



HAL
open science

À la recherche de caractéristiques d'un modèle de l'apprenant des concepts fondamentaux de la programmation (CFP)

Ph.D. Yves Boudreault, M.Sc.A. Anka Stoykova

► **To cite this version:**

Ph.D. Yves Boudreault, M.Sc.A. Anka Stoykova. À la recherche de caractéristiques d'un modèle de l'apprenant des concepts fondamentaux de la programmation (CFP). Jun 2007. hal-00161489

HAL Id: hal-00161489

<https://hal.archives-ouvertes.fr/hal-00161489>

Submitted on 10 Jul 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

À la recherche de caractéristiques d'un modèle de l'apprenant des concepts fondamentaux de la programmation (CFP)

Ph.D. Yves Boudreault*, M.Sc.A. Anka Stoykova**

* C.P. 6079, succ. Centre-Ville, Montréal
Québec, Canada, H3C 3A7
École Polytechnique de Montréal, bureau M4117
yves.boudreault@polymtl.ca

** C.P. 6079, succ. Centre-Ville, Montréal
Québec, Canada, H3C 3A7
École Polytechnique de Montréal, bureau M4202
anka-stoykova.mihaylova@polymtl.ca

RÉSUMÉ Cette recherche a pour objectif de contribuer à l'élaboration d'un modèle de l'apprenant des concepts fondamentaux de la programmation. Ce modèle s'inscrit dans un projet d'envergure dont l'objectif est la conception d'un EIAH qui tiendra compte des besoins évolutifs de l'apprenant et de produire des situations d'apprentissage tirant profit de ces styles d'apprentissages. Afin d'identifier certaines caractéristiques de l'apprenant, nous avons proposé à 18 étudiants de participer à une expérience. L'expérience consiste à compléter le test de Felder-Silverman et de réaliser ensuite deux exercices de programmation, l'un à caractère abstrait et l'autre à caractère concret. Le test de Felder-Silverman nous informe sur les préférences de l'étudiant selon le modèle des styles d'apprentissage des mêmes auteurs. Nous avons obtenu plusieurs informations intéressantes de la comparaison des résultats obtenus de chaque type d'activité en fonction du style d'apprentissage.

MOTS-CLÉS : stratégies d'apprentissage, programmation, modèles psychologiques, C++, informatique.

1. Description de l'expérimentation

L'expérience consiste, pour l'étudiant, à compléter le test de Felder-Silverman et de réaliser ensuite deux exercices de programmation, l'un à caractère abstrait et l'autre à caractère concret. La description de chacun de ces éléments est réalisée dans les sections suivantes.

1.1. Le modèle psychologique de l'apprentissage de Felder-Silverman

Il existe plusieurs modèles psychologiques d'apprentissage. Chacun a ses priorités et ses avantages. Nous avons choisi pour cette expérience le modèle de Felder-Silverman parce qu'il a été utilisé pour plusieurs études portant sur des étudiants en sciences et en génie, qu'il contient peu de questions, soit 44, et qu'il est disponible gratuitement.

D'après Felder (1998) l'apprentissage est un processus qui peut être divisé en deux parties: la réception de l'information et son traitement. Le modèle de Felder-Silverman classe les apprenants selon les moyens pertinents qu'ils utilisent pour percevoir l'information et traiter cette information. Il ne s'agit pas d'une classification stricte au sens où une personne d'un type particulier peut, au besoin ou selon la situation, se débrouiller avec des méthodes du type opposé. Le tableau 1 résume les types psychologiques selon le modèle de Felder-Silverman (Felder 1996).

<i>sensoriel</i> (concret, pratique, orienté vers les faits et procédures) ou <i>intuitif</i> (conceptuel, innovateur, orientés vers les théories)	SNS - INT
<i>visuel</i> (préfère les représentations visuelles avec des images, diagrammes...) ou <i>verbal</i> (préfère les explications écrites ou verbales)	VIS - VRB
<i>inductif</i> (préfère les présentations qui commencent par le spécifique vers le général) ou <i>déductif</i> (préfère les présentations qui commencent par le général vers le spécifique)	-
<i>actif</i> (préfère travailler en équipe) ou <i>réflexif</i> (préfère travailler seul)	ACT - REF
<i>séquentiel</i> (linéaire, bien ordonné) ou <i>global</i> (holistique, penseur systématique)	SEQ - GLO

Tableau 1 Types psychologiques du modèle de Felder-Silverman

1.2. Exercices à caractère abstrait

Un court texte décrit l'exercice de programmation à réaliser. Il s'agit d'imaginer le déplacement d'un robot. Plus précisément, l'exercice consiste à obtenir de l'utilisateur du programme la position initiale du robot et de le déplacer dans un espace bien défini, par exemple un rectangle 20×50. Le robot peut se déplacer selon les quatre points cardinaux : nord, sud, est et ouest. L'objectif est de s'assurer que le robot demeure dans le rectangle. Précisons que le programme s'exécute dans une fenêtre de type console et qu'il n'y a donc aucun élément de graphisme dans ce programme. Le programme doit prévoir un dialogue invitant l'utilisateur à choisir successivement la direction désirée pour le déplacement du robot jusqu'à spécifier

TERMINER pour mettre fin aux déplacements. Finalement, le programme doit sauvegarder et afficher tous les déplacements effectués et non effectués.

1.3. Exercices à caractère concret

Pour rendre concret l'exercice de programmation, nous avons recours à un montage électronique. La pièce maîtresse de ce montage est un microcontrôleur servant à manipuler des luminodiodes et un moteur. À l'aide de ce montage, les étudiants doivent réaliser deux exercices de programmation. Le premier consiste à concevoir un programme qui manipule un moteur, c.-à-d. le faire tourner et l'arrêter. Les étudiants doivent trouver deux impulsions, l'une qui démarre le moteur et l'autre qui l'arrête. Le deuxième exercice consiste à concevoir un programme qui impose d'allumer et d'éteindre quatre luminodiodes selon un ordre précis. Pour ces deux exercices, les étudiants devaient ajouter leurs instructions à programme existant. Cette stratégie a permis d'éviter toutes les complications de communication entre l'ordinateur et le montage. Le programme existant contient également une panoplie de fonctions nécessaires à la manipulation du montage électronique.

2. Les résultats

Tout d'abord les résultats concernant le test de Felder-Silverman. Tous les étudiants sont des représentants du groupe visuel VIS indépendamment qu'ils aient des préférences équilibrées ou bien marquées. Cette première constatation nous incite à exploiter des diagrammes, des images, des vidéos, etc. Une partie des étudiants qui ont participé à notre expérience sont des personnes équilibrées selon l'échelle ACT/REF et SEQ/GLO. Cela signifie que ces étudiants sont des représentants d'un groupe, qu'ils ont leurs propres préférences, mais peuvent au besoin se débrouiller avec des méthodes utilisées par les représentants de l'autre groupe. Le tableau 2 présente le nombre d'étudiants présents de chaque côté de l'échelle de préférence. Il est à remarquer la présence de représentants des deux côtés de l'échelle, soit : des étudiants qui suivent le processus linéaire (SEQ) pour résoudre un problème et d'autres qui cherchent de nouvelles méthodes (GLO) ; des étudiants qui se lancent aussitôt au travail (ACT) et d'autres qui aiment d'abord réfléchir et ensuite agir (REF).

ACT	7	11	REF
SNS	11	7	INT
VIS	18	-	VRB
SEQ	9	9	GLO

Tableau 2. Nombre d'étudiants selon la préférence

Selon Felder (1998), les étudiants en sciences et en génie sont généralement de types visuel, sensoriel, inductif et actif. Toujours selon Felder, les étudiants les plus créatifs seraient de type global. Considérant ces observations et les résultats obtenus, nous avons classé nos étudiants en deux groupes : actif et réflexif. Chacun de ces groupes est ensuite subdivisé en deux sous-groupes soit séquentiel ou global. Le tableau 3 présente ces subdivisions. L'analyse des productions des programmes s'effectuera en distinguant les membres de ces groupes.

ACTIF				REFLEXIF			
4 étud.		3 étud.		6 étud.		5 étud.	
ACT		ACT			REF		REF
SNS		SNS			INT		SNS
VIS		VIS		VIS			VIS
SEQ			GLO		GLO		SEQ

Tableau 3. Classification des étudiants participant à l'expérience

2.0. Comparaison des deux types d'exercices.

Une période de deux heures était prévue pour la rédaction des deux programmes. Nous avons invité ceux qui n'avaient pas terminé en deux heures à compléter leurs exercices à un moment de leur choix. Le pourcentage des programmes qui s'exécutent correctement selon le type de préférence est présenté au tableau 4. Les réflexifs remportent de toute évidence la palme. Le pourcentage des programmes qui s'exécutent correctement du groupe REF est 91% tandis que celui des ACT est de 57%. Pour le deuxième exercice, exercice à caractère concret, cette différence est moindre. Les deux groupes principaux (ACT et REF) ont des résultats similaires et comparables. Dans le tableau 4, précisons que le x des couples (x,y) sous les groupes SEQ et GLO correspond au nombre d'étudiants qui ont complété l'exercice abstrait et le y correspond au nombre d'étudiants qui ont complété l'exercice concret.

		Exercice à caractère abstrait			Exercice à caractère concret			
REF (11)	SEQ (5, 4)	OUI	80%	OUI	90.9%	37.5%	OUI	50%
		NON	20%			62.5%		
	GLO (6, 4)	OUI	100%	NON	9.1%	62.5%	NON	50%
		NON	0%			37.5%		
ACT (7)	SEQ (4, 4)	OUI	25%	OUI	57.1%	12.5%	OUI	41.7%
		NON	75%			87.5%		
	GLO (3, 2)	OUI	100%	NON	42.9%	100%	NON	58.3%
		NON	0%			0%		

Tableau 4. Pourcentage des programmes s'exécutant correctement

Pour l'exercice à caractère abstrait, tous les étudiants ont présenté des programmes qui montrent une bonne compréhension du sujet. Certains programmes

montrent une logique précise et d'autres dans lesquels il y a des segments mal conçus ou inutiles. Les représentants du groupe REF ont mieux travaillé et leurs résultats sont meilleurs que ceux du groupe ACT. Les étudiants qui sont plus réflexifs ont de meilleurs résultats que les étudiants actifs. Tous leurs programmes, sauf un, s'exécutent d'une façon correcte. Tous les étudiants, sauf un, de ce groupe font un contrôle des données à l'entrée. Neuf parmi eux font un affichage des résultats. La plupart ont utilisé des expressions booléennes complexes. Les étudiants du groupe ACT montrent une préférence vers utilisation de la structure `if` plutôt que `if-else`. Dans certains programmes de ces étudiants, on observe, soit l'absence d'un contrôle à l'entrée, soit l'absence d'affichage à la fin. Cependant, leurs programmes sont bien construits et le nombre de lignes de code est plus petit que celui des étudiants réflexifs. Les étudiants REF/SEQ utilisent plutôt la structure `do-while` tandis que les étudiants REF/GLO utilisent plutôt la structure `while`. Pour leur part, les étudiants ACT/SEQ se distinguent des étudiants ACT/GLO par des constructions algorithmiques plus élaborées.

Pour le deuxième exercice, celui à caractère concret, quatre étudiants n'ont malheureusement pas fait cet exercice. De plus, la deuxième partie de l'exercice manquait, celle avec les leds, pour deux des programmes obtenus. Les programmes manquants sont attribuables à trois représentants REF et un de type ACT. De nouveau les étudiants REF conçoivent des programmes avec un plus grand nombre de lignes de code. La construction algorithmique des programmes est plus élaborée chez les REF que chez les ACT. L'affichage chez les étudiants REF est mieux organisé que chez les ACT. Chez les ACT il y a des absences d'affichage des résultats. Les REF/SEQ utilisent plutôt la structure `do-while` tandis que les REF/GLO utilisent plutôt la structure `while`. Étonnamment, il y a peu de différence entre les deux groupes.

3. Conclusion

Initialement, nous envisageons un groupe expérimental d'environ soixante étudiants. Ayant réussi à convaincre que dix-huit étudiants, nous sommes conscients de la réserve avec laquelle nous devons interpréter nos résultats. Il nous faut dès lors qualifier ce travail d'exploratoire qui nous permet d'identifier quelques pistes.

Nous espérons discerner quels styles d'apprentissage seraient les plus appropriées dépendamment de l'activité à réaliser. Notre hypothèse de base était que les étudiants de type REF réussiraient mieux un exercice à caractère abstrait tandis que les étudiants de type ACT réussiraient mieux un exercice à caractère concret. Des résultats obtenus nous pouvons uniquement affirmer que les étudiants de type REF ont mieux performé dans l'exercice à caractère abstrait. Il nous faudra répéter l'expérience avec un plus grand nombre d'étudiants afin d'obtenir des résultats plus significatifs.

Tous les candidats qui ont participé à l'expérience sont de type VIS. Il nous apparaît essentiel qu'un EIAH dédié à l'apprentissage de la programmation doit

intégrer des éléments qui exploitent cette caractéristique. Les candidats se subdivisent ensuite en REF ou ACT. Cette subdivision nous suggère l'utilité de parcours adapté dans un EIAH. Nous pouvons à partir de l'information obtenue de la structure algorithmique des programmes réalisés imaginer certains parcours. Par exemple, un parcours offrant une décomposition selon une granularité fine des concepts pour les ACT et un parcours offrant une décomposition selon une granularité grossière avec une vue d'ensemble pour les REF. Il faudrait également porter une attention particulière à la spécification explicite des opérations à intégrer dans un programme, telles la vérification de l'exactitude de la donnée lue ou la présentation des résultats pour les ACT. Une autre facette à ajuster selon le type de l'apprenant concerne les structures de contrôles. Notre étude souligne certaines préférences de structure de contrôle selon le type. Une stratégie intéressante serait probablement d'introduire la structure exclue à partir de la structure préférée.

Au début du deuxième exercice, les étudiants ont observé comment le montage électronique fonctionne et ils devaient ensuite concevoir un programme réalisant le même comportement. Nous avons remarqué que dans cet exercice les différences entre les groupes se sont atténuées. Comme si les étudiants étaient dans des « conditions égales ». Les fautes commises sont similaires. Le pourcentage des programmes fonctionnels et non fonctionnels est quasi identique pour les deux groupes. Comment expliquer cela? Est-ce un type d'exercice souhaitable? Nous croyons qu'il s'agit d'une voie qu'il faut explorer d'avantage avant de tirer des conclusions.

Bibliographie

- [ADELSON 81] Adelson, B., « Problem solving and the development of abstract categories in programming languages. », *Memory and cognition*, 9 (4), 1981, p.422- 433.
- [ADELSON 85] Adelson, B., « Comparing natural and abstract categories: a case study from computer science. », *Cognitive Science*, 9, 1985, p.417- 430.
- [BROOKS 80] Brooks, R., « Studying programmers' behavior experimentally: the problems of proper methodology », *Communications of the ACM*, 23 (4), 1980, p. 207-213.
- [DÉTIENNE 98] Détienne, F., *Génie logiciel et psychologie de la programmation*. Éditions Hermès, Paris, 1998, p.184.
- [FELDER 96] Felder R. M., « Matters of Styles », *ASEE Prism*, 6(4), 1996, p.18-23.
- [FELDER 98] Felder R. M., « Learning and teaching Styles in Engineering Education », *Engr. Education*, 78(7), 1998, p.674-681.
- [RIST 86] Rist, R.S, « Plans in Programming: Definition, Demonstration, and Development ». In E. Soloway and S. Iyengar, (Eds.), *Empirical Studies of Programmers, First Workshop*, Norwood, NJ, Ablex Publishing Corporation, 1986, p.28-47.
- [SHNEIDERMAN 80] Shneiderman, B., « Software psychologist: Human factors in computer and Information Systems », Boston, Mass, Little, Brown and Company, 1980.
- [SOLOWAY et al. 82] Soloway, E, Erlich, K., Bonar, J. et Greenspan, J., « What do novices know about programming? » In A. Badre and B. Shneiderman, (Eds.), *Directions in Human - Computer Interaction*, New York, Ablex Publishing corporation, 1982, p.27-54.