



**Département d'informatique
et de mathématique**
Université du Québec à Chicoutimi

PLAN DE COURS

8INF853 - Architecture des applications d'entreprise

Objectifs généraux du cours

- Identifier les problèmes de développement des applications d'entreprise dans les différentes phases de développement de logiciels, surtout la relation entre les différentes composantes ou modules.
- Décrire les concepts et les techniques qui ont été développés pour faciliter les tâches de développement et d'implémentation des applications d'entreprises.
- Offrir une occasion de *critiquer* des articles existants et de *résoudre* des problèmes relatifs aux architectures d'applications d'entreprises: *Connecteurs, styles architecturaux, couplage, réutilisation, chargement dynamique, EJB (JEE), MVC, intégration (interopérabilité), Développement orienté service (SOA), Cloud Computing, Internet of Things (IoT), Big Data, ...*
- Encourager les étudiants à détecter des défis des applications d'entreprises, à préparer et à présenter des exposés pour montrer ces défis. Les étudiants sont invités à présenter et à étudier les approches existantes pour résoudre tels défis et à proposer des solutions (avec *preuve de concept*) : étude de cas, implémentation, modèle mathématique, etc.
- Offrir une vision globale liée à plusieurs domaines de recherche en informatique : invitation des chercheurs (IBM ou autres) ou collègues pour présenter leurs recherches, outils, etc.

Description officielle du cours

Amener l'étudiant à approfondir les concepts liés au développement et à l'architecture des applications d'entreprises. Amener l'étudiant à acquérir des compétences à travailler dans un environnement de programmation ayant des composantes complexes. L'étudiant sera exposé à la technologie Java, aux modèles objets et à l'architecture orientée services. Enfin, l'étudiant se familiarisera avec certaines technologies touchant les aspects d'infrastructure de développement et de déploiement d'applications d'entreprises.

Concepts fondamentaux d'interfaces usagers. Études de cas avec Java (Swing). Développement et déploiement d'applications d'entreprises (Enterprise applications). Technologie Java. EJB, Java Beans et architecture client-serveur RMI-Java. Architecture orientée Modèles (MDA). Architecture orientée service (SOA).

Description du cours (objectifs spécifiques)

Ce cours se concentre sur les concepts d'architecture d'applications d'entreprises. Nous étudions d'abord les concepts avancés pour le développement d'applications d'"entreprise" (*enterprise applications*) en Java (réflexion, chargement dynamique, EJB). Par la suite, nous allons approfondir les styles architecturaux et montrer, en particulier, les problèmes de couplage et de réutilisation. Les techniques de séparation de préoccupations seront utilisées pour mieux implémenter ces styles. Ces techniques pourraient supporter le couplage faible et améliorer la réutilisation. Le modèle MVC sera étudié comme un style architectural permettant de séparer les composantes logicielles en différentes parties. Finalement, les serveurs d'application et l'architecture orientée service seront étudiés.

Les principales caractéristiques d'une architecture d'applications d'entreprise sont: *réflexibilité, fiabilité, performance, disponibilité, adaptabilité, flexibilité* et *extensibilité*, etc. Ce cours vise à comprendre ces caractéristiques et à familiariser l'étudiant à la recherche et au développement des applications d'entreprise dans l'environnement de l'Internet:

- Maîtriser la terminologie architecturale d'une application
- Acquérir les concepts fondamentaux d'une architecture d'une application d'entreprise donnée
- Étudier les défis de ces applications.
- Introduire le développement orienté service pour les applications d'entreprises.
- Maîtriser les techniques d'architectures logicielles utilisées dans le paradigme orienté objets, composants, et, dans une version simplifiée, dans le paradigme orienté services.

Cours magistraux

Le cours permettra aux étudiants l'apprentissage des éléments suivants (**L'ordre et le contenu de ces éléments peuvent être modifiés**) :

Partie 1:

- Introduction
- Styles architecturaux : filtres, couches, reposeirs, client-serveur, etc.
- Architectures des applications d'entreprises: connecteurs, architectures (C/S, multi-tiers, noyau), Modules, composants

Partie 2:

- Chargement dynamique, plugins, réflexivité, ... Paquetage de réflexion en Java (Class, Method, Field, etc.)
- Sérialisation en Java
- Refactoring

Partie 3:

- Architecture et patrons de conception (application locale)
- Architecture et patrons à distance (remote patterns)
- Model-View-Controller MVC
- Séparation de préoccupations
- Message Oriented Middleware (MOM)

Partie 4:

- Méthode de développement orientée attributs ADD
- Problèmes (Attributs) et solutions (patrons et styles)

Partie 5:

- Architecture des applications avec les serveurs d'applications:
- JEE (EJB), JBoss, Spring, Service Web (SOAP & REST).
 - WebSphere : L'IBM va présenter ce thème en détail.

Partie 6:

- Introduction au développement orienté service (Service-Oriented Computing): SOA et ESB

Références :

1. Notes de cours 8INF843 et liste de sites intéressants (voir ma page web : utilisez vos comptes)
2. Bass, L. (2013). *Software architecture in practice* (3e éd.). Boston: Addison-Wesley.
Disponibilité à la bibliothèque: <http://go.uqac.ca/5pit>
3. Taylor, R. N. (2010). *Software architecture: foundations, theory, and practice*. Hoboken, N.J.: Wiley.
4. Printz, J. (2012). *Architecture logicielle : concevoir des applications simples, sûres et adaptables* (3e éd.). Paris: Dunod.
Accès à la version électronique <http://go.uqac.ca/kwxk>
Disponibilité à la bibliothèque: <http://go.uqac.ca/2z8z>
5. Blanc, X. (2005). *MDA en action : Ingénierie logicielle guidée par les modèles*. Paris, France: Eyrolles.
Accès à la version électronique <http://go.uqac.ca/9hkp>
6. Gamma, E. (1995). *Design patterns: elements of reusable Object-Oriented software*. Reading, Mass.: Addison-Wesley.
Disponibilité à la bibliothèque: <http://go.uqac.ca/KXeN>
7. Horstmann, C. S. (2006). *Object-oriented design & patterns* (2e éd.). Hoboken, N.J.: Wiley.
Disponibilité à la bibliothèque: <http://go.uqac.ca/Atda>
8. Kircher, M., & Jain, P. (2004). *Patterns for resource management (Pattern-oriented software architecture, vol. 3)*. Chichester, England: Wiley.
Disponibilité à la bibliothèque: <http://go.uqac.ca/QiqL>
9. Metsker, S. J. (2006). *Les Design Patterns en Java : les 23 modèles de conception fondamentaux*. Paris: CampusPress.
Disponibilité à la bibliothèque: <http://go.uqac.ca/UtGp>
10. Völter, M., Kircher, M., & Zdun, U. (2005). *Remoting patterns foundations of enterprise, internet and realtime distributed object middleware*. Chichester, England: John Wiley.
Disponibilité à la bibliothèque: <http://go.uqac.ca/CQ5w>
11. Martin Fowler: <http://www.martinfowler.com/articles/writingPatterns.html>
12. REST Architectural Style for distributed systems: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
13. Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns: Designing, Building, and deploying Messaging Solutions*. ISBN: 0321200683.
14. Nicolai M. Josttis, 2007. *SOA in Practice: The Art of Distributed System Design*. O'Reilly, ISBN: 9780596529550.
15. Paul C. Brown. 2008. *Implementing SOA: Total Architecture in Practice*. Addison-Wesley, ISBN: 978321504722.
16. SOA magazine sur <http://www.soamagazine.com>
17. OMG (Object Management Group) : <http://www.omg.org/spec/BPMN/>
18. Coulouris, G., Dollimore, J., Kindberg, T. 2005. *Distributed systems, Concepts and Design*. Third edition. Addison-Wesley.
19. Tanenbaum, Andrew et Maarten Van Steen. 2011. *Distributed Systems: Principles and Paradigms*. Prentice-Hall, 2ième edition, USA.
20. E. Comer M. Douglas. 2009. *Computer Networks and Internets* (5th Edition), Prentice Hall, 5th edition.
21. Budi, Kurniawan. *Java for the web with Servlet, JAS, and EJB*. New Riders. ISBN: 073571195. 2002.

Outils:

Eclipse: <http://www.eclipse.org/galileo/>

NetBeans: <http://netbeans.org/downloads/>

GlassFish (Download): <https://open-esb.dev.java.net/Downloads.html>

UML to Spring:

<http://www.skywayperspectives.org/portal/web/guest/ggfeaturedownload?gclid=CO74oLml50CFU1M5QodwCxe8w>

Explication (brièvement) de BPMN avec BizAgi: <http://www.bpmn.org/>

Introduction to BPMN by Stephen A. White

<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>

BPMN Specifications at <http://www.bpmn.org/>

<http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>

BizAgi Process Modeler works based on the BPMN Standard:

http://wiki.bizagi.com/en/index.php?title=Activity:_User_and_Service_Task

BizAgi Process Modeler visiting our Resource center at <http://bizagi.com>

<http://wiki.bizagi.com/en/index.php?title=Integration>

Understand BizAgi's architecture: <http://wiki.bizagi.com/en/index.php?title=Architecture>

Free courses of BPMN:

<http://elearning.bizagi.com/?counterId=3c59dc048e8850243be8079a5c74d079>