

LOG8430: Architecture logicielle et conception avancée

Hiver 2019

©M. Fokaefs et F. Guibault

Enseignant

- François Guibault – Professeur
- <https://www.polymtl.ca/expertises/guibault-francois>
- francois.guibault@polymtl.ca
- Local M-3424
- Recherche
 - Modélisation géométrique, génération et adaptation de maillage
 - Méthodes de simulation
 - Calcul haute-performance et traitement parallèle

Enseignant – François Guibault

- Diplôme en génie physique – Polytechnique
- Maîtrise en mathématiques appliquées – Polytechnique
 - Système de descriptions des éléments à ordre élevé dans un logiciel de simulation par éléments finis
- Doctorat en génie mécanique – Polytechnique
 - Génération de maillages hybrides tridimensionnels
- En GIGL depuis septembre 1999

Chargés de laboratoire

- Mme Mouna Abidi
 - mouna.abidi@polymtl.ca
 - Étudiante de Doctorat à Poly
- M. Théo Moffelein
 - theo.moffelein@polymtl.ca
 - Étudiant de maîtrise à Poly

Objectifs du cours

- Présenter les styles et les patrons d'architecture et de conception logicielle en mettant l'emphasis sur les systèmes distribués.
- Mesurer et maintenir la qualité de la conception et de l'architecture.
- Explorer et pratiquer la conception de logiciels distribués dans des cas réels : cadrage, systèmes infonuagiques et systèmes d'analyse de données.



À la fin du cours, l'étudiant sera en mesure de

- Concevoir l'architecture d'un logiciel en choisissant et en justifiant les styles d'architecture et les patrons de conception selon les exigences.
- Concevoir les composantes de l'architecture en utilisant des modèles spéciaux et des technologies nouvelles (architectures évènementielles, orientées services, orientées objets, infonuagiques).
- Évaluer la qualité d'une architecture ou d'une conception logicielle, et la maintenir en appliquant des changements.

Situation du cours

- LOG8430 termine une chaîne de cours :
 - INF1005C – programmation procédurale
 - INF1010 – programmation orientée objets
 - LOG2410 – conception logicielle
 - LOG3210 - Éléments de langages et compilateurs
 - LOG3430 - Méthodes de test et de validation du logiciel

Situation du cours

- LOG8430 est aussi un cours gradué
- Être à l'aise pour lire (en anglais) et écrire.
- Faire de la recherche
 - Il n'existe pas toujours une seule réponse.
 - Justifiez et vérifiez vos réponses.
 - Pensée critique.
 - Collaboration!
- Bien travaillez en équipe.
- Concepts avancés sur
 - OO
 - SOA
 - Infonuagique
 - Systèmes distribués

Format du cours

- 2 heures de cours magistraux – 1 heure de pratique en classe
 - Exercices sur le cours du jour.
 - En équipe.
 - Apportez vos ordinateurs portables!
- 3 heures de labo par 2 semaines
 - Outils
 - Exemples de systèmes
- 1 atelier pratique

3 Questions sur l'architecture logicielle

1. Qu'est-ce qu'une architecture logicielle ?
2. Quels sont les critères qui doivent guider un architecte logiciel dans le choix d'un style architectural particulier ?
3. Quel impact a le choix d'un style architectural sur le développement d'un système logiciel ?

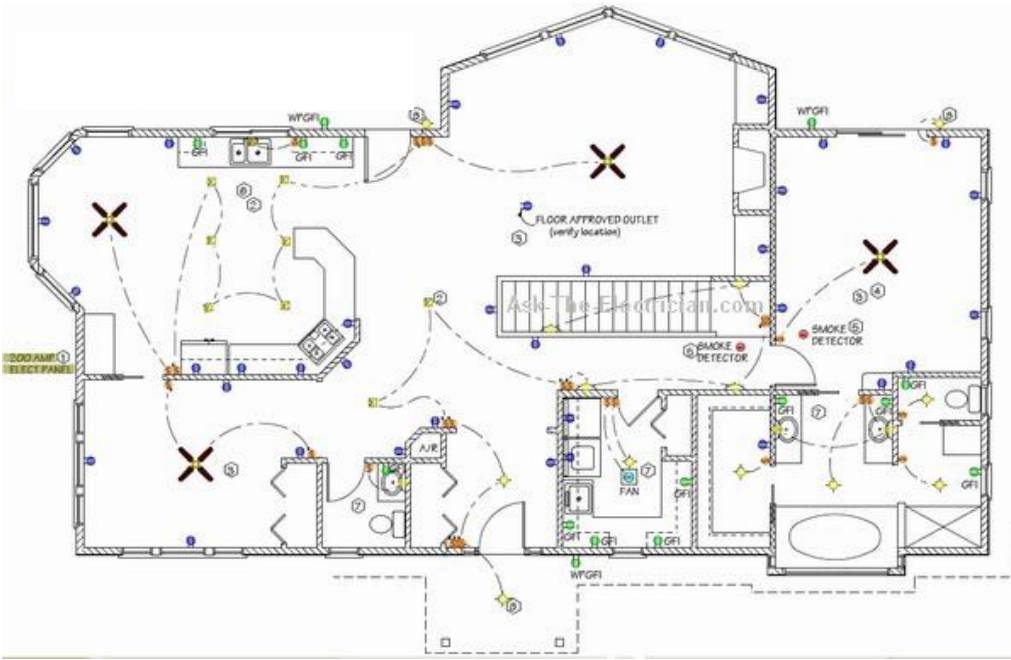
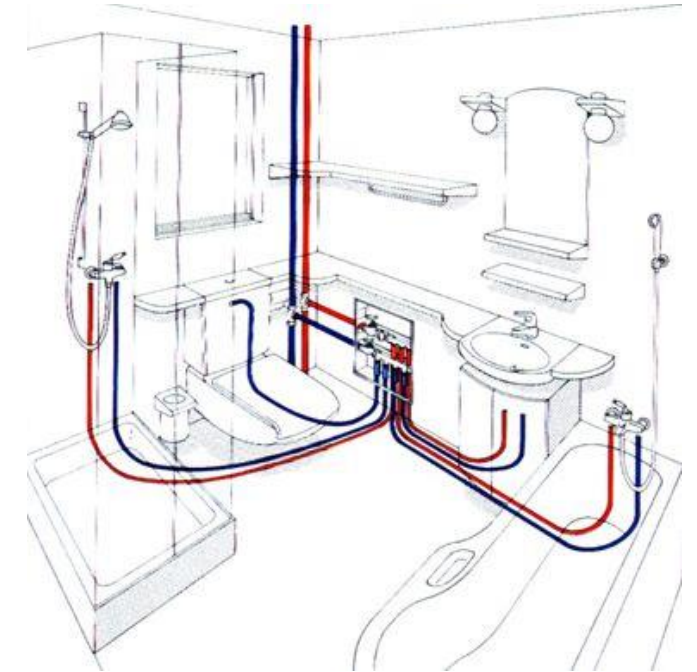
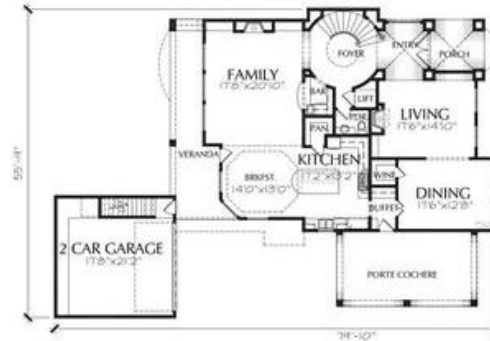
3 Questions sur la conception logicielle

1. Quelle est la différence entre l'architecture et la conception logicielle ?
2. Les décisions architecturales et de conception sont-elles indépendantes ?
3. Comment peut-on évaluer la qualité d'une architecture et la qualité d'une conception ?

L'architecture et la conception dans le domaine immobilier



L'architecture et la conception dans le domaine immobilier



3 Questions sur l'architecture logicielle

1. Qu'est-ce qu'une architecture logicielle ?
2. Quels sont les critères qui doivent guider un architecte logiciel dans le choix d'un style architectural particulier ?
3. Quel impact a le choix d'un style architectural sur le développement d'un système logiciel ?

3 Questions sur la conception logicielle

1. Quelle est la différence entre l'architecture et la conception logicielle ?
2. Les décisions architecturales et de conception sont-elles indépendantes ?
3. Comment peut-on évaluer la qualité d'une architecture et la qualité d'une conception ?

Programme du cours

| Cours | Contenu | Date |
|-------|--|------------|
| 01 | Introduction Concepts de base Principes SOLID Patrons de conception | 8 janvier |
| 02 | Architectures distribuées Patrons et styles d'architecture | 15 janvier |
| 03 | Qualité de la conception Attributs de qualité Métriques de qualité | 22 janvier |
| 04 | Mauvaise Conception Antipatrons / design smells Refactoring, refactoring to patterns | 29 janvier |

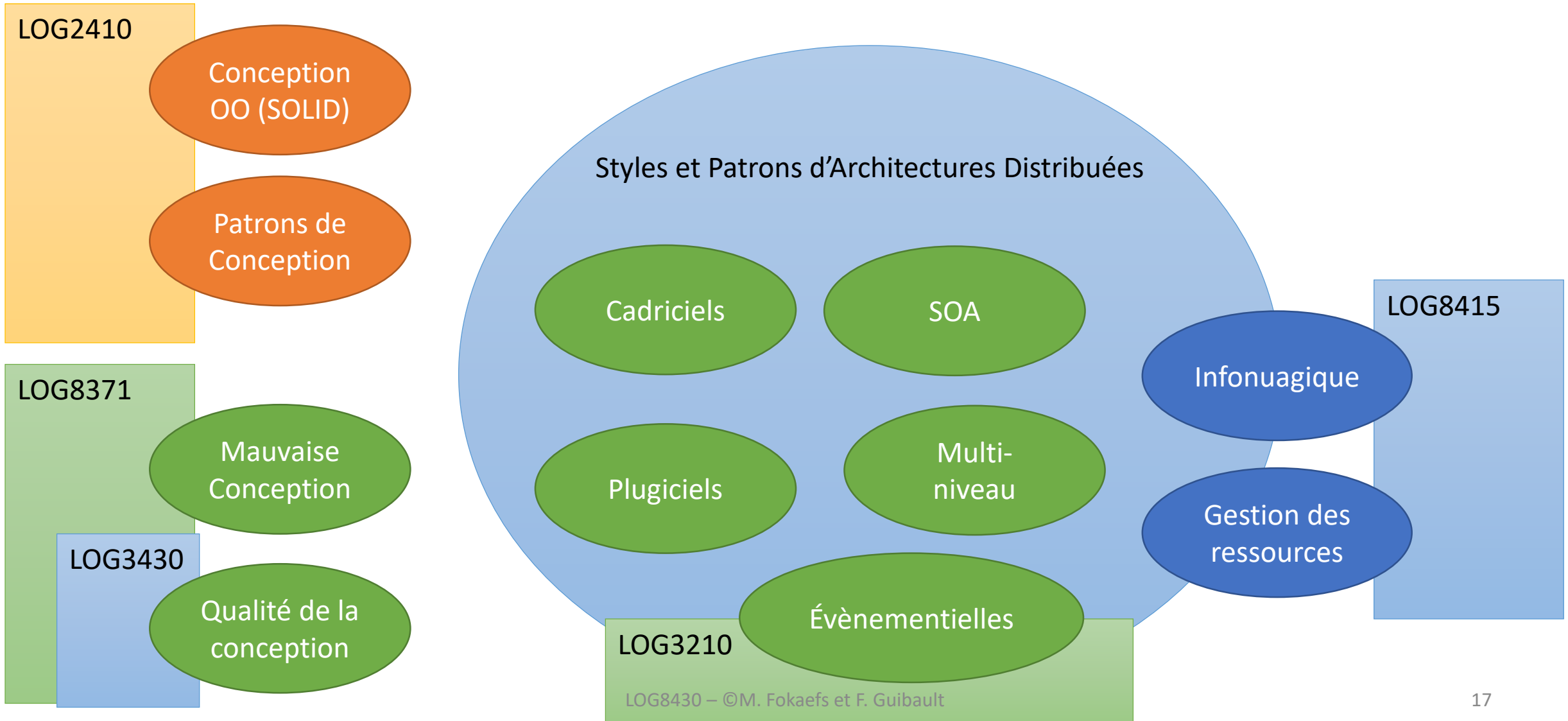
Programme du cours

| Cours | Contenu | Date |
|-------|--|------------|
| 05 | Architectures modulaires de composantes Cadriciels How to break an API? | 5 février |
| 06 | Architectures évènementielles Traitement d'évènements Synchronisation et concurrence | 12 février |
| 07 | Architecture logicielle orientée patrons | 19 février |
| 08 | Atelier pratique: Migration Refactoring d'architecture | 26 février |
| 09 | Développer un logiciel pour un système modulaire/distribué Plugiciels | 12 mars |

Programme du cours

| Cours | Contenu | Date |
|-------|--|---------|
| 10 | <p>Systemes distribués sur le web</p> <p>SOA</p> <p>Microservices</p> | 19 mars |
| 11 | <p>Architectures multi-niveaux</p> <p>Systemes de systemes</p> <p>Systemes distribués de mégadonnées</p> | 26 mars |
| 12 | <p>Déploiement dans le nuage</p> <p>Le rôle du réseau</p> <p>Le rôle du logiciel</p> | 2 avril |
| 13 | <p>Concepts avancés et émergents</p> <p>Performance et utilisation des ressources</p> <p>Autoscaling</p> | 9 avril |

Carte du cours



Ressources

- Pas de manuel principal, mais plusieurs ressources pour étudier et approfondir la matière :
 - Liens
 - Tutoriels
 - Manuels
 - Articles
 - Logiciels
- Disponibles sur Moodle pour chaque séance.

Outils

- Eclipse - <https://www.eclipse.org/>
- Git
 - GitHub - <https://github.com/>
 - Bitbucket - <https://bitbucket.org/>
- Conception et modélisation
 - Papyrus - <https://www.eclipse.org/papyrus/>
 - ArgoUML - <http://argouml.tigris.org/>
 - Enterprise Architect - <https://sparxsystems.com/products/ea/>
- Analyse de code source
 - JDeodorant - <https://marketplace.eclipse.org/content/jdeodorant>
 - Ptidej - <http://www.ptidej.net/>
- Architectures multi-niveaux
 - Tomcat - <http://tomcat.apache.org/>
 - Spark - <https://spark.apache.org/>
 - Cassandra - <http://cassandra.apache.org/>
 - Docker - <https://www.docker.com/>
 - VirtualBox - <https://www.virtualbox.org/>

Important

- Toute la matière présentée pendant les séances de cours ou de labo peut faire l'objet d'une évaluation.
- Le contenu du cours est dynamique!
- La matière sera disponible juste quelques jours avant.
- Des annonces précéderont les mises à jour du matériel et du cours.
- Les travaux remis en retard seront pénalisés de 10% par jour de retard.
- Toute demande d'examen différé doit passer par le bureau des affaires académiques.
- La collaboration est permise pour les TP, mais les règlements relatifs au plagiat s'appliquent en tout temps.

Évaluation du cours

- Travaux Pratiques (Microprojets 1 et 2) 60%
- Examen Final 40%
- Au moins 50% à l'examen final pour passer le cours.

Travaux Pratiques

- En équipe (3-4 max)
 - Déclarez vos équipes par courriel avant le 18 janvier.
- Objectifs
 - Maitriser la théorie
 - Compréhension et analyse
 - Conception et décision
- 2 microprojets – 3+1 rapports

Examen final

- Individuel
- 6 questions
 - Chaque question vaut 20 points.
 - Maximum 100 points.
- Théorie et pratique
 - Comme les exercices en classe (mais sur papier).
- Exemples de systèmes discutés en classe, au labo ou dans les TPs.

Exercices en classe

- Apportez vos ordinateurs portables!
- Apprentissage par la mise en pratique. Il n'y aura pas de note pour ces exercices.
- MAIS! Des exemples de ces exercices pourraient être utilisés à l'examen.
- On va se concentrer sur des systèmes simples, mais populaires et avec amplement de documentation
 - JHotDraw, MiniDraw (pendant le Microprojet 1)
 - Eclipse
 - Spark, Docker (pendant le Microprojet 2)

3 Questions sur l'architecture logicielle

1. Qu'est-ce qu'une architecture logicielle ?
2. Quels sont les critères qui doivent guider un architecte logiciel dans le choix d'un style architectural particulier ?
3. Quel impact a le choix d'un style architectural sur le développement d'un système logiciel ?

3 Questions sur la conception logicielle

1. Quelle est la différence entre l'architecture et la conception logicielle ?
2. Les décisions architecturales et de conception sont-elles indépendantes ?
3. Comment peut-on évaluer la qualité d'une architecture et la qualité d'une conception ?

Qu'est-ce que l'architecture?
Qu'est-ce que la conception?



Architecture logicielle (ANSI/IEEE Std 1471-2000)

L'architecture d'un système logiciel est définie comme l'**organisation** fondamentale du système, qui s'incarne dans ses **composantes**, leurs **relations** entre elles et avec leur **environnement**, et les principes qui guident sa **conception** et son **évolution**.

Architecture logicielle (définition SEI)

L'architecture d'un système logiciel est l'ensemble des **structures** nécessaires pour **raisonner** à propos du système, qui comprennent les **éléments** logiciels, les **relations** entre eux et les **propriétés** des chaque.



Architecture logicielle (définition Sangwan, 2015)

L'architecture d'un système logiciel se préoccupe fondamentalement de l'organisation d'un système en ses constituants et leurs interrelations afin de réaliser un objectif donné.

Architecture

- Elle répond à « quoi? ».
- Au niveau modèle
 - Paquetages, composantes et relations
- Niveau de décision plus abstrait.
 - Les décisions qui doivent être prises à l'avance.
 - Les décisions qui sont très coûteuses à changer pendant le développement.
- Pertinente pour les objectifs opérationnels et les exigences non-fonctionnelles.
- Elle inclut:
 - Modules
 - Bases/technologies de données
 - Environnement d'exécution et de déploiement

Conception

- Elle répond à « comment? ».
 - Comment on va implémenter l'architecture.
- Éléments plus concrets:
 - Classes
 - Méthodes/Fonctions
 - Tables de données
- Pertinent pour les exigences fonctionnelles.
- Éléments qui peuvent être changés pendant le développement (la phase de maintenance)
 - Réingénierie (*refactoring*)
 - Évolution
 - Adaptation

Architecture ou Conception?



« On veut une couche de GUI, une couche d'analyse et une couche de stockage des données. »

Architecture

Conception

« On veut une couche de GUI, une couche d'analyse et une couche de stockage des données. »

Architecture

Conception

« Toutes les nouvelles applications doivent étendre l'interface *Application*. »

Architecture

Conception

« Toutes les nouvelles applications doivent étendre l'interface *Application*. »

Architecture

Conception



« L'application sera disponible
comme un service déployé en
infonuagique. »

Architecture

Conception

« L'application sera disponible
comme un service déployé en
infonuagique. »

Architecture

Conception



« On a besoin d'une base de données NoSQL avec un taux de disponibilité élevé. »

Architecture

Conception

« On a besoin d'une base de données NoSQL avec un taux de disponibilité élevé. »

Architecture

Conception

« Une méthode prend le type d'un objet comme paramètre et retourne une instance de ce type en appelant le constructeur privé de la classe correspondante. »

Architecture

Conception

« Une méthode prend le type d'un objet comme paramètre et retourne une instance de ce type en appelant le constructeur privé de la classe correspondante. »

Architecture

Conception