

REPUBLICUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

SCIENTIFIQUE

UNIVERSITÉ EL-HADJ LAKHDAR – BATNA
FACULTÉ DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

THESE

Présentée par

REBAIAIA Mohamed-Larbi

pour obtenir le titre de

DOCTEUR en Sciences

Spécialité

INFORMATIQUE

Sujet de la thèse :

Spécification et Vérification des Systèmes Critiques :

Extension de l'Environnement VALID pour la Prise en Charge du Temps Réel

Soutenue publiquement le 16 février 2011 devant le jury composé de :

M. Noureddine BOUGHECHEL	Professeur	Président
M. Mohamed BENMOHAMED	Professeur	Directeur de Thèse
M. Djamel-Eddine SAIDOUNI	Professeur	Examineur
M. Allaoua CHAOUI	Maître de conférences	Examineur
M. Azzedine BILAMI	Maître de conférences	Examineur
M. Brahim BELATAR	Maître de conférences	Examineur

SOMMAIRE

CHAPITRE 1 : INTRODUCTION ET MOTIVATION

1.1	Problème étudié dans la thèse	2
1.2	Contexte de Nos Travaux	3
1.3	Plan de la Thèse	5

CHAPITRE 2 : ETAT DE L'ART

2.1	Introduction	8
2.2	Méthodes Formelles	8
2.3	Processus de conception	9
2.4	Spécification Formelle	10
2.5	Vérification Formelle	11
2.6	Vérification du Matériel (Hardware)	12
2.7	Techniques de Vérification du matériel	12
2.8	Les spécifications comportementales	13
2.8.1	Les spécifications Axiomatiques	14
2.8.2	Les spécifications Logiques	14
2.8.3	Les spécifications Algébriques	14
2.9	Explosion des états et Génération de Modèles	15
2.10	Quelques Exemples Notables de cas vérifiés	18
2.11	Conclusion	18

CHAPITRE 3 : DEFINITIONS DE BASE

3.1	Introduction	20
3.2	Concepts fondamentaux	20
3.3	Systèmes temps réel	21
3.3.1	Définition d'un système temps réel	21
3.3.2	Temps et contraintes temporelles	22
3.4	Les classes de systèmes temps réel	23
3.5	L'Approche Synchrone	23
3.6	L'Approche Flot de Données	24
3.7	L'Approche Asynchrone	25
3.7.1	Synchrone / Asynchrone	25
3.8	Langages et automates	25
3.9	Les multi-ensembles	27
3.10	Équivalences de Bisimulation	28
3.11	Conclusion	30

CHAPITRE 4 : LES DIAGRAMMES BINAIRES DE DECISION

4.1	Introduction	32
4.2	Diagrammes De Décision Binaire	33
4.2.1	Diagrammes De Décision Binaires Ordonnés (OBDD)	34
4.2.3	Comment Construire les BDDs	35
4.3	Implantation Informatique Des BDDs	36
4.3.1	Algorithme De Réduction Des BDDs	37
4.4	Manipulation Des ROBDDs	37
4.4.1	L'approche Depth-First search (profondeur d'abord)	37
4.4.2	L'approche Breadth-First Search	39

4.5	Algorithme ITE	40
4.6	Ordre Des Variables	41
4.7	Implantation des techniques de représentation et de minimisation des systèmes concurrents	41
4.8	Conclusion	42

CHAPITRE 5 : THEORIE DU MODEL-CHECKING

5.1	Introduction	44
5.2	Modèles Linéaires	45
5.2.1	La Logique Temporelle Linéaire (LTL)	45
5.3	Model-checking à la Volée	49
5.3.1	Les Automates de Büchi.	49
5.3.2	Quelques définitions de Base	50
5.4	L'Algorithme du Model-checking d'une Formule de LTL	52
5.4.1	Transformation d'un graphe de Kripke en un graphe de Buchi	52
5.4.2	Implémentation de la Procédure du Model-checking	53
5.4.3	Transformation d'une formule LTL en Automate de Büchi	54
5.5	Modèles Arborescents	57
5.5.1	La Logique arborescente CTL	57
5.5.2	Opérateurs Temporels Auxiliaires	59
5.5.3	Axiomatisation de la logique CTL	60
5.6	Model-checking de CTL	61
5.6.1	Calcul des points fixes de fonctions monotones : une approche pour CTL	62
5.6.2	Caractérisation des points fixe pour CTL	65
5.6.3	La génération de Contre-exemples.	68
5.7	Vérification des Systèmes Temps-réel	68
5.7.1	Modélisation des systèmes temps-réel	69
5.7.2	Systèmes de Transitions Temporisées	74
5.7.3	Composition d'Automates Temporisés	75
5.7.4	La Logique Temporelle Temporisée (TCTL)	76
5.7.5	Conclusion	77

CHAPITRE 6 : LA LOGIQUE DE REECRITURE, MAUDE ET FULL-MAUDE

6.1	Introduction	79
6.2	La réécriture	79
6.2.1	Quelques concepts fondamentaux	79
6.3	Notions de base des systèmes de réécriture	81
6.3.1	Syntaxe des systèmes de réécriture	81
6.4	Spécification équationnelle (syntaxe-sémantique)	82
6.4.1	Notions Générale	82
6.4.2	La déduction dans les Systèmes de réécriture	83
6.5	La logique de réécriture	85
6.5.1	Introduction	85
6.5.2	Expression de la Logique de Réécriture dans La Théorie des Catégories	86
6.6	Différents Type d'Algèbres Engendrées par la Réécriture	87
6.6.1	Many Sorted Algebra	88
6.6.2	Order-Sorted Algebra	89
6.7	Le Langage Maude	89
6.7.1	Généralités	89
6.7.2	Expression des Communications Synchrones et Asynchrones	91
6.7.3	Caractéristique du Module META-LEVEL et de la Réflexions	92
6.7.4	Full-Maude	94
6.8	Conclusion	94

CHAPITRE 7 : SPECIFICATION DES SYSTEMES TEMPS REEL DANS LE CADRE DE LA LOGIQUE DE REECRITURE

7.1	Introduction	96
7.1.1	Spécification des systèmes temps réels	96
7.2	Modèle temporisé	97
7.3	Expression du temps comme une action	99
7.4	Logique de réécriture en tant que sémantique pour les systèmes temps réel	100
7.4.1	Application aux automates temporisés	100
7.4.2	Systèmes temps-réel orienté-objets	101
7.4.3	Exemple d'application	101
7.5	Conclusion	102

CHAPITRE 8 : PROPOSITION D'UN ENVIRONNEMENT POUR LA MODELISATION DES SYSTEMES REACTIFS

8.1	Introduction.	104
8.2	Choix d'une méthodologie de modélisation	105
8.2.1	Choix d'un langage pour la méthodologie	106
8.3	La notation UML (Unified Modelling Language)	107
8.3.1	Les bénéfices d'UML	107
8.3.2	Les Classes	108
8.3.3	Diagramme de Classes	109
8.3.4	Diagramme de Séquences	109
8.3.5	Diagrammes de Collaboration	111
8.3.6	Diagramme de Statecharts (Diagramme de transitions)	112
8.3.7	Diagramme d'Activité	114
8.4	Vue Générale Sur le Langage OCL	115
8.4.1	POURQUOI OCL?	115
8.4.2	DANS QUEL CAS UTILISER OCL	116
8.4.3	Invariants	116
8.4.4	Expressions Générales	117
8.4.5	Caractéristiques Prédéfinies sur Tous les Objets	119
8.4.6	Collection	120
8.4.7	Valeurs Précédentes dans Les Post-conditions	121
8.5	Modélisation du système Train-Gate-Controller en UML/OCL	122
8.5.1	L'approche de modélisation par le langage UML	122
8.5.2	Spécification de Contraintes en OCL	123
8.6	L'abstraction des contraintes temporelles en UML/OCL	126
8.7	Traduction du langage UML vers Maude.	126
8.7.1	Contraintes: passage de l'OCL vers Maude	129
8.8	UML et le temps réel	131
8.8.1	Une représentation limitée du temps	131
8.8.2	Illustration d'UML pour la Modélisation des Systèmes Temps réel	132
8.8.3	Une approche pour la Spécification des Systèmes temps-réel en UML	133
8.8.4	Spécification des Systèmes Temps-réel	133
8.8.5	Spécification de l'Extension du Langage UML	135
8.8.6	Exemple de Spécification-Stéréotypes	136
8.9	Conclusion	137

CHAPITRE 9 : IMPLANTATION

9.1	Introduction	139
9.2	Module de Spécification et de Simulation des Systèmes Réactifs	139
9.2.1	Le Système VALID	139
9.2.2	Processus de Modélisation dans VALID	143
9.3	Simulation et Animation dans VALID 2	147
9.4	Transformation de Spécifications VALID en MAUDE	150
9.4.1	Vérification de la terminaison et de la confluence de la spécification.	150
9.4.2	Génération de Code MAUDE à Partir de Spécifications VALID	151
9.5	Proposition d'un Model Checker basé sur la Logique Temporelle PTL	152
9.6	Expérimentation	154
9.6.1	Exemple de Spécification d'un Système Hardware	154
9.6.2	Exemple de Spécification d'un Software	154
9.7	Conclusion	157
	Conclusion et Perspectives	157
	Bibliographie	157
	Annexe A	174
	Annexe B	178

Résumé : La technologie des systèmes mixtes matériels et logiciels connaît de nos jours une révolution sans pareille dans tous les domaines technologiques. Le fait de mélanger matériel et logiciel, ceci a donné naissance à des systèmes très complexes. La conception de tels systèmes, nécessite l'intervention de ressources humaine, financière et technologique très importante. Pour veiller au bon fonctionnement de leurs conceptions, ces systèmes doivent être soumis à des processus de validation très rigoureux. Une telle tâche n'est point évidente vue l'accroissement rapide de la complexité et de l'hétérogénéité des nouvelles architectures (matériel/logiciel). Aussi correcte que possible, la conception doit tenir compte de certaines propriétés dites critiques, qui sans leur présence risque d'engendrer une situation de catastrophe et même de ternir l'image de la compagnie.

Dans ce travail, nous nous sommes intéressées à la mise en place d'un environnement intégré pour le développement des systèmes réactifs temps-réel. A cet effet, nous avons proposé plusieurs solutions pour la modélisation, la spécification et la vérification de plateformes matériel/logiciel. Dans un premier temps, nous avons utilisé une technique simple et efficace pour implémenter les diagrammes de décision binaire utilisés pour la représentation et la vérification des réseaux booléens symbolisant les systèmes réactifs. Nous avons ensuite, montré que l'axiomatisation des logiques temporelles linéaire LTL et arborescente CTL, permet de cerner de peu le problème de la complexité combinatoire spacio-temporelle. Les automates temporisés et la logique TCTL ont permis de mettre en place un modèle très utile pour la spécification et la vérification des systèmes temps-réels.

En se basant d'un côté sur la logique de réécriture comme modèle de spécification algébrique et philosophie universelle de raisonnement formel, et du système Maude comme outil de programmation et d'exécution de ces spécifications, nous avons intégré le tout en tant que noyau de programmation de l'environnement en question et sur lequel un ensemble de modules intégrés et spécialisés sont venus se greffer. Nous avons d'autant plus compris durant cette recherche, que la description UML et son extension UML/OCL et UML-RT, constituent pour le moment les meilleurs moyens pour la modélisation "semi-formelle" des systèmes réactifs temps-réel. Finalement, nous avons montré l'aisance même de l'utilisation de cet environnement à travers plusieurs exemples et benchmarks connus.

Mots-clés : Spécification, Vérification, ROBDD, Logique de Réécriture, Maude, UML/OCL, UML-RT, Méthodes Formelles, Systèmes Réactifs Temps-réel.

Abstract : With the increasing emergence of hardware/software distributed systems, designers need in some ways efficient methods and tools to improve the safeness of such systems and to reduce drastically their design engineering time. In general, some traditional methods as simulation and testing are used to detect errors and bugs, but they are inadequate to certify the correctness of complex systems. To accelerate the design and to avoid distributed systems malfunctions, new mathematical-based techniques have been used to improve the deficiency of the old methods. During the last two decades, Model-checking and theorems proving have been the major research verification fields intensively used to check the design correctness of some big projects (NASA lance rockets, Pentium chips, etc.) In fact, despite the generalization of formal methods, some problems remain in suspense in producing coherent specification fully integrated semantics and avoiding the size complexity of designs.

In this Thesis, we present a hierarchical approach and an open environment to describe and to verify distributed and concurrent systems. The system currently uses UML and extension UML/OCL and UML-RT notations and provides rewriting logic, model checking, theorem proving, and simulation techniques. The Model checking algorithm developed in this research work is based on the axiomatization of LTL linear temporal model and tree temporal logic CTL. We have used temporal automata and TCTL to describe formal behavioral description and to verify timed-based properties. Our experience has shown that combining Temporal Logics and a theoretical well based techniques as the Rewriting logic embodied in the core of the fastest formal system (Maude), can improve the verification of the correctness of large circuits and critical software, and can decrease the time-consuming to perform a solution. Such good solutions are obtained due to the strategy rules incorporated in Maude system which are based on the recent theoretic researches by the rewriting community. When one would like to analyze the results depicted in the benchmark experiments results, he will be surprised because due to the execution CPU time which is, in general insignificant comparing with other verification commercial tools.

Keywords : Specification, Verification, ROBDD, Rewriting Logic, Maude Language, UML/OCL, UML-RT, Formal Methods, Reactive Real-Time Systems.