



Cours :	INF 734 – Méthodes formelles de spécification
Trimestre :	Hiver 2024
Enseignant :	Évariste Valéry BÉVO WANDJI

1. Mise en contexte

L'étape de spécification est reconnue comme fondamentale mais également ardue dans le processus de développement des systèmes informatiques en général, des systèmes logiciels dits « complexes » ou critiques (c'est-à-dire dont les défaillances peuvent avoir des conséquences dramatiques directes ou indirectes sur les personnes qui sont dans leur environnement), en particulier. De la qualité des spécifications dépend bien souvent la qualité du système logiciel résultant (fiabilité, maintenabilité, etc.) et aussi la maîtrise des coûts de maintenance associés. Les méthodes et outils de spécification jouant un rôle crucial dans la qualité des spécifications, il convient donc de leur accorder une attention particulière en développement logiciel. Si les méthodes dites « informelles » semblent plus simples et accessibles, les méthodes dites « formelles » permettent une formulation rigoureuse des besoins et une formalisation de la vérification et la validation des systèmes logiciels construits. De plus, l'application des méthodes formelles peut également guider le processus de test et constitue un atout pour la réutilisation. C'est donc sans surprise que les méthodes formelles occupent de plus en plus une place de choix en génie logiciel en général et en développement de logiciels dits « critiques » en particulier.

Le cours se propose de contribuer à la vulgarisation de ces méthodes en phase de spécification, en apportant une aide et un bagage de base aux étudiants et étudiantes qui en sont à leurs débuts dans la spécification formelle des logiciels.

2. Place du cours dans le programme

Qu'il s'agisse du programme de deuxième cycle en génie logiciel ou encore du programme de maîtrise en génie logiciel incluant un cheminement de type cours en technologies de l'information, le cours permettra à l'étudiante ou l'étudiant d'approfondir ses connaissances en matière de spécification de systèmes informatiques (utilisation des méthodes formelles de spécification). Le cours est conçu de façon à intégrer des étudiantes ou des étudiants ayant différents niveaux de connaissances et d'expérience en matière de développement logiciel, en fournissant un cadre de référence uniforme pour l'étude des méthodes formelles de spécification. Il est toutefois supposé que les étudiantes ou les étudiants sont familiers avec le processus de développement logiciel.

3. Objectifs généraux

(Selon l'annuaire des activités pédagogiques, disponible sur le site de l'Université : <https://www.usherbrooke.ca/admission/fiches-cours/INF734/>)

Cette activité pédagogique vise à développer chez l'étudiante ou l'étudiant les aptitudes suivantes :

- utilisation des notations formelles;
- lecture et compréhension des spécifications formelles;
- identification des différentes méthodes formelles de spécification;
- utilisation des méthodes formelles et justification de leur utilisation pour spécifier des systèmes et analyser les propriétés d'un système;

4. Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable de :

- justifier l'utilisation des méthodes formelles dans le processus développement logiciel en général et la phase de spécification en particulier ;
- classer et comparer les méthodes formelles de spécification ;
- utiliser une méthode formelle de spécification ;
- identifier les avantages et les limites de l'utilisation des méthodes formelles pour la spécification logicielle ;
- utiliser des notions de logique propositionnelle et de logique des prédicats du premier ordre dans la spécification de logiciels ;
- écrire, lire et comprendre des spécifications formelles de systèmes logiciels avec un langage logico-ensembliste en général et la notation B en particulier.

5. Planification hebdomadaire

Semaine	Contenu du cours	Travaux	Poids
1 & 2 (sur Moodle)	Logique et spécification : <ul style="list-style-type: none">• la logique propositionnelle ;- tour d'horizon- méthodes de démonstration		
1 & 2 (sur Moodle)	Logique et spécification : <ul style="list-style-type: none">• la logique des prédicats du premier ordre- tour d'horizon- démonstrations		
1	Présentation du contenu du cours et entente d'évaluation L'étape d'analyse et spécification dans le processus de développement Introduction aux méthodes formelles : <ul style="list-style-type: none">• terminologie;• classification des méthodes de spécification;• les langages de spécification;• les applications industrielles.		



2	Spécification formelle : <ul style="list-style-type: none"> • la méthode B <ul style="list-style-type: none"> - description - processus - obligation de preuve • le langage B <ul style="list-style-type: none"> - langage de composants - principales clauses - illustration 		
3	Spécification formelle : <ul style="list-style-type: none"> • le langage B <ul style="list-style-type: none"> - langage de prédicats - langage d'expressions - illustration 	<i>Contrôle continu (30min)</i> <i>Travail No 1 (énoncé)</i>	5%
4	Spécification formelle : <ul style="list-style-type: none"> • le langage B <ul style="list-style-type: none"> - langage de substitutions 		
5	Spécification formelle : <ul style="list-style-type: none"> • le langage B <ul style="list-style-type: none"> - langage de substitutions (<i>suite ...</i>) - illustration • le langage B <ul style="list-style-type: none"> - cohérence d'une spécification - illustration 	<i>Contrôle continu (30min)</i>	5%
6	Spécification formelle : <ul style="list-style-type: none"> • le langage B <ul style="list-style-type: none"> - cohérence d'une spécification (<i>suite ...</i>) - illustration 	<i>Travail No 1 (remise des travaux au professeur)</i> <i>Travail pratique de session (énoncé)</i>	15%
7	Spécification formelle : <ul style="list-style-type: none"> • la méthode B <ul style="list-style-type: none"> - Étude de cas 		
8	Spécification formelle : <ul style="list-style-type: none"> • la méthode B <ul style="list-style-type: none"> - Étude de cas (<i>suite ...</i>) 	<i>Contrôle continu (30min)</i>	5%
9	Spécification formelle : <ul style="list-style-type: none"> • la méthode B <ul style="list-style-type: none"> - Étude de cas (<i>suite ...</i>) • la notation OCL d'UML 		
10	Spécification formelle : <ul style="list-style-type: none"> • la notation OCL d'UML (<i>suite ...</i>) • illustration 		
11	Spécification formelle : <ul style="list-style-type: none"> • la notation OCL d'UML (<i>suite ...</i>) • illustration 	<i>Travail No 2 (énoncé)</i>	
12	Spécification formelle : <ul style="list-style-type: none"> • la notation OCL d'UML (<i>suite ...</i>) • illustration 	<i>Contrôle continu (30min)</i>	5%
13	Spécification formelle : <ul style="list-style-type: none"> • la notation OCL d'UML (<i>suite ...</i>) • illustration 		
14	<i>Synthèse & Consultation Travail de session</i>	<i>Travail No 2 (remise des travaux au professeur)</i>	15%

15	Examen final	<i>Examen final</i>	30%
		<i>Travail pratique de session (remise des travaux au professeur)</i>	20%

6. Approche pédagogique préconisée

L'approche pédagogique qui permettra d'atteindre les objectifs visés par le cours est la suivante : un enseignement hebdomadaire sous la forme d'un cours magistral, avec exemples et discussions en classe pour une période de trois heures. L'étudiant ou l'étudiante devra compléter sa formation par des travaux individuels proposés et des lectures personnelles suggérées. **En prélude au cours, l'étudiante ou l'étudiant devra compléter avant la troisième semaine de cours un module préparatoire disponible sur Moodle. Ce module porte sur la logique propositionnelle, la logique des prédicats et les démonstrations de formules. Le premier contrôle continu portera sur les notions couvertes dans ce module.** Le cours pourra être adapté selon les besoins des étudiantes et étudiants, si nécessaire.

7. Évaluation de l'apprentissage

Description sommaire	Pondération	Pondération individuelle	Pondération de groupe
Contrôles continus <ul style="list-style-type: none"> <i>travail individuel</i> <i>petits exercices ou questions de cours</i> 	4 x 5%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Travail No 1 (« take-home ») <ul style="list-style-type: none"> <i>travail individuel</i> <i>exercices (langage B)</i> 	15%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Travail No 2 (« take-home ») <ul style="list-style-type: none"> <i>travail individuel</i> <i>exercices (langage OCL)</i> 	15%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Travail pratique de session <ul style="list-style-type: none"> <i>travail de groupe (groupes de 3 au maximum)</i> <i>projet B</i> 	20%	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Examen final : <ul style="list-style-type: none"> <i>travail individuel</i> <i>questions de cours et exercices</i> 	30%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<u>Notez bien</u> : Pour tous les travaux, un rapport classique de TP (documents papier et électronique) doit être remis au professeur. La qualité du français compte pour 5% de la note. La pénalité pour le retard dans la remise d'un travail est de 5% de la note par jour ouvrable de retard.			

Toute modification liée à une date de remise doit avoir été acceptée par le groupe et la direction du CeFTI dans un délai plus grand qu'une semaine avant l'échéance de la remise.

8. Plagiat

Conformément à l'article 9 du Règlement des études de l'Université de Sherbrooke, le plagiat, soit le fait dans une activité évaluée de faire passer pour sien les idées et le travail d'autrui, est un délit académique qui peut être sanctionné par les autorités disciplinaires compétentes. Peuvent être imposées à titre de sanctions, l'une ou plusieurs des mesures suivantes :

- a) la réprimande simple ou sévère consignée au dossier étudiant pour la période fixée par l'autorité disciplinaire ou, à défaut, définitivement. En cas de réprimande fixée pour une période déterminée, la décision rendue demeure au dossier de la personne aux seules fins d'attester de l'existence du délit en cas de récidive;
- b) l'obligation de reprendre une production ou une activité pédagogique, dont la note pourra être établie en tenant compte du délit survenu antérieurement;
- c) la diminution de la note ou l'attribution de la note E ou 0;
- d) le renvoi du dossier à la personne responsable de l'évaluation d'une production ou d'une activité pédagogique pour qu'elle attribue une nouvelle note en tenant compte du délit.

Par plagiat, on entend notamment :

- copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets;
- reproduire des présentations, des dessins, des photographies, des graphiques, des données sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire;
- utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources;
- résumer ou paraphraser l'idée d'un auteur sans en indiquer la source;
- traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets;
- utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord);
- acheter un travail sur le Web ou ailleurs et le faire passer pour sien;
- utiliser sans autorisation le même travail pour deux activités différentes (auto plagiat).

9. Adresse électronique pour remise des travaux

evariste.valery.bevo.wandji@usherbrooke.ca



10. Bibliographie

- [1].FRAPPIER, M. and HABRIAS, H., *Software Specification Methods : An Overview Using a Case Study*, Springer Verlag, April 2006, 448 pages, ISBN: 1-85233-353-7.
- [2].HABRIAS, H., *Spécification formelle avec B*, Editions Hermès - Lavoisier, 2001.
- [3].Wang, Jiacun. *Formal Methods in Computer Science*. CRC Press, 2019.
- [4].Ghezzi C. (2018) Formal Methods and Agile Development: Towards a Happy Marriage. In: Gruhn V., Striemer R. (eds) *The Essence of Software Engineering*. Springer, Cham. https://doi.org/10.1007/978-3-319-73897-0_2
- [5].O'Regan, Gerard. *Concise guide to formal methods*. Springer, Cham, 2017.
- [6].Lecomte, T., Pinger, R., Romanovsky, A. *Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification*. First International Conference, RSSRail 2016, Paris, France, June 28-30, 2016, Proceedings
- [7].Jackson, D., *Software Abstractions, Logic, Language, and Analysis, Revised Edition*, MIT Press, 2012
- [8].Alagar V.S., Periyasamy K. *The B-Method*. In: *Specification of Software Systems. Texts in Computer Science*. Springer, London, 2011
- [9].TREMBLAY, G., *Modélisation et spécification formelle des logiciels (édition revue et augmentée)*, – Loze-Dion Editeurs Inc., Montréal, 4e trimestre 2004.
- [10]. KELLEY SOBEL, A.E. and CLARCKSON, M., “Formal Methods Application: An Empirical Tale of Software Development”, in *IEEE Transactions on Software Engineering*, Vol. 28, No 3, pp. 308-320, 2002.
- [11]. GIRAULT, C. and VALK, R., *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*, Springer-Verlag, 2002.
- [12]. PALSHIKAR, G.K., “Applying Formal Specifications to Real-World Software Development”, in *IEEE Software* , pp. 89-97, Novembre/Décembre 2001.
- [13]. SCHNEIDER, S., *The B-Method: an Introduction*, Palgrave, Cornerstones of Computing series, UK, October 2001.
- [14]. DIAZ, M., *Les Réseaux de Petri - Modèles fondamentaux*, Hermes Science, Paris, 2001.
- [15]. BOWEN, J.P. and HINCHEY, M.G. “The Ten Commandments of Formal Methods”, in *Computer* , Vol. 28, No 4, pp.56-63, 1995.
- [16]. HINCHEY, M. and BOWEN, J., *Applications of Formal Methods*, Prentice Hall International Series in Computer Science, series editor Prof. C.A.R. Hoare. 1995.
- [17]. MONIN, J.F. and HINCHEY, M.G., *Understanding formal methods*, Springer, 2003, ISBN 1852332476
- [18]. BERZINS, V. and LUQI, *Software Engineering with Abstractions*, Addison-Wesley Publishing Co., 1991.
- [19]. JONES, C.B., *Systematic Software Development using VDM*, Prentice Hall International, 1990.
- [20]. HALL, A., “Seven Myths of Formal Methods”, in *IEEE Software*, 7(5), pp. 11-19, 1990.

Quelques sites Web

 <http://www.methode-b.com/>

- ☞ Manuel de référence <http://www.atelierb.eu/wp-content/uploads/sites/3/ressources/manref1.8.6.fr.zip> (langage B)
- ☞ <http://www.fmeurope.org/>
- ☞ <http://dblp.uni-trier.de/db/conf/fm/> (congrès)
- ☞ <http://www.omg.org/spec/OCL/> (OCL)
- ☞ Langage Z
- ☞ Langage Spec)
- ☞ <http://www.atelierb.eu/> (outil Atelier B)