

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي  
جامعة العقيد الحاج لخضر - باتنة

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ COLONEL HADJ LAKHDAR - BATNA  
Faculté des Sciences de l'ingénieur كلية الهندسة  
Département d'informatique قسم الإعلام الآلي

UNE APPROCHE DE MODÉLISATION  
DES SYSTÈMES À ÉVÈNEMENTS  
DISCRETS UTILISANT LES  
CONCEPT-MAPS

THÈSE

*Présentée et soutenue publiquement le :*

20 Février 2011

*Pour l'obtention du grade de :*

**Doctorat en science**

(Spécialité : informatique)

Par

**BOUROUIS ABDELHABIB**

Devant le jury composé de :

Président	: BENMOHAMMED	MOHAMED	PROFESSEUR	UNIVERSITÉ DE CONSTANTINE
Rapporteur	: BELATTAR	BRAHIM	M. C. A	UNIVERSITÉ DE BATNA
Examineurs	: BILAMI	AZEDDINE	PROFESSEUR	UNIVERSITÉ DE BATNA
	: MAAMRI	RAMDANE	M. C. A	UNIVERSITÉ DE CONSTANTINE
	: ZIDANI	ABDELMADJID	M. C. A	UNIVERSITÉ DE BATNA
	: CHAOUI	ALLAOUA	M. C. A	UNIVERSITÉ DE CONSTANTINE

UNIVERSITÉ DE BATNA - 2011 -

# Résumé-Abstract

## Résumé

Les travaux réalisés dans ce manuscrit s'inscrivent dans le cadre de l'ingénierie dirigée par les modèles. En partant d'une investigation des possibilités d'utilisation des concept-maps dans les différentes étapes d'un projet de modélisation et de simulation, nous avons développé un formalisme visuel sous forme de concept-map pour la modélisation des systèmes à événements discrets. Ce formalisme baptisé EQNM<sup>2</sup>L s'inspire de la théorie des réseaux de files d'attente et s'avère extensible et expressif. Un format d'échange standard basé sur XML a été proposé pour favoriser l'interopérabilité entre les outils de simulation et d'analyse. Dans la perspective d'automatiser la génération du code de simulation (ou d'analyse), la transformation de modèles nous a servi de moyen pour obtenir un code Java complet selon la bibliothèque JAPROSIM développé aussi dans le cadre de ces recherches. La transformation de modèles a été réalisée utilisant XSLT.

## Mots clés

Concept-map, Systèmes à Événements Discrets, Langages Visuels, Explication Participative, Modélisation, IDM, DSML, Théorie des files d'attente, Évaluation de performances.

## Abstract

Research accomplished in this thesis adhere to the model driven engineering. Starting with an investigation of possible uses of concept-maps in different stages of a modeling and simulation project, we developed a visual formalism as a concept map for modeling discrete event systems. This formalism called EQNM<sup>2</sup>L based on the theory of queueing networks and is extensible and expressive. A XML-based standard exchange format has been proposed to promote interoperability between simulation and analysis tools. In the context of automated simulation (or analysis) code generation, model transformation has served as a means to obtain a complete Java code using the JAPROSIM library also developed through this research. The model transformation has been achieved using XSLT.

## KeyWords

Concept-map, Discrete Event Systems, Visual Languages, Participatory Explanation, Modeling, MDE, DSML, Queueing Theory, Performance Evaluation.

# Table des matières

Remerciements . . . . .	ii
Dédicace . . . . .	iii
<b>Résumé</b>	<b>iv</b>
<b>Introduction générale</b>	<b>v</b>
Problématique . . . . .	v
Objectifs spécifiques . . . . .	vi
Organisation de la thèse et organisation du travail . . . . .	vi
<hr/>	
<b>I Concept-maps et langages visuels</b>	<b>1</b>
<b>1 Fondements des concept-maps</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Origines . . . . .	2
1.3 Fondements psychologiques . . . . .	3
1.4 Fondements épistémologiques . . . . .	5
1.5 Définition . . . . .	6
1.6 Création de concept-maps . . . . .	6
1.7 Domaines d'application . . . . .	7
1.8 Techniques similaires . . . . .	8
1.8.1 Mind Map . . . . .	8
1.8.2 Cluster Map . . . . .	8
1.8.3 Argument-Map . . . . .	8
1.8.4 Topic Map . . . . .	9
1.8.5 Text Graph . . . . .	10
1.9 Conclusion . . . . .	10
<b>2 Langages visuels et spécifiques au domaine</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Les graphes . . . . .	14
2.3 Les langages visuels . . . . .	14
2.4 Classification des langages visuels . . . . .	14
2.4.1 Selon les systèmes de programmation . . . . .	15
2.4.2 Selon la technique de spécification . . . . .	15
2.4.3 Selon le domaine d'application . . . . .	15
2.4.4 Selon les relations spatiales . . . . .	15
2.5 Les langages spécifiques au domaine . . . . .	16
2.5.1 Le développement des DSLs . . . . .	17
2.5.2 Outils de développement des DSLs . . . . .	19

2.5.3	Architecture d'une DSM	20
2.6	Conclusion	22
<b>3</b>	<b>Ingénierie dirigée par les modèles et spécification formelle des DSMLs visuels</b>	<b>23</b>
3.1	Introduction	24
3.2	Ingénierie dirigée par les modèles	24
3.2.1	Principes	24
3.2.2	Modèle, métamodèle et mégamodèle	25
3.2.3	Transformation de modèles	26
3.2.3.1	Outils de transformation	28
3.2.4	Transformation basée sur la méta-modélisation	29
3.2.4.1	Définition des règles de transformation	29
3.2.4.2	Expression des règles de transformation	30
3.2.4.3	Exécution des règles de transformation	30
3.2.5	Transformation de graphes	31
3.3	Spécification formelle des DSMLs visuels	31
3.3.1	Les profils UML	31
3.3.2	La syntaxe	32
3.3.3	Sémantique formelle	34
3.3.3.1	Sémantique axiomatique	34
3.3.3.2	Sémantique dénotationnelle	35
3.3.3.3	Sémantique opérationnelle	35
3.4	Conclusion	36
<b>4</b>	<b>Concept-maps en modélisation et simulation</b>	<b>37</b>
4.1	Introduction	37
4.2	Projet de modélisation et de simulation	37
4.2.1	Formulation du problème	38
4.2.2	Objectifs et plan du projet	41
4.2.3	Élaboration du modèle conceptuel	43
4.2.4	Collecte de données	45
4.2.5	Translation du modèle	46
4.2.6	Vérification et validation du modèle	47
4.2.6.1	Techniques de validation subjectives	48
4.2.6.2	Techniques de validation objectives	49
4.2.7	Conception des expérimentations	49
4.2.8	Analyse des résultats	49
4.2.9	Documentation et implémentation	50
4.3	Choix d'un outils de simulation	50
4.4	L'explication en simulation	53
4.4.1	Connaissances explicatives	54
4.4.2	Connaissances factuelles du domaine de simulation	54
4.4.3	Stratégies d'explication	55
4.4.4	Le module d'explication	56
4.4.4.1	Architecture du module d'explication	56
4.5	Conclusion	58
<b>II</b>	<b>Conception et implémentation</b>	<b>60</b>
<b>5</b>	<b>La bibliothèque Japrosim</b>	<b>61</b>
5.1	Introduction	61

5.2	JAPROSIM : une vue générale . . . . .	61
5.3	Simulation à événements discrets par interaction de processus en JAVA . . . . .	62
5.4	Librairies similaires . . . . .	63
5.5	Conception et paquetages . . . . .	64
5.5.1	Le noyau . . . . .	64
5.5.2	Le paquetage RANDOM . . . . .	66
5.5.3	Le paquetage STATISTICS . . . . .	69
5.5.4	Le paquetage UTILITIES . . . . .	69
5.6	Scénario simple de files d'attente . . . . .	69
5.6.1	Solution analytique . . . . .	70
5.6.2	Solution par simulation utilisant JAPROSIM . . . . .	71
5.7	La collecte automatique des statistiques . . . . .	72
5.8	Conclusion . . . . .	73
<b>6</b>	<b>Extended Queueing Networks Modeling and Markup Language (EQNM<sup>2</sup>L)</b>	<b>76</b>
6.1	Introduction . . . . .	76
6.2	Phase de décision . . . . .	77
6.3	Phase d'analyse : notions de base . . . . .	78
6.3.1	Classes et priorités . . . . .	78
6.3.1.1	Classes . . . . .	78
6.3.1.2	Priorités . . . . .	79
6.3.2	Stations de service . . . . .	79
6.3.2.1	Station asymétrique . . . . .	79
6.3.2.2	Station Fork/Join . . . . .	79
6.3.2.3	Possession simultanée de ressources . . . . .	80
6.3.2.4	Choix . . . . .	80
6.3.3	Types de service . . . . .	80
6.3.3.1	Traitement par lot . . . . .	80
6.3.3.2	Préemption . . . . .	80
6.3.3.3	Blocage . . . . .	80
6.3.3.4	Service dépendant de la charge . . . . .	81
6.3.3.5	Pannes des serveurs . . . . .	81
6.3.4	Arrivées (Entrées) . . . . .	81
6.3.5	Départs (Sorties) . . . . .	81
6.3.6	Stratégie de routage . . . . .	81
6.3.7	Région à capacité limitée . . . . .	82
6.4	Phase de conception . . . . .	82
6.4.1	Le métamodèle . . . . .	82
6.4.2	Sémantique statique . . . . .	84
6.4.3	Syntaxe concrète et langage visuel . . . . .	85
6.4.4	Dialecte XML . . . . .	85
6.4.4.1	Interopérabilité . . . . .	85
6.4.4.2	Format d'échange . . . . .	87
6.5	Phase d'implémentation . . . . .	90
6.5.1	GME ToolKit . . . . .	90
6.5.2	L'environnement de modélisation graphique d'EQNM <sup>2</sup> L . . . . .	92
6.5.3	Génération automatique de code . . . . .	94
6.5.3.1	Métamodèle JAVA . . . . .	94
6.5.3.2	Spécification de règles . . . . .	97
6.6	Conclusion . . . . .	98
	<b>Conclusion générale et perspectives</b>	<b>100</b>