

Département d'informatique

IGL 301 – Spécification et vérification des exigences

Plan de cours Hiver 2016

Enseignant	Luc Lavoie			
	Courriel :	luc.lavoie@usherbrooke.ca		
	Local :	D4-2006		
	Téléphone :	(819) 821-8000 poste 62015		
	Site :	http://info.usherbrooke.ca/llavoie		
	Disponibilité :	sur rendez-vous		
Horaire	Groupe 01	lundi	08:30 à 10:20	D3-2037
		jeudi	08:30 à 10:20	D3-2037 (D4-1017 lors des TD)

Description officielle de l'activité pédagogique¹

Objectifs : Spécifier, valider et vérifier les exigences des clients; en déduire une architecture technologique.

Contenu : Spécifications fonctionnelles et non fonctionnelles. Diagramme de flux de données et modèles de données. Spécification textuelle des exigences. Cas d'utilisation et scénario. Validation des exigences. Génération de scénarios de tests d'acceptation. Élaboration de l'architecture. Présentation des normes de spécification IEEE.

Crédits : 3

Organisation : 3 heures d'exposé magistral par semaine
1 heure d'exercices par semaine
5 heures de travail personnel par semaine

Concomitante : IFT 232

Particularités : Aucune

Professeurs responsables

Marc Frappier et Luc Lavoie

¹ <http://www.usherbrooke.ca/fiches-cours/igl301>

1 Présentation

Cette section présente les objectifs spécifiques et le contenu détaillé de l'activité pédagogique.

1.1 Mise en contexte

Le génie logiciel traite de la configuration d'une machine universelle (ordinateur) dans le but d'atteindre un objectif spécifique. Le logiciel de configuration peut lui-même être vu comme une machine, mais il diffère d'autres machines en ce sens qu'il est intangible. Le génie logiciel (GL) doit son nom et sa constitution comme un domaine de connaissance propre à la tenue d'un séminaire organisé par l'OTAN à Garmisch-Partenkirchen en Allemagne en 1968.

Le logiciel de configuration d'une machine universelle est désigné sous plusieurs appellations différentes, selon la caractéristique mise de l'avant : logiciel (intangibilité), programme (déterminisme), système (complexité).

Puisqu'on construit généralement un système pour atteindre un objectif donné, il est préférable de déterminer et de détailler d'abord cet objectif et le but auquel l'objectif concourt. Ce qui nous amène à l'ingénierie des exigences (IE), la partie du génie logiciel qui permet de déterminer quel système doit être développé. Plusieurs études tendent à montrer que l'échec d'un projet de développement de système est souvent imputable à des lacunes d'IE. Par ailleurs, une erreur d'IE a généralement plus d'impact qu'une erreur commise à une autre étape du développement du système et ce, que l'impact soit mesuré en terme de coût, de durée, de portée ou de qualité.

La maîtrise du processus de développement logiciel passe par celle de l'IE et intéresse toutes les organisations tributaires de systèmes comprenant des logiciels.

Note : L'expression « spécification des exigences » est parfois utilisée pour désigner l'ingénierie des exigences dans son ensemble; nous préférons réserver cette expression pour désigner une activité précise au sein du processus d'ingénierie des exigences, à savoir l'activité par laquelle on déduit (ou conçoit) et met en forme, selon des critères rigoureux, les exigences découlant des activités préalables d'exploration et d'analyse.

1.2 Objectifs spécifiques

La réussite de l'activité pédagogique est déterminée par la capacité :

1. de décrire, d'adapter et d'appliquer différents procédés d'ingénierie des exigences;
2. d'appliquer différentes méthodes et techniques d'exploration et d'analyse des besoins;
3. d'appliquer différentes méthodes et techniques de modélisation de problèmes et de solutions;
4. d'appliquer différentes méthodes et techniques de spécification, de vérification, de validation des exigences;
5. de produire le cahier des charges d'un système logiciel de complexité moyenne.

1.3 Contenu détaillé

Tableau 1 – Contenu détaillé de l'activité

N°	Contenu	Heures ²	Objectifs ³	Travaux
1.	Introduction 1.1 – Place de l'expertise propre au sein de l'organisation 1.2 – Place du l'IE au sein du GL 1.3 – Artéfacts de l'IE 1.4 – Un premier procédé d'IE simple et typique	4	1	
2.	Exploration 2.1 – Techniques (a) enquêtes : profils d'acteurs, entrevue, sondage, etc.; (b) ateliers : remue-méninges, groupes de concertation, Delphi, JAD, etc.; (c) recherche et gestion documentaire : glossaire, wiki, CMS, etc.; (d) observation de tâches; (e) maquettage; (f) prototypage. 2.2 – Planification et exécution (a) contraintes et cibles de couverture; (b) pertinence et choix des techniques.	8	1, 2	TP1
3.	Analyse structurée (pilotée par les processus) (a) contexte, environnement et agents (DC) (b) processus et activités (DFD, TD, AEF, MEF, pseudo-code) (c) information et données (MCD, DD)	8	1, 2, 3	TP2
4.	Spécification (a) spécification textuelle; (b) spécification semi-formelle; (c) diagrammes UML-A, UML-C, UML-S, UML-U, etc.	4	1, 3	TP3
5.	Vérification et validation (a) principes; (b) grilles de couverture; (c) techniques de revue.	4	1, 4	TP4-TP6
6.	Sélection et traçabilité (a) principes; (b) grilles d'évaluation critériée; (c) techniques de suivi.	4	1	TP4-TP6
7.	Analyse pilotée par le problème (Jackson Frames) (a) modèle de Jackson; (b) éléments : atelier, contrôle, commande, conversion, information; (c) règles de composition et de décomposition.	8	2, 3	TP4-TP6
8.	Analyse orientée utilisateur (pilotée par les interactions) (a) modèle d'interaction; (b) cas d'utilisation et scénario; (c) règles de composition et de décomposition.	8	2, 3	TP4-TP6
9.	Analyse orientée objet (pilotée par le contexte) (a) modèle contextuel; (b) stéréotypes et patrons; (c) règles de composition et de décomposition.	8	2, 3	TP4-TP6
10.	Procédés d'IE (a) activités fondamentales: exploration, analyse, sélection, spécification, vérification et validation (b) procédés : V, QT, JAD, CSEM, Bray, etc. (c) grille comparative	4	1, 5	TP4-TP6
		60		

Si l'activité est limitée à 52 heures (cours et travaux dirigés), un seul des thèmes 8 ou 9 est traité, au choix de l'enseignant.

² Répartition des heures combinées des cours, des exercices et des travaux dirigés.

³ L'objectif spécifique 5 est atteint à l'aide des exercices, des travaux dirigés et des travaux pratiques.

2 Organisation

Cette section présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux.

2.1 Méthode pédagogique

Les périodes de cours hebdomadaires serviront aux exposés théoriques et aux exemples. Les travaux dirigés proposent des exercices individuels ou en groupe selon le cas. Des lectures complémentaires seront requises tout au long de l'activité; les références seront communiquées au fil des semaines.

2.2 Calendrier

Tableau 2 – Planification des activités et des lectures

S ^e	Semaine	Activités	Contenu	Bray	Hull	Autres	Échéance des travaux
1.	2016-01-04	cours	1	1, 2	1, 2		
2.	2016-01-11	cours	2	3	5	IAAT	
3.	2016-01-18	cours + TD	2	9, 11		WOL	TP1
4.	2016-01-25	cours	3	4.3	3		
5.	2016-02-01	cours + TD	3	12, 13, 14		Sommerville	TP2
6.	2016-02-08	cours	4	5	4	GLOGUS	
7.	2016-02-15	cours + TD	5	6	7	GLOGUS	TP3
8.	2016-02-22	examen	--				
9.	2016-02-29	relâche universitaire					
10.	2016-03-07	cours	6			Leffingwell	
11.	2016-03-14	cours	7	4.5			
12.	2016-03-21	cours + TD	7			Jackson	TP4
13.	2016-03-28	cours	8	4.4			
14.	2016-04-04	cours + TD	8			Larman	TP5
15.	2016-04-11	cours	10			De Lucia	
16.	2016-04-18	cours	10			Wood	TP6
17.	2016-04-xx	examen	--				

Le premier cours est programmé le jeudi 7 janvier et le dernier le lundi 18 avril. Il n'y a pas de cours le lundi 28 mars. Les dates de travaux sont sujettes à changement en fonction du rythme du cours. Les dates d'examen seront fixées ultérieurement par la Faculté des sciences. La date de remise des travaux est le dernier de la semaine (par exemple, le TP1 est à remettre au plus tard le dimanche 24 janvier à 23:59).

2.3 Évaluation

En plus des deux examens individuels, l'évaluation porte sur :

- trois travaux pratiques (TP1 à TP3) réalisés en tandem (deux personnes);
- un projet (TP4 à TP6) réalisé en équipe de trois à quatre personnes.

Les examens sont placés sous la responsabilité de la Faculté des sciences et organisés par elle. La durée des examens est de trois heures. La documentation personnelle est limitée à une feuille recto verso de format lettre américaine (216 × 279 mm); l'usage d'appareils informatiques, électroniques ou de communication (ordinateur, calculatrice, téléphone, etc.) est interdit sauf si l'examen a lieu dans un laboratoire auquel cas seuls les équipements du laboratoire doivent être utilisés.

Tableau 3 – Sommaire des évaluations

Évaluation	Valeur	Commentaire
examen 1	35 %	individuel
examen 2	35 %	individuel et récapitulatif
travaux	12 %	en tandem
projet	18 %	en équipe de trois à quatre
Total	100 %	

Tout étudiant, toute étudiante, qui omet de remettre un travail au moment prescrit doit rencontrer l'enseignant afin de déterminer une nouvelle date de remise. Dans tous les cas, une pénalité de 10 % par jour de retard est imposée.

L'évaluation est faite en tenant compte de la clarté des documents et du respect de la méthodologie du génie logiciel. Conformément à l'article 17 du Règlement facultaire d'évaluation des apprentissages⁴, l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation. Toute situation de plagiat sera traitée en conformité avec le Règlement des études⁵ de l'Université de Sherbrooke, notamment l'article 8.1.2.

En cas de circonstances extraordinaires au-delà du contrôle de l'Université de Sherbrooke et sur décision de celle-ci, l'évaluation des apprentissages de cette activité est sujette à changement.

2.4 Échéancier des travaux

Voir le calendrier (section 2.2).

3 Matériel nécessaire pour le cours

Le plan de cours et les présentations utilisées en cours sont disponibles sur le site de l'enseignant⁶.

4 Définitions

AEF	automate à états fini.
CA	(modèle ou diagramme) classe-association (parfois appelé orienté objet, OO).
CU	cas d'utilisation.
DC	diagramme de contexte
DCU	diagramme de cas d'utilisation.
DD	dictionnaire de données.
DFD	diagramme de flux de données
EA	(modèle ou diagramme) entité-association.
IEEE	<i>The Institute of Electrical and Electronics Engineers, inc.</i>
MCD	modèle conceptuel de données.
MEF	machine à états finie.
SAS	spécification d'architecture du système.
SCL	spécification de conception du logiciel (IEEE <i>SDD software design document</i>).
SEL	spécification des exigences du logiciel (IEEE <i>SRS software requirement specification</i>).
SES	spécification des exigences du système (IEEE <i>SRS system requirement specification</i>).

⁴ <http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-devaluation/>

⁵ <http://www.usherbrooke.ca/programmes/references/reglement/>

⁶ <http://info.usherbrooke.ca/llavoie/enseignement/IFT187>

SXL	spécification des essais du logiciel.
SXS	spécification des essais du système.
TD	table de décision.
UML	<i>Unified Modeling Language</i> .
UML-A	diagramme d'activité utilisant la notation UML; voir MEF.
UML-C	diagramme de classe utilisant la notation UML; voir CA.
UML-S	diagramme de séquence utilisant la notation UML.
UML-U	diagramme de cas d'utilisation utilisant la notation UML; voir DCU.

5 Références

5.1 Références essentielles

[Bray2002]

Ian K. BRAY.

An Introduction to requirements engineering.

Addison-Wesley, 2002; ISBN 0-201-76792-9; [UdeS QA 76.758 B744 2002].

[IGL 301]

Luc LAVOIE.

IGL 301 – Spécification et validation des exigences (notes complémentaires et synthétiques).

<http://info.usherbrooke.ca/llavoie/enseignement/IGL301>

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada, janvier 2012.

[GLOGUS]

GROUPE Μῆτις

GLOGUS – recueil de modèles de documents pour le développement logiciel.

<http://info.usherbrooke.ca/llavoie/glogus.php>

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada, janvier 2012.

[Hull2011]

Elizabeth HULL, Ken JACKSON, Jeremy DICK.

Requirements Engineering.

3rd édition, Springer, 2011; ISBN 978-1-84996-405-0.

[Printz2012]

Jacques PRINTZ.

Architecture logicielle — Concevoir des applications simples, sûres et adaptables.

Dunod, 3^e édition, 2012; ISBN 978-2-10-057865-8.

5.2 Références importantes

[Braude2011]

Eric J. BRAUDE, Michael E. BERNSTEIN.

Software engineering : modern approaches.

2nd édition, John Wiley & sons, 2011, ISBN 978-0-471-69208-9; [UdeS QA 76.758 B743 2011].

[Contantinidis2011]

Yves CONTANTINIDIS.

Expression des besoins pour le système d'information.

Eyrolles, 2011. ISBN 978-2-212-12783-6.

[Elmasri2007]

Ramez ELMASRI, Shamkant B. NAVATHE

Fundamentals of database systems.

5th édition, Pearson Addison Wesley, 2007, ISBN 0-321-36957-2.

[Leffingwell2003]

D. LEFFINGWELL, D. WIDRIG.

Managing software requirements – A use case approach.

2nd edition, Addison-Wesley, 2003, ISBN 0-321-12247-X; [UdeS QA 76.76 D47L44 2003].

[WOL2004]

Wall-On-Line : l'e-gouvernement wallon.

La boîte à outils : 15 méthodes d'implication des utilisateurs.

http://egov.wallonie.be/boite_outils_methodes/index.htm

(version en date du 17 décembre 2004 consultée le 11 mai 2007, disponible maintenant sous

<http://info.usherbrooke.ca/lavoie/projets/GLOGUS/wall-on-line.pdf>).

[Wood1995]

Jane WOOD, Denise SILVER.

Joint Application Development.

2nd edition, John Wiley & sons, 1995. ISBN 0-471-04299-4.

5.3 Références utiles

[Audibert2009]

Laurent AUDIBERT;

UML 2 : De l'apprentissage à la pratique;

Ellipses Marketing, Paris, 2009; ISBN 978-2729852696.

De larges extraits sont disponibles sous <http://laurent-audibert.developpez.com/Cours-UML/>
(consulté le 2012-01-08).

[Davis2007]

M. DAVIS;

Requirements Bibliography,

<http://web.uccs.edu/adavis/UCCS/reqbib.htm> (consulté le 2007-03-15).

[De Lucia2008]

Andrea DE LUCIA, Filomena FERRUCCI, Genny TORTORA, Maurizio TUCCI (ed).

Emerging Methods, Technologies, and Process Manafement in Software Engineering.

John Wiley and sons, 2008. ISBN 978-0-470-08571-4.

[GDT]

Grand dictionnaire terminologique.

Office québécois de la langue française.

<http://www.granddictionnaire.com>

(consulté le 2011-12-15).

[IEEE1233]

IEEE Guide for Developing System Requirements Specifications;

IEEE Std 1233-1998, IEEE, New York, 1998.

[IEEE830]

IEEE Recommended Practice for Software Requirements Specifications;

IEEE Std 830-1998, IEEE, New York, 1998.

[IEEE12207]

Industry Implementation of International Standard ISO/IEC 12207-1995;

IEEE Std 12207.0-1996, IEEE, New York, 1998.

[Jackson1995]

Michael JACKSON;

Software Requirements & Specifications;

Addison Wesley, 1995; ISBN 0-201-87712-0; [UdeS QA 76.76 D47J33 1995].

[Jackson2001]

Michael JACKSON;

Problem frames;

ACM Press Book, Addison Wesley, 2001; ISBN 0-201-59627-X; [UdeS QA 76.76 D47J32 2001].

- [Jacobson1999] (traduit en français, voir [Jacobson2000])
Ivar JACOBSON, Grady BOOCH, James RUMBAUGH;
The unified software development process;
Addison-Wesley, 1999; ISBN 0-201-57169-2.
- [Jacobson2000] (traduction de [Jacobson1999])
Ivar JACOBSON, Grady BOOCH, James RUMBAUGH;
Le processus unifié de développement logiciel;
Eyrolles, 2000; ISBN 2-212-09142-7; [UdeS 76.76 D47J3514 2000].
- [Larman2005]
Craig LARMAN;
Applying UML and patterns - an introduction to object-oriented analysis and design and iterative development;
3rd edition, Prentice-Hall, Upper Sadel River (NJ), 2005;
ISBN 0-13-148906-2; [UdeS QA 76.9 O35L3714 2005].
- [Pfleeger2010]
Shari Lawrence PFLEEGER, Joanne M. ATLEE.
Software Engineering – Theory and Practice.
4th edition, Prentice Hall, 2005; ISBN 978-0-13-606169-4.
- [Pressman2005]
Roger S. PRESSMAN.
Software Engineering - A practioner's Approach.
6th edition, McGraw-Hill, 2005; ISBN 0-07-301933-X; [UdeS QA 76.758 P73 2005].
- [Sommerville2011]
Ian SOMMERVILLE.
Software Engineering.
9th edition, Addison-Wesley, 2011; ISBN 978-0-13-703515-1.