

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université El Hadj Lakhdar - Batna

Faculté des sciences de l'ingénieur

Département d'informatique

Internet

Mémoire

présenté en vue de l'obtention du diplôme de

Magister en Informatique

Option : Informatique industrielle

Internet

**Intégration des moyens de modification dynamique des
contenus sur le WEB**

Internet

Samir ADOUANE

Jury :

Dr. S. Chikhi

Pr. M. Benmohammed

Dr. A. Zidani

Dr. S. Zidet

Université de Constantine

Université de Constantine

Université de Batna

Université de Batna

Président

Rapporteur

Examineur

Examineur

RESUME

Le monde de conception des applications est enrichi chaque jour par de nouveaux designs architecturaux, répondant à des problèmes d'implémentation récurrents, la majorité de ces solutions tournent autour de la programmation orientée objet qui facilite leur intégration et mise en œuvre.

Dans ce mémoire nous proposons une approche basée sur le modèle MVC que nous appellerons « les délégués des packages », cette approche vise à faciliter et à accélérer le processus de développement web. Elle tente d'apporter des éléments pratiques, facilitant le déploiement du modèle MVC dans un contexte de la programmation orientée objet.

Mots-clés : Modélisation, Méthode de conception, UML, Design Patterns, XML, XMI, application web, Entrepôt de données.

ABSTRACT

The world of applications design is enriched every day by new architectural designs that answer recurring implementation problems. The majority of these solutions address the Object-Oriented Programming (OOP) which facilitates their integration and implementation.

In this report we propose an approach based on the MVC model which we shall call "packages delegates", this approach aims to facilitate and to accelerate the process of web development. moreover it tries to come up with practical elements, facilitating the deployment of the MVC model in an OOP context.

Keywords : Modeling, Conception method, UML, Design Patterns, XML, Web application, Data Warehouse.

SOMMAIRE

| | |
|--|-----------|
| INTRODUCTION | 2 |
| LES APPLICATIONS WEB | 6 |
| 1.1 Introduction | 6 |
| 1.2 Définition d'une application WEB | 6 |
| 1.3 Sites ou applications web? | 6 |
| 1.4 Utilisations courantes des applications Web | 7 |
| 1.5 Le processus de développement | 7 |
| 1.5.1 Le modèle | 9 |
| 1.5.2 Le workflows | 10 |
| 1.5.3 Les risques | 14 |
| 1.5.4 Les itérations | 14 |
| 1.6 Contenu des sites web | 14 |
| 1.6.1 Page dynamique et page statique | 14 |
| 1.6.2 Traitement des pages Web statiques | 15 |
| 1.6.3 Traitement des pages dynamiques | 16 |
| 1.6.4 Accès à une base de données | 16 |
| 1.6.5 Création de pages dynamiques | 18 |
| 1.6.6 Choix d'une technologie de serveur | 20 |
| 1.7 Patterns architecturaux pour les applications WEB | 20 |
| 1.8 Sécurité des application web | 21 |
| 1.8.1 Introduction | 21 |
| 1.8.2 Principes fondamentaux de la sécurité | 21 |
| 1.8.3 Menaces, vulnérabilités et attaques : définitions | 22 |
| 1.8.4 Comment créer une application Web sécurisée ? | 23 |
| 1.8.5 Aspects architecturaux pour la conception des applications Web | 23 |
| 1.9 Critères ergonomiques pour l'évaluation des interfaces Web | 27 |
| 1.9.1 Guidage | 27 |
| 1.9.2 Charge de travail | 28 |
| 1.9.3 contrôle explicite | 28 |
| 1.9.4 Adaptabilité | 29 |
| 1.9.5 Gestion des erreurs | 29 |
| 1.9.6 Homogénéité/Cohérence | 29 |
| 1.9.7 Signifiante des codes et dénominations | 30 |
| 1.9.8 Compatibilité | 30 |
| Conclusion | 30 |
| TECHNOLOGIE XML | 31 |
| 2.1 Introduction | 31 |
| 2.2 Pourquoi utiliser XML ? | 32 |
| 2.3 Structure d'un document XML | 33 |
| 2.3.1 Les composantes des documents XML | 33 |
| 2.3.2 Unicode | 34 |
| 2.3.3 Unicode et XML | 34 |
| 2.3.4 La question de l'encodage | 34 |
| 2.4 XML opposé à HTML | 35 |
| 2.4.1 Exemple de HTML/XML : | 35 |
| 2.5 Les DTD (Document Type Définition/Définition de Type de Document) | 36 |
| 2.6 Les schémas XML | 38 |
| 2.6.1 langages de schéma | 39 |
| 2.7 DTD ou XML-Schema ? | 40 |
| 2.8 Feuilles de style | 40 |
| 2.8.1 CSS feuilles de style en cascade | 43 |
| 2.9 Les règles du jeu XML | 43 |
| 2.10 Les espaces de nommage (<i>namespaces</i>) | 44 |
| 2.10.1 Problématique: espaces de nommage et validation | 44 |
| 2.11 XML et les bases de données | 45 |

| | |
|---|-----------|
| 2.11.1 XML est-il une base de données ?----- | 46 |
| 2.12 Résumé ----- | 47 |
| Conclusion ----- | 49 |
| MODELISATION UML----- | 50 |
| 3.1 Introduction ----- | 50 |
| 3.2 Les principes de la modélisation ----- | 53 |
| 3.3 Pourquoi l'approche Objet ? ----- | 53 |
| 3.4 Les objets ----- | 55 |
| 3.5 Les diagrammes UML ----- | 56 |
| 3.5.1 Vue fonctionnelle : ----- | 57 |
| 3.5.2 Vue statique : ----- | 57 |
| 3.5.3 Vue dynamique : ----- | 58 |
| 3.6 Nouveauté de UML 2.0 ----- | 59 |
| 3.7 UML est un langage pour communiquer ----- | 60 |
| 3.8 Etude basée sur les cas d'utilisation ----- | 60 |
| 3.9 Le choix d'un outil de modélisation ----- | 62 |
| 3.10 UML et la Méta-modélisation ----- | 65 |
| 3.11 Les Design Patterns ----- | 69 |
| 3.11.1 Définition ----- | 70 |
| 3.11.2 Exemple de design Patterns ----- | 71 |
| APPROCHE DES DELEGUES DES PACKAGES ----- | 74 |
| 4.1 Le Modèle MVC----- | 74 |
| 4.1.1 Introduction ----- | 74 |
| 4.1.2 Problématique ----- | 75 |
| 4.1.3 Facteurs à prendre en compte----- | 75 |
| 4.1.4.Solution ----- | 76 |
| 4.1.5 Variations ----- | 77 |
| 4.1.6 Comportement du modèle actif----- | 77 |
| 4.1.7 Considérations relatives au test ----- | 79 |
| 4.2. Approche des délégués des packages----- | 81 |
| 4.2.1 Introduction : ----- | 81 |
| 4.2.2 Approche des délégués des packages : ----- | 81 |
| 4.2.3. Amélioration du faible couplage M-V-C ----- | 82 |
| 4.2.4 Avantages de Approche des délégués des packages ----- | 83 |
| 4.2.5 Inconvénients Approche des délégués des packages----- | 84 |
| 4.2.6 Exemple d'implémentation de l'approche des délégués des packages ----- | 84 |
| Conclusion ----- | 85 |
| APPLICATION & IMPLEMENTATION----- | 86 |
| 5.1 Introduction ----- | 86 |
| 5.2 Le projet MERLIN ----- | 86 |
| 5.2.1 Contexte ----- | 86 |
| 5.2.2 Dénomination du projet ----- | 87 |
| 5.2.3 Problématique ----- | 87 |
| 5.2.4 Solution ----- | 87 |
| 5.2.5 Technologies employées ----- | 88 |
| 5.2.6 Notion d'indicateur ----- | 88 |
| 5.2.7 Alimentation du Data Warehouse----- | 89 |
| 5.2.8 Diagramme des cas d'utilisation ----- | 90 |
| 5.2.9 Diagramme de séquence : Comportement dynamique du système ----- | 92 |
| 5.2.10 Diagramme des packages ----- | 94 |
| 5.2.11 Utilisation de l'approche des délégués des packages ----- | 94 |
| 5.2.12 Modèle de données : ----- | 96 |
| 5.2.13 Interface (Ecrans)----- | 103 |
| 5.2.14 Avantages constatés avec l'utilisation de l'approche des délégués des packages ----- | 106 |
| 5.3 Site WEB AFPLemans ----- | 107 |

| | |
|--|------------|
| 5.3.1 Préface ----- | 107 |
| 5.3.2 Contexte ----- | 107 |
| 5.3.3 Technologies utilisées ----- | 107 |
| 5.3.4 Choix de la technologie XML ----- | 108 |
| 5.3.5 Structure du site ----- | 108 |
| 5.3.6 Intégration des moyens de modification dynamique ----- | 108 |
| 5.3.7 Exemple d'outils facilitant l'édition des contenus : ----- | 109 |
| 5.3.8 Editeur des contenus intégré ----- | 109 |
| 5.3.9 Le fichier XML----- | 112 |
| 5.3.10 Interface de l'éditeur des contenus ----- | 113 |
| CONCLUSION ----- | 115 |
| ANNEXE A : DATA WAREHOUSE ----- | 117 |
| ANNEXE B : TUTORIAL RELAX NG [CLARK01] ----- | 121 |
| ANNEXE C : GLOSSAIRE DE VOCABULAIRE ----- | 127 |
| REFERENCES BIBLIOGRAPHIES ----- | 135 |