis. Ce modèle est formalisé par une sémantique observationnelle. Le travail développé dans ce rapport montre comment nous appliquons ce modèle aux solveurs CHR (*Constraint Handling Rules*) sur domaines finis, afin d'englober différents types de solveurs. Nous étudions dans cette approche la validité de ce type de modèle.

Mots-clés : programmation avec contrainte, programmation logique, débogage, trace générique, CIIR(Constraint Handling Rules), visualisation, sémantique observationnelle, solveur, domaines finis.

Abstract : Developing and maintaining Constraint Logic Programs (CLP) requires performance debugging tools based on visualization and explanation. However, existing tools are built in an ad hoc way and porting them from one platform to another is very difficult and experimentation of new tools remains limited. A generic trace model for constraint resolution was proposed. In this report, we propose a generic trace model for CHR (constraint Handling Rules) to study and improve it.

Key-words : constraint programming, logic programming, debugging, generic trace, CIIR(Constraint Handling Rules), visualization, observational semantics, solver, Finite Domain.

