

ETUDES INFORMATIQUES
ET LOGICIELS

collection dirigée par Nicolas Manson

La réalisation optimale de systèmes informatiques

Michel Dehès

hermes

Lavoisier

La réalisation optimale de systèmes informatiques

Michel Dehès



hermes
Science
— publications —

Lavoisier

TABLE DES MATIÈRES

| | |
|--|----|
| Avant-propos | 13 |
| Introduction | 15 |
| Chapitre 1. La démarche pour construire un système | 19 |
| 1.1. Système d'information et système informatique | 19 |
| 1.2. Les architectures d'un système informatique | 20 |
| 1.3. Togaf de l'Open Group | 21 |
| 1.4. <i>Unified Process</i> , le cycle de vie d'un logiciel. | 27 |
| 1.5. La démarche du cycle en « Y » | 29 |
| 1.6. Prise en compte d'un besoin d'entreprise dans le SI | 33 |
| 1.6.1. Les approches <i>top-down</i> et <i>bottom-up</i> | 33 |
| 1.6.1.1. L'approche <i>top-down</i> | 34 |
| 1.6.1.2. L'approche <i>bottom-up</i> | 35 |
| 1.6.2. La démarche pour prendre en compte un besoin dans le SI | 35 |
| Chapitre 2. L'expression des besoins | 39 |
| 2.1. Les enjeux et les objectifs | 39 |
| 2.2. Le besoin de la MOA et les exigences | 39 |
| 2.3. Les exigences fonctionnelles | 41 |
| 2.4. Les exigences techniques. | 46 |
| 2.4.1. Exigences d'utilisation et de performances. | 46 |
| 2.4.1.1. L'ouverture du service. | 46 |
| 2.4.1.2. La volumétrie | 47 |
| 2.4.1.3. Les performances demandées | 47 |

6 La réalisation optimale de systèmes informatiques

- 2.4.2. Exigences de fiabilité du service 48
- 2.4.3. Exigences de sécurité. 49
- 2.4.4. Exigences d'exploitabilité 51
 - 2.4.4.1. Administration des données et de l'applicatif 51
 - 2.4.4.2. Maintenabilité, outil de diagnostic. 51
 - 2.4.4.3. Supervision 51
 - 2.4.4.4. Sauvegardes et conservation des données 51
 - 2.4.4.5. Ordonnancement 51
 - 2.4.4.6. Gestion des changements 52
 - 2.4.4.7. Télédistribution et inventaire 52
- 2.5. Le suivi des exigences 52
- 2.6. Résumé 53

Chapitre 3. L'analyse métier 55

- 3.1. Les enjeux et les objectifs 55
- 3.2. Identifier les enjeux métier et les objectifs 56
- 3.3. Les acteurs, les *use cases* et les processus métier impactés. 57
- 3.4. Les objets métier. 61
- 3.5. Les flux métier. 62
- 3.6. La modélisation des exigences 63
- 3.7. Les services métier 64
- 3.8. L'enrichissement de la cartographie métier 65
- 3.9. Résumé 65

Chapitre 4. L'architecture fonctionnelle 67

- 4.1. La démarche pour bâtir l'architecture fonctionnelle 67
- 4.2. La présentation de l'architecture fonctionnelle 68
 - 4.2.1. Le positionnement du système dans le SI. 68
 - 4.2.2. La description de l'architecture fonctionnelle 70
 - 4.2.2.1. Présentation des services et des fonctions offerts
par le système. 71
 - 4.2.2.2. Les services fonctionnels externes utilisés par le système. 76
 - 4.2.2.3. Les flux avec les autres systèmes 77
 - 4.2.3. La structuration interne 80
 - 4.2.3.1. La décomposition du sous-système du SI
en systèmes fonctionnels 80
 - 4.2.3.2. La décomposition des systèmes fonctionnels
en sous-systèmes 80

| | |
|--|------------|
| 4.3. L'analyse système | 82 |
| 4.3.1. Le modèle détaillé des cas d'utilisation | 84 |
| 4.3.2. La description détaillée des <i>use cases</i> | 86 |
| 4.3.3. L'identification des objets | 88 |
| 4.3.4. Le modèle statique du système | 88 |
| 4.3.4.1. Identification des liens entre les objets | 88 |
| 4.3.4.2. Etablissement des responsabilités | 91 |
| 4.3.5. Le modèle dynamique du système | 95 |
| 4.3.5.1. Identifier les scénarios | 96 |
| 4.3.5.2. Diagramme de séquence « enregistrerArticle » | 97 |
| 4.3.5.3. Diagramme de séquence « terminerSaisie » | 101 |
| 4.3.5.4. Diagramme de séquence « enregistrerPaieement » | 103 |
| 4.3.5.5. Identification des états | 107 |
| 4.4. L'enrichissement de l'urbanisme fonctionnel | 111 |
| 4.5. Résumé | 111 |
| | |
| Chapitre 5. L'architecture technique | 113 |
| 5.1. Les enjeux et les objectifs | 113 |
| 5.2. Les services techniques et les infrastructures techniques | 114 |
| 5.2.1. Les services d'habilitation | 114 |
| 5.2.1.1. Le service d'habilitation interne | 114 |
| 5.2.1.2. Le service d'habilitation externe | 117 |
| 5.2.2. Les <i>middlewares</i> | 119 |
| 5.2.2.1. Cinématique des échanges | 123 |
| 5.2.3. L' <i>Enterprise Service Bus</i> | 123 |
| 5.2.4. La virtualisation des serveurs | 125 |
| 5.2.4.1. ESX Server | 126 |
| 5.2.4.2. Virtual Center | 126 |
| 5.2.4.3. Système de fichiers VMFS | 126 |
| 5.2.4.4. Vmware VMotion | 127 |
| 5.2.4.5. Vmware DRS | 128 |
| 5.2.4.6. Vmware HA | 129 |
| 5.2.5. Un exemple de consigne : la haute disponibilité | 129 |
| 5.2.5.1. Les causes d'incidents et de sinistres | 129 |
| 5.2.5.2. Mécanismes assurant la haute disponibilité | 129 |
| 5.2.5.3. La répartition de charge matérielle | 130 |
| 5.2.5.4. La répartition de charge logicielle | 130 |
| 5.2.5.5. Les <i>clusters</i> | 132 |
| 5.2.5.6. La recopie des données | 133 |

| | |
|---|------------|
| 5.3. La démarche pour bâtir l'architecture technique | 133 |
| 5.4. La description de l'architecture technique. | 136 |
| 5.4.1. Les réponses aux exigences d'utilisation | 136 |
| 5.4.2. Les réponses aux exigences de fiabilité de service | 139 |
| 5.4.2.1. Haute disponibilité | 139 |
| 5.4.2.2. PRA (plan de reprise d'activités) en cas de désastre site. | 142 |
| 5.4.3. Les réponses aux exigences de sécurité | 142 |
| 5.4.3.1. La sécurisation des accès | 142 |
| 5.4.4. Les réponses aux exigences de performances | 144 |
| 5.4.5. Les réponses aux exigences d'exploitabilité | 145 |
| 5.4.5.1. Administration des données, des produits et services techniques | 145 |
| 5.4.5.2. Maintenabilité, outil de diagnostic. | 145 |
| 5.4.5.3. Supervision | 145 |
| 5.4.5.4. Sauvegarde, archivage et historique. | 146 |
| 5.4.5.5. Ordonnancement | 146 |
| 5.4.5.6. Gestion des changements | 146 |
| 5.4.6. Le volet logiciel | 146 |
| 5.4.6.1. Couche présentation | 147 |
| 5.4.6.2. Couche applicative et métier | 147 |
| 5.4.6.3. Couche accès aux données | 147 |
| 5.4.7. Le volet technique | 149 |
| 5.4.8. Le volet production informatique | 151 |
| 5.5. L'analyse technique des choix proposés | 152 |
| 5.6. L'enrichissement de l'urbanisme technique. | 152 |
| 5.7. Résumé | 153 |
| | |
| Chapitre 6. L'architecture logicielle. | 155 |
| 6.1. Les enjeux et les objectifs | 155 |
| 6.2. Les règles et les modèles d'architecture logicielle | 156 |
| 6.2.1. L'architecture 3-tiers | 156 |
| 6.2.2. Les <i>design patterns</i> | 157 |
| 6.2.2.1. Le <i>design pattern factory method</i> | 158 |
| 6.2.2.2. Le <i>design pattern state</i> | 160 |
| 6.2.2.3. Le <i>design pattern façade</i> | 161 |
| 6.2.3. Le modèle JEE (anciennement J2EE) | 163 |
| 6.2.3.1. Couche présentation avec JSP et <i>servlets</i> | 164 |
| 6.2.3.2. Protocole http | 165 |
| 6.2.3.3. <i>Servlet</i> | 165 |
| 6.2.3.4. <i>Java Server Pages</i> | 167 |
| 6.2.3.5. Le modèle de conception MVCII appliqué à JEE. | 168 |

| | |
|--|-----|
| 6.2.3.6. <i>Java Server Faces</i> (JSF) | 169 |
| 6.2.3.7. Composants EJB | 169 |
| 6.2.4. PHP | 170 |
| 6.2.5. La technologie .Net de Microsoft | 172 |
| 6.2.6. Les <i>frameworks</i> | 174 |
| 6.2.6.1. Un <i>framework</i> de gestion de transactions. | 174 |
| 6.2.6.2. Les <i>frameworks</i> Spring et Hibernate | 176 |
| 6.2.7. Les <i>webservices</i> | 181 |
| 6.2.7.1. Le protocole SOAP | 183 |
| 6.2.7.2. Le WSDL | 187 |
| 6.2.7.3. Les piles SOAP | 189 |
| 6.2.7.4. La localisation des <i>webservices</i> | 189 |
| 6.2.7.5. Rest | 190 |
| 6.3. La démarche pour bâtir l'architecture logicielle | 191 |
| 6.4. La description de l'architecture logicielle | 192 |
| 6.4.1. La cohérence avec l'architecture fonctionnelle | 192 |
| 6.4.2. La cohérence avec l'architecture technique. | 193 |
| 6.4.3. La cohérence avec l'urbanisme applicatif. | 193 |
| 6.4.4. Le schéma d'architecture logicielle | 193 |
| 6.4.4.1. Couche présentation | 195 |
| 6.4.4.2. Couche applicative et métier | 195 |
| 6.4.4.3. Couche intégration (accès aux données) | 195 |
| 6.4.4.4. Modèles d'échanges | 196 |
| 6.5. La conception du système | 197 |
| 6.5.1. Identifier les classes participantes aux cas d'utilisation. | 197 |
| 6.5.2. Elaborer les diagrammes dynamiques associés au <i>use case realization</i> | 197 |
| 6.5.3. Simplifier la problématique en créant des packages. | 200 |
| 6.6. L'enrichissement de l'urbanisme applicatif | 201 |
| 6.7. Résumé | 202 |

Chapitre 7. L'implémentation technique. 203

| | |
|---|-----|
| 7.1. Les enjeux et les objectifs | 203 |
| 7.2. La démarche pour implémenter un système. | 204 |
| 7.3. Le modèle de la base de données | 204 |
| 7.4. Le codage | 205 |
| 7.4.1. Exemples de codage | 205 |
| 7.4.1.1. Implémentation de la classe Vente | 205 |
| 7.4.1.2. Le <i>design factory method</i> | 209 |
| 7.4.1.3. Codage des <i>webservices</i> | 211 |
| 7.5. La description physique des bases de données | 214 |

| | |
|--|------------|
| 7.5.1. Notions sur le stockage disque | 214 |
| 7.5.2. Règles de répartition d'une base de données | 215 |
| 7.5.3. Mise en œuvre de la répartition des données | 216 |
| 7.5.4. Règles de répartition des données | 216 |
| 7.6. Le plan de nommage | 219 |
| 7.7. Le plan d'ordonnancement | 220 |
| 7.8. Le plan de sauvegarde | 221 |
| 7.9. Le plan de supervision | 223 |
| 7.9.1. Système de fichiers (surveillance de la taille) | 224 |
| 7.9.2. Présence ou absence d'un fichier | 224 |
| 7.9.3. Présence ou absence d'un mot-clé | 225 |
| 7.9.4. Autres | 225 |
| 7.10. La télédistribution et la gestion des inventaires | 226 |
| 7.11. Les audits | 227 |
| 7.12. Résumé | 227 |
| | |
| Chapitre 8. L'exemple de la refonte de la <i>supply chain</i> | 229 |
| 8.1. L'expression des besoins | 230 |
| 8.1.1. Les exigences fonctionnelles | 230 |
| 8.1.2. Les exigences techniques | 230 |
| 8.2. L'architecture fonctionnelle | 231 |
| 8.2.1. L'analyse métier | 231 |
| 8.2.1.1. Présentation du modèle SCOR | 231 |
| 8.2.1.2. Les <i>use cases</i> et les processus métier impactés | 233 |
| 8.2.1.3. Les objets métier | 235 |
| 8.2.1.4. Les flux métier | 237 |
| 8.2.2. La description de l'architecture fonctionnelle | 238 |
| 8.2.2.1. Le positionnement du système dans le SI | 239 |
| 8.2.2.2. La description détaillée des <i>use cases</i> système | 242 |
| 8.3. L'architecture technique | 243 |
| 8.4. L'architecture logicielle | 247 |
| 8.5. Le développement | 248 |
| 8.6. Résumé | 249 |
| | |
| Conclusion | 251 |
| | |
| Annexes. Les concepts objets et UML | 253 |
| A.1. Les acteurs et les <i>use cases</i> | 253 |
| A.2. Objet/classe | 254 |

| | |
|---|------------|
| A.3. L'héritage | 256 |
| A.4. Le polymorphisme | 258 |
| A.5. UML | 259 |
| A.6. La visibilité entre objets | 261 |
| A.7. Exemple de diagramme de communication. | 265 |
| Glossaire | 269 |
| Index | 277 |

Cet ouvrage présente une démarche globale pour réaliser un système informatique permettant la prise en compte des besoins, aussi bien métiers que techniques, des entreprises ou des organismes, et ce, de manière automatique.

Illustré d'exemples précis et complets, *La réalisation optimale de systèmes informatiques* propose de factoriser les concepts (rassembler les éléments communs) afin de simplifier la problématique tout en assurant la cohérence. Il présente les aspects métiers, fonctionnels, techniques et logiciels puis développe l'implémentation des systèmes. Ces différentes étapes doivent répondre au cahier des charges de la maîtrise d'ouvrage et respecter les règles et les modèles d'architecture tout en s'inscrivant dans les plans d'urbanisme cibles du système d'information.

Par une bonne connaissance de tous ces aspects et la traçabilité de l'information entre les différents niveaux, la réalisation des systèmes est ainsi automatisée, ce qui améliore considérablement la productivité. Les coûts et les délais sont minimisés et la qualité optimisée.

L'auteur

Architecte de systèmes d'information au sein d'un grand groupe international, Michel Dehès possède une longue expérience dans les différentes architectures des systèmes et dans les nouvelles technologies informatiques.