
PLAN DE COURS

8INF956
Développement avancé de logiciels :
Patrons et Modèles

Département d'informatique et de mathématique



Objectifs généraux du cours

- Comprendre et maîtriser les fondements de formation concernant la communication, la dépendance et l'environnement des modules (composantes, classes, aspects, ou autres).
- 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.
- Permettre une souplesse dans l'architecture et la configuration des applications en utilisant les techniques des patrons (et IOC) et modèles MDA et assurer la qualité de logiciels, en particulier: facilité le changement des comportements dynamiques d'une application, favoriser la réutilisation de composantes logicielles, faciliter les tâches de développement des applications d'entreprise, et faciliter la maintenance.
- Offrir une occasion de *critiquer* des articles existants et de *résoudre* des problèmes identifiés et relatifs au développement de logiciels: application et réutilisation de patrons et modèles pour résoudre des problèmes récurrents dans les divers environnements.
- Encourager les étudiants à détecter des défis au développement de logiciels, à préparer et à présenter des exposés pour montrer ces défis. Les étudiants sont invités à présenter les approches existantes pour résoudre tels défis et à proposer des solutions.
- Offrir une vision globale liée à plusieurs domaines de recherche en informatique liés à l'application de patrons et modèles : invitation des chercheurs ou collègues pour présenter leurs recherches.

Description du cours (objectifs spécifiques)

Ce cours se concentre sur les problèmes récurrents, les patrons et le modèle trouvés dans les diverses phases de développement de logiciels et en particulier la réutilisation de composants logiciels (patrons, modèles, bibliothèques, plateformes). Nous étudions d'abord les concepts avancés pour le développement d'applications d'"entreprise", surtout la qualité de logiciel, les problèmes de couplage (IOC), la réutilisation et la maintenance. Par la suite, nous allons aborder les différents types de patrons (implémentation, conception, architecture et analyse). Les techniques de séparation de préoccupations (par exemple, aspect-oriented computing AOP, Spring framework, OSGi framework) seront utilisées pour mieux implémenter ces patrons. Ces techniques pourraient supporter le couplage faible et améliorer la réutilisation. Les modèles MVC et publish-subscribe seront étudiés comme patrons architecturaux permettant de séparer les composantes logicielles en différentes parties. Le développement orienté modèle et son architecture seront aussi étudiés pour faciliter la transformation d'un modèle à un autre.

À la fin du cours, l'étudiant devrait pouvoir

- collaborer et communiquer avec les membres d'une équipe de développement de logiciel dans la réalisation d'un projet complexe, en se basant sur les patrons et modèles.
- améliorer la qualité de logiciels selon les patrons existants et le développement orienté modèle MDA
- être en mesure de mener simultanément une démarche qui combine créativité dans la conception de logiciel, rigueur dans la modélisation et dans l'implémentation du logiciel et dans le suivi d'un projet, et pragmatisme dans l'atteinte des objectifs d'un projet en tenant compte de contraintes temporelles et budgétaires.



Insertion du cours dans le programme

Le cours est optionnel dans les divers programmes des cycles supérieurs du département d'informatique et de mathématique (DIM) de l'université. Ce cours nécessite la connaissance des concepts de base d'un développement de logiciel (méthodes de développement, cycle de vie, UML, ...), divers diagrammes (cas d'utilisation, de séquence, ...), programmation objet (Java, C/C++, C#), composant (concept, interaction), AspectJ, Spring, Eclipse, etc. De plus, il demande une capacité d'analyse rigoureuse et méthodique.

Situation du cours dans le programme

Le cours 8INF956- Développement avancé de logiciels est un cours optionnel offert à tous les cycles supérieurs dans le département d'informatique et de mathématiques. Aucun cours n'est préalable au cours 8I

Références

1. Notes de cours 8INF956 et liste de sites intéressants (voir ma page web)
2. Steven John Metsker, and William C. Wake. Design Patterns in Java, Addison-Wesley, 2006.
3. Cay Horstmann. Object-Oriented Design and Patterns, Wiley, 2006.
4. Xavier Blanc. MDA en Action. Eyrolles 2005.
5. Markus Volter, Micheal Kricher and Uwe Zdun. Remoting Pattern: Foundations of Enterprise, Internet and Realtime, Distributed Object Middleware, John Wiley & Sons, England, 2005.
6. Micheal Kricher and Prashant Jain. Pattern-Oriented Software Architecture: Pattern for Resource Management. Volume 3. John Wiley & Sons, England, 2004.
7. Len Bass, Paul Clements, and Rick Kazman. Software Architecture in Practice, Addison-Wesley, 2003.
8. Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy. Software Architecture: Foundations, Theory and Practice. Wiley, 2009.
9. Martin Fowler. Patterns of Enterprise Application Architecture. Addison-Wesley Professional. 2004.
10. Martin Fowler. Analysis Patterns: Reusable Object Models. Addison-Wesley Longman, 1997.
11. Mary Shaw and David Garlan. Software Architecture. Prentice-Hall, 1996.
12. Jacques Printz. Architecture Logicielle : concevoir des applications simples, sûres, et adaptables. Dunod. 2009.
13. Gregor Hohpe and Bobby Woolf. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley. ISBN: 0321200683, 2004.
14. Eric Gamma, Richard Helm, and Ralph Johnson. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional; 1er edition, 1995.
15. Martin fowler: <http://www.martinfowler.com/articles/writingPatterns.html>