

This is the html version of the file <https://moodle.polymtl.ca/mod/resource/view.php?id=236382>.
Google automatically generates html versions of documents as we crawl the web.

LOG8430: Architecture logicielle et conception avancée

Présentation de Cours

Automne 2017

Fabio Petrillo

Chargé de Cours

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

Page 2

Introduction

- Chargé de course et chargé de lab.
- Objectifs de l'informatique
- Objectifs du cours
- Programme de cours
- Évaluation
- Sondage

2

Page 3

Chargé de cours et labo

Fabio Petrillo

<http://fabiopetrillo.com>

fabio@petrillo.com

fabio.petrillo@polymtl.com

[@petrillofabio](#)

Local M-4221

3

Page 4

Chargé de cours - Fabio Petrillo

- Diplôme d'ingénieur en génie électrique
- Maîtrise en génie logiciel
 - Méthode agile pour l'industrie du jeux vidéo
- **Doctorat en génie logiciel**
 - Analyse, compréhension et visualisations des programmes

- Débogage - Swarm Debugging
- Plus de 20 ans d'expérience en génie logiciel
 - Architecte logiciel au Brésil (Government)
 - Gestionnaire
 - Expérience en systèmes complexes
- Chargé de Course et Lab à Poly (LOG8430, INF3710, et LOG8371)
- Research Assistant (Postdoc) at Concordia University (Prof. Yann-Gaël)

Page 5

Chargé de labo

Manel Abdellatif

manel-2.abdellatif@polymtl.ca

Étudiante de Doctorat à Poly

Expert en migration vers service

Page 6

Objectifs

- Présenter les principaux types d'architectures logicielles et de critères de choix d'un type d'architecture
- Présenter des approches basées sur des patrons de conception pour la réalisation d'applications (distribuées)
- Donner un aperçu des approches émergentes de conception logicielle

Page 7

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

- **Concevoir** l'architecture d'un logiciel en **choisissant** et en **justifiant** les architectures, les patrons de conception et les techniques de conception avancée utilisés (entre autres à la méta-programmation, la programmation par composants, les patrons d'architecture distribuée, les architectures orientées services) ;
- Concevoir les composantes de l'architecture d'un logiciel (classes orientées-objets, composants logiciels ou services logiciels) à l'aide de techniques avancées et à partir de patrons d'architecture distribuée parmi les plus utilisés et de justifier ses choix ;
- Juger de la conception de l'architecture d'un logiciel et de suggérer des changements à cette architecture.

7

Situation du cours

LOG8430 termine une **chaîne** de cours :

- INF1005C - programmation **procédurale**
- INF1010 - programmation **orientée objets**

- LOG2410 - **conception** logicielle

8

Page 9

Situation du cours

- LOG8430 est aussi un cours **gradué**
- Être à l'aise pour lire et écrire (en l'anglais)
- Faire de la recherche
 - Pensée critique
 - Créer des connaissances
- Concepts avancés
 - Être à l'aise avec les concepts OO
 - Être à l'aise avec des langage de programmation
 - Être à l'aise avec installation des outils (se **débrouiller**)
 - **Être à l'aise pour prendre des décisions et les soutenir**

Page 10

Programme de cours

Cours	Contenus
-------	----------

00	Plan de cours
----	---------------

01 - 04	Fondements de architecture logicielle
---------	---------------------------------------

	Modélisation, concepts de la POO
--	----------------------------------

	Patrons des projets et les styles architecturaux
--	--

	Analyse et évaluation de la qualité logicielle
--	--

	Récupération d'architecture logicielle (reverse engineering)
--	--

	Documentation d'architecture logicielle
--	---

05 - 08	Conception d'architecture logicielle
---------	--------------------------------------

	Process, gestion et évolution et d'architecture logicielle
--	--

	Méta-programmation
--	--------------------

	Cadriciels (Frameworks) et platforms
--	--------------------------------------

09 - 10	Systèmes distribués; Architecture orientée services (SOA), Microservices,
---------	---

	Serverless, Platform as a service (PaaS), Cloud Computing Architecture
--	--

	Architectures émergentes: Machine Learning, Blockchain, ...
--	---

11 - 12	Présentations des Papiers et Révision
---------	---------------------------------------

10

Page 11

Principaux manuel du cours essentielles

- *Software systems architecture: working with stakeholders using viewpoints and perspectives* , 2nd ed.; Rozanski, Nick. Addison-Wesley, 2011.
- *Designing Software Architectures: A Practical Approach*, Humberto Cervantes; Rick Kazman, Addison-Wesley Professional, 2016.
- *Software and Systems Architecture in Action*, Raghvinder S. Sangwan; Auerbach Publications, 2014.
- *Documenting software architectures: views and beyond*, Paul Clements, 2nd; Addison-Wesley, 2011.
- *Patterns of Enterprise Application Architecture*, Martin Fowler; Addison Wesley Professional, 2002.

11

Page 12

Outils

- Moodle
- Slack - <https://log8430.slack.com>
- Trello - <https://trello.com>

- Overleaf - <https://www.overleaf.com>
- Mendeley - <https://www.mendeley.com/>
- Git - <https://github.com/>
- CASE/Modelling Tools - Modelio, Papyrus, ArgoUML, Visual Paradigm, Enterprise Architect, Archi Tool,
- Static analysis tools: Ptidj, Understand, Better Code Hub, SonarQube, CodeScene,

12

Page 13

Important

- Toutes les matières présentées pendant les séances de cours ou de lab. sont contrôlable
- 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
- Les travaux pratiques se font par équipes de **3 étudiantEs (4 max)**. Chaque équipe doit soumettre le travaux sur Moodle.

Page 14

Important

Les règlements relatifs au plagiat s'appliquent en tout temps

Page 15

Évaluation du cours

- Travaux Pratiques 30%
- Papier de Recherche 30%
- Examen Finale 40%

15

Page 16

Travaux Pratiques

- En équipe (3 - 4 max)
- Objectifs
 - Compréhension et analysis

- Conception
- Implementation
- 3 rapports (TP1 - 3) et une présentation finale (TP4)
- **Pondération 30%**

Organisation des travaux pratiques

TP	Sujet	Pondération	Remise
TP1	Étude et analyse d'architecture	30%	15/10/2017
TP2	Conception and implementation	30%	12/11/2017
TP3	Migration - Arch. orienté à services	30%	03/12/2017
TP4	Présentation finale	10%	04/12/2017

Papier de recherche

- Papier de recherche sur un sujet de architecture logicielle
- **Individuelle ou en groupe**
- Nombre des pages: nombre d'auteurs x **5 pages**
 - Individuelle - 5 pages (short paper)
 - 2 auteurs - 10 pages (full conference paper)
 - 3 auteurs - 15 pager (journal paper)
 - 4 auteurs - 20 pager (top journal paper)
- Format finale **Latex/Doc IEEEtrans**
- Langue: **anglais** (préférable) ou français
- Outils d'édition: Overleaf (<http://www.overleaf.com>)
- **Soumission:** <http://icsa-conferences.org/2018/> (pas obligatoire)
- **Pondération 30%**

Structure de base pour le papier

- Abstract: une synthèse de votre papier pour guider le lectures/attirer pour lire le papier
- Introduction: problème ciblé, motivation, proposition, organization du papier
 - avec une (ou des) question(s) de recherche
- Background: les concepts/contexte nécessaire pour comprendre votre papier (references)
- Travaux connexes
- Approche/méthodologie: votre approche pour le problème ciblé
- Résultats: les résultats par rapport l'application de votre approche
- Discussion
- Threats to the validity
- Conclusion: les conclusions et travaux futures

19

Papier - Points évalués

- Importance de la **contribution** du papier et avantages que d'autres peuvent tirer de la contribution: pourquoi la contribution et les avantages sont-ils

importants?

- **Originalité** du travail: quelles nouvelles idées, discussions ou approches sont introduites?
- **Validité** du travail présenté: comment les chercheurs et les praticiens peuvent-ils utiliser les résultats avec confiance?
- **Clarté** de présentation;
- **Travaux antérieurs** pertinents: le travail préalable a-t-il fait l'objet d'un examen adéquat?

20

Page 21

Idées pour la recherche

- Révision de la littérature
- Surveys
 - SLR
 - Architects
 - Artefacts (Github, StackOverFlow, software projects)
- Évaluations
 - Outils, practices, styles, patterns, techniques, ...
- Nouveaux étuds

- The state of practice on software architecture modeling and representation
- The state of practice on software architecture tools: an Open Source perspective
- Architectural styles on Open Source Softwares: an exploratory study
- Who is the software architect?

Page 22

Remises pour le papier de reserche

Remise	Livrable	Pondération	Date de Remise
R1	Définition du sujet	5%	10/09/2017
R2	Introduction, Background, Travaux connexes et méthodologie	20%	24/09/2017
R3	Results, Discussion, Threats to Validity, Abstract et Conclusion (brouillon complet)	25%	22/10/2017
RF	Version finale Présentation	50%	19/11/2017 22 e 29/11/2017

Examen Final

- Individuel
- Résumé fait **à la main** d'une feuille A4 (max) recto-verso
- 4~5 questions obligatoire et de questions optionnelle
- **Pondération 40%**
- Élaboration des questions par les étudiantEs
 - a chaque cours peut demander aux présentes à proposer de questions avec la réponse
 - sujet de la séance ou les prochaines (étudiant en avance!!!)
 - je vais utiliser les bonnes questions dans l'examen
 - **l'auteur recevra les points sans faire la question!**

23

Activité et sondage

1) **Activité:** sur papier, en utilisant votre connaissance actuelle, proposer un

modèle architectural pour un “clone” du **Twitter** (le plus complet possible

- utilisez votre créativité!).

2) Après, svp, réponse le **sondage** suivant:

- a) Origine - Quel «génie» (Polytechnique) / École d'origine ?
- b) Bac, DESS, Master ou Doctorat?
- c) Expérience professionnelle?
- d) Intérêt à suivre le cours - Pourquoi je suis là?
- e) Quelles sont vos attentes à l'égard du cours?