

وزارة التعليم العالي و البحث العلمي

Université Badji Mokhtar –Annaba-

Badji Mokhtar –Annaba- University



جامعة باجي مختار- عنابة-

Année: 2015-2016

Faculté des sciences de l'ingénierie
Département d'informatique

THÈSE

Pour obtenir le diplôme de
Docteur en science

L'évaluation de l'apprenant dans les environnements d'apprentissage de TéLéTPs

Filière : Informatique

Spécialité : Intelligence Artificielle

Préparée par

Karima Boussaha

Devant le jury

Président :	Pr Meslati Djamel	Professeur	Université Badji Mokhtar ,Annaba
Directeur de thèse :	Pr Farid Mokhati	Professeur	Université larbi Ben M'hidi, Oum El Bouaghi
Examineurs :	Pr Seridi Hassina	Professeur	Université Badji Mokhtar ,Annaba
	Pr Amirat .A. El Karim	Professeur	Université Cherif Msaadia ,Souk Ahras
	Pr Laffi Yacine	Professeur	Université 8 mai 45, Guelma,

Dédicaces

À mes parents

À mon mari et ma chère petite fille Haoua

À tout(e)(s) mes sœurs et frères

Remerciements

Je tiens tout particulièrement à remercier le professeur Farid Mokhati mon directeur de thèse d'avoir dirigé mon travail de recherche, pour sa disponibilité, sa patience, son immense gentillesse, et ses conseils. C'est la seule personne qui m'a fait une confiance et pris mes mains et me donner un espoir de terminer mes recherches dans une période très difficile et grâce à lui que je peux terminer mes recherches de doctorat qui aurait arrêté la deuxième année de thèse merci beaucoup mon cher professeur.

Mes remerciements s'adressent aussi aux membres du jury: Seridi Hassina, professeur à l'université Badji Mokhtar Annaba, Amirat Abd El Karim, professeur à l'université Cherif Msaadia de Souk Ahras, Lafifi Yacine, professeur à l'université 8 mai 45 de Guelma, d'avoir la gentillesse d'accepter de juger ce travail et pour l'intérêt qu'ils ont manifesté au sujet.

Mes remerciements s'adressent aussi à Meslati Djamel, professeur à l'Université Badji Mokhtar Annaba, de m'avoir fait l'honneur de présider le jury de ma soutenance.

J'adresse mes remerciements à Monsieur Benmouhamed Hcen, maître de conférences à l'université de LYON pour ses conseils et son orientation ainsi que sa femme Fadila pour son immense gentillesse.

Sans oublier de remercier les membres de l'équipe Silex, Laboratoire de Recherche en Informatique d'université Claude Bernard (LIRIS), avec lesquelles j'ai pu avoir de nombreux échanges, et évoluer mon esprit de débat scientifique.

J'adresse mes remerciements à monsieur Bouri Elhadj, maître de conférences dans le département d'anglais à l'université d'oum Elbouaghi pour son aide et ses corrections multiples de mon article de publication.

J'adresse mes remerciements à toutes mes amies : Farida et ses enfants, Sabrina et ces filles, Razika, Zineb, Samia et son fils , Nawel, Samira et ces enfants , Meriem et son fils, Aicha et ces enfants et Anissa.

Tous mes remerciements vont aussi à tous les étudiants des départements informatiques, de l'université d'Oum-El-Bouaghi pour leurs commentaires et leurs participations de plus près aux enquêtes et aux expérimentations que j'ai menées.

Karima Boussaha , le 14 Avril 2016

Table des matières

Table des matières	I
Liste des figures	VIII
Liste des tableaux.....	XII
Abstract.....	XIV
Résumé.....	XV
Introduction Générale	01
Cadre et Motivations.....	01
Problématique et Contribution.....	02
Organisation du manuscrit.....	03
Chapitre 01 : Les EIAH et les TélÉTPs	05
1. Introduction	05
Partie 1	
2. Les EIAH.....	06
2.1. Définition	06
2.2. Terminologies concernant quelques modes de formations.....	06
2.3. Les systèmes LMS et LCMS	10
a. Le LMS.....	10
b. LCMS:.....	10
3. Quelques plateformes d'apprentissage.....	14
3.1. Claroline.....	14
3.2. Moodle.....	16
3.3.SAKAI.....	18
4. Étude comparative	19
Partie 2	
5. les TélÉTPs.....	21
6. Les travaux pratiques(TPs).....	21
6.1. Définition.....	21
6.2. Les objectifs pédagogiques d'un TP.....	22

6.3. Les interactions dans les TPs.....	23
6.4. Typologie d'un TP.....	23
6.5. Phases de déroulement d'un TP.....	25
6.6. Limites d'un TP.....	27
6.7. Position de nos travaux.....	27
7. Les travaux pratiques à distance (TéLéTPs).....	27
7.1. Définition	27
7.2. Apport et limites des TéLéTPs	28
7.3. Typologies des situations	28
8. Quelques EIAH consacrés à l'apprentissage des TéLéTPs.....	28
8.1. Tpline.....	29
8.2. Évariste.....	30
8.3. ICTT@Lab.....	30
8.4. E-TéLéTPC@AALP.....	31
9. Conclusion.....	32

Chapitre 02 : Évaluation des apprenants dans les EIAH

33

1. Introduction.....	33
2. L'évaluation dans les activités d'apprentissage.....	33
2.1.D'où vient le mot évaluation.....	33
2.2.Évaluation, contrôle : Quelles sont les différences !.....	34
2.3. Les acteurs de catégorisation des buts de l'évaluation?.....	35
2.4. Les critères d'évaluation	36
2.5. Correspondance entre objectifs pédagogiques et outils d'évaluation.....	37
2.6. Les différentes formes d'évaluation.....	39
3. Étude comparative entre les différentes formes de l'évaluation.....	42
4. L'évaluation dans les EIAH.....	43
4.1. Évaluation, session d'apprentissage quel relation !.....	43
4.2. La portée de l'évaluation.....	44
4.3. L'évaluation sommative et pronostique.....	45
4.4. L'évaluation diagnostique des compétences.....	45

4.5. L'évaluation sommative et diagnostique	46
5. Évaluation de production : l'auto-évaluation.....	47
5.1. L'évaluation dans GenEval.....	47
6. Évaluation de la démarche : l'assistance à l'évaluation.....	48
7. L'évaluation dans les EIAH dédiées à L'apprentissage des travaux pratiques à distance	48
7.1. L'évaluation dans la plateforme TEB.....	49
7.2. OASYS.....	50
8. Synthèse des types d'évaluation dans les EIAH.....	51
9. L'évaluation de l'apprenant dans notre plateforme : E-TéléTPC@AALP.....	52
9.1. Évaluation appariée : une nouvelle forme d'évaluation proposée.....	52
10. Conclusion	53

Chapitre 03 : Techniques d'Appariement de Graphes et Calcul de Similarité 54

1. Introduction.....	54
2. Les techniques d'appariement des graphes.....	55
2.1. Notions de base sur les graphes	55
2.2. Appariement de graphes	57
2.3. Les algorithmes de recherche d'un meilleur appariement.....	67
2.3.1 Algorithme glouton	67
2.3.2. Algorithme de Recherche locale Taboue réactive (RTS, Reactive Tabu Search).....	67
2.3.3. Algorithme d'optimisation par colonie de fourmis.....	68
2.4. Classification des techniques d'appariement	68
2.5. Types d'apparieurs.....	69
2.6. Représentation des diagrammes UML sous forme de graphes UML.....	70
2.7. Implémentation d'un programme orienté objet C++ en un modèle objet et vice versa.....	70
2.7.1. Diagramme de classes	70
2.7.2. Diagramme de communication.....	72
3. Conclusion.....	73

Chapitre 04 : La programmation orientée objet: un domaine d'application pour l'apprentissage des activités de TélÉTPs **74**

1. Introduction	74
2. Épistémologie de la programmation.....	74
2.1. Un peu d'historique.....	74
2.2. Définition de la programmation.....	76
2.3. Décomposition de l'activité de la programmation.....	76
3. Principales Difficultés dans l'apprentissage des TélÉTPs de la programmation, d'où viennent-elles?.....	78
3.1. Première hypothèse : difficultés à cause des caractéristiques intrinsèques de la discipline de la programmation.....	78
3.2. Deuxième hypothèse : difficultés d'utilisation des méthodes classiques (en présentiel) dans l'apprentissage.....	81
3.2.1. Difficultés de l'apprentissage présentiels des TP en programmation.....	81
3.2.2. L'efficacité d'utilisation de l'apprentissage à distance dans les activités de la programmation.....	82
4. L'activité de la programmation et les méthodes d'évaluation automatiques.....	87
4.1. L'approche statique.....	87
4.2. L'approche dynamique.....	88
5. Avantages de La programmation orientée objet (la POO).....	88
6. Conclusion.....	89

Chapitre 05 : Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant **91**

1. Introduction	91
2. Vue générale sur la plateforme E-TéléTPC@AALP	93
2.1. Architecture de la plateforme E-TéléTPC@AALP	93
2.2. Le principe de fonctionnement de la plateforme E-TéléTPC@AALP.....	95
2.2.1. Interactions de l'administrateur avec la plateforme et les autres acteurs	95
2.2.2. Interactions de l'apprenant avec la plateforme et les autres acteurs.....	95
2.2.3. Interactions du tuteur avec la plateforme et les autres acteurs.....	96
2.2.4. Interactions de l'enseignant avec la plateforme et les autres acteurs.....	96

3. Le générateur automatique de diagrammes UML.....	96
3.1. Pour le diagramme de classes.....	96
3.2. Pour le diagramme de communication.....	96
4. L'approche proposée pour l'évaluation de l'apprenant.....	97
4.1. Types d'apparieurs.....	97
4.1.1. Apparieurs hybrides	97
4.1.2. Apparieurs composites.....	98
4.2. Le processus d'évaluation de l'apprenant	98
• Phase 1.....	99
• Phase 2.....	99
• Phase 3.....	99
• Phase 4.....	100
5. Comparaison entre le programme de l'apprenant et celui de l'enseignant.....	101
5.1. Technique d'appariement proposée.....	101
5.1.1. Type d'appariement appliqué.....	102
5.1.2. La cardinalité de l'appariement.....	102
5.1.3. Algorithme de comparaison	102
5.1.4. La technique d'analyse utilisée.....	102
5.1.5. Types de diagrammes à comparer	103
➤ Diagramme sous-spécifié.....	103
➤ Diagramme sur spécifié.....	103
➤ Diagramme identiques.....	103
5.2. L'algorithme d'appariement développé	103
5.2.1. Critères de comparaison.....	104
5.2.2. Algorithme.....	104
5.2.3. Les étapes de déroulement de l'algorithme.....	104
6. Conclusion.....	108

Chapitre 06 : Environnement d'évaluation de l'apprenant	109
--	------------

1. Introduction	109
2. L'architecture générale de l'outil d'évaluation.....	110
2.1. L'outil éditeur.....	110

2.2. L’outil générateur.....	110
2.3. L’outil apparieur.....	110
3. Présentation de l’environnement	111
3.1. L’outil éditeur.....	111
3.2. L’outil générateur.....	112
3.3. L’outil apparieur.....	113
3.3.1. Processus de comparaison de diagramme de classes générés.....	113
• Measure de similarités	114
- Similarité par rapport à l’appariement m	114
- Similarité par rapport aux contraintes	115
- Similarité totale	115
3.3.2. Processus de comparaison de diagrammes de communication générés.....	116
3.3.3. Le calcul de la similarité globale entre les diagrammes de l’enseignant et les diagrammes de l’apprenant.....	117
3.3.4. .Liste des erreurs détectées.....	118
4. Évaluation de l’outil.....	119
4.1. L’utilité de notre outil.....	119
4.1.1. Pour l’enseignant.....	119
4.1.2. Pour les apprenants	119
4.2. Intégration de l’outil d’évaluation dans la plateforme analyses et discussion...	120
4.2.1. Résultat et analyses pour les apprenants.....	120
4.2.1.1. Pré-test (Avant l’intégration de l’outil d’évaluation dans la plateforme...	120
4.2.1.2. Post-test (Après l’intégration de l’outil d’évaluation dans la plateforme).	121
.....	121
4.2.2. Résultat et analyses pour les enseignants assistants de TP.....	122
5. Conclusion	123

Conclusion et Perspectives	124
-----------------------------------	------------

1. Bilan	124
2. Perspectives.....	127
A. Perspectives expérimentales	127
B. Perspectives de recherche.....	127

Bibliographie	128
----------------------	------------

Annexe	140
1. Annexe A.....	140
2. Annexe B.....	148
Notations	150
Travaux réalisés	153

Liste des figures

Chapitre 01 : Les EIAH et les TéléTPs

Figure 1.1 : Relation entre les différents modes de formations [Benmouhamed,2007]	09
Figure 1.2 : Les acteurs d'une plateforme d'apprentissage.....	12
Figure 1.3 : Page d'accueil d'un cours dans Claroline - mode de vue étudiant [Dossou et al. 2008].....	15
Figure 1.4 : La page d'accueil du cours dans la plateforme moodle [Oubahssi, 2014].....	16
Figure 1.5 : Page d'accueil d'un cours sur Sakai [Oubahssi, 2014].....	18
Figure 1.6 : Les différentes interactions dans un TP en présentiel [Benmohamed, 2007].....	23
Figure 1.7 : Typologie des systèmes de TP adaptée et inspiré de [[Benmohamed, 2007] [Bouabid, 2013].....	25
Figure 1.8 : Phases de déroulement d'un TP adapté et inspiré de [Gateau et al., 2007].....	26

Chapitre 02 : Évaluation des apprenants dans les EIAH

Figure 2.1 : Correspondances entre objectifs d'apprentissage et outils d'évaluation.	39
Figure 2.2 : Relation entre L'évaluation et la session d'apprentissage.....	43
Figure 2.3 Le principe de CAT [Durand, 2006].....	45
Figure 2.4 : Les trois composants de logiciel Pépite [Delozanne, 2002].....	46
Figure 2.5 : Le logiciel GenEval [Durand, 2006].....	48
Figure 2.6 : L'interface de la plateforme TEB [Tanana, 2009].....	49
Figure 2.7 : OASYS : interface de correction [Durand, 2006].....	50

Chapitre 03 : Techniques d'Appariement de Graphes et Calcul de Similarité

Figure 3.1 : Un graphe non-étiqueté non-orienté.....	56
Figure 3.2 : Un graphe étiqueté non-orienté.....	56
Figure 3.3 : Un graphe non-étiqueté orienté.....	56
Figure 3.4 : Un graphe étiqueté orienté.....	56

Figure 3.5 : Le processus d'appariement adapté et inspiré de [Auxepaules, 2009]...	58
Figure 3.6 : Exemple d'isomorphisme de graphe inspiré et adapté de [Abbaci,2013]	61
Figure 3.7 : Exemple d'isomorphisme de sous graphe [Abbaci, 2013].....	62
Figure 3.8 : Exemple de graphes similaires [Abbaci, 2013].....	63
Figure 3.9 : Le plus grand sous graphe commun [Tanana,2009].....	64
Figure 3.10 : Différents types d'apparieurs.....	69
Figure 3.11 : Implémentation d'une classe en C++.....	71
Figure 3.12 : Implémentation d'héritage en C++.....	71
Figure 3.13 : Implémentation d'agrégation en C++.....	71
Figure 3.14 : Implémentation de composition en C++.....	72
Figure 3.15 : Implémentation d'un diagramme de communication en C++.....	72

Chapitre 04 : La programmation orientée objet: un domaine d'application pour l'apprentissage des activités de TéléTPs

Figure 4.1 : la tâche de la programmation [Duchâteau a ,1992].....	76
Figure 4.2 : la décomposition de l'activité de la programmation [Guittet et al., 2005]..	76
Figure 4.3 : Distances séparant l'ordinateur et la tâche qu'il souhaite accomplir, selon [Norman ,1990].....	79
Figure 4.4 : l'activité de la programmation : une source de difficulté.....	81
Figure 4.5 : L'intérêt de faire un TP à distance.....	83
Figure 4.6 : La tâche d'édition de code.....	84
Figure 4.7 : La tâche de compilation.....	84
Figure 4.8 : La tâche de correction des erreurs.....	84
Figure 4.9 : La tâche d'exécution.....	85
Figure 4.10 : Les réactions des apprenants devant les tâches difficiles.....	85
Figure 4.11 : l'évaluation préférée chez les apprenants.....	86
Figure 4.12 : Les préférences des apprenants pour la réalisation des quatre tâches de la programmation.....	86
Figure 4.13 : L'utilité d'utilisation d'un module d'évaluation dans la plateforme....	87

Chapitre 05 : Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

Figure 5.1 :L'architecture de la plateforme E-TéléTPC@AALP [Boussaha,2009]....	94
Figure 5.2 : Méthodologie de l'approche proposée [Boussaha et al. , 2012].....	101
Figure 5.3 : L'algorithme d'appariement proposé [Boussaha et al.,2015].....	106

Chapitre 06 : Environnement d'évaluation de l'apprenant

Figure 6.1 : L'architecture générale de l'outil d'évaluation.....	110
Figure 6.2 : Programme de l'enseignant.....	111
Figure 6.3 : Programme de l'apprenant.....	111
Figure 6.4: Diagramme de classe de l'enseignant.....	112
Figure 6.5: diagramme de classe de l'apprenant.....	112
Figure 6.6: Diagramme de communication de l'enseignant.....	113
Figure 6.7: Diagramme de communication de l'apprenant.....	113
Figure 6.8: l'ensemble des nœuds et des arcs étiquetés dans le diagramme de classe de l'enseignant et de l'apprenant.....	114
Figure 6.9: l'appariement m généré à partir de diagramme de classe de l'enseignant	114
Figure 6.10: Mesure de similarité entre les deux diagrammes de classe (enseignant, apprenant) par apport à l'appariement m.....	115
Figure 6.11: Mesure de similarité totale entre les deux diagrammes de classe (enseignant, apprenant).....	115
Figure 6.12: L'ensemble des nœuds et des arcs étiquetés dans le diagramme de communication de l'enseignant et de l'apprenant.....	116
Figure 6.13: L'appariement m' généré à partir de diagramme de communication de l'enseignant.....	116
Figure 6.14: Mesure de similarité totale entre les deux diagrammes de communication (enseignant, apprenant).....	117
Figure 6.15: Mesure de similarité globale entre les deux programmes (enseignant, apprenant).....	118
Figure 6.16: La liste des erreurs détectées dans le programme de l'apprenant.....	119
Figure 6.17: Les résultats avant l'intégration de l'outil d'évaluation dans la plateforme.....	121
Figure 6.18: Les effets de l'outil pour la maîtrise des concepts et la résolution des situations de blocage pour les apprenants.....	122
Figure 6.19: Les effets de l'outil pour diminuer le nombre de demande d'interventions en ligne des assistants de TP.....	123

Annexe : Annexe A

Figure A.1 : L'intérêt de faire un TP à distance.....	144
Figure A.2 : La tâche d'édition de code.....	144
Figure A.3 : La tâche de compilation.....	144
Figure A.4 : La tâche de correction des erreurs.....	145
Figure A.5 : La tâche d'exécution.....	145
Figure A.6 : Les réactions des apprenants devant les tâches difficiles.....	145
Figure A.7 : Les modes de communication préférés dans un TéléTP.....	146
Figure A.8 : L'évaluation préférée chez les apprenants.....	146
Figure A.9 : Les préférences des apprenants pour la réalisation des quatre tâches de la programmation.....	147
Figure A.10 : l'utilité d'utilisation d'un module d'évaluation dans la plateforme...	147

Liste des tables

Chapitre 01 : Les EIAH et les TéléTPs

Table 1.1 : Récapitulatif des fonctionnalités de systèmes LMS et LCMS et la relation existante entre eux.....	11
Table 1.2 : Les avantages et les inconvénients des Classes des plateformes.....	14
Table 1.3 : Récapitulatif des fonctionnalités et services communs fournis par la plateforme Moodle /Claroline.....	17
Table 1.4 : Étude comparative entre les plateformes [Dossou et al. ,2008].....	20

Chapitre 02 : Évaluation des apprenants dans les EIAH

Table 2.1 : Comparaison entre des différentes formes de l'évaluation inspirée de [Durand, 2006] et étendue.....	42
Table 2.2 : Les types d'évaluation dans les EIAH et les EIAH dédiées aux TéléTPs inspirée de [Durand, 2006] et étendue.....	52

Chapitre 04 : La programmation orientée objet: un domaine d'application pour l'apprentissage des activités de TéléTPs

Table 4.1 : Le résultat de module programmation orientée objet(POO) et le taux de réussite.....	80
Table 4.2 : Comparaisons de taux de réussite de programmation orientée objet(POO) par rapport d'autres modules.....	80

Chapitre 06 : Environnement d'évaluation de l'apprenant

Table 6.1 : T-test Avant l'intégration de l'outil d'évaluation dans la plateforme.....	120
Table 6.2 : T-test après l'intégration de l'outil d'évaluation dans la plateforme....	122

ملخص

تقييم الطلبة , الموضوع المعالج في هذه الأطروحة, يعتبر من أهم مواضيع البحث في بيئات التعلم عن بعد بصفة عامة و بيئات التعلم عن بعد الموجهة لتعلم الأعمال التطبيقية بصفة خاصة, بالنظر إلى الأهمية البيداغوجية للتقييم أثناء كل فترة تعلم فهي تتموقع وسط كل فترة تعلم لأنها الوسيلة الوحيدة التي تضمن مرور المعلومات إلى المتعلمين .

لكن للأسف جل بيئات التعلم عن بعد لم يأخذوا بعين الاعتبار التقييم كوظيفة أساسية أثناء فترة التعلم وإن وجد يكون على شكل تمارين بسيطة تعرض على طريقة الأسئلة الاختيارية هذا لأن إنشاء و إدماج التقييم كوظيفة في بيئات تعلم الأعمال التطبيقية عن بعد يعتبر من أصعب المهام المنوطة بالمبرمجين لهذه الأخيرة .

مجال تطبيقنا هو تطبيقات البرمجة عن بعد التي تعتبر منذ نشأتها صعبة للإلقاء بالنسبة للأساتذة, ومستعصية الفهم والاستيعاب بالنسبة للمتعلمين وهذا يرجع إلى الخصائص الجوهرية التي تميز هذا المجال التطبيقي والطرق القديمة المنتهجة لتعليمها.

هذا العمل العلمي يعتبر نقطة تقاطع بين هذين المحورين فإنه بطبيعة الحال يرث الصعوبات الموجودة في كل محور منهما عاى حد سواء . هذا العمل موجه خصيصا لإنشاء طريقة بيداغوجية لتقييم المتعلمين عن بعد في مجال تطبيقات البرمجة عن بعد عن طريق بيئة تعلم أنشأت خصيصا لهذا الغرض .

منهجيتنا تسمح بتقييم الأعمال التطبيقية للطلبة في ما يخص لغة برمجة معينة عن بعد , الفكرة الأساسية تتمحور حول المقارنة غير المباشرة لبرنامج المتعلم مع برنامج المعلم عن طريق تمثيل كل من البرنامجين بطريقة الرسوم البيانية , الهدف من الفرضية هو إيجاد أوجه الشبه والاختلاف بين البرنامجين .

الكلمات المفتاحية : الأعمال التطبيقية عن بعد, التقييم الذاتي , المهارات , بيئات التعلم , لغات البرمجة .

Abstract

The learners' assessment, the topic addressed by this thesis, is one of the most important research areas in computer environments for human learning (CEHL).

The assessment is at the heart of any learning process. It is the means to regulate learning. It is fundamental because it is the only way to ensure good transmission of knowledge to the learners.

Most of the platforms developed for teaching have little interest in the assessment, they usually incorporate simple generators questionnaire and thus avoid the difficulty of the evaluation and the complexity of its implementation in a CEHL. Moreover, the nature of knowledge to assess (declarative knowledge, procedural knowledge) exponentially increases this difficulty.

Our field of application, object-oriented programming was from birth a difficult discipline for the teacher to teach and complex for the learners to assimilate. This is due to the intrinsic characteristics of the material and the way in which teachers (the majority) teach its foundations.

This research is at the confluence of these two axes and therefore inherits difficulties of both.

Our approach allows to evaluate the learners' practical work C++ objects oriented programs The idea is to make an indirect comparison of two object-oriented programs (learner, teacher) through their graphs described by two UML diagrams (class diagram and communication diagram). Our approach aims to find the similarities and differences between production of the learner and the teacher. The technique of matching, we offer takes into account several parameters.

Keywords: Practical works, self-assessment, e-learning platforms, UML, object oriented programming, C++, learners' skills , learners' assessment.

Résumé

L'évaluation des apprenants, le sujet traité par cette thèse, est l'un des plus importants axes de recherche dans les environnements informatiques pour l'apprentissage humain(EIAH).

L'évaluation se situe au cœur de n'importe quel processus d'apprentissage. Elle est le moyen de réguler et d'améliorer l'apprentissage. Elle joue un rôle fondamental, car c'est le seul moyen d'assurer un bon transfert des connaissances à l'apprenant.

La plupart des plateformes développées pour l'enseignement s'intéressent peu à l'évaluation des apprenants, elles intègrent généralement des générateurs simples de questionnaire et évitent ainsi la difficulté de l'évaluation et la complexité de sa mise en œuvre dans un EIAH. De plus, la nature des connaissances à évaluer (connaissances déclaratives, connaissances procédurales) accroît exponentiellement cette difficulté.

Notre domaine d'application, la programmation orientée objet, a été depuis sa naissance une discipline un peu difficile à enseigner pour l'enseignant et complexe à assimiler pour l'apprenant. Cela est dû aux caractéristiques intrinsèques de la matière et à la manière avec laquelle les enseignants (la majorité) enseignent ses fondements.

Ce travail de recherche s'inscrit au confluent de ces deux axes, et par conséquent, hérite des difficultés des deux. Il vise plus précisément à élaborer une approche d'évaluation des compétences en programmation orientée objet dans un EIAH consacré à l'apprentissage des travaux pratiques à distance.

Notre approche permet d'évaluer les travaux pratiques des apprenants sur le développement de programmes C++ orientés objets. L'idée est de procéder à une comparaison indirecte de deux programmes orientés objets (apprenant, enseignant) à travers leurs représentations graphiques décrites par deux diagrammes UML (diagramme de classes et diagramme de communication). Notre approche a pour objet de trouver les similarités et les différences entre la production de l'apprenant et celle de l'enseignant. La technique d'appariement que nous proposons tient compte de plusieurs paramètres.

Mots clés : TéléTPs, Auto-Évaluation, les Plateformes, UML, Programmation Orientée Objet, C++. Compétences des apprenants, l'évaluation des apprenants.

Introduction générale

L'apprenant, pour lequel on recherche à améliorer et à progresser le processus de son apprentissage, et l'enseignant, dont on cherche à simplifier sa tâche d'enseignement, sont deux acteurs et deux objectifs qui sont visés par l'introduction des TIC dans l'éducation (TICE) et les EIAH (Environnements Informatiques pour l'Apprentissage Humain) dans l'enseignement [Tanana, 2009].

Plusieurs EIAH sont conçus pour couvrir ces deux objectifs cités au-dessus et aider ces deux acteurs. Malheureusement, la plus part des auteurs de l'ingénierie des EIAH ne prennent pas en considération l'activité de l'évaluation des apprenants lors de la conception de leurs EIAH.

Cadre et Motivations

Le présent travail se place dans le contexte des EIAH et vise l'évaluation des compétences en résolution des problèmes, dans les activités des TELLÉTPs, chez les apprenants débutants en domaine de la programmation orientée objet.

Il est bien connu que plupart des étudiants trouvent l'apprentissage de l'algorithmique et de la programmation comme concepts difficiles à maîtriser que ce soit dans les séances des TD ou les sessions de TP avec la méthode pédagogique d'enseignement classique en présentiel et souffrent de plusieurs problèmes majeurs techniques et opérationnels [Boussaha, 2009]. En ce qui concerne les problèmes techniques, nous pouvons citer, entre autres, la désinstallation des outils de programmation par le groupe précédent et les problèmes de fonctionnement des PCs, blocage du groupe. Sur le plan opérationnel, plusieurs problèmes, on peut noter la mauvaise compréhension des consignes de travail, la perte du programme réalisé (le cas des étudiants de la première année qui ne savent pas sauvegarder leurs programmes) et l'apparition d'erreurs pendant la tâche de correction des erreurs.

Ces problèmes et d'autres [Boussaha, 2009] influencent directement sur le taux de réussite des étudiants dans ces deux domaines. Notre étude menée dans notre université montre un taux de réussite de 09,16 % (pour plus de détails voir le chapitre 4).

Ce problème d'échec des étudiants dans ces deux domaines est abordé dans tous les instituts et n'est pas limité à notre université, plusieurs études menées aux niveaux national et international ont convergé vers la même conclusion : l'apprentissage de

l'algorithmique / la programmation était toujours une source de difficultés pour les deux acteurs d'apprentissages à la fois (les étudiants et pour les enseignants aussi).

Problématique et Contribution

Face à ces problèmes : insuffisance des méthodes classiques en présentiel dans l'apprentissage de la programmation, le taux d'échec élevé chez les apprenants, la rareté des plateformes d'apprentissage des TélecTPs, et les problèmes d'encombrement des apprenants et son influence sur la longue durée d'attente des enseignants assistants de TPs (soit pour aider l'apprenant à se débloquer soit pour évaluer le TP réalisé par l'apprenant), nous avons commencé de réfléchir d'améliorer l'apprentissage de ce domaine (TP en programmation) depuis l'année 2009 par la conception et le développement de la plateforme (E-TélecTPC@AALP)[Boussaha,2009] dédiée à l'apprentissage à distance des TPs en programmation (dans le cadre de mon projet de recherche de magister) cette plateforme a l'avantage de :

- offrir des TPs à jour à tout moment.
- offrir la possibilité de collaborer avec d'autres personnes ayant différentes idées que celles des groupes de la salle.
- diminuer le problème d'encombrement dans les groupes et compléter les TPs programmés dans le programme.
- éviter le problème d'attente de sollicitation de l'assistant.

Dans le cadre de ce travail de thèse, nous avons amélioré et étendu les services de cette plateforme par la proposition et l'introduction d'un nouveau mécanisme d'évaluation des apprenants.

Le travail élaboré dans cette thèse passe par quatre principales étapes. La première concerne la conception d'un modèle à base des graphes pour la représentation d'un domaine d'enseignement pour un apprentissage des TélecTPs. La deuxième étape consiste à la construction d'une approche d'évaluation des connaissances des apprenants basée sur ce modèle de graphe et sur le concept d'erreur. Cette évaluation est un processus qui consiste à détecter les erreurs de l'apprenant à travers un mécanisme de feedback. La troisième étape présente des applications de cette approche à un domaine d'enseignement spécifique (les TélecTPs dans la programmation orientée Objet) qui est un module de l'enseignement de base des informaticiens. Pour ce domaine, un prototype est développé et des expérimentations sont réalisées avec des étudiants pour étudier l'importance et les

apports du système réalisé. La quatrième et la dernière étape, c'est l'intégration de cet outil dans notre plateforme E-TéléTPC@AALP consacrée à l'apprentissage des TéléTPs avec la réalisation des expérimentations afin de déterminer la portée du système réalisé par rapport à l'apprentissage et l'évaluation classiques des TP (en présentiel).

Organisation du manuscrit

Cette thèse est organisée autour de six chapitres.

Le premier chapitre est consacré à un large tour d'horizon, et un état de l'art sur les environnements informatiques pour l'apprentissage humain en prenant comme exemple les environnements informatiques pour l'apprentissage humain des travaux pratiques à distance.

Dans le deuxième chapitre, nous avons mis l'accent sur le côté théorique de l'évaluation comme une activité pédagogique distincte dans un processus d'apprentissage ou de formation. Nous nous intéressons ensuite, à l'évaluation dans les environnements informatiques pour l'apprentissage humain et l'évaluation des apprenants en utilisant les nouvelles technologies. Nous définissons quelques notions et concepts liées à l'évaluation dans les EIAH et l'évaluation dans les EIAH dédiés à l'apprentissage des TéléTPs. Nous achevons ce chapitre par l'introduction d'un travail de synthèse portant sur un ensemble de systèmes dédiés à l'apprentissage et à l'évaluation des productions des apprenants

Le troisième chapitre présente le concept de graphes et introduit la possibilité de son utilisation en tant que formalisme de représentation de connaissances dans les systèmes EIAH dédiés à l'apprentissage des TéléTPs.

Une étude épistémologique de la programmation orientée objet, notre domaine d'application, est présenté dans le quatrième chapitre afin de mieux cerner les Différentes caractéristiques de ce champ d'application.

Dans le cinquième chapitre, nous présentons notre contribution fondée sur une modélisation d'une solution de TP avec les graphes, un algorithme d'appariement et une mesure de similarité sont proposés pour répondre à nos besoins d'évaluation d'une production de l'apprenant.

Dans le sixième chapitre, l'implémentation de cette approche donne naissance à un environnement d'évaluation qui est un système d'évaluation des compétences en programmation orientée objet. L'intégration de cet environnement dans la plateforme (E-TéléTPC@AALP), La validation de celui-ci et la discussion des résultats obtenus sont aussi présentées.

Introduction générale

Nous terminerons cette thèse par une conclusion générale, mettant l'accent sur nos contributions majeures. Enfin, en ouvrant d'autres perspectives de recherche.

CHAPITRE 1

Les EIAH et les TélÉTPs

1. Introduction

Le développement rapide de la technologie Internet et sa popularité croissante a eu un impact énorme sur le processus d'enseignement/apprentissage. Actuellement, l'apprentissage par Internet est devenu important dans l'enseignement.

Depuis des années, nous avons vu l'émergence de plusieurs développements dans les domaines pédagogiques ce qui entraîne l'émergence de nouvelles technologies. L'utilisation de l'ordinateur dans le domaine de l'éducation n'est pas un phénomène nouveau et a abouti à l'émergence du terme EAO (Enseignement Assisté par Ordinateur) dans les années 60. L'intégration des techniques d'intelligence artificielle à l'EAO a engendré l'apparition, dans une première acception, de l'EIAO (Enseignement Intelligemment Assisté par Ordinateur).

Puis une deuxième notion a vu le jour qui mettait l'accent sur l'importance fondamentale de l'interactivité des systèmes, il s'agit des *Environnement Interactif d'Apprentissage avec Ordinateur*. Plus récemment, c'est le terme **EIAH** (*Environnement Informatique pour l'Apprentissage Humain*) qui a remplacé les termes précédents [Benmohamed, 2007].

Ce chapitre est consacré à la définition du contexte général de notre travail : les EIAH pour les travaux pratiques à distance. Le chapitre est organisé en deux parties :

La première partie est dédiée à la présentation de la chronologie des EIAH ainsi à la définition des concepts concernant quelques modes de formation afin de situer notre recherche vis-à-vis de ces modes.

La seconde partie est réservée pour l'étude des environnements d'apprentissages des TP/TélÉTPs. Nous concluons cette partie par une synthèse qui permet de dégager notre problématique liée aux EIAH dédiées à l'apprentissage des TélÉTPs.

PARTIE 1

2. Les EIAH

2.1. Définition

Un EIAH au sens large « *est un environnement qui intègre des agents humains (apprenants, enseignants, tuteurs), et artificiels (agents informatiques, qui peuvent, eux aussi, tenir différents rôles) en leur offrant des conditions d'interactions, localement au à travers les réseaux informatiques, ainsi que des conditions d'accès à des ressources formatives (humaines et/ou médiatisées), locales ou distribuées* » [Techounikine et al., 2004]. Ce type d'environnement est conçu dans le dessein de favoriser *l'apprentissage humain*.

Afin de situer nos travaux dans ce domaine de recherche (les EIAH) nous préférons de donner un aperçu des étapes marquantes de l'e-formation, or notre objectif n'est pas ici de retracer l'histoire des EIAH.

2.2. Terminologies concernant quelques modes de Formations

Quelques terminologies qui ont participé à l'apparition de nouvelles formes de formation : l'e-formation et la *téléformation* sont les objets de discussion dans la section suivante.

2.2.1. La FOAD : « Formation Ouverte et A Distance »

La formation ouverte et à distance est un type d'enseignement relativement récent (apparu dans les années 1990). L'adaptation à l'individu et la modularité de la formation, l'intégration des technologies de l'information et de la communication sont des contraintes principales exigées dans la FOAD comme l'indique son acronyme [Benmohamed, 2007] [Frayssinhes, 2011]. Elle se caractérise par un dispositif de formation fondé sur une prise en compte des besoins des apprenants, articulant les contenus de formation à des services variés (tutorat, forum, exercices ou simulation), libérant des contraintes de lieux et de moment. *La FOAD* possède deux qualificatifs [Frayssinhes, 2011]:

- « **ouvert** » : le qualificatif «ouvert» désigne l'introduction des notions supplémentaires. Il caractérise des situations qui offrent une plus grande accessibilité, une plus grande souplesse ou flexibilité dans leur mode d'organisation pédagogique que ce soit pour les apprenants ou les enseignants [Frayssinhes, 2011].

- « à distance » : la distance, c'est-à-dire la dispersion géographique et /ou la différence temporelle des acteurs participants dans le processus de formation.

Il y a deux sortes de formation ouverte et à distance :

- **FOAD synchrone** : en temps réel, aussi appelé « *classe virtuelle* ».
- **FOAD asynchrone** : en différé, favorisant l'autonomie des apprenants.

L'UNESCO¹ caractérise les formations ouvertes par « *une liberté d'accès aux ressources pédagogiques mises à disposition de l'apprenant, sans aucune restriction à savoir : absence de conditions d'admission, itinéraire et rythme de formation choisis par l'apprenant selon sa disponibilité de conclusion d'un contrat entre l'apprenant et l'institution* » ; qui prend en compte la singularité des personnes dans leurs dimensions individuelle et collectives. Elle repose sur des situations d'apprentissage complémentaires et plurielles en termes de temps, de lieux de médiations pédagogiques humaines et technologiques, et de ressources [Panagiotakopoulos et al., 2003].

La « commission européenne- initiative e-learning » définit la **FOAD** comme suit: « *l'utilisation des nouvelles technologies du multimédia et de l'Internet afin d'améliorer la qualité de l'éducation et de la formation à travers l'accès à distance à des ressources et des services, ainsi qu'à des collaborations et des échanges* ».

2.2.1.1 Utilisation de FOAD

Bien qu'il soit particulièrement bien adapté aux cours théoriques ou à la conduite de projets, ce type d'enseignement ne semble pas convenir aux travaux pratiques nécessitant l'emploi d'un matériel spécifique, tel qu'il est utilisé dans l'industrie.

Ce mode de formation répond aux attentes de trois catégories sociales :

- ✓ les étudiants de l'enseignement supérieur,
- ✓ le public à mobilité réduite de tout niveau,
- ✓ les personnes éloignées des lieux de formation.

2.2.2. La FAD

La **FAD (Formation A Distance)** peut être simplement définie par : « *toute situation éducative dans laquelle la transmission du savoir ou l'activité d'apprentissage est faite hors de la situation présentielle entre les différents acteurs, la notion de « distance »* »

¹ UNESCO : *United Nations Educational, Scientific and Cultural Organization*, L'Organisation des Nations unies pour l'éducation, la science et la culture est une institution spécialisée de l'Organisation des Nations unies créée le 16 novembre 1945.

repr sente une dispersion g ographique ou temporelle des participants» [Benmohamed, 2007].

Les formations   distance rev tent naturellement de nombreux aspects de niveaux de diff rentes qualit s, mais on peut le plus souvent y relever les caract ristiques communes suivantes : une **FAD** est :

-une formation souple et individualis e : Perriault dans [Perriault, 1997] affirme que la formation   distance offre des degr s de libert  en plus, par rapport   la formation en pr sentiel dans le sens de la souplesse dans le mode d'organisation p dagogique, donc selon les choix p dagogiques des formateurs, les contenus sont accessibles   tout moment, ils sont souvent organis s en modules o  le choix d termine le parcours de l'apprenant.

-un d veloppement de l'autonomie et du travail d' quipe: Ouerfelli dans [Ouerfelli, 2004] d clare que le processus d'apprentissage repose essentiellement sur l'interaction dans un sens transitive entre les acteurs de la FAD qu'il s'agisse des formateurs, des tuteurs, ou des stagiaires, dans ce sens, le processus d'apprentissage assur  par la FAD est autant individuel que collectif selon une n gociation individuelle et sociale. On parle alors de *communaut s d'apprentissage* ou de *communaut s virtuelles*. Cependant, chacun doit g rer sa part personnelle d'apprentissage et ses contributions   la production commune.

-la garantie d'un accompagnement : dans une formation   distance les formateurs sont charg s d'encadrer les apprenants, ces derniers b n ficient du soutien du tuteur en ligne qui suivent leur appropriation des contenus et qu'ils peuvent interpeller   tout moment de leur parcours [Benaily, 2000].

-un dispositif  tay  par plusieurs familles d'outils en ligne cette caract ristique permet de mettre   la disposition des apprenants et des formateurs les services n cessaires aux deux types de phases **asynchrones** et **synchrones** de la formation [Benaily, 2000] :

-pour les phases asynchrones, on utilise une *plateforme* de formation en ligne : *affichage de contenus, partage de fichiers, forums, messagerie, suivi des parcours* ;

-pour les phases synchrones, on utilise une *classe virtuelle* qui permet aux apprenants de participer   des r unions de formation en ligne accompagn s de leurs formateurs et de leurs tuteurs. Les *classes virtuelles* sont utilis es pour les *travaux pratiques* ou des * tudes de cas* et pour un nombre beaucoup plus restreint, une dizaine de participants.

2.2.3. La e-formation

  l'origine, la *e-formation* « e :  lectronique » est la formation par la voix  lectronique ». Elle peut  tre *locale* (produits p dagogiques pr sents sur CD-ROM, disque dur...) ou   *distance* [Benmohamed, 2007], [Mbala-Hikolo,2003],[FOAD, 2002].

2.2.4. La T l formation

Consid r e comme   un sous-ensemble de la *FAD*, la t l formation est un dispositif de formation   distance s'appuie sur l'utilisation des TICE² via Internet, intranet ou extranet » [Mbala-Hikolo, 2003]. Ce terme n'indique pas n cessairement que le dispositif soit ouvert.

Son but est donc de fournir un *enseignement /apprentissage*   tout individu ou groupe d'individus distants de l'organisme prestataire du service de formation selon des modalit s simples adapt es au mieux aux possibilit s de chacun des individus en formation » [Benmohamed, 2007][FOAD, 2002].

2.2.5. Situation de T l TPs

Les travaux de recherche sur les T l TPs qui constituent le contexte de notre travail, se situent dans le cadre des t l formations.

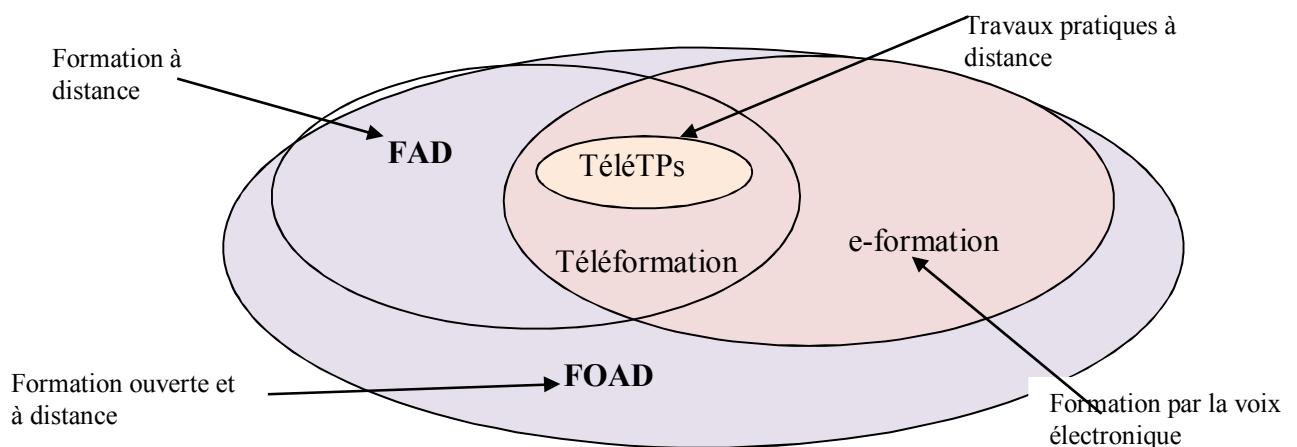


Figure 1.1 : Relation entre les diff rents modes de formations [Benmouhamed, 2007].

² TICE est l'acronyme cr e par le minist re de l' ducation Nationale pour Technologies de l'Information et de la Communication Educatives. Les TICE d signent l'ensemble des technologies utilis es dans le traitement et la transmission des informations, auxquelles les enseignants font appel dans l'enseignement et l' ducation.

2.3. Les syst  mes LMS et LCMS

Les plateformes de t  l  formations s'articulent autour de deux syst  mes : **le LMS** « *Learning Management System* » et le **LCMS** « *Learning Content Management System* ».

a. Le LMS : est un site web qui h  berge du contenu didactique et facilite la mise en   uvre de strat  gies p  dagogiques [Kerschenbaum, 2002]. La plupart de ces syst  mes pr  sentent   galement des g  n  rateurs internes des tests d'  valuation que l'on retrouve sous forme de **QCM**, **vrai /faux**, **texte    trous**, appariements.

Ces activit  s sont soit soumises    validation par l'enseignant soit propos  es comme activit  s de r  gulation en auto-  valuation.

b. LCMS: Applications qui permettent de cr  er, valider, publier, et g  rer des contenus d'apprentissage [Kerschenbaum, 2002].

Il est    noter qu'il existe une diff  rence fondamentale entre un **LMS** et un **LCMS**.

LMS : gestion de la formation, des r  sultats, diffusion de contenus pr   tablis, ordonnancement de modules de formation.

LCMS : toutes les capacit  s du **LMS**, auxquelles on doit ajouter la capacit   de g  n  rer des contenus de formation nouveaux,    partir d'objets d'apprentissages qui   voluent en quantit   et en qualit  .

Dans la table 1.1 on r  sume toutes les fonctionnalit  s discut  es dans [Kerschenbaum, 2002],[Mbala-Hikolo,2003],[EdD.S.I, 2005], de chacun des deux syst  mes ainsi que la relation existante entre eux.

	LMS	LCMS
Fonctionnalit��s	<ul style="list-style-type: none"> -h��berger le contenu p��dagogique selon diff��rents formats (textuel et multim��dia), -contr��ler l'acc��s aux ressources, -offrir des activit��s p��dagogiques, -faciliter les activit��s de tutorat et de pilotage de la formation, 	<ul style="list-style-type: none"> - cr��er le contenu, -Stocker le contenu, -Diffuser le contenu d'une fa��on dynamique, -valider de contenu -g��rer le contenu - publier le contenu

	-faciliter le pilotage des ressources de l'organisme de formation (gestion des formateurs, des moyens logistiques et techniques), - g��rer la communaut�� d'apprenants, - permettre la gestion administrative des documents associ��s �� la formation (attestations de formations par exemple),	
Relation LMS/LCMS	<i>LCMS=LMS+CMS.</i>	

Table 1.1 : R  capitulatif des fonctionnalit  s de syst  mes LMS et LCMS et la relation existante entre eux

2.3.1. Les Acteurs d'une plateforme d'apprentissage

Selon Taziri dans [Taziri, 2005], on distingue plusieurs types d'acteurs (figure 1.2):

- **Administrateur de la plateforme** qui g  re la plateforme pour mettre en place les modules, installe et assure la maintenance de syst  me.
- **Enseignant** : qui cr  e les modules, anime les sessions en ligne, et effectue le suivi de l'apprenant.
- **Apprenant** : qui suit la formation en consultant les modules et participants aux activit  s p  dagogiques en ligne.
- **Tuteur** qui encadre les apprenants. Ses droits sont plus limit  s que celui de l'enseignant.
- **Invit   (ou visiteur)** qui ne dispose pas d'identifiants. Il peut acc  der aux cours publics de la plateforme.



Figure 1.2 : Les acteurs d'une plateforme d'apprentissage [Dossou et al. , 2008]

2.3.2. Points cl  s pour choisir une plateforme d'apprentissage

Comme il est convenu, une plateforme ne se limite pas aux d  p  ts de documents et    un travail collaboratif simple : communication entre (apprenant, enseignants) et (apprenant, apprenants). Choisir une plate-forme adapt  e aux besoins de l'ing  nierie p  dagogique est une   tape importante pour l'optimisation du dispositif et l'atteinte des objectifs. Ce choix peut ainsi   tre conditionn   par des crit  res p  dagogiques et autres techniques.

2.3.2.1. Crit  res p  dagogiques

Une plateforme doit, en effet, r  pondre aux besoins p  dagogiques du plus grand nombre en termes de m  thodes p  dagogiques possibles et d'outils num  riques utilisables. Plus particuli  rement, elle doit [Dossou et al. , 2008] :

-   tre simple d'utilisation, avec une prise en main rapide et facile de mani  re      tre accessible    la plupart des enseignants.
-   tre utilisable par la majorit   des enseignants
- permettre la gestion des parcours p  dagogiques tels que la gestion des individus (les enseignants et les apprenants), la gestion de groupes et la gestion des contenus de formation.
- donner la possibilit   de relier certaines fonctionnalit  s (par exemple, offrir un tableau de bord des rendus de devoirs de chaque individu, masquer ou rendre visible rapidement un document ou un ensemble de documents, passer facilement d'une vision enseignant    une vision apprenant, donner la possibilit   d'adresser les m  mes

contenus   diff rentes sessions, cr er des quiz not s, effectuer des statistiques fines, manipuler des forums  volu s...).

- offrir une ergonomie et utilisabilit  des plateformes pour les enseignants comme pour les apprenants.
- assurer des besoins d'industrialisation (duplication, affectation de documents).
- r pondre   des besoins de capitalisation des formations (organisation "ferm e" et structur e sur une formation) en int grant toutes les fonctions de gestion de groupes et/ou d'individus dans la dur e.

2.3.2.2. Crit res techniques

Les crit res techniques d'une plateforme d'apprentissage sont essentiellement :

- La capacit  de g rer un grand nombre d'utilisateurs [Kurilovas, 2000].
- L'int gration de sp cifications techniques et de standards comme les normes SCORM, AICC et LOM et des contenus issus de logiciels de cr ation de contenus comme Questionmark, HotPotatoes [Kurilovas, 2000].

2.3.3. Classification des plateformes d'apprentissage

Selon la classification qui a  t  faite par Dossou, dans [Dossou et al. ,2008], les plateformes d'apprentissage   distance peuvent  tre class es en trois grandes familles:

- **Plateforme maison** : plateforme d velopp e par une institution.
- **Plateforme libre** : plateforme d velopp e par une communaut  dont le code source est accessible   tous les utilisateurs.
- **Plateforme propri taire** : plateforme commerciale d velopp e par une entreprise.

La table 1.2 suivante montre les avantages et les inconv nients de ces trois familles.

Famille	Avantages	Inconv�nients
Plateforme maison	r�ponse aux besoins r�els de l'institution connaissance du code institution symbolique d�veloppement assur� � l'interne	co�t de d�veloppement difficult� � avoir une apparence professionnelle probl�me � exporter l'outil
Plateforme libre	acc�s au code source licence permettant la diffusion libre de ces ressources. co�ts r�duit	dynamisme de la communaut� fiabilit� d�veloppement futur
Plateforme propri�taire	garantie sous forme d'une licence soutien technique fourni par la compagnie	acc�s interdit au code source qualit� du soutien technique fournie co�ts �lev�

Table 1.2 : Les avantages et les inconv nients des Classes des plateformes [Dossou et al. , 2008]

3. Quelques plateformes d'apprentissage

Il existe, dans la litt rature, plus de 250 plateformes d'apprentissage   distance dont une trentaine sous licences libres. Nous pr sentons dans ce qui suit quelques-unes.

3.1. Claroline

3.1.1. G n ralit s

Claroline³ (figure 1.3) est une plateforme open source d'e-Learning d velopp e par l'Universit  Catholique de Louvain-la-Neuve (UCL, Belgique), en 2001 sous licence Open Source(GPL⁴). Elle permet   un formateur de cr er des espaces de cours en ligne et de g rer des activit s de formation sur Internet. Comme elle permet aux apprenants d'assister aux activit s de formation attendues. Claroline est r alis e gr ce   des technologies libres comme PHP et MySQL et utilise des standards comme SCORM⁵ et

³ Claroline : <http://www.claroline.net>

⁴ GPL : *General Purpose License*

⁵ SCORM : **S**harable **C**ontent **O**bject **R**eference **M**odel : est une sp cification de codage permettant de cr er des objets p dagogiques structur s. Visant   r pondre   des exigences d'accessibilit , d'adaptabilit , de durabilit , d'interop rabilit  et de r utilisabilit , les normes du mod le informatique SCORM cherchent   faciliter les  changes entre plates-formes de formation en ligne.

IMS/QTI⁶ pour les  changes de contenu. Elle pr sente de nombreuses similitudes dans ses fonctionnalit s et ses services avec la plateforme « Moodle » [Dossou et al., 2008]. Ses fonctionnalit s et services communs sont r sum s dans la table 1.3.

Claroline s'appuie sur une communaut  mondiale d'utilisateurs et de d veloppeurs (plus de 80 pays). Elle est disponible en 35 langues.

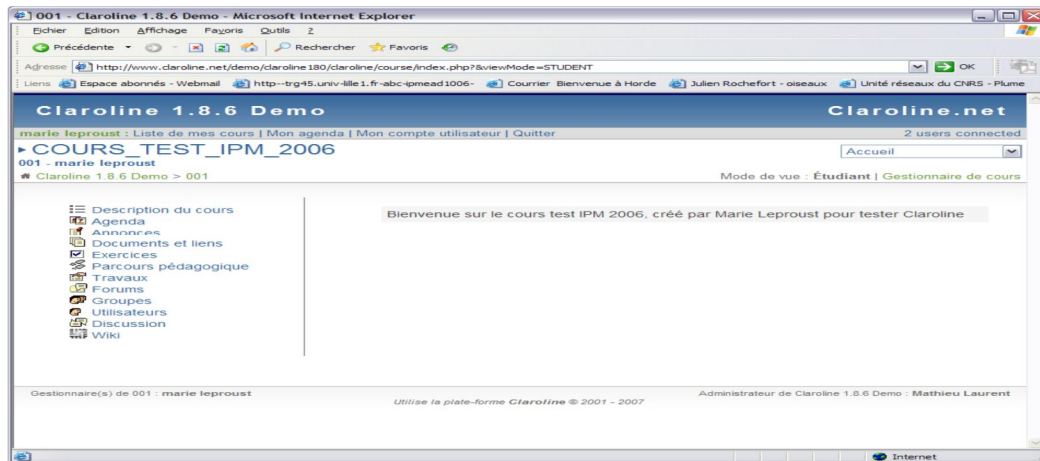


Figure 1.3 : Page d'accueil d'un cours dans Claroline-mode de vue  tudiant [Dossou et al., 2008]

3.1.2. Caract ristiques

La plateforme [Claroline, 2007] Claroline est :

- **simple** : facile   installer et   utiliser au quotidien. Claroline ne n cessite aucune comp tence technique priv e. Le formateur peut s'approprier l'outil dans un temps limit  et s'investir ainsi davantage dans la qualit  des p dagogies d velopp es. [Claroline, 2007].
- **vivante** : les  quipes de d veloppement de Claroline font  voluer la plateforme en accord avec les besoins et les contextes d'utilisation.
- **efficace** : la plateforme offre   l'utilisateur un environnement de travail flexible et sur mesure. Et elle implique l'apprenant dans sa formation, tout en lui laissant une grande autonomie dans l'organisation de son travail.

⁶ IMS/QTI :IMS Question and Test Interoperability c'est une norme de repr sentation des questions et des r ponses.

3.2. Moodle

Moodle (Modular Object Oriented Dynamic Learning Environment) est une plateforme d'apprentissage en ligne sous licence open source servant    cr  er des communaut  s d'apprenants autour de contenus et d'activit  s p  dagogiques [Dossou et al. ,2008] [Oubahssi, 2014].

Consid  r  e comme   tant un outil tr  s performant et riche aux fonctionnalit  s nombreuses, MOODLE⁷ est bas  e sur du code PHP et faite appel    une base de donn  es MYSQL.



Figure 1.4 : La page d'accueil du cours dans la plateforme Moodle [Oubahssi, 2014].

3.2.1. Principales fonctionnalit  s

La plateforme Moodle offre plusieurs outils et fonctionnalit  s communs avec la plateforme Claroline dont les principales sont r  sum  es dans la table 1.3 suivante :

⁷ Moodle: <http://www.moodle.com>

Fonctionnalit� / service	Description	Communication
Espace enseignant concepteur	<ul style="list-style-type: none"> -productions des supports de cours. -�diteur WYSIWYG - interface web. -tests. -Editeur de questionnaire de test -questions � choix multiples, � r�ponse num�riques, � r�ponses courtes -tests limit�s dans le temps (date limite de rentr�e, date de disponibilit�). - param�tres suppl�mentaires (nombre de tentatives autoris�es) 	
Espace enseignant	<ul style="list-style-type: none"> - travail en groupe. -Tutorat. -Communication asynchrone -Communication asynchrone (formes fichiers attach�s) -Communication synchrone (messagerie instantan�e). 	enseignant- enseignant enseignant-Apprenant
Espace apprenant	<ul style="list-style-type: none"> -acc�s au support de cours -espace priv� et personnalisation -communication asynchrone -outil de travail en groupe. -communication synchrone 	apprenant-apprenant. apprenant-apprenant
Discussion direct	messagerie instantan�e.	apprenant-apprenant enseignant –apprenant
Discussion indirect	<ul style="list-style-type: none"> -forum de discussion (publique et priv�s). -outil de travail en groupe. 	apprenant-apprenant apprenant-apprenant
Outils p�dagogiques	<ul style="list-style-type: none"> -tests. -sondages. -consultation p�dagogique. -suivi des r�sultats aux tests. -devoirs- le�ons- ateliers. 	
Administration	<ul style="list-style-type: none"> -inscription et suivi des apprenants. - gestion des dossiers des apprenants -cr�ation de groupes d'utilisateurs 	

Table 1.3 : R capitulatif des fonctionnalit s et services communs fournis par la plateforme Moodle /Claroline

3.3. Sakai

Sakai est une Plateforme p  dagogique modulaire cr  e par un consortium d'universit   et organisme d'enseignement, libre et gratuite. Son architecture permet de r  pondre aux besoins des grandes entreprises et des gouvernements [Oubahssi, 2014].

Le projet SAKAI a   t   lanc  , en 2004, par 4 universit  s am  ricaines avec pour objectif de consolider leurs d  veloppements en mati  re de *plateforme d'apprentissage*. La version 1 est parue en octobre 2004. L'objectif de SAKAI est de produire une plateforme compl  te de gestion de cours libre de qualit     quivalente ou sup  rieure aux autres produits retrouv  s sur le march  .

Le *framework* ainsi que les outils d  velopp  s sont g  n  riques et peuvent   tre utilis  s comme base pour la construction d'outils de collaboration ou d'apprentissage dans n'importe quel domaine.

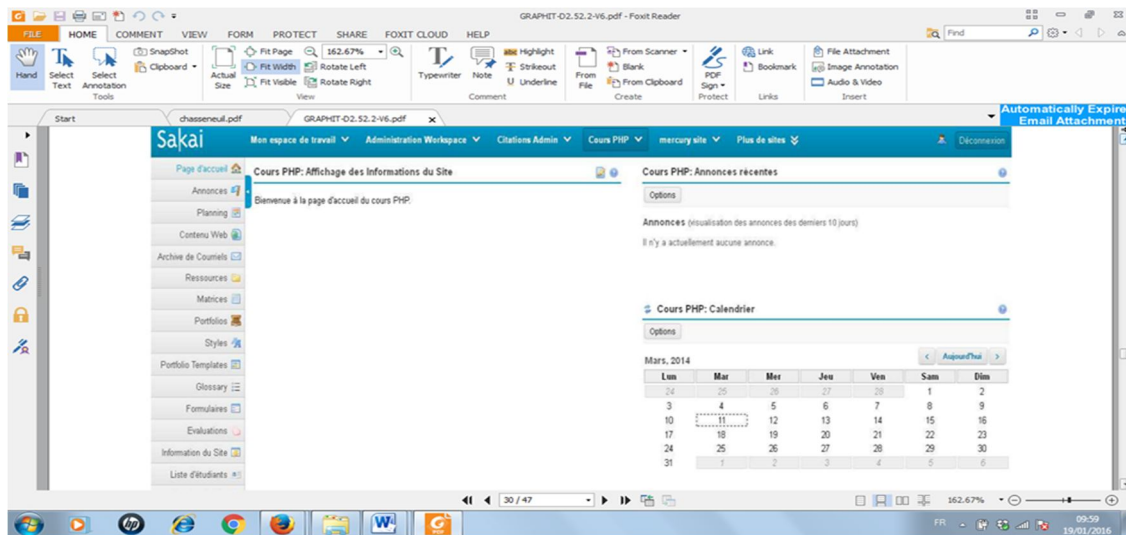


Figure 1.5 : Page d'accueil d'un cours sur Sakai [Oubahssi, 2014]

Sakai se d  compose en trois composants principaux : *les outils*, *les services* et *le framework*. Le framework agit comme le point central de l'application. Son r  le est de g  rer et de r  f  rencer les services et outils. Les services regroupent l'ensemble des op  rations relatives    la plateforme. Il s'agit, entre autres, de l'authentification, de la gestion des utilisateurs, de la gestion des contenus, de l'acc  s aux donn  es et de la gestion des groupes, etc...

Les outils correspondent aux fonctionnalit  s pr  sent  es aux utilisateurs (barres de navigation, textes de pr  sentation, forums de discussion, calendrier...), seuls les outils g  n  rent du contenu graphique. Le contenu qu'ils pr  sentent est accessible via les services [Oubahssi, 2014].

3.3.1. Principales fonctionnalités

Dans ce qui suit, nous citons les principales fonctionnalités de SAKAI regroupées par catégorie :

- ***outils de communication*** (les forums, échange de fichiers, courriel, note et journaux, clavardage, audio-vidéo conférence, tableau blanc, partage d'application, liste des personnes connectées).
- ***outils de productivité*** (signet, aide en ligne, outil de recherche, calendrier et agenda commun).
- ***outils d'administration*** (authentification, autorisation de cours, inscription).
- ***outils d'implication d'étudiants*** (auto-évaluation, port folio).
- ***outils de diffusion de cours*** (Gestions des cours, statistique de suivi des cours, évaluations en ligne, suivi des étudiants, tests et notations automatisés).

4. Étude comparative

Dans cette section, nous présentons une étude comparative (Table 1.4) élaborée par [Dossou et al. ,2008] entre les trois plateformes : Claroline , Moodle et Sakai. Des « smileys » sont utilisés pour illustrer les comparaisons :



: La plateforme répond au critère.



: La plateforme ne répond pas au critère.































Crit��res d'��valuation	Claroline	Moodle	Sakai
1. Communaut��, dynamisme, international			
2. Documentation			
3. Gestion du nombre d'utilisateurs			
4. Outils collaboratifs			
5. Adaptabilit�� et modularit��			
6. Int��gration de sp��cification et standards			
7. Installation et gestion			
8. Adaptation de la charte graphique			
9. Syst��mes d'exploitation, navigateurs, plug-in client			
10. Ergonomie et utilisabilit��			

Table 1.4 :   tude comparative entre les plateformes [Dossou et al. ,2008]

Bien que ces plateformes aient apport   de r  elles avances dans la litt  rature, elles ne traitent pas les travaux pratiques. Elles se fondent, soient sur les T  l  cours soient sur les T  l  TD sans r  elle activit   de la part de l'apprenant. Les T  L  TPs ne sont pas pris en consid  ration. Nous donnons dans le reste de ce chapitre une vue g  n  rale sur l'apprentissage des T  L  TPs.

PARTIE 2

5. Les TélÉTPs

Afin de mettre en pratique les connaissances acquises par l'apprenant ainsi que ses compétences, toutes les informations nécessitent une confrontation avec le monde réel par la pratique dans le domaine [Benmouhamed, 2007].

Le modèle pédagogique français a adopté, le triptyque: cours magistral, TD, TP (enrichi et complété ces dernières années par « le projet »). Ci-après le découpage étudié par [Ouamer, 1992].

a- Le cours magistral

Est une pratique mise en œuvre par un enseignant visant à transmettre des connaissances (savoir, savoir-faire, compétences...) à un élève, un étudiant ou tout autre public dans le cadre d'une institution éducative.

b- Les travaux dirigés

Le petit dictionnaire de français Larousse définit les travaux dirigés comme suit « *Applications concrètes des cours magistraux réalisées en groupes restreints sous forme d'exercices écrits ou d'exposés* ». Donc, l'élève ici est beaucoup plus acteur.

c- Le projet

Le projet est global et non méthodologique. Il ne donne généralement pas accès à des dispositifs techniques.

d- Les travaux pratiques

Ils offrent aux élèves une confrontation entre ce qui a été abordé en cours, TD, et la réalité en proposant des dispositifs concrets où ils peuvent expérimenter leurs connaissances.

6. Les travaux pratiques (TPs)

6.1. Définition

Les Travaux Pratiques sont des activités complémentaires aux parties cours et travaux dirigés. Ce type d'activités peut être mené dans un laboratoire (souvent le cas) ou non (étude et analyse chimique et biologique de la terre dans un champ agricole par exemple). Dans la littérature, on trouve plusieurs définitions des TPs :

Neau [Neau, 2003] a défini les travaux pratiques par « *l'activité qui consiste, pour les étudiants, d'effectuer une ou plusieurs expérimentations, c'est-à-dire des manipulations de dispositifs expérimentaux avec l'objectif de mettre en évidence un phénomène* ». Par contre [Singer, 2005], a proposé la définition suivante « *les travaux pratiques donnent l'occasion pour les étudiants d'interagir directement avec un dispositif expérimental (ou avec des données issues d'un dispositif expérimental) utilisant des outils, des techniques de collecte de données, des modèles et des théories scientifiques, ceci inclut :*

- ✓ *la manipulation d'instruments: TP de chimie, biologie, physique ;*
- ✓ *l'utilisation des simulateurs pédagogiques ;*
- ✓ *l'utilisation et l'accès à des bases de données expérimentales : les étudiants peuvent utiliser des données expérimentales issues d'instruments de mesures et représentées sous diverses formes (images pour l'étude de la lune par exemple). Ces données peuvent être incluses dans des films, DVD, ou programmes informatiques ;*
- ✓ *Les laboratoires distants.*

Enfin, Bouabid dans [Bouabid, 2012], a défini les TP comme suit « *les travaux pratiques dans l'enseignement sont une forme d'activité d'apprentissage qui se fait dans un laboratoire et qui permet à un groupe d'étudiants d'expérimenter les principes théoriques appris en suivant les consignes de l'enseignant* ».

6.2. Les objectifs pédagogiques d'un TP

Les théories socioconstructivistes connues dans l'apprentissage démontrent l'importance des travaux pratiques comme étant un élément favorisant les interactions entre apprenants et apprenants-tuteurs d'un côté et mettant en œuvre le principe de l'apprentissage par essai/erreur d'un autre côté [Meyer et al. 2002].

Mechta dans [Mechta, 2012], cite quelques objectifs pédagogiques derrière les travaux pratiques:

- fournir des illustrations et démonstrations des principes enseignés et donc une meilleure assimilation des apprenants.
- motiver les étudiants et focaliser les interactions entre étudiants et entre étudiant-tuteurs.
- développer des compétences pratiques considérées comme importantes d'un point de vue professionnel.

- d  veloppement des comp  tences du travail collaboratif en   quipe.
- introduire les   tudiants dans la communaut   de pratique des ing  nieurs et des scientifiques.
- donner un contexte pour l'enseignement de l'analyse de donn  es.

6.3. Les interactions dans les TPs

Benmohamed dans [Benmohamed, 2007] explique les diff  rentes interactions dans un TP en pr  sentiel comme le montre la figure 1.6 suivante :

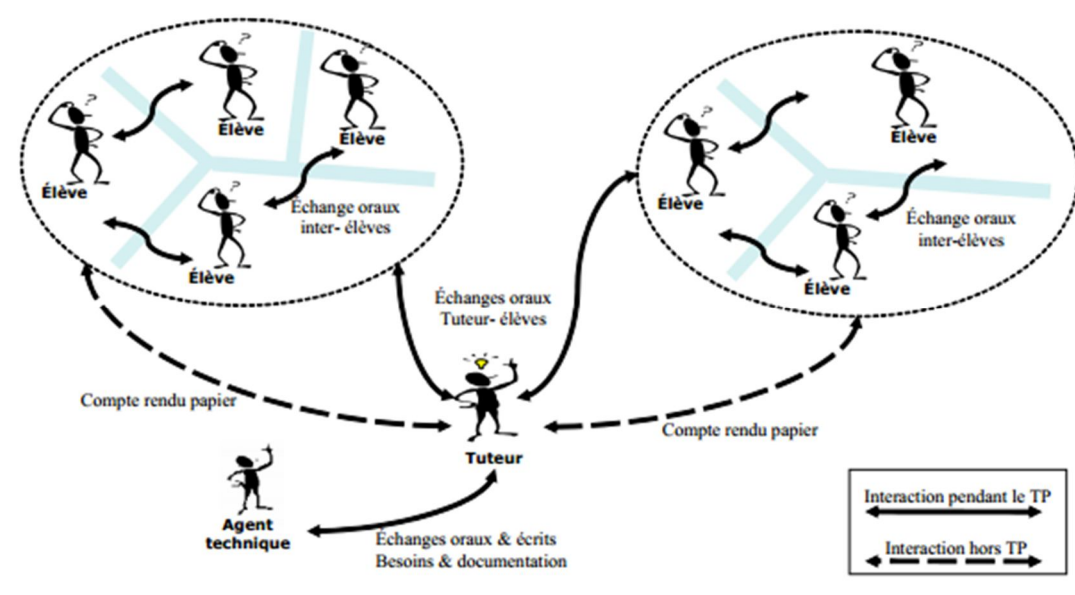


Figure 1.6 : Les diff  rentes interactions dans un TP en pr  sentiel [Benmohamed, 2007]

Dans un TP, le mat  riel (le dispositif) est peut-  tre manipul   et visualis   par les apprenants. La s  ance de TP est le lieu d'un   change privil  gi   avec l'enseignant. Les apprenants encadr  s en petits groupes posent plus facilement des questions [Millar, 2009]. C'est aussi l'occasion de plus nombreux   changes avec les autres apprenants et d'un travail en groupe. Le r  le des enseignants dans les s  ances de TP est important, il s'agit d'aider les apprenants    construire un r  f  rentiel exp  rimental, d'  valuer leurs r  actions, leur capacit   de travail et leur   volution de fa  on individuelle, leur capacit      travailler en groupe, de leur faire prendre conscience de la performance et de l'importance de la s  curit  .

6.4. Typologie d'un TP

Nous avons r  sum   les typologies des TPs existantes dans la litt  rature   tudi  e [L  lev  , 2002],[Millar,2009],[Benmohamed,2007],[Bouabid,2012],[Mechta, 2012] dans la section suivante. Donc, Un TP peut s'organiser autour de contexte :

a- un syst  me en vraie grandeur et contexte :(unit   industrielle de production, atelier, magasin). Il s'agit de former du personnel qualifi      la conduite de ces unit  s.

b- un syst  me r  el    caract  re p  dagogique : Il s'agit d'un syst  me r  el d  connect   de son contexte d'origine pour   tre d  di      la formation.

c- un simple dispositif technique : dans ce cas, c'est le dispositif r  el qui est directement utilis   et contextualis   par le sc  nario p  dagogique.

d - une maquette p  dagogique:

Le petit dictionnaire de fran  ais Larousse d  finit les maquettes comme suit : « *Repr  sentation en trois dimensions, le plus souvent   chelle r  duite, mais fid  le dans ses proportions, d'une construction, d'un appareil, d'un d  cor, d'un objet quelconque* ». L'utilisation des maquettes est connue depuis les anciens si  cles (maison cr  toise, navire   gyptien, etc. trouv   dans les tombes) et    travers toutes les   poques : celles qui nous sont parvenues sont en g  n  ral li  es    l'architecture, aux arts militaires (plan-relief) et    la navigation.

e - un simulateur p  dagogique:

Le dictionnaire fran  ais *Le Robert* d  finit le mot simulateur comme suit « *Le mot « simulateur » vient du latin « simulator » : c'est un dispositif exp  rimental ou un programme informatique qui permet de reproduire    l'aide d'une maquette ou d'un programme informatique, le fonctionnement r  el d'un appareil    des fins d'  tudes, de d  monstration, ou d'explication* ».

f- l'analyse de donn  es exp  rimentales : Ces donn  es exp  rimentales (r  elles) n  cessitent une longue p  riode d'acquisition et elles sont issues d'instruments sophistiqu  s et co  teux. G  n  ralement, ces donn  es utilis  es pour valider des mod  les th  oriques acquis lors des cours magistraux, ou TDs [Benmohamed,2007][Bouabid,2012].

g-la mise en application d'un concept ou outil informatique:

TP portant sur les langages de programmation, les syst  mes d'exploitation, ou l'utilisation d'outil de mod  lisation (merise, UML).

Ces diff  rentes situations sont regroup  es dans de nouvelles classes par Mechta et Benmohamed dans [Mechta, 2012] et [Benmohamed, 2007] :

a : syst  mes r  els :

Le terme *r  el* veut dire que le dispositif utilis   p  dagogiquement est le dispositif d'origine par exemple: un simple dispositif technique.

b : systèmes physiques :

Le terme *physique* exprime que le système utilisé est de nature physique, c'est le cas des trois premiers exemples et des maquettes pédagogiques.

c: systèmes virtuels :

Les simulateurs sont les meilleurs exemples des principaux systèmes de cette classe. Le terme *virtuel* désigne tout objet non physique.

d : systèmes hybrides :

Un système composé d'une partie *physique* et d'une *partie virtuelle* (Figure 1.7). L'exemple- type de ce système est l'intégration d'un sous-système isolé (maquette par exemple) dans un système global simulé (système de production).

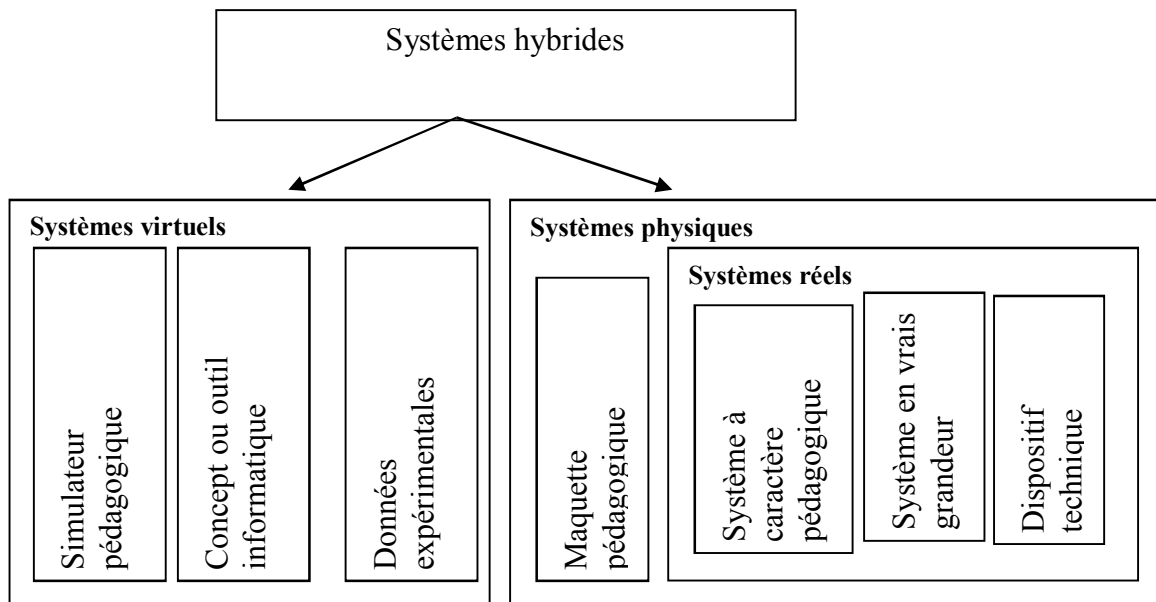


Figure 1.7 : Typologie des systèmes de TP adaptée et inspirée de [[Benmohamed, 2007] [Bouabid, 2013].

6.5. Phases de déroulement d'un TP

Les TP sont un type d'enseignement fondé sur l'apprentissage pratique avec, en particulier, la réalisation d'expérience qui permet aux apprenants de vérifier et de compléter ses connaissances dispensées dans les cours théoriques. Donc, la caractéristique intrinsèque qui différencie les séances de travaux pratiques par rapport aux autres types de cours qui se passent exclusivement à l'oral ou à l'écrit est l'exigence souvent de manipulation d'un matériel spécifique.

Gateau dans [Gateau et al., 2007] affirme que généralement un TP se déroule en trois phases:

- *Une phase pr paratoire   l'exp rience:* o  les apprenants sont pr par s   travers un rappel de cours et des renvois vers des lectures, initiation   l'utilisation des outils du laboratoire,  ventuellement un test d'entr e pour v rifier leur ma trise des pr requis avant d'entrer dans l'exp rience.
- *La phase d'exp rimentation:* il s'agit ici de conduire une ou plusieurs exp riences, selon l'activit  prescrite par les enseignants.
- *La phase post-exp rimentation:* consacr e   l'analyse et l' valuation des r sultats de(s) exp rience(s) men (es), la r daction d'une synth se ou d'un compte rendu individuel et/ou collectif et enfin l' valuation par le tuteur.

Par ailleurs, nous pouvons distinguer deux types de TP comme il est discut  dans [Millar, 2009]:

- ✓ *Les TPs de niveau introduction:* Les concepts pr sent s dans les lectures en classe ou dans les lectures individuelles sont renforc s par ce type de TP. Sont dirig s et con us pour cette fin.
- ✓ *Les TPs de niveau interm diaire et avanc :* Ce type de TPs exigent des  tudiants de mod liser et impl menter des solutions, mod liser des exp riences pour acqu rir des donn es afin de compl ter la mod lisation ou mesurer diverses caract ristiques. Donc, ils doivent inclure des probl mes ouverts.

Le d roulement des trois phases d'un TP est illustr  par la figure 1.8 suivante :

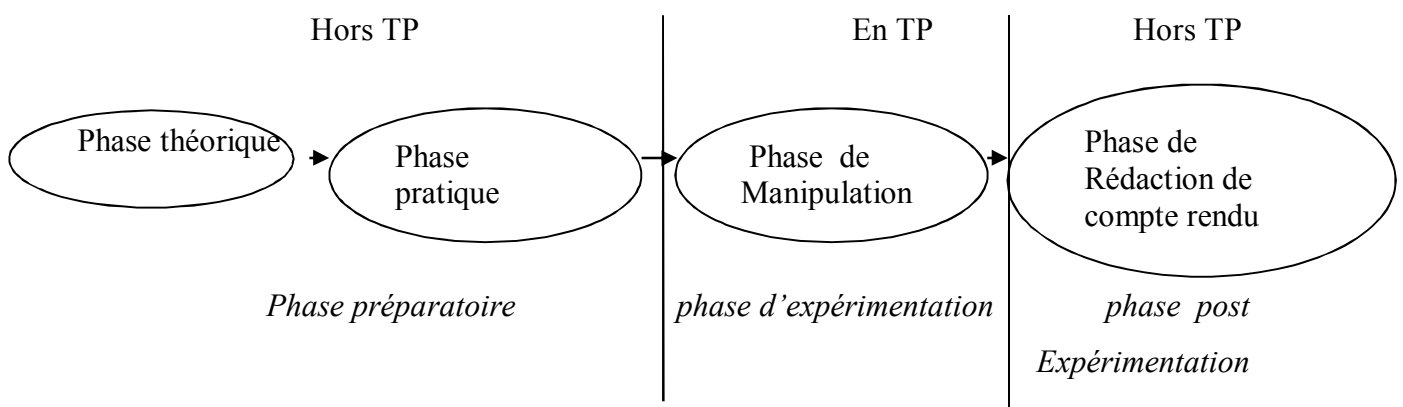


Figure 1.8 : Phases de d roulement d'un TP adapt es et inspir es de[Gateau et al., 2007]

6.6. Limites de TPs

Quelques limitations intrins ques des travaux pratiques classiques sont pr sent es par Lelev  en 2004 [Lelev , 2004]:

- le volume horaire est limit   dans l'emploi du temps et les rotations de TP sont insuffisantes si l'on compare le nombre d'  l  ves qui sont invit  s pour utiliser un nombre faible de syst  mes accessibles, en outre, les contraintes des emplois du temps, font que certains TP arrivent avant le cours magistral. Ce type de probl  me est class   dans *la limite du temps*.
- dans la majorit   des cas, la documentation est limit  e et parfois le sujet est donn   au moment de la session de TP, et dans certains cas, il devient indisponible en dehors des s  ances de TP, ce type de probl  me concerne *la limite documentation*.
- certains TP n  cessitent un   quipement lourd et on  reux que l'on ne peut pas dupliquer ce qui pose le probl  me de manque d'*'  quipement*.

6.7. Position de nos travaux

Les principaux objectifs des travaux pratiques sont de confronter les connaissances des apprenants    la r  alit  . Nous apportons ici une dimension suppl  mentaire : la distance au niveau de la mise en application d'un *concept* ou *un outil informatique*. (r  aliser un TP sur un langage de programmation    distance et   valuer les apprenants qui le r  alise).

Dans la section suivante, nous pr  sentons les travaux pratiques    distance (*T  L  TPs*) avec l'apparition d'un nouveau terme « *la distance* ».

7. Les travaux pratiques    distance (T  L  TPs)

7.1. D  finition

« Travaux pratiques    distance ou T  L  TPs, nous utilisons arbitrairement l'une ou l'autre d  nomination pour d  signer un TP pr  sentiel classique qui a   t     ventuellement modifi  , mais surtout   tendu afin d'  tre acc  d      distance via Internet » [Benmouhamed, 2007].

La distance d  signe la dispersion g  ographique et/ou la diff  rence temporelle des acteurs humains et mat  riels (auteurs, apprenants, tuteurs, dispositifs p  dagogiques) impliqu  s dans un syst  me de T  L  TP. Pour cette raison, « e -TP » est un terme ignor   dans notre terminologie. En effet, il n'implique pas n  cessairement la distance de la m  me mani  re que la « e-formation » o   le « e » d  signe la formation par la voix «   lectronique », c'  st-  -dire m  diatis  e par l'informatique.

7.2. Apport et limites des TélÉTPs

7.2.1. Apports

Les problèmes rencontrés dans les TP présentiels sont cernés et traités par les TélÉTPs, sur plusieurs plans (économique, pédagogique, etc.). Mechta dans [Mechta, 2012] discute l'ensemble des apports de TélÉTPs:

- Sur le plan économique :

- le partage d'instruments et d'équipements lourds et coûteux entre institutions [léléve, 2004]
- l'accès aux instruments d'un « laboratoire » n'importe quand et de n'importe où.

- Sur le plan pédagogique :

- l'illustration de Télé-Cours ou Télé-TD pour des apprenants distants ne pouvant pas accéder à des dispositifs réels (équivalent à une démonstration).
- l'intégration d'outils (simulateurs, réalité virtuelle, etc.) apportés par les TIC.
- l'ouverture à des utilisateurs autres que la population d'origine : scénarios pédagogiques adaptés à des niveaux scolaires différents (du lycée à l'école d'ingénieurs) et même au grand public en dehors des créneaux réservés aux apprenants.
- les démonstrations mettant en valeur l'équipement et le savoir-faire de l'établissement auprès d'un plus grand public.

7.2.2. Limites

Les TélÉTPs ont des limites selon lélévé dans [lélévé, 2005] dues à la mise à distance des manipulations qui engendre de nombreuses difficultés en termes de commandabilité (le recours aux techniques de la réalité virtuelle peut résoudre ce problème), observabilité (une dégradation dans l'observation sensorielle du système piloté) et sécurité (sécurité du système informatique gérant l'expérimentation et sécurité du matériel à télé-manipuler).

7.3. Typologies des situations

Par analogie d'un TP présentiel et leur classification, nous distinguons trois formes de TélÉTP :

- les TélÉTPs fondés sur un système virtuel autrement nommé « laboratoire virtuel » traduction du concept anglophone « virtual laboratory ». C'est ce type de TélÉTP qui a constitué l'axe principal de nos travaux.
- les TélÉTPs fondés sur la manipulation d'un système physique. C'est le cas des laboratoires réels à distance connus sous le terme « laboratoire distant » traduit littéralement de l'anglais « remote laboratory ».
- les TélÉTPs fondés sur un système hybride (physique-virtuel). Il s'agit d'un système physique intégré dans un système virtuel simulateur.

8. Quelques EIAH consacrés à l'apprentissage des TélÉTPs

Les TélÉTPs ont commencé vus le jour dans les travaux de recherche pour combler les manques de la formation à distance dans ce domaine-là au début du 21eme siècle. Plusieurs projets et multitudes d'études de faisabilité traitant ce champ particulier de recherche (TélÉTPs) dans divers domaines ont vu le jour résultant pour la plupart en environnement propriétaires et utilisables uniquement dans le laboratoire ou l'université à l'origine du projet. C'est pourquoi malgré de nombreux sites traitant des TélÉTPs que nous avons vus sur Internet, peu sont visibles et donc encore moins en démonstration [Bouabid, 2012].

Nous avons fait une étude de quelques projets afin de nous permettre de retracer le panorama des solutions existantes en termes de TélÉTP.

8.1. TPLine

TPLine [TPLine] est une plateforme gratuite permet de mettre à la disposition des apprenants et des formateurs des travaux pratiques à partir de mesure faite sur des systèmes industriels réels mets à distance [Lombard,2013]. Tous les niveaux en partant du collègue jusqu'au Post Bac en passant par le lycée et lycée professionnel sont introduits dans des TPs sur cette plateforme. Actuellement, des TPs sont proposés sur plusieurs systèmes tels quels (stations météo, tapis roulant, Le contrôle de la consommation électrique dans un logement, etc.).

Quoique le fonctionnement de TPLine soit très pratique pour les apprenants, cette plateforme ne permet pas réellement le « pilotage » des systèmes réels. Elle se limite à l'utilisation des mesures faites sur ces systèmes. La notion de session de manipulation a été retrouvée avec la possibilité de sauvegarder des configurations et réponses de l'apprenant,

l'apprenant peut aussi charger ses sessions de manipulation précédentes. En parle ici la notion de traçabilité des apprenants [Lombard, 2013].

8.2. Évariste

Un grand nombre de chercheurs et d'enseignants appartient à L'association Évariste (Étude et Valorisation des Applications de la Recherche en Informatique sur les Systèmes Tutoriels d'enseignement) (de secondaire et de supérieur). Ceux-ci travaillent à la conception, à la réalisation et à l'expérimentation d'ensembles logiciels et matériels éducatifs, ils utilisent l'informatique comme outil pour mesurer, contrôler, représenter des phénomènes physiques dans un but commun ceci est de favoriser la démarche d'apprentissage et l'activité de modélisation /réflexions menées par les élèves. Ces logiciels proposent des TP dans différents domaines : physique, chimie, Science et Vie de la Terre, dans ces logiciels, les TP sont réalisés sur des données déjà acquises, les apprenants interprètent les données, mais ne pilotent pas les systèmes [Relier, 2003], [Marcotte, 2004], [Lombard, 2013].

8.3. ICTT@Lab

C'est un projet résultant des travaux de thèse de Hcene Benmohamed rendu, en 2007, et développé dans le laboratoire ICTT de l'INSA de Lyon. ICTT@Lab (generIC framework for remoTe and virTu@l LABoratory integration) est un framework d'intégration des TéléTPs au sein d'une plateforme de formation en ligne existante.

Pour faire le lien entre l'EIAH existant et le laboratoire distant, une partie interface a été rajoutée à l'environnement d'origine dans le dessein de supporter les TéléTPs. Cette partie interface que nous avons parlé est composée de deux composants ELaMS et OntoServ [Lombard, 2013] [Bouabid, 2012].

ELaMS (Electronic Laboratory Management System) représente le middleware, son rôle est d'assurer le dialogue entre la couche téléformation et la couche téléopération. Les nouveaux dispositifs technologiques disponibles dans le laboratoire distant sont installés et référencés par ce middleware, il est également destiné pour gérer l'utilisation de ces derniers (Les nouveaux dispositifs technologiques) par les apprenants et les formateurs (gestion des droits d'accès aux dispositifs, choix et utilisation du dispositif, ...) [Lombard, 2013].

OntoServ (Serveur d'Ontologies) est le conteneur des fichiers d'ontologies créés [Lombard, 2013]. Les dispositifs technologiques à disposition dans le laboratoire distant sont décrits graphiquement par les ontologies. Pour chaque dispositif, on retrouve une ontologie mère qui permet de décrire le dispositif de façon globale avec ses principaux composants. Chaque composant et ses principales fonctionnalités sont décrits par Les ontologies filles quant à elles [Lombard, 2013]. Ces travaux de thèse ont expérimenté des TéléTP seulement sur deux systèmes automatisés didactiques (le magasin vertical et le four industriel). La communication avec les API de ces deux systèmes didactiques a été réalisée par l'ajout d'un module serveur web permettant la communication TCP/IP. Malheureusement, dans ces travaux de thèse Hcene Benmohamed ne présente pas les protocoles de communication utilisés pour communiquer avec l'API.

Le composant ELaMS souffre d'une insuffisance majeure, c'est que L'intégration d'un nouveau dispositif n'est pas faite d'une façon automatique, il est exigé à l'administrateur de rentrer les informations nécessaires (le nom du dispositif associe les URL permettant de trouver l'IHM de commande réalisé avec le langage VRML, les dispositifs environnementaux (caméra, micro) et, l'automate permettant de communiquer avec lui) via un formulaire [Lombard, 2013].

8.4. E-TéléTPC@AALP

E-TéléTPC@AALP (environnement de travaux pratiques collaboratifs à distance: application à l'apprentissage des langages de programmation) c'est une plateforme développée, en 2009, dans le cadre de nos travaux de recherche de magister.

Notre hypothèse se fonde sur la mise en place d'un TéléTP collaboratif autour des langages de programmation en informatique. De là, nous proposons une architecture d'un environnement d'apprentissage collaboratif « plateforme ».

Notre plateforme proposée contient trois niveaux : le niveau téléformation, le niveau interface et le niveau téléprogrammation. Notre plateforme contient plusieurs espaces :

- *espace apprenant* : pour accéder à l'espace de l'apprenant.
- *espace concepteur de TP* : permet à l'enseignant concepteur de TP d'accéder à son espace.
- *espace tuteur* : Pour accéder à l'espace tuteur.

9. Conclusion

Dans ce chapitre, nous avons exploré principalement le contexte de nos travaux. Nous avons commencé par présenter les concepts-clés pour la compréhension des EIAH. En considérant une plateforme de travaux pratiques à distance comme étant un EIAH à usage spécifique et dépendant d'un environnement classique de téléformation (LMS, LCMS).

Ensuite, nous avons donné une vue générale sur les activités des travaux pratiques et travaux pratiques à distance. Nous avons conclu ce chapitre par une exposition sur quelques EIAH ayant proposés des TéléTPs sur des systèmes didactiques en mettant l'accent sur notre EIAH(E-TéléTPC@AALP) consacré à l'apprentissage des TéléTPs des langages de programmation en informatique.

Un retour d'usage sur cette première conception nous a permis de proposer une nouvelle conception améliorée notamment par l'intégration d'un mécanisme d'évaluation de l'apprenant au cours de son processus d'apprentissage. Dans le deuxième chapitre, nous allons étudier l'évaluation des apprenants dans les plateformes d'apprentissage.

Évaluation des apprenants dans les EIAH

1. Introduction

L'objet des travaux de recherche relatifs aux EIAH est d'étudier les situations pédagogiques informatisées et les logiciels qui permettent ces situations [Tchounikine, 2009]. Ce type de contribution de recherche favorise les interactions entre l'apprenant et l'outil, donc, la représentation de l'évolution de la progression de l'apprentissage de l'apprenant devient une exigence. L'évaluation des connaissances des apprenants est l'activité pédagogique qui est chargée d'identifier cette progression [Bouarab-Dahmani, 2010].

L'évaluation est une activité clé dans n'importe quel domaine d'apprentissage. Elle est importante et indispensable tant pour l'apprenant que pour l'enseignant. Actuellement, cette activité attire, de plus en plus, l'attention des chercheurs travaillant dans le domaine de l'apprentissage électronique (e-learning). Dans la littérature, plusieurs travaux de recherche ont été réalisés dans ce contexte dans l'objectif de mettre en l'évaluation de l'apprenant.

Dans ce chapitre, nous mettons en place une partie du cadre théorique qui concerne l'évaluation comme activité pédagogique dans un processus d'apprentissage ou de formation. Ensuite, nous nous intéressons à l'évaluation dans les environnements informatiques pour l'apprentissage humain et l'évaluation des apprenants. Nous définissons quelques notions et concepts liées à l'évaluation dans les EIAH. Nous insistons sur les types d'évaluation proposés dans la littérature ainsi que sur les stratégies d'évaluation utilisées dans les EIAH consacrés à l'apprentissage des TêLÉTPs.

2. L'évaluation dans les activités d'apprentissage

2.1. D'où vient le mot évaluation

Le terme *évaluer* vient du latin « *Valère* » signifiant valoir. Dans le dictionnaire Larousse, on trouve la définition de ce mot : *Déterminer, fixer, apprécier la valeur, le prix de quelque chose, d'un bien, etc.*

La première apparition des modes d'évaluation était vers le 16^{ème} siècle dans les pays européens par les collèges des Jésuites. L'objectif était d'élaborer un système sélectif efficace dans les collèges. Il s'agissait de privilégier les plus méritants et d'éliminer les autres [Tanana, 2009].

Comme décision à prendre et après un classement élémentaire qui a été effectué selon des critères choisis par des spécialistes (par exemple, les copies corrigées étaient classées selon leur mérite) Les meilleurs élèves étaient encouragés par la « distribution d'un prix ». Ensuite, des appréciations chiffrées étaient remplacées par les indications de rang, tout en gardant la notion de classement, ce type d'évaluation généralement fait en fin d'année scolaire [Tanana, 2009].

Pour stimuler de plus en plus la compétitivité et la rivalité entre les élèves, l'attribution de divers trophées, fait son apparition. Or, en France, l'évaluation est officialisée vers le 19^{ème} siècle, elle utilise le système de notation de 0 à 20. Par la suite, des nouveaux concepts sont apparus tel que la notion de moyenne et le redoublement [Tanana, 2009].

Au début du 20^{ème} siècle, un nouveau mouvement de rationalisation de l'évaluation voit le jour d'où l'apparition de deux nouvelles sciences de l'évaluation: la « docimologie » ou « Science de l'évaluation » et l'Edumétrie.

- **la docimologie** : Dans le vocabulaire de la psychologie de Piéron, la docimologie est définie comme suit : "*l'étude systématique des examens (modes de notation, variabilité interindividuelle et intra-individuelle des examineurs, facteurs subjectifs, etc.)*".

André Quinton dans [Quinton, 2005] définit **la docimologie** comme suit « *La docimologie est la science des examens. Le terme a été créé par Henri PIERON à partir du grec dokime signifiant épreuve* ».

La docimologie est constituée de deux approches différentes mais étroitement liées :

- *la docimologie critique* : qui s'occupe du fonctionnement des pratiques d'évaluation

- *la docimologie prescriptive* : qui s'occupe de l'amélioration du fonctionnement de ces pratiques.

➤ **Edumétrie** : Selon Marc Demeuse dans [Demeuse, 2002], la première utilisation du terme «édumétrie» en langue française peut être trouvée dans l'ouvrage de Viviane de Landsheere intitulé «Faire réussir, faire échouer. La compétence minimale et son évaluation» publié, en 1988, aux Presses universitaires de France (p.59): « *Edumétrie. Mot créé par Carver (1974), sur le modèle de psychométrie, pour désigner l'étude quantitative des variables relatives aux apprentissages suscités par l'éducation: influence d'une action pédagogique, performance effective par rapport à une performance attendue, épreuves centrées sur les objectifs...* », est une discipline qui s'intéresse aux mesures des résultats d'apprentissages. Alors que la psychométrie classe les individus soit avant, soit après l'étude les uns par rapport aux autres, donc, on peut dire que le progrès réalisé par un individu est évalué par l'édumétrie entre les étapes successives de son apprentissage [Bey, 2013].

Pour Charles Hadji [Hadji, 1997] et dans le cadre de l'apprentissage humain, « *évaluer signifie : interpréter, vérifier ce qui a été appris, compris, retenu, vérifier les acquis dans le cadre d'une progression, juger un travail en fonction des critères donnés, estimer le niveau de compétence d'un apprenant, situer l'apprenant par rapport à ses compétences, déterminer le niveau d'une production donnée par l'apprenant* ».

2.2. Évaluation, contrôle : Quelles sont les différences !

On résume les différences entre l'évaluation et le contrôle dans quelques points comme suit :

L'évaluation est une activité qui ne peut être que partagée par les acteurs, elle privilégie le qualitatif sur le quantitatif. Au-delà du contrôle analytique, elle est conçue comme un processus intervenant dans un système ouvert, en évolution, en vue d'en élucider le fonctionnement et l'évolution. Elle englobe et dépasse le contrôle. Tandis que le contrôle est destiné pour mesurer des écarts entre des produits, des démarches et une

norme extérieure, préétablie. Aussi, le contrôle est régi par des critères de conformité, de logique, de cohérence, son rôle est la vérification pour valider ou rejeter, corriger ou sanctionner [Simmons-Hariss, 2002].

2.3. les acteurs de catégorisation des buts de l'évaluation?

[Berry, 2014] déclare que deux critères essentiels sont pris en considération pour catégoriser les buts de l'évaluation : selon le but final de l'évaluation et selon les acteurs qui sont impliqués dans le processus de l'évaluation, ces acteurs sont : apprenants, enseignants ou ceux qui s'intéressent aux résultats de l'évaluation : institutions et parents.

Pour le premier critère, deux raisons justificatives pour évaluer sont citées dans [Berry, 2014] :

- pour établir un jugement sur la performance des apprenants ou l'efficacité d'un système d'enseignement,
- pour améliorer l'apprentissage.

2.3.1. L'évaluation pour l'enseignant

Les enseignants évaluent leurs apprenants pour vérifier si les objectifs qu'ils ont définis ont été atteints. Dans ce cas, l'évaluation est indispensable, plusieurs utilités de l'évaluation sont citées dans [Durand, 2006] [Bey, 2013] en ce qui concerne enseignant. Elle permet de:

- donner du sens à partir des analyses des réponses des apprenants;
- faire en sorte qu'elles soient utiles dans la classe, réfléchir sur les besoins spécifiques des apprenants;
- créer des groupes de besoins en ciblant les difficultés;
- agir après l'évaluation;
- évaluer pour prendre des décisions;
- comptabiliser ce qui vient d'être compris que remédier aux dysfonctionnements de la leçon;
- contrôler en permanence la progression d'un apprenant;

2.3.2. L'évaluation pour l'apprenant

L'évaluation est une activité très impopulaire au sein de la communauté des apprenants parce que cette dernière avec sa forme de pratique dominante est toujours mal perçue par eux en voyant toujours cette activité comme étant la responsable de leur échec. Malgré ses utilités variantes à l'apprenant, ces utilités sont citées par Ross et Ramond dans [Ross, 2006], [Ramond, 2010] :

- elle permet de juger le niveau des apprenants
- inciter l'apprenant à identifier ses erreurs par des questions précises ou des références à des outils précis;
- sélectionner les erreurs prioritaires.
- favoriser la motivation et l'autonomie;
- elle permet la représentation de l'état cognitif de l'apprenant

2.3.3. L'évaluation pour l'institution

Comme l'évaluation est une activité importante pour les deux acteurs cités précédemment, elle est utile aussi pour l'institution. Ses utilités sont citées dans certaines références.

- agir après l'évaluation [Ramond, 2010] ;
- évaluer pour prendre des décisions [Jorro, 2000] ;
- diagnostiquer, prévoir, positionner les formés [Ramond, 2010] ;
- faire le point, conduire l'action, régulée pour les formateurs [Jorro, 2000] ;
- juger de l'efficacité : pédagogique, dans la profession, sur le terrain [Jorro, 2000].

2.4. Les critères d'évaluation

À partir de réalisations observables et en fonction des objectifs d'apprentissage visés, les enseignants sont les seuls responsables de fixer les critères d'évaluation [Tanana, 2009]. André Peretti dans [Peretti, 2009] divise les critères d'évaluation en deux groupes :

2.4.1. Les critères quantitatifs

Les valeurs mesurables sont des meilleurs exemples concernés par ces critères, André Peretti dans [Peretti, 2009] cite des exemples de valeurs mesurables :

1. le temps de réponse mis pour répondre à chaque question ;
2. évaluation terminée dans les délais ;
3. les résultats à des questions fermées ;
6. etc.

2.4.2. Les critères qualitatifs

Les valeurs observables, difficiles à mesurer, sont des exemples concernées par ces critères. André Peretti dans [Peretti, 2009] donne quelques exemples de ces valeurs :

1. la démarche stratégique de résolution utilisée ;
2. comportement face à une situation imprévue ;
3. temps de résolution limité ;

4. employer des sources d'informations ;
5. évaluation individuelle ou en groupe ;
6. etc.

2.5. Correspondance entre objectifs pédagogiques et outils d'évaluation

En (1956) Bloom [Bloom, 1956] a proposé une classification hiérarchisée qui permet de classer les objectifs pédagogiques (ou d'apprentissage) en 6 niveaux ordonnés en fonction de leurs complexités comme suit :

1. la connaissance,
2. la compréhension,
3. l'application,
4. l'analyse,
5. la synthèse,
6. l'évaluation

L'enseignant, lorsqu'il veut évaluer son apprenant, il doit, dans un premier temps, viser l'objectif pédagogique à évaluer. Dans le domaine d'évaluation assistée par ordinateur, il est convenu que l'outil de suivi pédagogique ou d'évaluation est qualifié par le type de réponse qu'il implique. Trois sortes d'outils d'évaluation sont proposées. Les premiers outils offrent des réponses dites plus fermées (comme les QCM) tandis que les seconds offrent des réponses dites très ouvertes (comme la production d'un travail personnel), les derniers offrent des réponses semi ouvertes [Bouarab-Dahmani, 2010].

Entre les formes de fermeture et d'ouverture, il existe toute une variété d'outils où le degré de liberté laissé à l'apprenant est variable.

Nous pouvons distinguer trois types de questions [Tanana, 2009][Lafifi, 2007], [Bouarab-Dahmani, 2010] que nous pouvons utiliser dans les évaluations.

2.5.1. Les questions ouvertes

Elles sont beaucoup plus subjectives et donc plus difficiles à corriger, parce que l'apprenant répond avec ses propres mots. Le principal avantage de ce type de question est la possibilité d'évaluer des objectifs pédagogiques de niveau cognitif élevé [Bouarab-Dahmani, 2010][lafifi, 2007].

2.5.2. Les questions fermées

Dans ce cas, la réponse se trouve parmi une liste de choix proposée à l'apprenant. Les questions fermées sont plus faciles à implémenter et sont corrigées d'une manière objective comme par exemple celles citées dans [Tanana, 2009] :

2.5.2.1. Les QCM (questions à choix Multiples)

Les QCM ont été intensivement utilisés dans les évaluations pédagogiques, car ils sont faciles à implémenter et permettent un traitement rapide des réponses. Il existe plusieurs variantes de QCM : vrai/faux, trouver la seule bonne réponse, trouver la meilleure réponse, trouver la seule réponse fausse, trouver toutes les bonnes réponses, etc.. [Bouarab-Dahmani, 2010][Lafifi, 2007].

2.5.2.2. Question de types réarrangement

Dans ce type de question, l'apprenant doit réarranger une liste d'éléments qui font partie de l'énoncé dans le bon ordre [Tanana, 2009].

2.5.2.3. Question de types association ou appariement

Dans ce type de question, deux listes de choix sont proposées à l'apprenant. Ce dernier doit faire une correspondance ou une association entre un ou plusieurs éléments de la première liste avec un ou plusieurs éléments de la deuxième liste. Cette caractéristique de bijectivité de correspondance entre les éléments augmente la complexité de la question elle-même [Tanana, 2009].

2.5.3. Les questions semi ouvertes

Dans ce type de question, la réponse de l'apprenant est considérée comme un compromis entre les deux types précédents, cela désigne que l'apprenant ne peut pas exprimer sa réponse avec son propre langage, mais il a la possibilité de répondre de plusieurs façons différentes que celle proposées par le système sauf qu'elle doit respecter certaines contraintes : la réponse à ce genre de question doit être brève et précise. Donc, on peut dire que ce type de question comporte des réponses proposées et offrent la possibilité d'ajouter des réponses libres [Lafifi, 2007], [Tanana, 2009], nous distinguons :

2.5.3.1. Les questions à réponse courte

Dans ce genre de question, l'enseignant doit poser des questions claires et précises qui imposent des réponses brèves et spécifiques par l'apprenant. Ce dernier de son rôle doit répondre à ces questions par un ou plusieurs mots [Tanana, 2009].

2.5.3.2. Les questions de type « carte conceptuelle »

Dans l'année 2005, Delorme dans [Delorme et al., 2005] utilisait ce type de questions pour le développement d'un outil générique pour l'évaluation des connaissances de l'apprenant pour les hypermédias adaptatifs. Cet outil a les objectifs suivants :

- il permet de générer un rapport complet sur la production de l'apprenant,
- il permet ainsi de savoir si l'apprenant a compris ou non une notion du cours
- il permet d'extraire les difficultés rencontrées afin d'y remédier.

Dans la figure (2.1) suivante, on exprime les correspondances entre objectifs d'apprentissage et outils d'évaluation utilisés.

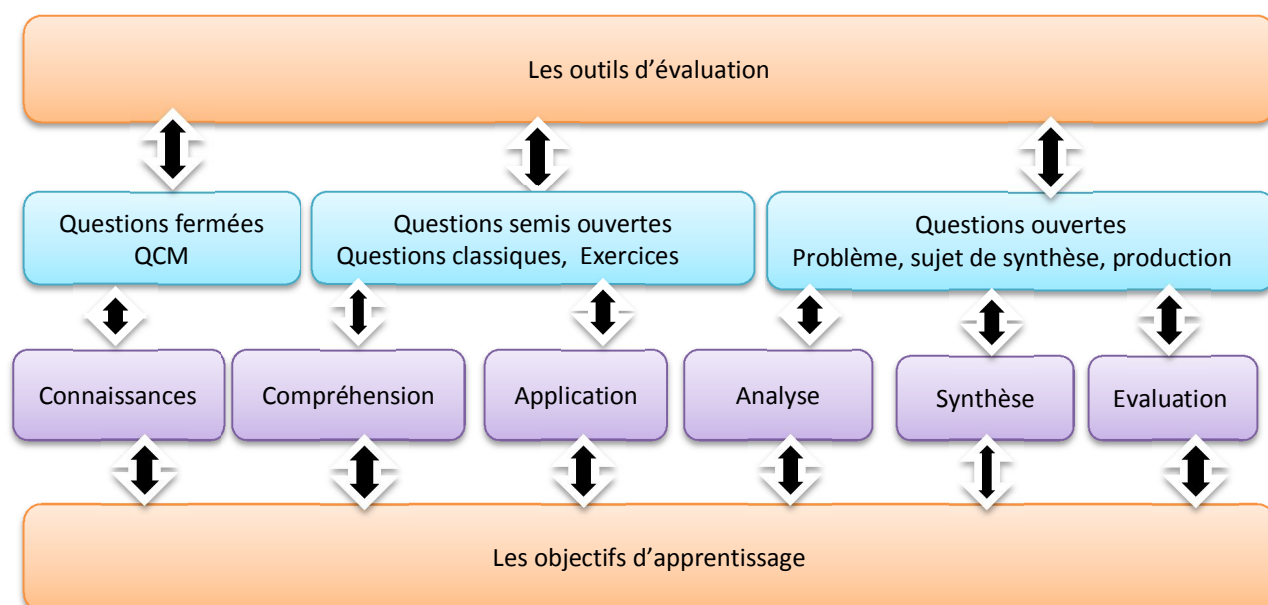


Figure 2.1 : Correspondances entre objectifs d'apprentissage et outils d'évaluation.

2.6. Les différentes formes d'évaluation

L'évaluation a principalement deux objectifs : évaluation pour l'apprentissage ou de l'apprentissage suivant le rôle qu'elle doit jouer. L'évaluation se présente sous des formes différentes. En 2006 Durand dans [Durand, 2006] citait les différentes formes d'évaluation comme suit [Durand, 2006]:

- l'évaluation pronostique ;
- l'évaluation formative ;
- l'évaluation formatrice ;
- l'évaluation diagnostique ;
- l'évaluation sommative.

Sans oublier de cette typologie les évaluations normatives et critériées. L'évaluation normative permet de comparer les résultats d'un apprenant par rapport à ceux d'autres apprenants. Tous les types d'évaluation cités précédemment reposent sur des critères soient explicites ou implicites. Ce qui nous a conduit de parler de l'évaluation critériée [Durand, 2006]. Ce dernier type d'évaluation ne situe pas les élèves les uns par rapport aux autres ; elle prend plutôt compte des performances (que ce soit en plus ou en moins) par rapport à ensemble de compétences (savoirs, savoir-faire, savoir-être). C'est une activité qui permet de situer un comportement par rapport à une cible (le critère qui correspond à l'objectif à atteindre). Nous pouvons dire autrement que l'évaluation compare le résultat par rapport à un critère/performance [Bey, 2013].

Les évaluations normatives et critériées sont donc transversales aux cinq autres présentées ci-dessus [Durand, 2006].

2.6.1. L'évaluation pronostique

Permet d'évaluer les prérequis d'un apprenant pour poursuivre un apprentissage, un cycle d'étude ou à exercer une profession. Le but de ce type d'évaluation est de vérifier si l'apprenant a les connaissances nécessaires pour poursuivre une formation donnée. Pour cela, l'évaluation pronostique est généralement effectuée au début ou en fin de formation pour des formations ultérieures [Durand, 2006].

2.6.2. L'évaluation formative

L'évaluation formative est une « évaluation dont l'ambition est de contribuer à la formation » [Lafifi, 2007]. Son but est pédagogique que ce soit, pour l'enseignant ou pour l'apprenant, elle permet de réguler l'enseignement par la fourniture des informations permettant à l'enseignant d'adapter son enseignement, afin que les apprenants puissent

suivre son cours [Hanna, 2004]. Elle entre dans le cadre d'un enseignement différencié⁸ [Durand, 2006].

C'est une évaluation qui se produit à n'importe quel moment au cours d'un processus d'apprentissage. Son but est de guider et de réguler l'apprenant dans son apprentissage, ainsi que de situer les connaissances acquises (ou performances accomplies) par lui par rapport à un objectif d'apprentissage donné.

2.6.3. L'évaluation formatrice

C'est une forme particulière d'évaluation formative. Dans la formation formative c'est l'enseignant qui est concerné par la régulation de l'apprentissage, cependant dans l'évaluation formatrice, la régulation est assurée par l'apprenant [Bey, 2013].

2.6.4. L'évaluation diagnostique

L'évaluation formative est basée en partie sur l'évaluation diagnostique [Durand, 2006]. L'évaluation diagnostique peut aider à identifier les connaissances actuelles des apprenants pour un sujet donné, leurs compétences et capacités, et de dissiper les malentendus avant que l'enseignement ait lieu. Connaître les forces et les faiblesses des apprenants peut aider à mieux planifier ce qu'il faut enseigner et comment l'enseigner [Hanna, 2004]. Dans le cadre d'une évaluation formative, ce diagnostic permet la remédiation et la mise en œuvre d'une pédagogie différenciée.

2.6.5. L'évaluation sommative

L'évaluation sommative a lieu après que l'apprentissage est terminé. Elle peut aussi apparaître tout au long du processus de formation. Elle permet de fournir des informations et des commentaires qui résument le processus de l'enseignement et de l'apprentissage. L'évaluation sommative s'intéresse essentiellement aux productions réalisées par les apprenants. En général, son but est de montrer si l'apprenant a été capable d'atteindre les objectifs visés par la formation. Elle est en opposition avec l'évaluation formative, elle ne régule pas l'apprentissage, elle le contrôle. L'évaluation sommative est plus axée sur le produit et évalue le produit final, alors que l'évaluation formative met l'accent sur le

⁸ Enseignement différencié : Basée sur l'individualisation des parcours de formation, centrée sur l'apprenant ou le modèle de l'apprenant, de telle façon que « chaque apprenant se trouve, aussi souvent que possible, dans des situations d'apprentissage fécondes pour lui, c'est-à-dire susceptibles de le faire progresser ».

processus vers l'achèvement du produit. Une fois le projet terminé, pas d'autres révisions peuvent être faites [Hanna, 2004].

3. Étude comparative entre les différentes formes de l'évaluation

La table 2.1 suivante récapitule les évaluations existantes en termes d'objectifs (utilisation de l'évaluation), d'insertion dans les processus d'apprentissage (à quel moment l'évaluation est menée) et d'influence de facteur de subjectivité sur l'activité d'évaluation.

Types d'évaluation	Moment d'insertion	Objectifs	La subjectivité	Pour ou de L'apprentissage
pronostique	Avant l'apprentissage	Prédire, Contrôler l'accès à un cycle ou une année d'étude.	oui	pour
formative	Pendant l'apprentissage	Informar, Réguler l'activité de l'apprenant.	oui	Pour / de
formatrice	Pendant l'apprentissage	Informe, Permettre à l'apprenant de réguler son apprentissage	oui	Pour / de
diagnostique	Juste avant un apprentissage	Informar, Évaluer un niveau de compétences.	oui	pour
Sommative	Après l'apprentissage	Certifier, Établir un bilan certifié Des résultats de l'apprenant.	oui	de

Table 2.1: Comparaison entre des différentes formes de l'évaluation inspirée de [Durand, 2006] et étendue.

Nous pouvons déduire de la précédente étude que toutes les formes d'évaluation sont subjectives, c'est-à-dire qu'elles sont influencées par la personnalité de l'évaluateur engendrant.

Avec l'apparition des nouvelles technologies, l'émergence de l'évaluation automatisée ou l'évaluation assistée par ordinateur devient une solution très efficace pour réduire le problème de la subjectivité et rendre l'évaluation plus constructive. C'est dans ce sens qu'une extension des méthodes et des techniques traditionnelles d'évaluation a vu le jour, surtout avec l'apparition des EIAH. C'est ce concept que nous nous proposons d'examiner d'une façon plus précise dans la section suivante.

4. L'évaluation dans les EIAH

Les environnements informatiques pour l'apprentissage humain sont les meilleurs outils permettant d'assurer les activités d'apprentissage et d'évaluation chez les apprenants. Si l'on prend l'activité d'évaluation, l'utilité des EIAH réside en matière d'un gain énorme en termes de consistance contrairement à l'évaluation à la main qui est un processus coûteux en temps et un lourd fardeau.

4.1. Évaluation, session d'apprentissage : quelle relation !

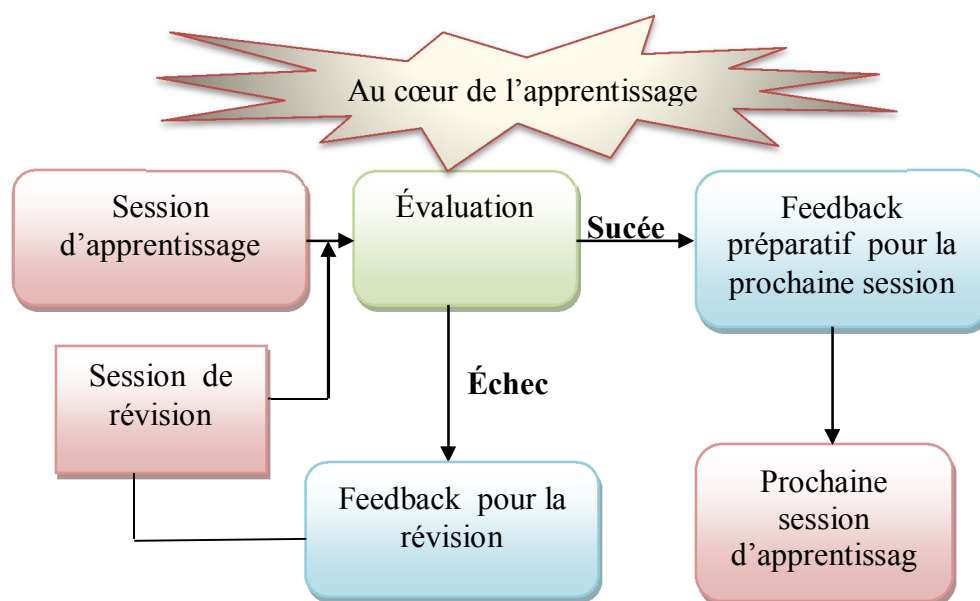


Figure 2.2: Relation entre l'évaluation et la session d'apprentissage adaptée et inspirée de [Bey, 2013]

Notre interprétation de la figure 2.2 se concentre sur les deux points essentiels suivants : (1) la location stratégique de l'activité de l'évaluation : elle se positionne au cœur de n'importe quel processus d'apprentissage [Bey,2013] et (2) son rôle principal : elle est considérée comme le moyen de régulation de l'apprentissage en prenant en compte sa localisation au milieu d'une session d'apprentissage, de telle sorte qu'elle permet des retours en arrière en cas de l'échec de la session d'apprentissage par la présentation d'un feedback fourni à l'apprenant pour plus d'éclaircissement et la session d'apprentissage sera révisée puis réévaluée, ou elle permet à l'apprenant de se préparer à la prochaine session d'apprentissage par la fourniture d'un feedback de préparation de la session d'apprentissage suivante . Donc si en prend en compte, la localisation stratégique de cette activité conviviale deux objectifs principaux sont tirés :

- évaluation pour l'apprentissage : dans ce cas l'évaluation se positionne avant la session d'apprentissage.
- évaluation de l'apprentissage : dans ce cas l'évaluation se positionne après la session d'apprentissage.

4.2. La portée de l'évaluation

Dans les EIAH, on peut aussi distinguer les évaluations en fonction de leur portée ou leur impact. Une portée est faible, moyenne ou haute en fonction des conséquences de l'évaluation pour l'apprenant [Adafer, 2005] :

- une évaluation dont l'impact est **faible** est habituellement formative et ses résultats demeurent locaux.
- une évaluation dont l'impact est **moyen** peut avoir des résultats locaux ou nationaux, mais ceux-ci ne sont pas déterminants dans la vie de l'apprenant évaluée.
- une évaluation à **fort impact** modifie le parcours d'un apprenant, ses rôles et activités à venir.

Aussi en EIAH, les cinq grandes catégories d'évaluation sont présentées. Chaque type d'évaluation en EIAH est détaillé par l'introduction d'exemples illustratifs :

4.3. L'évaluation sommative et pronostique

L'évaluation sommative est généralement menée dans des environnements d'apprentissage fondés sur des banques de questionnaires à choix multiples (QCM). Ce type d'évaluation fait partie de ce qu'on appelle les tests adaptatifs par ordinateurs ou **CAT** (*Computer Adaptive Test*). Ces tests permettent de certifier des connaissances de base. [Durand, 2006].

Dans un CAT, l'évaluation globale menée est sommative et individuelle et on évalue des réponses à un QCM. Avec un CAT, les apprenants sont assis devant des ordinateurs et passent des tests qui sont adaptés à leurs niveaux de capacité. Un CAT fonctionne de la manière suivante : si un apprenant répond à une question particulière correctement, on demande alors à l'apprenant de répondre à un élément plus difficile. Inversement, si l'apprenant donne une mauvaise réponse, un élément plus facile sera présenté [Linacre, 2000].

Un CAT permet de certifier un niveau de compétence, que ce soit en anglais pour le TOEFL ou sur les connaissances pré-universitaires pour le GRE.

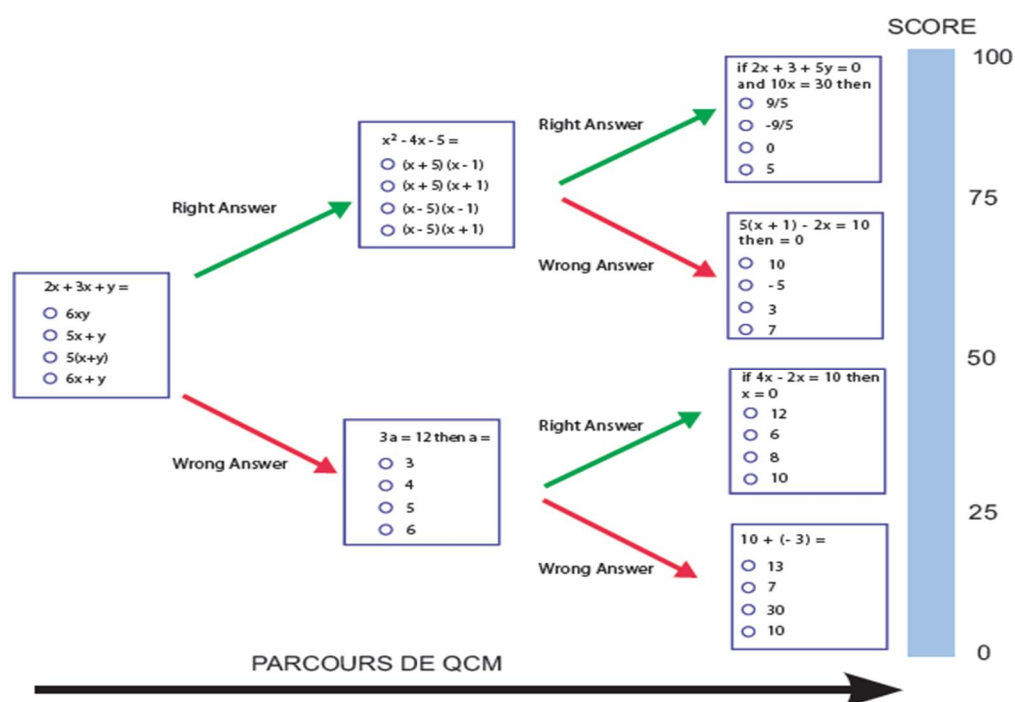


Figure 2.3 : Le principe de CAT [Durand, 2006].

4.4. L'évaluation diagnostique des compétences

Des diagnostics de compétences de l'apprenant sont proposés par les systèmes d'évaluation diagnostique. Ces derniers ne se contentent pas de corriger les productions des apprenants et de comptabiliser des points. Le meilleur exemple de ce type de système est le logiciel *Pépité* [Delozanne, 2002] [Prévit et al., 2003].

Pépité : est un logiciel permet aux enseignants de diagnostiquer les compétences de leurs apprenants en algèbre élémentaire. C'est un outil logiciel composé de trois parties :

- *PepiTest* : qui permet aux apprenants de résoudre une liste de 22 exercices en algèbre. Ses questions sont de type fermées et ouvertes.

- *PepiDiag* : chargé de l'interprétation des réponses des apprenants, en utilisant des analyses didactiques élaborées en amont du projet.

- *PepiProf* : destiné pour établir le profil de chaque apprenant. Il permet de modifier les réponses des apprenants dans le cas où l'enseignant les trouverait contestables [Delozanne, 2002].

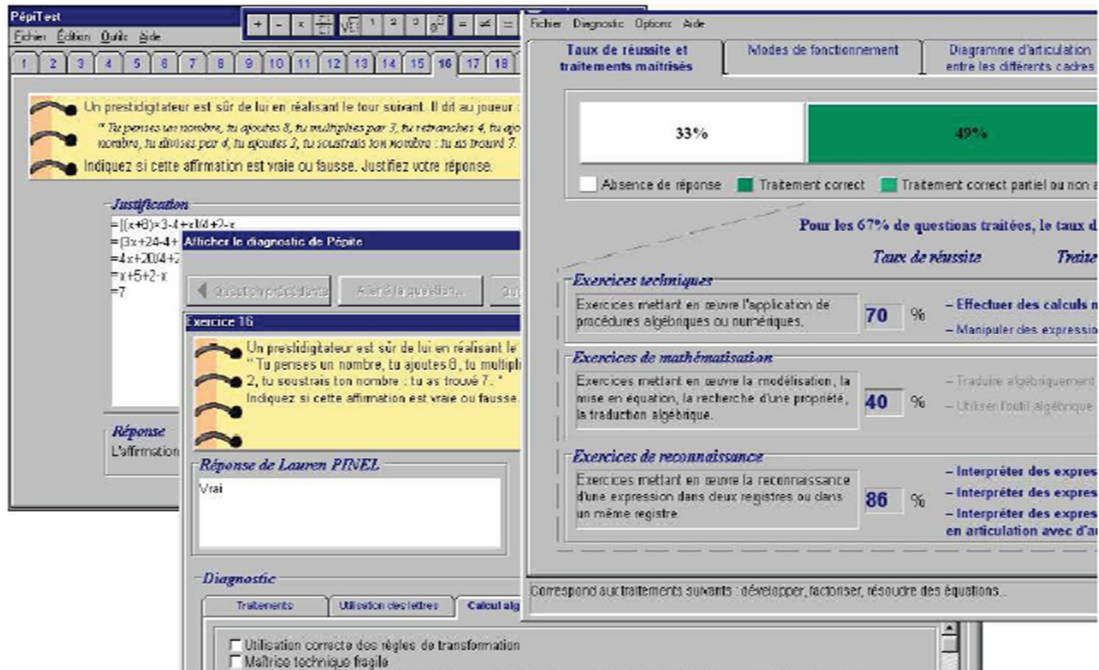


Figure 2.4 : Les trois composants de logiciel Pépite [Delozanne, 2002]

4.5. L'évaluation sommative et diagnostique

TDmaths est un exerciceur⁹ de mathématiques destiné aux apprenants de niveau collège. *TDmaths* intègrent les deux types d'évaluation, sommative et diagnostique. *TDmaths* prend aussi en compte l'évaluation normative. *TDmaths*, propose trois types d'évaluation [Souchard, 2003].

- Une évaluation sommative individuelle des performances assurée par des scores obtenus dans les exercices de mathématiques [Durand, 2006]. ;

- Une évaluation diagnostique des apprenants. Un diagnostic graphique est proposé à l'enseignant après la réalisation d'une évaluation des compétences transversales des apprenants. L'enseignant peut alors envisager des remédiations à partir de cette évaluation. Dans ce cas, l'évaluation s'inscrit alors dans un cadre formatif [Souchard, 2003].

- Une évaluation normative des apprenants. Le logiciel propose de positionner les résultats de ces évaluations par rapport à la classe. Que ce soit au niveau des scores ou des compétences transversales [Durand, 2006].

L'évaluation est une activité conviviale, on outre, comme elle joue différents rôles en EIAH, elle porte sur des objets différents. Si elle se concentre sur les productions des

⁹ Exerciceur : Logiciels proposant des énoncés et en mesure de valider ou d'invalider de manière interactive la réponse de l'apprenant.

apprenants, on l'appelle évaluation de production. Par contre, si elle porte sur la démarche mise en œuvre pour réaliser des productions, on parle d'évaluation de la démarche [Durand, 2006].

5. Évaluation de production : l'auto-évaluation

Dans certains cas, l'auto-évaluation des apprenants est utile pour favoriser l'autorégulation de leur apprentissage en s'insérant dans le cadre d'une évaluation formative [David, 2003]. GenEval (Générateur d'exercices d'auto-évaluation)[David, 2003] est un logiciel qui permet de générer des exercices d'auto-évaluation. Il permet à un formateur de créer des exercices similaires à ceux proposés dans une séance de travail dirigé. Ce logiciel propose une liste d'exercices, chaque exercice propose un cadre de résolution, avec des accès hypermédias à des informations relatives aux concepts étudiés, ainsi que des aides méthodologiques progressives. Pour chaque question de l'exercice, l'apprenant peut bénéficier ou non d'indications pour répondre. Une fois la réponse donnée est corrigée, l'apprenant évalue sa maîtrise des compétences mises en jeu dans les questions en se donnant une note. L'étudiant s'auto-évalue par rapport à la solution proposée par le formateur [David, 2003].

5.1. L'évaluation dans GenEval

GenEval propose deux types d'évaluation conjointement [Durand, 2006]:

- Une évaluation individuelle sommative. *GenEval* donne une correction à l'apprenant lorsque ce dernier répond à une question.
- Une auto-évaluation. Avec ce type d'évaluation, l'apprenant évalue sa maîtrise des compétences vue que cette évaluation favorise l'autorégulation de l'apprentissage.

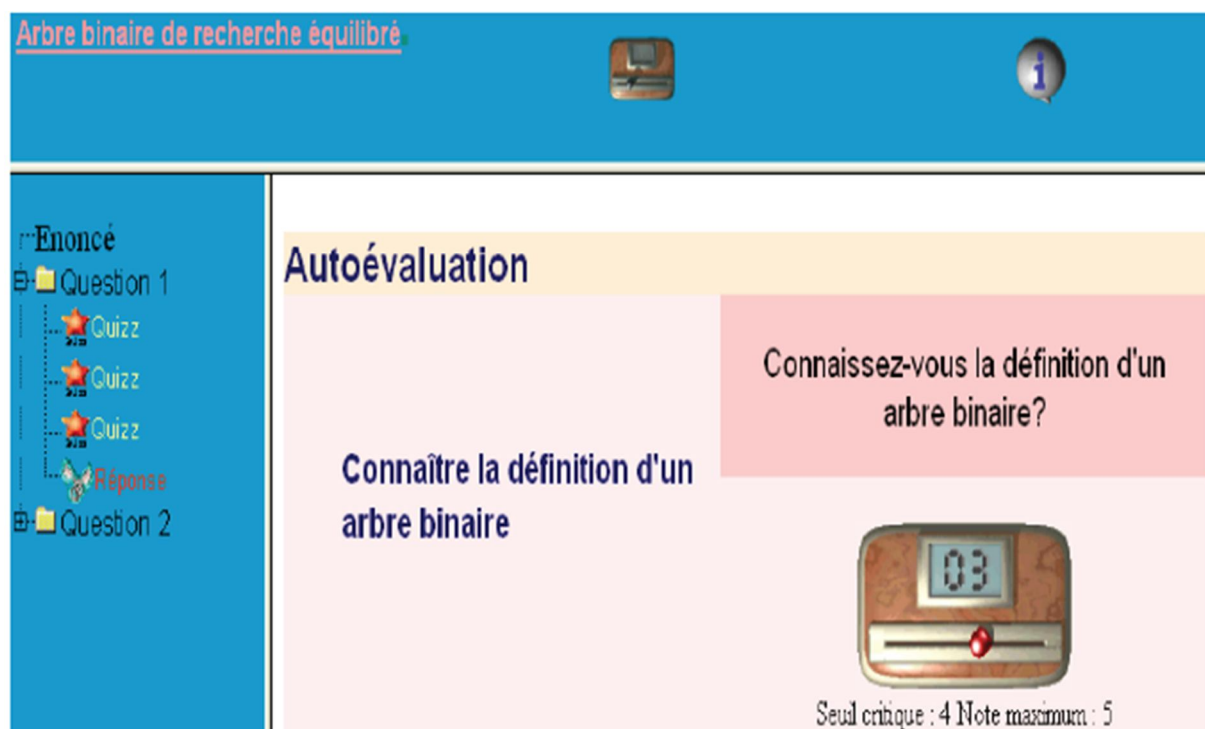


Figure 2.5: Le logiciel GenEval [Durand, 2006].

6. Évaluation de la démarche : l'assistance à l'évaluation

L'évaluation d'une production se trouve dans une opposition avec celle d'une démarche. Aussi, qu'il s'agisse d'activités individuelles ou collectives, elle propose des solutions d'assistance à l'évaluation. Elle fournit à l'enseignant des vues sur les activités par le biais d'indicateurs, de mesures, dans le dessein d'évaluer le déroulement de l'activité [Merceron, 2004]. Dans ce contexte, plus que des outils de mesure, il existe des systèmes qui proposent de superviser l'activité, le meilleur exemple est le système **GISMO**.

GISMO (Graphical Interactive Student Monitoring System) permet à l'enseignant de maîtriser ce qui se déroule durant l'activité pédagogique, et ce par le biais d'interfaces de visualisation d'indicateurs. Il utilise Moodle comme un système de gestion d'apprentissage [Mazza, 2005].

7. L'évaluation dans les EIAH dédiées à L'apprentissage des travaux pratiques à distance

Dans ce contexte, nous citons **MELBA** [Guibert, 2006], **Easyalgo**[Benabbou,2006]. Malheureusement, les travaux que nous citons s'intéressent à l'activité d'apprentissage qui est dépourvue de l'activité de l'évaluation des apprenants. Nous avons trouvé un seul travail, c'est la plate-forme **TEB (pour Travaux pratiques d'Électronique Binaire)**

[Tanana, 2009] qui a été développée dans l'objectif de gérer et d'évaluer des schémas électroniques des apprenants. Cette application est conçue pour trois catégories d'utilisateurs :

- l'enseignant, c'est la personne qui est chargée de proposer des sujets de travaux pratiques et d'évaluer des schémas de circuits électroniques réalisés par les apprenants, même en grand nombre ;
- aux apprenants, qui sont des acteurs responsables pour concevoir et réaliser, à distance ou en présentiel, des schémas de circuits électroniques numériques ;
- et aux administrateurs, pour la gestion des comptes utilisateurs (enseignant et apprenant).



Figure 2.6: L'interface de la plateforme TEB[Tanana, 2009]

7.1. L'évaluation dans la plateforme TEB

Une méthode d'évaluation formative du « savoir-faire » de l'apprenant fondée sur *TEB*. L'enseignant récupère les travaux des différents apprenants et lance le « moteur d'évaluation ». Le rôle de ce dernier permet à l'enseignant de comparer automatiquement son schéma de référence et tous ceux réalisés par les apprenants. Le moteur d'évaluation compare les deux tables de vérité « apprenant-enseignant » à l'aide d'un simulateur appelé

7.2.1. L'évaluation dans OASYS

Dans OASYS, l'apprenant peut évaluer trois autres apprenants dont les copies lui sont automatiquement attribuées si seulement s'il devient un évaluateur. Pour devenir un évaluateur, il doit répondre tout d'abord aux questions qui lui sont posées. Dans une phase préliminaire, l'enseignant définit les éléments sur lesquels l'apprenant va devoir se focaliser pour évaluer la réponse de chaque question. Un questionnaire qui apprécie le respect des critères sélectionnés par l'enseignant est mis à la disposition de l'apprenant, ce dernier doit utiliser ce questionnaire. Dans le processus d'évaluation, l'enseignant joue le rôle de modérateur. Il est prévenu si un écart entre les différentes évaluations d'une même copie est important [Ward, 2004].

Des nouvelles fonctionnalités sont intégrées dans la version **OASYS2**. Dans la phase d'évaluation, les apprenants peuvent travailler en groupe, un chat peut être introduit pour leur permettre d'échanger leurs points de vue. Ils évaluent ensemble les réponses d'autres groupes. À la fin du processus d'évaluation, chaque apprenant reçoit son résultat [Ward, 2004].

8. Synthèse des types d'évaluation dans les EIAH

L'évaluation en **EIAH** est une extension des méthodes et des techniques traditionnelles d'évaluation. Or nous pouvons dire que les pratiques référencées dans les **EIAH** couvrent les grandes catégories classiques d'évaluation. La table 2.2 le synthétise.

Types d'évaluation	Moment d'insertion	objectifs	EIAH	EIAH des TélÉTPs
pronostique	Avant l'apprentissage	Prédire, Contrôler l'accès à un cycle ou une année d'étude.	CAT(GRE)	
formative	Pendant l'apprentissage	Informé, Réguler l'activité de l'apprenant.	GISMO SPLACH	TEB E- TéléTPC@AALP
formatrice	Pendant l'apprentissage	Informe, Permettre à l'apprenant de réguler son apprentissage	GenEval APOM	OASYS
diagnostique	Juste avant un	Informé,	Pépite	

	apprentissage	Évaluer un niveau de compétences.	TDmaths	E-TéléTPC@AALP
Sommative	Après l'apprentissage	Certifier, Établir un bilan certifié Des résultats de l'apprenant.	CAT(TOEFL) TDmaths	E-ETéléTPC@AALP

Table 2.2: Les types d'évaluation dans les EIAH et les EIAH dédiées aux TéléTPs inspirés de [Durand, 2006] et étendus.

9. L'évaluation de l'apprenant dans notre plateforme :E-TéléTPC@AALP

E-TéléTPC@AALP [Boussaha, 2009] se veut un environnement pour l'enseignement et l'apprentissage à distance des travaux pratiques en programmation en informatique. Réalisée en 2009, elle possède une architecture de trois niveaux (téléformation, télé programmation, interface). La plateforme est développée en ASP/ MYSQL et fonctionne sur n'importe quel environnement.

9.1. Évaluation appariée : une nouvelle forme d'évaluation proposée

Comme nous l'avons cité plus haut, lorsqu'un enseignant veut évaluer un apprenant, il commence par déterminer l'objectif pédagogique visé. Ensuite, en fonction du niveau cognitif de cet objectif, il va choisir un ou plusieurs types d'exercice. Plus le niveau cognitif à évaluer est « haut », plus il faut laisser de liberté à l'apprenant.

Nous souhaitons évaluer le savoir-faire de l'apprenant qui correspond au niveau cognitif d'application de la taxonomie de Bloom. Nous allons donc nous intéresser à des outils adaptés à ce choix.

Nous proposons une nouvelle méthode d'évaluation qui permet d'assurer une évaluation appariée de l'apprenant dans l'activité d'apprentissage des TéléTPs basée sur les algorithmes d'appariement.

Le but de cette nouvelle évaluation est l'identification des incohérences et des différences entre les programmes produits par l'apprenant et ceux de référence décrivant une interprétation correcte du programme de l'enseignant. La première phase de cette méthode proposée est supportée par un outil que nous avons développé et qui permet de générer des diagrammes de classes et de communication à partir de programme C++. La

deuxième phase de la méthode proposée est le développement d'un algorithme d'appariement utilisé pour la comparaison entre les productions de l'apprenant et les productions de l'expert afin de pouvoir extraire premièrement la liste des couples similaires et différents entre de différents diagrammes et deuxièmement pour détecter les différentes erreurs produites par l'apprenant.

10. Conclusion

Dans la première partie, ce chapitre expose un état de l'art sur l'évaluation dans la littérature, ensuite le contexte des environnements informatiques. Dans cette partie sont tout d'abord introduites les différentes formes d'évaluation existantes ainsi que les points de vue des disciplines qui les étudient. Ce chapitre est suivi d'une synthèse sur l'existant en matière d'évaluation dans les EIAH, et plus particulièrement dans les EIAH des TÉLÉTPs (notre domaine de recherche). C'est un état de l'art étendu (étude de contexte d'évaluation dans Les EIAH, classement de ces EIAH selon le type de la technique d'évaluation utilisée). Nous terminons ce chapitre par l'introduction de notre méthode d'évaluation appariée. La méthode proposée est fondée sur un algorithme d'appariement des graphes. C'est ce concept que nous nous proposons d'examiner d'une façon plus détaillée.

Techniques d'Appariement de Graphes et Calcul de Similarité

1. Introduction

L'utilisation de graphes dans la représentation des connaissances a balayé de nombreux domaines d'application, y compris la bio-informatique, la chimie, la reconnaissance d'image, les réseaux sociaux, la recherche d'information. Etc. Ils sont devenus de plus en plus importants dans la modélisation des objets complexes et structurés, tels que les composants chimiques, les services Web, les images médicales ou autres, etc.. [Riesen et al., 2007] [Abbaci, 2013].

Dans notre cas nous utilisons les graphes pour la modélisation des programmes de l'apprenant et l'enseignant dans le domaine d'apprentissage des travaux pratiques à distance [Boussaha et al., 2012].

Notre domaine d'application étant la programmation orientée objet en informatique, nous essayons de modéliser chaque programme réalisé par un diagramme de classes et de communication UML, ensuite, on applique les techniques d'appariement sur ces graphes afin de les comparer avec ceux de référence.

Dans le présent chapitre, nous introduisons quelques notions sur les graphes en général, nous illustrons les techniques d'appariement et plus particulièrement les appariements entre les graphes. Nous présentons tout d'abord, les définitions et les termes employés dans le cadre du processus d'appariement de modèles puis nous qualifions les différents problèmes généraux d'appariement dans le contexte des graphes.

2. Les techniques d'appariement de graphes

2.1. Notions de base sur les graphes

2.1.1. Définition

Un graphe est une structure de données représentant une relation binaire (arêtes) entre des sommets [Abbaci, 2013]. Pour assurer une représentation efficace d'objets complexes et aussi une comparaison pertinente entre deux objets, les graphes sont devenus des outils de représentation très connus, ils véhiculent un maximum d'informations nécessaires pour modéliser ces objets.

Un objet complexe est modélisé par un graphe dans lequel les nœuds et les arêtes représentent respectivement ses entités (ses composants) et les relations entre ces entités.

La définition formelle d'un graphe est [Bonnici et al., 2013]:

Un graphe est un couple $G = (S, A)$ tel que

- S est un ensemble fini de sommets;
- $A \subseteq S \times S$ est une relation binaire sur S.

2.1.2. Type de graphes

2.1.2.1. Graphe (orienté / non orienté)

Soit un graphe $G = (S, A)$. Si $\forall (x, y) \in A, (y, x) \in A$,

Alors le graphe est dit non orienté et les éléments de A sont appelés arêtes du graphe. Dans ce cas, on note indifféremment une arête : $a \in A, (s, s') \in A$ avec s et s' dans S, ou encore (s', s) . Dans le cas contraire, le graphe est dit orienté et les éléments de A sont appelés arcs du graphe [Camille, 2014].

2.1.2.2. Graphe étiqueté

Un graphe étiqueté est un tuple $G = (S, A, E, e)$ tel que

- S est un ensemble fini de sommets ;
- $A \subseteq S \times S$ est un ensemble d'arêtes ;
- E est l'ensemble des différentes étiquettes possibles ;
- $e : S \cup A \rightarrow E$, est une fonction qui associe une étiquette à chaque sommet et chaque arête [Camille, 2014].

2.1.2.3. Graphe multi étiqueté

Les sommets et les arcs d'un graphe multi étiqueté, sont toujours associés à plusieurs étiquettes (labels) décrivant leurs propriétés. Étant donné LV (respectivement LE) un ensemble d'étiquettes de sommets (respectivement d'arcs), un graphe multi-étiqueté est défini par un triplet $G = (V, rV, rE)$ tel que [Auxepaules, 2009] :

- V est un ensemble de sommets ;
- $rV \subseteq V \times LV$ est une relation associant les sommets à leurs étiquettes, *i.e.*, rV est l'ensemble des couples (v, l) tels que le sommet v a pour étiquette l ;
- $rE \subseteq V \times V \times LE$ est une relation associant les arcs à leurs étiquettes, *i.e.*, rE est l'ensemble des triplets (v, v', l) tels que l'arc (v, v') a pour étiquette l ;
- l'ensemble E des arcs est défini par $E = \{(v, v') / \exists l, (v, v', l) \in rE\}$.

2.1.2.4. Graphe simple

Un graphe est dit simple s'il ne comporte pas de boucles (sommets reliés à eux-mêmes par une arête) ni de couples de sommets connectés par plusieurs arêtes distinctes [Müller, 2011].

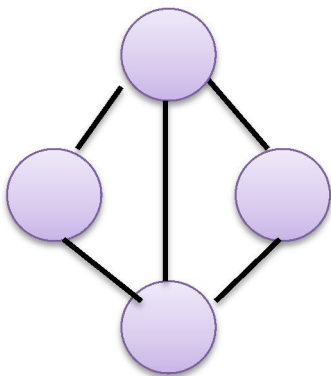


Figure 3.1 : Un graphe non-étiqueté non-orienté.

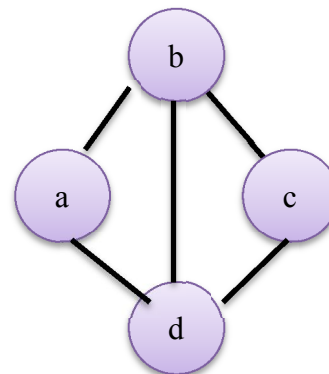


Figure 3.2 : Un graphe étiqueté non-orienté.

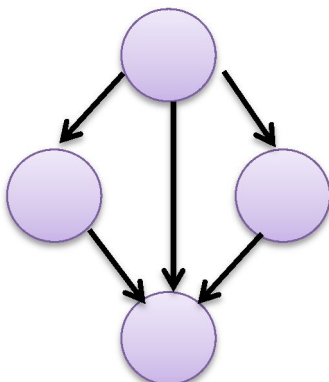


Figure 3.3 : Un graphe non-étiqueté orienté.

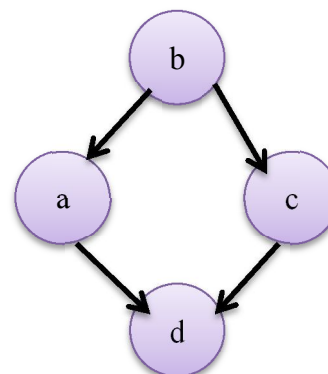


Figure 3.4 : Un graphe étiqueté orienté.

2.2. Appariement des graphes

2.2.1. Définition de l'appariement

Nous avons pu constater que la terminologie relative au problème d'appariement diffère entre les différents domaines et même au sein d'un même domaine. Cela dépend de la nature des modèles à analyser (ontologies, schémas, graphes).

2.2.1.1. Appariement des ontologies

L'appariement des ontologies (*ontology matching*) est le processus de découverte des relations (ou correspondances) entre les entités de différentes ontologies [Euzenat & Shvaiko 2007]. Dans ce domaine, nous trouvons, le terme *alignement* est le plus couramment utilisé. L'alignement d'ontologies met en évidence les relations sémantiques de plusieurs ontologies à confronter (équivalence, subsomption, incompatibilité, etc.). L'expression des correspondances, appelée aussi alignement, peut par la suite être utilisée par exemple pour fusionner les ontologies, migrer des données entre ontologies ou traduire des requêtes formulées en fonction d'une ontologie vers une autre [Djoufak Kengue et al. 2008].

2.2.1.2. Appariement de graphes

L'appariement de graphes (*graph matching*) consiste à mettre en correspondance les sommets (noeuds) de deux graphes, l'objectif étant généralement de comparer les objets modélisés par les graphes [Bengoetxea, 2002].

2.2.1.3. Appariement de schémas

L'appariement de schémas (*schema matching*) consiste à trouver les correspondances sémantiques (i.e. appariements) entre deux schémas [Nottelmann, 2006] [Do & Rahm 2007]. L'appariement peut être considéré comme une opération ou un opérateur (match) qui prend deux schémas en entrée et produit un mapping entre les éléments des deux schémas correspondant sémantiquement les uns aux autres [Rahm & Bernstein 2001].

2.2.1.4. Appariement de formes ou de motifs

L'appariement de forme ou de motif (pattern matching) est la mise en correspondance de formes en se fondant sur un ensemble prédéfini de règles ou de

critères. L'appariement de formes se ramène à un problème de filtrage [Auxepaules, 2009]. Dans un programme informatique, le filtrage par motif (une autre nomination de pattern matching) est le processus de vérifier la présence des composants d'un motif donné. À la différence de la reconnaissance de formes (pattern recognition), les motifs concernent conventionnellement des séquences ou des arbres et sont spécifiés strictement. Pour vérifier qu'un objet filtré a une structure désirée, le filtrage par motif est utilisé, pour trouver une structure appropriée, pour retrouver des parties alignées ou pour substituer les motifs reconnus par quelque chose d'autre [Auxepaules, 2009].

Dans notre cas, on s'intéresse à l'appariement de graphe, on considérerait chaque diagramme que ce soit de classes ou de communication (engendrés à partir de programme orienté objet) comme un graphe étiqueté.

2.2.2. Processus de l'appariement

Le processus d'appariement peut être vu comme une fonction f qui, à partir d'une paire de modèles à appairer m et m' , un alignement en entrée A , un ensemble de paramètres p et un ensemble de ressources externes s , retourne un alignement A' entre ces modèles [Hinnoutondji, 2011].

$$A' = f(m, m', A, p, s)$$

Le processus d'appariement peut être représenté schématiquement comme c'est le cas dans la figure 3.5 suivante.

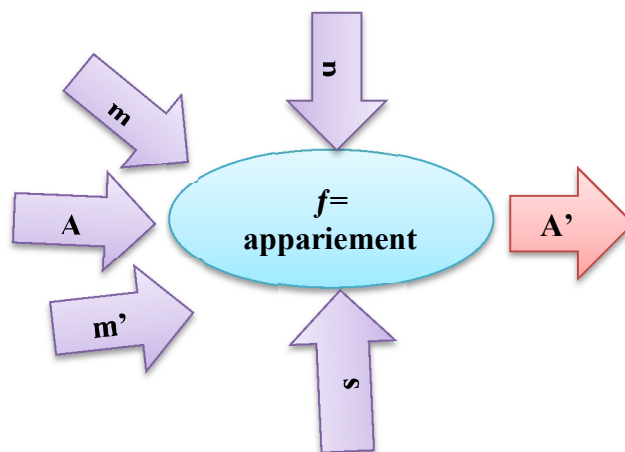


Figure 3.5: Le processus d'appariement adapté et inspiré de [Auxepaules, 2009].

Dans le cas où le processus d'appariement serait mis en correspondance entre plusieurs modèles, cet appariement est nommé appariement multiple (multiple matching).

Alors ce dernier est défini formellement comme le premier sauf dans ce dernier cas la fonction a plusieurs modèles en entrés [Auxepaules, 2009].

$$A' = f(m1, \dots mn, A, p, s)$$

2.2.3. Le résultat d'un processus d'appariement de graphe

Appariement, alignement ou mapping est le résultat du processus d'appariement (match result), indique quels éléments des modèles en entrée correspondent les uns aux autres [Hinnoutondji, 2011]. Par conséquent, on peut dire qu'une correspondance est définie par les deux entités reliées (qui peuvent être des classes, des instances, des propriétés, des termes, mais aussi des combinaisons complexes de ceux-ci), la relation liant ces entités (équivalence, subsomption, incompatibilité, etc.) et si possible une mesure de confiance dans cette correspondance [Hinnoutondji, 2011].

Pour comparer deux graphes, un processus d'appariement est donc utilisé pour effectuer une mise en correspondance de leurs nœuds et/ou de leurs arêtes.

2.2.4. Les types d'appariement de graphes

Les types d'appariement qui existent dans la littérature sont classifiés soit selon le critère de cardinalité ou suivant les caractéristiques des contraintes de l'appariement.

2.2.4.1. Type d'appariement selon les cardinalités

2.2.4.1.1. Appariement bijectif : cardinalité = (1, 1)

Tous les sommets d'un graphe sont appariés à un et un seul sommet de l'autre graphe et inversement. Il n'existe pas donc deux sommets d'un graphe liés à un même sommet de l'autre graphe [Thi Hong Hiep, 2009].

2.2.4.1.2. Appariement injectif : cardinalité = (1, 0..1)

Un appariement injectif de G1 et G2 lie chaque sommet du graphe G1 à au maximum un sommet du graphe G2. Il est impossible d'apparier deux sommets de G1 à un même sommet de G2 [Thi Hong Hiep, 2009].

2.2.4.1.3. Appariement univoque : cardinalité = (0..1, 0..1)

Dans un appariement univoque entre deux graphes G1 et G2, chaque sommet d'un graphe est apparié au maximum à un sommet de l'autre graphe. Chaque sommet n'a donc

que deux possibilités : soit il n'est pas apparié, soit il est apparié à un sommet unique [Jouili, 2008].

2.2.4.1.4. *Appariement multivoque* : cardinalité = $(0..|V1|, 0..|V2|)$

Chaque sommet d'un graphe est apparié à un ensemble (éventuellement vide) de sommets de l'autre graphe. Dans un appariement multivoque, un sommet est peut-être apparié en même temps à plusieurs sommets [Jouili, 2008].

Un appariement multivoque d'un sommet d'un graphe avec plusieurs autres sommets de l'autre graphe se nomme généralement *éclatement* de sommets (cardinalité d'appariement 1:n). Inversement, un appariement multivoque de plusieurs sommets associés à un seul sommet de l'autre graphe est une *fusion* de sommets (cardinalité d'appariement n:1) [Sorlin et al., 2006].

2.2.4.2. Type d'appariement selon les caractéristiques de contraintes

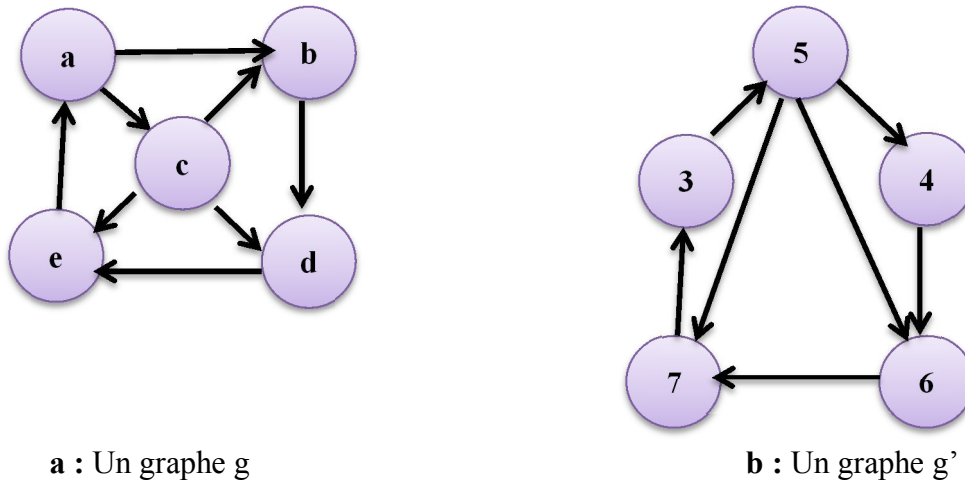
2.2.4.2.1. *L'appariement exact de graphes*

Ce type d'appariement impose des contraintes obligatoires sur le processus de mise en correspondance entre deux graphes. L'objectif d'un appariement exact est de déterminer si deux graphes, ou des parties de deux graphes sont identiques suivant leurs structures et leurs étiquettes [Sorlin et al., 2006]. Cette égalité entre deux graphes est alors définie par une fonction bijective, dite isomorphisme de graphe [Hinnoutondji, 2011].

2.2.4.2.1.1. *Isomorphisme de graphe*

Ce type d'appariement permet de vérifier si deux graphes sont structurellement identiques [Zaslavskiy, 2010].

Le graphe g est dit isomorphe au graphe g' , s'il existe un isomorphisme (une fonction bijective) entre les deux graphes qui permet de retrouver une mise en correspondance entre tous les noeuds et/ou arêtes du graphe source et du graphe cible tout en respectant les fonctions d'étiquetage [Gaillard, 2011]. L'isomorphisme de graphes consiste donc à déterminer une égalité entre deux graphes en termes de structure et d'étiquettes [Abbaci, 2013]. La figure 3.6 donne un exemple de deux graphes isomorphes en appliquant une fonction bijective f telle que : $f(a) = 3$, $f(b) = 4$, $f(c) = 5$, $f(d) = 6$ et $f(e) = 7$.



a : Un graphe g

b : Un graphe g'

Figure 3.6: Exemple d'isomorphisme de graphe inspiré et adapté de [Abbaci, 2013].

2.2.4.2.1.2. Isomorphisme de sous graphes

Permet de vérifier si un graphe est « inclus » dans un autre [Zaslavskiy, 2010]. Cet appariement permet de retrouver une partie du graphe cible identique au graphe source. Autrement dit, le problème est ramené à celui de trouver un isomorphisme de sous-graphe entre le graphe source et le graphe cible [Sorlin et al., 2006].

Dans certains cas, la recherche des objets contenant un autre objet peut être plus intéressant. Prenant par exemples, en chimie organique, les chimistes ont parfois besoin de trouver toutes les molécules complexes contenant une autre molécule afin d'étudier ses comportements et ses propriétés chimiques. Dans les services Web, il est parfois nécessaire de rechercher un service particulier contenu dans d'autres services Web. En reconnaissance d'images, la recherche peut être réalisée en récupérant toutes les images qui contiennent un objet particulier [Abbaci, 2013].

Dans les cas cités précédemment, on doit appliquer un appariement de sous graphes pour démontrer si le graphe de plus petite taille est inclus dans le graphe de plus grande taille.

Un isomorphisme de sous-graphe existe entre deux graphes g et g', s'il existe un sous-graphe g'' de g' (i.e. g'' appartient à g') tel que g'' et g sont isomorphes de graphe. Autrement dit, l'isomorphisme de sous-graphe entre deux graphes indique que le graphe de plus petite taille est inclus dans le graphe de grande taille [Bonnici et al., 2013].

Un exemple d'isomorphisme de sous-graphe est illustré dans la figure 3.7 en appliquant une fonction injective f telle que : $f(a) = 1$, $f(b) = 2$, $f(c) = 3$, $f(d) = 4$ et $f(e) = 5$.

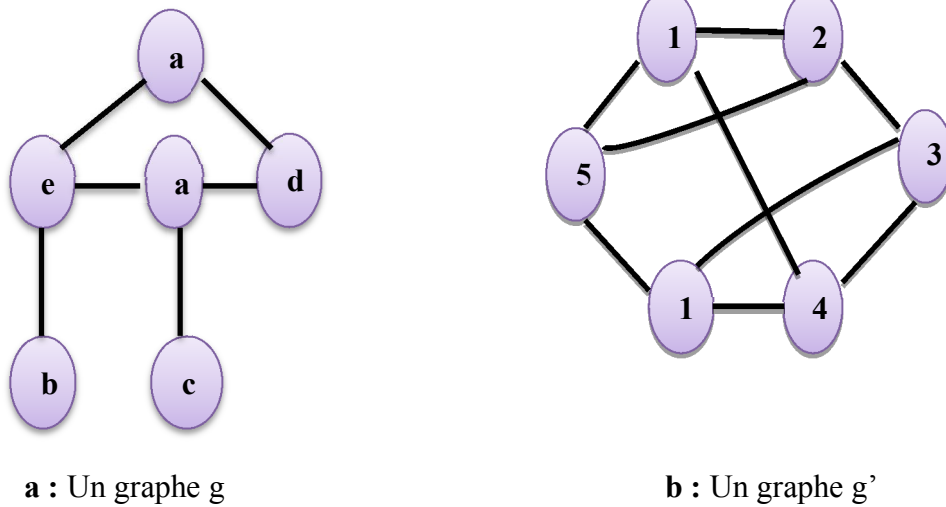


Figure 3.7: Exemple d'isomorphisme de sous-graphe [Abbaci, 2013].

2.2.4.2.2. L'appariement approximatif de graphes

Les contraintes utilisées dans ce type d'appariement sont plus souples, de telle sorte, il permettra une mise en correspondance avec une certaine tolérance d'erreurs. Dans l'appariement approximatif de graphes, des dissemblances entre un certain nombre de nœuds et/ou d'arêtes, ainsi que des suppressions et des insertions de nœuds et/ou d'arêtes des deux graphes comparés sont tolérées [Hinnoutondji, 2011].

La détermination d'une correspondance entre deux graphes avec certaines tolérances est l'objectif d'un appariement approximatif de graphes. Prenant les deux graphes g et g' de la figure 3.8, ces deux graphes sont considérés similaires si l'on applique le processus d'appariement approximatif de sous-graphes, parce qu'il évalue leur ressemblance avec tolérance de certaines de leurs différences. Également il est clair que ces deux graphes sont très similaires, au regard de leurs nœuds et arêtes qui sont identiques. Néanmoins, ces graphes ne sont pas isomorphes de (sous-)graphes. Ainsi, lors de l'application d'un appariement exact, les graphes sont considérés comme différents [Zaslavskiy, 2010].

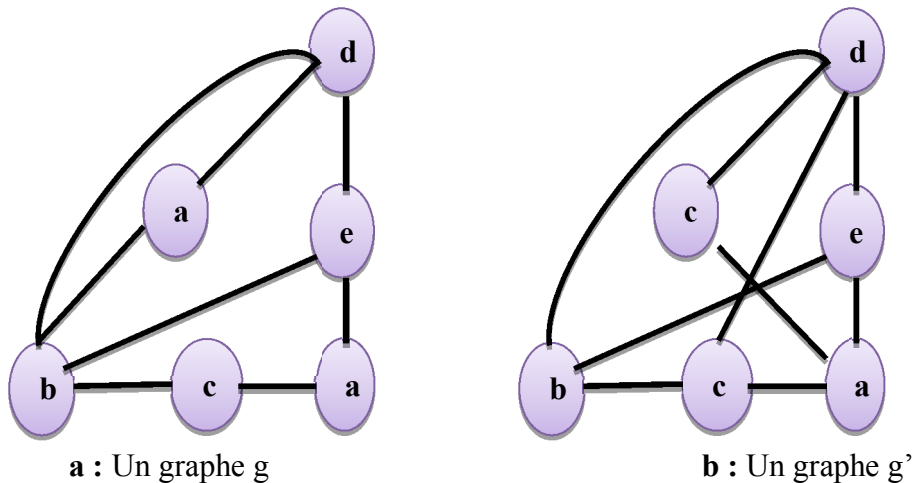


Figure 3.8: Exemple de graphes similaires [Abbaci, 2013].

La comparaison de deux graphes à l'aide d'un processus d'appariement approximatif de graphes ou de sous-graphes exige une méthode de calcul de similarité ou de distance entre graphes.

- *la mesure de similarité* : consiste à identifier et à quantifier les points communs entre deux objets.
- *la mesure de distance* : est l'identification et la quantification des différences entre deux objets.

La distance et la similarité sont deux concepts duaux dont l'objectif est commun, détermination des différences et/ou des similarités entre deux objets. Nous nous intéressons dans notre travail, dans le contexte des graphes, au calcul de la distance ou de la similarité de deux structures de graphes pour chercher :

- **le meilleur appariement** : dans le cas d'appariement approximatif.
- **une correspondance ou une inclusion** : dans le cas d'appariement exact.

Dans la littérature, plusieurs méthodes ont été proposées. Dans la sous-section suivante, nous présentons, en détail, les mesures les plus utilisées pour calculer la similarité entre deux graphes.

2.2.4.2.2.1. Plus grand sous graphe commun

Le plus grand sous-graphe commun de deux graphes : g_1 et g_2 est le plus grand graphe qui soit un sous-graphe isomorphe de g_1 et g_2 . La taille du sous-graphe commun pouvant être définie par le nombre de sommets et/ou le nombre d'arcs [Solon, 2005].

Le principe est le suivant : g_3 (figure 3.9) est le plus grand sous-graphe commun à deux graphes g_1 et g_2 (noté $mcs(g_1, g_2)$ pour maximum common subgraph), si g_3 est un sous-graphe commun à g_1 et g_2 ayant un nombre de sommets maximum, c'est-à-dire qu'il

n'existe aucun autre sous-graphe commun à g_1 et g_2 avec un ordre (ou nombre de sommets) strictement plus grand que celui de g_3 .

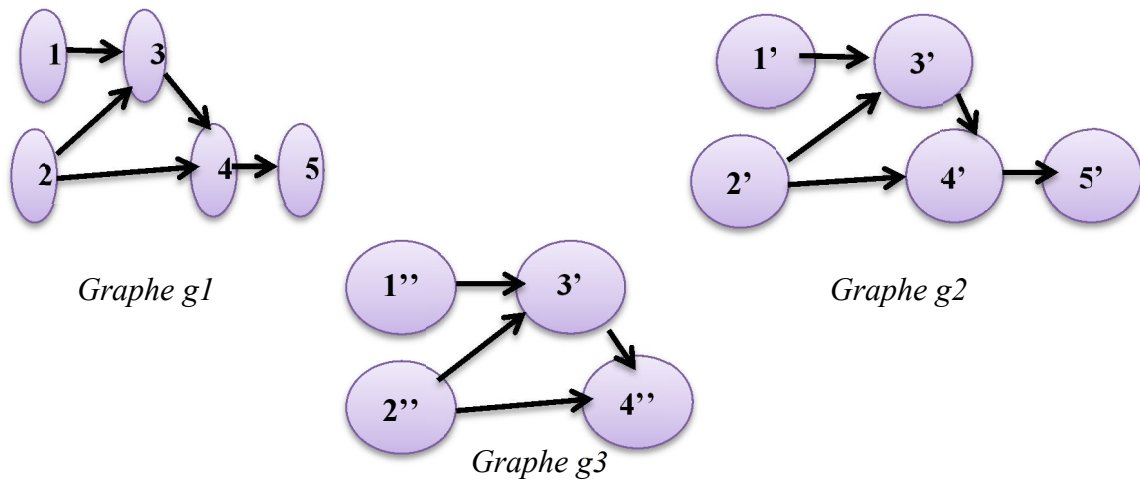


Figure 3.9: Le plus grand sous graphe commun [Tanana, 2009]

Dans [Bunke, 2006], Bunke propose une définition d'une mesure de similarité entre deux graphes basée sur la méthode du plus grand sous-graphe commun par l'équation(1) suivante :

$$d(g_1, g_2) = 1 - \frac{|mcs(g_1, g_2)|}{\max(|g_1|, |g_2|)} \quad (1)$$

où $|mcs(g_1, g_2)|$ est le nombre de sommets du plus grand sous-graphe commun à g_1 et g_2 , $|g_1|$ (respectivement $|g_2|$) est le nombre de sommets du graphe g_1 (respectivement g_2).

2.2.4.2.2.2. Distance d'édition de graphes

Cette méthode est très largement utilisée pour déterminer des appariements de graphes tolérants aux erreurs. Dans [Bunke, 2006], Bunke a proposé plusieurs algorithmes de mesure de similarité basés sur la distance d'édition de graphes ou « Graph Edit Distance (GED)».

Le principe de ces algorithmes est le suivant : soient deux graphes g_1 et g_2 , l'idée est de transformer g_1 en g_2 , en appliquant une suite d'opérations d'édition telles que : insertion, substitution et suppression de sommets ou d'arcs sur le premier graphe pour obtenir le deuxième graphe. Autrement dit, ce processus tend à déformer le modèle du graphe initial jusqu'à ce qu'il s'identifie avec le modèle de l'autre graphe [Bunke, 2006].

Chaque transformation est associée d'un coût prédéfini, le coût d'une séquence de transformations est égal à la somme pondérée des coûts de chaque transformation. La

séquence de transformation qui obtient un coût minimal représente la distance d'édition entre les deux graphes. La formule de calcul de similarité proposée par Bunke est illustrée par l'équation (2) suivante.

$$d(g1, g2) = \min\left\{\sum_{i=1}^{K=card(t)} c(ei)\right\} \quad (2)$$

avec $t \in E(g1, g2)$, « ei » est appelée une opération d'édition. « $t = (e1, \dots, ek)$ » est une séquence d'opérations d'édition transformant $g1$ en $g2$.

3242221

2.2.4.2.2.1. Les opérations d'édition

Une opération d'édition transforme partiellement un graphe en un autre graphe. Une transformation est effectuée soit sur un sommet ou sur un arc. Il existe six types de transformations possibles (substitution / suppression / insertion de sommet, substitution / suppression / insertion d'arc), Tanana, en 2009, dans [Tanana, 2009] expliquait en détail ces opérations de transformation.

Dans les sections précédentes de notre chapitre, nous avons expliqué en détail plusieurs types d'appariements qui sont largement utilisés pour comparer deux graphes tels que: l'isomorphisme de graphes qui consiste à vérifier que deux graphes sont identiques, l'isomorphisme de sous-graphes qui permet de vérifier qu'un graphe est inclus dans un autre, et la recherche du plus grand sous-graphe commun, qui est utilisé pour évaluer l'intersection de deux graphes et la distance d'édition de graphe. Cependant, tous les types d'appariements cités en dessus souffrent d'un inconvénient majeur, ils ne sont pas complètement satisfaisants pour mesurer la similarité d'objets. En particulier, ils ne permettent pas d'apparier un sommet d'un graphe avec un ensemble de sommets de l'autre graphe, et ils ne permettent pas de personnaliser facilement la mesure aux objectifs de l'utilisateur.

Nous introduisons dans la section suivante une nouvelle mesure pour évaluer la similarité de graphes proposée par Pierre-Antoine Champin et Christine Solnon [sorlin et al. ,2003].

Notre étude approfondie et critique de cette similarité nous a permis de déduire les avantages qui la distinguent en la comparant avec les autres cités ci-dessus. Ces avantages sont :

- elle permet d'identifier et d'expliquer les différences et les caractéristiques communes entre deux graphes.
- cette mesure est directement proportionnelle aux nombres de caractéristiques communes aux deux graphes par rapport au nombre total de caractéristiques.
- elle permet également de comparer des graphes décrivant des objets à différents niveaux de granularité, car elle est fondée sur des appariements multivoques (permettant d'associer un sommet d'un graphe à un ensemble de sommets de l'autre graphe).
- elle est calculée par rapport à un appariement.
- enfin, cette mesure est générique et elle est paramétrée par deux fonctions qui permettent d'intégrer des connaissances spécifiques à l'application.

Un graphe étiqueté est caractérisé par ses composants et les étiquettes qui leurs sont attachées. Chaque couple (sommet, étiquette) ou (arête, étiquette) est considéré comme une caractéristique du graphe. La similarité de graphes proposée par Pierre-Antoine Champin et Christine Solnon [sorlin et al. ,2003] est calculée par la formule illustrée dans l'équation (3).

$$sim(g1, g2) = \max \frac{f(descr(g1) \cap_m descr(g2)) - g(splits(m))}{f(descr(g1) \cup_m descr(g2))} \quad (3)$$

où $(descr(g1) \cap_m descr(g2))$: est l'ensemble des caractéristiques communes entre les deux graphes $g1$ et $g2$.

$(descr(g1) \cup_m descr(g2))$: est l'ensemble de toutes les caractéristiques de deux graphes.

$splits(m)$: ensemble de sommets appariés a plusieurs sommets.

f : fonction de quantification des caractéristiques.

g : fonction de quantification des splits.

m : le meilleur appariement engendré.

Donc, l'idée principale de calcul de cette mesure est de trouver le meilleur appariement qui maximise la similarité entre $g1$ et $g2$. Donc c'est un problème d'optimisation combinatoire. Dans la section suivante nous détaillons quelques algorithmes qui permettent de trouver le meilleur appariement.

2.3. Les algorithmes de recherche d'un meilleur appariement

2.3.1. Algorithme glouton

Le premier algorithme proposé est un algorithme glouton, il retourne un appariement localement optimal et peut être exécuté plusieurs fois pour retourner le meilleur appariement trouvé. Il commence à partir d'un appariement vide et ajoute itérativement des couples de sommets choisis dans l'ensemble des candidats jusqu'à ce que l'ajout d'aucun autre couple ne puisse augmenter la similarité. À chaque étape, le couple à ajouter est choisi aléatoirement parmi l'ensemble des couples qui augmentent le plus la similarité [Sorlin et al., 2003].

2.3.2. Algorithme de Recherche locale Taboue réactive (RTS, Reactive Tabu Search).

Le second algorithme est fondé sur une recherche locale taboue réactive (RTS, Reactive Tabu Search) [Sorlin, 2003] permettant d'améliorer une solution courante en explorant son voisinage. Les voisins d'un appariement m sont les appariements obtenus en ajoutant ou en supprimant un seul couple de sommets à m . En démarrant à partir d'un appariement initial (en général défini à l'aide d'un algorithme glouton), une recherche locale explore l'espace de recherche en se déplaçant de voisin en voisin jusqu'à l'obtention de la solution optimale ou jusqu'à un nombre maximal d'itérations autorisé [Sorlin, 2003] [Sorlin et al b.,2006]. À chaque itération, le prochain voisin à explorer est choisi selon une heuristique. Le voisin qui maximise la similarité est toujours sélectionné en premier. Une liste taboue est utilisée pour mémoriser les k derniers mouvements réalisés afin d'interdire les mouvements inverses durant k itérations et ainsi ne pas rester autour d'un maximum local en réalisant toujours les mêmes mouvements [Sorlin, 2003][Sorlin et al b., 2006] .

Pour rendre la recherche réactive, la longueur de la liste taboue est dynamiquement adaptée pendant la recherche. Il faut évaluer les besoins de diversification pendant la recherche : cela permet de détecter la redondance, quand un même appariement est exploré deux fois.

Pour chaque appariement visité, une clé de hachage est mémorisée. Si une collision survient dans la table de hachage, cela signifie que l'appariement courant a probablement déjà été exploré et la liste de taboue est allongée. Si au contraire il n'y a pas eu de collisions pendant un certain nombre de mouvements (signe que la recherche ne stagne pas autour d'un optimum local), la liste taboue est raccourcie [Sorlin et al b., 2006].

2.3.3. Algorithme d'optimisation par colonie de fourmis

Les fourmis sont capables de résoudre d'une manière collective des problèmes complexes, comme trouver le plus court chemin entre deux points dans un environnement accidenté. Pour cela, elles communiquent entre elles de façon locale et indirecte, grâce à une hormone volatile, appelée phéromone [Solon, 2005]. Au cours de sa progression, une fourmi laisse derrière elle une trace de phéromone, qui augmente la probabilité que d'autres fourmis passant à proximité choisissent le même chemin. Ce mécanisme de résolution collective de problèmes est à l'origine des algorithmes à base de fourmis artificielles comme il est proposé dans l'algorithme proposé par [Solon, 2005] [Sorlin et al b., 2006].

Cet algorithme appelé ACO (Ant Colony Optimization) ou l'optimisation par colonie de fourmis consiste à reformuler le problème à résoudre en un problème de recherche d'un chemin optimal dans un graphe (appelé graphe de construction) et à utiliser des fourmis artificielles pour trouver les bons chemins de ce graphe [Sorlin.2005] [Sammoud, 2006]. À chaque itération de l'algorithme, chaque fourmi de la colonie construit aléatoirement un chemin du graphe (une solution du problème) et de la phéromone est déposée sur les meilleurs chemins découverts lors de ce cycle [Sammoud, 2006][Sorlin.2005]. Lors des itérations suivantes, les fourmis construisent de nouveaux chemins avec une probabilité dépendant de la phéromone déposée lors des cycles précédents et d'une heuristique propre au problème considéré. La colonie de fourmis converge alors peu à peu vers les meilleures solutions [Sorlin et al b., 2006] [Solon, 2005].

2.4. Classification des techniques d'appariement

Euzenat et Shvaiko, dans [Euzenat & Shvaiko, 2007] expliquaient en détail les différentes approches d'appariement. Ces dernières sont classées selon les deux classifications suivantes :

- *Granularity/Input Interpretation classification* : cette classification est fondée sur la granularité des apparieurs (niveau élément ou structure) et ensuite sur la manière dont les techniques interprètent les informations fournies.
- *Kind of Input classification* : cette deuxième classification est basée sur la catégorie d'entrées qui est utilisée par des techniques élémentaires d'appariement.

2.5. Types d'apparieurs

Il faut sélectionner les apparieurs en fonction du domaine d'application et des types de modèles car dans un processus d'implantation d'appariement, il est possible d'utiliser plusieurs types d'apparieurs. Cela entraîne la combinaison de plusieurs approches d'appariement qui peuvent être évaluées simultanément ou dans un ordre spécifique. L'utilisation de plusieurs apparieurs induit deux sous problèmes [Auxepaules, 2009] :

D'un côté, il y a la réalisation des apparieurs simples qui calculent un appariement en se fondant sur un seul critère d'appariement. D'un autre côté, il faut définir la manière de combinaison des apparieurs simples dans le système complet. Cette combinaison peut se faire sous la forme d'un apparieur hybride qui intègre des critères multiples d'appariement (par exemple, une égalité des noms et une égalité des types) ou bien sous la forme d'un apparieur composite qui combine les résultats produits par différents algorithmes simples d'appariement. Cette classification sous forme d'apparieurs simples et combinés est illustrée par la figure 3.10 [Auxepaules, 2009].

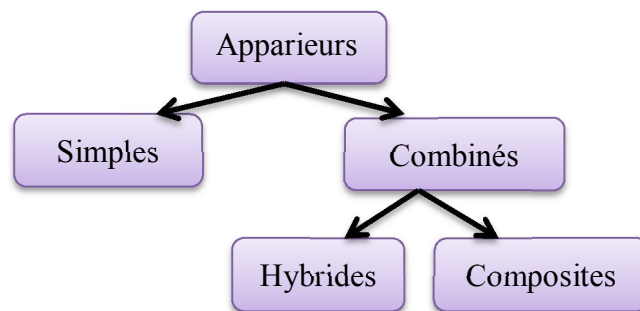


Figure 3.10 : Différents types d'apparieurs

2.5.1. Apparieurs simples

Les apparieurs simples calculent un appariement en se fondant sur un seul critère d'appariement.

2.5.2. Apparieurs hybrides

Les apparieurs hybrides combinent directement plusieurs approches d'appariement pour déterminer les candidats à l'appariement en se fondant sur des critères et des sources d'informations multiples [Auxepaules, 2009]. Ils peuvent fournir des appariements meilleurs et avec une meilleure performance qu'en exécutant de manière séparée plusieurs

apparieurs (réduction du nombre de passes effectuées sur les modèles [Auxepaules, 2009] [Meijer, 2008]).

2.5.3. Apparieurs composites

Les apparieurs composites combinent les résultats évalués indépendamment par plusieurs apparieurs dont des apparieurs hybrides. Un apparieur composite peut permettre un ordonnancement flexible des apparieurs. Il permet de les exécutés soit simultanément et /ou soit séquentiellement. Cette capacité à combiner les apparieurs est plus flexible que la combinaison rigide de techniques particulières d'appariement exécutées simultanément ou dans un ordre fixe dans les apparieurs hybrides [Auxepaules, 2009].

Dans notre travail, nous appliquons tous ces concepts d'appariement sur les diagrammes UML (de classes et de communication) qui représentent une interprétation indirecte des programmes orientés objets et qui sont considérés comme des graphes.

Dans la section suivante, nous présentons la représentation des diagrammes UML sous forme de graphes.

2.6. Représentation des diagrammes UML sous forme de graphes UML

Nous pouvons transformer un diagramme de classes UML en graphe UML, on exprime directement, tous les éléments UML et leurs liens explicites à l'aide de nœuds et d'arcs. Ces derniers sont les héritages, les relations de propriété et de typage entre les différents éléments. Par contre, la majorité des éléments de modélisation (classificateurs, associations, attributs, fins d'association et opérations) sont des sommets de graphe, les classes (des sommets) possèdent (des arcs) des attributs (des sommets) qui peuvent être typés (arc) par celles-ci. Les associations possèdent des extrémités d'association (sommets) qui peuvent être typées (arc) par des classes (sommets) [Auxepaules, 2009].

Les dernières représentations sous forme de graphe expriment clairement les liens entretenus au niveau de leurs éléments et de leurs caractéristiques dans les diagrammes UML. Chaque arc relie deux nœuds et est associé d'une direction. Les nœuds et les arcs sont caractérisés au moyen d'un type et d'étiquettes représentant leurs attributs (nom, visibilité, abstraction, agrégation, multiplicités) [Auxepaules, 2009].

2.7. Implémentation d'un programme orienté objet C++ en un modèle objet et vice versa

2.7.1. Diagramme de classes

2.7.1.1. Classe



```

Class Chambre {
private :
    int prix ;
protected :
    int étage ;
Public :
    void reserve (String nom) {} ;
    Bool estvide () {} ;
};
    
```

Figure 3.11: Implémentation d'une classe en C++

2.7.1.2. Héritage

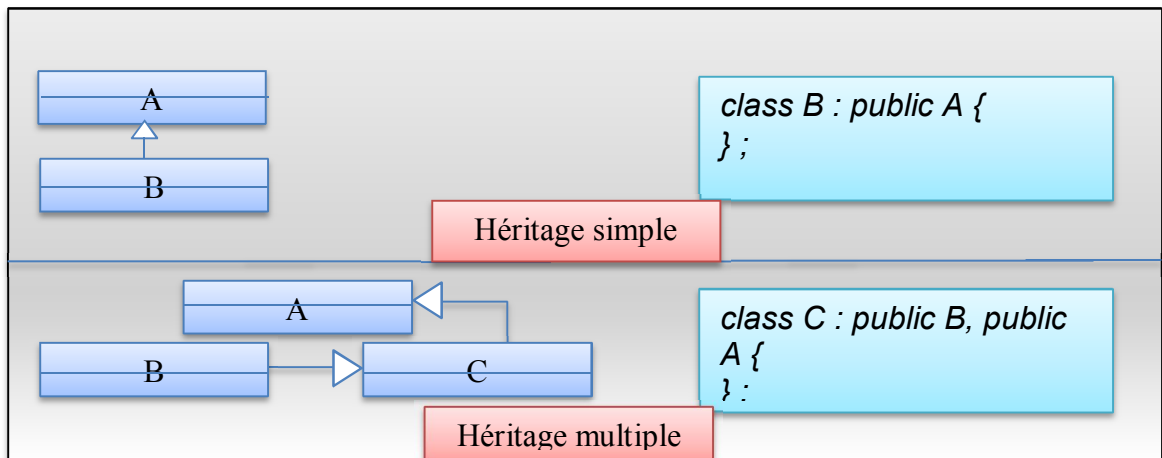


Figure 3.12: Implémentation d'héritage en C++

2.7.1.3. Agrégation

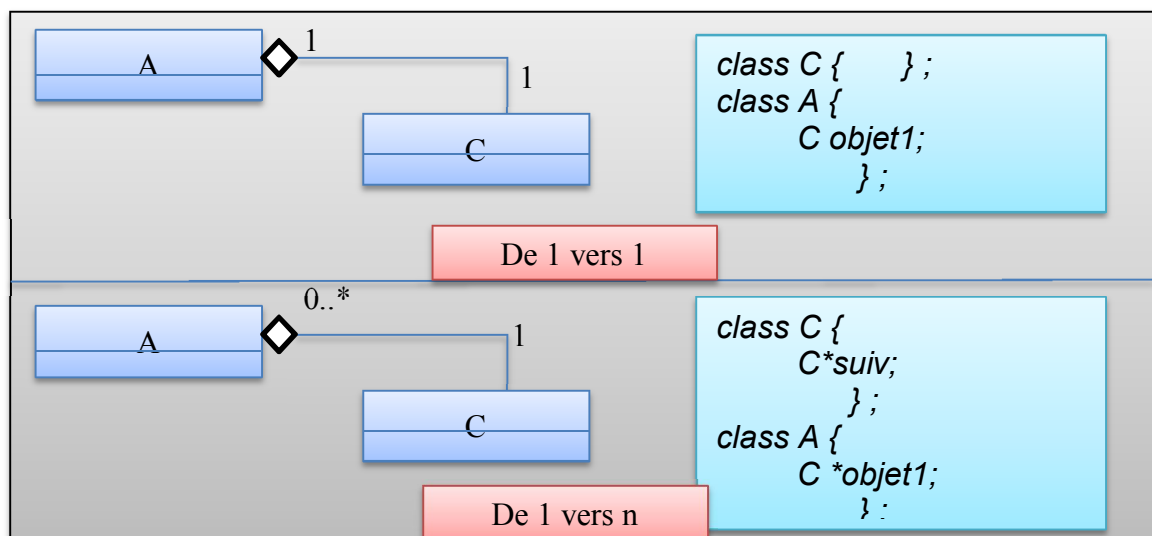


Figure 3.13: Implémentation d'agrégation en C++

2.7.1.4. Composition

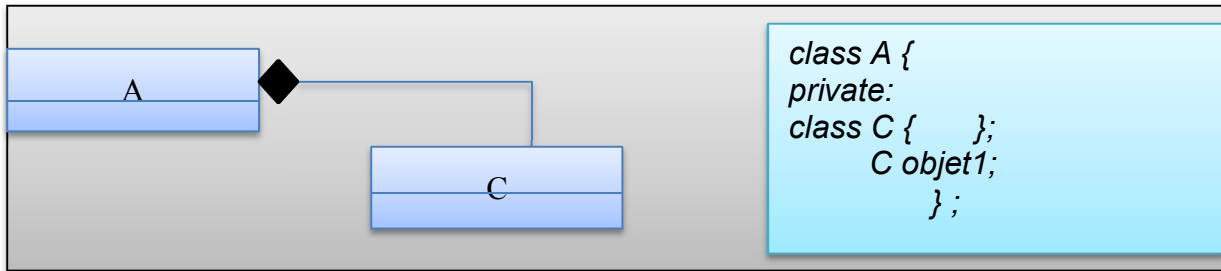


Figure 3.14: Implémentation de composition en C++

2.7.2. Diagramme de communication

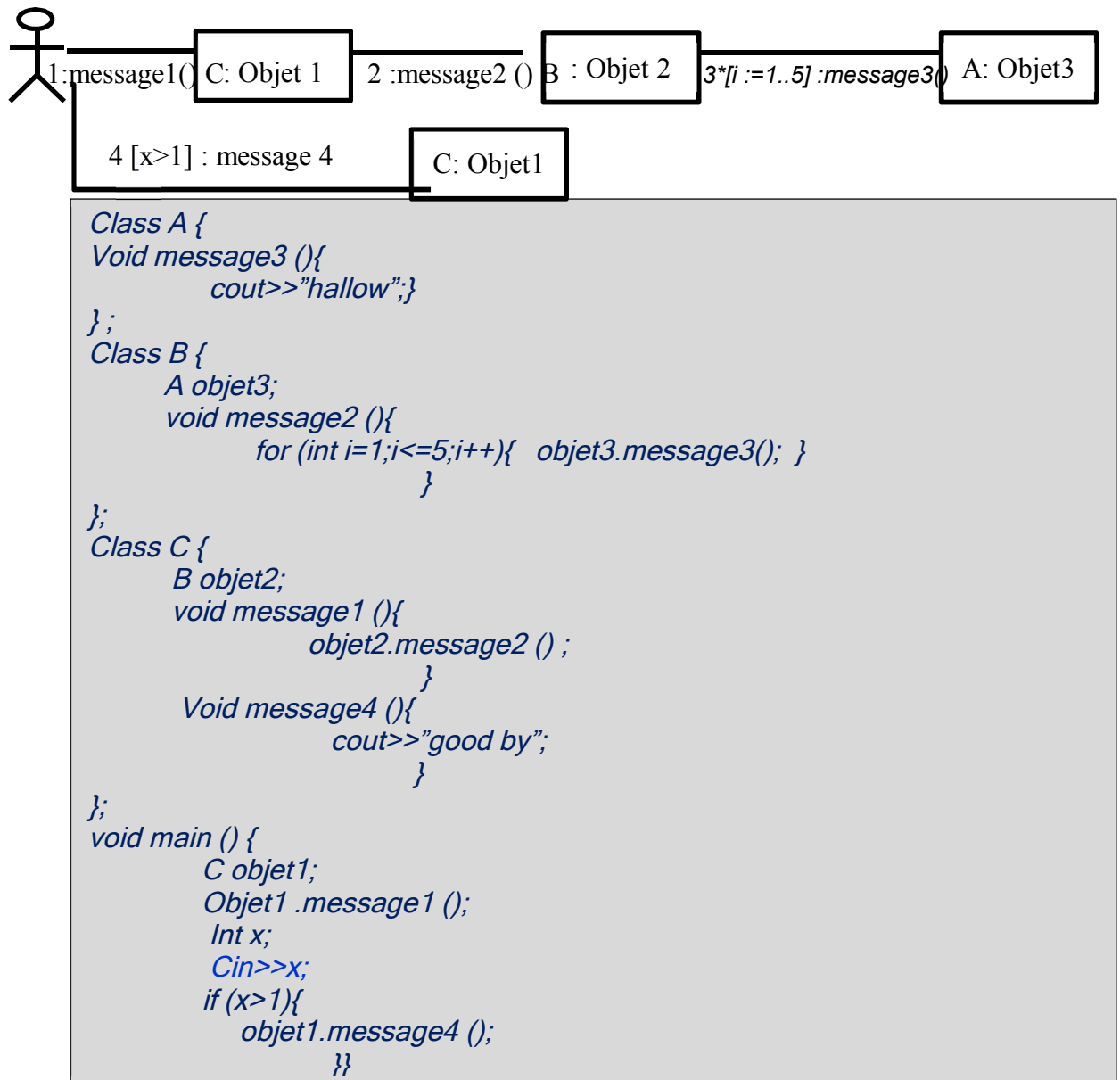


Figure 3.15: Implémentation d'un diagramme de communication en C++

3. Conclusion

L'objectif principal du présent chapitre a été l'introduction des principaux concepts liés à notre contribution, à savoir l'évaluation des apprenants dans les environnements d'apprentissage des T  L  TPs en utilisant les graphes et les appariements des graphes dans le calcul de similarit   et la comparaison entre le programme de l'apprenant et l'enseignant. Sachant que chaque programme doit   tre transform   en graphes UML. Donc la comparaison entre les deux programmes n  cessite un processus d'appariement entre les deux graphes repr  sentant ces deux programmes, pour comparer le programme de l'apprenant avec celui de l'enseignant.

Deux types d'appariement de graphes ont   t   introduits dans ce chapitre : (i) un appariement exact et (ii) un appariement approximatif. Le premier exige soit une   galit   entre les deux graphes compar  s, soit une inclusion totale du plus petit graphe dans le plus grand. L'appariement approximatif, quant    lui, offre une tol  rance par rapport au nombre d'erreurs de mise en correspondance entre les n  uds et/ou les ar  tes des deux graphes appari  s. Les deux approches prouvent ces   checs dans le cas o   le graphe est grand et elles ne peuvent pas   tre appliqu  es dans le cas de l'appariement multivoque.

Une autre mesure de similarit   propos  e par Pierre-Antoine Champin et Christine Solnon [sorlin ,2003] qui a r  solu le probl  me a   t     galement introduite dans ce chapitre. Cette mesure se base sur le calcul de meilleur appariement qui maximise la similarit   entre les deux graphes. Pour calculer le meilleur appariement, on a introduit trois algorithmes : glouton, recherche locale taboue et r  active et l'optimisation par colonie de fourmis. Notre approche d'  valuation des apprenants s'agit d'une comparaison entre les productions des apprenants et ceux de l'enseignant en se basant sur la similarit   propos  e par champin et Solnon et l'algorithme glouton. Comme domaine d'application, nous avons opt   pour la programmation orient  e objet comme qui fera l'objet du chapitre suivant.

CHAPITRE 4

La programmation orientée objet: un domaine d'application pour l'apprentissage des activités de T  L  TPs

1. Introduction

L'apprentissage de la programmation chez les   tudiants d  butants est toujours r  put  e difficile : jusqu'   80% des   tudiants dans un cours d'initiation    la programmation   chouent ou abandonnent [Guittet et al. 2005][Guibert, 2006].

Dans ce chapitre, nous   mettons les hypoth  ses : qu'une partie de la difficult   et de la frustration exp  riment  es par certains apprenants en phase d'initiation provient, d'un c  t  , des caract  ristiques intrins  ques de la discipline elle-m  me. Et d'un autre c  t   de l'usage des m  thodes classiques dans l'apprentissage de la programmation (des s  ances de TPs en pr  sentiels). Ce type d'apprentissage souffre de plusieurs inconv  nients comme il est montr   dans [Boussaha, 2009]. Dans les sections qui suivent, nous essayons de r  pondre    ces deux hypoth  ses en se fondant sur nos deux exp  rimentations r  alis  es avec l'utilisation de notre plateforme d'apprentissage des activit  s des T  L  TPs de la programmation orient  e objet. Avant de confirmer ces deux hypoth  ses, nous donnons une vue g  n  rale sur l'activit   de la programmation.

2. Epist  mologie de la programmation

2.1. Un peu d'histoire

L'histoire de la discipline de la programmation au cours des ann  es passe par des grandes dates marquantes, r  sum  es dans ces quelques lignes suivantes :

Chapitre 4 La programmation orientée objet : un domaine d'application pour l'apprentissage des activités de TéléTPs

Au début des années 50, les praticiens au domaine de l'informatique, proposaient des définitions claires de la programmation : Hartree, en 1950, dans [Hartree, 1950] définit la programmation comme suit « *le processus de préparation d'un calcul pour une machine se divise en deux parties, programmer et coder. La programmation est le processus d'élaboration de la séquence d'opérations requise pour réaliser le calcul* »; Wilkes dans [Wilkes, 1956] donnait une autre définition de la programmation comme suit : « *la séquence d'ordre est connue sous le nom de programme, et la machine l'exécute automatiquement sans intervention de l'utilisateur* ». Ces définitions sont apparues avec l'utilisation en parallèle des ordinateurs dans les calculs, cela entraînant, l'apparition des premiers langages de programmation, citons à titre d'exemples : (Fortran, mathematical FORMula TRANslating system- et LISP -LIST Processing language) [Blackwell, 2002] [Guibert, 2006].

Ensuite, à la fin de la décennie, les notions de « programme », et « de programmation » sont fusionnées dans une seule définition comme il est cité dans la définition de Wrubel [Wrubel, 1959]: « *cette séquence est appelée programme et son processus de préparation est appelé programmer* ».

Cependant, au cours des années, et surtout avec l'expansion de l'utilisation de l'ordinateur et de l'invasion de différents domaines, toute une nouvelle génération de langages est apparue, tels Pascal (1970), C, Prolog et SmallTalk (1972), ou Basic (1975), dans les années 70. La généralisation de la programmation objet grâce aux performances des micro-ordinateurs, Java, Perl, Python s'ajoutent aux langages micros, au cours des années 90.

Ces dernières années sont caractérisées par l'introduction d'un nouveau terme : c'est la programmation sans langage ou bien comme il est appelé « *end user* ». Ce sont des outils de programmation destinés aux non-informaticiens comme le **HTML** pour la conception d'un site Internet, LaTeX pour la rédaction des documents scientifiques. Enfin, l'émergence d'un nouveau champ de recherche en psychologie, la « psychologie de la programmation », est introduite et devient une nécessité [Hague, 2001][Guibert, 2006].

Avec l'augmentation de la taille des applications informatiques. Le nombre de programmeurs associés à un projet doit être augmenté, en mettant en exergue une dimension sociale dans l'activité du programmeur. La « psychologie de la programmation »

s'intéresse aux processus cognitifs mis en œuvre par les programmeurs, et à la dimension sociale de leur travail [Blackwell, 2002].

2.2. Définition de la programmation

Le didacticien Charles Duchâteau dans [Duchâteau a,1992], a défini l'activité du programmeur comme suit « *faire faire une tâche par un exécutant - l'ordinateur -* ». Pour cela, il va concevoir *l'algorithme*, qui pilotera la réalisation ultérieure de la tâche. La figure 4.1 illustre comment se déroule la tâche de la programmation.

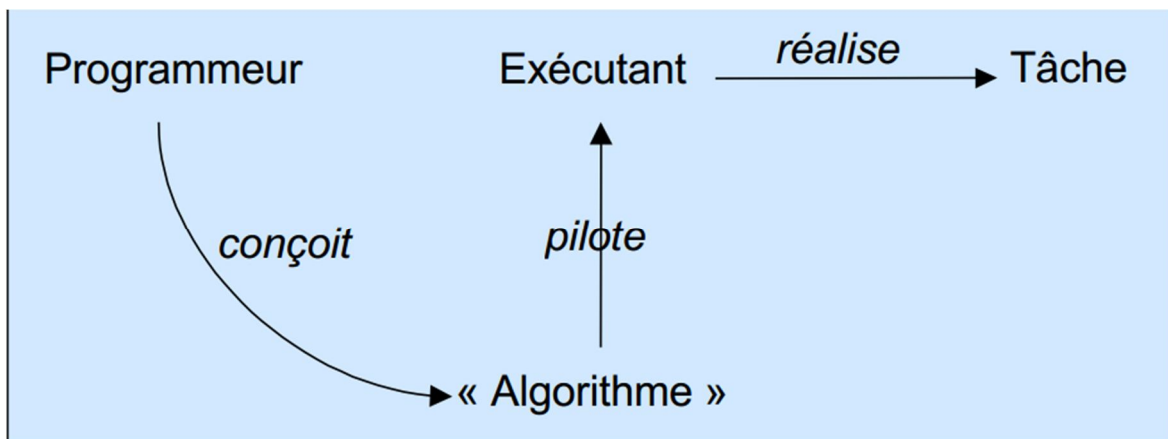


Figure 4.1: La tâche de la programmation [Duchâteau a,1992]

2.3. Décomposition de l'activité de la programmation

La figure 4.2 représente la décomposition de l'activité de la programmation

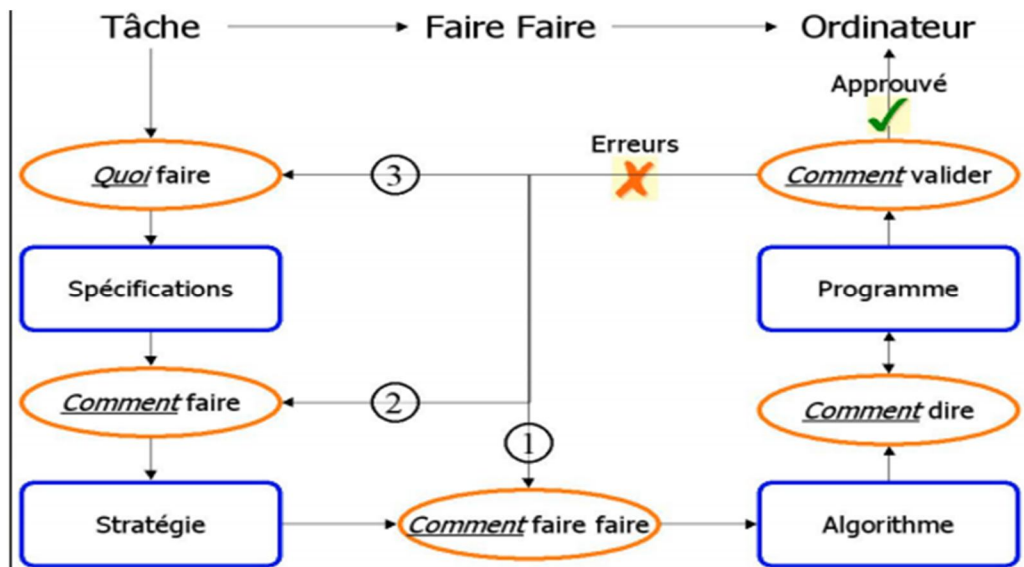


Figure 4.2: La décomposition de l'activité de la programmation [Guittet et al., 2005]

Afin *de faire faire* une tâche par l'ordinateur, le programmeur passera donc par les étapes suivantes :

- ***Quoi faire***

La première étape de l'activité du programmeur consistera le plus souvent en l'analyse de la tâche que la machine aura à traiter, où les informaticiens doivent poser les questions pertinentes pour l'automatisation de cette tâche devant un client qui dispose du *savoir-faire* (généralement en partie inconscient) sur la tâche considérée [Pair, 1988] [Guittet et al., 2005].

- ***Comment faire***

Cette deuxième étape permettant *d'abstraire* tous les comportements de la machine lors de l'exécution. Cette phase doit se faire, à la fois, d'une manière :

- exhaustif : il s'agit de couvrir tous les comportements que l'on attend du programme
- explicite : consiste à passer du domaine de l'inconscient à celui du conscient prenant un exemple d'enseignant qui demande à ces apprenants de trouver le plus petit élément dans une liste d'entiers. Le problème qui se pose ici, ce n'est le savoir-faire, mais du savoir, « comment faire » [Pair, 1989].

- ***Comment faire faire***

A ce stade, le programmeur doit décrire les objets d'intérêt de la tâche sous une forme compréhensible par l'ordinateur. Autrement dit il faut franchir la barrière cognitive entre sa représentation de l'univers de la tâche, et celle, complètement formelle et le plus souvent numérique, employée par l'ordinateur [Duchâteau b, 1992][HOC, 1982].

- ***Comment dire***

Pour être compréhensible par l'ordinateur, la description des objets d'intérêt de la tâche doit ensuite être traduite dans un langage de programmation. Cette étape se caractérise par la prééminence de la syntaxe sur la sémantique, et il convient de s'y exprimer de façon rigoureuse, tant grammaticalement qu'orthographiquement [Guibert, 2006].

- ***Comment valider***

Cette étape est difficile pour le programmeur professionnel et elle l'est encore plus pour le débutant. Après avoir effectué de (nombreuses) modifications au programme pour

le rendre syntaxiquement valide, le programmeur doit alors passer dans une étape d'exécutions successives pour vérifier la sémantique. Il y a presque toujours des erreurs à la fin de cette séquence. Ce sont les erreurs algorithmiques qui sont généralement détectées en premier, après quoi l'on s'aperçoit d'erreurs de stratégie (modélisation de la tâche non exhaustive, ou stratégie incorrecte) et finalement, on peut avoir la mauvaise surprise de ne pas avoir réalisé l'outil attendu par le client [HOC, 1982].

3. Principales difficultés dans l'apprentissage des TéléTPs de la programmation, d'où viennent-elles?

Comme il a été supposé précédemment, ces difficultés proviennent de deux causes :

- les caractéristiques intrinsèques de la discipline elle-même;
- l'utilisation des méthodes classiques dans l'apprentissage des TéléTPs de la programmation (TPs en présentiel).

3.1. Première hypothèse : difficultés à cause des caractéristiques intrinsèques de la discipline de la programmation

Blackwell dans [Blackwell, 2002] déterminait des caractéristiques essentielles à la programmation :

- ***La perte de la manipulation directe***

Dans un système de manipulation directe, l'état actuel du système doit être représenté de façon continue à l'utilisateur, une action unique devrait avoir un effet visible unique dans la représentation, et le rétablissement de l'état de la représentation à celui avant l'action doit rétablir la situation [Blackwell, 2002].

Dans les systèmes de programmation, aucune de ces choses n'est nécessairement vraie. La situation dans laquelle le programme doit être exécuté peut ne pas être disponible pour inspection, car le programme peut être appliqué à un plus grand éventail de situations de données que celles actuellement visibles chez le programmeur ou parce qu'il peut être utilisé ultérieurement.

Toutes les tâches de programmation ont en commun de spécifier un comportement futur de la machine. Cette absence d'écho immédiat se révèle un élément-clé et intrinsèque dans la difficulté à programmer [Blackwell, 2002].

- ***L'usage de notations pour supporter l'abstraction***

La deuxième caractéristique universelle de la programmation est que le programme est représenté dans un ensemble de notations, à travers lesquelles le programmeur exprime le comportement futur de la machine. Cet usage de notations se justifie par la possibilité de décrire de façon abstraite une multitude d'exécutions différentes.

D'autre part, les « programmes » sont composés « d'instruction ». C'est-à-dire que l'écriture de programmes requiert non pas un « savoir-faire » mais un « savoir-faire faire » qui nécessite de la part de l'étudiant une prise de recul par rapport à ses connaissances sur la tâche [Liskov, 1974].

- ***La programmation et la distance entre le model mental du programmeur***

Les différences entre la représentation du langage et le modèle qu'en fait l'utilisateur sont exprimées dans [Norman, 1990] par des distances. La distance sémantique figure 4.3 attrait à la connaissance des objets manipulés et à la signification des commandes et des retours d'informations (franchies dans les phases d'intention et d'évaluation). Comme son nom l'indique, la distance articulatoire mesure la difficulté à faire le lien entre les grandeurs et observables du système et le modèle mental de l'utilisateur (spécification et interprétation) [Guibert, 2006].

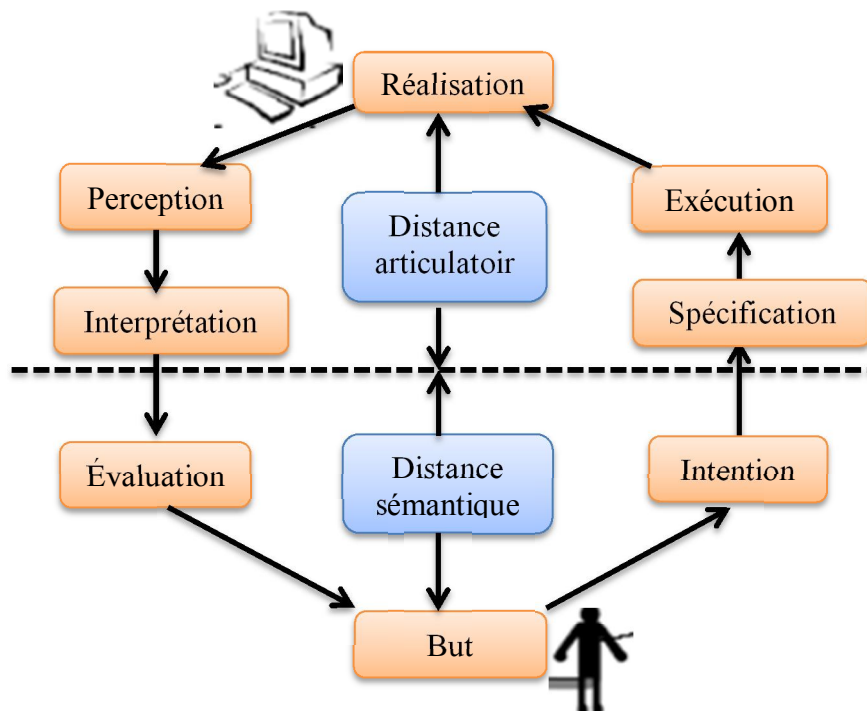


Figure 4.3: Distances séparant l'ordinateur et la tâche qu'il souhaite accomplir, selon [Norman, 1990].

Chapitre 4 La programmation orientée objet : un domaine d'application pour l'apprentissage des activités de TéléTPs

✓ **Expérimentation 1 : confirmation de la première hypothèse**

Un suivi des résultats des étudiants de deuxième année informatique de notre université est réalisé après la fin de troisième semestre pendant 02 années successives (2014-2015 ; 2015-2016). Ce suivi consiste à comparer le taux de réussites des étudiants en programmation orientée objet avec d'autres modules enseignés pendant les mêmes années. Le résultat obtenu montre un taux de réussites de 09.16 % sur les deux ans consécutifs concernés par le suivi.

Dans cette étude, nous avons suivi l'évolution du niveau des étudiants via leurs résultats en programmation orientée objet. Ensuite, nous avons validé les notes obtenues en les comparant avec celles obtenues dans les autres unités de valeur du même semestre. Les tables 4.1 et 4.2 suivantes montrent les résultats obtenus.

NOTE	2014-2015	2015-2016
	Janvier	Janvier
<10	91,21%	90,48%
>= 10	08,79 %	09.52%

Table 4.1 : Le résultat de module programmation orientée objet (POO) et le taux de réussite

Année	note	THL	ASD	MN	SI	Architec	LM	POO
2014-	<10	67 ,03%	42,86%	78,03%	61,54%	71,43%	91 ,21%	91,21%
2015	>=10	32,97%	57,14%	21,97%	38 ,46%	28,57%	08,79%	08,79%
2015-	<10	87,48%	45,24%	90 ,48%	30,95%	50%	76,19%	90,48%
2016	>=10	12,52%	54,76%	09,52%	69,05%	50%	23,81%	09,52%

Table 4.2 : Comparaisons de taux de réussite de la programmation orientée objet par rapport à d'autres modules

Ce problème d'échecs en programmation ne concerne pas seulement notre institution. Plusieurs études sur l'apprentissage de l'algorithmique/programmation menées par différentes institutions dans d'autres pays ont convergé vers la même conclusion : l'apprentissage de l'algorithmique/programmation était toujours une source de difficultés non seulement pour les étudiants, mais pour les enseignants aussi. Cela est dû à des caractéristiques intrinsèques de la discipline.

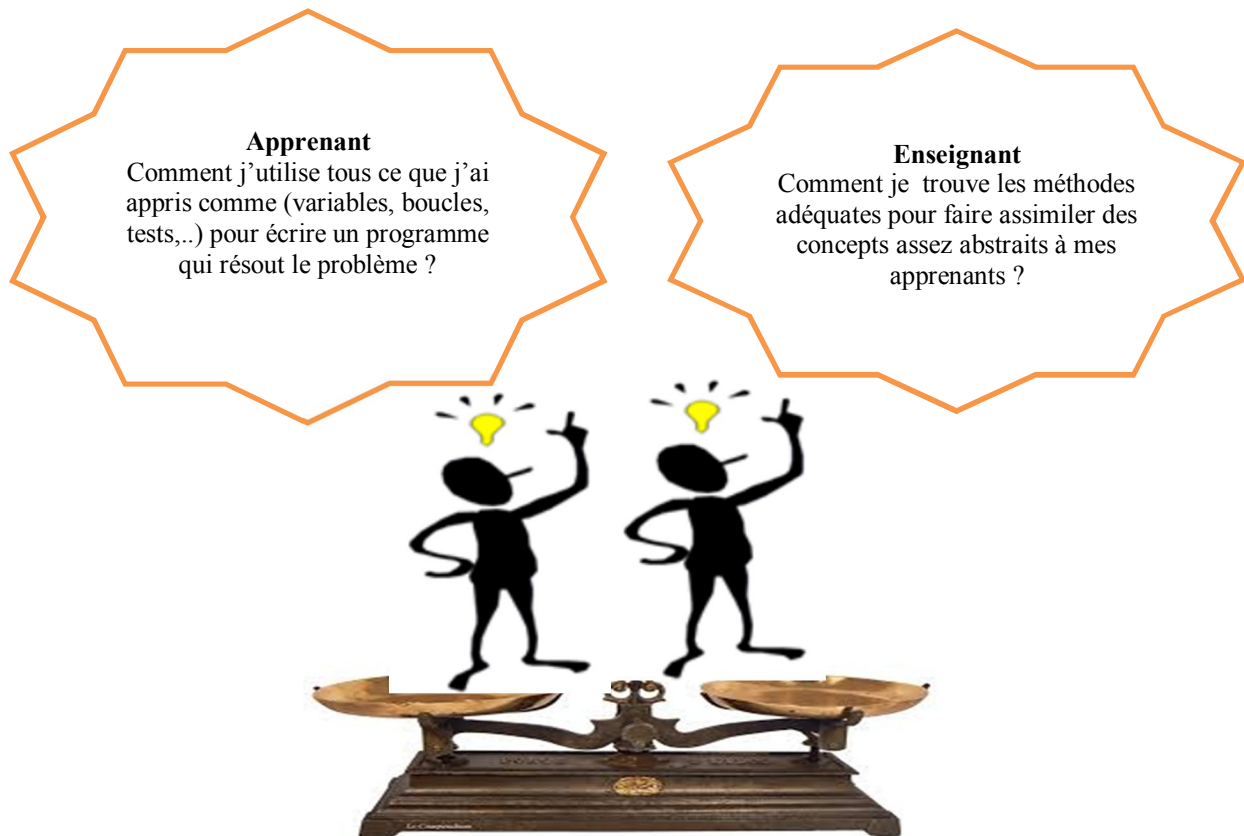


Figure 4.4: L'activité de la programmation : une source de difficulté.

3.2. Deuxième hypothèse : difficultés d'utilisation des méthodes classiques (en présentiel) dans l'apprentissage

Comme nous avons constaté précédemment, les difficultés de la discipline de la programmation proviennent de deux causes : l'utilisation des méthodes classiques dans l'apprentissage des TéléTPs de la programmation et les caractéristiques intrinsèques de la discipline elle-même. Dans la sous-section suivante, nous essayons de démontrer les apports et les avantages de l'utilisation des plateformes en ligne pour l'apprentissage des TéléTPs dans la programmation par rapport aux méthodes classiques utilisées dans ce type d'activités (TPs en présentiel). Mais avant d'aborder les avantages de l'apprentissage à distance de ce type d'activités, nous identifions les inconvénients des méthodes en présentiels [Boussaha, 2009].

3.2.1 Difficultés de l'apprentissage présentiels des TPs en programmation

Plusieurs problèmes sont cités dans [Boussaha, 2009], dont nous citons :

- a- problèmes techniques :

Chapitre 4 La programmation orientée objet : un domaine d'application pour l'apprentissage des activités de TéléTPs

- ✓ désinstallation des outils de programmation par le groupe précédent ;
- ✓ les problèmes de fonctionnement des PCs ;
- ✓ blocage du groupe ;
- b- des problèmes opérationnels :
 - ✓ mauvaise compréhension des consignes de travail.
 - ✓ perte du programme réalisé (le cas des étudiants de la première année qui ne savent pas sauvegarder leurs programmes).
 - ✓ apparition d'erreurs pendant la tâche de correction des erreurs.
 - ✓ la longue durée d'attente de sollicitation des assistants de TP

En effet, le « problème » posé dans les TP d'initiation à la programmation n'est pas tant lié aux tâches à accomplir (calculer la moyenne d'une liste de nombres, trouver le plus petit (ou grand) élément, trier par ordre croissant (décroissant)... etc.) mais aussi relative à la disponibilité du matériel. En effet, la difficulté « instrumentale » n'y apparaît pas clairement, alors qu'elle est très importante dans l'apprentissage de la programmation.

Donc, face à ces problèmes, l'enseignement à distance de ce type d'activités devient indispensable pour remédier à ces obstacles.

Notre petite expérience avec notre plateforme conçue et réalisée pour ces objectifs justifie l'efficacité d'utilisation de ce type de plateforme dans l'apprentissage de ce type d'activités.

3.2.2. L'efficacité d'utilisation de l'apprentissage à distance dans les activités de la programmation

✓ Expérimentation 2 : confirmation de la deuxième hypothèse

Nous avons demandé aux apprenants qui ont l'expérience d'utiliser la plateforme E-TéléTPC@AALP de signaler les différents apports des TéléTPs. La majorité des étudiants jugent que les TéléTPs offrent plusieurs avantages par rapport aux TP en présentiel sur les langages de programmation [Boussaha, 2009].

Ils permettent de :

- ✓ offrir des TP à jour à tout moment.
- ✓ offrir davantage la possibilité de collaborer avec d'autres personnes ayant différentes idées que celles des groupes de la salle.

Chapitre 4 La programmation orientée objet : un domaine d'application pour l'apprentissage des activités de TéléTPs

- ✓ diminuer le problème d'encombrement dans les groupes et continuer les TP programmés pour la formation des étudiants.
- ✓ éviter le problème d'attente de sollicitation de l'assistant.

L'utilisation continue de cette plateforme depuis 2009 (l'année de réalisation de cette plateforme) dans l'apprentissage à distance des TP en programmation montre l'efficacité de ce type d'environnements par rapport aux méthodes en présentiels. Il est intéressant de signaler qu'environ 61,54% des étudiants enquêtés jugent que les travaux pratiques à distance dans la programmation orientée objet sont intéressants. Et dans le cadre d'un travail de recherche lié à la conception et l'extension de services de notre plateforme de travaux pratiques à distance (E-TéléTPC@AALP) sur les langages de programmation et spécifiquement l'intégration de module d'évaluation dans cette plateforme, nous avons besoin de recueillir certaines informations. Nous avons distribué un questionnaire sur les deux groupes participants à l'expérimentation, ce questionnaire contient un ensemble de questions réparties en deux catégories, le premier ensemble concerne l'utilisation de la plateforme E-TéléTPC@AALP et le deuxième ensemble concerne l'évaluation des apprenants. Dans ce qui suit, nous exprimons quelques réponses des apprenants au questionnaire sous forme d'histogrammes. Les autres réponses sont exprimées dans l'annexe A.

Q I.1 : Que pensez-vous de faire un TP à distance ?

- Intéressant peu intéressant pas intéressant

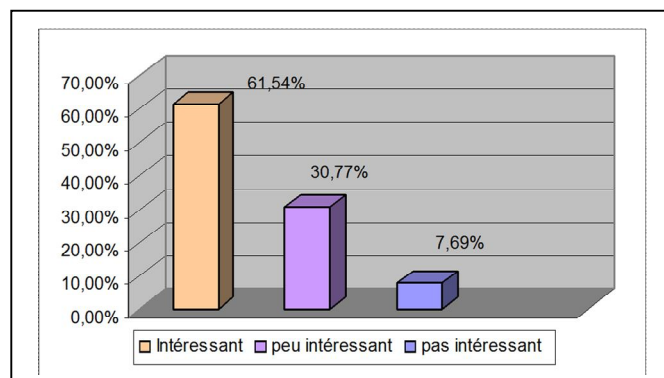


Figure 4.5 : L'intérêt exprimé de faire un TP à distance

Q I.4 : Pour réaliser un TéléTPs avec la plateforme E-TéléTPC@AALP il y a 4 tâches essentielles. Pour chaque tâche, déterminez le degré de difficulté ?

- a. Edition de code

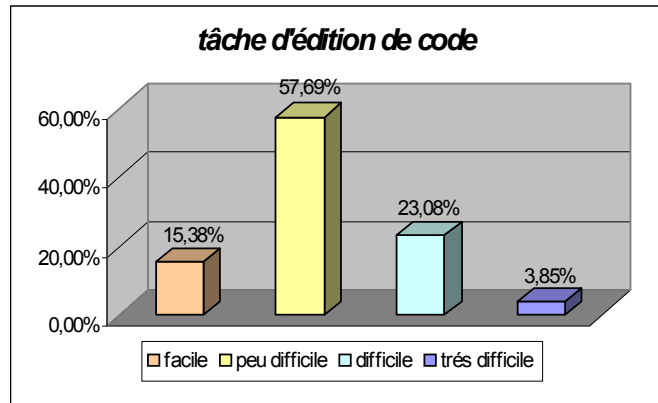


Figure 4.6: La tâche d'édition de code

b. Compilation

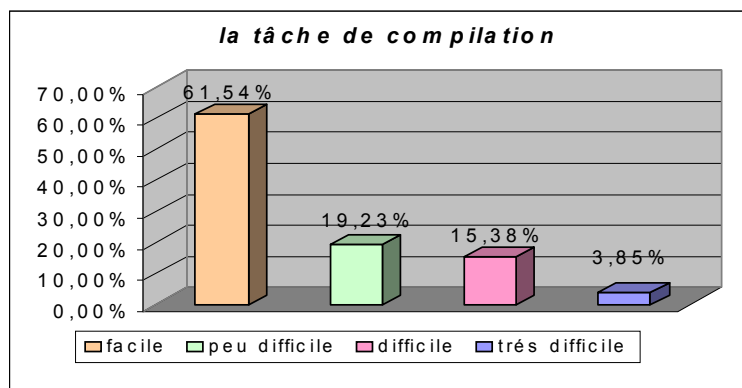


Figure 4.7 : La tâche de compilation

c. Correction des erreurs

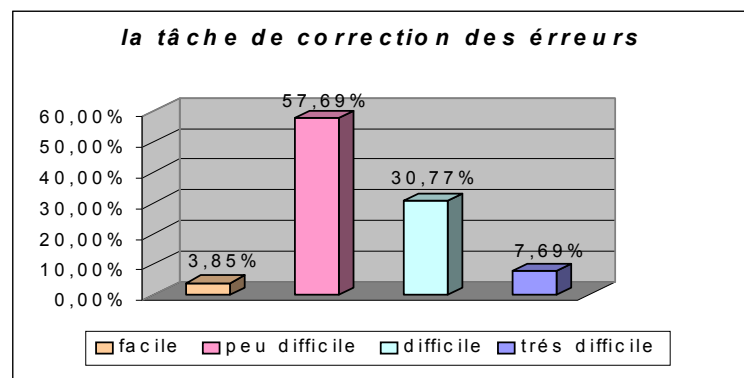


Figure 4.8: La tâche de correction des erreurs

d. Exécution

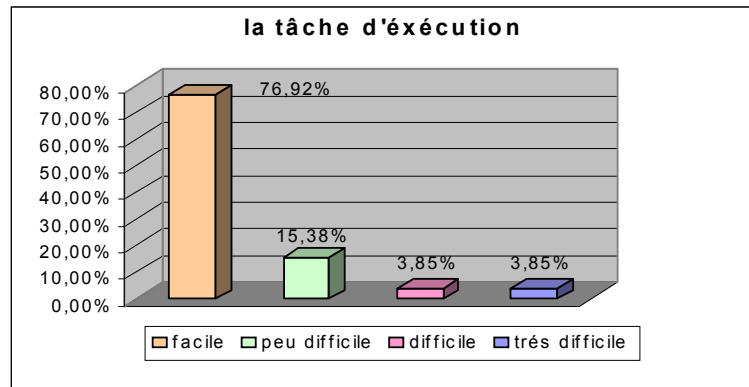


Figure 4.9 : La tâche d'exécution

Q I.5 : Dans les tâches difficiles comment surmontez-vous les difficultés ?

- Demander l'aide collaborer autres

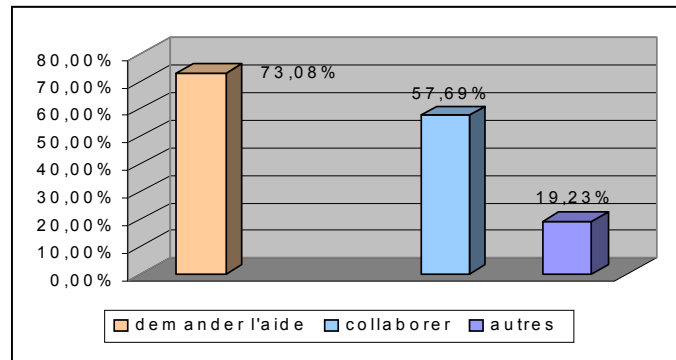


Figure 4.10: Les réactions des apprenants devant les tâches difficiles

Q II.1 : Dans un environnement de travaux pratiques à distance quel mode d'évaluation préférez-vous ?

- Manuel (compte rendu)
- QCM Automatique
- Outil d'auto évaluation
- Autre, préciser
-
-

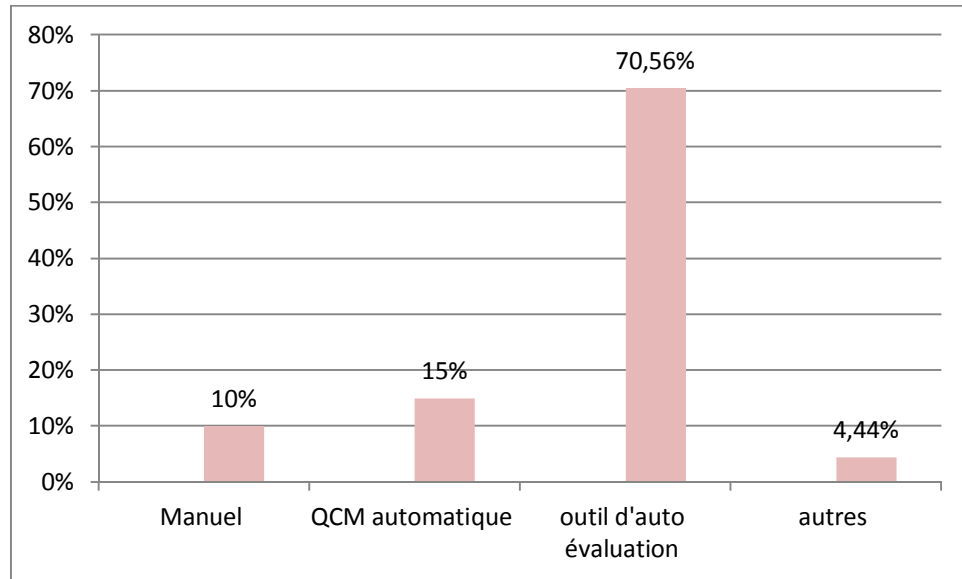


Figure 4.11: L'évaluation préférée chez les apprenants

Q II.2 : Pour les différentes tâches à effectuer au cours du votre TêLêTP déterminez si la tâche doit se faire individuellement ou à l'aide d'un outil assistant ?

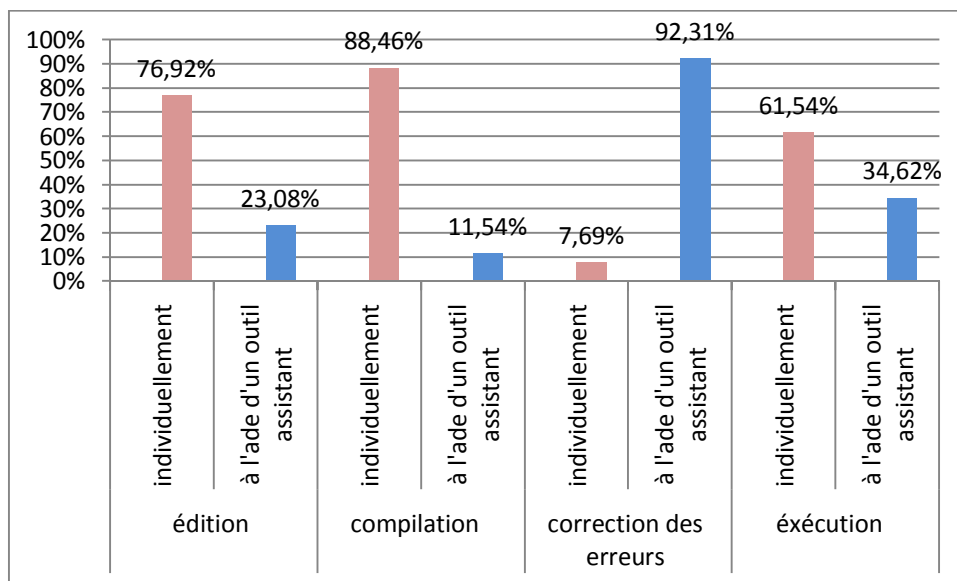


Figure 4.12 : Les préférences des apprenants pour la réalisation des quatre tâches de la programmation

Q.II.4.commet trouvez-vous l'intérêt d'ajout d'un module d'évaluation dans la plateforme

- catalyseur pour l'apprentissage
- frustrant pour la l'apprentissage
- autres

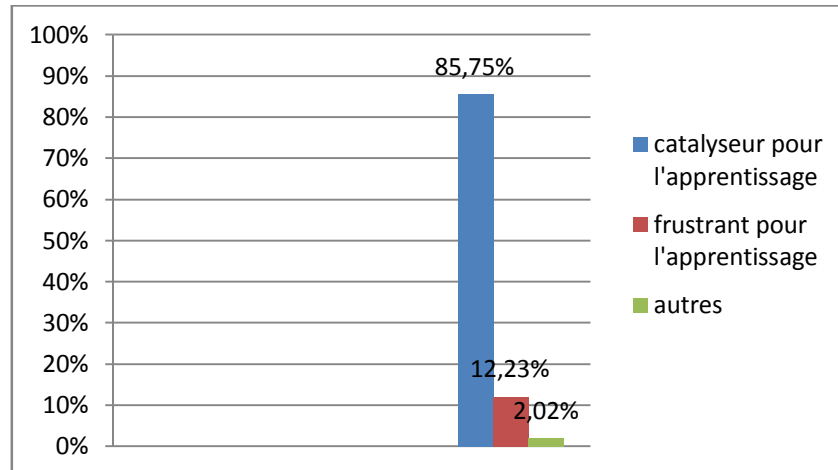


Figure 4.13: L'utilité d'utilisation d'un module d'évaluation dans la plateforme

Le taux d'échecs élevé dans l'apprentissage de la programmation, nous amène à poser les questions suivantes :

- *quelles sont les sources de difficultés des apprenants dans leurs apprentissages en programmation et quelles sont les erreurs communes chez les apprenants débutants en programmation ?*
- *en quoi les environnements d'apprentissage en ligne peuvent-ils remédier à ces difficultés ?*
- *avec quelles méthodes pédagogiques peut-on améliorer l'apprentissage de la programmation?*

4. L'activité de la programmation et les méthodes d'évaluation automatiques

Pour évaluer un apprenant à travers son programme, deux approches d'évaluation automatiques sont utilisées dans la littérature :

4.1. L'approche statique

Cette approche est fondée sur la collecte des informations de code de programme sans lancer son exécution. Elle se base sur la vérification visuelle de code en prenant en compte plusieurs aspects pour l'évaluation tels que : style de codes, erreurs de programmation, métriques logicielles, conceptions et d'autres aspects sont cités en détail dans [Bouacha, 2015].

4.2. L'approche dynamique

Cette approche se base sur l'exécution de programme pour l'évaluer, elle est fondée sur le comportement dynamique de programme en prenant en compte plusieurs aspects tels que : la fonctionnalité, l'efficacité, jeux de tests, et d'autres aspects sont expliqués en détail dans [Bouacha, 2015].

Malheureusement, nous ne pouvons pas dire que la première approche est efficace par rapport à la deuxième, et vice versa, parce que chaque méthode a ses avantages et ses inconvénients comparés à la seconde méthode, mais nous pouvons confirmer que les deux ne fournissent pas une évaluation totalement automatique. Les deux approches ont besoin d'intervention de l'être humain dans certains cas, où la situation de prendre la décision est vague et ambiguë.

Dans les sections précédentes, nous avons essayé de répondre aux questions soulevées liées à l'apprentissage de l'activité de la programmation et pourquoi cette activité est toujours une source de difficultés et ce par la présentation et la confirmation de deux hypothèses avec deux expérimentations réalisées.

D'après notre enquête auprès des étudiants et leur réponse à la question suivante : *comment trouvez-vous l'intérêt d'utilisation d'un module d'auto-évaluation dans la plateforme ?* Nous avons trouvé que 85.75% parmi les étudiants enquêtés jugent que l'ajout d'un module d'auto-évaluation est un catalyseur pour eux pour progresser dans leur l'apprentissage. Ce résultat de l'enquête et l'échec des deux approches automatiques d'évaluation dans certains cas, nous amènent à poser la question suivante :

- *avec quels outils d'évaluation peut-on progresser l'apprentissage de la programmation chez les apprenants ?*

Avant de répondre à cette question en présentant notre contribution dans le chapitre suivant, nous citons, ci-après, les avantages de la programmation orientée objet, notre champ d'application.

5. Avantages de la programmation orientée objet

- L'intérêt principal de l'OO réside dans le fait que l'on ne décrit plus par le code des actions à réaliser de façon linéaire mais, par des ensembles cohérents appelés objets. Cette description présente d'énormes avantages: facilité d'organisation,

Chapitre 4 La programmation orientée objet : un domaine d'application pour l'apprentissage des activités de TéléTPs

réutilisation, méthode plus intuitive, possibilité d'héritage, facilité de correction, projets plus faciles à gérer [Curé, 2005].

- Dans un modèle à objets, toute entité du monde réel est un objet, et réciproquement tout objet représente une entité du monde réel donc l'OO est facilement concevable, car il décrit des entités comme il en existe dans le monde réel. Ainsi, les modèles à objets ont été créés pour modéliser le monde réel. [Duris, 2003] [Paetau, 2005].
- L'OO permet d'écrire des programmes facilement lisibles avec un minimum d'expérience, de taille minimale et à la correction aisée. Ces programmes sont, de plus, souvent très stables [Paetau, 2005].
- Les informations concernant un domaine étant centralisées en objets, il est facile de sécuriser le programme en interdisant ou autorisant l'accès à ces objets aux autres parties du programme [Curé, 2005].
- La programmation objet permet d'obtenir des projets modulaires. En général, en OO, cohabitent deux types de développeurs: ceux qui conçoivent les objets et ceux qui utilisent ces objets dans leurs programmes. De cette façon, la programmation objet permet de *diviser la complexité*. Chaque programmeur ne travaille que sur des implémentations le concernant [Paetau, 2005][Duris,2003].

6. Conclusion

Dans ce chapitre, nous avons étudié les caractéristiques du domaine de la programmation, en général, et plus spécifiquement la programmation orientée objet qui est notre champ d'application. Nous avons également présenté deux études pratiques dans ce domaine, la première montre les échecs de l'apprentissage de la programmation avec des méthodes classiques (des séances de TP en présentiel). Quant à la deuxième, elle concentre l'utilisation des plateformes d'apprentissage en ligne dans ce type d'activités et l'utilité d'évaluation automatique dans ces plateformes. L'analyse du questionnaire distribué aux apprenants nous donne les justificatifs suivants :

Il faut d'abord remarquer que la majorité des apprenants concernés par l'enquête n'ont jamais entendu parler d'un TP à distance. Les autres avaient des avis partagés : certains étudiants ont estimé que la distance pourrait être gênante; d'autres ont montré de l'intérêt pour l'aspect collaboratif; enfin certains d'autres ont accepté l'idée, mais posent le problème de l'assistance, il ne se voit pas faire un TP sans assistant à proximité. Mais d'une manière générale, l'ensemble des apprenants interrogés acceptent l'idée du

Chapitre 4 La programmation orientée objet : un domaine d'application pour l'apprentissage des activités de TéléTPs

TéléTP si l'environnement proposé assurait surtout l'assistance. La majorité des étudiants qui ont eu l'occasion d'utiliser la plateforme E-TéléTPC@AALP pour la réalisation de leurs TP jugent que les TéléTPs offrent plusieurs apports par rapport aux TP en présentiel sur les langages de programmation.

Cependant, la majorité des apprenants estiment que la tâche la plus difficile dans la programmation est la correction des erreurs et le blocage devant une erreur est un facteur frustrant pour les apprenants de progresser leurs apprentissages. Ce problème est le principal catalyseur pour nous de réfléchir à améliorer les services de notre plateforme par l'ajout d'un mécanisme d'évaluation qui offre un feedback pour les apprenants pour corriger leurs erreurs. Dans le chapitre suivant, nous exprimons, notre contribution concernant la mise en œuvre d'un mécanisme d'évaluation des apprenants dans les activités des TéléTPs en utilisant la plateforme E-TéléTPC@AALP consacrée à l'apprentissage des TéléTPs en programmation orientée objet.

Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

1. Introduction

Dans les environnements informatiques pour l'apprentissage humain (EIAH), l'enseignant n'est pas face à face avec ses apprenants pour déterminer, à partir de leurs expressions faciales et leurs questions et leurs interactions, s'ils ont acquis les connaissances présentées [Boussaha, 2011] [Durand, 2006].

La majorité des plateformes e-learning fournissent des indicateurs typiques de tutorat et de l'évaluation des apprenants, tels que le nombre de connexions de l'apprenant, la date limite de la session, la dernière visite de départ du cours, etc. Pour évaluer l'apprenant, les plateformes utilisent souvent des outils classiques intéressants comme les questions à choix multiples (QCM) [Tchounikine, 2009]. Cependant, ils ne permettent pas à l'enseignant d'avoir une perception (collection) de l'activité réelle de l'apprenant [Tchounikine, 2009]. Bien qu'elle représente une activité importante pendant la formation de l'apprenant, l'évaluation de l'apprenant dans les travaux pratiques avec les EIAH n'a pas atteint un niveau de maturité élevé.

Dans ce contexte, très peu de travaux [Bennouna et al, 2009] ont été émergés dans la littérature. Autres travaux qui traitent l'évaluation de l'apprenant dans le processus de modélisation [Auxepaules, 2009] [Mitrovic & Suraweera, 2004] ont également été proposés.

Bien sûr, ces travaux ont considérablement amélioré le domaine en proposant de nouvelles stratégies pour l'évaluation de l'apprenant. Cependant, ils ne traitent pas de l'apprentissage des langages de programmation.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

Dans ce chapitre, nous nous intéressons à l'évaluation des apprenants dans les travaux pratiques. L'évaluation permet à l'apprenant d'évaluer et de développer ses compétences (l'analyse, compréhension, application et de synthèse) dans un domaine donné. C'est une opération qui s'avère compliquée dans les travaux pratiques qui représente une partie importante du e-learning.

Bien que ce type d'évaluation soit à la fois important et approprié pour les apprenants dans les disciplines scientifiques et techniques (c'est-à-dire qu'il contribue à l'avancement des compétences de l'apprenant, d'encourager les activités de formation et aider les apprenants à se préparer pour les tests), il est encore un sujet qui n'a pas été suffisamment exploré.

En outre, les activités d'évaluation des travaux pratiques sont parmi les plus délicats dans un EIAH, spécifiquement, parce que les travaux pratiques en informatique sont caractérisés par la multitude des solutions pour un problème donné. Cette caractéristique augmente la difficulté de l'évaluation dans les systèmes d'apprentissage: l'expert dans le domaine de l'expérience a la difficulté de trouver toutes les solutions possibles à un problème pour les intégrer dans la base de solutions.

Nous présentons, dans ce chapitre, une nouvelle approche d'évaluation de l'apprenant qui étend notre plateforme E-TéléTPC@AALP [Boussaha, 2009]. Notre approche permet d'évaluer les travaux pratiques des apprenants sur le développement de programmes C++ orientés objets. L'idée est de procéder à une comparaison indirecte de deux programmes orientés objets (apprenant, enseignant) à travers leurs représentations graphiques décrites par deux diagrammes UML (diagramme de classes et diagramme de communication).

Notre approche a pour objet de trouver les similarités et les différences entre la production de l'apprenant et celle de l'enseignant. Pour répondre à ces objectifs, notre travail passe par deux étapes essentielles :

Dans une première étape, nous avons concentré sur la conception d'un générateur automatique des diagrammes UML, le rôle de cet outil est de fournir pour chaque programme solution une seule représentation graphique afin de la comparer avec celle de l'enseignant.

Par la suite, dans la deuxième étape, nous avons proposé une technique d'appariement afin de pouvoir comparer les différents diagrammes engendrés par le générateur proposé (les diagrammes engendrés considérés comme des graphes).

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

Avant de détailler notre approche d'évaluation de l'apprenant, nous donnons une vue générale sur notre plateforme E-TéléTPC@AALP (environnement de travaux pratiques collaboratifs à distance: application à l'apprentissage des langages de programmation) [Boussaha, 2009].

2. Vue générale sur la plateforme E-TéléTPC@AALP

Réalisé en 2009, E-TéléTPC@AALP [Boussaha, 2009], la figure 5.1 est une plateforme pour l'enseignement et la formation à distance des travaux pratiques dans les langages de programmation en informatique. Son architecture est organisée en trois niveaux (téléformation, téléprogrammation et interfaces). La plateforme est développée en ASP / MYSQL et elle fonctionne sur les deux systèmes d'exploitation: Windows et Unix. Elle offre les services suivants:

- **Pour l'apprenant**

Elle permet à l'apprenant d'effectuer des travaux pratiques à distance, pour s'adapter à la plateforme, de rechercher des collaborateurs, de choisir un langage de programmation informatique, de télécharger un langage de programmation et de consulter les états de travail pratique à distance.

- **Pour l'enseignant concepteur de TP**

Elle permet à l'enseignant d'enregistrer, de consulter le travail pratique à distance envisagé dans la formation, pour créer un nouveau travail pratique à distance et de contacter d'autres enseignants.

- **Pour le tuteur**

Elle permet au tuteur de surveiller les apprenants et leur offre des questionnaires à choix multiples afin de les aider à se rappeler les principaux concepts des langages de programmation.

2.1. Architecture de la plateforme E-TéléTPC@AALP

Comme il est cité ci-dessus, la plateforme proposée présentée par la figure 5.1 est organisée en trois niveaux :

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

- niveau téléformation : ce niveau décrit les interactions entre les acteurs humains (apprenant, tuteur, enseignant), les modèles d'interaction dans les travaux pratiques à distance, toutes les configurations possibles (différentes situations de formation entre les acteurs) dans les travaux pratiques à distance sur les langages de programmation, les situations d'enseignement dans les travaux pratiques à distance, la collaboration entre les acteurs, etc.
- niveau interface : il joue le rôle d'intermédiaire entre le niveau de téléformation et le niveau téléprogrammation.
- niveau de téléprogrammation: Il présente les activités liées à la programmation à distance (édition de code, la compilation, la correction d'erreur, exécution, etc.).

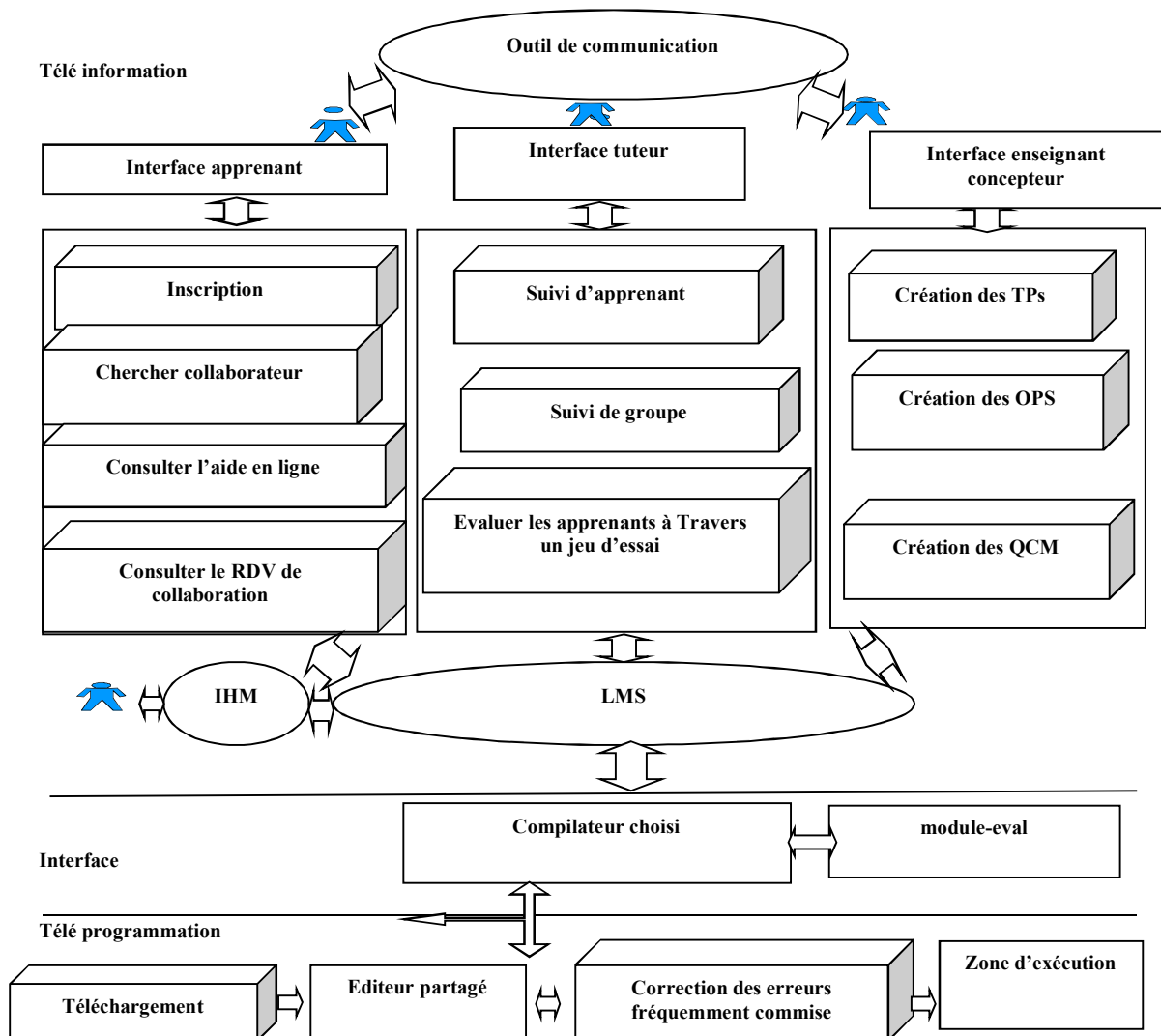


Figure 5.1: L'architecture de la plateforme E-TéléTPC@AALP [Boussaha & Bensebaa, 2009].

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

2.2. Le Principe de fonctionnement de la plateforme E-TéléTPC@AALP

Dans cette plateforme, il y a quatre acteurs humains (administrateur, tuteur, l'enseignant et les apprenants). Ces acteurs peuvent interagir, selon le choix et l'activité à réaliser, à l'aide de la plateforme et utilisent les trois niveaux : (téléformation, téléprogrammation et l'interface). Par exemple, si l'apprenant veut s'inscrire dans la plateforme, il doit utiliser le niveau téléformation, mais s'il veut écrire son programme, il doit utiliser le niveau téléprogrammation.

2.2.1. Interactions de l'administrateur avec la plateforme et les autres acteurs

L'administrateur est la personne responsable de la gestion de la plateforme. Il peut ajouter ou supprimer des apprenants dans un travail pratique, ajouter ou supprimer des enseignants (concepteurs d'un travail pratique) et ajouter ou supprimer des tuteurs. Il peut réaliser ces tâches en dehors de la session de travaux pratiques (lorsque les apprenants n'utilisent pas la plateforme pour la programmation) à travers une interface homme-machine (IHM).

2.2.2. Interactions de l'apprenant avec la plateforme et les autres acteurs

L'apprenant est une personne engagée dans l'apprentissage sur la plateforme. Il interagit avec la plateforme via les trois niveaux (téléformation, téléprogrammation, l'interface). Tout d'abord, l'apprenant doit s'inscrire en spécifiant un mot de passe et un nom d'utilisateur (niveau téléformation) afin d'être capable d'utiliser la plateforme. Une fois qu'il s'enregistre, il peut écrire son travail pratique (niveau téléprogrammation) . Dans le cas où il aurait des difficultés avec le travail pratique, il a la possibilité de consulter l'aide en ligne, demander de l'aide à son tuteur ou rechercher une réunion de collaboration avec d'autres apprenants qui sont inscrits dans la plateforme (niveau téléformation) s'il préfère travailler en groupe. Pour l'écriture de son programme en utilisant un langage de programmation existant, la plateforme offre aux apprenants un éditeur partagé et une zone d'exécution partagée avec le module compilateur choisi (niveau de l'interface). Si le langage de programmation n'est pas disponible, la plateforme permet aux apprenants de le télécharger.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

2.2.3. Interactions du tuteur avec la plateforme et les autres acteurs

Le tuteur est une personne chargée d'assister les apprenants. Tout d'abord, le tuteur doit s'inscrire via un mot de passe et un nom d'utilisateur. Une fois qu'il s'enregistre, il peut assister l'apprenant ou le groupe d'apprenants qui travaillent ensemble. L'assistance des apprenants est faite en les surveillant et en les aidant (si l'apprenant a les concepts du langage de programmation) ou avec l'utilisation d'un QCM (si l'apprenant ne connaît pas les concepts de base des travaux pratiques). Le tuteur peut également échanger les informations de travaux pratiques avec l'enseignant pour avoir plus de détails sur le travail demandé aux apprenants afin d'être capable de les aider dans leur travail pratique.

2.2.4. Interactions de l'enseignant avec la plateforme et les autres acteurs

L'enseignant peut, à travers la plateforme, créer un travail pratique, ajouter de nouvelles parties dans un travail pratique qui existe et de créer des QCM considérés comme des rappels sur les concepts prérequis et les concepts nécessaires pour résoudre le problème proposé. L'enseignant peut également communiquer avec le tuteur pour coordonner leurs stratégies d'enseignement.

3. Le générateur automatique de diagrammes UML

La première partie abordée est la conception d'un générateur automatique des diagrammes de classes et de communication UML.

Le générateur proposé permet à partir d'un code source écrit en langage C++ de générer deux diagrammes UML, un pour l'enseignant et l'autre pour l'apprenant.

- Diagramme de classes pour le côté statique
- Diagramme de communication pour le côté dynamique

3.1. Pour le diagramme de classes

On extrait une liste des classes principales. Cette liste contient elle-même deux autres listes des classes, la première liste contient l'ensemble des classes pour le diagramme de l'apprenant et la deuxième liste contient l'ensemble des classes pour le diagramme de l'enseignant. Chaque liste des classes est décomposée en liste des attributs, des méthodes, des héritages, des compositions et des agrégations.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

3.2. Pour le diagramme de communication

On extraira deux listes, l'une contient la liste des objets et l'autre contient la liste des messages :

- *La liste des objets* contient tous les objets et ses classes d'appartenance.
- *La liste des messages* est décomposée en trois colonnes, la première colonne contient l'objet émetteur, la deuxième colonne contient le message envoyé et la troisième colonne contient l'objet récepteur.

4. L'approche proposée pour l'évaluation de l'apprenant

Lors de la conception de notre approche, deux types *d'apparieurs (hybrides et composites)* sont utilisés pour comparer les diagrammes UML de l'apprenant à ceux de l'enseignant. Ces apparieurs sont complexes, car ils peuvent utiliser différents types de données.

4.1. Types d'apparieurs

4.1.1. Apparieurs hybrides

Les apparieurs hybrides sont des apparieurs qui utilisent un mélange de propriété en même temps, par exemple, propriétés comportementales entrelacées avec des propriétés structurelles [Meijer, 2008]. Ce sont une combinaison de différentes techniques d'appariement travaillée ensemble d'une manière prédéfinie fixe. Cette combinaison permet de déterminer les candidats de l'appariement en mettant l'accent sur les critères et de multiples sources d'information. Les apparieurs hybrides peuvent offrir un meilleur appariement avec une meilleure performance [Auxepaules, 2009].

Chaque *apparieur simple* a un résultat pour un type de donnée. Le résultat de l'apparieur d'hybride est obtenu par la combinaison des résultats obtenus par les apparieurs simples via la combinaison de différentes opérations arithmétiques (addition, soustraction, multiplication, division) dans le même apparieur hybride.

Par exemple, dans notre travail, *l'apparieur hybride 1* utilise les résultats calculés par *l'apparieur simple* qui calcule le *nombre de toutes les caractéristiques* et *l'apparieur simple* qui calcule le *nombre de caractéristiques communes* pour le *diagramme de classes*.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

L'apparieur hybride 2 utilise les résultats obtenus par plusieurs apparieurs simples (apparieur simple qui calcule la similarité de classe, apparieur simple qui calcule la similarité des attributs, apparieur simple qui calcule la similarité des méthodes) pour le diagramme de classes.

L'apparieur hybride 3 utilise les résultats donnés par l'apparieur simple qui calcule le nombre de toutes les caractéristiques et l'apparieur simple qui calcule le nombre de caractéristiques communes pour le diagramme de communication.

4.1.2. Apparieurs composites

Les *apparieurs composites* combinent les résultats évalués indépendamment par plusieurs apparieurs hybrides. Cette capacité à combiner des apparieurs est plus souple que la combinaison rigide de certaines techniques correspondantes qui sont exécutées simultanément ou dans un ordre fixe (dans les apparieurs hybrides). En revanche, un apparieur composite sélectionne à partir d'un ensemble d'apparieurs modulaires ceux qui sont basés, par exemple, sur le domaine d'application ou le formalisme des modèles considérés. De plus, un apparieur composite peut permettre à un ordonnancement flexible des apparieurs en l'exécutant à la fois simultanément et séquentiellement. Dans ce dernier cas, le résultat d'un premier apparieur est consommé et prolongé par un second apparieur pour réaliser une amélioration itérative du résultat de mise en correspondance [Auxepaules, 2009].

Par exemple, dans notre travail, le résultat de *l'apparieur composite 1* est obtenu par l'addition et la division des résultats obtenus par *l'apparieur hybride 1 et 2*.

Le résultat de l'apparieur composite 2 est obtenu par l'addition et la division des résultats obtenus par l'apparieur composite 1 et l'apparieur hybride 3.

4.2. Le processus d'évaluation de l'apprenant

Le processus d'évaluation de l'apprenant proposé consiste en quatre phases principales comme illustré dans la figure 5.2. Durant les deux phases 1 et 2 (zone en pointillés1), les productions de l'enseignant et celles de l'apprenant sont traitées séparément. Mais la réalisation des phases 3 et 4 (boîte en pointillés 2) nécessite la combinaison des productions finales (diagrammes de classes et de communication

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

décomposés) de l'enseignant et de l'apprenant, nous détaillons dans ce qui suit les quatre phases.

- **Phase 1**

Au cours de cette phase, l'apprenant et l'enseignant doivent développer deux programmes orientés objet C++ qui sont soumis à un processus d'analyse pour générer des diagrammes UML (diagrammes de classes et de communication).

Le diagramme de classes exprime la structure statique du programme, en matière de classes et de relations entre les classes. Les classes sont essentiellement organisées par agrégation, héritage ou relations d'association.

Le diagramme de communication décrit comment un ensemble d'objets collaborent-ils pour accomplir une tâche spécifique. Ils mettent l'accent sur les interactions dynamiques entre les objets (échange de messages), ainsi que leur synchronisation.

- **Phase 2**

Dans la deuxième phase, les diagrammes générés doivent subir un processus de décomposition. Il consiste à décomposer chaque diagramme en sous-diagrammes composants. Le diagramme de classes est décomposé en un ensemble de classes, un ensemble d'attributs et un ensemble de méthodes et un ensemble de relations entre les classes. Le diagramme de communication est décomposé en un ensemble d'objets et un ensemble de messages envoyés et un ensemble de messages reçus.

- **Phase 3**

Cette phase consiste à appliquer la technique d'appariement que nous avons adoptée et qui propose trois types d'apparieurs hybrides. Les deux premiers apparieurs (apparieur hybride 1 et apparieur hybride 2) sont appliqués sur les deux diagrammes de classes (apprenant, enseignant), tandis que le troisième (apparieur hybride 3) est appliqué sur les deux diagrammes de communication (apprenant, enseignant).

Dans cette phase, chaque apparieur hybride est appliqué entre les paires de sous-diagrammes (sous-diagramme apprenant, sous-diagramme enseignant). Le résultat issu de cette phase est :

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

- *alignement diagramme de classes (apprenant, enseignant)*

Ce résultat est obtenu après une application des deux apparieurs hybrides entre à la paire, diagramme de classes apprenant décomposé et diagramme de classes de l'enseignant décomposé.

- *alignement diagramme de communication (apprenant, enseignant)*

Ce résultat est obtenu après une application de troisième apparieur hybride à la deuxième paire, diagramme de communication de l'apprenant et diagramme de communication de l'enseignant.

- **Phase 4**

Lors de la dernière étape, nous appliquons deux apparieurs composites. Le premier apparieur composite (*apparieur Composite 1*) est utilisé pour combiner les résultats de *l'apparieur hybride 1* et *l'apparieur hybride 2* tandis que le second *apparieur composite 2* est utilisé pour combiner les résultats de *l'apparieur composite 1* et *L'apparieur hybride 3* afin d'obtenir une liste des similarités et des différences entre le programme de l'apprenant et le programme de l'enseignant.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

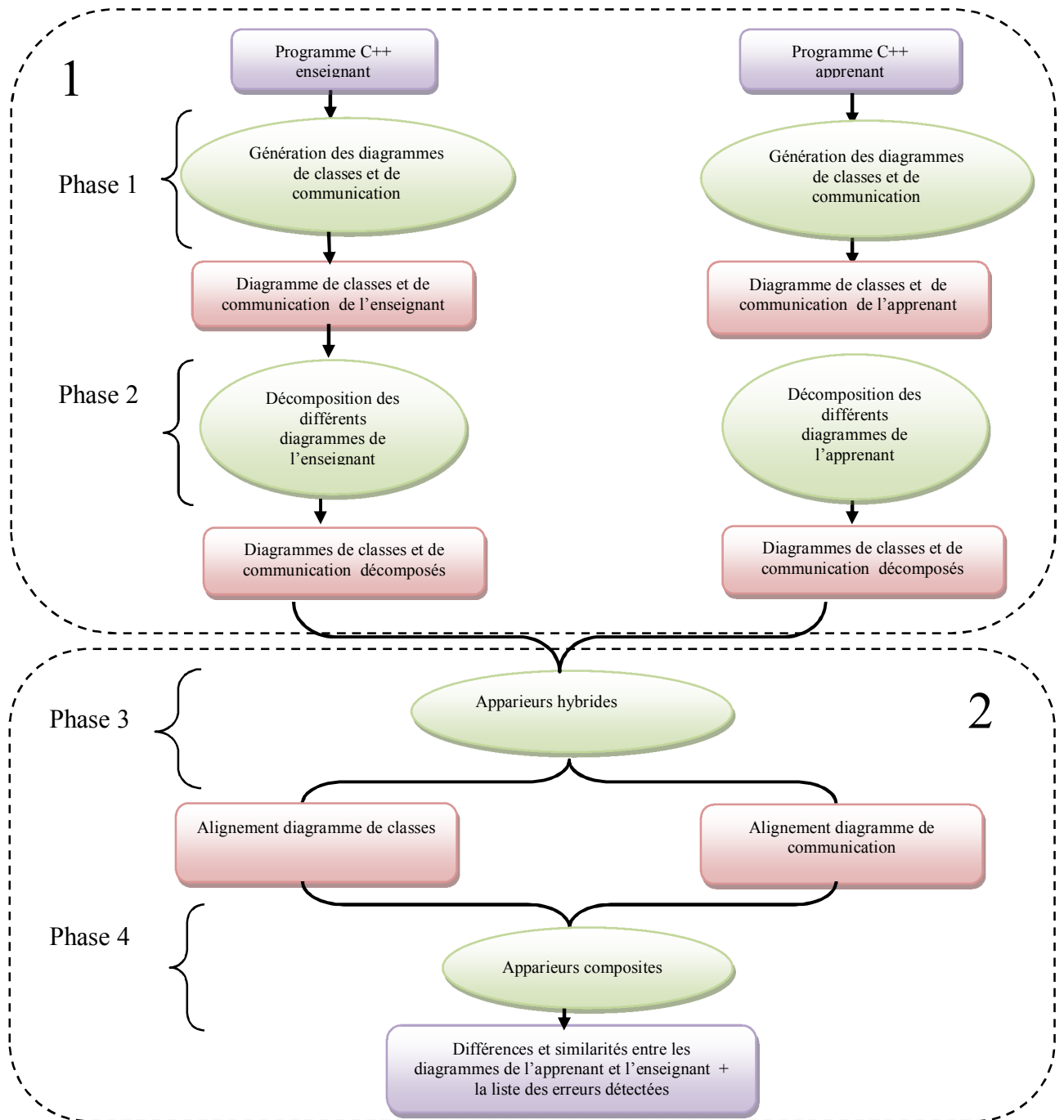


Figure 5.2: Méthodologie de l'approche proposée [Boussaha et al. , 2012]

5. Comparaison entre le programme de l'apprenant et celui de l'enseignant

5.1. Technique d'appariement proposée

Afin d'être capable d'appliquer notre technique d'appariement, les diagrammes UML générés doivent subir un processus de décomposition. Il consiste à décomposer chaque

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

diagramme en *sous-diagrammes* composants. Notez que nous nous intéressons, à la *comparaison syntaxique* entre les productions des apprenants avec celles qui sont élaborées par l'enseignant. En fait, deux représentations UML différentes peuvent toujours capturer une solution précise. Pour faire face à ce cas, nous devons prendre en compte, dans un autre projet à moyen termes, les aspects sémantiques (par exemple, l'utilisation des ontologies) de ces représentations UML.

La technique d'appariement, que nous proposons tient compte de plusieurs paramètres :

5.1.1. Type d'appariement appliqué

Parmi les différents types d'appariement que nous avons déjà étudiés (appariement de forme, appariement de schémas, appariement de graphe) nous avons choisi *l'appariement de graphe* parce que nous avons considéré chaque diagramme comme un graphe et notre objectif est de comparer les différents diagrammes engendrés sous forme de graphe.

5.1.2. La cardinalité de l'appariement

Nous avons utilisé *l'appariement univoque* avec une cardinalité (1..1) c'est-à-dire on va comparer chaque sommet d'un graphe avec un seul sommet de l'autre graphe.

La comparaison est fondée sur les identifiants de chaque composant dans le diagramme de classes et selon l'élément émetteur et l'élément récepteur et le message envoyé dans le diagramme de communication.

5.1.3. Algorithme de comparaison

Nous avons appliqué *l'algorithme Top-down*, car nous réalisons un appariement sur les structures hiérarchiques et plus l'algorithme Top down n'est moins coûteux qu'un *algorithme Bottom-down*.

5.1.4. La technique d'analyse utilisée

Les techniques fondées sur les contraintes sont les plus adaptées à notre hypothèse parce qu'elles se focalisent sur les contraintes internes des modèles.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

5.1.5. Types de diagrammes à comparer

La comparaison doit se faire pour les diagrammes *sous spécifiés*, *sur spécifiés* et *identiques*.

- Diagramme sous-spécifié : cela signifie que certains composants ne sont pas saisis dans les diagrammes de l'apprenant (diagramme de classes ou diagramme de communication) par rapport au diagramme de l'enseignant.
- Diagramme sur spécifié : cela signifie que certains composants sont ajoutés dans les diagrammes de l'apprenant par rapport au diagramme de l'enseignant.
- Diagramme identique : cela signifie que les diagrammes de l'apprenant et les diagrammes de l'enseignant sont tout à fait semblables.

Dans la méthodologie proposée, on va utiliser deux types d'apparieurs, apparieurs *hybrides* et d'autres *composites*.

5.2. Algorithme d'appariement développé

L'algorithme d'appariement développé est fondé sur le principe de *l'algorithme glouton* et les similarités entre les graphes décrites dans (sébastien sorlin et christine solnon, 2006).

L'algorithme accepte comme données d'entrée : Deux appariements m et m' (m pour les diagrammes de classes UML et m' correspond aux diagrammes de communication), deux diagrammes de classes UML (apprenant, enseignant) et deux diagrammes de communication UML (apprenant, enseignant). Comme il est cité ci-dessus, notre algorithme proposé utilise deux types d'apparieurs : apparieur hybride et composite en tenant compte de certains critères de comparaison. Ces apparieurs sont destinés à calculer :

- *apparieur hybride 1* : calcule la similarité entre les diagrammes de classes (apprenant, enseignant) par rapport à l'appariement m engendrait à partir de diagramme de classes.
- *apparieur hybride 2* : calcule la similarité entre les diagrammes de classes (apprenant, enseignant) par rapport aux contraintes prises en compte de chaque composant de diagramme de classes.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

- *apparieur hybride 3* : calcule la similarité entre les diagrammes de communication (apprenant, enseignant) par rapport à l'appariement m' engendré à partir de diagramme de communication.
- *apparieur composite 1* : calcule la similarité totale entre l'apparieur hybride 1 et 2.
- *apparieur composite 2* : calcule la similarité globale entre l'apparieur composite 1 et l'apparieur hybride 3.

5.2.1. Critères de comparaison

- *Critère1*: pour comparer deux classes (classe dans le diagramme de l'apprenant avec la classe appropriée dans le diagramme de l'enseignant) nous devons comparer leurs noms.
- *Critère2*: pour comparer deux attributs (attribut dans le diagramme de l'apprenant avec l'attribut approprié dans le diagramme de l'enseignant), nous devons comparer leurs noms, leurs visibilitées et leurs types.
- *Critère 3*: afin de comparer deux méthodes (méthode dans le diagramme de l'apprenant avec la méthode appropriée dans le diagramme de l'enseignant), nous devons comparer leurs noms, leurs visibilitées, attributs et leurs types de retour).

5.2.2. Algorithm

Entrées: Let:

$G1A (V1A, E1A)$: Diagramme de classes de l'apprenant , $V1A$: ensemble de sommets , $E1A$: ensemble d'arcs.

$G1T(V1T, E1T)$: Diagramme de classes de l'enseignant , $V1T$: ensemble de sommets , $E1T$: ensemble d'arcs.

$G2A(V2A, E2A)$: Diagramme de communication de l'apprenant, $V2A$: ensemble de sommets, $E2A$: ensemble d'arcs.

$G2T(V2T, E2T)$: Diagramme de communication de l'enseignant, $V2T$: ensemble de sommets, $E2T$: ensemble d'arcs.

m, m' : Appariements correspondant aux diagrammes de classes et de communication de l'enseignant respectivement.

$NbrClasses, NbrAttributes, NbrMethods$: nombre de classes, nombre d'attributs et le nombre de méthodes respectivement.

Sorties: Similarités entre le programme de l'apprenant et celui de l'enseignant.

5.2.3. Les étapes de déroulement de l'algorithme

1. La représentation des diagrammes de classes et de communication $G1A, G1T, G2A, G2T$ sous forme de *graphes étiquetés*.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

2. L'étiquetage des sommets et des arcs pour chaque graphe (*V1A, E1A, V1T, E1T, V2A, E2A, V2T, E2T*) pour les deux types de diagrammes (classes et communication).
3. Générer deux appariements (*m* et *m'*) le premier à partir de diagramme de classes de l'enseignant et le second à partir de diagramme de communication de l'enseignant.
4. Extraire l'ensemble de toutes les caractéristiques entre les diagrammes de classes *G1T* et *G1A*.
5. Extraire l'ensemble de *caractéristiques communes* entre les diagrammes de classes *G1T* et *G1A* par rapport à l'appariement *m*.
6. Extraire l'ensemble de *toutes les caractéristiques* entre les diagrammes de communication *G2T* et *G2A*.
7. Extraire l'ensemble de *caractéristiques communes* entre les diagrammes de communication *G2T* et *G2A* par rapport à l'appariement *m'*.
8. Calcul de *similarité* entre les deux types de diagrammes (classe, communication) : pour le diagramme de classes *G1T* et *G1A* nous avons calculé deux similarités : une par rapport à l'appariement *m* et l'autre par rapport *aux contraintes prises en compte par chaque composant de diagramme de classes* (classe, attribut, méthode). Par contre pour les diagrammes de communication *G2T* et *G2A* nous avons calculé une seule *similarité* par rapport à l'appariement *m'*.
9. Calcul de la *similarité totale* entre les diagrammes de classes *G1T* et *G1A*.
10. Calcul de la *similarité globale* entre les deux similarités (similarités totales de diagramme de classes et la similarité de communication).

Cet algorithme peut être décrit formellement comme suit :

Entrés

G1A,G1T :Diagrammes de classes
G2A,G2T :Diagramme de communication
V1A ,V1T,V2A,V2T : ensembles de sommets
E1A ,E1T ,E2A,E2T : ensembles d'arcs

Sorties

VarSim_m, Sim_m', Sim_c, Total_Sim,, Global_Sim : Real

Etape1.

[1] Representing_In_Labled_Graph (G1A, G1T, G2A, G2T)

Etape2.

For each $v_i \in V1A, v_j \in V1T, v_k \in V2A, v_l \in V2T$ do

[2] Associate_Label_To_Vertice (v_i, v_j, v_k, v_l)

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

```

For each  $e_i \in E1A, e_j \in E1T, e_k \in E2A, e_l \in E2T$  do
[3] Associate_Label_To_Edge( $e_i, e_j, e_k, e_l$ )
Etape3.
[4]  $m \leftarrow$  Generate-matching( $G1T$ )
[5]  $m' \leftarrow$  Generate-matching ( $G2T$ )
Etapes 4,5,6,7.
[6] Extract_All_Features( $G1T, G1A$ )
[7] Extract_Common_Features ( $G1T, G1A, m$ )
[8] Extract_All_Features( $G2T, G2A$ )
[9] Extract_Common_Features ( $G2T, G2A, m'$ )
Etape8.
[10]  $Sim\_m \leftarrow$  Calculate_Similarity1 ( $G1A, G1T, m$ ) // Apparieur hybride 1
[11]  $Sim\_c \leftarrow$  Calculate_Similarity2 ( $G1A, G1T, c$ ) // Apparieur hybride2
[12]  $Sim\_m' \leftarrow$  Calculate_Similarity1 ( $G2A, G2T, m'$ ) // Apparieur hybride3
Etape9.
[13]  $Total\_Sim \leftarrow$  Calculate_Total_Similarity( $Sim\_m, Sim\_c$ ) // Apparieur composite 1
Etape10.
[14]  $Global\_Sim \leftarrow$  Calculate_Global_Similarity( $Total\_Sim, Sim\_m'$ ) // Apparieur composite 2
[15] If ( $Global\_Sim = 1$ ) then (les deux diagrammes sont similaires)
[16] else (les deux diagrammes sont différents)

```

Figure 5.3 : l'algorithme d'appariement proposé [Boussaha et al. 2015]

La fonction **Representing_In_Labled_Graph** (ligne [1]) consiste à étiqueter les différents diagrammes pour extraire leurs caractéristiques.

Les deux fonctions **Associate_Label_To_Vertice** (ligne [2]) et **Associate_Label_To_Edge** (ligne [3]) associent les étiquettes à tous les sommets et les arcs de chaque graphique respectivement.

La fonction **Generate-matching** (lignes [4, 5]) génère l'appariement pour les diagrammes de l'enseignant (diagramme de classes ou de communication).

Pour extraire toutes les caractéristiques des deux diagrammes donnés, nous avons utilisé la fonction **Extract_All_Features** (lignes [6,8]). La fonction **Extract_Common_Features** (ligne [7, 9]) est utilisée pour extraire les caractéristiques communes entre les deux diagrammes donnés en utilisant l'appariement correspondant.

Selon l'appariement, la fonction **Calculate_Similarity1** (lignes [10, 12]) calcule la similarité entre deux diagrammes donnés du même genre ; diagramme de classes ou de communication respectivement. De la même manière, la fonction **Calculate_Similarity2** (ligne [11]) calcule la similarité Sim_c entre deux diagrammes de classes donnés à l'aide d'une liste de contraintes données.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

En utilisant les résultats retournés par les deux fonctions `Calculate_Similarity1` (ligne [10]) et `Calculate_Similarity2` (ligne [11]), nous pouvons calculer la similarité totale ***Total_Sim*** (ligne [13]) entre les diagrammes de classes de l'enseignant et de l'apprenant.

En utilisant la notion de pondération, la similarité totale peut être considérée comme $(W_m * Sim_m + W_c * Sim_c) / 2$, où W_m et W_c sont les poids attribués respectivement à la similarité entre diagrammes de classes de l'enseignant et de l'apprenant qui est basée sur l'appariement m et à la similarité entre les mêmes diagrammes qui est basée sur la liste des contraintes c .

Après l'obtention de similarité totale entre les diagrammes de classes de l'enseignant et de l'apprenant (*Total_Sim*), nous calculons la similarité globale (ligne [14]) entre les diagrammes de l'enseignant et de l'apprenant à l'aide de la fonction ***Calculate_Global_Similarity***. La similarité globale peut être considérée comme $(W_{cld} * Total_Sim + W_{comd} * Sim_m') / 2$, où W_{cld} et W_{comd} sont les poids attribués respectivement à des aspects statiques et dynamiques du programme C++ décrit par les diagrammes de classes et de communication.

Notons que ces poids (W_m , W_c , W_{cld} , W_{comd}) sont définis par l'enseignant en fonction de son objectif et de l'importance du programme C++.

Les différentes mesures citées ci-dessus ont une valeur pédagogique importante pour l'apprenant. Ils lui permettent de détecter et de corriger les erreurs qu'il a commises dans ses programmes à différents niveaux. Pour des exemples, la métrique sim_m' (appariement hybride 3) permet à l'apprenant de détecter et de corriger les erreurs pouvant apparaître dans son diagramme de communication, tandis que la valeur de la métrique *Total_Sim* (appariement Composite 1) permet à l'apprenant de détecter et de corriger les erreurs pouvant apparaître dans son diagramme de classes.

Par conséquent, ces mesures contribuent, en général, dans l'amélioration des compétences de programmation de l'apprenant. Ils aident l'apprenant, d'une part, à travailler de façon efficace, dans son travail pratique et, d'autre part, à planifier l'étape suivante.

Notez que l'approche proposée et l'environnement associés ne traitent pas avec le niveau de la conception de logiciels. Ils acceptent en entrée des deux programmes orientés objet écrits en C++.

Chapitre 5 Une approche basée sur l'appariement des graphes pour l'évaluation de l'apprenant.

6. Conclusion

Dans ce chapitre, nous avons présenté notre approche d'évaluation de l'apprenant. Il s'agit de comparer deux programmes orientés objet écrits en C++, l'un développé par l'enseignant et l'autre par l'apprenant. L'idée sous-jacente est en fait, de procéder à une comparaison indirecte des programmes à travers leurs représentations graphiques décrites à l'aide de digrammes UML (diagrammes de classes et de communication). Pour ce faire, nous avons développé un outil implémentant un algorithme d'appariement inspiré des algorithmes d'appariement des graphes.

Dans le chapitre suivant nous présentons notre environnement d'évaluation de l'apprenant comme un EIAH (environnement informatique pour l'apprentissage humain). L'environnement développé permet essentiellement : (1) la génération de diagrammes UML à partir du programme de l'apprenant, et (2) l'application des algorithmes d'appariement pour mesurer les degrés de similarité et de différence entre les diagrammes de l'apprenant et les diagrammes de l'enseignant. Notre idée a été testée sur des exemples concrets et les résultats obtenus ont montré une bonne fiabilité de l'approche proposée. L'environnement développé est illustré par une étude de cas concrète.

Environnement d'évaluation de l'apprenant

1. Introduction

L'environnement d'évaluation repose sur une démarche pédagogique totalement socioconstructiviste¹⁰, destinée à l'évaluation des compétences des apprenants dans le domaine des TélecTPs en langages de programmation orientés objets dans un EIAH.

La situation d'apprentissage est présentée sous forme de résolution de problèmes. Elle consiste à évaluer une solution de TélecTPs pour un problème donné, cette évaluation étant formative et sommative.

L'approche proposée dans le chapitre précédent part de l'idée que pour résoudre un problème, il faut le représenter en graphes (diagramme de classes et de communication). Donc, l'environnement développé permet essentiellement: (1) la génération de diagrammes UML à partir du programme de l'apprenant, et (2) l'application des algorithmes d'appariement pour mesurer les degrés de similarité et de différence entre les diagrammes de l'apprenant et ceux de l'enseignant.

L'environnement développé a été testé en utilisant le test de Student afin d'enquêter sur les avantages de notre approche. Nous avons effectué deux expérimentations en utilisant notre plateforme E-TélecTPC@AALP avec les étudiants débutants dans les TPs de la programmation orientée objet. Les résultats obtenus des expérimentations sont satisfaisants.

¹⁰ **Socioconstructiviste** : cette pédagogie est centrée sur l'apprenant. C'est l'élève qui apprend par l'intermédiaire de ses représentations. La construction d'un savoir bien que personnelle s'effectue dans un cadre social. Les informations sont en lien avec le milieu social, le contexte et proviennent à la fois de ce que l'on pense et de ce que les autres apportent comme interactions.

2. L'architecture générale de l'outil d'évaluation

L'environnement d'évaluation développé est composé de trois modules principaux: l'outil éditeur, l'outil générateur, et l'outil apparieur. La composante la plus importante est l'outil apparieur qui est implémenté en utilisant l'algorithme d'appariement développé qui évalue la solution de l'apprenant en termes de valeurs de similarité. Son architecture est présentée dans la figure 6.1.

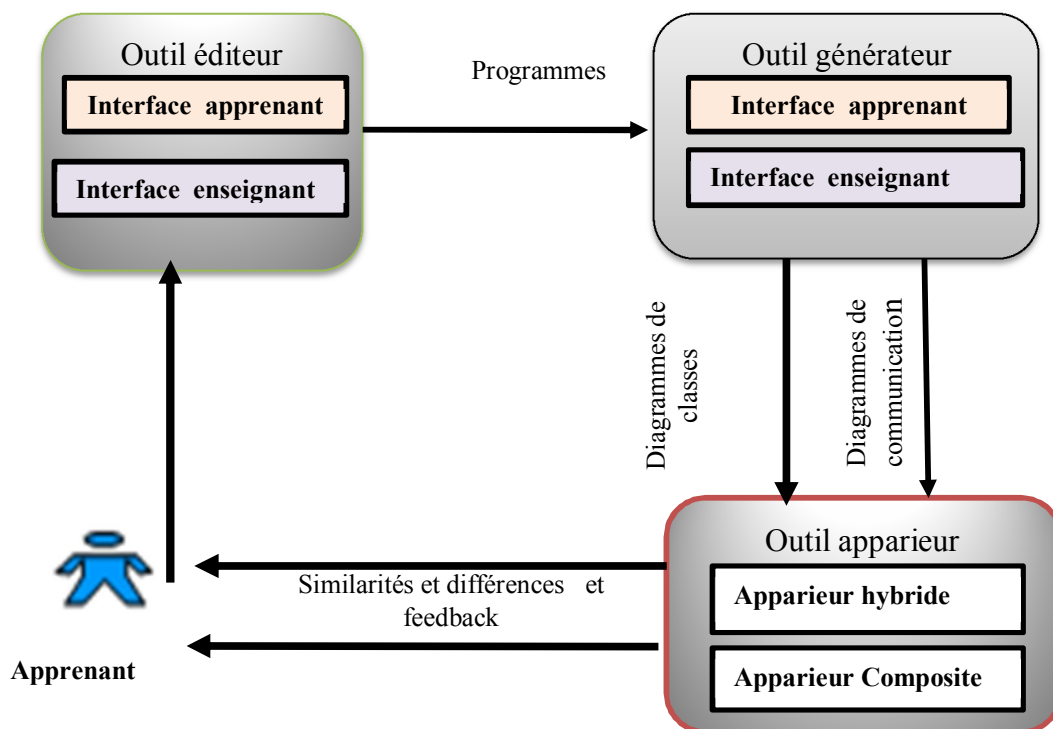


Figure 6.1 : L'architecture générale de l'outil d'évaluation

2.1. L'outil éditeur

Permet aux apprenants et enseignants d'écrire les codes des programmes.

2.2. L'outil générateur

Permet aux apprenants et enseignants de générer les différents diagrammes (diagrammes de classes et de communication) à partir de leurs codes.

2.3. L'outil apparieur

Utilise l'algorithme d'appariement pour comparer les programmes de l'apprenant à ceux de l'enseignant. Il indique également à l'apprenant l'ensemble des similitudes et des

différences. Un feedback montre l'ensemble des erreurs détectées dans le programme de l'apprenant.

3. Présentation de l'environnement

Nous avons développé, le long de ce travail, un environnement soutenant l'approche proposée. L'environnement offre plusieurs outils pour guider les utilisateurs (enseignants et apprenants) pendant l'exercice de leurs activités. Nous présentons dans ce qui suit notre environnement en utilisant une étude de cas concrète. Il s'agit d'un processus de réservation d'hôtel simple, quand un client arrive à l'hôtel, il présente sa carte d'identité, le réceptionniste de l'hôtel vérifie la disponibilité des chambres, puis propose au client une liste des pièces disponibles, ainsi que leur type et leur tarif. Le client en choisit une chambre et informe de la date de début de réservation, le réceptionniste de l'hôtel lui donne alors le numéro de la chambre choisie.

3.1. L'outil éditeur

Avec l'utilisation de l'environnement développé, l'enseignant et l'apprenant peuvent écrire leurs programmes orientés objet en C++ pour la mise en œuvre du processus de réservation de l'hôtel. Les figures 6.2 et 6.3 illustrent les programmes de l'enseignant et de l'apprenant respectivement. Les parties qui ne figurent pas dans le programme de l'apprenant sont marquées en couleur bleue. Notons que l'apprenant ne peut utiliser l'environnement qu'après édition du programme de l'enseignant et génération des diagrammes UML correspondants.

```

auto assessment tool
Teacher Learner Matching
c++ Editor
diagrams generator
UML class diagram
UML communication diagram

class person{
public:
string first_name;
string last_name;
string address;
};
class room_booking{
private:
int id_card;
int room_number;
int start_date;
int end_date;
bool reserve;
bool smoking;
};
class hotel_booking{
private:
room_booking *rb;
int nb_room;
public:
bool existe(){
int get_nbfreeroom(){
void reserve(int nb,int nb_ch){
};
class the hotel receptionist:person{

```

Figure 6.2 : Programme de l'enseignant

```

auto assessment tool
Teacher Learner Matching
c++ Editor
diagrams generator
UML Class diagram
UML communication diagram

class person_a{
public:
string first_name;
string last_name;
string address;
};
class hotel_booking_a{
private:
int nb_room;
public:
void reserve(int nb,int nb_ch){
};
class the_hotel_receptionist_a:person_a{
private:
int id;
hotel_booking_a hotelbooking;
void verif(){
void set(int id){hotelbooking.reserve(id,nb);
};
class client_a:person_a{
the_hotel_receptionist_a p;
int id card;

```

Figure 6.3 : Programme de l'apprenant

3.2. L'outil générateur

Comme il est mentionné ci-dessus, la comparaison des deux programmes édités se fait par comparaison de leurs représentations graphiques. Tout d'abord, l'enseignant utilise l'outil générateur afin de générer des diagrammes de classes et de communication (figures 6.4 et 6.6) de son programme C++. Ces diagrammes doivent être stockés pour une utilisation ultérieure. Après cela, l'apprenant doit, à son tour, générer des diagrammes de classes et de communication (figure 6.5 et 6.7) de son programme. Les deux diagrammes de classes (figure 6.4 et 6.5) ne sont pas complètement identiques. La classe *room_booking* et les trois membres : (l'attribut *rb* et les deux méthodes *Existe ()* et *get_nbrfreeroom ()*) ne sont pas saisis dans le diagramme de classes de l'apprenant. En outre, les deux messages *Existe ()* et *get_nbrfreeroom()* manquent au diagramme de communication de l'apprenant.



Figure 6.4: Diagramme de classe de l'enseignant

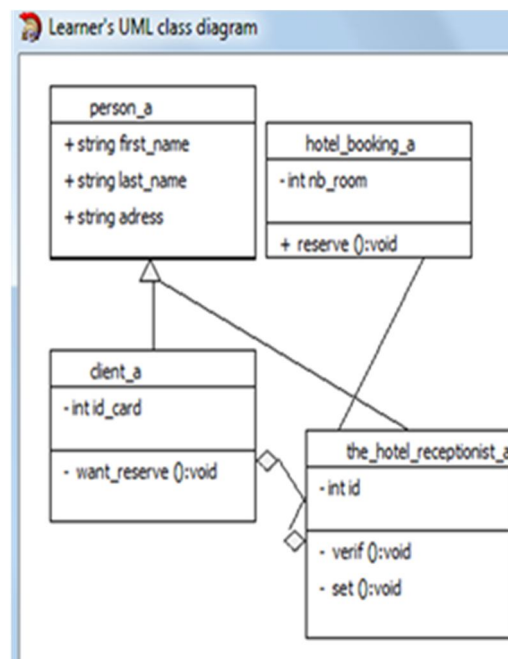


Figure 6.5 : Diagramme de classe de l'apprenant

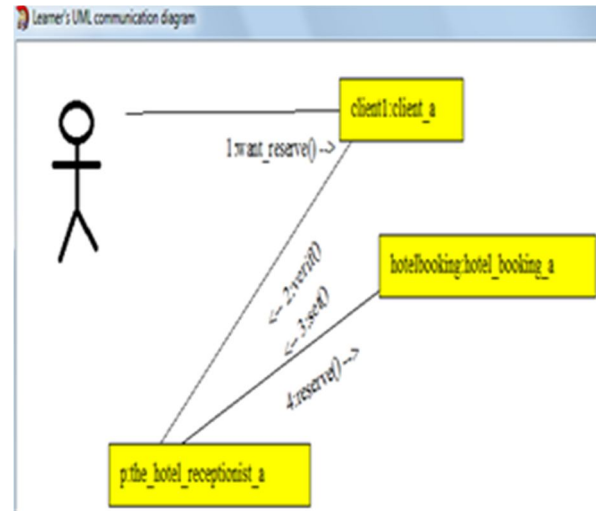
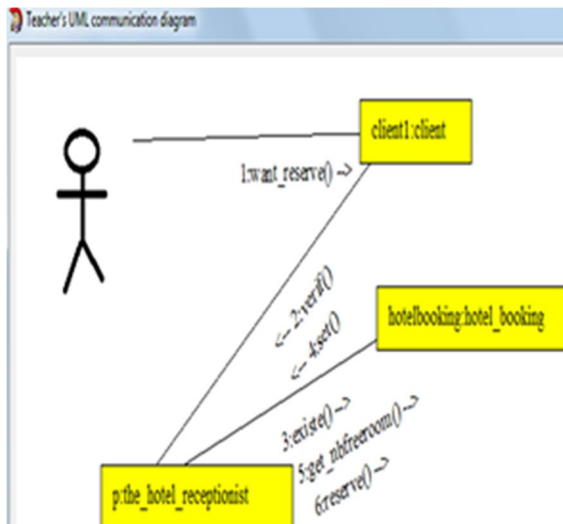


Figure 6.6 : Diagramme de communication de l'enseignant **Figure 6.7 :** Diagramme de communication de l'apprenant

Avant de présenter le reste des caractéristiques de notre environnement, il est important de noter que, parmi les résultats engendrés par l'environnement (illustrés par les différentes figures), il y en a ceux qui sont directement pertinents pour l'apprenant (figures 6.5, 6.7, 6.15, 6.16) alors que d'autres (figures 6.8, 6.9, 6.10, 6.11, 6.12, 6.13 et 6.14) sont optionnelles (l'apprenant n'est pas obligé de les consulter ou de les comprendre), ils sont justes utilisés pour plus de précision (c'est-à-dire ils sont utilisés juste pour montrer et valider la méthodologie de notre approche).

3.3. L'outil apparieur

L'objectif principal de l'environnement mis au point permet à des apprenants d'évaluer leurs compétences en programmation orientée objet C++, c'est pourquoi, dans la suite du TP, l'apprenant lui-même s'auto-évalue (l'enseignant n'intervient pas dans le reste du processus d'évaluation).

3.3.1. Processus de comparaison de diagrammes de classes générés

Pour comparer les deux diagrammes de classes générés plusieurs étapes sont nécessaires :

Etape 1 : *Etiquetage des sommets et des arcs pour chaque diagramme*

La figure 6.8 représente tous les sommets et les arcs étiquetés pour les diagrammes de classes de l'enseignant et de l'apprenant respectivement.

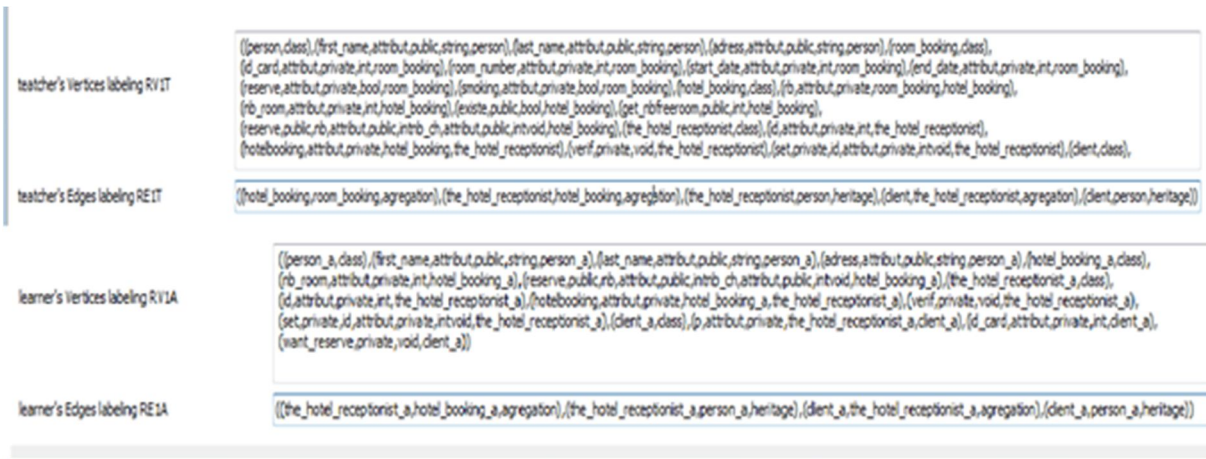


Figure 6.8 : L'ensemble des nœuds et des arcs étiquetés dans le diagramme de classes de l'enseignant et de l'apprenant

Etape 2 : La génération de l'appariement m à partir de diagramme de classes de l'enseignant

L'appariement engendré à partir de diagramme de classes de l'enseignant est représenté par la figure 6.9. C'est un ensemble de couples (class_name1, class_name2) où le nom_class1, nom_class2 représente les noms de classes dans le diagramme de classes de l'enseignant et dans le diagramme de classes de l'apprenant respectivement.

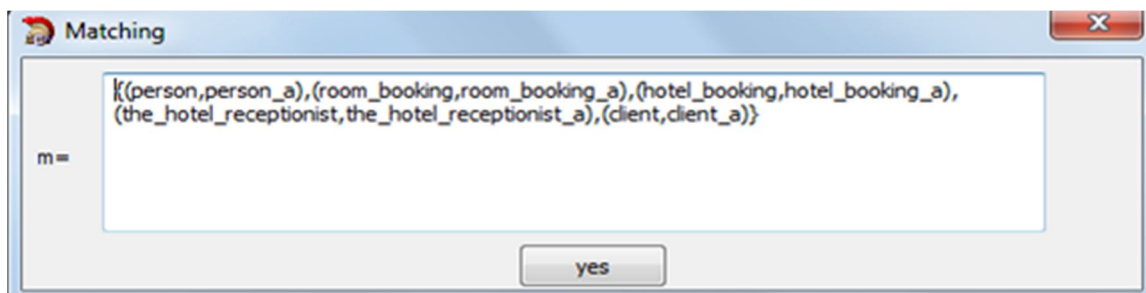


Figure 6.9 : L'appariement m généré à partir de diagramme de classes de l'enseignant

- **Mesure de similarités**

- *Similarité par rapport l'appariement m*

Pour calculer la similarité Sim_m entre les deux diagrammes de classes UML selon l'appariement m , nous devons utiliser l'apparieur hybride 1 (ligne [10] dans l'algorithme proposé). Pour cela, nous avons tout d'abord extrait les caractéristiques communes entre les deux diagrammes de classes et toutes les caractéristiques des deux diagrammes de classes par rapport à l'appariement m . Il y a quarante caractéristiques communes entre les deux diagrammes et cinquante et une caractéristiques des deux diagrammes (figure 6.10).

Cette similarité est obtenue par la division du nombre de caractéristiques communes ($G1A \cap G1T$) par le nombre de toutes les caractéristiques ($RV1A \cup RV1T \cup RE1A \cup RE1T$) par rapport à l'appariement m (soit 40/51). Le résultat est approximativement égal à peu près 0,784 (figure (6.10)), parce que les deux diagrammes de classes sont différents.

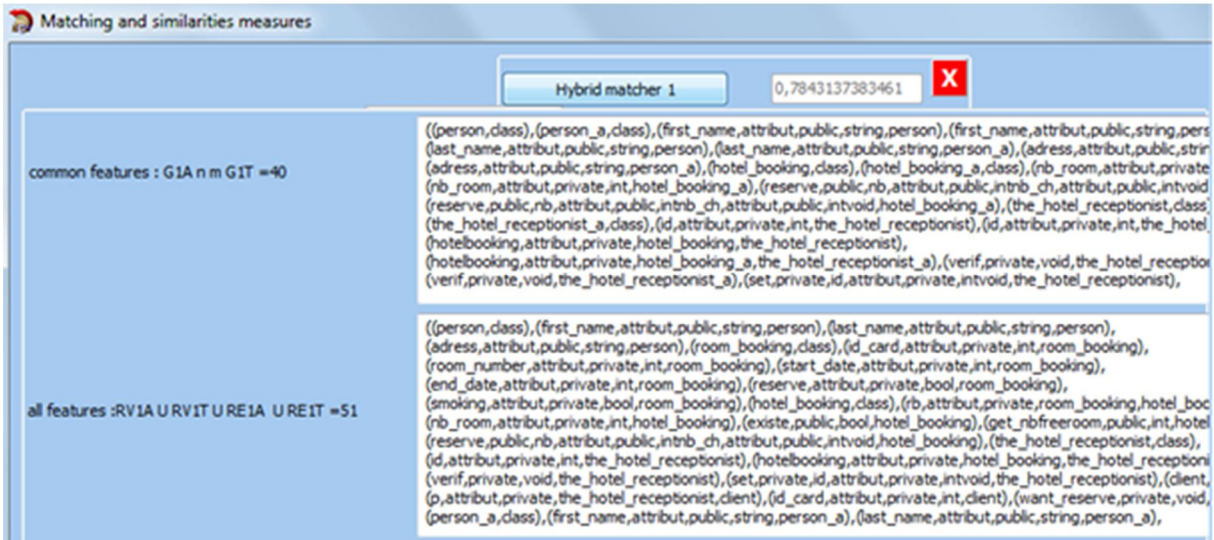


Figure 6.10: Mesure de similarité entre les deux diagrammes de classes (enseignant, apprenant) par rapport à l'appariement m

- *Similarité par rapport aux contraintes*

De la même manière, pour calculer la similarité Sim_c entre les deux diagrammes de classes par rapport aux contraintes proposées, nous devons utiliser l'apparieur hybride 2 (ligne [11] dans l'algorithme proposé). Le résultat obtenu est approximativement égal à 0,722 (figure 6.11). Ce résultat signifie qu'il y a des différences entre le diagramme de classes de l'apprenant et de celui de l'enseignant.

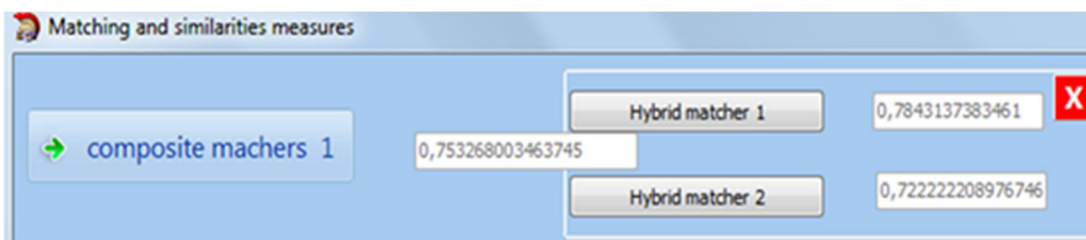


Figure 6.11: Mesure de similarité totale entre les deux diagrammes de classes (enseignant, apprenant)

- *Similarité Totale*

Pour calculer la similarité totale ($Total_Sim$) entre les deux diagrammes de classes UML, nous devons appeler l'apparieur composite 1 (ligne [13] dans l'algorithme proposé) (figure 6.11). Ce dernier a besoin du résultat de l'apparieur hybride 1 et de

l'apparieur hybride2. Total_Sim est calculé par moyenne pondérée des deux résultats $(Sim_m * Wm + Sim_c * Wc) / 2$. Notons que les poids Wm et Wc sont tous deux censés égales à la valeur 1. Donc, $(1 * 0,784 + 1 * 0,722) / 2 = 0,753$. Ce résultat signifie que le diagramme de classes de l'apprenant est un peu différent de celui de l'enseignant (deux diagrammes de classes sont tout à fait similaires si la similarité totale est égale à la valeur 1). Lorsque cette mesure est égale à un l'environnement affiche le message : "il n'y a pas d'erreurs" mais, lorsque cette mesure est inférieure à un le message d'erreur : "il y a des erreurs" apparaît. Dans ce cas, l'environnement engendre un retour à la saisie du texte pour identifier toutes les erreurs et offres de l'aide pour les corriger.

3.3.2. Processus de comparaison de diagrammes de communication générés

De la même manière, nous reprenons les mêmes étapes suivies pendant le processus de comparaison des diagrammes de classes pour comparer les diagrammes de communication générés. La figure 6.12 montre l'ensemble des sommets et des arcs étiquetés pour les diagrammes de communication de l'enseignant et de l'apprenant respectivement.

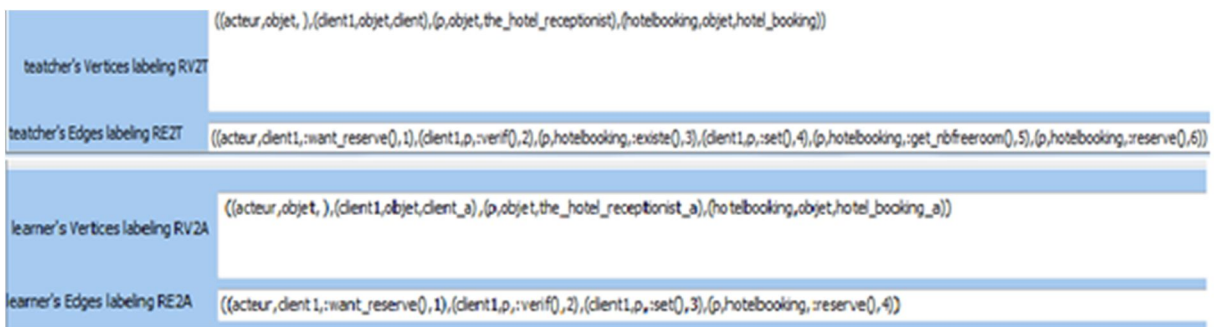


Figure 6.12: L'ensemble des nœuds et des arcs étiquetés dans le diagramme de communication de l'enseignant et de l'apprenant.

Tout comme pour les diagrammes de classes, la comparaison de diagramme de communication se fait par rapport à l'appariement m' (figure 6.13).

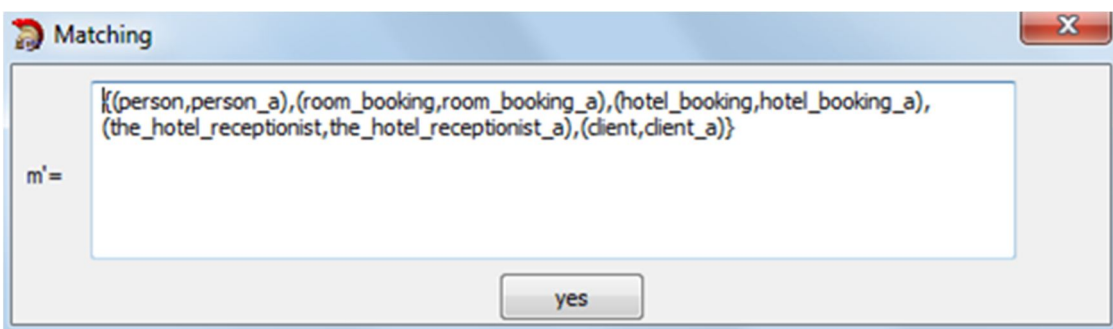


Figure 6.13: L'appariement m' généré à partir de diagramme de communication de l'enseignant

Pour calculer la similarité Sim_m' entre les deux diagrammes de communication par rapport l'appariement m' , nous devons appeler l'apparieur hybride 3 (ligne [12] dans l'algorithme proposé). Pour cela, nous devons, tout d'abord, extraire les caractéristiques communes et toutes les caractéristiques entre les diagrammes de communication de l'apprenant et de l'enseignant par rapport l'appariement m' . Il y a douze caractéristiques communes entre les deux diagrammes et dix-huit caractéristiques des deux diagrammes (Figure 6. 14).

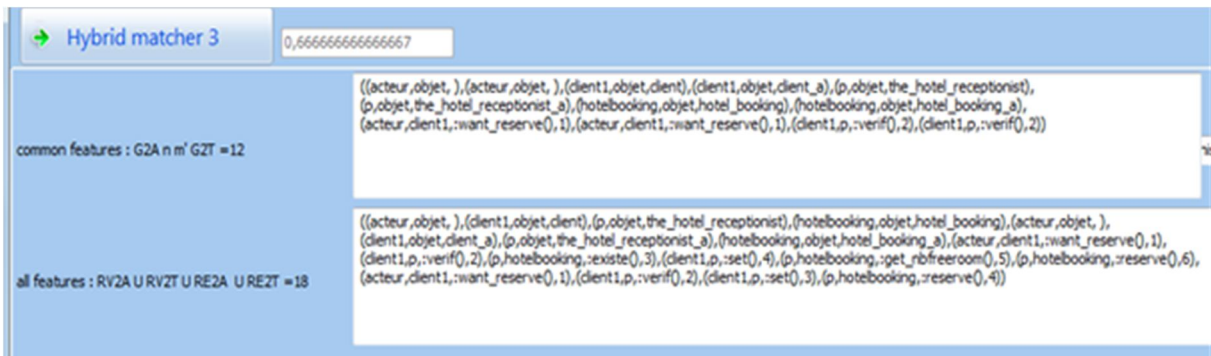


Figure 6.14: *Mesure de similarité totale entre les deux diagrammes de communication (enseignant, apprenant)*

L'apparieur hybride 3 (figure 6.14) est mis en œuvre en tant que la division du nombre de caractéristiques communes ($G2A \cap m'G2T$) par le nombre de toutes les caractéristiques ($RV2A \cup RV2T \cup RE2A \cup RE2T$) par rapport de l'appariement m' . Donc, $18/12 = 0,666$. Ce résultat indique que le diagramme de communication de l'apprenant est différent de celui de l'enseignant.

3.3.3. Le calcul de la similarité globale entre les diagrammes de l'enseignant et les diagrammes de l'apprenant

La figure 6.15 illustre les différentes mesures de similarités calculées, en particulier, la similarité globale $Global_Sim$ entre les diagrammes de l'enseignant et de l'apprenant. Cette dernière est calculée par l'apparieur composite 2 ((ligne [14] dans l'algorithme proposé). Ce dernier a besoin du résultat de l'apparieur composite 1 et de l'apparieur hybride 3. $Global_Sim$ est calculée par $(Wcld * Total_Sim + Wcomd * Sim_m') / 2$. Donc, $(0,753 * 1 + 0,666 * 1) / 2$. Le résultat signifie que le programme de l'apprenant est différent de celui de l'enseignant.

sont absents, les méthodes et les attributs sont manquants ou incorrects, etc.). La figure 6.16 illustre les erreurs détectées dans le programme de l'apprenant.

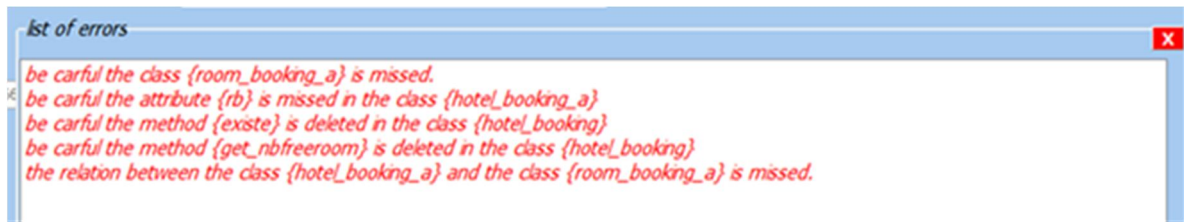


Figure 6. 16 : La liste des erreurs détectées dans le programme de l'apprenant

4. Évaluation de l'outil

4.1. l'utilité de notre outil

L'environnement d'auto-évaluation développé est un outil d'évaluation des étudiants débutants en programmation orientée objet dans lequel ils peuvent effectuer leurs travaux pratiques dans la programmation orientée objet avec la plateforme E-TéléTPC@AALP. L'efficacité de notre outil a été évaluée à la fois pour les enseignants et les apprenants:

4.1.1. Pour l'enseignant

- Réduire le problème d'encombrement des apprenants.
- Diminuer le problème d'attente des enseignants par les apprenants.
- Libérer les enseignants des tâches répétitives et de routine de l'évaluation des apprenants.

4.1.2. Pour les apprenants

- Motiver les apprenants,
- Encourager les activités de formation en programmation et aider les apprenants à se préparer pour les tests,
- Contribuer à l'avancement des compétences des apprenants,
- La formation de bons programmeurs et développeurs (i e, il ne se concentre pas uniquement sur les résultats obtenus par le code écrit. Mais il fournit des conseils aux apprenants pour écrire des codes lisibles et optimaux grâce à l'approche d'appariement, que nous avons appliquée).

4.2. Intégration de l'outil d'évaluation dans la plateforme : analyses et discussions

Dans ce qui suit, nous étudions les avantages de l'utilisation de l'outil d'évaluation au cours des travaux pratiques. Pour atteindre cet objectif, et avant de discuter notre expérience en ligne, soixante (60) étudiants ont été sélectionnés au hasard pour participer à cette expérience.

Tous les étudiants sont novices dans le domaine de la programmation orientée objet. Par conséquent, ils n'avaient pas de prérequis au travail pratique de langage de programmation orienté objet. Les participants ont été divisés au hasard en deux groupes qui sont le groupe 2 (n = 30) - dans lequel l'outil d'évaluation a été intégré dans la plateforme -, et groupe1 - dans lequel l'outil d'évaluation n'a pas été intégré dans la plateforme-. Nous étudions les effets de l'outil d'évaluation, d'une part, pour la maîtrise des concepts du langage de programmation orienté objet et de la résolution des situations de blocage par les apprenants. D'autre part, nous étudions les effets de cet outil pour le nombre d'interventions des enseignants assistants des travaux pratiques. Nous voulons prouver que les étudiants du groupe 2 étaient mieux en mesure de résoudre leurs problèmes que les étudiants dans le groupe 1.

Dans la première étape, les étudiants ont été instruits dans la connaissance de base de la programmation orientée objet, langage C ++, Avec travaux pratiques. Après avoir reçu cette connaissance fondamentale, nous avons demandé aux étudiants de passer un pré-test, qui visait à évaluer leurs connaissances de base avant de participer à l'apprentissage. Dans la deuxième étape, les étudiants du groupe 2 étaient disposés à apprendre avec la plateforme qui intègre l'outil d'évaluation. Alors que, les étudiants du groupe1 apprennent avec une autre version du système sans outil d'évaluation. Après avoir effectué l'activité d'apprentissage, les étudiants ont passé un post-test.

4.2.1. Résultat et analyses pour les apprenants

4.2.1.1. Pré-test (avant l'intégration de l'outil d'évaluation dans la plateforme)

	Group 1	Group2
Moyenne	10,025	09,96
SD	2,1298	2,1825
T-test		0,1197
df		58
p-valeur		0,9051

Table 6.1: T-test Avant l'intégration de l'outil d'évaluation dans la plateforme

SD: déviation standard, t-test: test de Student, df: degré de liberté , P-valeur: valeur de probabilité.

Le t-test a été effectué en premier, afin de déterminer si le groupe 1 et groupe 2 avaient la même connaissance préalable dans le domaine étudié. Comme on peut le voir dans la table 6.1 et la figure 6.17, il n'y avait aucune différence significative entre le groupe 1 et le groupe 2 dans leur connaissance préalable ($P > 0,05$). Ce résultat implique que le niveau de ces deux groupes ne diffère pas de manière significative avant l'expérience. Autrement dit, les deux groupes d'étudiants avaient des capacités statistiquement équivalentes avant de prendre l'unité de sujet.

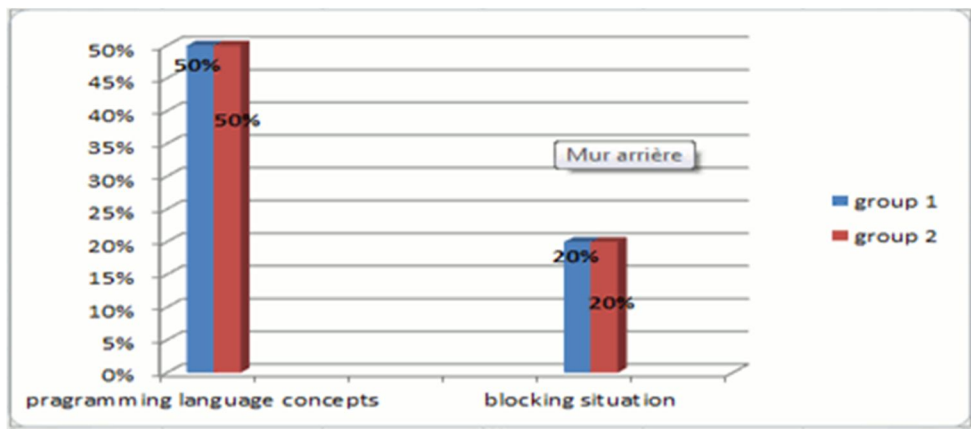


Figure 6.17 : Les résultats avant l'intégration de l'outil d'évaluation dans la plateforme

Comme le montre la figure 6.17, pour le groupe 1 et le groupe 2 seulement 50% des apprenants maîtrisent les concepts du langage de programmation orienté objet et seulement 20 % peuvent résoudre leurs situations de blocage. Donc, il n'y avait pas de différences significatives entre le groupe 1 et le groupe 2 dans leur connaissance préalable.

4.2.1.2. Post-test (Après l'intégration de l'outil d'évaluation dans la plateforme)

Le t-test a été réalisé afin de comparer la moyenne des résultats pour les deux groupes. Le t-test a déterminé que les différences mesurées entre les moyennes du groupe 1 et le groupe 2 étaient significativement différentes. Cette différence est due à l'apprentissage grâce à l'outil d'évaluation donné au groupe 2. Les résultats montrent que le groupe 2 maîtrise nettement mieux que le groupe 1 dans le post-test ($t = - 2,0545$, $df = 58$, $p = 0,0444 < 0,05$). La table 6. 2 et la figure 6.18 montrent les résultats de la comparaison.

	Group 1	Group2
Moyenne	10,0833	11,25
SD	2,0734	2,3184
T-test		-2,0545
df		58
p-valeur		0,0444*

Table 6.2 : *T-test après l'intégration de l'outil d'évaluation dans la plateforme*

*Différence significative, SD: déviation standard, T-test: test de Student, df: degré de liberté, P-valeur: valeur de probabilité.

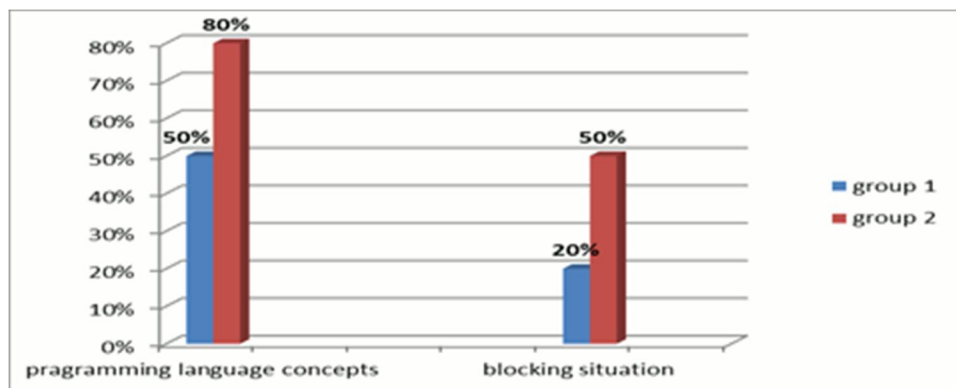


Figure 6.18: *Les effets de l'outil pour la maîtrise des concepts et la résolution des situations de blocage pour les apprenants*

Comme le montre la figure 6.18, pour le groupe 1 seulement 50% des apprenants peuvent maîtriser les concepts du langage de programmation orienté objet et seulement 20% peuvent résoudre leurs situations de blocage. En revanche pour le groupe 2, 80 % des apprenants peuvent maîtriser les concepts du langage de programmation orienté objet et 50% d'entre eux qui peuvent résoudre leurs situations de blocage. Nous remarquons que le groupe dans lequel l'outil d'évaluation est intégré était mieux en mesure de résoudre leur travail pratique que celui dans lequel l'outil d'évaluation n'est pas intégré.

4.2.2. Résultat et analyses pour les enseignants assistants de TP

Dans la figure 6.19, on observe pour le groupe 1, que les demandes d'intervention en ligne de l'assistant de travaux pratiques (tuteurs) est de 90 interventions pour une session de 2 heures (3 interventions par apprenant), mais pour le groupe 2 les demandes d'intervention en ligne devient 60 (2 interventions par apprenant). Nous remarquons que l'outil d'évaluation libère mieux les enseignants assistants de travaux pratiques.



Figure 6.19 : Les effets de l'outil pour diminuer le nombre de demandes d'intervention en ligne des assistants de TP

5. Conclusion

Parmi les objectifs de cet outil est l'amélioration des compétences de résolution des problèmes en programmation à travers une évaluation formative.

Avec cet outil, on peut évaluer des productions d'apprenants et assurer l'objectivité dans la correction ainsi qu'une grande fidélité à l'évaluation, même pour des grands groupes d'apprenants, contrairement à l'évaluation de l'enseignant humain qui est toujours biaisée par des facteurs liés à l'évaluateur (sa personnalité, etc.).

L'apport de cet outil est considérable aussi bien du côté des apprenants que du côté des enseignants : en effet, il a motivé les apprenants et les a encouragés dans leurs activités de formation en programmation et les a aidés à se préparer pour les tests. Du côté des assistants, cet outil permet de diminuer la perte de temps aux apprenants dans l'attente des assistants et permet de libérer les enseignants des tâches répétitives et de routine dans l'évaluation des apprenants.

Conclusion générale et perspectives

Bilan

Pour renforcer l'apprentissage chez les apprenants plusieurs EIAH ont été développées et plusieurs sont disponibles sur le web en accès libre. Malheureusement, très peu entre ces EIAH qui sont centrés sur l'apprentissage des travaux pratiques à distance.

Quoique l'enseignement des travaux pratiques se soit révélé indispensable dans les formations scientifiques et techniques, en présentiel comme en distanciel, il manque d'un véritable besoin. Souvent, les apprenants sont privés de cet indispensable enseignement et pour causes: le problème de disponibilité des assistants, le problème d'encombrement des apprenants, le matériel est onéreux et ne peut être dupliqué. Afin de minimiser ces problèmes, l'enseignement à distance devrait répondre à ces besoins. Par ailleurs, les auteurs de ces environnements s'intéressent généralement à faire apprendre une masse de connaissances sans s'assurer de son acquisition chez les apprenants. L'évaluation, cet acteur fréquemment manquant dans plusieurs activités pédagogiques et souvent mis à part, est une véritable garante de réussite et de mesure d'efficacité. Elle est l'instrument d'orchestration de n'importe quelle formation. Son rôle est de promouvoir l'apprentissage.

Le présent travail de thèse traite d'une problématique s'articulant sur les deux axes de recherche cités en dessus : les environnements informatiques d'apprentissage humain dans les TéléTPs, et l'évaluation automatisée des connaissances des apprenants. L'intérêt de ce type de problématique est particulièrement significatif de nos jours du fait dans l'apprentissage humain.

Dans la littérature, la majorité des systèmes d'évaluation des productions des apprenants sont en algorithmiques, le domaine de TP en programmation est omis. Ces systèmes se fondent sur deux approches : exécution de programmes ou analyse de programme. Les points faibles majeurs de ces deux dernières approches sont qu'on ne peut pas évaluer les compétences de conception de programme d'un côté, prenant en compte par exemple la première approche, elle est comparée toujours les résultats obtenus par rapport aux résultats attendus et ne traite pas ni le code ni les compétences de conceptions, la même chose pour la deuxième approche, elle concentre sur l'analyse de programme par rapport à des programmes prédéfinis ce qui réduit la liberté des apprenants dans la

Conclusion générale et perspectives

conception des solutions. D'un autre côté, nous pouvons confirmer que les deux approches ne fournissent pas une évaluation totalement automatique, elles restent dans le besoin d'intervention de l'être humain dans certains cas, où la situation de prendre la décision est vague et ambiguë.

L'objectif de notre travail a consisté à valider l'importance d'apprentissage des TéléTTPs (particulièrement dans le domaine de la programmation orientée) avec des EIAH consacré à ce type de disciplines pour supporter la difficile phase d'initiation à la programmation et éliminer les problèmes de ce type d'activités d'apprentissage avec les méthodes classiques (en présentiel). Pour cela, nous avons conçu, sur la base de mon travail de magister sur la conception et le développement d'une plateforme d'apprentissage des TéléTTPs dans le domaine de la programmation en informatique, un environnement original exploitant le principe des situations d'apprentissage actif, et soumis à une évaluation appariée basée sur un algorithme d'appariement des graphes pour évaluer le savoir-faire des apprenants.

L'approche d'évaluation proposée est une approche socioconstructiviste assurant une évaluation trilatérale : évaluation sommative, (quantifier les compétences et les connaissances réellement possédées), formative (aider l'apprenant à progresser) et diagnostique (diagnostiquer les erreurs et les lacunes).

Notre contribution au traitement de cette problématique intègre les quatre aspects suivants :

Contribution 1 : une approche d'évaluation appariée

Notre approche permet d'évaluer les travaux pratiques des apprenants sur le développement de programmes C++ orientés objets. L'idée est de procéder à une comparaison indirecte de deux programmes orientés objets (apprenant, enseignant) à travers leurs représentations graphiques décrites par deux diagrammes UML (diagramme de classes et diagramme de communication). Notre approche a pour objectif de trouver les similarités et les différences entre la production de l'apprenant et celle de l'enseignant. La technique d'appariement que nous proposons tient compte de plusieurs paramètres (voir chapitre 5 section 5).

Contribution 2 : un algorithme d'appariement

L'algorithme proposé est destiné pour calculer la similarité entre la production de l'apprenant et celle de l'enseignant. Il accepte comme données d'entrée: Deux

Conclusion générale et perspectives

appariements m et m' (m pour les diagrammes de classes UML et m' correspondant pour diagrammes de communication), deux diagrammes de classes UML (apprenant, enseignant) et deux diagrammes de communication UML (apprenant, enseignant). Notre algorithme utilise deux types d'apparieurs: apparieur hybride et apparieur composite en tenant compte de certains critères de comparaison dans le calcul de similarité. En sortie, l'algorithme donne le résultat le calcul de similarité entre les deux programmes (apprenant et enseignant).

Contribution 3 : un environnement d'évaluation des productions des apprenants en programmation orientée objet

L'environnement développé permet essentiellement: (1) la génération de diagrammes UML à partir du programme de l'apprenant, et (2) l'application des algorithmes d'appariement pour mesurer les degrés de similarité et de différence entre les diagrammes de l'apprenant et les diagrammes de l'enseignant. L'environnement d'évaluation développé est composé de trois modules principaux: l'outil éditeur, l'outil générateur, et l'outil apparieur. La composante la plus importante est l'outil apparieur qui est implémenté en utilisant l'algorithme d'appariement développé qui évalue la solution de l'apprenant en termes de valeurs de similarité et offre un feedback aux apprenants pour les aider à corriger leurs erreurs commises.

Contribution 4 : l'intégration de l'outil d'évaluation dans la plateforme E-TéléTPC@AALP

L'intégration de l'outil d'évaluation dans notre plateforme nous a permis de prouver les effets de cet outil d'évaluation, d'une part, pour la maîtrise des concepts du langage de programmation orienté objet C++ et de la résolution des situations de blocage par les apprenants. D'autre part, nous a permis de justifier les effets de cet outil pour diminuer le nombre de demandes d'intervention en ligne des enseignants assistants des travaux pratiques dans une session de TP.

Un des problèmes rencontrés est que la modélisation d'une solution dans notre outil n'est pas assez riche. Plus précisément, en utilisant les graphes pour modéliser les solutions cette modélisation nous n'a permis que la comparaison syntaxique entre les programmes, mais la comparaison sémantique n'est pas encore prise en considération.

Perspectives

Nous sommes conscients que notre travail est loin d'être complet et qu'il pourra évoluer dans un futur plus ou moins proche. Il est loin d'être achevé et nécessite encore beaucoup d'efforts et de persévérance. Différentes orientations pratiques et différentes perspectives de recherches découlent des travaux de cette thèse.

A. Perspectives expérimentales

- Transformer notre plateforme d'une plateforme maison (utilisée par une institution) en une plateforme libre dont le code source est accessible à tous les utilisateurs.

B. Perspectives de recherche

- L'utilisation des graphes pour la modélisation des solutions, dans le cadre de cette thèse, s'est limitée au contexte d'utilisation de l'activité d'évaluation des connaissances des apprenants. Il nous n'a permis que la comparaison syntaxique des programmes. Différentes autres utilisations des outils de modélisation sont possibles en particulier avec les ontologies pour assurer aussi la comparaison sémantique des programmes. En ce qui concerne le processus d'évaluation, en modifiant la représentation des solutions à comparer par les ontologies.
- Il faudra changer la mesure de similarité proposée au départ. Puisque cette dernière a été conçue par rapport à un formalisme bien précis (les graphes).
- Améliorer le processus d'évaluation de l'apprenant en appliquant les techniques de tests fonctionnels sur sa production.
- Etendre l'approche proposée en tenant compte d'autres types de relations entre classes supportées par UML.
- Comme il a été montré, quoique cette approche ait été conçue pour l'évaluation des productions en programmation, elle peut, très facilement, être adaptée à n'importe quel domaine manipulant des connaissances procédurales. Nous envisageons de poursuivre notre travail vers une méthode générique d'évaluation du savoir-faire.

Bibliographie

[Abbaci, 2013] Abbaci, K. (2013), « Contribution à l'interrogation flexible et personnalisée d'objets complexes modélisés par des graphes », thèse de doctorat en informatique, l'université de rennes 1, 2013, France.

[Adafer,2005] Adafer.F. (2005), « Etat de l'art sur l'évaluation de l'apprenant dans un hypermédia adaptatif », Rapport interne, projet de recherche EVAL.

[Alonso & Py, 2009] Alonso, M. and Py, D.(2009), « Une évaluation des rétroactions dans DIAGRAM, un EIAH pour la modélisation orientée objet », dans the 4ème conférence EIAH'09, Le Mans, France.

[Alonso et al., 2008] Alonso, M., Auxepaules, L., Lemeunier, T., and Py, D.(2008), «Design of Pedagogical Feedbacks in a Learning Environment for Object-Oriented Modeling», in: Promoting Software Modeling through Active Education, dans Proceedings of Educators Symposium of the ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS'08), Michal Smialek (Eds.), Toulouse, France ,PP. 39-50.

[Antoniol et al., 2010 a] Antoniol, G., Galinier, P., and Kpodjedo, S.(2010), «Enhancing a tabu algorithm for approximate graph matching with a similarity measure», dans the EvoCOP'10, Eur. Conf. on Evolutionary Computation in Combinatorial Optimisation, Istanbul, Turkey.

[Antoniol et al., 2010 b] Antoniol, G., Galinier, P., and Kpodjedo, S.(2010 b), «On the use of local similarity measures for approximate graph matching»,Journal of Electronic Notes in Discrete Mathematics, 36,.Vol 36 (Elsevier),pp. 687–694.

[Antoniol et al., 2011] Antoniol, G., Galinier, P., and Kpodjedo, S.(2011), « Design evolution metrics for defect prediction in object oriented systems». An International Journal Empirical Software Engineering, Vol 16 Issue 1, February 2011,pp 141-175.

[Auxepaules et al., 2007] Auxepaules, L., Dominique, P., and Thierry, L.(2007), « Une méthode de diagnostic reposant sur l'appariement structurel de diagrammes de classes dans un EIAH pour la modélisation orientée objet. », dans the Environnements Informatiques pour l'Apprentissage Humain, Lausanne ,France,2007.

[Auxepaules ,2009] Auxepaules, L. (2009), « Analyse des diagrammes de l'apprenant dans un EIAH de la modélisation orientée objet », Thèse de doctorat, Université du Maine, France, 2009.

[Benabbou, 2006] Benabbou.F, Hanoune. M. (2006), « easyalgo :un environnement d'apprentissage et d'autoévaluation de l'algorithmique », [en ligne] disponible sur : http://admiroutes.asso.fr/larevue/2006/76/Article_Apprentissage.pdf consulté [le 28 janvier 2016].

Bibliographie

- [Benaily,2000] Benaily.M.(2000), « Formations Ouvertes et à Distance l'accompagnement pédagogique et organisationnel »,Conférence de Consensus 27, 28 & 29 mars 2000, France.
- [Benmohamed a, 2006] Lelevé. A, Prévot. P, Benmohamed. H.(2006), «distance closing: closing the gap between remote labs and learning management system. rapport», 2006, Hammamet, Tunisia.
- [Benmohamed b, 2006] Lelevé. A, Prévot. P, Benmohamed. H.(2006), « télé-tps: évolution d'une plateforme générique »,Rapports, 2006, Toulouse, France.
- [Bouarab-Dahmani, 2010] Bouarab-Dahmani.F.(2010), « Modélisation basée ontologies pour l'apprentissage interactif - Application à l'évaluation des connaissances de l'apprenant », thèse de doctorat en informatique , université UMMTO, Algérie.
- [Benmohamed, 2007] Benmohamed. H.(2007), « ICCT@LAB: un environnement informatique pour la génération et l'exécution de scénarios de télé TP », Thèse de doctorat, Université de Lyon, France, 2007.
- [Bengoetxea, 2002] Bengoetxea.E.(2002), «Inexact Graph Matching Using Estimation of Distribution Algorithms»,Thèse de Doctoral Spécialité Signal et Images, Telecom Paris ,France.
- [Berry,2014] Berry.R.(2014), «Assessment for Learning: Questioning and Feedback to understand and Enhance Learning», The International Conference of EDUCA 2014 The 7th Annual Congress for Teacher Professional Development, Bangkok (15 17 Oct 2014).
- [Blackwell, 2002] Blackwell. A.(2002), «What is Programming? » .PPIG, Brunel University, London, UK, 2002.
- [Bloom,1956] Bloom.B.(1956), «Taxonmy of educational objectives: The classification of educational goals»,Handbook I, cognitive domain. Longman, New York.
- [Bonnici et al., 2013] Bonnici.V, Giugno.R, Pulvirenti.A, Shasha.D,and Ferro .A(2013), «A subgraph isomorphism algorithm and its application to biochemical data. BMC Bioinformatics»,2013 14(Suppl 7):S13 DOI: 10.1186/1471-2105-14-S7-S13.
- [Bennouna et al., 2008] Bennouna. M., Delestre.N., Pécuchet. J.-P., and Tanana. M(2008), «Évaluation du savoir-faire en électronique numérique à l'aide d'un algorithme de classification», Dans the conférence Technologies de l'Information et de la Communication pour l'Éducation (TICE'08), Paris, France, 2008.
- [Bennouna et al., 2009] Bennouna.M., Delestre.N., Pécuchet. J.-P., and Tanana. M(2009), «Génération d'exemples pour l'évaluation de l'apprenant en électronique numérique à l'aide d'un algorithme de classification», dans the Conférence EIAH 2009, le mans, France ,2009.
- [Bouabid, 2012] Bouabid. M.L. (2012), «de la conception à l'exploitation des travaux pratiques en ligne : application d'une approche générique à l'enseignement de l'informatique», thèse de doctorat en informatique, Université de Toulouse, France,2012.

Bibliographie

[Bouacha,2015] Bouacha.I.(2015), «Un EIAH pour l’algorithmique», Thèse de doctorat en informatique, Université Badji Mokhtar ; Annaba, Algérie, 2015.

[Boussaha & Bensebaa, 2009] Boussaha .K., and Bensebaa ,T.(2009), «Design of an environment of Remote practice works between realities and prospect», Dans the international conference of novel digital technology. 31 July 2009, republic check.

[Boussaha , 2009] : Boussaha. K.(2009), «conception d’un environnement de TéléTPs». Mémoire de magister en informatique, Annaba, Algérie ,2009.

[Boussaha, 2011] Boussaha.K.(2011), « Modélisation d'une situation d'évaluation de l'apprenant avec UML: CAS d'application pour l'apprentissage des langages de programmation » .dans the 8eme colloque sur l’optimisation et les systèmes d’information, COSI’2011, université 08 Mai 45 Guelma, les 24-27 avril 2011, Algérie.

[Boussaha et al., 2012] Boussaha.K., Mokhati.F., and Taleb.N.(2012), «a novel learner self-assessment approach – application to practical work» , dans 4 th international conference on computer supported Education, the CSEDU 2012 ,Porto ,Portugal.

[Boussaha et al.,2015] Boussaha. K., Mokhati.F., and Chaoua,Z.(2015), «Architecture of a specific platform for training practical works: integration of learners assessment component», dans : International Journal of Technology Enhanced Learning(IJTEL) , Vol. 7, No. 3, 2015.

[Buche, 2005] Buche.C.(2005), « Un système tutoriel intelligent et adaptatif pour l’apprentissage de compétences en environnement virtuel de formation », Thèse de doctorat, Université de Bretagne Occidentale, pages 246.

[Bunke, 2000] Bunke.H.(2000) «Graph matching: Theoretical foundations, algorithms, and applications», Dans Proc. Vision Interface , Montreal, pp. 82-88.

[Camille, 2014] Camille, C. (2014), « Mesures de similarité pour cartes généralisées », thèse de doctorat en informatique, université U.C.B.L, France, 2014.

[Chilowicz, 2010]Chilowicz,M.(2010), « Recherche de similarité dans du code source »,thèse de doctorat , université Paris –Est,France,2010 .

[Claroline ,2007]Manuel d'utilisation de Claroline Claroline 1.8 - Septembre 2007 , Version Administrateur , [en Ligne], Disponible sur, http://www.joseouin.fr/allpdf/claroline_manuel_administrateur.pdf.

[Curé, 2005] Curé.O.(2005), « La programmation orientée objet ». cours [en ligne] disponible sur : http://perso.univ-mlv.fr/ocure/imm/immJava0809_cm2.pdf [consulté le 09 février 2016], université de Marne la vallée, France.

[David,2003]David J.P.(2000), « Modélisation et production d’objets pédagogiques ». *Sciences et Techniques Educatives*, Hors série 2003, Ressources numériques, XML et éducation, Hermès, avril 2003.

Bibliographie

[Delorme et al. ,2005] Delorme,F., Delestre,N., Pecuchet,J.(2005), « Évaluer l'apprenant à l'aide de cartes conceptuelles », dans Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie, Oct 2005,Université de Technologie de Compiègne,France. pp.25-31, 2004, ISBN : 2-913923-12-7.

[Delozanne, 2002] Delozanne.E , Grugeon .B .(2002), « projet pépite et lingo des logiciels pour faciliter la régulation par les enseignants des apprentissage en algèbre » [en ligne] disponible sur : <http://pepите.univ-lemans.fr/Telechargement/francais/19-Cahier%20Ed&Dev%2004.pdf> [consulté le 20 aout 2015].

[Demeuse, 2002], Demeuse.M.(2002), « Edumétrie et psychométrie: deux disciplines différentes ? », dans le Bulletin de l'Association pour le Développement des Méthodologies d'Evaluation en Education (ADMEE Europe) (2002/2).

[Després, 2001] Després,C. (2001), « Modélisation et Conception d'un Environnement de Suivi Pédagogique Synchrones d'Activités d'Apprentissage à Distance », Thèse de doctorat, Université du Maine, Le Mans, France.

[Djoulak Kengue *et al.* 2008] Djoulak Kengue J. F., Euzenat J., Valtchev P.(2008), « Alignement d'ontologies dirigé par la structure », dans Actes de la 2ème Conférence Francophone sur les Architectures Logicielles (CAL 2008), Revue des Nouvelles Technologies de l'Information RNTI-L-2, Cépaduès (Eds.), 155, mars 2008 Montréal, Québec, Canada.

[Do & Rahm, 2007] : Do H.-H., Rahm E.(2007), «Matching Large Schemas: Approaches and Evaluation», dans the Journal on Information Systems, vol. 32, n°6, 857-885, septembre 2007.

[Dossou et al., 2008] Dossou, A-K., DOGBE, S., Anne, D. (2008), « Etude comparative de plates-formes de formation à distance », dans le cadre du Projet @2L.

[Dranidis, 2007] Dranidis, D. (2007), «Evaluation of Student UML: an Educational Tool for Consistent Modelling with UML », dans Proceedings of the Informatics Education Europe II (IEEII), Thessaloniki, Grèce, pp.248-256.

[Duchâteau a, 1992] Duchâteau, C. (1992), «From "DOING IT ..." to " HAVING IT DONE BY ...": The Heart of Programming. Some Didactical Thoughts», NATO Advanced Research Workshop "Cognitive Models and Intelligent Environments for Learning Programming", S Margherita, Italy.

[Duchâteau b,1992]. Duchâteau.C.(1992) « De "FAIRE" à "FAIRE FAIRE" :au coeur de la programmation : quelques réflexions didactiques », Conférence présentée au NATO Advanced Research Workshop "Cognitive Models and Intelligent Environments for Learning Programming" S;Margherita, Italie, 17-21 mars 1992.

[Durand, 2006] Durand, G. (2006). « Vers une scénarisation de l'évaluation en EIAH », dans les proceedings de rjc_eiah 2006.

Bibliographie

[Duris,2003] Duris.E.(2003), « Les bases de la programmation orientée objet avec Java », Cours [en ligne] disponible sur : <http://igm.univ-mlv.fr/~duris/JAVA/IR1/JavaLesBases.pdf> [consulté le 31 janvier 2016].

[EdD.S.I, 2005],EdD.S.I, Mowat.J.(2005), «Learning Content Management System (LCMS) », [en ligne] disponible sur : http://www.herridgegroup.com/pdfs/LCMS_chptr_Aug29%202005_%20Final%20Sept%2006%2005.pdf [consulté le 29 janvier 2016].

[Euzenat & Shvaiko, 2007] Euzenat J., Shvaiko P.(2007), «Ontology Matching», Springer-Verlag, Berlin Heidelberg, 334 pages, 2007.

[FOAD, 2002] : glossaire de la formation ouverte et à distance , Le GIP ARIFOR, Centre de Ressources régional sur la formation professionnelle, région de Champagne-Ardenne 2002. [en Ligne], Disponible sur : <http://www.tribunes.com/tribune/alliage/29-30/perr.htm>, [consulté le 10 novembre 2015].

[Frayssinhes, 2011] Frayssinhes.J,(2011), « Les pratiques d'apprentissage des adultes en FOAD : effet des styles et de l'auto-apprentissage », thèse de doctorat , université de Toulouse, France.

[Gaillard, 2011] Gaillard.A.L(2011), « identification de motifs au sein des structures biologiques arborescentes », thèse de doctorat en informatique, université bordeaux i, France.

[Gateau et al., 2007] Gateau.G, Fontes.G, Roux.N, Bonnet.F, Defay.F(2007), « Les nouvelles technologies au service de l'aide à la préparation des travaux pratiques », conférence CETSIS 2007.

[George, 2001] George S. (2001). « Apprentissage collectif à distance. SPLACH : un environnement informatique support d'une pédagogie de projet » , Thèse d'informatique, Université du Maine.

[Ghédira et al., 2006] Ghédira,K .,Olfa ,S.,Solnon,S.,and Sorlin,S.(2006) , « comparaison de l'optimisation par colonies de fourmis et d'une recherche réactive sur des problèmes d'appariement de graphes », dans the conference JFPC , Nîmes Alés , France.

[Guibert, 2006] Guibert, N.(2006), « Validation d'une approche basée sur l'exemple pour l'initiation à la programmation », thèse de doctorat en informatique, université de Poitiers, France, 2006.

[Durand, 2006] : Guillaume, D.(2006) « La scénarisation de l'évaluation des activités pédagogiques utilisant les Environnements Informatiques d'Apprentissage Humain », thèse de doctorat en informatique, université de Savoie, France. 2006.

[Guittet et al., 2005] Guittet, L., Girard. P, Guibert.N.(2005), « a study of the efficiency of an alternative programming paradigm to teach the basics of programming». LISI / ENSMA, 86961 Futuroscope Chasseneuil cedex,France.

[Hadji,1990] Hadji, C. (1990), « Evaluation, les règles du jeu », ESF.

Bibliographie

- [Hadji, 1997] Hadji, C. (1977), « L'évaluation démystifiée », EME Editions Sociales Francaises (ESF), , Paris, 1997
- [Hague,2001] Hague. R.(2001), «Designing a programming language for home automation», In G. Kadoda (Ed.) Proceedings of the 13th Annual Workshop of the Psychology of Programming Interest Group (PIIG 2001), 85-103.
- [Hanna,2004]Hanna, G. S., & Dettmer, P. A.(2004), «formative and summative assessment», Northern Illinois University, Faculty Development and Instructional Design
- [Hartree, 1950]: Hartree, D. R. (1950), «Calculating instruments and machines», Cambridge University Press.
- [Hinnoutondji, 2011] Hinnoutondji,K .(2011), «Approximate graph matching for software engineering», Thèse de doctorat, université de Montréal, Canada.
- [Hoc,1982] Hoc. J.M. (1982), « L'étude psychologique de l'activité de programmation » : une revue de la question - Technique et Science Informatiques 1 (1982), p 383-392
- [JEAN, 2000] JEAN,S.(2000), « PEPITE : un système d'assistance au diagnostic de compétences », Thèse de doctorat en informatique, Université du Maine,France, 2000.
- [Jolion et al., 2007] Jolion J.-M.,Solnon C.,and Sorlin S.(2007), «A Generic Graph Distance Measure Based on Multivalent Matchings», dans the Applied Graph Theory in Computer Vision and Pattern Recognition book , Vol. 52, Series "Studies in Computational Intelligence", Springer, pp 151-182 2007.
- [Jorro ,2000] Jorro.A.(2000), « L'enseignant et l'évaluation : des gestes évaluatifs en question. »- De Boeck université , 2000.- 184 p..-Pratiques pédagogiques .
- [Jouili,2008] Jouili.S, Tabbone.S.(2008), « .Applications des graphes en traitement d'images », international Conference on Relations, Orders and Graphs: Interaction with Computer Science - ROGICS'08, May 2008, Mahdia, Tunisie. pp.434-442, 2008.
- [Jouili & Tabbone, 2009] Jouili, S., and Tabbone, S. (2009). «Graph matching based on node signatures», dans Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition,pp.154 – 163,Nancy France.
- [Kerschenbaum, 2002] Kerschenbaum.S: LMS(2002), « Selection Best Practices » [en Ligne], Disponible sur :https://cdns3.trainingindustry.com/media/2068137/lmsselection_full.pdf [consulté le 10 novembre 2015].
- [Kurilovas, 2000]. Kurilovas.E.(2000), « Methods of Multiple Criteria Evaluation of the Quality of Learning Management Systems for Personalised Learners Needs», Institute of Mathematics and Informatics, Vilnius, Lithuania , Disponible sur : http://wwwis.win.tue.nl/lms-ale-09/Kurilovas_paper.pdf [Consulté le 26/06/2015].
- [Lafifi, 2007] Lafifi Y.(2007), « SACA : un système d'apprentissage collaboratif », Thèse de doctorat en informatique, Université d'Annaba, Algérie.

Bibliographie

[Lelevé a, 2003] Lelevé A, Benmohamed H, Prevot P, Meyer C(2003), « Remote Laboratory Towards an integrated training system », Quatrième conférence internationale sur l'éducation et la formation basées sur les technologies de l'information (ITHET03)(Information Technology Based Higher Education and Training) Marrakech, Maroc du 7 au 9 juillet 2003 .

[Lelevé b, 2003] Lelevé A, Prevot P, Subai C, Noterman D, Guillemot M.(2003), « Toward remote laboratory platforms with dynamic scenarios » , multi-conférence en systémique,cybernétique et informatique (SCI2003), Juillet 2003 ,Orlando, Floride, USA.

[Lelevé a, 2004] Lelevé A, Benmohamed H, Prevot P, Benadi M (2004), «Generic e-lab platforms and e-learning standards », actes de la conférence (CALIE 04)(Computer Aided Learning In Engineering Education) Grenoble, France du 16 au 18 Février 2004.

[Lelevé b, 2004] Lelevé A, Meyer C, Prevot P(2004), « Télé-TP: premiers pas vers une modélisation », Actes du Symposium on Technology of Information and Communication in education for engineering and industry, Lyon, p. 203-211.

[Lelevé *et al.*, 2004] Lelevé, A., Benmohamed, H. , Prévôt, P(2004), «Sharing a System between Simultaneous Learners in Remote Laboratories», 2nd IFAC Workshop on Internet Based Control Education (IBCE), 5-7 Septembre 2004, Grenoble, France.

[Lelevé1, 2005] Lelevé A, Benmohamed H, Prevot P(2005), «Implémentation d'une chaîne d'édition générique pour télé-TPs »,actes de la conférence sur la robotique pédagogique.

[Lelevé *et al.*, 2005] Lelevé, A., Benmohamed, H. , Prévôt, P.(2005), « Implémentation d'une chaîne d'édition générique pour téléTPs » , 8ème colloque francophone de robotique pédagogique, 2-4 Mai 2005, Le Mans, France.

[Linacre, 2000] Linacre.J.M(2000), «Computer-Adaptive Testing:A Methodology Whose Time Has Come», Ph.D.MESA Psychometric Laboratory, University of Chicago,2000

[Liskov,1974] Liskov.B.H, Zilles.S.N(1974),. «Programming with abstract data types». – SIGPLAN Notices 9 (1974), p 50-59

[Lombard,2013] Lombard.A. (2013), « TéléTP : intégration de systèmes didactiques dans une plateforme d'apprentissage en ligne », thèse de doctorat, université de Toulouse ,2013.

[Marcotte, 2004] Marcotte. A(2004), « Les apports de l'expérimentation assistée par ordinateur (ExAO) en pédagogie par projet en Sciences de la nature au collégial », thèse de doctorat, Université de Montréal, Canada.

Bibliographie

[Mazza,2005] Mazza.R and Milani.C,(2005), «Exploring usage analysis in learning systems : Gaining insights from visualisations»,. In *AIED Workshops (AIED'05)*, juillet 2005.

[Mbala-Hikolo,2003], Mbala-Hikolo.A.(2003), « Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance »,thèse de doctorat , université de Franche-Comté,France,2003.

[Mechta, 2012] Mechta, D. (2012), « Plate-forme pour les travaux pratiques à distance sur le Web », thèse de doctorat en informatique, université de Ferhat Abbas de Sétif, 2012, Algérie.

[Meijer, 2008] Meijer M.M. (2008), « On a method for XML Schema matching Using edit distance and schema trees for structural matching of elements», 8th Twente Student Conference on IT, Enschede, January 25th, 2008.

[Merceron,2004] Merceron.A, Yacef.K. (2004), «Train, store, analyse for more adaptive teaching», in *Proceedings of International Symposium Information and Knowledge Technologies in Higher Education and Industry'*, TICE'04, Technical University of Compiegne, Compiegne, France, pp. 52–59.

[Meyer et al., 2002] Meyer, C., Lelevé, A.,Prévot ,P.(2002), « télé-TPs: premier pas vers une modélisation », rapport, 2002, Lyon, France.

[Millar,2009] Millar.R,(2009), «Analysing practical activities to assess and improve effectiveness: The Practical Activity Analysis Inventory (PAAI) » , Published by the Centre for Innovation and Research in Science Education, Department of Educational Studies, University of York, Heslington,

[Mitrovic & Suraweera,2004] Mitrovic, A, and Suraweera, P. (2004), «An Intelligent Tutoring System for Entity Relationship Modelling », *International Journal of Artificial Intelligence in Education* 14 (3–4),pp. 375–417.

[Müller,2011] Müller.D.(2011), « Introduction à la théorie des graphes », *CAHIERS DE LA CRM*,2011.

[Monbet,2009] Monbet. V.(2009), «Tests statistiques Notes de cours». [Cours en ligne] disponible sur : https://perso.univ-rennes1.fr/valerie.monbet/doc/cours/Cours_Tests_2009.pdf
[consulté le 01 mars 2016].

[Neau ,2003] Neau,N. (2003), « Argograph : un support au débat scientifique dans le cadre de travaux pratiques pour l'apprentissage des sciences expérimentales »,thèse de doctorat en informatique, le Mans, université de Maine, 2003,287p.

[Norman ,1990] Norman, D. A. (1990),« The design of every day things », NewYork NY, USA, Doubleday Currency.

Bibliographie

[Nottelmann, 2006]Nottelmann.H, Straccia.U(2006), «Information retrieval and machine learning for probabilistic schema matching», *information Processing and Management* 43 (2007) 552–576.

[Ouamer ,1992] Ouamer Ahmed ,R. (1992), « un atelier de génie didacticiel pour les applications industrielles AGEDI », thèse de doctorat en informatique ,INSA de lyon ,1992,299p.

[Oubahssi, 2014] Oubahss.L, Nour EL MAWAS(2014),« étude comparative de plates-formes de formation à distance » , [en Ligne], Disponible sur : <http://www-ilium.univ-lemans.fr/~laforcad/graphit/wp-content/uploads/2014/08/GRAPHIT-D2.52.2-V6.pdf>, [consulté le 19 juillet 2015].

[Ouerfelli,2004]Ouerfelli; T.(2004), «Stratégie pédagogique et formation à distance des professionnels : éléments de réflexion à travers le projet de mise en place d'une formation à distance des documentalistes- bibliothécaires en Tunisie », Colloque TICE Méditerranéenne-NICE. 26 et 27 novembre 2004.

[Paetau,2005] paetau.p.(2005), «on the benefits and problems of the object-oriented paradigm including a finnish study», Department of Management and Organization (Information Systems Science) Vaasa, Finland

[Pair, 1988] Pair.C. (1988), « Je ne sais (toujours) pas enseigner la programmation » - *Informatiques 2 revue pour l'enseignement de l'informatique* ; n° 2. - pp. 5-14.- ISSN : 0987-7312.

[Pair,1989] Pair.C.(1989), « l'apprentissage de la programmation », Centre de Recherche en Informatique de Nancy,1989, France.

[Panagiotakopoulos et al.,2003] Panagiotakopoulos .C, Lionarakis.A, Xenos.M.(2003), « open and distance learning: tools of information and communication technologies for effective learning», *Proceedings of the sixth Hellenic European Research on Computer Mathematics and its Applications Conference,HERCMA2003,Athens,Greece,2003.*

[Peretti ,2009]Peretti.A(2009), «Encyclopédie de l'évaluation en formation et en éducation Guide pratique», 4ème édition. Issy-les-Moulineaux : ESF éditeur, Lyon, France

[Perrault,1997]Perrault, J. « Formation à distance et culture scientifique et technique, *Alliage*, n°29-30 », [en Ligne], 1996, Disponible sur : <http://www.tribunes.com/tribune/alliage/29-30/perr.htm>, [consulté le 18 janvier 2016].

[Prévit et al ,2003] Prévit.D , Delozanne.E , Grugeon.B.(2003), « Modélisation cognitive en algèbre élémentaire une conception itérative », [en ligne],disponible sur : <http://pepite.univ-lemans.fr/Telechargement/francais/15-DP-ED-BG-TICE04.pdf>[consulté le 28 janvier 2016]

[Quinton, 2005], Quinton.A,(2005), « Docimologie, D.U. de Pédagogie » – Mai 2005, [en Ligne], Disponible sur <http://www.crame.u-bordeaux2.fr/pdf/docimologie.pdf>.

Bibliographie

- [Rahm & Bernstein 2001] Rahm E., Bernstein P. A.(2001), «A survey of approaches to automatic schema matching», dans: The international Very Large Data Bases Journal (VLDB Journal), vol. 10, n°4, Springer Berlin / Heidelberg, 334-350, décembre 2001.
- [Ramond,2010] Ramond.S.(2010), « de la formation à l'autoformation des enseignants d'économie- gestion : pour une analyse des boucles de reproduction pédagogique », Actes du congrès de l'Actualité de la recherche en éducation et en formation (AREF), Université de Genève, septembre 2010.
- [Rellier,2003] Rellier.C (2003), « les histoires d'évariste », [en Ligne], Disponible sur, http://www.epi.asso.fr/fic_pdf/b60p115.pdf
- [Riesen et al.,2007] Riesen.K, Neuhaus.M , Bunke.H(2007), « Bipartite graph matching for computing the edit distance of graphs. Graph-Based Representations in Pattern Recognition», pages 1 – 12, 2007.
- [Ross, 2006]. Ross. J. A .(2006), « The Reliability, Validity, and Utility of Self-Assessment», *Practical Assessment, Research & Evaluation journal* , Volume 11 Number 10, November 2006.
- [Sammoud,2006]Sammoud.O , Solnon.C , Ghedira.K(2006), « Ant Colony Optimization for Multivalent Graph Matching Problems», ECAI Workshop on Evolutionary Computation,2006.
- [Simmon-Hariss,2002] Simmons-Harris.D,(2002),«School Self-evaluation in a Democracy», School -Based Evaluation in D. Nevo (ed.), School-based Evaluation: An International Perspective, JAI Press, Oxford, pp. 17-34.
- [Singer et al. ,2005] Singer, Susan R. Margaret L. Hilton, Heidi A, (2005). «America's Lab Report: Investigations in High School Science» Schweingruber, Editors, Committee on High School Science Laboratories: Role and Vision, National Research Council ISBN: 0-309-65286-3, 254 pages, 7 x 10, (2005).
- [Sonja A. Irlbeck.E, 2005] Sonja A. Irlbeck, EdD, Joanne Mowat,(2005), «Learning Content Management System (LCMS) »,rapport, Capella University , [en Ligne], Disponible sur : http://www.herridgroup.com/pdfs/LCMS_chptr_Aug29%202005_%20Final%20Sept%2006%2005.pdf [consulté le 01 septembre 2015].
- [Solon, 2005] : Solnon, C.(2005), « Contributions à la résolution pratique de problèmes combinatoires : des fourmis et des graphes », Habilitation à Diriger des Recherches, Laboratoire d'Informatique en Image et Systèmes d'information (LIRIS), Université Claude Bernard Lyon 1 , France, 115 pages, décembre 2005.
- [Solon & Fenet, 2006] Solnon,C., and Fenet, S.(2006), «A study of aco capabilities for solving the maximum clique problem», dans Journal of Heuristics- Springer, 2006.
- [Sorlin,2003] Sorlin.S, Solnon.C.(2003), « Similarité de graphes : une mesure générique et un algorithme tabou réactif », IC 2003,France

Bibliographie

- [Sorlin et al.,2003] Sorlin,S., Champin,P,A. and Solnon,C. (2003), « Mesurer la similarité de graphes étiquetés », dans les actes *JNPC'03*,
- [Sorlin & Solnon,2005] Sorlin, S., and Solnon ,C.(2005), « Une contrainte globale pour le problème de l'isomorphisme de graphes », Laboratoire d'Informatique en Image et Systèmes d'information (LIRIS), Université Claude Bernard Lyon 1.
- [Sorlin.2005]Sorlin.S and Solnon.C.(2005), «Reactive tabu search for measuring graph similarity», 5th IAPR Workshop on Graph-based Representations in Pattern Recognition (GbR 2005),LNCS 3434 Springer :172–182, 2005.
- [Sorlin 2006] Sorlin S.(2006), « Mesurer la similarité de graphes », Thèse du Laboratoire d'Informatique en Image et Systèmes d'information (LIRIS), Université Claude Bernard Lyon 1, France, 142 pages, novembre 2006.
- [Sorlin et al a.,2006]Sorlin.S, Sammoud.O , Solnon.C, Ghédira.K.(2006), « de l'optimisation par colonies de fourmis et d'une recherche réactive sur des problèmes d'appariement de graphes », Actes JFPC 2006.
- [Sorlin et al b. 2006] Sorlin S., Sammound O., Solnon C., Jolion J. M.(2006),« Mesurer la Similarité de Graphes », dans Extraction de connaissance à partir d'Images (ECOI 2006), Atelier de Extraction et Gestion de Connaissances (EGC 2006), 21 -30, 2006.
- [Sorlin et al. 2007] Sorlin S., Solnon C., Jolion J.-M.(2007), « A Generic Graph Distance Measure Based on Multivalent Matchings»,dans Applied Graph Theory in Computer Vision and Pattern Recognition, 151 –182, 2007.
- [Souchard,2003] Souchard.L.(2003), « Analyse des ressources de logiciels tutoriels fermées dans l'enseignement pré-algébrique ». Lagrange, Jun 2003, Reims, France.[en ligne] disponible sur : <https://hal.archives-ouvertes.fr/edutice-00001363/document> [consulté le 20 aout 2015].
- [Tanana, 2009] Tanana,M., (2009), «Evaluation formative du savoir-faire des apprenants à l'aide d'algorithmes de classification : application à l'électronique numérique», thèse de doctorat en informatique, université de INSA de Rouen,France,2009.
- [Taziri,2005]Taziri, A.B., & Allouche, B. (2005), « *Les acteurs de l'enseignement à distance.* », 3rd International Conference: Sciences of electronic, technologies of information and telecommunications, Tunisie.
- [Tchounikine 2002] Tchounikine P.(2002), « Quelques éléments sur la conception et l'ingénierie des EIAH », dans Actes du GDR I3 (Information-Interaction-Intelligence), 233-245, 2002.
- [Techounikine et al., 2004] Techounikine P, Baker M, Balacheff N, Baron M, Derycke A, Guin D,Nigaud J-F, Rabardel P (2004), « platon 1: quelques dimensions pour l'analyse des travaux de recherche en conception d'EIAH », rapport d'action spécifique du CNRS,2004,19p.
- [Techounikine, 2009] Tchounikine, P. (2009), « Précis de recherche en ingénierie des eiah », From [http://membresliglab. imag.fr/tchounikine/Precis.html](http://membresliglab.imag.fr/tchounikine/Precis.html).

Bibliographie

[Thi Hong Hiep,2009] Thi Hong Hiep.N.(2009), « Appariement multivoque de graphes par la recherche locale »,Mémoire de fin d'études, Université catholique de Louvain

TPLINE. Les travaux pratiques de sciences industrielles sur des systèmes en fonctionnement en temps réel. Disponible sur : <http://www.tpline.net/index.php> (consulté le 26/06/2015).

[Ward,2004] Ward.A , Sitthiworachart.J, and Joy.M.(2004), «Aspects of web-based peer assessment systems for teaching and learning computer programming», In *Proceedings of the IASTED International Conference*, 16-18 février 2004.

[Wilkes, 1956] Wilkes, M. V. (1956), «Automatic digital computers», Cambridge University Press.

[Wrubel ,1959] Wrubel, M. H. (1959), «A primer of programming for digital computers», Mc Graw Hill.

[Zarrouk, 2012]Zarrouk .F.(2012), « les statistiques inferentielles (test de student) » ,[cours en ligne] disponible sur : http://www.issep-ks.rnu.tn/fileadmin/templates/Fcad/Test__de_Student.pdf consulté le [01 mars 2016].

[Zaslavskiy, 2010] Zaslavskiy.M.(2010), « Graph matching and its application in computer vision and bioinformatics», thèse de doctorat en Bioinformatique,, l'Ecole nationale supérieure des mines de Paris ,France.

Annexe A

Questionnaire

Dans le cadre d'un travail de recherche lié à la conception et l'extension des services de notre plateforme de travaux pratiques à distance (E-TéléTPC@AALP) sur les langages de programmation et spécifiquement l'intégration de module d'évaluation dans cette plateforme, nous avons besoin de recueillir certaines informations. Les étudiants des deux groupes participant à l'expérimentation sont invités à renseigner le questionnaire ci-dessous.

Auriez-vous l'amabilité de préciser dans ce questionnaire ce que vous faites dans votre TP sur les langages de programmation en informatique ?

Nous vous en remercions.

I. TP à distance (TéléTP)

QI.1. Que pensez-vous de faire un TP à distance (TéléTP) à l'aide de la plateforme E-TéléTPC@AALP?

- Intéressant
- Peu intéressant
- Pas intéressant

QI.2. Pour vous, quel est l'apport des TéléTPs sur les langages de programmation par rapport aux TP en présentiel ?

- Offrir plus de possibilité de collaboration
- Des TP à jour en tout moment
- Diminuer le problème d'encombrement dans les groupes, et offrir aux apprenants l'occasion de faire leurs TP à l'aise
- Une communication fiable entre les apprenants
- Une standardisation des méthodes de travail

Annexe A

- Autre, préciser
-
-

QI.3. Dans un environnement pour l'apprentissage des TPs à distance sur les langages de programmation quels sont les outils que vous pensez retrouver ?

A- Pour la programmation

.....

.....

.....

B- Pour la documentation

.....

.....

.....

C - Pour l'évaluation

.....

.....

.....

D- Pour la communication avec les co-TP et avec les tuteurs

.....

.....

.....

QI.4. Pour réaliser un TélecTPs avec la plateforme E-TélecTPC@AALP il y a 4 tâches essentielles. Pour chaque tâche, déterminez le degré de difficulté ?

A. Edition de code

- Facile
- Peu difficile
- Difficile
- Très difficile

B. Compilation

- Facile
- Peu difficile
- Difficile

Annexe A

Très difficile

C. Correction des erreurs

Facile

Peu délicate

Difficile

Très difficile

D. Exécution

Facile

Peu difficile

Difficile

Très difficile

QI.5. Dans les tâches difficiles, comment surmontez-vous les difficultés ?

consulter l'aide en ligne

Collaborer entre vous

Autre, préciser

QI.6. Quel mode de communication préférez-vous dans un environnement de TéléTP?

Mode synchrone

Chat

Messagerie instantanée

Autre :

Mode asynchrone

Forum

Email

Autre :

II. Evaluation

QII.1. Dans un environnement des travaux pratiques à distance quel mode d'évaluation préférez-vous ?

Manuel (compte rendu)

QCM Automatique

Annexe A

- Outil d'auto-évaluation
- Autre, précisez
-
-

QII.2. Pour les différentes tâches à effectuer au cours de votre TéléTP, déterminez si la tâche doit se faire individuellement ou à l'aide d'un outil assistant?

A. Edition de code

- Individuellement
- Outil assistant

B. Compilation

- Individuellement
- Outil assistant

C. Correction des erreurs

- Individuellement
- Outil assistant

D. Exécution

- Individuellement
- Outil assistant

QII.3. Selon vous, quel est le rôle d'un module d'auto-évaluation dans un Télé TP ?

- Aider à surmonter les difficultés de programmation
- Assurer le déroulement de la séance dans de meilleures conditions
- Aider de progresser l'apprentissage
- Autre, précisez

QII.4. comment trouvez-vous l'intérêt d'utilisation d'un module d'évaluation dans la plateforme ?

- Catalyseur pour l'apprentissage
- Frustrant pour l'apprentissage
- Autres

QII.5. Vos commentaires Merci beaucoup!!!!!!!!!!!!

Représentations graphiques des Réponses au questionnaire

I. TP à distance (TéléTP)

Q1.1.

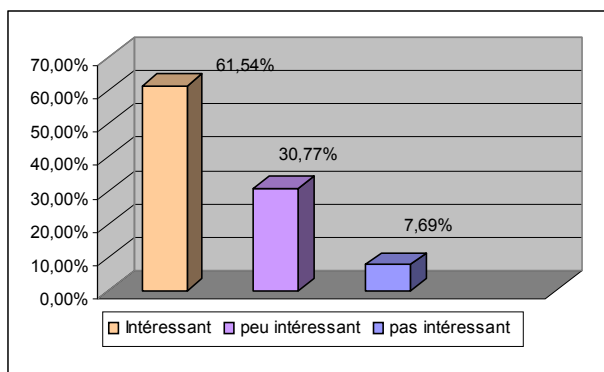


Figure A.1 : *L'intérêt de faire un TP à distance*

Q1.4.

e. Edition de code

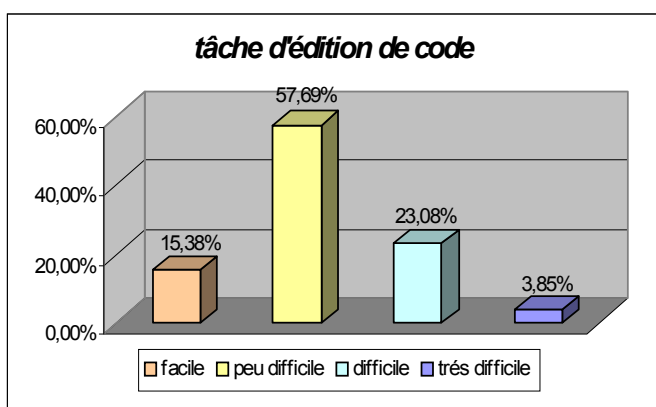


Figure A.2 : *La tâche d'édition de code*

f. Compilation

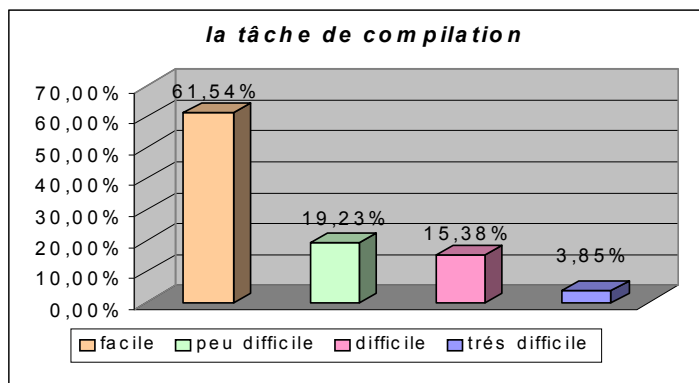


Figure A.3 : *La tâche de compilation*

Annexe A

g. Correction des erreurs

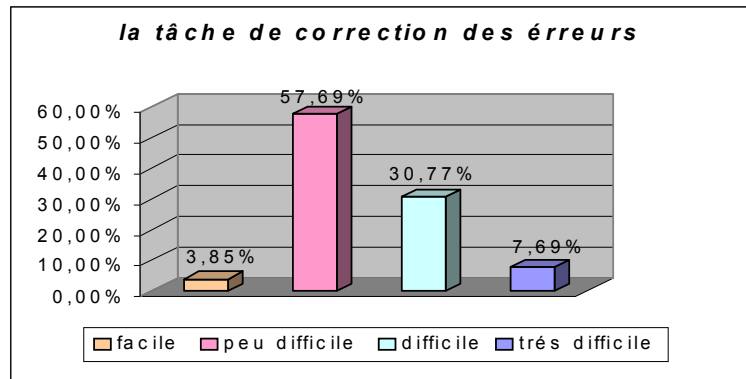


Figure A.4 : La tâche de correction des erreurs

h. Exécution

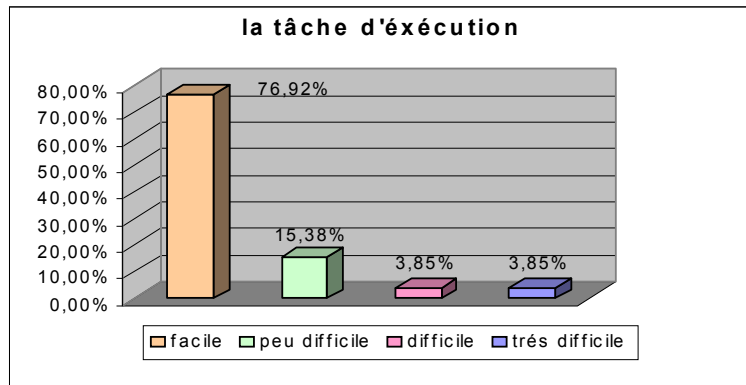


Figure A.5: La tâche d'exécution

Q1.5

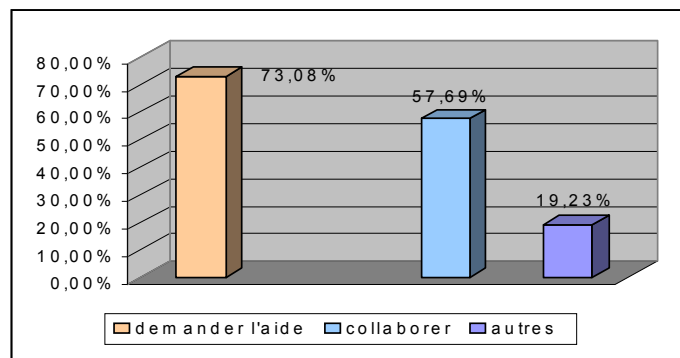


Figure : A.6: Les réactions des apprenants devant les tâches difficiles

QI.6.

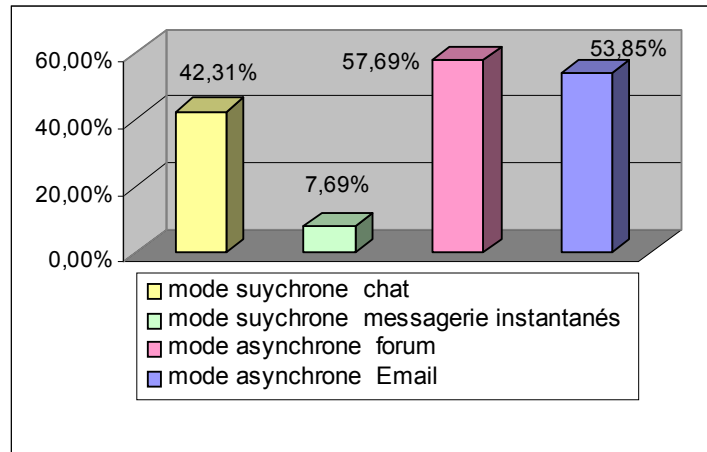


Figure A.7: Les modes de communication préférés dans un TélÉTP

II. Evaluation

QII.1.

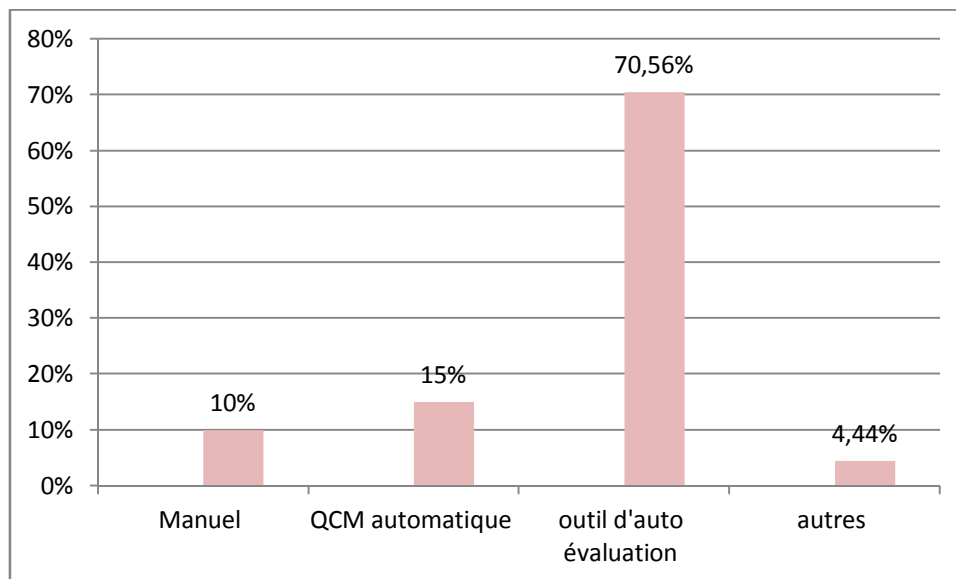


Figure A.8 : L'évaluation préférée chez les apprenants

QII.2.

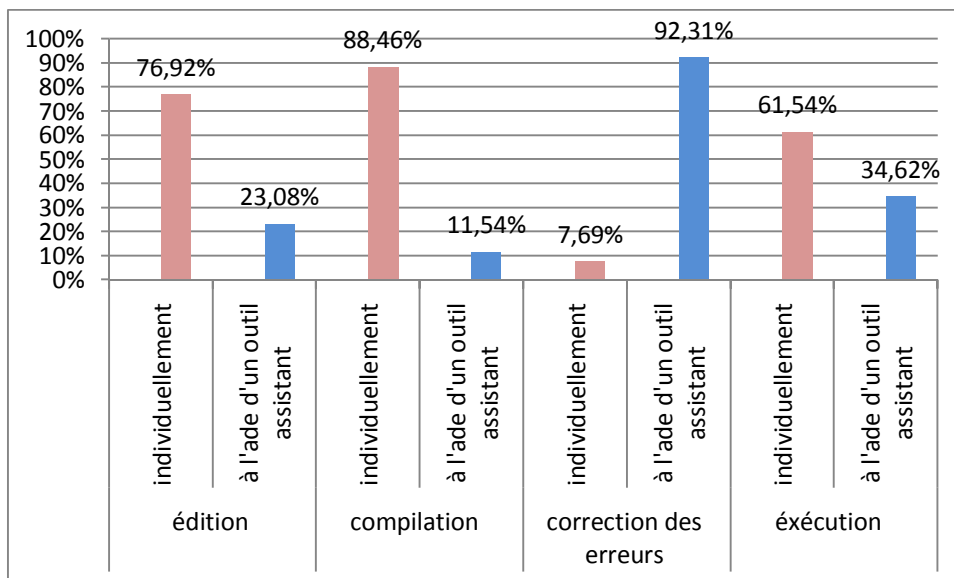


Figure A.9 : Les préférences des apprenants pour la réalisation des quatre tâches de la programmation

QII.4.

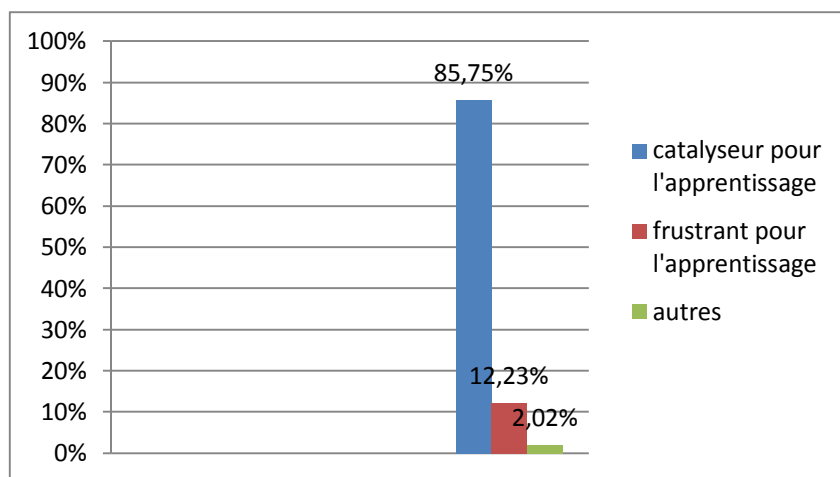


Figure A.10 : L'utilité d'utilisation d'un module d'évaluation dans la plateforme

Annexe B

Le test de Student

1. Définition

Un test statistique est un mécanisme qui permet de trancher entre deux hypothèses au vu des résultats d'un échantillon [Monbet, 2009]. La notion de test est incluse dans le domaine des statistiques d'inférence qui, contrairement à la statistique descriptive, ne se contente pas de décrire des observations, mais extrapole les constatations faites à un ensemble plus vaste et permet de tester des hypothèses sur cet ensemble ainsi que de prendre des décisions [Zarrouk, 2012].

Soient H_0 et H_1 deux hypothèses (H_0 est appelée hypothèse nulle, H_1 hypothèse alternative) [Zarrouk, 2012], dont une et une seule est vraie. La décision consiste à retenir H_0 ou H_1 .

Il existe plusieurs tests tels que le test Z, Test de Mann-Whitney-Wilcoxon, test de Fisher, test de Student.

Dans notre travail on a utilisé le test de Student . Ce test permet de comparer [Monbet, 2009] :

- une moyenne d'un échantillon à une valeur donnée
- les moyennes de deux échantillons indépendants
- les moyennes de deux échantillons appariés.

Donc on utilise ce test pour comparer deux moyennes de deux échantillons indépendants (deux groupes des apprenants avec deux tailles n_1 et n_2).

Question : le module d'évaluation ajouté fait-il améliorer significativement l'apprentissage des TéléTPs chez les étudiants débutants en langage de programmation avec la plateforme E-TéléTPC@AALP ?

Les hypothèses :

H0 : le module d'évaluation ajouté est inefficace

H1 : le module ajouté fait améliorer l'apprentissage chez les apprenants

Réponse : comparer deux échantillons à travers la différences entre leurs moyennes

2. Les formules de calcul de test de Student

On calcule la valeur t qui suit une variable aléatoire de Student aux degrés de liberté (df) tel que :

$$df = n_1 + n_2 - 2.$$

2.1. Calculer les moyennes et les écarts types de chacun des groupes

$$\text{Moyenne} = \Sigma Xi / n$$

$$\sigma^2 = \sqrt{\frac{\Sigma(xi - \text{moyenne})^2}{n-1}}$$

2.2. Calculer la variance totale

$$S^2 = (n_1 * \sigma_1^2) + (n_2 * \sigma_2^2) / df$$

2.3. Calcul de t de student

$$t = \left| (\text{moyenne1} - \text{moyenne2}) \right| s \sqrt{\left(\frac{1}{n_1} + \frac{1}{n_2} \right)}$$

3. Analyse de la valeur de test de student

Comparer le t calculé au t de la table, avec un degré de liberté $n = N_1 + N_2 - 2$. Si t calculé est supérieur au t de la table il existe une différence significative (c'est H1 qui est retenue). Si t calculé est inférieur au t de la table, il n'existe pas de différence significative (c'est H0 qui est retenue) [Monbet, 2009].

Ou bien on extrait la valeur de probabilité qui correspond au t calculé. Si la valeur de probabilité est supérieure à 0.05 il n'existe pas de différence significative (c'est H0 qui est retenue). Si la valeur de probabilité est inférieure à 0.05 il existe une différence significative (c'est H1 qui est retenue) [Zarrouk, 2012].

Remarque : il existe des sites¹¹ qui calculent ces valeurs d'une manière automatique. Les résultats de notre expérience sont calculés par les deux méthodes (Manuelle avec des formules et automatiques à l'aide des sites).

¹¹<http://www.sthda.com/french/rsthda/unpaired-t-test.php>

Chapitre 01

EAO : enseignement assisté par ordinateur.

EIAO : enseignement intelligemment assisté par ordinateur.

EIAH : environnement informatique pour l'apprentissage Humain.

FOAD : formation ouverte et à distance.

FAD : formation à distance.

TICE : Technologies de l'information et de la communication.

LMS: learning management system.

LCMS: learning content management system.

QCM : Question à Choix Multiples.

SCORM: Sharable Content Object Reference Model.

LOM : Learning Object Metadata.

TéLÉTPs : travaux pratiques à distance.

UML: Unified Modeling Language.

ELaMS: Electronic Laboratory Management System.

OntoServ : Serveur d'Ontologies.

IHM : interface homme machine.

VRML: Virtual Reality Markup Language.

E-TéLÉTPC@AALP: environnement de travaux pratiques collaboratifs à distance: application à l'apprentissage des langages de programmation.

Chapitre 02

CAT: Computer Adaptive Test.

GISMO: Graphical Interactive Student Monitoring System.

TEB : Travaux pratiques d'Électronique Binaire.

Chapitre 03

mcs : maximum common subgraph (le plus grand sous graphe commun)

GED : Graph Edit Distance (la distance d'édition de graphe)

Notations

RTS : Reactive Tabu Search. (Recherche locale Taboue réactive)

ACO : Ant Colony Optimization (optimisation par colonie de fourmis)

Chapitre 04

HTML : Hypertext Markup Language

MI : math informatique.

POO : programmation orientée objet.

OO : orienté objet.

Chapitre 05

m : l'appariement généré à partir de diagramme de classe de l'enseignant.

m' : l'appariement généré à partir de diagramme de communication de l'enseignant.

Sim_m : similarité entre les deux diagrammes de classe (apprenant, enseignant) par rapport à l'appariement m.

Sim_c : similarité entre les deux diagrammes de classe (apprenant, enseignant) par rapport aux contraintes.

Sim_{m'} : similarité entre les deux diagrammes de communication (apprenant, enseignant) par rapport à l'appariement m'.

Total_Sim : la similarité totale entre les deux diagrammes de classes (apprenant, enseignant).

Global_Sim : la similarité globale entre les deux programmes.

W_m : le poids attribué à la similarité entre diagrammes de classes de l'enseignant et de l'apprenant qui est basé sur l'appariement m.

W_c : les poids attribué à la similarité entre les diagrammes de classes de l'enseignant et de l'apprenant qui est basé sur la liste des contraintes c.

W_{cd} : le poids attribué à des aspects statiques du programme C++ décrit par les diagrammes de classes.

W_{cmd} : le poids attribué à des aspects dynamiques du programme C++ décrit par les diagrammes de communication.

G_{1A} ∩_m G_{1T} : le nombre de caractéristiques communes entre les diagrammes de classe (apprenant, enseignant) par rapport à l'appariement m.

RV_{1A} ∪ RV_{1T} ∪ RE_{1A} ∪ RE_{1T} : le nombre de toutes les caractéristiques pour les diagrammes de classe (apprenant, enseignant) par rapport à l'appariement m.

Notations

$G2A \cap m' G2T$: le nombre de caractéristiques communes entre les diagrammes de communication (apprenant, enseignant) par rapport à l'appariement m' .

$RV2A \cup RV2T \cup RE2A \cup RE2T$: le nombre de toutes les caractéristiques pour les diagrammes de communication (apprenant, enseignant) par rapport à l'appariement m' .

Travaux réalisés au cours de cette thèse

I. Publications Internationales dans des revues : (1).

1. **BOUSSAHA Karima** ,Mokhati Farid,Chaoua Zakaria, «Architecture of a specific platform for training practical works: integration of learners assessment component», à apparaitre dans : *International Journal of Technology Enhanced Learning(IJTEL)* , Vol. 7, No.3 , 2015.

II. Communications Internationales avec ISBN: (1).

1. **BOUSSAHA Karima** ,Mokhati Farid,Taleb Nora, «a novel learner self-assessment approach – application to practical works » ,4 international conference on computer supported Education ,CSEDU 2012 ,ISBN 978-989-8565-06-8,Porto ,Portugal ,april 16-18,2012

III. Communications Internationales avec comité de lecture: (2).

1. **BOUSSAHA Karima**, « Modélisation d'une situation d'évaluation de l'apprenant avec UML: CAS d'application pour l'apprentissage des langages de programmation », 8eme colloque sur l'optimisation et les systèmes d'information, COSI'2011 , université 08 Mai 45 Guelma, les 24-27 avril 2011, Algérie.
2. **BOUSSAHA Karima** ,Mokhati Farid,Chaoua Zakaria , «Using matching approach to assess learners in their Practical works activity with a specific CEHL», The 2nd International Conference on Multimedia Information processing ,CITIM'2015 Mustapha Stambouli University -Faculty of Science and Technology, Mascara May 12-13, 2015,Algérie.

IV. Formations & stages

1. Stage de courte durée ,1 mois, dans le laboratoire LIRIS, équipe « Silex » université Claude Bernard Lyon 1 France novembre-décembre 2012.
2. Formation, « défis et solutions du Télé-Enseignement et des e-Test en Milieu Universitaire », 1ère Ecole de printemps sur le télé-enseignement, université Saad Dahlab ,27-30 avril, 2013, Blida.
3. Formation: « Améliorer sa pédagogie, réussir son cours, en techno-pédagogie Universitaire », Université d'Oum El Bouaghi, 15-17 mars 2015.
4. Formation, « Latex, en techno-pédagogie Universitaire », Agence universitaire de la Francophonie(AUF) en partenariat avec l'université d'Oum El Bouaghi, 3-5 mai 2015,Oum El Bouaghi