

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE**

**UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE  
HOUARI BOUMEDIENNE  
USTHB**

**INSTITUT D'INFORMATIQUE**

**Thèse  
pour l'obtention du grade de  
Magistère**

**Spécialité :  
INFORMATIQUE**

**Présentée par  
Missous Fatma-Zohra**

**Sujet**

**Spécialisation du système orienté objet Exodus  
aux applications multimédias**

**Jury :**

**M<sup>me</sup> Mokhtari Aicha, Maître de conférence, USTHB  
M<sup>me</sup> Alimazighi Zaia, Chargée de cours, USTHB  
Mr M. Ahmed-Nacer, Maître de conférence, USTHB  
M<sup>r</sup> Atif, Maître de conférence, USTHB  
M<sup>r</sup> Boufaïda, Maître de conférence, U. de constantine  
M<sup>me</sup> F. Azrou, Chargée de cours, USTHB**

**Présidente  
Rapporteur  
Examineur  
Examineur  
Examineur  
Examineur  
Année : 1998**

## Résumé

Notre sujet de thèse de Magistère se situe dans le cadre des systèmes de gestion de bases de données orientées objets. Il consiste en la spécialisation du système orienté objet Exodus aux applications multimédias. Le projet a été défini pour pallier aux limites du modèle relationnel pour gérer des données hiérarchiques tels que les documents. Son objectif principal est d'aboutir à un système de gestion de bases de données orientées objets prenant en compte les liens complexes qui peuvent exister entre données multimédias.

Dans ce sens, le système orienté objet Exodus, constitué d'un système de stockage orienté objet et d'un langage de programmation orienté objet persistant, nommé E, a été pris comme noyau pour notre système de gestion de bases de données. Ainsi, un langage de définition de données orientées objets prenant en compte les différents liens de composition qui peuvent exister entre des données complexes, a été mis en œuvre autour du système Exodus. Par la suite, un langage d'interrogation de base de données représentées sous le modèle précédent, a été implémenté à travers une algèbre orienté objet [Van91]. Enfin, pour ne pas négliger le problème de l'évolution de schéma dans le cas des applications multimédias, une étude a été faite en proposant une approche de propagation de modification de schéma sur les instances de la base de données, à travers un compromis entre la propagation automatique [Pen87] et la propagation retardée [Kim87b].

**Mots clés :** modèle de données, objet, langage de définition, base de données, Exodus, langage de requêtes, évolution de schéma, règles ECA.

### Abstract :

The subject of our master thesis is situated in the scope of object oriented databases. It consists in the specialisation of the object oriented system Exodus to multimedia applications. The project has been defined to offset the limits of the relational model to manage hierarchical data as documents. Its main goal is to end up in an object oriented database management system taking in account complex links that could exist between multimedia data.

In this way, the object oriented system Exodus, which is composed of an object oriented storing system and a persistent object oriented programming language, named E, has been taken as the Kernel of our database management system. So an object oriented data definition language taking in account the different composite links that could exist between complex objects, has been implemented on the Exodus system. Then a query language of a database represented in the previous data definition language has been implemented through an object oriented Algebra. Finally, for not disregarding the data base schema evolution problem in the case of multimedia applications, an investigation has been made by suggesting a schema modification propagation approach on the instances, through a combination of the automatic propagation and the delayed propagation.

**Key words :** data model, object, data definition language, data bases, Exodus, query language, schema evolution, ECA rules.

# SOMMAIRE

<b>Chapitre I : INTRODUCTION</b> .....	2
<b>Chapitre II : Les systèmes de gestion de base de données orientées-objet : concepts et état de l'art</b> .....	4
II.1. Introduction.....	5
II.2. Principes et terminologies Orientés-Objet .....	5
II.2.1. Notion d'objet.....	5
II.2.2. Identité d'objet .....	5
II.2.3. Les méthodes .....	5
II.2.4. Encapsulation.....	5
II.2.5. Les classes.....	6
II.2.6. Le lien d'héritage entre classes.....	6
II.3. Les systèmes de gestion de base de données orientées-objet .....	6
II.3.1. Le système Orion.....	6
II.3.1.1. Le modèle de données.....	6
II.3.1.2. Le langage de Requêtes.....	8
II.3.1.3. Gestion de schéma .....	8
II.3.1.4. Architecture.....	9
II.3.1.5. Gestion d'objets.....	10
II.3.1.6. Gestion de transactions.....	10
II.3.2. Le système O2 .....	11
II.3.2.1. Le modèle de données .....	11
II.3.2.2. Le langage de requêtes .....	12
II.3.2.3. Gestion de schémas.....	12
II.3.2.4. Gestion de classe dans O2.....	12
II.3.2.5. Architecture.....	13
II.3.2.6. Le gestionnaire d'objets de O2 .....	13
II.3.2.7. Gestion de transactions.....	14
II.3.3. Le système GEODE.....	14
II.3.4. Le système Exodus.....	16
II.3.5. Conclusion.....	16
<b>Chapitre III : Le système Exodus</b> .....	18
III.1 Introduction.....	19
III.2. L'approche du système Exodus .....	19
III.3. Architecture du système Exodus .....	20
III.3.1 Le gestionnaire de stockage.....	21
III.3.1.1 Le gérant d'E/S physique.....	21
III.3.1.2 Le gérant du buffer.....	22
III.3.1.3 Le support d'objet.....	22
III.3.1.3.1 Gestion des versions.....	22
III.3.1.3.2 Gestion des transactions.....	23
III.3.1.3.3 Le regroupement des objets.....	23

III.3.1.4 L'interface d'Exodus.....	24
III.3.2 Le langage E et son compilateur.....	24
III.3.2.1 La classe collection.....	25
III.3.2.2 Les itérateurs.....	26
III.3.2.3 Gestion des transactions.....	27
<b>Chapitre IV : Le modèle de données.....</b>	<b>29</b>
IV.1. Conception du modèle de données.....	30
IV.1.1. Présentation du modèle de données Extra.....	30
IV.1.1.1. Définition des types de schéma .....	30
IV.1.1.2. Définition des méthodes dans Extra.....	32
IV.1.1.3. Les objets composés.....	33
IV.1.2. Quelques modifications.....	34
IV.2. Implémentation du modèle de données.....	36
IV.2.1. Introduction .....	36
IV.2.2. Implémentation des constructeurs base de données.....	36
IV.2.3. Architecture de l'interpréteur.....	37
IV.2.3.1. Analyse lexicale .....	38
IV.2.3.2. Analyse syntaxique.....	39
IV.2.3.3. Analyse sémantique.....	39
IV.2.3.3.1. Vérification de la validité sémantique du schéma conceptuel.....	39
IV.2.3.3.2. Génération de code source E.....	39
IV.2.3.3.3. Remplissage du catalogue.....	41
<b>Chapitre V : Le langage de requêtes.....</b>	<b>45</b>
V.1. Conception du langage de requêtes.....	46
V.1.1. Introduction .....	46
V.1.2. Choix de conception.....	46
V.1.3. Présentation informelle.....	47
V.1.3.1 Requêtes sur le tuple.....	47
V.1.3.2 Requête sur l'ensemble.....	47
V.1.3.3 Requête sur la liste.....	47
V.1.3.4 Requête sur les collections imbriquées.....	48
V.1.3.5 Requête de jointure.....	48
V.1.3.6 Comment éviter des résultats "oids".....	48
V.1.3.7 Requête utilisant les fonctions agrégats.....	49
V.1.4. Présentation de l'algèbre orientée-objet.....	49
V.1.4.1 Les structures de l'algèbre.....	49
V.1.4.2 Les opérateurs algébriques.....	51
V.1.4.2.1 Les opérateurs sur les multi-ensembles.....	51
V.1.4.2.2. Les opérateurs sur le tuple.....	53
V.1.4.2.3. Les opérateurs sur un vecteur.....	55
V.1.4.2.4 Les opérateurs sur les références.....	56
V.1.4.2.5 Un opérateur de prédicat.....	57
V.1.4.2.6 Exemples sur les opérateurs algébriques.....	57

V.2. Implémentation du langage de requêtes.....	59
V.2.1. Les générateurs de compilateur lex et yacc.....	59
V.2.1.1 L'Analyseur lexical .....	61
V.2.1.2 Analyseur syntaxique.....	61
V.2.1.3 Analyse sémantique.....	61
V.2.1.3.1 Validité sémantique.....	61
V.2.1.3.2. Le traducteur du langage.....	63
V.2.1.3.2.1. Présentation du traducteur.....	65
V.2.1.3.2.2. Le schéma général de traduction d'une requête.....	66
V.2.1.3.2.3. Exemples illustratifs.....	69
V.2.1.3.3. L'interpréteur de l'arbre de requête.....	74
V.2.1.3.3.1 Présentation de l'interpréteur de l'arborescence.....	74
<b>Chapitre VI : Evolution de schéma dans les bases de données orientées objet.....</b>	<b>77</b>
VI.1. Etat de l'art.....	77
VI.1.1. Introduction.....	77
VI.1.2. Orion.....	78
VI.1.2.1. Le modèle.....	78
VI.1.2.2. L'évolutivité.....	78
VI.1.2.2.1. Taxonomie et sémantique d'évolution de schéma...78	
VI.1.2.2.1.1. Changements relatifs à la définition d'une classe.....	78
VI.1.2.2.1.2. Changements sur les liens is-a.....	79
VI.1.2.2.1.3. Changements sur la classe Objet.....	79
VI.1.2.2.2. Invariants d'évolution de schéma.....	79
VI.1.2.2.3. Règles d'évolution de schéma.....	79
VI.1.2.2.3.1. Règles de résolution de conflit.....	80
VI.1.2.2.3.2. Règles de propagation des modifications portant sur les propriétés des classes.....	80
VI.1.2.2.3.3. Règles de manipulation du graphe.....	81
VI.1.2.2.3.4. Règles concernant les objets composés.....	82
VI.1.2.2.3. Impact de l'évolutivité.....	82
VI.1.2.3.1 Dynamique du schéma.....	82
VI.1.2.3.2 Gestion de versions.....	82
VI.1.3. Gemstone.....	86
VI.1.3.1. Le modèle.....	86
VI.1.3.2. L'évolutivité.....	86
VI.1.4. Encore.....	87
VI.1.4.1. Le modèle.....	87
VI.1.4.2. L'évolutivité.....	87
VI.1.5. Remarque.....	88
VI.2. Combinaison de la propagation automatique et de la propagation retardée.....	90
VI.2.1. Introduction.....	90
VI.2.1.1. Conversion.....	90

---

VI.2.1.1.1 Conversion immédiate.....	90
VI.2.1.1.2. Conversion différée.....	90
VI.2.1.2. Emulation.....	90
VI.2.1.3. Versionnement d'objets.....	90
VI.2.2. Modèle.....	92
VI.2.2.1. Les règles ECA.....	92
VI.2.2.2 Modes de couplage et les transactions.....	92
VI.2.2.3 Modèle de transaction imbriqué étendu.....	92
VI.2.3. Combinaison de la propagation retardée et de la propagation automatique.....	95
VI.2.3.1. Comment assembler la propagation retardée et la propagation automatique ?.....	96
VI.2.3.2. Mise en oeuvre de notre approche de propagation.....	99
VI.2.4. Conclusion.....	101
<b>Chapitre VII : Conclusion.....</b>	<b>103</b>