

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
L'université de Batna 2 - (Mostefa Ben Boulaïd)
Faculté des Mathématiques et Informatique

Thèse

En vue de l'obtention du diplôme de
Doctorat en Sciences en Informatique

Développement d'un environnement pédagogique de support de la programmation coopérative des apprenants

Présentée Par
Atef CHORFI

Soutenue le 19 Mai 2021

Devant le jury composé de :

<i>Président</i>	Rachid SEGHIR	<i>Prof., Université de Batna 2</i>
<i>Rapporteur</i>	Djalal HEDJAZI	<i>MCA., Université de Batna 2</i>
<i>Examineurs</i>	Mohammed BENMOHAMMED	<i>Prof., Université de Constantine 2</i>
	Hichem HOUASSI	<i>MCA., Université de Khenchela</i>
	Rafik MAHDAOUI	<i>MCA., Université de Khenchela</i>
<i>Invité</i>	Sofiane AOUAG	<i>MCA., Université de Batna 2</i>

Remerciement

Le modeste travail de recherche décrit dans cette thèse est le résultat de plusieurs années de travail et de patience, et grâce à l'aide précieuse de certaines personnes. Particulièrement, Docteur *Djalal HEDJAZI* Maître de Conférences-A à l'Université de Batna 2, à qui revient le très grand mérite d'avoir accepté de poursuivre la direction de ce travail et de me permettre de proposer les contributions adéquates pour atteindre les objectifs de la thèse. Il m'a offert d'excellentes idées et visions sur le travail de la thèse tout au long de son évolution et n'a en aucun cas préservé aucun effort pour me conseiller en me permettant de concrétiser mes idées librement. Qu'il trouve ici l'expression de mes plus sincères remerciements.

Son soutien permanent et son expérience m'ont été d'un grand apport et ont conforté mon enthousiasme dans cette recherche, c'est au Professeur *Abdelmadjid ZIDANI* que revient aussi le mérite d'avoir initié ce travail, auxquelles j'exprime mes plus franches remerciements.

Je témoigne également toute ma gratitude au Professeur *Rachid SEGHIR*, Professeur à l'université de Batna 2, pour l'honneur qu'il me fait de présider mon jury de thèse.

Je tiens à exprimer toute ma reconnaissance au Professeur *Mohammed BENMOHAMMED*, Professeur à l'Université de Constantine 2, d'avoir bien accepté de faire partie de mon jury de thèse.

Je témoigne toute ma sympathie au Docteur *Hichem HOUASSI*, Maître de Conférences-A à l'Université de Khenchela, d'avoir accepté de faire partie de mon jury de thèse.

Je témoigne toute ma sympathie au Docteur *Rafik MAHDAOUI*, Maître de Conférences-A à l'Université de Khenchela, d'avoir accepté de faire partie de mon jury de thèse.

Je remercie sincèrement le Docteur *Sofiane AOUAG*, Maître de Conférences-A à l'Université de Batna 2, d'avoir accepté l'invitation pour participer à mon jury de thèse.

Un grand merci à toute ma famille, en particulier, ma femme et mes parents, qui ont toujours été présents pour me soutenir. Ce travail leur est dédié.

Enfin, mes remerciements vont également à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Résumé

Le contexte de *l'apprentissage collaboratif de la programmation* a évolué très rapidement ces dernières décennies, notamment avec l'avancement considérable qu'a connu le domaine des technologies de l'Information (en anglais *IT : Information Technology*). Ces technologies, qui offrent des perspectives très intéressantes, permettent la création non seulement de moyens de communication, mais aussi des supports pour les interactions de groupe. En effet, plusieurs acteurs (apprenants, enseignants et tuteurs) peuvent utiliser ces supports afin d'interagir, indépendamment de leurs localisations géographiques, au sein d'une session d'apprentissage de la programmation informatique.

L'apprentissage par problème (en anglais *PBL : Problem-Based Learning*) représente l'une des méthodes pédagogiques les plus efficaces que nous pouvons appliquer pour compléter les méthodes classiques. Dans une telle méthode, les apprenants, organisés en groupes, collaborent ensemble autour de la résolution d'un problème de la programmation informatique généralement soulevé par un enseignant. Le problème pour lequel ils n'ont suivi aucune formation particulière sera étudié pour effectuer des apprentissages de contenus et à améliorer des compétences de résolution de problèmes.

Le travail de recherche, décrit dans de cette thèse, s'inscrit dans une direction à travers laquelle les sessions d'apprentissage classiques de la programmation informatique peuvent être enrichies par des supports de l'apprentissage collaboratif assisté par ordinateur (en anglais *CSCL : Computer Supported Collaborative Learning*), tout en appliquant la méthode pédagogique d'apprentissage par problème. Par conséquent, un nouvel environnement pédagogique baptisé *PBPCLG (Problem-Based Programming' Collaborative Learning Groupware)* est proposé. *PBPCLG* offre aussi bien un espace de travail partagé pour l'édition de code source qu'un ensemble d'outils collaboratifs synchrones/asynchrones avec lesquels les apprenants peuvent interagir pour développer un programme résolvant un problème de programmation donné.

Cette thèse vise un double objectif, d'une part, concevoir et développer un environnement collaboratif pour le support du téléapprentissage de la programmation et, d'autre part, mener une étude expérimentale pour mesurer le degré d'acceptabilité par ses utilisateurs effectifs d'un tel environnement.

Mots-clés : *Apprentissage à distance, Apprentissage par problèmes, Travail coopératif assisté par ordinateur, Interaction synchrone/asynchrone, Notification d'évènements, Architecture MVC, Édition partagée de code source.*

Abstract

The context of programming collaborative learning has rapidly evolved these recent decades, especially with the considerable advancement marked in the field of Information Technology (IT). Technologies that offer very interesting perspectives allowing the creation not only the means for communication, but also supports for group interactions. Thus, several actors (learners and tutors) can use these supports in order to interact, regardless of their geographic locations, within programming learning sessions.

Problem-Based Learning (PBL) is one of the effective learning methods that we can apply to complete traditional methods. In such a method, learners, organized in groups, collaborate around solving a programming problem typically posed by a teacher. The problem, for which they did not take any special training, will be study to perform content learning and improve problem-solving skills.

The research work, described in this thesis, goes in a direction through which the classic learning sessions of computer programming can be completed by new approaches based on problem-based learning using computer-assisted collaborative learning supports (CSCL: *Computer Supported Collaborative Learning*). Therefore, a new learning environment called PBPCLG (*Problem-Based Programming 'Collaborative Learning Groupware*) is proposed. It offers both a shared workspace for source code editing and a set of synchronous/asynchronous collaborative tools through which learners can interact with each other to develop a program solving a programming issue raised by a tutor. This thesis has double objectives, on the one hand, to design and develop a collaborative environment to support programming learning and, on the other hand, to conduct an experimental study to measure the degree of acceptability of such environment, notably by its real users.

Keywords: *E-learning, Problem-based learning, Computer assisted cooperative work, Synchronous/Asynchronous interaction, Event notification, MVC Pattern, Client/Server design, Collaborative editing of source code.*

ملخص

التقدم الكبير والسريع في مجال التعلم التعاوني للبرمجة خلال العقود الاخيرة، خاصة مع التقدم الكبير الملحوظ في مجال تكنولوجيا المعلومات . ومع ظهور تقنيات جديدة التي تقدم وجهات نظر مثيرة للاهيام حيث تسمح بإنشاء ليس فقط وسائل الاتصال، ولكن أيضا تدعم التفاعلات الجماعية. وبالتالي، يمكن للعديد من الجهات الفاعله (المتعلمين والمعلمات) استخدام هذه الدعام للتعامل، بغض النظر عن مواقعهم الجغرافية، في جلسات تعلم البرمجة. يعد التعلم القائم على حل المشكلات أحد أساليب التعلم الفعالي التي يمكننا تطبيقها والتي من خلالها يتعاون المتعلمون، المنظمون في مجموعات، لإيجاد حلول لمشكلات البرمجة التي يطرحها عادة المعلم. هذه المشكلات التي لم يتلقوا أي تدريب خاص بشأنها، تكون موضوع دراسة من خلالها يتم تعلم بعض المفاهيم الخاصة بالبرمجة الى جانب تحسين مهارات حل المشكلات .

يسير العمل البحثي، الموصوف في هذه الأطروحة، في اتجاه يمكن من خلاله تدعم جلسات التعلم الكلاسيكية لبرمجة الكمبيوتر بمناهج جديدة تعتمد على التعلم القائم على حل المشكلات باستخدام دعم التعلم التعاوني بمساعدة الكمبيوتر. لذلك، يتم اقتراح برنامج تعليمي جديد يسمى PBPCLG. يوفر هذا الأخير مساحة عمل مشتركة لتحرر التعلّات البرمجية إلى جانب مجموعة من الأدوات التعاونية الميرامنة/غير الميرامنة التي يمكن للمتعلمين من خلالها التفاعل مع بعضهم البعض لتطوير برنامج لحل مشكله مطروحة من طرف المعلم. ومن هنا فإن هذه الأطروحة لها هدف مزدوج، من ناحية، يهدف إلى تصميم وتطوير برنامج تعاوني لدعم تعلم البرمجة عن بعد بين المتعلمين، ومن ناحية أخرى، لإجراء دراسة تجريبية لقياس درجة قبول هذا البرنامج، لا سيما من قبل مستخدمي الحاسوبين.

الكلمات الرئيسية : التعلم الإلكتروني، التعلم القائم على المسائل، العمل التعاوني بمساعدة الحاسوب، التفاعل الميرامن/غير الميرامن، الكتابة التعاونية أو التشاركية لشفرة المصدر.

Table des Matières

Remerciement.....	I
Résumé	II
Abstract.....	III
ملخص	IV
Table des Matières	V
Liste des figures	VII
Liste des tables	VIII
Liste des abréviations et des sigles.....	IX
Introduction générale	1
1. Introduction	1
2. Contexte de travail	2
3. Problématique de recherche	3
4. Objectifs et méthodologie de recherche	3
5. Organisation de la thèse	5
Chapitre 1 : Aspects théoriques de l'apprentissage.....	6
1.1 Introduction.....	6
1.2 Théories de l'apprentissage	7
1.2.1 Béhaviorisme	7
1.2.2 Constructivisme et Socioconstructivisme	7
1.2.3 Interactionnisme	8
1.3 L'apprentissage selon la pédagogie ouverte	8
1.4 L'apprentissage en groupe-classe.....	9
1.4.1 Interaction.....	10
1.4.2 Participation	11
1.4.3 Communication	11
1.4.4 Prise de décision	11
1.5 Apprentissage situé	12
1.6 Apprentissage selon la théorie de l'activité.....	13
1.7 Tendances éducatives.....	15
1.7.1 Pédagogie classique	15
1.7.2 Pédagogie active	15
1.7.2.1 Apprentissage par découverte.....	16
1.7.2.2 Apprentissage collectif.....	17
1.7.2.3 Apprentissage par projet	20
1.7.2.4 Apprentissage par problème	22
1.7.2.4.1 Avantages de l'APP	24
1.8 Conclusion	24
Chapitre 2 : Apprentissage collaboratif assisté par ordinateur.....	25
2.1. Introduction.....	25
2.2. Enseignement assisté par ordinateur	26
2.3. Enseignement Intelligemment Assisté par Ordinateur	27
2.4. Environnements Interactifs d'Apprentissage Assisté par Ordinateur	28
2.5. Environnements Interactifs pour l'Apprentissage à Distance	28
2.6. Formation à Distance.....	29
2.6.1. Plateformes de FAD	30
2.7. Apprentissage Collaboratif Assisté par Ordinateur.....	31
2.7.1. TCAO	32

2.7.2.	ACAO.....	33
2.8.	Conclusion	36
Chapitre 3 : Systèmes Collaboratifs dédiés à l'apprentissage de la programmation		37
3.1.	Introduction.....	37
3.2.	Programmation Informatique	38
3.2.1.	Spécificités de la programmation informatique	39
3.2.1.1.	<i>Spécificités techniques.....</i>	<i>41</i>
3.2.2.	Apprentissage de la programmation informatique	42
3.2.2.1.	<i>Logiciels d'apprentissage de la programmation</i>	<i>42</i>
3.2.2.2.	<i>Quelques environnements similaires</i>	<i>44</i>
3.3.	Conclusion	47
Chapitre 4 : Présentation de l'environnement PBPCLG.....		49
4.1.	Introduction.....	49
4.2.	Aspects sociaux et conscience de groupe	49
4.2.1.	Interaction sociale.....	50
4.2.1.1.	<i>Modes d'interaction</i>	<i>51</i>
4.2.1.2.	<i>Attribution des rôles</i>	<i>52</i>
4.2.1.3.	<i>Diagramme de classe décoré interactif</i>	<i>53</i>
4.2.2.	Mécanismes de la conscience de groupe	54
4.3.	Scénario	56
4.4.	Implémentation de PBPCLG	60
4.4.1.	Architecture logicielle	60
4.4.2.	Choix technologiques d'implémentation	62
4.5.	Conclusion	62
Chapitre 5 : Facteurs d'acceptation et d'utilisation du PBPCLG : Application du modèle UTAUT		64
5.1.	Introduction.....	64
5.2.	Théories créatrices des modèles de l'intention	64
5.2.1.	Modèles d'intention d'utilisation des technologies	66
5.2.1.1.	<i>Modèle TAM.....</i>	<i>67</i>
5.2.1.2.	<i>Modèle des 3P</i>	<i>67</i>
5.2.1.3.	<i>Modèle UTAUT.....</i>	<i>68</i>
5.3.	Notre étude expérimentale	69
5.3.1.	Développement des hypothèses.....	70
5.3.1.1.	<i>Performance Espérée (PE).....</i>	<i>70</i>
5.3.1.2.	<i>Effort Espéré (EE)</i>	<i>71</i>
5.3.1.3.	<i>Influence Sociale (SI)</i>	<i>72</i>
5.3.1.4.	<i>Conditions Facilitantes (FC).....</i>	<i>72</i>
5.3.2.	Méthodologie	73
5.3.2.1.	<i>Collecte des données.....</i>	<i>73</i>
5.3.2.2.	<i>Échantillonnage et procédure expérimentale</i>	<i>75</i>
5.3.2.3.	<i>Analyse des résultats</i>	<i>76</i>
5.3.2.3.1.	<i>Modèle de mesure.....</i>	<i>77</i>
5.3.2.3.2.	<i>Modèle structurel.....</i>	<i>79</i>
5.3.3.	Discussion.....	81
5.4.	Conclusion	83
Conclusion générale et perspectives.....		84
Références.....		86
Annexe 1. Notre Production Scientifique		100

Liste des figures

FIGURE 1. 1 : FACTEURS INFLUENÇANT LA DYNAMIQUE MOTIVATIONNELLE DE L'APPRENANT.	10
FIGURE 1. 2 : SYSTEME D'ACTIVITE D'ENGESTRÖM [ENGESTRÖM 01].	14
FIGURE 1. 3 : DEUX SYSTEMES D'ACTIVITE [ENGESTRÖM 01].	14
FIGURE 1. 4 : APPRENTISSAGE PAR DECOUVERTE.	17
FIGURE 1. 5 : COLLABORATION VS COOPERATION.	18
FIGURE 1. 6 : DIFFERENCE ENTRE APPRENTISSAGE COLLABORATIF/COOPERATIF [LAFIFI 07].	19
FIGURE 1. 7 : CRITERES FAVORISANT L'APPRENTISSAGE PAR PROJET [LEBRUN 07].	21
FIGURE 1. 8 : PROCESSUS DE REALISATION D'UN PROJET [THIEN 14].	22
FIGURE 2. 1 : SCHEMA GENERAL D'UNE PLATEFORME DE LA FAD.	31
FIGURE 2. 2 : TETRAEDRE PEDAGOGIQUE DE FAERBER (2002).	34
FIGURE 2. 3 : SCHEMA DE FONCTIONNEMENT DU PROCESSUS PEDAGOGIQUE D'ACAO [EL HASSAN ET LAZREK 07].	35
FIGURE 3. 1 : PROCESSUS DE RESOLUTION DE PROBLEMES DE LA PROGRAMMATION : ENTITES INTERVENANTES [ROGALSKI 88].	39
FIGURE 3. 2 : PROCESSUS DE DEVELOPPEMENT D'UN PROGRAMME	40
FIGURE 3. 3 : TRADUCTION DU CODE SOURCE EN LANGAGE MACHINE.	41
FIGURE 4. 1 : DIAGRAMME DE CLASSE DECORE [CHORFI ET AL. 20].	54
FIGURE 4. 2 : POSITION DU CURSEUR D'UN APPRENANT « REDACTEUR ».	56
FIGURE 4. 3 : AUTHENTIFICATION D'UN APPRENANT VIA PSEUDO ET MOT DE PASSE.	57
FIGURE 4. 4 : INTERFACE PRINCIPALE DU PBPCLG.	57
FIGURE 4. 5 : DESCRIPTION TEXTUELLE D'UN PROBLEME DE PROGRAMMATION.	58
FIGURE 4. 6 : UNE DISCUSSION ENTRE BOUBICHE ET CHORFI.	59
FIGURE 4. 7 : FENETRE D'ENGAGEMENT DE NOTIFICATION.	60
FIGURE 4. 8 : ARCHITECTURE LOGICIELLE DU PBPCLG.	61
FIGURE 5. 1 : SCHEMA DE LA TAR [DAVIS ET AL. 89].	65
FIGURE 5. 2 : SCHEMA DE LA TCP [AJZEN 91].	66
FIGURE 5. 3 : SCHEMA DU MODELE TAM [DAVIS 89].	67
FIGURE 5. 4 : SCHEMA DU MODELE DES 3P [DILLON ET MORRIS 99].	68
FIGURE 5. 5 : SCHEMA DU MODELE UTAUT [VENKATESH ET AL. 03].	69
FIGURE 5. 6 : MODELE DE RECHERCHE UTAUT MODIFIE.	70
FIGURE 5. 7 : RESULTATS DE L'ANALYSE DE LA MODELISATION STRUCTURELLE.	81

Liste des tables

TABLEAU 1.1: TABLEAU COMPARATIF ENTRE L'APPRENTISSAGE COOPERATIF ET COLLABORATIF.	20
TABLEAU 5. 1 : QUESTIONNAIRE QUANTITATIVE ADOPTE PAR NOTRE ETUDE.	74
TABLEAU 5. 2 : APERÇU DES CONSTRUCTIONS DE RECHERCHE UTILISEES DANS LE QUESTIONNAIRE.	75
TABLEAU 5. 3 : CHARGES FACTORIELLES DES ITEMS.	77
TABLEAU 5. 4 : VALIDITE CONVERGENTE ET LA FIABILITE DES MESURES.	78
TABLEAU 5. 5 : VALIDITE DISCRIMINANTE DES MESURES.	78
TABLEAU 5. 6 : <i>INDICES D'AJUSTEMENT</i>	79
TABLEAU 5. 7 : <i>RELATION ENTRE LES VARIABLES EXOGENES ET ENDOGENES</i>	80

Liste des abréviations et des sigles

3P <i>Pouvoir- Performance-Perceptions</i>	
AC	Apprentissage Collaboratif
ACAO	Apprentissage Collaboratif Assisté par Ordinateur
AGFI	Aadjusted GFI
APP	Apprentissage Par Problème
CBT	Computer Based Training
CFI	Bentler comparative fit index
CG	Conscience de Groupe
CL	Collaborative Learning
CSSL	Computer Supported Collaborative Learning
CSCW	Computed Supported Collaborative Work
DCD	Diagramme de Classe Décorée
EAD	Enseignement à Distance
EAO	Enseignement Assisté par Ordinateur
EDI	Environnement de Développement Intégré
EIAD	Environnements Interactifs pour l'Apprentissage à Distance
EIAH	Environnement Informatique pour l'Apprentissage Humain
EIAO	Enseignement Intelligemment Assisté par Ordinateur
FAD	Formation à Distance
GFI	Goodness of Fit Index
GUI	Graphique User Interface
IA	Intelligence Artificielle
ICAI	Intelligent Computer Aided Instruction
IHM	Interface Homme Machine
ILE	Intelligent Learning Environment
IT	Information Technology
ITS	Intelligent Tutoring System
PBL	Problem Based Learning
PBPCLG	Problem Based Programming Collaborative Learning Groupware
PIFAD	Plateforme Informatique pour la Formation À Distance
PP	Programmation en Paires (Pair Programming)
RMSEA	Root Mean Square Error of Approximation
STI	Système Tuteur Intelligent
TAM	Technology Acceptation Model
TAR	Théorie de l'Action Raisonnée
TCAO	Travail Coopérative Assisté par Ordinateur
TCI	Théorie des Comportements Interpersonnels
TCP	Théorie du Comportement Planifié
TIC	Technologies de l'Information et de la Communication
UTAUT	Unified Theory of Acceptance and Use of Technology
WYSIWIS	What You See Is What I See

Introduction générale

1. Introduction

Le progrès important perçu, ses dernières décennies, dans le domaine des réseaux et internet a ouvert des perspectives extrêmement intéressantes, notamment en termes d'interaction sociale à distance, tout en permettant l'utilisation de nouveaux moyens de communication et de collaboration dans un large éventail de domaines tels que *l'e-gouvernement, e-business, e-maintenance, e-learning*, etc. Le monde éducatif est l'un des domaines les plus intéressés par l'adoption de la technologie pour améliorer le processus d'apprentissage collaboratif. Dans cette perspective, de nombreuses études et recherches ont été menées afin de modéliser, concevoir, analyser et mesurer des environnements virtuels assistant l'apprentissage collaboratif [Herrera-Pavo 21][Verbruggen et al. 20][Hernández-Sellés et al. 19][Jeong et al. 19][Hedjazi 18][Ferradji 17][Akgün et Akkoyunlu 13][Hedjazi 11].

Par ailleurs, l'Apprentissage Collaboratif (AC) en présentiel, qui représente le contexte de notre travail de recherche, peut être considéré comme étant une démarche qui se base aussi bien sur l'effort personnel de l'apprenant que les échanges, les confrontations, les négociations et les interactions interpersonnelles. Il s'agit donc d'un travail collectif impliquant plusieurs acteurs dans un objectif commun qui est la construction de connaissance individuelle et collective. Une telle démarche propose ainsi à l'apprenant un processus dynamique et réflexif de construction de connaissance [Hedjazi et Zidani 11]. L'amélioration d'un tel processus par de nouvelles technologies permettant aux apprenants de suivre des sessions d'apprentissage, à n'importe quel moment et à partir de n'importe quel endroit géographique, constitue actuellement, le souci majeur des organisations éducationnelles. En effet, il est possible d'explorer les technologies de l'information (en Anglais *IT: Information Technologies*) pour surmonter l'obstacle de

distance tout en supportant l'AC et en conséquence réduire considérablement les coûts associés à un tel apprentissage. Ceci donne naissance d'un nouveau domaine de recherche baptisé Apprentissage Collaboratif Assisté par Ordinateur (ACAO en anglais CSCL : *Computer Supported Collaborative Learning*). Les études menées dans ce domaine de l'ACAO s'appuient principalement sur les idées de plusieurs disciplines académiques à savoir : la psychologie, la sociologie et l'informatique [Gašević et al. 18]. Elles examinent également l'utilisation de l'informatique pour améliorer l'interaction et la collaboration des apprenants, notamment pour améliorer la réussite de l'apprentissage [Reis et al. 18] [Stahl et al. 06].

2. Contexte de travail

Le travail de recherche, réalisé dans le cadre de cette thèse, s'inscrit dans le contexte de l'ACAO, plus particulièrement l'ACAO de la programmation informatique. La programmation informatique est un processus qui peut se résumer en la rédaction d'une suite de lignes de code appelées code source ou programme source. Ce dernier décrit la séquence d'instructions à exécuter par la machine pour résoudre un problème donné. L'apprentissage de la programmation informatique est un processus souvent collaboratif qui ne diffère pas de celui de n'importe quelle autre discipline. En apercevant le déroulement d'un tel processus dans une classe classique (c.-à-d. *en face à face*), il a été révélé que les apprenants apprennent mieux la matière en se confrontant à un problème, et l'apprentissage évolue graduellement en partant généralement du plus simple au plus difficile. Cette méthode dite *l'Apprentissage Par Problème (APP)* constitue l'une des stratégies les plus répandues dans le domaine éducatif ainsi qu'elle est, depuis plusieurs années, devenues un thème populaire [Chiang 17]. D'autre part, lors de la résolution des problèmes de la programmation, il est avéré que les apprenants, en plus de résolution collective du problème, accomplissent des tâches de synchronisation, de communication, de négociation et d'interaction de groupe. C'est ainsi que les chercheurs travaillant sur l'apprentissage collaboratif s'accordent souvent avec la théorie de *Vygotsky* stipulant que la construction des connaissances se réalise d'abord par l'interaction sociale avant d'être internalisée [Charlier et al. 98]. Par conséquent, l'interaction sociale encourage les apprenants à partager et à exprimer une variété de points de vue et finalement élargir leur réflexion au-delà des compétences personnelles [Isohäätä et al. 17]. Pour toutes ces raisons que les tendances actuelles se sont concentrées sur l'AC, qui a largement démontré son efficacité, notamment en termes d'amélioration de la motivation des apprenants ainsi que la simplification de l'acquisition des connaissances [Koivuniemi et al. 18] [Scager et al. 16].

Dans le secteur de l'enseignement supérieur, l'AC est une stratégie utile et précieuse, car elle forme les étudiants à des métiers où ils travaillent en groupe [De Hei et al. 15]. Ainsi,

la combinaison de la stratégie d'AC avec la méthode APP semble assez prometteuse et utile, dans l'enseignement de la programmation informatique, du fait qu'elle offre aux apprenants plusieurs possibilités d'interaction de groupe, en particulier pour poser des questions, échanger des opinions et discuter des résultats. Cela permet aux apprenants d'améliorer aisément leurs compétences de développement de programmes. Toutefois, de nombreux chercheurs ont argumenté que pour atteindre une qualité satisfaisante d'AC, il est essentiel de concentrer l'intention sur l'interaction sociale [Molinillo et al. 18] [Hurst et al. 13][Spears 12][Jung et al. 02][Moore 93][Vygotsky 78].

En fait, notre défi majeur, dans ce contexte, n'est pas uniquement technique, mais surtout social. Il sera donc essentiel de comprendre comment adapter et affiner nos supports collaboratifs pour favoriser l'activité d'apprentissage de la programmation informatique et stimuler efficacement l'enthousiasme et l'engagement des apprenants.

3. Problématique de recherche

En utilisant les technologies collaboratives, apprendre à distance la programmation informatique à plusieurs va-t-il y avoir une influence sur l'apprentissage ? C'était la question principale de départ de nos recherches. En effet, pour répondre de manière objective à cette question nous avons adopté une méthodologie s'appuyant sur deux points essentiels : d'une part, concevoir et implémenter un prototype de système *Groupware* supportant l'ACAO de la programmation informatique, et d'autre part, étudier l'applicabilité du modèle de la *Théorie Unifiée de l'Acceptation et de l'Utilisation de la Technologie (UTAUT)* de Venkatesh et al. (2003) dans un cadre d'enseignement supérieur Algérien afin d'expliquer les facteurs influençant les intentions des apprenants d'adopter un tel prototype pour suivre à distance des sessions d'apprentissage de la programmation.

4. Objectifs et méthodologie de recherche

L'objectif principal, ciblé par les travaux de recherche de cette thèse, peut se résumer en la proposition d'une nouvelle approche permettant de répondre objectivement aux besoins des facultés informatiques Algériennes. Nos facultés d'informatique qui souhaitent également explorer l'ACAO pour saisir à la fois les opportunités d'améliorer les méthodologies d'apprentissage habituelles dans l'enseignement informatique, en particulier dans l'apprentissage de la programmation et de la nécessité de former les futurs développeurs de logiciels à la collaboration à distance. Dans cette perspective, en s'accordant avec l'idée Johnson (1998) stipulant que « *le processus d'analyse et de critique des artefacts logiciels produits par d'autres personnes est une méthode puissante pour apprendre les langages de programmation et les techniques de conception* », nous avons abordé notre

problématique, qui permet d'étudier les difficultés que rencontrent les apprenants en suivant des séances d'AC de programmation informatique en présentiel, tout en proposant une nouvelle approche basée sur les fondements de l'ACAO. Par conséquent, nous avons développé un prototype d'un *Groupware* appelé *PBPCLG* (*Problem-Based Programming' Collaborative Learning Groupware*) de support de l'AC à distance de la programmation informatique. Un tel *Groupware* intègre plusieurs outils supportant la collaboration, la coordination et l'interaction entre un groupe d'apprenants géographiquement dispersés. Il offre, en effet, un espace de travail partagé assistant le déroulement d'une session d'apprentissage de la programmation, tout en résolvant un problème particulier. Cet espace partagé offre un éditeur collaboratif de code source ainsi qu'un nombre d'outils collaboratifs, fonctionnant avec deux modes d'interaction à travers lesquels les apprenants peuvent contribuer individuellement ou collectivement aux processus de développement du code source partagé.

L'espace partagé comporte aussi un espace de modélisation graphique qui donne, aux apprenants et tuteurs, une vision globale du programme source en cours de développement. En fait, un diagramme de classe UML décoré nommé *DCD* (*Decorated Class Diagram*) [Chorfi et al. 20] modélisant les modules du programme source partagé peut jouer un rôle primordial, notamment pour synchroniser et coordonner les activités collaboratives des apprenants.

Par ailleurs, par le fait que le concept de conscience de groupe constitue un facteur clé de l'efficacité de tout travail collaboratif, nous lui accordons une importance extrême dans notre démarche, et ce dans le but de préserver l'homogénéité de la mémoire collective des collaborateurs (apprenants et tuteurs). Plusieurs mécanismes de conscience de groupe ont été intégrés dans *PBPCLG* à savoir : un mécanisme d'affichage de la *Liste des participants présents* qui affiche en temps réel tous les apprenants participants à la session en cours ; un mécanisme d'*audioconférence* qui peut être utilisé pour échanger des messages vocaux entre les participants. Un mécanisme de *chat* avec son module de gestion d'historique pour gérer les interactions textuelles échangées ; un système de *notifications* qui peut être exploité pour diffuser aux participants tout changement d'état effectué dans l'espace de travail partagé ; etc.

Enfin, l'étude expérimentale de notre prototype *PBPCLG*, dans un contexte réel d'apprentissage de la programmation, représente la dernière étape de notre démarche. Elle joue un rôle important pour tirer les bonnes conclusions, notamment en termes de spécifications des facteurs influençant l'adoption, par nos étudiants informatiques, d'une telle approche. Nous avons ainsi mené plusieurs expérimentations faisant participer au total 119 étudiants de l'Université *d'Oum el Bouaghi* - Algérie.

5. Organisation de la thèse

Nous avons organisé notre thèse en cinq chapitres dont les trois qui suivent cette introduction générale, décrit la partie état de l'art des trois domaines de recherche connexes à ce travail, ensuite les derniers chapitres présentent respectivement les parties : conception, implémentation et expérimentation du prototype concrétisant notre approche.

En conséquence, le *chapitre 1* intitulé « *Aspects théoriques de l'apprentissage* » présente de manière claire les différents aspects théoriques liés aux activités d'apprentissage, particulièrement l'apprentissage collaboratif.

Le *chapitre 2* expose les fondements de notre contexte de recherche qui est l'« *Apprentissage Collaboratif Assisté par Ordinateur (ACAO)* » en anglais : *Computer-Supported Collaborative Learning (CSCL)*.

Le *chapitre 3* intitulé « *Systèmes collaboratifs dédiés à l'apprentissage de la programmation* », explique également les caractéristiques essentielles de quelques systèmes proposés dans littérature pour supporter l'apprentissage/développement collaboratif de la programmation informatique.

Le *chapitre 4* intitulé « *Présentation de l'environnement PBPCLG* », aborde les choix conceptuels et techniques adoptés pour mettre en œuvre le *Groupware* proposé.

Le *chapitre 5* intitulé « *Facteurs d'acceptation et d'utilisation de l'environnement PBPCLG : preuves empiriques du modèle UTAUT* », décrit méthodologiquement l'aspect expérimental de ce travail de recherche, qui se base également sur l'application du modèle **UTAUT** afin de tirer les bonnes conclusions à propos des effets du *Groupware* proposé sur le comportement de ses utilisateurs effectifs à savoir les apprenants et tuteur.

Enfin, nous terminerons ce manuscrit par un bilan résumant nos contributions proposées dans le cadre de cette thèse ainsi qu'une proposition de quelques perspectives à envisager pour accomplir ce travail.

Chapitre 1 : Aspects théoriques de l'apprentissage

1.1 Introduction

L'apprentissage est, depuis longtemps, un domaine d'intérêt pour l'homme. Différentes hypothèses et théories de l'apprentissage ont émergées lors de l'évolution de la pensée humaine et du développement de l'appréhension des connaissances par la méthode de recherche de *l'apprentissage/enseignement*. Par conséquent, les théories de l'apprentissage tiennent en compte de l'explication du phénomène d'acquisition des connaissances. Selon les psychologues, le processus d'acquisition de connaissances forme la plateforme essentielle sur laquelle les théories d'apprentissage appuient afin de donner des explications objectives à l'apprentissage. Dans notre contexte, nous nous intéressons à l'apprentissage collectif autrement dit apprentissage collaboratif.

Nous avons préféré de consacré tout un chapitre à l'étude des théories de l'apprentissage afin de bien comprendre leurs visons et de tirer profit de leurs résultats et conclusions pour pouvoir y prendre en considération lors de la conception et l'implémentation de nos solutions technologiques assistant le processus de l'apprentissage collaboratif de la programmation informatique.

Toutefois, le développement de la psychologie de l'apprentissage a contribué, par son évolution, à l'amélioration de la qualité et de l'efficacité du processus pédagogique global.

Il existe de nombreuses théories de l'apprentissage, des méthodes d'enseignement, parfois convergentes, quelquefois divergentes, des différents modèles de rôles de l'enseignant et de l'enseigné. En effet, les différents aspects liés au comportement de l'apprenant, individuellement ou en groupe, sont également misent en exergue dans ce chapitre.

1.2 Théories de l'apprentissage

La littérature sur l'apprentissage discute de plusieurs théories d'apprentissage. Nous évoquons dans cette section, sans prétendre à l'exhaustivité, les grandes théories de l'apprentissage à savoir : *Béhaviorisme*, *Constructivisme*, *Socioconstructivisme*, *Interactionnisme*

1.2.1 Béhaviorisme

Le *béhaviorisme* qui représente la 1^{ère} grande théorie de l'apprentissage qui a beaucoup repéré les domaines de l'enseignement, de l'éducation et de la formation. La notion *béhaviorisme* est construite à partir du mot anglo-saxon « *behavior* » qui signifie « *comportement* ». La théorie *béhaviorisme* voit le processus d'apprentissage comme étant une modification durable du comportement produit de l'effet d'un entraînement particulier. Dans cette optique, les éléments suivants peuvent constituer une base sur laquelle l'apprentissage appuie :

- La matière à apprendre est divisée en un déroulement d'éléments courts pour assurer un renforcement le plus vite facile,
- Dans le but de favoriser un apprentissage sans erreur, l'enchaînement des contenus doit se faire de manière graduelle partant du niveau le plus facile jusqu'aux niveaux compliqués,
- Les renforcements positifs constituent un facteur clé de l'apprentissage. Ils doivent être considérés le plus vite possible

1.2.2 Constructivisme et Socioconstructivisme

Selon la théorie du *Constructivisme*, l'apprenant est situé au centre du processus d'acquisition de connaissances. Ces dernières se développent par l'apprenant lui-même en exploitant les connaissances déjà acquises pour s'adapter à des nouvelles conditions [Ferradji 17].

La *Socioconstructivisme* est une théorie développée par *Vygotsky* en 1985. Ce dernier trouve que la dimension sociale ne doit pas être ignorée, mais elle doit être considérée lors du processus d'acquisition de connaissance. Par ailleurs, cette théorie représente un modèle d'extension du modèle de la théorie constructiviste de manière à intégrer, dans le processus d'acquisition de connaissances, les aspects interaction, communication et négociation entre

les apprenants et les enseignants. Par conséquent, aborder le processus de l'apprentissage consiste à étudier le processus d'acquisition de connaissances comme étant une activité sociale, tout en permettant aux apprenants de confronter leurs savoirs et expériences à ceux des autres.

1.2.3 Interactionnisme

Quant à l'*interactionnisme* est un terme formé par la concaténation du mot *interaction* et le suffixe *-isme* qui sert une théorie de l'apprentissage développée aux *Etats-Unis* au milieu du *XXème siècle*. Cette théorie considère le processus d'acquisition de connaissance de l'apprenant au sein du lien social à l'intérieur d'un système de sens et de valeurs. En d'autres termes, les *interactionnistes* stipulent que l'individu et la société se construisent lors d'interactions.

1.3 L'apprentissage selon la pédagogie ouverte

La pédagogie ouverte [Paré 77] est un concept composé également de deux mots. Le mot « *ouverte* », qui fait référence aux ressources éducatives ouvertes, qui sont déterminées par l'*UNESCO* comme étant « *tout type de matériel éducatif qui est dans le public* ». Quant au mot « *pédagogie* », il représente la pratique et la méthode d'éducation définissant ainsi la manière d'éducation, plutôt que ce que l'éducation elle-même. En fait, la pédagogie ouverte est l'utilisation de ressources éducatives ouvertes pour soutenir l'apprentissage. En utilisant une telle pédagogie dans une classe, les apprenants sont appelés à faire partie du processus d'apprentissage, à participer à la construction de connaissances.

La pédagogie ouverte est alors un modèle d'enseignement qui est axé aussi bien sur l'apprenant que sur l'évolution de ses habiletés intellectuelles, sociales, émotionnelles, morales et religieuses. Elle reconnaît, entre autres, que l'apprenant est le principal élément de son apprentissage et qu'il est en interaction avec son environnement. Par conséquent, il est focalisé sur les interactions de groupe qui peuvent être envisagées entre l'apprenant et son environnement éducatif. C'est suite à ces interactions que se dégageront des liens significatifs pour l'apprenant et qui lui permettront de provoquer son processus d'apprentissage. En d'autres termes, selon la pédagogie ouverte l'apprenant doit apprendre à être autonome.

La pédagogie ouverte ne s'intéresse pas uniquement à l'interaction entre l'apprenant et son environnement, mais aussi à l'interaction mutuelle entre l'apprentissage des apprenants et les interventions de l'éducateur (enseignant ou tuteur) qui favorisent le développement de l'environnement éducatif [Paquette 76][Paquette 79]. En effet, les éducateurs ont un rôle primordial dans le processus d'apprentissage. Ils s'intègrent dans le groupe en tant que *conducteurs et/ou participants*, tout en aidant le groupe à comprendre

les facteurs *cognitifs* qui ont une influence sur l'apprentissage. Par conséquent, ils contribuent en premier lieu à bien aménager l'environnement éducatif de manière à considérer trois niveaux ou dimensions à savoir : le niveau physique d'une classe, niveau des activités d'apprentissage et enfin le niveau d'intervention de l'éducateur.

La pédagogie ouverte cible à mettre en œuvre un processus éducatif centré sur la compréhension de l'autre, la considération des diversités, le partage des opinions, produisant ainsi une école effectivement démocratique. Elle privilégie divers moyens pédagogiques susceptibles d'actualiser les compétences espérées des apprenants.

1.4 L'apprentissage en groupe-classe

À travers certaines définitions de la classe, nous pouvons la considérer ici comme étant un système, un groupe capable de traiter des informations, un ensemble de personnes dans un état d'interaction directe, dans le sens où chaque personne agit en interaction avec les autres [Homans 50] [Homans 61].

Le point essentiel de cette notion recouvre les interactions spécifiques d'un petit groupe. En effet, pour *Homans* l'interaction est un processus élémentaire, ainsi que l'activité (*toutes les tâches réalisées par les participants*) et le sentiment (*les états intérieurs, les émotions, les attitudes qui se dégagent, entre autres, par l'interaction*), se conçoivent comme un élément de base du comportement non seulement individuel, mais aussi de groupe. Il convient de noter que ces éléments (*activité, interaction et sentiment*) doivent être intégrés dans un *système externe*, puis dans un *système interne*.

À partir du *système externe*, qui désigne l'adaptation du groupe à son environnement se met en place le *système interne* qui assure la survie au niveau du groupe, sans contrainte liée à l'environnement. Ce *système interne*, qui implique des relations mutuelles entre les interactions, les sentiments et les activités, permet de tenir compte de la dynamique interne du groupe. Cette dynamique est réalisée par le groupe et pour ses membres, bien que le groupe soit organisé en fonction des influences du *système externe*.

Le groupe de classe est un ensemble qui porte un *système externe* et un *système interne* en interaction. Dans une sorte de processus circulaire, le *système externe* approvisionne le *système interne* et réciproquement. Le groupe de classe est donc défini non seulement par les éléments qui le constituent, mais aussi par la structure qui rend son fonctionnement opérationnel. La multiplicité des éléments séparés, les interrelations des éléments et la cohésion de l'ensemble constituent les caractéristiques de ce système. Le groupe de classe est donc un système qui apprend à produire des résultats à partir de divers facteurs de participation à savoir : des facteurs relatifs à la classe, des facteurs relatifs à l'école (organisation éducative), des facteurs relatifs à la société, et des facteurs relatifs à la vie

de l'apprenant. Tous ces facteurs engendrent ce qu'on appelle la *dynamique motivationnelle de l'apprenant* (Figure 1.1).

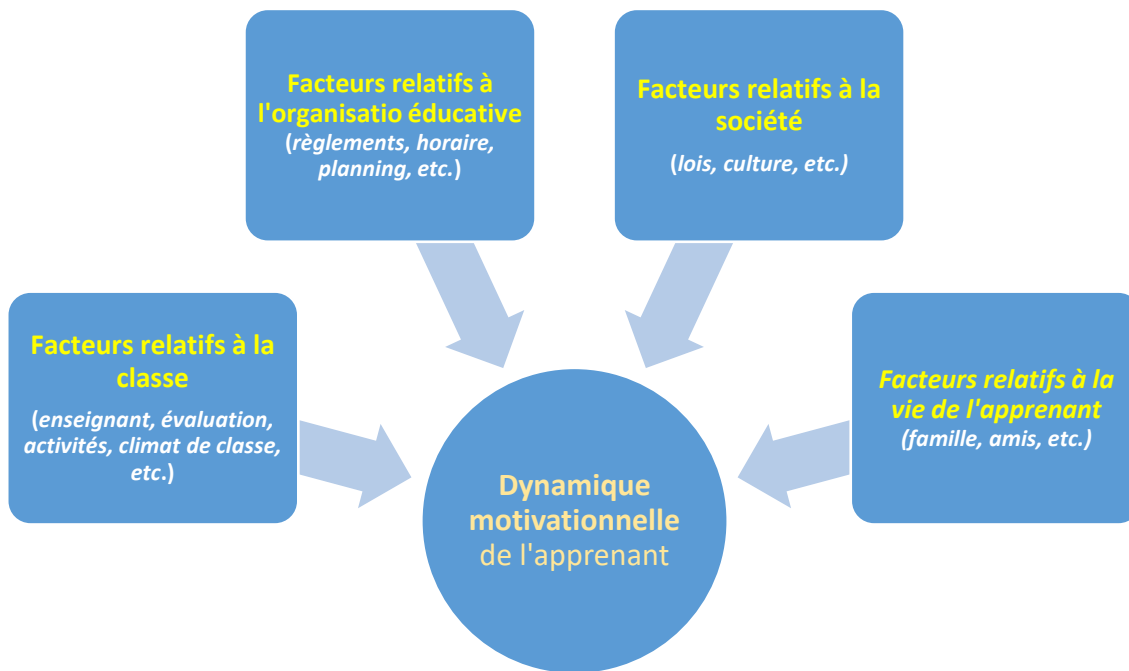


Figure 1. 1 : Facteurs influençant la Dynamique motivationnelle de l'apprenant.

Le groupe de classe étant considéré comme un système d'apprentissage. Une classe doit être envisagée en tant que groupe lorsque les éléments propres à ce groupe sont présents dans la classe. Ces principaux éléments sont : *l'interaction, la participation, la communication, le leadership et la prise de décision.*

1.4.1 Interaction

Dans une situation où l'apprentissage se déroule en présentiel, lorsqu'un enseignant se présente devant ses apprenants, une interaction dans la classe est également produite. Par conséquent, il se peut y avoir deux formes d'interaction : l'interaction fonctionnelle, qui est focalisée sur l'activité à exécuter et l'interaction affective qui est axée sur le principe stipulant un mieux-être avec soi et autrui. C'est ainsi que Morval (1967) a également défini la notion d'interaction dans le groupe comme étant « *un système d'actions réciproques qui permet un échange d'énergie et d'opinions entre partenaires de l'interaction* » [Morval 67].

Il est utile de préciser que l'interaction dans une telle situation d'apprentissage peut avoir deux configurations possibles : l'interaction à sens unique et l'interaction à double sens. L'interaction à sens unique aura lieu lorsque l'un des participants génère un comportement actif, alors que l'autre est passif ou inactif. Quant à l'interaction à double sens, elle n'est produite que lorsque deux ou plusieurs individus interagissent pour une

durée de temps bien déterminée. Elle se détermine, en effet, par une stimulation réciproque autrement dite « *interaction réciproque* » [Morval 71]. Selon *Bany et Johnson* (1971) une interaction réciproque se déroule entre un groupe et un autre, entre un groupe et l'enseignant, ou entre des sous-groupes au l'intérieur du groupe [Bany et Johnson 71].

1.4.2 Participation

La participation est un phénomène qui ne peut se produire que lorsqu'il y'a une interaction entre les membres du groupe. Cette notion a été intégrée dans le modèle *St-Arnaud* (1978) de façon à placer chaque apprenant sur son axe de participation par rapport aux objectifs initialement définis pour attribuer un sens à l'apprentissage. C'est à travers cet axe qu'il est possible de mesurer le degré d'engagement de chaque apprenant par rapport aux objectifs partagés de l'apprentissage [St-Arnaud 79]. En effet, une notion dite « *distance psychologique* » a été définie pour préciser l'intervalle qui peut exister entre un apprenant et l'objectif de la classe.

1.4.3 Communication

Ce modèle permet de confirmer que la participation et la communication sont des formes d'interaction, car une personne interagit généralement avec les autres afin de participer, de partager des informations et de communiquer.

Selon *Anzieu et Martin* (1968) et *Ardoino* (1966), la communication au sein d'un groupe de classe n'est donc pas une simple diffusion de l'information, mais plutôt une mise en relation entre les membres de la classe [Anzieu et Martin 68] [Ardoino 66]. Par conséquent, la communication se fait présenter au moment dont un apprenant ou un enseignant prend en compte une information qui a été délivrée par autrui.

1.4.4 Prise de décision

L'interaction, la participation et la communication engendrent une influence relationnelle au sein du groupe (baptisée autorité). Le *leadership*, qui résulte des forces du groupe, est considéré comme une aptitude à influencer et à être influencé. Chez l'apprenant, lorsque ces aptitudes sont bien contrôlées, ce dernier participera aisément et efficacement au processus de prise de décision au sein du groupe-classe. En fait, la décision, résultat d'un processus décisionnel, sera d'autant plus satisfaisante quand les apprenants/enseignant sont positivement engagés aux débats et dotés d'une attitude compréhensive.

1.5 Apprentissage situé

Du fait que l'être humain n'est pas conçu à être développé dans un environnement isolé, mais il est inséré dans un contexte social inséparable, *Lave & Wenger (1991)* ont contredit le principe de l'apprentissage classique qui est considéré comme une activité de transmission directe de la connaissance. En effet, Il a proposé une nouvelle théorie baptisée « *Apprentissage Situé* » qui aperçoit l'apprentissage comme une activité qui ne doit pas être séparée de son contexte social. Cela contraste avec ce qui se passe habituellement aux organisations éducatives, où le contenu pédagogique, qui représente la connaissance, est souvent donné de façon abstraite et dissociée du contexte.

L'apprentissage situé (en anglais : *situated learning*) autrement dit « *Apprentissage Contextualisé* » est une théorie qui s'inscrit dans des environnements pédagogiques qui prennent en charge les préoccupations des apprenants. Il s'agit donc d'un processus de construction de connaissances à partir de l'action [Tardif 98]. Selon cette théorie, l'apprentissage se développe, à tout moment, et dans n'importe quel contexte où peut se dérouler une activité humaine. Le déroulement de l'apprentissage situé, qui est adéquat pour le travail, l'environnement et la culture dans laquelle il se développe, s'effectue en étudiant la nature des problèmes rencontrés de manière à appliquer la théorie à la pratique. L'apprentissage situé est donc basé sur la réalité afin de proposer des activités similaires, notamment à des problèmes professionnels. Dans ce cas, l'apprentissage a un sens pour les apprenants du fait qu'il inclut des activités qui sont à proximité de leurs vies privées, besoins et intérêts.

Dans cette dialectique, nous pouvons juger que le but d'un tel apprentissage n'est pas uniquement la rétention de la connaissance, mais plutôt la mise en pratique d'une telle connaissance. L'accent est mis ainsi sur l'activation d'un processus mental d'un niveau supérieur qui permet l'application de la connaissance dans son contexte. C'est pour cette raison que l'enseignant aura une tâche fastidieuse, particulièrement pour trouver des liens entre le contenu pédagogique et des situations réelles. Cela encourage largement l'interaction sociale et la collaboration dans l'environnement d'apprentissage des apprenants. Enfin, les caractéristiques essentielles de l'apprentissage situé peuvent être résumées en ce qui suit :

- *L'apprentissage situé est caractérisé par sa grande liaison avec le contexte socio-culturel de l'apprenant, son environnement d'apprentissage et ses connaissances.*
- *Il est aussi vu comme une action de participation et d'intégration à un contexte social en maîtrisant graduellement le principe de son évolution.*
- *Il surmonte la situation de transfert classique du savoir, en allant vers le développement de l'expertise de l'apprenant en interaction avec des acteurs professionnels dans un climat qui*

favorise autant la formalisation et la négociation du sens des connaissances et savoir-faire, que l'expérience et la contribution à une pratique professionnelle.

1.6 Apprentissage selon la théorie de l'activité

La théorie de l'action est l'investigation de la nature de l'action humaine dans son contexte aussi bien individuel que collectif. Revenant à l'origine d'une telle théorie, elle représente une filière de la philosophie concernée par l'analyse de ce que les humains effectuent volontairement. De nombreuses théories se réfèrent comme théorie de l'action telles que : Théorie générale de l'action de *Parsons* (1951), Théorie de l'action de *Norman* (1986), Théorie de l'action située introduite par *Suchman* (1987), etc.

Dans un contexte collaboratif où les êtres humains interagissent autour d'un objectif commun, les théories de l'action semblent insuffisantes, notamment pour étudier certains problèmes complexes inhérents de l'activité collective. Elles peuvent, en effet, être complétées par les « *théories de l'activité* ».

L'apprentissage est certainement une activité humaine, et qui n'a de perception que dans une structure propice au modèle de l'activité. La théorie de l'activité, qui est d'origine introduite par les spécialistes en psychologie, peut faire l'objet d'une adaptation pour créer un climat propice particulièrement à l'apprentissage, à la construction de connaissances et à l'acquisition d'expertises. Par conséquent, une 1^{ère} génération d'une telle théorie s'appuie sur le principe que l'être humain détermine ses activités suite aux activités de ses partenaires (acteurs de l'environnement), tout en prenant en compte le contexte dans lequel s'inscrit un apprentissage [Vygotski 34]. Ensuite, de nombreuses recherches ont souligné que la compréhension de la vie humaine dans son environnement, notamment celui d'apprentissage nécessite l'interprétation des interactions de l'individu humain avec le monde comme étant médiatisées par les objets, les méthodes, les règles, communauté, etc. C'est ainsi qu'*Engeström* (1997, 2001) a accentué dans son système d'activité (Figure 1.2) que l'analyse de l'activité se situe, au sein d'une communauté qui se structure, et se pose sur des règles et une répartition des tâches (division du travail). En analysant le système décrit sur la figure 1.3, plusieurs concepts sont à considérer :

sujet, qui représente l'individu ou le sous-groupe concerné par l'analyse ; *objet*, qui est la transformation de l'environnement visée par l'activité ; *outils*, qui peuvent être matériels ou symboliques, utilisés pour médiatiser l'activité ; *communauté* qui un ensemble des sujets qui visent la production du même objet ; la division du travail entre les sujets ou les membres de la communauté ; les *règles*, qui font référence aux normes, conventions, habitudes qui maintiennent et régulent les interactions à l'intérieur du système.

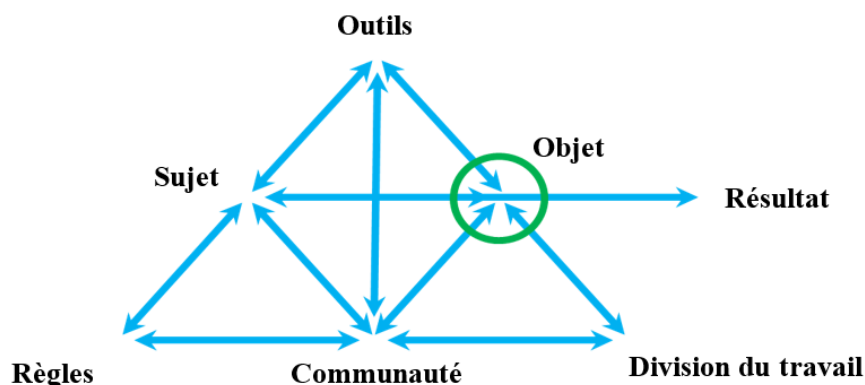


Figure 1. 2 : Système d'activité d'Engeström [Engeström 01].

Une 2^{ème} génération de la théorie de l'activité est inspirée des travaux de *Leontiev*. *Leontiev*, un des pères de la théorie de l'activité, traite l'activité comme étant un phénomène de travail collectif, tout en spécifiant trois niveaux d'activités à savoir : les *activités*, les *actions* et les *opérations*. Une *activité*, qui provient d'un besoin, est liée à un motif ainsi qu'elle est menée par la communauté. Une *action*, qui vise un but bien déterminé, est menée par l'individu ou un groupe d'individus. Enfin, une *opération*, qui est une procédure compilée et inconsciente, est orientée vers des conditions et moyens d'exécution. Lorsque ces conditions d'exécution ont changé, une opération peut à nouveau devenir une action pour être spécialisée et adaptée à ses nouvelles conditions. Une opération est menée par une routine exécutée soit par un être humain ou par une machine. La relation entre l'activité et les actions, et les opérations est également dynamique. Ainsi, une action peut servir une ou plusieurs activités et une opération sert à exécuter une action selon des conditions nécessaires.

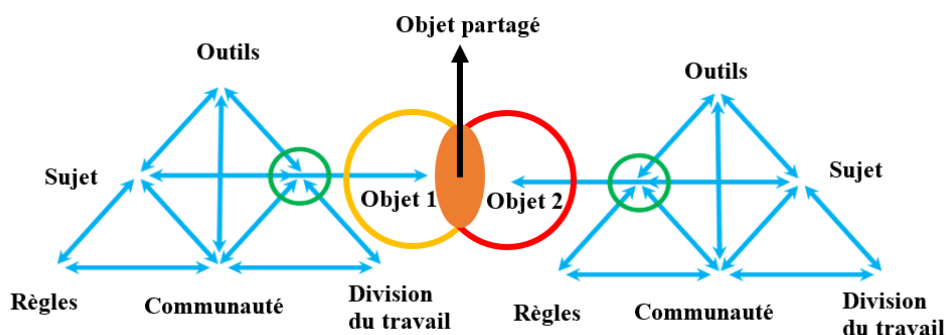


Figure 1. 3 : Deux systèmes d'activité [Engeström 01].

En effet, l'apprenant ou le groupe d'apprenants au sein d'une classe sont un déterminant fondamental de l'activité de l'éducateur. L'adoption des scénarios ainsi que leurs progressions, en classe, se base également sur les prérequis des apprenants, les

modes de fonctionnement que l'éducateur évalue comme possibles dans la classe. En classe, les activités pédagogiques de l'éducateur dépendent de l'activité des apprenants, individuellement ou collectivement. Un feedback est toujours renvoyé à l'éducateur afin d'adopter certains éventuels ajustements sur ses tâches.

1.7 Tendances éducatives

Les théories d'apprentissage que nous avons évoqué dans la première partie de ce chapitre forment une base importante pour différencier les tendances éducatives (autrement dites courants pédagogiques) d'un côté, et pour adopter la méthode d'apprentissage de l'autre côté. En effet, nous pouvons juger que le modèle béhavioriste correspond à la pédagogie traditionnelle. Le constructivisme et le socioconstructivisme correspondent à une pédagogie active qui se décline en apprentissage collectif, en apprentissage par problèmes et en apprentissage par projet [Abdallah 09][Thien 14]. Dans la suite de cette section, nous présentons de manière générale les caractéristiques de ces deux courants (*pédagogie classique et active*).

1.7.1 Pédagogie classique

Cette tendance, qui est centrée sur la matière pédagogique enseignée, se base également sur la transmission du savoir entre l'enseignant et l'apprenant. C'est ainsi que, dans certaines littératures, elle est baptisée pédagogie transmissive. Les caractéristiques essentielles d'une telle pédagogie ont été bien discernées par *Legendre* (1993). Elles concernent aussi bien l'activité de l'enseignant que celle des apprenants en classe. L'enseignant s'occupe de plusieurs tâches, notamment dans la classe à savoir : la gestion du temps, la transmission du contenu pédagogique théorique tel que les concepts et notions, les théorèmes, les règles, etc., la proposition des exercices relativement liés aux contenus théoriques acquis par les apprenants, le suivi de la résolution de problèmes. Quant aux apprenants, la pédagogie classique leur permet le développement de leurs compétences, tout en se limitant uniquement à la rétention, à l'écoute, reconstruction de la connaissance abstraite. Par ailleurs, le déroulement de l'apprentissage s'appuie également sur la communication orale des savoirs. Enfin, le processus d'évaluation se fait à travers des examens écrits et corrigés par l'enseignant en attribuant des notes aux apprenants.

1.7.2 Pédagogie active

Les fondateurs principaux de la pédagogie active sont également les deux grands pédagogues *Friedrich Fröbel Johann* et *Heinrich Pestalozzi*. Ce type de pédagogie est reconnu notamment pour donner une liberté à l'apprenant, en fonction de ses besoins, d'agir sur son apprentissage. Contrairement à la pédagogie traditionnelle (classique) qui

est centrée sur le contenu pédagogique, la pédagogie active est centrée sur l'apprenant. Ainsi, les connaissances, au lieu d'être apprises de façon directe, elles sont construites par les apprenants. L'accent est mis alors sur le principe de rendre l'apprenant acteur principal de ses apprentissages afin qu'il construise ses savoirs. C'est lui qui choisit le contenu pédagogique à apprendre et l'heure de l'apprentissage. De plus, la position de l'enseignant par rapport à l'apprenant est totalement changée, de manière qu'il ne devient qu'un simple médiateur entre le savoir et l'apprenant.

Par ailleurs, la pédagogie active favorise les situations réelles de recherche et d'étude qu'à travers lesquelles l'apprenant peut maîtriser certaines connaissances ciblées par l'enseignant. Ces situations peuvent être concrétisées dans des séances en utilisant plusieurs types d'apprentissage à savoir : l'apprentissage par problèmes, l'apprentissage collectif, l'apprentissage par découverte et l'apprentissage par projet [Thien 14].

1.7.2.1 Apprentissage par découverte

L'apprentissage par découverte est considéré comme une des méthodes pédagogiques associées aux théories constructivistes de l'apprentissage. Lors d'un tel apprentissage, les apprenants sont guidés par un éducateur ou enseignant pour découvrir des notions, des règles, des concepts, etc., d'une thématique donnée. Au contraire à l'apprentissage classique dont les unités d'apprentissage et leurs contenus sont généralement détachés les uns des autres, cette méthode exige d'établir une relation entre ces unités. Selon *Bruner*, chaque apprenant bien guidé par son formateur peut découvrir la structure d'une thématique et l'apprendre. Par conséquent, ce type de pédagogie s'appuie sur des thèmes présentés aux apprenants et démontre son fondement dans la pratique. Elle favorise donc l'apprentissage par projet avec le travail collectif (collaboratif/coopératif).

Deux manières peuvent avoir lieu afin de mettre en place l'apprentissage par découverte : l'apprentissage non guidé, qui limite le rôle de l'enseignant. Toutefois, les apprenants n'ont souvent pas les connaissances antérieurement acquises et qui sont jugées utiles pour mener à bien leur tâche. C'est ainsi que *Bruner* n'admettait pas cette manière. Quant à l'apprentissage guidé, où l'enseignant accompagne et guide ses apprenants même en termes de rappel des prérequis. Il accepte, rejette et corrige les hypothèses formulées par les apprenants. Il pose parfois des questions pour orienter la démarche des apprenants (Figure .1.4).

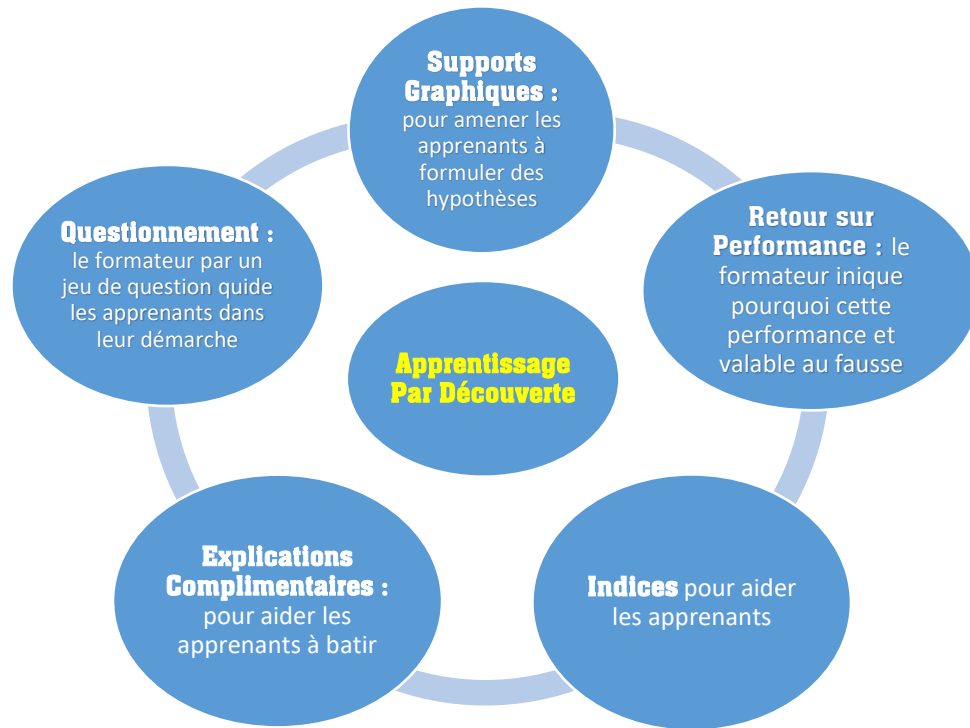


Figure 1. 4 : Apprentissage par découverte.

1.7.2.2 Apprentissage collectif

En se concentrant sur la différence entre individuel et collectif, il est à présent possible de mettre en évidence la hiérarchie du travail collectif dans l'apprentissage, en particulier de nombreuses études confirment que l'apprentissage collectif est plus efficace que l'apprentissage individuel [Legros et al. 02].

Les recherches dans le domaine de l'apprentissage collectif soulèvent souvent une différence entre la notion de *collaboration* et celle de *coopération*. La **collaboration** est un phénomène qui peut se présenter lorsque les membres du groupe interviennent, de la même façon, à l'activité partagée. Les mêmes tâches sont, en effet, effectuées par tous les participants. Quant à la **coopération**, l'activité partagée est décomposée en plusieurs sous-activités sur lesquelles les participants interviennent avec des rôles différents (Figure 1.5). La collaboration nécessite une intention commune des apprenants ainsi que des efforts mutuels et cohérents pour résoudre les problèmes [Dillenbourg 99]. La coopération, en revanche, exige une division des tâches beaucoup plus grande, Il est globalement évident que les groupes de coopération et de collaboration agissent et travaillent pour atteindre un *but commun* ou *partagé* [Henri et al. 01].

Dans ce contexte, le terme "*partage*" fait référence à deux significations différentes : premièrement, il se réfère à la *décomposition* d'un ensemble en plusieurs parties et

deuxièmement, il peut signifier aussi la *participation* à quelque chose [Henri et al. 01]. La coopération exige le partage dans le but de "*diviser*", alors que la collaboration exige la "*participation*". Dans cette optique, la coopération est basée sur la *répartition des tâches* et des *responsabilités* entre les membres du groupe. Chaque membre est responsable de prendre une action, de réaliser une opération ou d'accomplir une sous-tâche [Henri et al. 01].

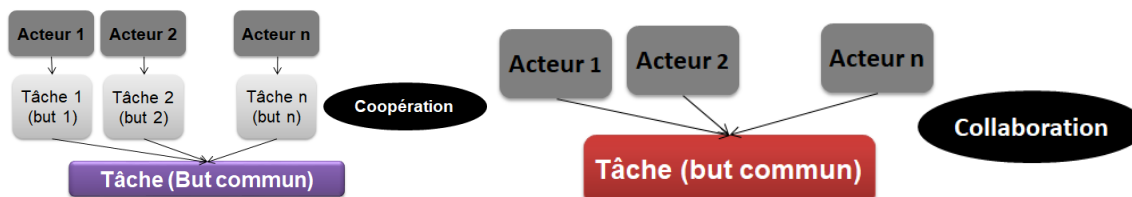


Figure 1. 5 : Collaboration vs Coopération.

Partant de ces éclaircissements portés sur la différence entre la notion de collaboration et coopération, nous pouvons considérer l'apprentissage collectif comme étant une activité qui intervienne pour faire participer un groupe d'apprenants, les rassembler autour de l'apprentissage d'un contenu pédagogique donné, les conduire à développer des savoir-faire. En effet, les connaissances aussi bien individuelles que collectives se construisent à la fois par les efforts personnels des apprenants ainsi que par les interactions, les échanges et le partage des informations entre les participants.

En fait, le but partagé de l'apprentissage collectif (collaboratif/coopératif) est également la construction de la connaissance individuelle et collective. Tous les efforts, actions et activités sont mis en place afin de conduire le groupe à ce but. C'est le groupe, en tant qu'entité, qui atteint le but partagé ; C'est lui qui accomplit la tâche à partir de laquelle se crée une production collective. Par conséquent, chaque apprenant participe à la réalisation du but partagé, en apportant une contribution adaptée à l'apprentissage collectif.

Par ailleurs, les membres du groupe, dans un contexte de collaboration, établissent un but commun. Cependant, par ses activités, les membres du groupe contribuent à la réalisation de ce but, de manière individuelle. Chaque membre vise son propre but en utilisant ses propres moyens, ce qui constitue le véritable consensus du groupe. Il en résultera avec de nombreuses productions, une production collective et les productions individuelles des apprenants.

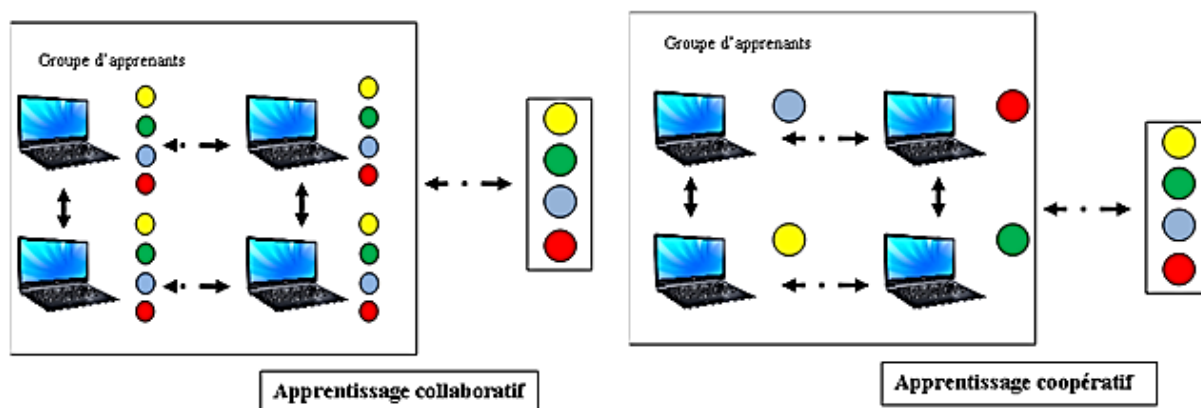


Figure 1. 6 : Différence entre apprentissage collaboratif/coopératif [Lafifi 07].

En revanche, la distinction entre "coopératif" et "collaboratif" est parfois estompée, surtout que les deux peuvent, dans certaines situations, ressembler. Néanmoins, comme il est bien illustré sur la figure 1.6, la **collaboration** se caractérise par des relations plus égalitaires entre les acteurs : apprenants, formateurs, concepteurs de formation et gestionnaires. Elle encourage un processus plus dynamique que la coopération, en permettant aux apprenants de se prendre en charge dans un environnement ouvert et solidaire. En fonction de leur motivation, de leur responsabilité et de leur autonomie, les apprenants, tout en participant à des activités de groupe, sont capables de faire des choix en fonction de leur parcours et d'utiliser les stratégies d'apprentissage qui leur conviennent le mieux. Pour favoriser l'apprentissage, la collaboration se concentre sur les tâches à accomplir par l'apprenant et les membres du groupe, par contre la coopération suggère à l'apprenant de réaliser une sous-tâche qui permet au groupe d'accomplir le travail demandé. Cependant, rien n'empêche un groupe collaboratif d'exécuter une tâche de manière coopérative s'il le désire. Dans une approche collaborative, la tâche coopérative n'est donc pas totalement exclue dans la mesure où elle est considérée comme un choix imposé par le groupe, mais elle peut avoir lieu si elle est vue comme pertinente, notamment pour atteindre efficacement le but partagé [Henri et al. 01]. La table 1.1 décerne clairement l'aspect comparatif entre l'apprentissage collaboratif et l'apprentissage coopératif, et ce selon plusieurs critères.

Critères distinctifs	Apprentissage coopératif	Apprentissage collaboratif
Échanges Interactions	Structurés (Principe d'interdépendance)	Non structurés (Partage, mise en commun des savoirs)
Contrôle de l'enseignant	Réel (Observation des groupes)	Faible (Autonomie des élèves)

Responsabilisation des élèves	Garantie par l'interdépendance	Incertaine (À la discrétion de chacun)
Équité entre élèves	Impossible (Caractère hétérogène des groupes)	Improbable (Organisation libre des groupes)
Rôles tenus par les élèves	Risque de spécialisation	Risque d'émiettement
Apprentissages visés	Savoirs fondamentaux liés aux différentes activités scolaires	Savoirs non fondamentaux : esprit critique, raisonnement, découverte collective.

Tableau 01.1: Tableau comparatif entre l'apprentissage coopératif et collaboratif.

1.7.2.3 Apprentissage par projet

L'apprentissage par projet (en anglais : *project-based learning*), comme son nom l'indique, consiste à apprendre des connaissances en réalisant un projet qui est en relation étroite avec le contexte dans lequel il s'inscrit. La naissance d'un tel apprentissage remonte au philosophe *John Dewey*. C'est le terme de « projet » qui doit être clarifié ici. Selon *Dewey* un projet signifie une réalisation, par les membres d'un petit groupe, d'un ensemble de tâches, en divisant le travail. Les apprenants contribuent à la réalisation des projets, qu'ils s'exécuteront suivant une division bien déterminée du travail. Cette division est caractérisée par une rotation fréquente dans l'affectation des rôles, notamment les rôles de responsabilité [Dewey 68].

Selon *Capra & Arpin* (2002), l'apprentissage par projet s'inscrit dans la théorie socioconstructiviste dans laquelle les apprenants s'engagent profondément dans le processus de construction de la connaissance à travers l'interaction avec l'environnement. De plus, l'enseignant n'est qu'un simple médiateur entre le savoir et l'apprenant [Capra et Arpin 02]. Pour *Proulx* (2004), l'apprenant mobilise, de manière contenue, ses connaissances antérieurement acquises afin de réaliser le projet en cours. Ainsi, les connaissances apprises à travers le présent projet peuvent éventuellement être appliquées dans d'autres situations. D'autre part, *Proulx* met exergue le résultat de l'apprentissage, la démarche et le temps du déroulement du projet [Proulx 04].

Comme il est illustré sur la figure 1.7, l'apprentissage par projet se base également sur une problématique qui provienne de la réalité [Lebrun 07], surtout une problématique intéressante pour la société et susceptible de provoquer la curiosité et la motivation de la part des apprenants. En fait, *Abdallah* (2009) a souligné aussi qu'une problématique ayant un caractère de curiosité permet de créer une certaine volonté chez les apprenants pour acquérir les compétences et les connaissances visées par le projet. Ensuite, les apprenants

contribuent à la réalisation concrète du projet, avec des rôles bien définis, en s'appuyant principalement sur leur souhait et désir. Cette idée est aussi appuyée par *Huber* (2005) qui a souligné qu'un projet doit être en liaison étroite avec un produit qui ne doit pas se limiter au produit relative à la classe, mais doit concerner même les produits hors la classe. Les projets impliquent les apprenants dans des rôles interactifs (résolveur de problèmes, décideur, chercheur, etc.). Contrairement à la méthode *APP*, où l'acquisition de connaissances nécessite la résolution d'un problème unique dans une durée déterminée, l'apprentissage par projet peut être interdisciplinaire et relativement long [Larmer 14].

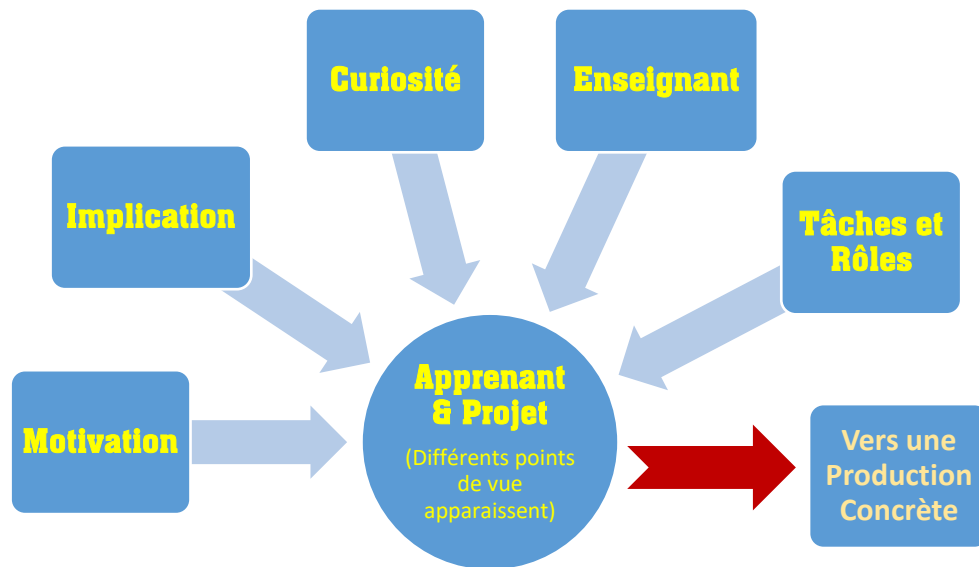


Figure 1. 7 : Critères favorisant l'apprentissage par projet [Lebrun 07].

Par ailleurs, la figure 1.8 illustre clairement les phases essentielles pour la réalisation d'un projet pédagogique. Cette illustration est tirée des travaux de *Thien* (2014). De même, *Schlemminger* (2003) a résumé les étapes de la réalisation d'un projet comme suit :

- Le choix et la définition du (des) projet(s) à réaliser font l'objet d'un débat collectif ;
- La répartition des tâches et des rôles.
- La planification de l'exécution des tâches dans le temps et dans l'espace ;
- L'évaluation des besoins en termes d'outils, de ressources et de moyens ;
- Des échanges fréquents sur la progression des travaux ;
- La présentation des résultats obtenus intermédiaires et finaux ;
- La discussion et les critiques en cas de modification et d'amélioration de ces derniers ;
- La diffusion des travaux finis sous différentes formes.

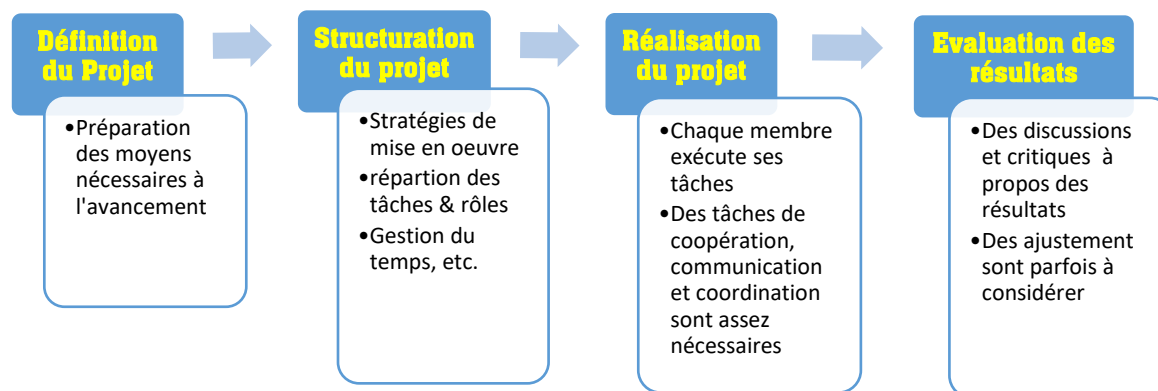


Figure 1. 8 : Processus de réalisation d'un projet [Thien 14].

Selon Grappin (2004), l'apprentissage par projet favorise largement la construction de la connaissance dans l'action ainsi que cette construction se réalise par les apprenants eux-mêmes. De plus, ces connaissances servent à donner des réponses aux interrogations qui peuvent être posées par les apprenants afin d'assurer l'avancement du projet.

Un projet pédagogique implique souvent un groupe d'apprenants, qui à leur tour sont appelés à la co-construction de la connaissance. En effet, des nombreuses activités liées aux phénomènes de collaboration, de coopération et de coordination sont à considérer lors de la réalisation d'un projet [Proulx 04] [Raby et al. 07].

Néanmoins, comme toute tendance, l'apprentissage par projet présente plusieurs inconvénients entre autres : la gestion complexe du temps de réalisation du projet [Bordallo et Ginestet 93]. Cet inconvénient a été souligné aussi par Proulx (2004) qui a trouvé qu'une telle méthode d'apprentissage nécessite beaucoup de temps ainsi qu'un grand effort mental et physique de la part des participants (enseignant & apprenants). Il est utile de noter que cette méthode est trop liée à l'établissement et l'environnement, notamment en termes d'approvisionnement des moyens nécessaires pour l'avancement du projet. Cela peut engendrer parfois des confusions dans les objectifs de l'apprentissage.

1.7.2.4 Apprentissage par problème

Les stratégies de résolution des problèmes sont récemment devenues un sujet populaire dans la recherche en éducation [Chiang 17]. Parmi ces stratégies, l'apprentissage basé sur la résolution de problèmes (en anglais : *Problem-Based Learning*) qui peut faire l'objet d'une méthode pédagogique favorisant l'apprentissage de la *programmation informatique*. Cette méthode d'apprentissage par problème (APP) a été présentée pour la première fois à la faculté de médecine de l'université *McMaster* au Canada dès 1970. Dès lors, les apprenants, qui sont structurés en équipes de travail, travaillent ensemble pour

résoudre un problème soulevé par un enseignant et pour lequel les apprenants ne suivent aucune formation particulière. À partir de cette problématique, le processus d'apprentissage se déroule de manière à laisser les apprenants obtenir des connaissances et des savoir-faire, à découvrir de nouveaux concepts de manière dynamique. D'autre part, les caractéristiques essentielles de cette méthode pédagogique peuvent être bien éclairées en rappelant la définition proposée par *Atmani & Stainier (2000)* : L'APP transfère la charge d'apprentissage du formateur (enseignant) à l'apprenant. Le processus d'apprentissage s'entame par la définition d'un problème. Les apprenants sont ensuite invités à discuter entre eux sur le sujet de ce problème. Ils soulèvent parfois des questions qui peuvent éventuellement être converties en thématiques ciblées par l'apprentissage. L'étude du problème provoque souvent la réactivation des prérequis (connaissances antérieurement acquises) ainsi que les apprenants sont motivés à chercher des réponses à leurs propres objectifs d'apprentissage. Dans ce cas, le savoir non seulement est mieux retenue, mais aussi plus aisément retrouvé lors de la résolution des problèmes pratiques provenus de la réalité.

En outre, l'apprentissage coopératif ou collaboratif (AC) est l'une des principales stratégies pédagogiques par lesquelles les apprenants peuvent améliorer les compétences en matière de résolution de problèmes [Ferradji et Hedjazi 17].

L'APP considère souvent des étapes d'auto-apprentissage pour acquérir les connaissances nécessaires à la résolution du problème et des étapes d'interaction collective pour valider ces connaissances. L'objectif principal de cette méthode pédagogique est la structuration et la construction de connaissances en résolvant des problèmes soulevés dans des contextes précis. Un tuteur, dans ce cas, joue un rôle d'un facilitateur pour orienter le processus de la résolution du problème et en conséquence rendre l'apprentissage plus efficace [Savery 06]. L'APP est fondé sur des scénarios adaptés à la réalité, dans lesquels des rôles bien déterminés sont attribués aux apprenants comme le modérateur qui gère les discussions.

Il est utile de préciser que l'acte de programmer est une tâche intellectuelle très complexe, du fait qu'elle fait recours à tout un ensemble d'expertises [Chaguiboff 85]. Pour construire de telles expertises, il nécessaire d'apprendre les fondements de la programmation informatique. L'apprentissage de la programmation informatique n'est pas forcément l'apprentissage d'un langage de programmation donné, mais plutôt l'apprentissage des techniques permettant de doter l'ordinateur par des programmes capables de faire des traitements à des fins de résolution de problèmes. Apprendre à coder ou à développer des programmes informatiques s'inscrit bien dans les tendances actives, notamment grâce à sa méthode d'APP. Les relations entre les problèmes posés

pour apprendre le codage des programmes ont été, depuis plusieurs années, soulignées par plusieurs chercheurs comme [Michard 82] [Maureen et Mildred 81] [Mayer 81]. Nous ciblons dans le cadre de cette thèse, la problématique permettant de concevoir et mettre en œuvre une nouvelle approche basée ACAO afin d'assister le processus d'apprentissage de la programmation, en utilisant la tendance active basée sur l'APP. L'adoption de la méthode APP pour la mise en œuvre de notre solution *Groupware* peut être motivée par les avantages que nous présenterons dans la section suivante.

1.7.2.4.1 Avantages de l'APP

La Méthode APP autorise les apprenants à avoir des aptitudes qui ne se limitent pas au niveau cognitif en leur donnant la possibilité d'acquérir des compétences métacognitives et sociales [Gholami et al. 16]. Nous essayons dans les points suivants de décrire les avantages apportés par une telle méthode d'enseignement, sur la base d'une étude réalisée par Dolmans et Schmidt "*Les apprenants apprennent mieux et rappellent davantage les connaissances acquises à cause de l'activation des anciennes connaissances, l'élaboration et l'apprentissage contextuel*" [Dolmans et Schmidt 96].

- L'APP offre aux apprenants la possibilité d'avoir des connaissances solides à travers l'activation et l'utilisation des connaissances théoriques dans la résolution des problèmes dans la vie professionnelle.
- L'APP favorise les compétences de l'*auto-apprentissage*.
- L'APP favorise les *compétences sociales*.

1.8 Conclusion

Le progrès considérable, qu'a connu le monde actuellement en termes de Technologies de l'Information et de la Communication (TIC), a incité la communauté académique de pencher vers l'adoption de l'apprentissage à distance. Ce dernier se distinguait de la méthode traditionnelle par une séparation physique entre les membres actifs dans le processus d'apprentissage. En effet, les participants peuvent, indépendamment de leurs localisations géographiques, collaborer, coopérer et interagir, à tout moment, autour d'un but commun d'apprentissage d'une discipline donnée. Ces interactions collaboratives peuvent s'effectuer même en temps réel. Néanmoins, plusieurs problèmes peuvent être soulevés afin de trouver des solutions informatiques assistant de telles activités. Ces problèmes sont principalement traités par un domaine de recherche appelé « ACAO : *Apprentissage Collaboratif Assisté par Ordinateur en anglais CSCL : Computer Supported Collaboratif Learning* » [Lafifi 07]. Les détails de ce domaine de recherche seront bien discutés dans le chapitre suivant.

Chapitre 2 : Apprentissage collaboratif assisté par ordinateur

2.1. Introduction

Le champ de Travail Coopératif Assisté par Ordinateur (*TCAO* ou *CSCW* : *Computer Supported Cooperative Work*) [Bannon et al. 91] est très vaste. Ainsi, il est effectivement possible de faire réunir un groupe de participants humains autour d'un travail partagé en utilisant des outils informatiques d'assistance. L'Apprentissage Collaboratif Assisté par Ordinateur (*ACAO*) est un sous-axe du grand domaine *TCAO*. Il analyse une variété de pratiques pédagogiques dans lesquelles l'interaction à distance entre pairs représente l'élément clé et le plus important de l'apprentissage [Dillenbourg et al. 09]. Le rôle principal de cette interaction ne se limite pas à assurer une simple communication entre les apprenants, mais plutôt d'inaugurer une situation dans laquelle les participants (*apprenants & enseignants*) construisent de nouvelles connaissances qu'ils n'avaient absolument pas préalablement [Cress et al. 15]. Par ailleurs, les préoccupations du domaine de recherche *ACAO* ont été résumées par George (2001) dont il a indiqué qu'elles tournent principalement autour de quatre problématiques : l'apprentissage humain, l'apprentissage collaboratif, le soutien à l'apprentissage collaboratif et l'utilisation de l'outil informatique pour assurer ce soutien.

Dans ce chapitre, nous présentons un état de l'art sur les environnements d'ACAO. En premier lieu, nous présenterons, de manière détaillée, le principe de l'enseignement Assisté par Ordinateur (EAO) suivi par la présentation des environnements dédiés à l'Enseignement Interactifs Assisté par Ordinateur (EIAO). Puis, nous présenterons les principaux fondements des environnements interactifs d'enseignement à distance. Ensuite, nous discuterons quelques détails du domaine TCAO qui représente le point de départ faisant émerger le domaine ACAO. Enfin, nous terminerons le chapitre par la présentation des spécificités du domaine ACAO ainsi que quelques environnements qu'ils assistent.

2.2. Enseignement assisté par ordinateur

L'Enseignement Assisté par Ordinateur (EAO) (en anglais, *Computer Based Training (CBT)*) est un sous-domaine d'informatique, qui tient en compte de la problématique de l'apprentissage des personnes afin de leur proposer des outils logiciels d'assistance. En jetant un coup d'œil dans la littérature, nous pouvons nous référer à plusieurs fondateurs de ce sous-domaine comme *Skinner*, *Crowder*, et *Jean Piaget*. Sans oublier *Seymour Papert* qui représente l'un des chercheurs contemporains qui ont améliorés EAO, notamment à travers la proposition du langage *Logo*.

Les premières tentatives d'enseignement en utilisant des machines apparaissent dans les années 1950. Dans cette époque, les premières machines ne permettent qu'une automatisation des cours sur un support papier. Ce type d'enseignement est également basé sur le modèle de la psychologie comportementale de *Skinner* [Skinner 54]. Le principe d'un tel enseignement se base sur le découpage du contenu pédagogique en plusieurs petites unités d'enseignement ainsi que le processus d'apprentissage est assuré de façon graduelle (par étapes). Suite à une simple évaluation des requis de l'unité précédente, le système propose à l'apprenant un contenu pédagogique d'une unité d'enseignement. Cette évaluation est basée sur la proposition d'un questionnaire composé d'une série de questions dont l'apprenant doit répondre.

C'est à partir des années 80 que les bases de l'EAO ont été publiées, surtout avec la naissance des premières machines personnelles. En effet, des applications éducatives, autrement dits didacticiels, ont joué un rôle primordial pour faire améliorer l'enseignement [Benadi 04].

Bien que les didacticiels aient apporté un plus au domaine de l'enseignement, ils restent encore incapables de résoudre certains problèmes qu'ils posent eux-mêmes à savoir : l'organisation et la gestion du dialogue apprenant-machine [Lafifi 07]. De plus, la gestion des didacticiels est figée de manière qu'ils ne donnent pas à l'enseignant assez de

flexibilité pour améliorer son contenu pédagogique. Les interactions apprenant-machine sont aussi fixées au moment du développement de l'application et toute amélioration exige un redéveloppement de tout le système [Boufaïda 95]. C'est pour ces raisons qu'un nouveau type d'environnement a vu le jour. Il s'agit des environnements *EIAO* que nous présenterons dans la section suivante.

2.3. Enseignement Intelligemment Assisté par Ordinateur

Dans le but d'éliminer certains problèmes issus des systèmes d'*EAO*, l'intelligence artificielle (*IA*) les sciences cognitives ont joués un rôle primordial, notamment pour rendre les logiciels d'enseignement plus intelligents et plus efficaces. Ceci a donné naissance de l'axe de recherche baptisé L'Enseignement Intelligemment Assisté par Ordinateur (*EIAO*) (en anglais *ICAI : Intelligent Computer Aided Instruction*) au début des années 80. Il s'agit d'un axe de recherche convergeant plusieurs disciplines comme : l'*IA*, l'éducation et la psychologie cognitive. Par conséquent, une variété de systèmes a été proposée pour caractériser les outils logiciels qui en sont issus. Parmi ces systèmes, nous citons : Système Tuteur Intelligent (*STI*) (en anglais *ITS : Intelligent Tutoring System*) ou encore Environnement d'Apprentissage Intelligent (en anglais *ILE : Intelligent Learning Environment*) [Dillenbourg 99].

Les *STI* marquent le passage de l'*EAO* à l'*EIAO*. Pour s'adapter à l'apprenant en situation d'apprentissage, le modèle général des *STI* est fondé sur une triple compétence : celle du domaine à enseigner (expert en matière), celle de l'enseignement (contenu pédagogique) et celle des compétences et connaissances de l'étudiant (modèle de l'apprenant). La considération de tous ces facteurs reste délicate et les résultats de ces systèmes ne sont pas toujours à la mesure des exigences. Ces approches se sont focalisées sur le processus d'enseignement et les outils informatiques essayent principalement à jouer le rôle de l'enseignant. Une deuxième stratégie a été développée, en même temps, dans laquelle on considère la machine comme un moyen pour permettre l'apprentissage par découverte, tout en offrant un certain degré de liberté à l'apprenant. Ce courant est basé sur les théories constructivistes de l'apprentissage [Piaget 35]. L'objectif est de permettre aux apprenants de réaliser des activités créatives. La notion de *micro-monde* émerge alors, l'idée sous-jacente étant de créer des environnements dans lesquels les apprenants agissent sur des objets issus d'un environnement plus simplifié pour pouvoir manipuler des concepts abstraits et construire leurs propres connaissances. Cette approche permet à l'apprenant de faire la transition entre le concret et l'abstrait. Le représentant le plus célèbre de l'approche du *micro-monde* est certainement *LOGO* [Papert 81].

Par ailleurs, l'évolution des interfaces utilisateurs graphiques a énormément augmenté les opportunités d'interaction en améliorant la maniabilité des environnements. La désagrégation de ces environnements d'apprentissage ouverts est principalement liée à leur absence de soutien à l'apprenant dans sa démarche. Si l'apprenant rencontrait des obstacles pour atteindre son objectif, le système ne pourrait pas l'aider.

2.4. Environnements Interactifs d'Apprentissage Assisté par Ordinateur

Les tuteurs intelligents et les environnements d'apprentissage ouverts apparaissent parfois aux extrémités d'un processus continu qui évalue le degré de dynamisme du système. Entre ces extrêmes, nous constatons des environnements qui cherchent à combiner l'apprentissage par découverte avec l'apprentissage par problème, c'est-à-dire qui visent un compromis entre une liberté totale pour l'apprenant et un contrôle strict de son activité. Au début des années 1990, des environnements interactifs d'apprentissage assisté par ordinateur ont vu le jour [Baron et al. 91]. En réalité, la signification du terme "*interactif*" est importante, car elle couvre d'une part les activités qui permettent à l'apprenant de participer et, d'autre part, son aptitude à intervenir de manière pertinente et à s'adapter à ses besoins [Baron et al. 91]. Le facteur décisif est l'interaction entre l'apprenant et son environnement dans lequel il réalise ses activités d'apprentissage. On constate aussi la disparition du terme "*enseignement*", remplacé par "*apprentissage*". Ce changement révèle qu'il y a moins d'intérêt pour le transfert de connaissances et plus pour la construction de connaissances par l'apprenant. Enfin, le terme "*par ordinateur*" devient "*avec ordinateur*" et met ainsi l'accent sur le fait que la machine est perçue comme un support permettant aux apprenants de poursuivre leur apprentissage.

2.5. Environnements Interactifs pour l'Apprentissage à Distance

Nous constatons une évolution des *EIAO* depuis de nombreuses années vers des environnements permettant la communication et l'interaction entre les machines et les apprenants géographiquement réparties. Dans ce sens, un nouveau terme est apparu à partir de la fin des années 90, il s'agit de l'Environnement informatique pour l'apprentissage humain (*EIAH*) [Balacheff et al. 97]. Cet acronyme permet de prendre en compte l'évolution récente *TIC* et de se rapprocher de l'apprentissage à distance. Le terme *EIAH* couvre à la fois *EIAO* et les *EIAO* favorisant l'enseignement à distance, que nous appelons environnements interactifs pour l'apprentissage à distance (*EIAD*).

2.6. Formation à Distance

Il y'a plusieurs années depuis que l'enseignement à distance a fait son apparition dans de nombreux systèmes éducatifs nationaux. L'accélération du progrès technologique et le besoin croissant de formation continue permettent de prévoir que la Formation à Distance (*FAD*) se renforcera encore sur les prochaines années. Nous présentons tout d'abord une clarification de certaines appellations que nous trouvons souvent dans littérature de ce domaine. Premièrement, un Enseignement à Distance (*EAD*) est le concept le plus utilisé, qui vise à établir une situation éducative permettant la transmission de connaissances en dehors d'un contact direct entre l'enseignant et l'apprenant, c'est-à-dire en dehors d'une situation de coprésence (en anglais *face to face*). Ensuite, le terme Formation à Distance (*FAD*) a été mentionné de manière générale dans plusieurs références. Il intègre les deux processus éducatifs : l'*EAD* (du point de vue de l'enseignant) et l'Apprentissage à Distance (du point de vue de l'apprenant).

Les premiers témoignages de *FAD* remontent au milieu du XIXe siècle. En 1840, *Isaac Pitman* a créé le premier cours par correspondance en Angleterre [Marot et Darnige 96]. Ce nouveau mode d'enseignement par correspondance se développait alors dans de nombreux pays. Son but était déjà de faciliter l'accès au savoir à tous ceux qui ne pouvaient pas aller à l'école en raison d'un handicap ou de l'isolement géographique. Sa croissance s'est accompagnée du développement du réseau des bureaux de poste : la correspondance des cours, des tutorats et des devoirs a permis ainsi de faciliter l'accès à la connaissance.

Les principales raisons qui conduisent les personnes à choisir la formation à distance sont diverses, mais désormais bien définies [Perriault 96] :

- *pour obtenir une compétence pendant qu'ils travaillent ou habitent chez eux ;*
- *pour se cultiver en vue d'obtenir une valeur ajoutée dans les échanges professionnels.*
- *avoir du temps, de la liberté, de la flexibilité.*

Le résultat obtenu ne correspond pourtant pas toujours à ce que nous espérons. En effet, le pourcentage de réussite aux examens est à environ 30 à 40 %. Ce taux de réussite assez bas s'explique notamment par la sous-estimation de la difficulté d'apprendre seule [Perriault 96].

Pour mieux satisfaire les besoins des apprenants et maximiser leurs possibilités de réussite, les institutions de *FAD* sont en évolution permanente. Les *TIC* sont au cœur des développements actuels et de nombreuses plateformes informatiques apparaissent pour l'*EAD*.

2.6.1. Plateformes de FAD

La technologie évolue de plus en plus vite : Internet, Intranet, réseaux à haut débit constituent les nouvelles technologies de l'information et de la communication. Ces outils offrent une véritable complémentarité organisationnelle et pédagogique à l'EAD en permettant aussi bien l'accès à distance à des contenus pédagogiques que les échanges d'informations entre les différents acteurs (apprenants & enseignants). Nous désignons sous le nom de Plateforme Informatique pour la Formation À Distance (*PIFAD*) les dispositifs d'apprentissage à distance utilisant les réseaux informatiques comme support. En d'autres termes, la plateforme est un support de diffusion et de gestion des connaissances, associant des contenus pédagogiques à des outils de communication, des outils d'apprentissage et d'évaluation. Les plateformes sont très nombreuses et offrent un choix très varié de services pour les usagers.

Les *PIFAD* visent la consultation à distance des contenus pédagogiques, l'individualisation de l'apprentissage et le tutorat à distance. Ces plateformes intègrent des outils d'assistance facilitant les tâches des acteurs (*enseignant, concepteur informatique, tuteur, apprenant et administrateur*) impliqués dans le processus de FAD. Le premier rôle est celui de l'enseignant qui est souvent responsable de la conception du contenu pédagogique. Le concepteur informatique, ou parfois l'enseignant lui-même, qui créent les médias (texte, image, vidéo, etc.). En outre, l'enseignant crée des trajectoires pédagogiques typiques bien adaptées aux objectifs de l'enseignement. Un autre rôle baptisé *Tuteur* est extrêmement important, notamment pour guider les apprenants dans leurs activités d'apprentissage. Il peut parfois ajuster leurs parcours pédagogiques pour rendre l'apprentissage plus efficace. L'apprenant consulte en ligne où télécharge le contenu pédagogique proposé, organise sa démarche, résout des exercices et transmet ses résultats au tuteur pour des fins d'évaluation. L'administrateur assure l'installation et la maintenance de la plateforme, l'inscription des apprenants, les droits d'accès aux ressources pédagogiques. L'administrateur doit comprendre qu'il s'agit d'un rôle spécifique à la préparation de la plateforme et non pas d'un rôle administratif habituel. La figure 2.1 illustre un modèle *PIFAD* avec cinq acteurs.

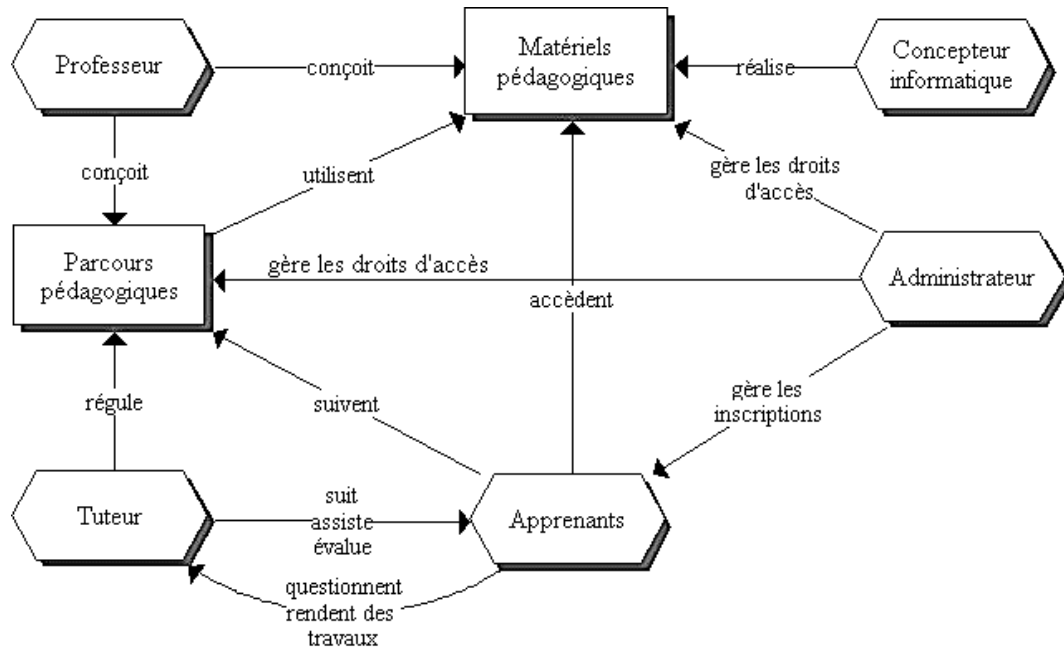


Figure 2. 1 : Schéma général d'une Plateforme de la FAD.

Les diverses plateformes prennent rarement en compte tous ces acteurs ou ne les séparent pas complètement. Par exemple, les rôles de l'enseignant et du concepteur informatique ou de l'enseignant et du tuteur sont parfois fusionnés. Malgré tout, nous prenons ce modèle avec cinq acteurs comme référence pour expliquer le principe général du PIFAD.

De nombreuses plateformes appuient leur modèle d'enseignement principalement sur des ressources pédagogiques (par exemple des pages *HTML* de cours) à mettre à la disposition des apprenants. Les activités d'apprentissage sont fréquemment réduites à la consultation des cours et à la réalisation d'exercices de type choix multiple. Parmi les plateformes indiquées dans la littérature, nous trouvons plusieurs plateformes comme *WebCT*, *Virtual-U*, *Learning Space*, *Librarian*, etc [Talhi 07].

Bien que ces plateformes jouent un rôle crucial pour améliorer considérablement le monde de l'enseignement et de formation à distance, elles restent encore insuffisantes, notamment pour supporter la collaboration, l'interaction et la conscience de groupe. C'est ainsi que les outils d'Apprentissage Collaboratif Assisté par Ordinateur (ACAO) ont vu leur naissance.

2.7. Apprentissage Collaboratif Assisté par Ordinateur

L'apprentissage collaboratif assisté par ordinateur (ACAO) ou en anglais *CSCL* pour désigner *Computer Supported Collaborative Learning* est devenu un domaine de recherche à part entière. Depuis plusieurs années que de nombreuses manifestations s'organise à travers le

monde pour créer un lieu d'échange des idées, des expériences et des résultats obtenus par les chercheurs. Ce domaine de recherche s'appuie également sur les fondements du grand domaine de recherche baptisé Travail Collaboratif Assisté par Ordinateur (TCAO) ou en anglais CSCW pour désigner *Computer Supported Collaborative Work*.

Nous pouvons apercevoir le terme CSCL ou ACAO comme une dérivation du terme CSCW ou TCAO. Ces deux domaines présentent clairement un nombre de points en commun et profitent de leurs contributions mutuelles à la compréhension et au soutien des interactions humaines. Toutefois, des particularités et des spécificités propres à chacun des domaines sont à distinguer. Nous essayons à travers cette section de mettre en exergue les caractéristiques communes et les différences.

2.7.1. TCAO

Le concept de TCAO a été initialement référencé par *Paul Cashman & Irene Greif* (1984), et ce dans le cadre d'une série de conférences financées par ACM ciblant l'étude de la nature du travail de groupe humain ainsi que la façon dont la technologie peut assister un tel travail.

Les technologies basées TCAO sont largement utilisées dans d'une multitude de contextes comme l'industrie, l'administration, l'enseignement, les jeux, le commerce, etc. Bien que ces contextes n'aient pas les mêmes objectifs, leurs groupes individuels impliqués dans les activités collectives sont presque régis par les mêmes règles. En effet, les *collecticiels* (ou *Groupwares*), qui représentent les logiciels de groupes doivent prendre en considération les spécificités de ces groupes. Ainsi, la réalisation par les membres du groupe de l'activité collective, à partir de leurs localisations géographiques en utilisant leurs machines de travail (*station de travail, portables, tablettes, smartphone, etc.*) constitue un autre point caractérisant les *Groupwares* [David et al. 03a] [David et al. 03b]. En considérant l'aspect collaboratif qui fait partie intégrante de la majorité de nos activités, le TCAO tient en compte de nouvelles dimensions de l'IHM telle que la conscience de groupe ou (en anglais : *Group Awareness*) [Dourish et Bellotti 92]. Ce dernier concept a été beaucoup abordé par de nombreux chercheurs travaillant dans les domaines du TCAO et de l'IHM, notamment pour étudier leurs conséquences sur l'efficacité des *Groupwares*. Dans cette perspective, certains travaux de recherche ont été proposés pour décerner l'apport de la conscience de groupe sur la qualité des *Groupwares* [Sohlenkamp 98][Gutwin et al. 96][Dourish et Bellotti 92]. D'autres travaux ont étudié les effets de la conscience de groupe sur l'activité collaborative [Weisband 02].

Le fruit de ce domaine de recherche est également la mise en place des applications de type *Groupwares*. Ces applications servent à implémenter les modules nécessaires pour permettre à un groupe de personnes, souvent éloignées, de travailler ensemble autour

d'un but partagé. Bien que le travail de groupe existe depuis l'aube de l'humanité, il présente un autre avantage lorsque les Groupwares peuvent apporter un gain de temps précieux, en déchargeant les membres du groupe des déplacements corporels et des tâches fatigantes liés à la diffusion de l'information en utilisant par exemple le téléphone, le fax, le courrier, etc.

2.7.2. ACAO

Apprendre, c'est donc le déroulement d'un processus d'apprentissage dans un contexte éducatif souvent régi, selon l'institution dans lequel se déroule, par des règles bien déterminées concrétisées par un contrat didactique. L'apprentissage est également un phénomène qui s'intéresse à la transmission du savoir en prenant en compte deux acteurs importants. Le premier acteur (apprenant), gagne lorsqu'il génère spontanément un certain nombre d'attitudes décrivant le savoir-faire transféré par le second acteur. Les activités d'apprentissage impliquent aussi des activités de collaboration, de coordinateur d'interaction de groupe, qui s'appuient notamment sur la diffusion, d'un acteur à une autre, de certaines connaissances. Le second acteur (enseignant) gagne lorsque le premier acteur gagne. Dans le chapitre précédent, la différenciation que nous avons évoquée entre l'apprentissage coopératif et l'apprentissage collaboratif justifie bien que les activités collaboratives conviennent mieux à l'apprenant qui sais preuve d'autonomie est qui est apte à s'engager volontairement dans un apprentissage collectif. Ainsi, doter des applications informatiques d'apprentissage par des mécanismes favorisant la collaboration constitue, sans aucun doute, un élément clé de la réussite de telles applications. C'est à travers de ces mécanismes que les apprenants peuvent trouver accomplir leurs activités aussi bien individuelles que collectives.

Par ailleurs, afin de bénéficier des avantages des *TIC*, les gens pensent, dans un premier temps, à mettre à la disposition des apprenants des contenus pédagogiques à étudier de manière autonome. Néanmoins, le sentiment d'isolement, la difficulté à conduire son propre apprentissage, le manque d'expérience, etc. amenaient à un taux d'abandon important. Mettre en place un enseignement collaboratif assisté par la technologie pour faire réunir un ensemble d'apprenants pour des fins d'apprentissage semble une stratégie plus qu'utile. Dans cette dialectique, un nouveau domaine de recherche baptisé *ACAO* qui dérive de *TCAO* a vu le jour.

La notion *ACAO*, souvent nommée en utilisant l'acronyme anglo-saxonne *CSCL*. Elle a été initialement référencée lors des travaux d'un *Workshop* lancé lors des journées d'une conférence internationale organisée en Italie en 1989 [Bannon 94]. Cette conférence a également choisi les environnements médiatisés comme cible de travail, particulièrement pour décerner leurs effets sur les interactions collaboratives des apprenants. L'*ACAO* représente l'un des domaines de recherche qui fait intervenir plusieurs disciplines totalement hétérogènes comme la psychologie, la sociologie, les sciences de l'éducation,

etc. L'objectif principal de ce domaine est de chercher à améliorer le domaine éducatif par la proposition de Groupwares favorisant l'apprentissage collaboratif à distance, en prenant en considération des aspects pédagogique et social du phénomène. Toutefois, notant ici qu'une multitude de travaux de recherche, dans le domaine d'ACAO, ne se placent pas uniquement dans un contexte de distance, mais se chargent aussi d'analyser les possibilités qui peuvent être offertes par les ordinateurs pour soutenir l'apprentissage collaboratif en classe [Ferradji 17]. En effet, des Groupwares assistant l'apprentissage collaboratif en coprésence (en classes) peuvent être proposés pour des fins d'éducation [Stahl et al. 06].

D'autre part, en s'appuyant sur le tétraèdre de *Faerber (2002)* décrivant un modèle pédagogique reposant sur les quatre pôles essentiels dans le processus d'ACAO, un Groupware d'apprentissage doit être en mesure de répondre aux besoins de ses utilisateurs, notamment en termes des rapports entre ces pôles. Comme il est illustré sur la figure 2.2, au centre du tétraèdre se trouvent les outils de médiation permettant de supporter les interactions et les échanges entre les différents pôles. Dans le cas d'ACAO, un Groupware d'apprentissage doit, entre autres, intégrer plusieurs outils pour assister les tâches inhérentes à l'interaction de groupe.

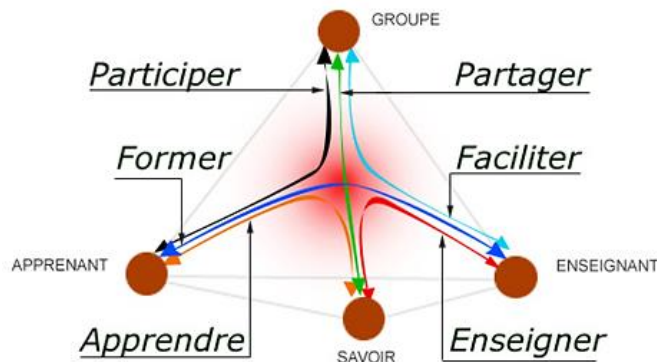


Figure 2. 2 : Tétraèdre pédagogique de Faerber (2002).

En effet, les acteurs (apprenants, enseignants) doivent être sensés de partager, de participer, de coordonner, etc. pour mener à bien leur apprentissage. De plus, les quatre pôles sont connectés par des liens spécifiant une variété de processus pédagogiques à savoir : le processus enseigner, former, apprendre, participer, faciliter, et partager. Le processus « enseigner » réponde également aux besoins de la pédagogie classique (voir section 1.7.1) qui est centré sur le contenu se base beaucoup sur les tâches de l'enseignant qui se charge de l'explication des savoirs en les mettant le mieux possible à la portée des apprenants. Le second processus appelé « former » favorise plutôt les aspects humains tels que les émotions dans les rapports de l'enseignant avec l'apprenant. Quant au processus « apprendre », il ressemble aux réflexions des apprenants face aux connaissances qu'ils doivent comprendre afin de les retenir. Le processus « participer » décrit les rapports

(communication, coordination, collaboration, etc.) que les apprenants produisent pour s'interagir. Le processus « *faciliter* » satisfait les besoins du groupe en termes de conduite de l'apprentissage et l'enseignant, dans ce cas, joue le rôle d'un tuteur. Le dernier processus est le processus « *partager* » qui correspond à un apprentissage mutuel au sein du groupe qui diffuse les connaissances acquises. Il utile de préciser que les travaux réalisés, dans le contexte de ce tétraèdre, ont été menés dans des contextes de téléformations déroulées au sein de l'université de Strasbourg en France.

Plusieurs formes d'apprentissage peuvent avoir lieu en combinant les six processus que nous venons de définir par exemple l'apprentissage par projet (ou par problème) qui sera ciblé sur l'apprenant et mettre en jeux les processus « *former* » et « *apprendre* ». L'apprentissage privilégiant la transmission de la connaissance sera ciblé le savoir et mettre en déroulement les processus « *apprendre* », « *enseigner* » et « *partager* ». Ainsi, L'apprentissage favorisant un travail collaboratif sera centré sur le groupe d'apprenants et il sera appuyé sur les processus « *participer* », « *faciliter* » et « *partager* ».

Partant de ce constat, nous pouvons dire que l'apprentissage peut être vu comme un processus regroupant un ensemble de moyens et d'activités qui transforme un individu entrant avec des prérequis en un individu sortant plus expérimenté. En mappant cette affirmation sur notre contexte, nous obtenons un schéma graphique décrivant le processus pédagogique de l'ACAO (figure 2.3).

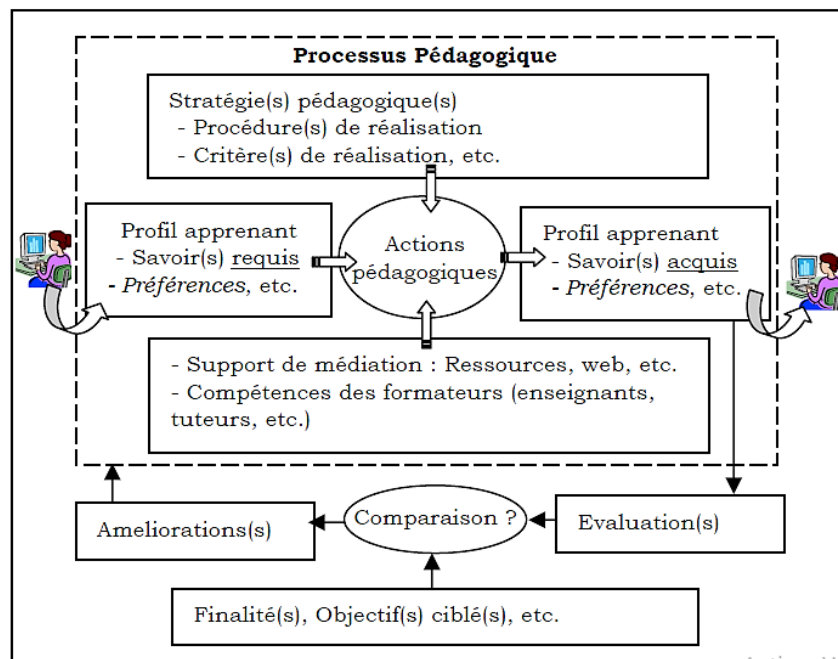


Figure 2. 3 : Schéma de fonctionnement du processus pédagogique d'ACAO [El Hassan et Lazrek 07].

2.8. Conclusion

À travers les sections qui composent ce chapitre, nous avons essayé de mettre en exergue le domaine de recherche d'*ACAO*, qui représente également une dérivation du grand domaine *TCAO*. D'autre part, une présentation de l'évolution qui conduit à la création de ce domaine semble assez importante. C'est pour cette raison que nous avons commencé par donner une brève explication aux différentes notions qui nous apparaissent connexes à ce domaine. De nombreuses notions ont été également discutées telles que : l'Enseignement Assisté Par Ordinateur, l'Enseignement Intelligemment Assisté par Ordinateur, les Environnements Interactifs d'Apprentissage Assisté par Ordinateur, les Environnements interactifs pour l'Apprentissage à Distance, la Formation à Distance, et le Travail Collaboratif Assisté par Ordinateur.

Le produit final des travaux de recherche de l'*ACAO* et *TCAO* est, bien entendu, le Groupware. Ce dernier n'est qu'un système informatique qui assiste un groupe de personnes engagées dans une tâche commune (ou but commun) et qui fournit une interface à un environnement partagé [Ellis 91]. Dans notre contexte, un Groupware doit prendre en considération les spécificités des activités collectives des apprenants et enseignants, particulièrement pendant le déroulement de l'apprentissage en leur proposant des alternatives technologiques. Par conséquent, ils doivent non seulement intégrer des fonctionnalités de production, mais aussi des fonctionnalités de communication, de coordination et d'interaction de groupe.

Par ailleurs, le champ d'application des Groupwares est actuellement très large parmi lequel nous distinguons le domaine éducatif. L'apprentissage de l'informatique, particulièrement l'apprentissage de la programmation informatique constitue l'un des champs d'application de cette technologie.

Chapitre 3 : Systèmes Collaboratifs dédiés à l'apprentissage de la programmation

3.1. Introduction

Dans le domaine de l'informatique, la programmation est une activité humaine qui vise à rédiger un ensemble d'instructions appelées code source en obéissant aux règles d'un langage de programmation. Elle implique généralement plusieurs tâches débutant par la formulation du problème et conception en terminant par la mise en marche du programme final. D'autre part, la programmation peut être aperçue selon de nombreux angles : l'angle technique qui tient en compte de l'écriture du code, l'angle conception qui s'intéresse aux cycles de développement et l'angle réflexion qui s'occupe de la pensée informatique. Le développement d'un projet informatique de grande taille requiert souvent la contribution de plusieurs acteurs comme les architectes des architectures logicielles, les développeurs de code source, les testeurs des fragments de codes, etc. La collaboration, le partage et l'interaction entre tous ces acteurs paraient utiles. En effet, l'apprentissage de la programmation informatique à des apprenants débutants soulève de nombreux problèmes éducatifs assez sérieux que ceux rencontrés lors de l'apprentissage des autres disciplines. La programmation informatique y semble à la fois comme une science, relevant d'un domaine scientifique avec ses notions propres, et comme un moyen pour concourir à résoudre des problèmes soulevés dans d'autres disciplines.

D'autre part, parmi ses caractéristiques essentielles, nous distinguons celle d'être en relation étroite avec de multiples pratiques professionnelles exigeant souvent des interactions collectives, qui conduisent entre autres à étudier plusieurs méthodes d'apprentissage. Par conséquent, l'apprentissage de la programmation avec une méthode adéquate, dès les premières séances, prépare bien les développeurs à être en mesure de travailler dans un contexte collectif. Dans cette perspective, l'apprentissage de cette discipline ne doit pas être fait avec uniquement le principe de la pédagogie classique, mais le recours à la pédagogie active semble plus qu'important. Parmi les méthodes d'une telle pédagogie, nous distinguons l'*APP* (voir section 1.7.2.4) qui constitue l'une des méthodes utilisées dans notre contribution. Nous avons ainsi combiné l'*APP* avec l'*ACAO* afin de proposer un nouveau Groupware pour assister les activités d'apprentissage de la programmation informatique. Dans cette perspective, nous avons organisé ce chapitre en deux parties : la première présente les spécificités de l'apprentissage de la programmation informatique, notamment dans son contexte classique, tandis que la deuxième partie s'intéresse à dresser un état de l'art sur les systèmes informatiques proposés pour assister l'apprentissage collaboratif de ladite programmation.

3.2. Programmation Informatique

La programmation informatique a largement attiré l'attention au cours de cette dernière décennie, non seulement en raison des développements technologiques rapides, mais aussi parce qu'elle est censée favoriser d'autres compétences, y compris la résolution de problèmes, la pensée logique et la créativité [Scherer et al. 20]. De plus, le secteur éducatif du monde entier est en train d'élaborer des programmes qui mettent en œuvre la programmation informatique sous forme d'un concept qui contextualise la programmation informatique et les compétences connexes en tant que forme de résolution de problèmes [Shute et al. 17], ou comme un thème indépendant ou intégré dans d'autres disciplines [Yadav et al. 17].

Dans le contexte éducatif, la programmation informatique est souvent confondue avec à la notion de codage ou écriture du code source. En informatique, ces deux notions sont différentes. La programmation comporte généralement plusieurs étapes débutant par l'analyser du problème suivi par la proposition d'un algorithme, puis la traduction, dans un langage de programmation, de l'algorithme, et enfin les tests et la mise en marche du programme. Quand on entend l'appellation *codage*, notre esprit va directement à l'écriture, dans un langage de programmation, d'une séquence d'instructions concrétisant ainsi un programme informatique. En effet, le codage est loin d'être un

processus de raisonnement d'un problème qu'une simple traduction d'un algorithme en un langage de programmation.

Dans cette dialectique, la programmation informatique peut être considérée comme une situation de résolution de problème et de codage. Il s'agit, par conséquent, à partir de l'énoncé d'un problème, concevoir et mettre en œuvre un programme capable d'y traiter. La première réflexion à exprimer est-ce que l'on distingue par résoudre un problème de programmation. Il s'agit donc d'un processus de passage d'un problème décrit par des entités du monde à un code source d'un programme exécutable par un ordinateur ou un autre dispositif de calcul. La figure 3.1 illustre clairement ce processus.

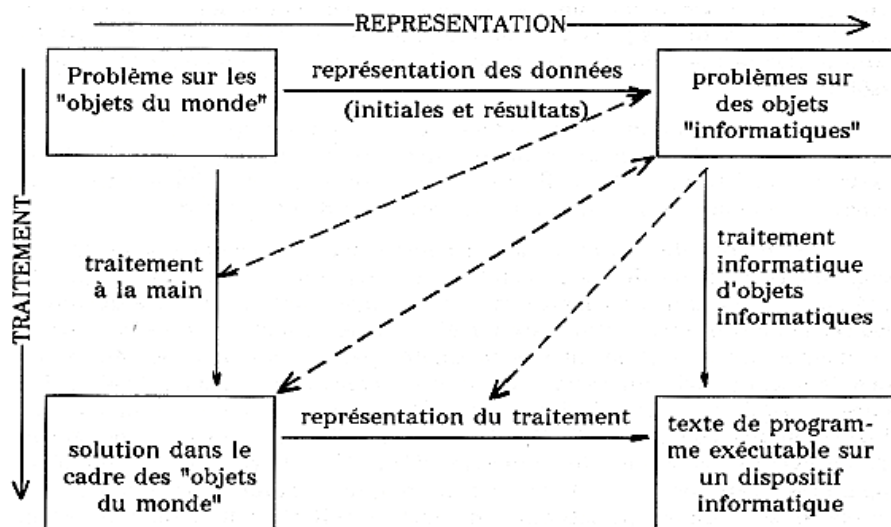


Figure 3. 1 : Processus de résolution de problèmes de la programmation : entités intervenantes [Rogalski 88].

3.2.1. Spécificités de la programmation informatique

La programmation informatique se distingue des autres disciplines par plusieurs spécificités que nous pouvons les regrouper principalement en deux grandes catégories de spécificités à savoir : les spécificités conceptuelles et les spécificités techniques.

Spécificités conceptuelles

La programmation informatique est un processus cyclique (figure 3.2.) qui se déroule selon une méthodologie de développement intégrant un nombre d'étapes que nous pouvons les synthétiser en trois grandes phases. La première phase, qui décrit les exigences (ou les besoins) sous forme d'un cahier de charge, permet d'intégrer les étapes qui précèdent le processus de la programmation. Elle sert à décrire le programme en s'axant prioritairement sur les besoins notamment fonctionnels ; Le terme spécifier c'est

« Désigner par son trait spécifique, exprimer, déterminer en particulier, en détail » c'est en opposition avec : « on va décrire en gros ». Quant au terme spécification, c'est « ce que doit faire le programme », ensemble de critères qui doivent satisfaire son fonctionnement interne et ses interactions avec son environnement. Ensuite, la phase de construction qui regroupe toutes les étapes de développement du programme depuis la conception, jusqu'au codage, etc. Enfin, la phase de test qui fait réunir toutes les activités liées à la vérification du programme. Ces trois phases sont étroitement liées et peuvent, en conséquence, s'influencer mutuellement. Toutefois, la phase de construction reste la plus essentielle pour pouvoir produire une application de qualité.

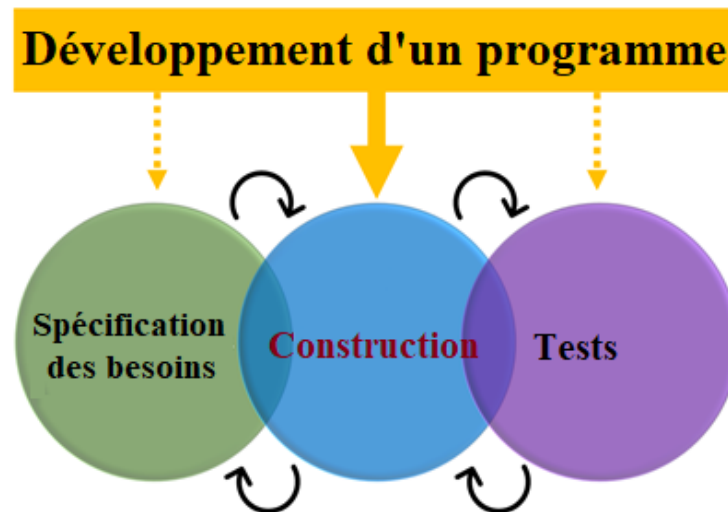


Figure 3. 2 : Processus de développement d'un programme

Dans ce processus, le programmeur joue un rôle très important, du fait que c'est lui qui doit déterminer ce qui se passe à l'intérieur du programme, en d'autres termes décrits quelle réaction le programme va adopter pour répondre aux actions de l'utilisateur. Plus particulièrement, dans la majorité des situations, l'activité de « programmer » est le fait d'offrir à l'utilisateur final du programme la possibilité de transformer ses données d'entrées en résultats de sortie. Afin d'assurer cette transformation, il est évident que le programmeur doit expliciter, par des instructions écrites dans un langage de programmation, la manière de transformation. En effet, le programmeur doit avoir ce qu'on appelle la *pensée computationnelle* (en anglais *Computational thinking*) qui se développe généralement par expérience. La pensée computationnelle est un concept qui fait référence à un ensemble d'habiletés, compétences et procédures facilitant la résolution de problèmes en se basant sur l'informatique théorique. Elle implique donc la résolution de problèmes, la conception de programmes et la compréhension du comportement humain [Wing 06].

D'autre part, une caractéristique spécifique à la programmation informatique réside dans l'exécution du programme résultant. Cette exécution se réalise dans des conditions distinctives par rapport à la phase de conception du programme. Premièrement, elle est effectuée, sans avoir intervenir le concepteur, dans un temps différé. Deuxièmement, les données d'exécution ne sont pas contrôlées par le concepteur. Néanmoins, beaucoup de langages de programmation offrent la possibilité d'exécuter le programme au fur et à mesure de rédaction du code. De plus, la réalisation d'un programme est fortement liée aux spécificités du langage de programmation ainsi qu'à l'outil informatique sur lequel s'exécute le code exécutable.

Par ailleurs, tout fragment de code (procédure, fonction, etc.) d'un programme doit être rédigé de façon séquentielle et le concepteur doit jouer un double rôle. D'un côté, il doit se représenter le fonctionnement du programme, et de l'autre côté, il doit se positionner à place de l'utilisateur dans l'exécution du programme dans son contexte d'exécution.

3.2.1.1. Spécificités techniques

Pratiquement, la programmation consiste en la résolution d'un problème de façon automatisée en s'appuyant sur un algorithme. Ce dernier n'est qu'un pseudocode indépendant des règles de tout langage de programmation. Un algorithme est une séquence d'instructions, décrites dans un langage proche du langage naturel, qui sont exécutées par le calculateur du dispositif informatique sur lequel s'exécute son programme exécutable correspondant. Le dispositif d'exécution ne peut exécuter le programme source sauf s'il est totalement traduit en langage machine. Cette traduction est également assurée par des compilateurs (Figure 3.3). Il y'a pour chaque langage de programmation un compilateur destiné à un type de plateforme (système d'exploitation & machine). Du moment que le programmeur ne rédige plus en langage machine (en binaire), les langages de programmation offrent une syntaxe plus proche de celle du langage naturel.

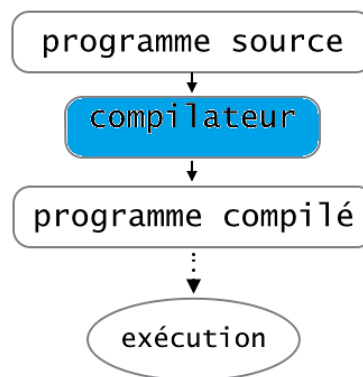


Figure 3. 3 : Traduction du code source en langage machine.

3.2.2. Apprentissage de la programmation informatique

L'apprentissage de l'informatique est devenu actuellement un phénomène de masse, notamment avec le progrès considérable qu'a connu le domaine des réseaux et d'internet. Ceci peut être justifié aussi par la quantité des offres d'emplois qu'a connu le monde actuellement, notamment en termes de développeurs en informatique. Il faut noter que l'apprentissage de l'informatique, notamment la programmation est considérée aussi comme une stratégie d'envergure mondiale, aux Etats-Unis par exemple le président a déclaré au début de l'année 2016 une décision stipulant que « les sciences informatiques pour tous » avec même un budget de 4 milliards de dollars pour l'éducation de l'informatique dans les écoles. En 2013, les décideurs de la *Royaume-Uni*, ont décidé de rendre obligatoire l'éducation de la programmation informatique à l'école, et ce à tous les niveaux scolaires. En France, par ailleurs, les améliorations apportées au domaine éducatif ont également saisi ce point par la prise de toutes les mesures nécessaires pour que l'élève soit mis en relation plus fusionnelle avec les dispositifs informatiques, les *IHM* et les applications informatiques [Ben Henda 17]. En *Algérie*, depuis plusieurs années, que le gouvernement a décidé d'incorporer l'enseignement de la matière informatique dans le secteur éducatif, notamment dans les niveaux moyen et secondaire.

3.2.2.1. Logiciels d'apprentissage de la programmation

De nombreuses méthodes pédagogiques ont été proposées et utilisées afin de guider l'apprentissage de la programmation. Chacune d'elles a ses spécificités en présentant des avantages et des inconvénients. Parmi ses méthodes nous distinguons celle qui se base l'utilisation de *l'hypertexte* et à *l'hypermédia*. Ces deux dernières technologies sont utilisées à des fins d'enseignement et ils l'assimilent continuellement, d'une part, à un tutoriel évolué [Leclercq et Denis 01] et d'autre part, à un outil intellectuel [Linard 01]. Afin de produire chez l'apprenant le sens et la pertinence, les outils logiciels à base d'hypertextes permettent d'établir des liens de toutes sortes entre les entités du monde réel et les entités de la programmation (ex. variables). Une autre tendance consiste à exploiter à des buts purement pédagogiques des technologies qui ne sont développées qu'à des fins commerciaux comme les logiciels de traitement de textes, de dessin, de présentation, de gestion de courrier électronique, etc. Ces logiciels peuvent jouer un rôle primordial, notamment pour assister le processus d'apprentissage de la programmation.

D'autre part, les méthodes les plus souvent utilisées s'appuient sur l'utilisation d'un langage de programmation préalablement maîtrisé. Ainsi, lors du déroulement de l'apprentissage, l'enseignant pourrait proposer des projets que les apprenants réalisent en utilisant ce langage maîtrisé ainsi que certains outils didacticiels. Nous soulignons ici qu'il n'y a absolument pas de théorie pédagogique spécifique derrière ces outils.

L'avantage de telle méthode, c'est qu'elle favorise mieux le travail individuel sans guidage et donc l'apprenant peut pénétrer l'ère de l'informatique de façon autonome.

Une méthode faisant appel à une théorie d'apprentissage s'est particulièrement développée en *Logo*, en adoptant notamment la théorie constructiviste. Selon le fondateur de *Logo* (voir section 2.3), qui est *Papert*, le dispositif informatique est considéré comme acteur principal dans le processus d'apprentissage de manière qu'il intervient comme un outil intellectuel afin de réfléchir avec l'apprenant. Dans cette optique, la connaissance n'est pas directement transmissible à l'apprenant, mais elle est déductive [Papert 81].

Dans le but d'alléger la charge cognitive de l'apprenant, plusieurs outils logiciels ont été développés. Certains de ces logiciels adoptent des représentations graphiques détachant ainsi l'apprenant, notamment l'apprenant novice de la complexité du code source en se focalisant sur la solution algorithmique. Nous citons parmi ces logiciels : *StarLogo The Next Generation* [Klopfer et Yoon 05], *Scratch* [Maloney et al. 04], *Alice2* [Kelleher et al. 02] et *Cleogo* [Cockburn et Bryant 98]. Ce type de logiciels est d'une importance extrême, du fait qu'ils incitent les apprenants à apprendre la logique de la programmation, sans qu'ils aient pour autant une connaissance formelle des notions. Une autre méthode consiste à utiliser la concurrence pour inciter des apprenants déjà expérimentés en programmation. Nous citons ici les deux logiciels *Robocode*¹ et *Robocup*², qui proposent aux apprenants des situations de programmation en utilisant des solutions dérivées de l'IA, et ce en vue d'orienter des robots dans des séances d'apprentissage bien organisées. D'autres logiciels, se basant sur les jeux vidéos, ont été développés pour assister des séances d'apprentissage de la programmation avancée. *Raclet* a proposé un nouveau protocole d'apprentissage de la programmation dans lequel les séquences de travail sont structurées en cycles : le premier cycle, l'apprenant développe de façon individuelle une partie de code d'un logiciel ensuite pour favoriser les interactions entre pairs une ou plusieurs phases de revue de code sont introduites. Une plateforme dite *elastic* est également utilisée [Raclet et al. 20].

Le choix, des outils d'assistance à l'apprentissage de la programmation, se fait généralement sur la base de certains critères parmi lesquels la favorisation d'apprentissage semble ignorante [Henry et Vandeput 18]. En effet, les gens (apprenants et enseignants) choisissent un logiciel, car son dépouillement est considéré comme formateur, car l'institution d'enseignement l'exige, car le secteur professionnel le préfère

¹ <https://robocode.sourceforge.io/>

² <http://www.robocup.org/>

et donc il y'a une absence totale de critères pédagogiques. Or, nous pensons que les activités collaboratives introduites par l'apprentissage en groupe peuvent être très utiles pour aussi bien pour l'apprenant que l'enseignant [Chorfi et al. 20]. C'est ainsi que nous avons inscrit les travaux de notre thèse dans ce contexte pour soutenir l'apprentissage de la programmation via une approche ACAO. En fait nous a avons développé un prototype d'outil *Groupware* assistant les activités d'apprentissage collaboratif de la programmation informatique. Ledit *Groupware* gère les interactions collaboratives produites lors du déroulement des sessions d'apprentissage d'un groupe d'apprenants géographiquement dispersés. Toutefois, avant d'entrer dans les détails de notre approche, une présentation de quelques outils collaboratifs proposés à des fins pédagogiques semble assez importante.

3.2.2.2. Quelques environnements similaires

Le développement, par un groupe d'apprenants, des codes sources partagés peut souvent être suggéré, notamment dans les situations d'apprentissage de la programmation à distance. C'est pourquoi la mise en place de systèmes qui permettent et soutiennent l'interaction de groupe semble très importante. La méthode pédagogique basée sur l'ACAO est une science dédiée à étudier la façon dont les individus peuvent apprendre ensemble en utilisant des ordinateurs et des réseaux [Stahl et al. 06]. Cette méthode exploite les TIC comme ressource principale d'interaction de groupe afin de permettre à des apprenants de partager et de construire des connaissances de manière collaborative. Dans le contexte de l'apprentissage de la programmation, la combinaison de l'interaction sociale avec les fonctionnalités de développement de programmes (édition, compilation, débogage et exécution de code) constitue une base importante sur laquelle l'ACAO est fondé [Serrano et al. 14]. Néanmoins, un sérieux problème peut être considéré à ce niveau, est celui de la manière de concevoir des mécanismes favorisant les activités relatives à L'ACAO et répondent aux besoins des apprenants en termes de développement de programmes. La première tendance est d'étendre les mécanismes d'ACAO en intégrant certaines fonctionnalités liées aux tâches de la programmation comme dans SICODE [Pérez et al. 06] et COLLEGE [Duque et Bravo 08]. Grâce à ces systèmes, les programmeurs communiquent autour d'un code partagé par le biais du chat.

Une autre tendance consiste à incorporer des fonctionnalités d'ACAO dans des outils professionnels d'édition de programmes basés sur l'EDI (*Environnement de Développement Intégré*) comme *Eclipse*. Dans cette perspective, de nombreuses tentatives ont été faites pour inclure des mécanismes de collaboration par le biais de plusieurs plugins à savoir : Jazz Sangam [Devide et al. 08], XPairtise [Pedersen et al. 09], Saros [Salinger et al. 10]. Olshed (*On-Line SHared EDiting*) [Allen et al. 02], est un autre outil qui se base sur EDI pour permettre

l'édition partagée de codes sources. Il s'agit d'une simple bibliothèque Java basée sur les swings pour l'édition partagée en ligne. Il s'avère facilement intégrable dans des applications Java déjà existantes ainsi qu'il offre de bonnes opportunités, notamment en termes d'éducation du langage Java.

Comme la synchronisation, dans l'espace de travail partagé, nécessite que tous les programmeurs aient dès le départ la même version du programme source dans l'outil d'édition partagé, Jazz Sangam intègre un mécanisme de contrôle de versions. En outre, une messagerie instantanée permettant la communication entre les programmeurs est également intégrée au système. Cependant, ce plugin souffre de plusieurs limitations comme l'absence de mécanismes de sensibilisation des membres du groupe ainsi que l'absence de supports de coordination particuliers. Saro permet aux programmeurs de participer simultanément au processus d'édition du code source et la gestion de la cohérence des copies des différents programmeurs est automatiquement assurée. Saros intègre, outre les mécanismes de transparence, des mécanismes de rétroaction de groupe et les programmeurs sont implicitement informés de la position des curseurs des autres participants dans le code source partagé.

COLLECE (COLLaborative Edition, Compilation and Execution of programs) est un système de type Groupware qui est proposé par *Bravo*, pour offrir de nombreuses fonctionnalités de programmation collaborative synchrone telles que l'édition partagée, la compilation et l'exécution de programmes. Ce système intègre plusieurs outils avancés, notamment pour la communication, la coordination et la sensibilisation du groupe (en anglais : *Group awareness*). L'expérimentation de *COLLECE* a démontré que les outils de transparence intégrés, qui fournissent des informations aux apprenants sur le déroulement des activités de l'apprentissage collaboratif, constituent un élément clé de la réussite de ce système. Elle a justifié aussi que lorsque les programmeurs répartis ont suffisamment d'expérience dans l'utilisation du Groupware, leurs tâches collaboratives de programmation produisent de meilleurs programmes que les programmeurs travaillant de manière individuelle [Bravo et al. 13].

Du fait que les applications basées sur web ont, depuis plusieurs années, démontré leurs efficacités, notamment avec l'avancement constaté ces derniers temps en termes des *TIC*, plusieurs applications web dédiées à l'édition partagée de codes sources ont été proposées. Nous pouvons mentionner *Collabode*, qui supporte l'édition collaborative en temps réel de codes sources. Il dispose en effet de nombreuses fonctions qui peuvent être extrêmement avantageuses pour les situations d'éducation. *Collabode* se caractérise aussi par la possibilité de pouvoir suivre en temps réel les changements de plusieurs programmeurs, et chaque membre peut interagir avec le système en utilisant différentes

interfaces graphiques utilisateur (*GUI*) liées à son rôle dans le travail collaboratif de programmation. Dans le contexte de l'éducation, les apprenants travaillent souvent avec leurs enseignants pour atteindre des objectifs pédagogiques, mais leurs rôles sont très asymétriques : enseignant structure le processus d'apprentissage et l'apprenant le met en pratique. L'effet des outils de *Collabode* sur l'apprentissage de la programmation a été discuté par les mêmes auteurs, qui ont prouvé qu'ils peuvent jouer un rôle important comme l'analyse de la progression de l'apprenant dans son environnement d'apprentissage de la programmation. Cela améliore le processus d'apprentissage en améliorant l'attention [Goldman et al. 11]. *WPASC* est l'acronyme de "*Web-based Programming Assisted System for Cooperation*", qui représente un environnement d'apprentissage collaboratif. Grâce à *WPASC*, les apprenants peuvent entrer, exécuter et déboguer des codes sources. Ils sont encouragés à trouver de l'assistance par les pairs. L'effet de *WPASC* sur les performances d'apprentissage a également été examiné par *Hwang* (2012). Les résultats de l'expérience ont permis de tirer une conclusion utile stipulant que la majorité des apprenants saisissent que l'activité d'apprentissage assistée par *WPASC* était utile pour l'apprentissage collaboratif de la programmation. En outre, les résultats ont démontré que l'activité de résolution de problèmes n'avait aucun effet sur l'apprentissage des apprenants [Hwang et al. 12]. Dans le même contexte, *Moreno* (2004) a proposé un environnement d'apprentissage collaboratif appelé *JeCo* (*Jeliot Collaboratively*) qui est une combinaison d'un système de visualisation de code Java (appelé *Jeliot 3*) et d'un système coopératif (appelé *Woven Stories*). *Jeliot 3* est un système d'apprentissage de langage *Java* offrant des cours d'initiations à la programmation. L'outil de rédaction collaborative en mode asynchrone est soutenu par le système *Woven Stories* pour permettre aux apprenants de développer et de partager des programmes sources. Un outil de chat est incorporé dans le système à des fins de conversation inter-apprenants donnant en conséquence des opportunités de collaboration synchrone. L'impact du système sur l'apprentissage de la programmation élémentaire a été étudié par *Ben-Ari* (2008). Les résultats de l'expérience ont démontré que [Ben-Ari 08]:

- Le système encourage l'apprentissage en améliorant l'attention des apprenants ainsi qu'il présente une véritable illustration des aspects dynamiques de l'enseignement ;
- Bien que de nombreux enseignants adoptent, de façon intensive, l'utilisation du système, d'autres observent le manque de contrôle qui peut, en conséquence, réduire ou parfois abandonner l'utilisation du système ;
- L'outil de visualisation proposé a un effet positif sur la participation des apprenants aux activités collaboratives d'apprentissage.

Eliph est un outil éducatif d'apprentissage de la programmation basé web. Il intègre un questionnaire de versions afin de permettre aux participants de consulter, à tout

moment, l'historique de développement du code source développé [Park et al. 17]. Le suivi d'un tel historique peut se faire par bloc d'instructions, notamment pour soulager la surcharge cognitive chez les apprenants en manipulant des programmes plus au moins importants. La principale caractéristique de ce système réside dans sa capacité à aider les enseignants dans leurs tâches d'évaluation de leurs apprenants en leur permettant, par exemple, de consulter les erreurs de compilation du code. *Eliph* a été évalué par une étude expérimentale dans le cadre d'enseignement des cours en informatique au profit des étudiants de la Licence. Les résultats de cette étude démontrent également les effets de l'outil d'affichage de l'historique du code sur la qualité du feedback des pairs, en permettant d'avoir une compréhension claire sur l'intention et la pensée computationnelle des apprenants (*voir section 3.2.1.1*) [Park et al. 17].

3.3. Conclusion

Dans ce chapitre, après avoir présenté les spécificités conceptuelles et techniques de la matière de programmation informatique, nous avons expliqué le principe de l'apprentissage de telle matière. Ensuite, nous avons dressé un état de l'art sur les outils logiciels proposés à des fins d'apprentissage de la programmation ou le développement de code informatique. Nous avons constaté suite à cette étude de l'état de l'art, que tous les outils logiciels proposés dans la littérature présentent une limite importante du fait qu'ils n'adoptent pas, dans leur principe de fonctionnement, une méthode pédagogique justifiant son efficacité dans l'apprentissage de la matière de programmation informatique par exemple la méthode d'apprentissage par problème ou par projet.

La méthode d'APP, qui est fondée sur les théories cognitives de l'apprentissage, implique un apprentissage en groupe. Cette méthode continue d'être une caractéristique essentielle des programmes d'enseignement supérieur à travers le monde [Rovers et al. 18]. L'apprentissage de la programmation informatique, qui est une matière déductive nécessitant beaucoup de réflexions, semble plus efficace si elle est guidée par la méthode pédagogique par problème [Kay et al. 20]. Cette idée a été encore justifiée par de nombreux chercheurs [Chis et al. 18][Bawamohiddin et Razali 14][Bawamohiddin et Razali 17][Peng 10][Bellström et Kilbrink 09][Nuutila et al. 08]. En effet, nous pensons que les méthodes d'APP et d'apprentissage par projet constituent les seuls moyens pour concrétiser l'apprentissage collaboratif de la programmation. Par ailleurs, l'APP offre plusieurs opportunités en termes de favorisation de l'auto-construction des connaissances ainsi qu'il relie le contenu de la matière à des problématiques de la vie réelle. De plus, une contribution permettant de combiner l'APP avec les TIC, nous apparaît importante et peut dégager plusieurs avantages, surtout en termes d'optimisation du coût d'apprentissage en supprimant les coûts de déplacement. En conséquence, nous pouvons assurer le

déroulement de l'apprentissage même dans des situations critiques comme celles causées par le coronavirus 2019 (*COVID 19*) que nous avons vécues récemment.

Partant de cette synthèse, nous pensons qu'une contribution qui s'appuie à la fois sur les fondements du domaine *d'ACAO* et les spécificités de la méthode pédagogique *d'APP*, semble assez utile et prometteuse. C'est ainsi que nous avons proposé un nouvel outil logiciel de type *Groupware* nommé *PBPCLG : Problem-Based Programming' Collaborative Learning Groupware*. La présentation détaillée des caractéristiques de ce *Groupware* est bien donnée au chapitre suivant.

Chapitre 4 : Présentation de l'environnement PBPCLG

4.1. Introduction

PBPCLG est prototype qui est fondamentalement proposé pour assister l'apprentissage collaboratif de programmation avec le langage *Java*. En utilisant un tel prototype, deux types de participants sont reconnus : les *tuteurs* et les *apprenants*. Un *tuteur*, qui est un facilitateur de l'apprentissage, agit dans le processus de construction des connaissances comme un guide pour rendre efficace l'apprentissage des apprenants. Il a un rôle extrêmement important, particulièrement lors de l'analyse des problèmes posés. Le *Groupware* proposé permet aux apprenants d'éditer, de manière collaborative, un code source résolvant un problème de programmation donné. À travers *PBPCLG*, une session d'apprentissage est définie par la définition de plusieurs éléments tels que la description d'un problème de programmation, les ressources pédagogiques requises, les objectifs de la séance d'apprentissage, la durée de la séance, etc. Ce chapitre s'intéressera à la présentation détaillée du *Groupware PBPCLG*. A cet effet, nous commencerons par une présentation de la dimension sociale en décrivant de manière claire plusieurs paramètres, entre autres, les différents modes d'interaction, les rôles d'intervention et leurs attributions, l'interaction via l'espace de travail partagé, etc. Ensuite, nous expliquerons la partie conception du *Groupware* tout en donnant plus d'importance à la conception de l'architecture logicielle adoptée. Nous terminerons le chapitre par une description des choix techniques opter pour mettre en œuvre notre prototype.

4.2. Aspects sociaux et conscience de groupe

Dans le domaine d'*ACAO* la dimension sociale de l'interaction de groupe d'apprenant est d'une importance capitale pour l'efficacité de l'apprentissage collective. Cette

dimension a incité un grand nombre de chercheurs comme en témoigne la multiplicité de publications qui ont été faites par exemple [Bouton et al. 20][Zhou et al. 20][Kwon et al. 14][Bulu 12]. De même, le concept de la conscience de groupe a aussi été de plusieurs années le centre d'intérêt de nombreux chercheurs dans le domaine d'ACAO. Dans cette section, nous allons présenter les possibilités fournies par *PBPCLG* en termes d'organisation du groupe d'apprenants, des modes d'interaction, des rôles disponibles, et de conscience de groupe.

4.2.1. Interaction sociale

L'apprentissage collaboratif est centré sur la théorie de l'apprentissage socialement constructiviste [Gergen 99], qui montre que l'apprentissage et la construction des connaissances sont influencés par l'interaction de groupe [Lee et al. 11] [Molinillo et al. 18]. Le développement d'un environnement collaboratif pour l'apprentissage de la programmation est une tâche ardue, car elle doit tenir en compte plusieurs aspects implicites et explicites, notamment ceux liés à l'activité de groupe dans le contexte classique d'éducation. Transformer ces aspects en mécanismes informatiques explicites n'est pas assez facile, mais implique d'identifier en premier lieu les règles sociales régissant les activités d'apprentissage collaboratif en présentiel (face-à-face). Pour soutenir l'interaction sociale dans un contexte virtuel d'ACAO, il est utile de proposer des mécanismes informatiques permettant de reproduire les situations d'apprentissage face-à-face tout en s'appuyant sur les technologies informatiques et de réseaux [Woo et Reeves 07][Bannan 02].

En analysant l'activité d'apprentissage dans un contexte classique, nous avons constaté que lors du déroulement des séances d'APP en classes, les apprenants interviennent intentionnellement à la résolution des problèmes avec différents rôles. Le code source partagé, qui est souvent rédigé sur un tableau, constitue un moyen efficace pour concrétiser l'espace de travail partagé. Ainsi, s'est avéré que les interactions de groupe qui se déroulent autour de cet espace ont rôle important pour gérer certains problèmes issus de l'activité collaborative comme la concurrence d'accès, les conflits, etc. Nous avons remarqué aussi que les apprenants permutent dynamiquement, de manière implicite et explicite, des rôles sur le code source partagé sur un tableau ou un écran d'affiche ainsi que les interventions sur ce dernier se font avec plusieurs modes d'interactions réelles.

Dans le reste de cette section, nous décrirons d'abord les modes d'interaction adoptés par *PBPCLG* afin de reproduire les modes réels d'interaction des apprenants. Ces modes offrent à l'apprenant plusieurs opportunités comme, par exemple, les échanges des idées, l'interaction avec leurs tuteurs, l'interaction avec les autres apprenants, etc. Ensuite, nous discuterons les différents rôles considérés par *PBPCLG*, et avec lesquels l'apprenant peut participer dans l'espace de travail partagé. Après, nous présenterons le

principe de notre proposition d'un *Diagramme de Classe Décoré (DCD)* pour asseoir une coordination, des activités du groupe, basée sur les artéfacts partagés (autrement dit les objets d'interactions). Enfin, nous détaillerons les différents mécanismes intégrés dans *PBPCLG* afin de gérer le concept de la conscience de groupe [Kwon et al. 14][Lee et al. 11].

4.2.1.1. *Modes d'interaction*

En partant de l'idée stipulant qu'un modèle d'interaction collaborative efficace doit reposer deux paramètres essentiels :

- *Ses capacités à réduire les contraintes logicielles pour les participants*
- *Ses capacités à reproduire les modes d'interaction naturels.*

En effet, nous avons défini pour le *PBPCLG* quatre modes d'interaction. Ces derniers ont été également inspirés des travaux de recherche de *Hedjazi (2011)*, réalisés dans le contexte de la télémaintenance coopérative et les travaux de *Ferradji (2017)*, menés dans le contexte de l'apprentissage collaboratif du raisonnement clinique.

Responsabilité individuelle : Elle exprime précisément la responsabilité individuelle que l'apprenant peut avoir afin de développer un fragment de code source. Le code source partagé est également décomposé en fragments qui sont souvent des objets ou des procédures/fonctions. Chaque fragment doit être développé par au moins un apprenant. Plusieurs apprenants peuvent accéder simultanément à un fragment de code, toutefois un seul qui aura l'accès avec ce rôle à un moment donné.

Responsabilité collective : À travers ce mode, les apprenants peuvent travailler ensemble sur un même fragment de code source. Ce mode est inspiré de la réalité, en particulier au sein d'une classe, lorsque plusieurs apprenants travaillent sur le même ordinateur pour produire un code source. Un seul apprenant est responsable de l'écriture et les autres ne peuvent que voir les modifications à l'écran ou générer des commentaires. En fait, via *PBPCLG*, deux ou plusieurs apprenants peuvent travailler ensemble sur le même fragment selon le mode *WYSIWIS (What You See Is What I See) relaxé* [Greenberg et al. 96]. *WYSIWIS* mentionne un modèle dans la zone de conception de l'interface graphique où plusieurs participants interagissent avec la même perception graphique de l'espace de travail partagé. Ce mode présente une plus grande flexibilité d'interaction de groupe du fait qu'il est plus naturel et mieux préféré pour le contexte d'apprentissage de la programmation que le mode *WYSIWIS strict*. À travers *WYSIWIS relaxé*, les tâches exécutées par un apprenant ne sont immédiatement affichées que pour les collaborateurs partageant le même fragment.

Actions alternatives : Différents apprenants peuvent développer des versions alternatives d'un même fragment de code. Ceci permet, sans aucun doute, de dégager

plusieurs solutions au même problème en combinant les versions alternatives développées.

Échange dynamique de tâches : Tout au long d'une session d'apprentissage, les apprenants peuvent, à tout moment, échanger dynamiquement des responsabilités sur des fragments. Dans cette perspective, des messages vocaux ou textuels peuvent jouer un rôle extraordinaire, comme par exemple, le cas où un apprenant désireux contribuer au développement d'un fragment de code en cours de modification par un autre apprenant peut lui envoyer un message pour libérer l'accès en écriture. Ce principe permet l'incarnation de certaines règles sociales reflétant les habitudes d'apprentissage classiques.

Par ailleurs, les modes d'interaction présentés concrétisent parfaitement le travail synchrone et asynchrone comme dans l'apprentissage collaboratif en coprésence. En outre, l'attribution des modes aux différents fragments de code est décidée conjointement par le groupe, mais son exécution est spécifiquement autorisée aux tuteurs.

4.2.1.2. Attribution des rôles

Un rôle est une désignation nominative pour un participant qui agira en conséquence. Pratiquement, lors de la programmation en binôme (en anglais *PP* : *Pair Programming*), où deux personnes travaillent ensemble sur un clavier partagé, seuls deux rôles sont considérés : Le *pilote* (en anglais *Driver*) et l'*observateur* ou *navigateur*. Dans une session *PP*, un participant agit en tant que *pilote* et implémente le code contrôlant le clavier et la souris ; un autre participant agit, en même temps, en tant qu'*observateur* ou *navigateur* et est responsable de la révision du code, de la prévention et de l'identification des erreurs logiques et syntaxiques dans le code [da Silva et Prikladnicki 15].

Néanmoins, nous pensons que, dans notre contexte, ces deux rôles sont insuffisants, pour de nombreuses raisons :

- *Les apprenants sont séparés géographiquement ;*
- *Ils apprennent ensemble en résolvant un même problème de programmation.*
- *Ils peuvent agir simultanément sur leurs différents claviers pour développer des fragments de code source.*

En fait, nous avons décidé d'adapter, à notre contexte, les rôles définis par *Posner & Baeker* (1992) pour des fins d'édition partagée, notamment pour permettre aux apprenants de rédiger, de consulter, d'annoter et de commenter des fragments de code d'un programme source partagé. Dans cette perspective, *PBPCLG* offre plusieurs rôles avec lesquels les apprenants peuvent intervenir à l'activité partagée. Le premier rôle baptisé « *Rédacteur* » qui est attribué à l'apprenant désireux introduire ou modifier le code

source d'un fragment donné, tandis que le rôle « *lecteur* » qui est également assigné à un apprenant pour lui permettre de consulter le contenu d'un fragment spécifique. Un autre rôle lié aux tâches de supervision du code source peut également être attribué aux participants. Il s'agit du rôle de « *superviseur* » avec lequel un apprenant peut suivre de manière synchrone ou asynchrone le processus de développement d'un fragment de code. L'avantage de ce rôle est principalement d'améliorer le sens de responsabilité chez les apprenants ainsi que de favoriser leur implication effective dans l'avancement de la tâche partagée en cours et de leur donner par conséquent un sentiment de volonté de contribuer à la résolution du problème. Le dernier rôle, appelé « *commentateur* », est prévu pour les apprenants désirant générer des commentaires à propos de certains fragments développés par les autres.

Un apprenant pourrait avoir à la fois plusieurs rôles différents. Il peut, entre autres, être autorisé à modifier certains fragments et à ne lire ou à commenter que certains autres. Comme il peut superviser l'avancement de certains autres fragments.

4.2.1.3. *Diagramme de classe décoré interactif*

Contrairement à la programmation à usage individuel, la participation de plusieurs apprenants programmeurs travaillant simultanément sur le même code source nécessite de mettre à leur disposition des outils de gestion des conflits et concurrence d'accès. Ces outils doivent, en outre, se charger de la visualisation, à tout moment, de l'état global du programme source partagé. En effet, nous avons intégré dans *PBPCLG* un outil graphique permettant aux apprenants de fragmenter leur programme source partagé en plusieurs parties. Ainsi, vu que notre *Groupware* s'intéresse à l'apprentissage de la programmation via *Java*, les fragments du code sont souvent des classes d'objets. Cette structuration crée donc une représentation graphique du code source. Elle est également appelée « *Diagramme de Classe Décoré (DCD)* ». Les accès, par les apprenants, aux différentes parties (fragments) du programme source partagé ne sont effectués qu'avec les rôles précédemment définis. En outre, l'état de chaque fragment du code doit clairement apparaître sur le *DCD* en lui ajoutant en conséquence des informations décoratives dynamiques. Dans cette vision, l'interaction de groupe se repose sur ce *DCD* et les apprenants peuvent, à travers les mécanismes de la conscience de groupe attachés au *DCD*, être informés à n'importe quel moment sur l'état de l'activité collaborative.

La figure 3.6 illustre un *DCD* avec ses différents fragments où chacun apparaît avec son identifiant. Afin de réduire la charge cognitive chez les apprenants, des informations complémentaires comme les couleurs, les icônes, les images, etc., sont insérées dynamiquement sur le diagramme partagé afin d'y doter par des informations de la conscience de groupe. Par exemple, l'icône du *verrouillage* et la *couleur noire* en haut du

fragment "Professor" indiquent que celui-ci est verrouillé et que son code est en cours de développement par le participant *Chorfi*. Parallèlement le participant *Iyad* est en train de consulter le contenu d'un tel fragment. Le fragment "Student" est en train d'être consulté par le participant *Aouag*. Les participants *Hedjazi* et *Boubiche* sont en cours de développement du fragment "Seminar" en utilisant le mode d'interaction « Actions alternatives » (voir section 4.2.1.1.). Enfin, le fragment "Enrollment" est à l'état libre.

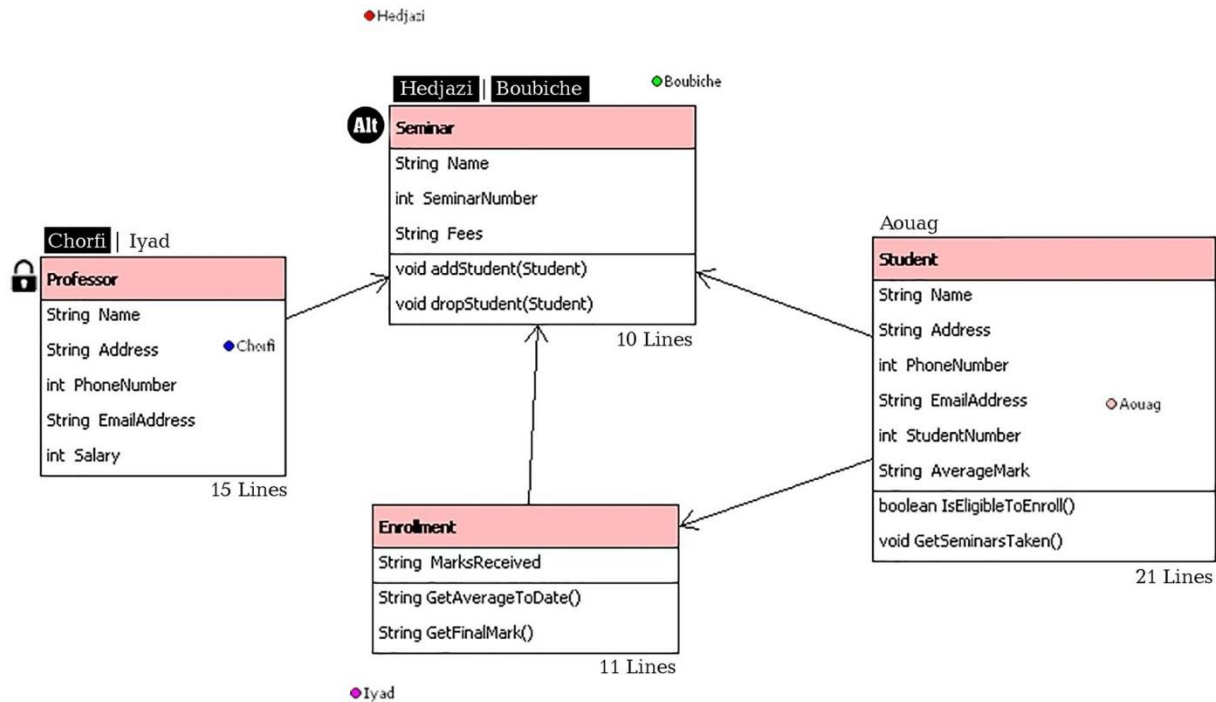


Figure 4. 1 : Diagramme de Classe Décoré [Chorfi et al. 20].

4.2.2. Mécanismes de la conscience de groupe

La Conscience de Groupe (CG) [Dourish et Bellotti 92] est un concept très utile, particulièrement en analysant les activités collectives. Il est largement abordé par les chercheurs, notamment ceux intéressés par les domaines TCAO et ACAO [Hedjazi et Zidani 14]. Nous avons accordé une importance extrême à ce concept, car il représente l'un des facteurs clés de la réussite et l'efficacité du travail de groupe.

De nombreuses définitions ont été proposées afin d'éclaircir le concept de la CG. Quelques définitions, que nous jugeons utiles pour notre travail de recherche, ont été largement discutées dans les travaux de thèse de Hedjazi (2011). Parmi ces définitions, nous invoquons celle de Dourish (1992), qui a considéré la CG comme étant « la compréhension des activités des autres qui fournit un contexte pour l'activité propre de chaque participant. ».

Selon *Liccardi et al.* (2007), la CG est « *la conscience des membres du groupe sur les rôles, responsabilités, activités, actions et position des autres membres durant le processus* ». D'autres chercheurs ont attaché ce concept à l'activité de coordination, par exemple *Beaudouin-Lafon* (1992), qui a soulevé que « *chaque individu doit être conscient de ce que font ses collègues pour faciliter la coordination.* »

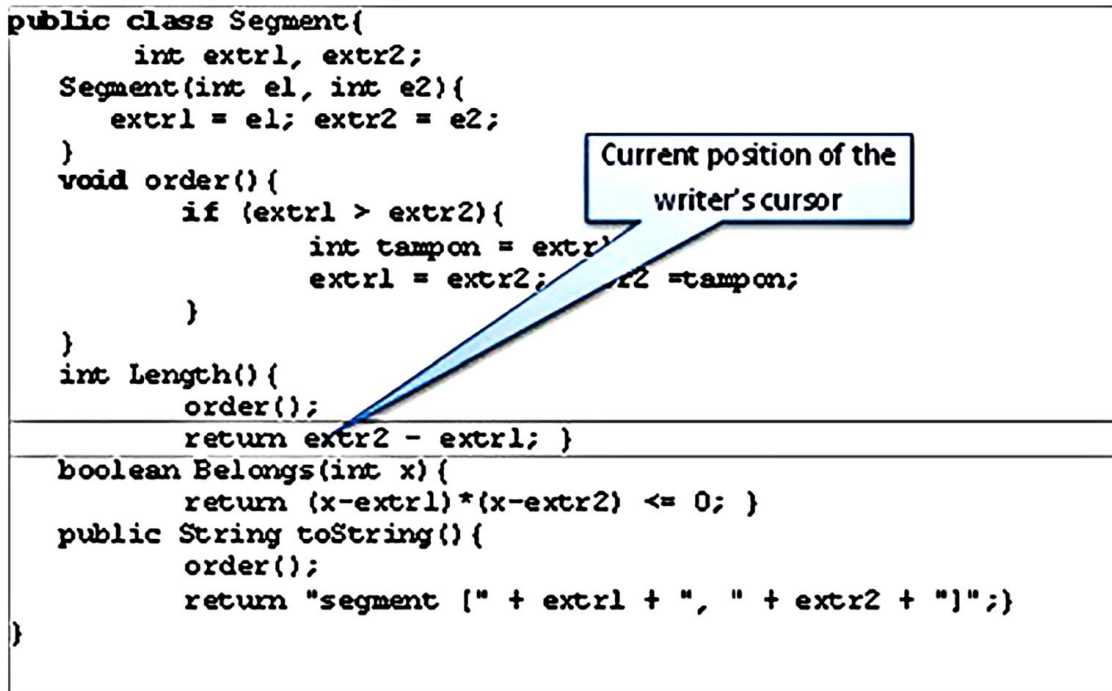
Partant de ces définitions, *Hedjazi* (2011) a jugé que la CG reflète l'état mental d'un acteur et son comportement dans l'activité collective.

Dans notre contexte où les apprenants sont géographiquement dispersés, un sérieux problème peut être soulevé, par exemple l'absence de contacts directs entre apprenants les uns des autres et leur environnement d'apprentissage, qui engendre en conséquence une insuffisance de nombreuses informations implicites de la CG, comme les expressions faciales, les expressions corporelles, etc. Toutefois, les mécanismes virtuels de CG peuvent constituer une alternative pour reconstruire ces informations. Par ailleurs, les informations propagées par ces mécanismes peuvent diminuer les efforts mentaux des apprenants en coordonnant leurs actions mutuelles et peuvent, en conséquence, améliorer l'efficacité de l'activité collaborative autour de la construction de la connaissance. En outre, les mécanismes de CG peuvent augmenter, chez les apprenants, le comportement d'engagement dans le processus d'apprentissage [Liu et al. 18].

CG est devenu une notion importante dans le domaine de l'ACAO depuis son introduction jusqu'à présent Liu (2018). Elle se réfère aux informations, sur l'environnement d'éducation, que les apprenants obtiennent au fur et à mesure du déroulement du processus d'apprentissage. Cela peut aider les apprenants à combiner les tâches individuelles et les tâches collectives, à créer un contexte dans lequel ils peuvent interpréter les actions de leurs collaborateurs et à réduire l'effort mental nécessaire pour coordonner les actions collaboratives au sein de l'espace de travail partagé.

Un nombre important d'études ont révélé l'influence positive du soutien de CG sur l'apprentissage collaboratif [Liu et al. 18] [Erkens et Bodemer 19]. Ainsi, afin de rendre la collaboration de groupe facile et efficace, les *Groupwares* doivent tenir compte de ce concept de CG en développant des supports appropriés. En conséquence, le PBPCLG offre de nombreux mécanismes de CG à savoir : un mécanisme nommé *liste de participants*, qui affiche en temps réel la liste des apprenants online ; les mécanismes de *chat vocal et textuel*, qui assurent la transmission, synchrone, des différentes conversations textuelles (chat) et vocales (audio). PBPCLG intègre un outil de *chat* avec un module d'enregistrement de l'historique, qui sert à stocker, de façon automatique, toutes les conversations déroulées dans l'espace partagé ; un mécanisme optionnel de *télépointeur*

qui précise immédiatement l'emplacement, dans l'espace partagé, des différents pointeurs que les apprenants manipulent lors du déroulement de la session d'apprentissage. Des étiquettes affichant les identités des apprenants sont attachées aux différents pointeurs qui se sont distingués les uns des autres par leurs couleurs ; un mécanisme précisant l'endroit des différents curseurs manipulés par les apprenants dans l'éditeur du code source (Figure 4.2) ; un mécanisme de notification pour diffuser en temps réel, aux apprenants concernés, les différents événements produits dans l'espace de travail partagé.



```

public class Segment{
    int extr1, extr2;
    Segment(int e1, int e2){
        extr1 = e1; extr2 = e2;
    }
    void order(){
        if (extr1 > extr2){
            int tampon = extr1;
            extr1 = extr2; extr2 = tampon;
        }
    }
    int Length(){
        order();
        return extr2 - extr1; }
    boolean Belongs(int x){
        return (x-extr1)*(x-extr2) <= 0; }
    public String toString(){
        order();
        return "segment [" + extr1 + ", " + extr2 + "];"}
}

```

Figure 4. 2 : Position du curseur d'un apprenant « Rédacteur ».

4.3. Scénario

Supposons un groupe d'apprenants composé de quatre membres : *Chorfi, Iyad, Boubiche* et *Aouag*, qui collaborent via *PBPCLG*, en résolvant un problème de programmation commun. Chaque apprenant rejoint la session d'apprentissage à travers l'introduction des informations d'authentification (figure 4.3). Ensuite, l'interface principale du *Groupware* (figure 4.4.) s'affiche sur son écran en lui présentant toutes les informations nécessaires au déroulement de son activité collaborative d'apprentissage. Ces informations concernent, entre autres, les apprenants présents en sessions (Liste des participants), le tuteur de la session, le problème autour duquel la session va se dérouler, etc.

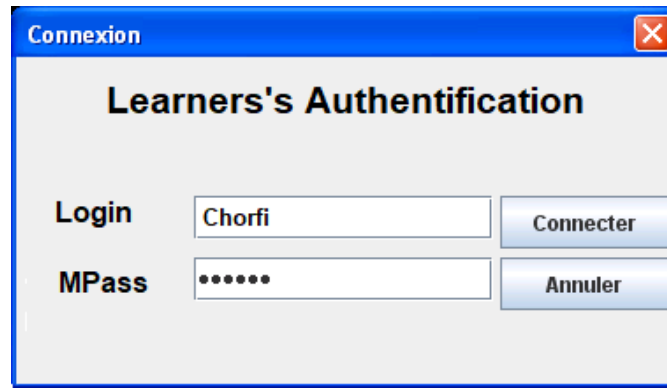


Figure 4. 3 : Authentification d'un apprenant via Pseudo et Mot de passe.

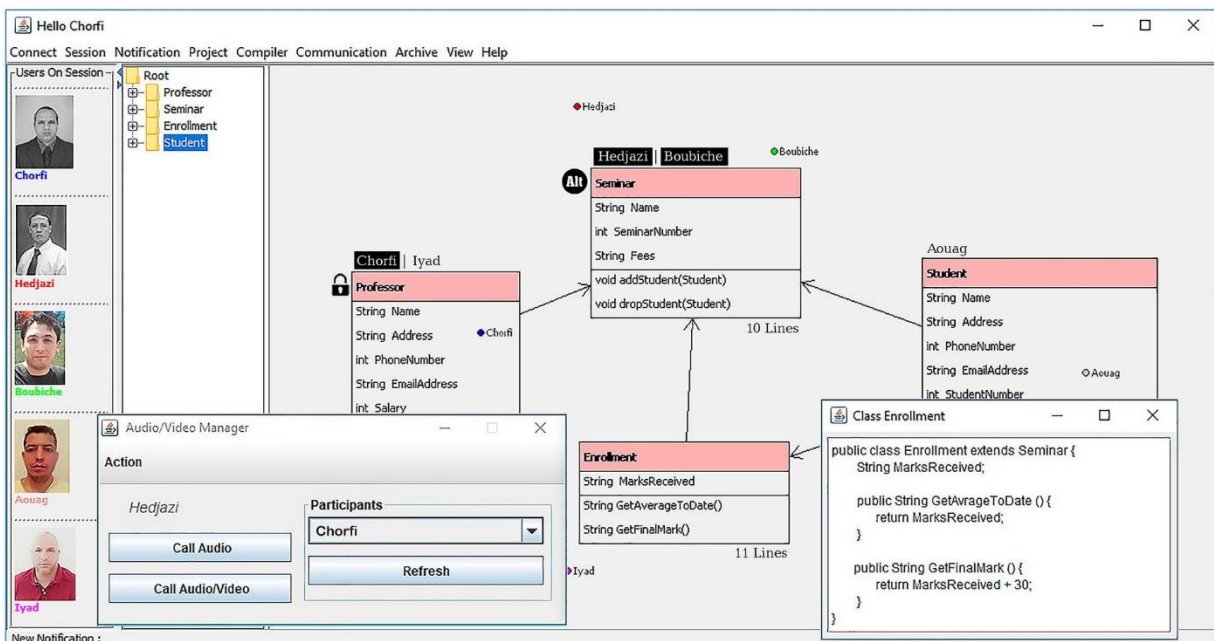


Figure 4. 4 : Interface principale du PBPCLG.

La description du problème (figure 4.5) est préparée par le tuteur *Hedjazi* qui est considéré comme un membre important dans le groupe. Lorsque les membres rejoignent la session d'apprentissage, ils analysent d'abord le contenu de cette description. Pour des fins de clarifications, les apprenants entrent souvent en discussions (en utilisant les mécanismes de conversations vocaux/chat) dont lesquelles le tuteur joue un rôle crucial. Parfois les apprenants sont motivés à lancer des recherches sur le web pour consulter certaines ressources pédagogiques jugées utiles. Progressivement, ils convergent vers une compréhension du problème.

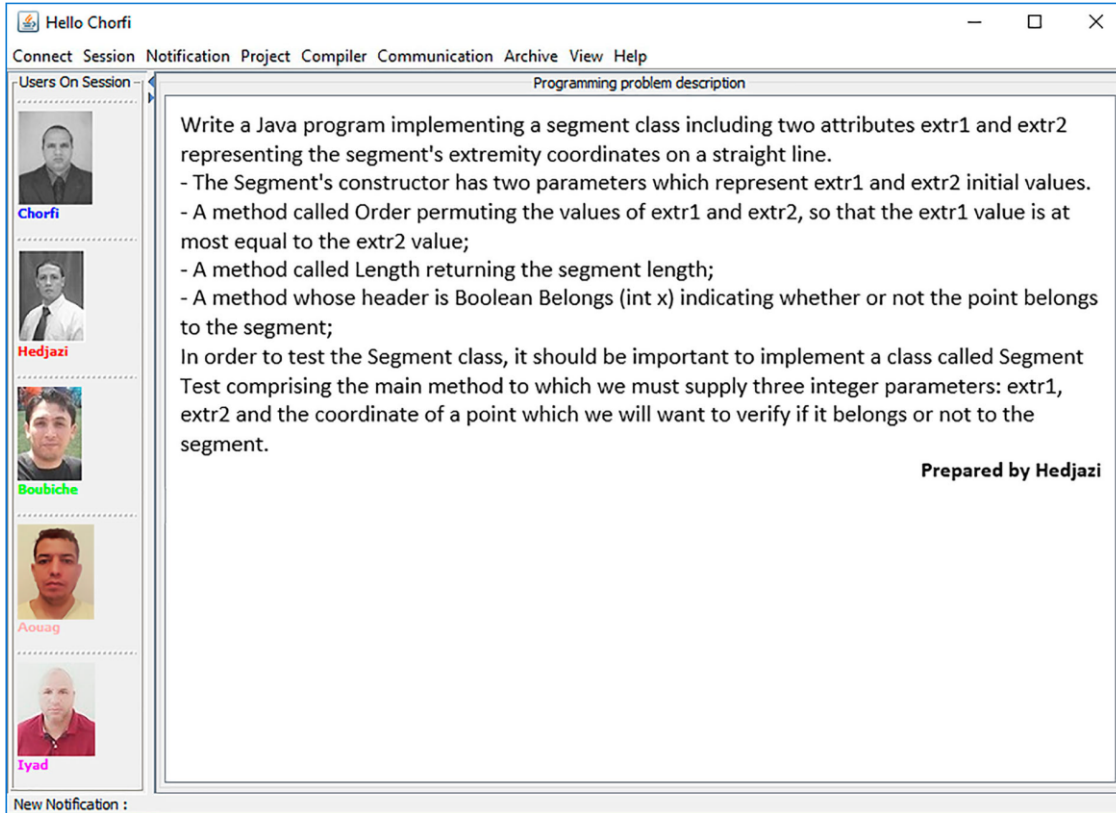


Figure 4. 5 : Description textuelle d'un problème de programmation.

Ensuite, les apprenants décident de structurer le programme cible en plusieurs fragments de code construisant ainsi la forme graphique du *DCD* (figure 4.1). De plus, les rôles sont assignés aux différents apprenants pour qu'ils puissent intervenir à l'activité partagée. En fait, certains apprenants peuvent se charger de la tâche de rédaction du code source associé aux fragments, tandis que d'autres peuvent jouer le rôle de révision. Les apprenants peuvent développer le programme de manière simultanée. Comme il est illustré sur la figure 4.6., les apprenants *Chorfi* et *Boubiche* travaillent ensemble sur le même fragment de code source associé à l'objet *Segment*. Après avoir terminé la rédaction du code d'un tel fragment, *chorfi* a sollicité *Boubiche* pour faire des révisions. Ils ont décidé, en effet, d'échanger un certain nombre de messages.

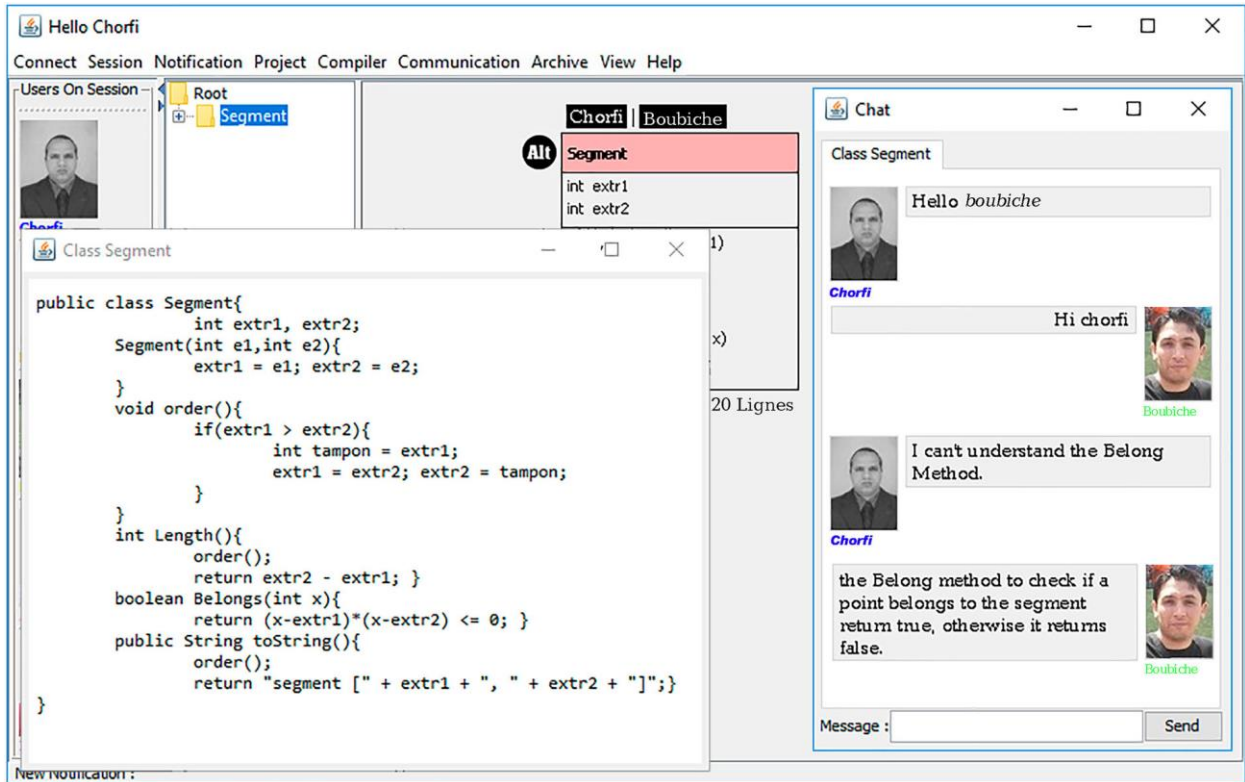


Figure 4. 6 : Une discussion entre Boubiche et Chorfi.

Lors d'une session, plusieurs fragments peuvent être développés simultanément par les apprenants en utilisant les modes d'interaction discutés dans le chapitre 4. Deux ou plusieurs apprenants peuvent, par exemple, partager le même fragment afin de développer son code de manière collaborative. Le mode *WYSIWIS* intégré dans le *PBPCLG* permet aux apprenants d'afficher le code source partagé et toutes les modifications apportées par n'importe quel apprenant « Rédacteur » apparaissent immédiatement sur les écrans de leurs collaborateurs. De plus, chaque participant peut demander à être informé chaque fois qu'un évènement se produit dans l'espace de travail partagé. En utilisant le mécanisme de *notification*, la figure 4.7 présente un exemple dans lequel le participant *Hedjazi* a établi un engagement de notification sur le changement de statut du fragment « Segment ». Il a également précisé la méthode dont il souhaite être informé.

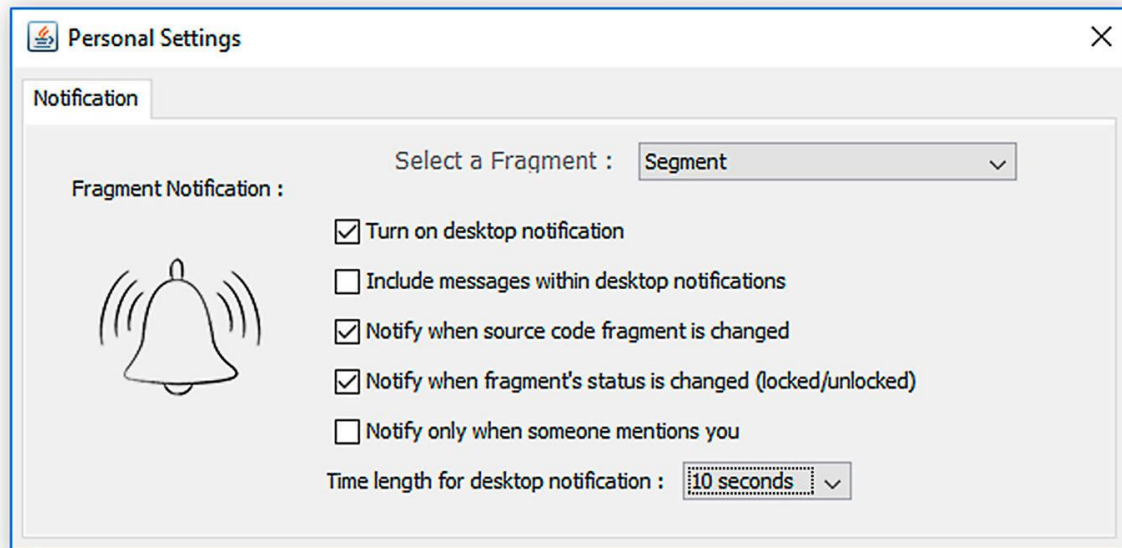


Figure 4. 7 : Fenêtre d'engagement de notification.

4.4. Implémentation de PBPCLG

Nous présenterons dans cette section, les aspects techniques de notre contribution. Nous commencerons d'abord par une explication de l'architecture logicielle adoptée pour mettre en œuvre notre prototype *PBPCLG*. Ensuite, nous discuterons les détails liés à nos choix technologiques utilisés pour développer notre *Groupware*.

4.4.1. Architecture logicielle

L'implémentation du *PBPCLG* se repose principalement sur un modèle d'architecture logicielle en couches avec une approche hybride (figure 4.8). Ce modèle résulte de l'instanciation du méta-modèle *Dewan* (1999) qui combine les deux approches répliquée et centralisée. Ainsi, nous avons décomposé le noyau fonctionnel du *Groupware* en plusieurs fonctionnalités. En utilisant la taxonomie de *Salber* (1995), ces fonctionnalités sont regroupées, selon leurs natures, dans trois espaces : espace de communication, espace de coordination et espace de production. Ces trois espaces sont présents au niveau de chaque couche de l'architecture. L'espace de production rassemble toutes les fonctionnalités de gestion de la représentation graphique du *DCD* ainsi que d'édition du texte de programme source à savoir : la saisie, la modification, la suppression, les sauvegardes, etc. Quant à l'espace de coordination, il intègre toutes les tâches de la gestion de la synchronisation et la concurrence d'accès. Enfin, l'espace de communication qui regroupe les fonctionnalités d'échange de messages vocaux et textuels (chat).

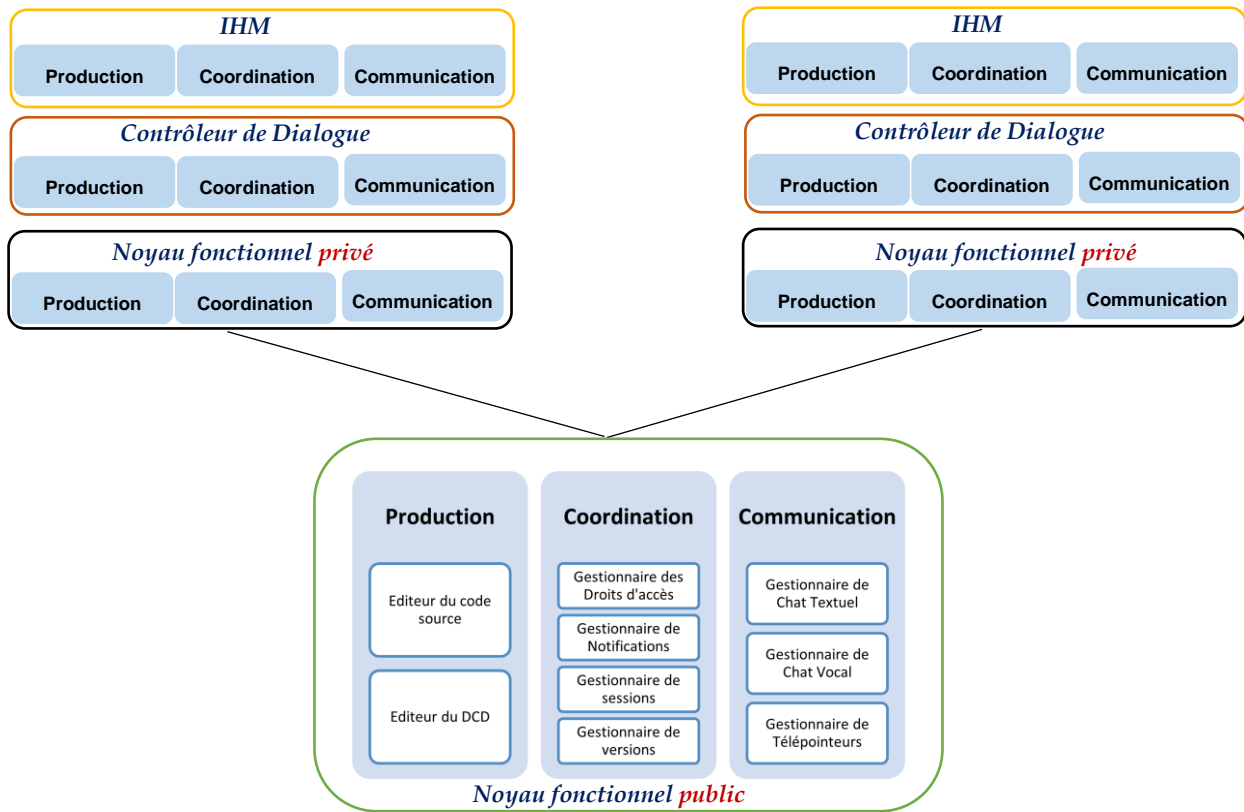


Figure 4. 8 : Architecture logicielle du PBPCLG.

L'avantage de cette architecture est que l'activité de la synchronisation devient plus au moins facile, tout en préservant la cohérence des données partagées du moment qu'elles sont centralisées au niveau du serveur. Néanmoins, certaines données privées (propre à chaque apprenant) sont déportées aux stations clientes, et ce dans le but d'optimiser le nombre de messages échangés entre les clients et le serveur. En effet, les traitements sur ces données sont immédiatement effectués par la station locale et la couche « *Contrôleur de dialogue* » doit interroger les composants de la couche *IHM* pour préserver la cohérence entre les *GUI* et les données. Lorsqu'il s'agit d'une action qui nécessite une coordination avec les autres stations, un composant coordination est intégré dans chaque couche afin de jouer un tel rôle.

D'autre part, il est utile de préciser que le processus de développement de tout système *Groupware* repose principalement sur les fondements des systèmes distribués. Ces derniers utilisent beaucoup la notion d'évènement. Dans notre architecture logicielle, cette notion primordiale d'évènement permet au noyau fonctionnel aussi bien privé que public de préserver la cohérence des interfaces utilisateurs et des données manipulées par les différents participants. Par conséquent, les évènements produits par les différents

apprenants au niveau de leurs *GUIs* sont immédiatement interceptés par la couche *Contrôleur de dialogue* afin de générer toutes les diffusions nécessaires en avertissant le *noyau fonctionnel privé* afin de lancer les traitements locaux adéquats. Lorsqu'un évènement provoque des mises à jour au niveau des stations distantes, une communication entre le *Noyau fonctionnel privé* et le *Noyau fonctionnel public* (centralisé au niveau du serveur) doit être établie. Ce dernier à son tour doit avertira tous les autres *Contrôleurs de Dialogue* des autres apprenants. De ce fait, toute action collaborative réalisée au niveau de *GUI* d'un apprenant donné doit être traitée par le *Noyau fonctionnel public* centralisé au niveau du serveur. A titre d'exemple, lorsqu'un apprenant souhaite participer au développement du code source d'un fragment donné, il doit solliciter ses droits d'intervention. En effet, cette action ne pourra s'autoriser sauf si le *Noyau fonctionnel public* donnera sa permission en sollicitant le module *droit d'accès* du composant *Coordination*. Dans ce cas les autres *GUIs* doivent être mis à jour comme la mise à jour du *DCD* partagé, la mise à jour du fichier correspondant au fragment de code modifié, etc.

4.4.2. Choix technologiques d'implémentation

Le PBPCLG, est entièrement développé en langage *Java*. Le modèle d'exécution que nous avons adopté pour mettre en œuvre la distribution des tâches/données et la communication interprocessus est également le modèle *client/serveur*. Ainsi, les opérations de contrôle qui sont mises en œuvre sur aussi bien le serveur que les clients permettent d'assurer la cohérence aussi bien des données partagées que les différentes *GUIs* ; et chaque *client* doit interagir avec le *serveur* en amont de chaque action sur l'espace de travail partagé. L'interaction avec les participants est gérée par les composants de la première couche (*IHM*) qui mettent en œuvre les fonctionnalités de gestion des interfaces graphiques. Les fonctionnalités relatives à la gestion de l'interaction collaborative sont également intégrées dans les noyaux fonctionnels privés et publics.

Par ailleurs, les données manipulées par le *Groupware* sont accessibles par le biais du composant *JDBC*. De plus, la communication entre les différents processus distribués est assurée par le mécanisme d'invocation de méthode à distance offert par la technologie *Java-RMI*. Ce choix peut être justifié par sa caractéristique qui rend abstraits les détails de bas niveau de la communication et simplifie ainsi le processus de mise en œuvre. Enfin, pour les échanges de message de type audio, *PBPCLG* utilise l'API *Java Media Framework (JMF)*. Cette dernière nous a permis de développer facilement les médias acoustiques.

4.5. Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux sur lesquels repose le fonctionnement du *PBPCLG* afin de permettre aux apprenants d'apprendre, à distance et en collaboration avec d'autres apprenants, les bases de la programmation

informatique. Le principe de l'apprentissage adopté par notre prototype de *Groupware* se base sur la méthode pédagogique active baptisée « *APP* ». Cette méthode offre aux apprenants les opportunités de construire leurs connaissances en s'appuyant sur la résolution de problèmes et les interactions de groupe. C'est ainsi que nous avons défini pour *PBPCLG* plusieurs modes d'interaction qui sont bien expliqués dans ce chapitre. D'autre part, un autre point jugé extrêmement essentiel a été aussi invoqué dans ce chapitre. Il s'agit de l'attribution des rôles avec lesquels les apprenants peuvent participer à l'activité collaborative tout en préservant la cohérence aussi bien de leurs GUIs que les données partagées. Enfin, les détails du processus d'implémentation de notre *Groupware* ont été clairement présentés à la fin du chapitre.

Arrivant à ce stade, nous ne pouvons pas confirmer que notre travail de recherche a atteint ces objectifs, toutefois, il fallait étudier les aspects expérimentaux de notre contribution. L'un des sujets les plus essentiels, et qui attire l'attention de beaucoup de chercheurs dans ce contexte, est celui qui s'intéresse à mesurer le degré d'acceptabilité, par les apprenants, des systèmes *Groupware* orienté apprentissage. En effet, nous avons mené une étude expérimentale mesurant l'intention d'adoption du *PBPCLG* par les apprenants en informatique. Cette étude se repose également sur le modèle proposé par *Venkatesh et al.* (2003). Les détails d'une telle étude ainsi que ses résultats obtenus sont bien présentés dans le chapitre suivant.

Chapitre 5 : Facteurs d'acceptation et d'utilisation du PBPCLG : Application du modèle UTAUT

5.1. Introduction

Le besoin croissant en informatique et en *TIC*, dans les organisations socioéconomiques, est considérable. Certaines études montrent que, depuis les années 80, environ 50 % de tous les nouveaux investissements en matière de ressources ont été effectués dans les TIC [Westland et Clark 00]. D'autre part, l'acceptation de ces technologies, par ses utilisateurs effectifs, a constitué le sujet d'intérêt de multiples études expérimentales. De telles études se reposent souvent sur des modèles théoriques.

En conséquence, pour mesurer l'adoption et l'usage de notre contribution concrétisée par le Groupware *PBPCLG*, nous devons choisir un modèle théorique explicatif de l'adoption des technologies. Ceci nous mène, tout d'abord, à étudier, de manière générale, les caractéristiques des principales théories créatrices de quelques modèles introduits dans la littérature. Ensuite, nous pensons que la présentation de quelques modèles de l'intention d'usage semble importante afin de bien justifier notre choix du modèle de *Venkatesh et al.* (2003). Enfin, les détails de notre étude expérimentale sont bien expliqués en analysant et discutant les résultats obtenus.

5.2. Théories créatrices des modèles de l'intention

Les modèles de l'intention d'usage reposent sur des théories psychosociales du comportement de l'être humain. Plusieurs théories sont à considérer dans cette étude

comme la *Théorie de l'Action Raisonnée (TAR)* [Fishbein et Ajzen 75], la *Théorie du Comportement Planifié (TCP)* [Ajzen 91] et la *Théorie des comportements interpersonnels (TCI)* [Triandis 80].

La *TAR*, qui est un modèle issu de la psychologie sociale, estime que les excitations (ou stimulus) externes n'influencent les comportements qu'à travers des transformations au sein de la structure des croyances de l'être humain. Elle permet de donner des explications, notamment en termes de relations entre attitude et comportement lors du déroulement d'une action humaine donnée. Par conséquent, une décision à prendre par un être humain pour adopter un comportement spécifique est basée sur les conséquences qu'il espère aboutir à la suite d'un tel comportement.

La *TAR* vise à interpréter le comportement décidé par l'être humain. Cette interprétation cherche à trouver des liens entre les motivations et les résultats d'une action humaine. Il s'agit donc d'une explication fournie en apercevant l'individu dans l'action. En d'autres termes, tout comportement humain (autrement dit *intention comportementale*) est produit suite à une croyance que possède l'être humain à propos du comportement et le résultat qu'il mènera (figure 5.1). Néanmoins, ceci est insuffisant pour déterminer les intentions comportementales, il fallait considérer les normes subjectives ainsi que l'attitude à l'égard du comportement.

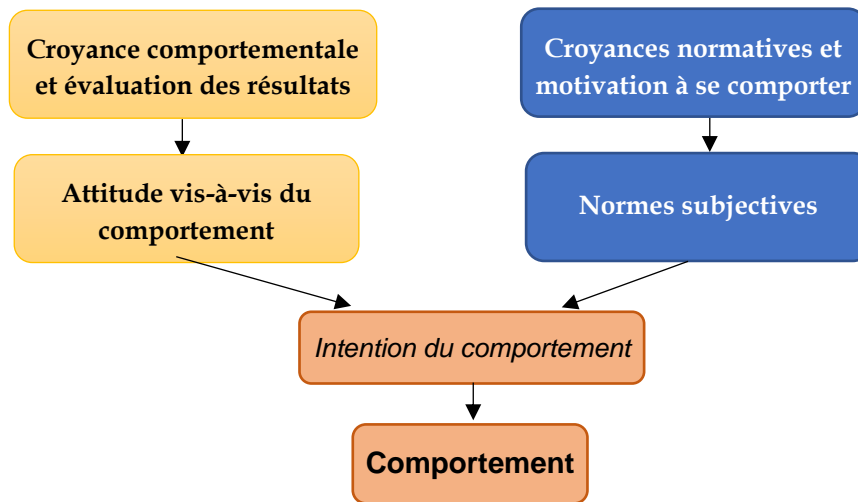


Figure 5. 1 : Schéma de la TAR [Davis et al. 89].

Ce schéma peut être formulé par la simple équation suivante :

$$IC = (AC \times P) + (NS \times P) \dots \dots \dots (1)$$

Où :

- **IC** : Intention Comportementale ;
- **AC** : Attitude Comportementale ;
- **NS** : Norme Subjective ;
- **P** : Pondération déterminée de manière empirique.

La théorie *TCP* a été proposée pour améliorer la théorie *TAR*. Elle sert à donner des explications aux comportements humains en répondant à trois types de questions permettant d'établir des liens les attitudes, la sensation d'auto-efficacité et les règles sociales (figure 5.2). En d'autres termes, le comportement d'un être humain, pour qu'il devienne physique, il doit au préalable être décidé et planifié.

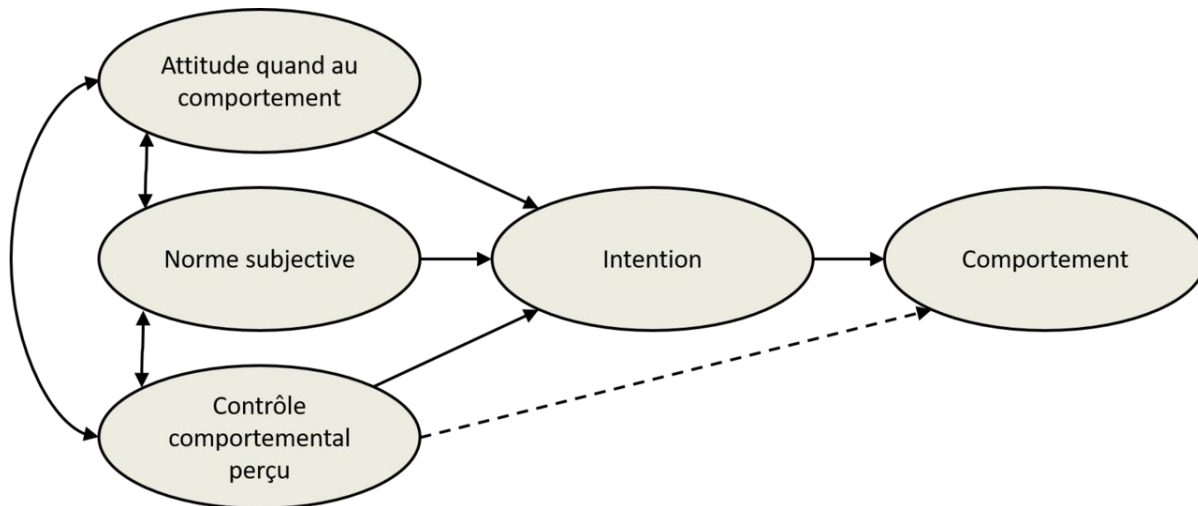


Figure 5. 2 : Schéma de la TCP [Ajzen 91].

Cette théorie a été appliquée dans plusieurs contextes ayant une grande importance dans la vie sociale de l'être humain par exemple : le contexte scolaire pour chercher des explications aux comportements d'abandon scolaire, le contexte professionnel afin de trouver les raisons qui provoquent l'absentéisme au travail, le contexte économique pour comprendre les comportements relatifs à la consommation de certains produits, etc.

La *TCI* est une hybridation des deux théories précédentes. Elle englobe ainsi tous les paramètres considérés dans *TAR* et *TCP*. Suivant le principe d'une telle théorie, trois facteurs sont à considérer pour déterminer tout comportement humain. Il s'agit de *l'intention*, *l'habitude* et les *conditions qui facilitent et soutiennent* l'adoption dudit comportement. Dans cette optique, le facteur d'*intention*, à son tour, considère d'autres facteurs importants comme l'impact social, les effets associés au comportement lui-même, les attitudes par rapport au comportement, et les convictions personnelles.

Bien que cette théorie semble pertinente et très riche en termes de déterminants, peu d'études, dans le contexte d'analyse du comportement humain à l'égard des systèmes informatiques, ont opté pour les modèles qui se basent une telle théorie.

5.2.1. Modèles d'intention d'utilisation des technologies

Dans la littérature, de nombreux modèles ont été proposés afin d'évaluer l'intention humaine à l'égard de l'usage de la technologie. Ces modèles, qui se basent sur les

principes des trois théories que nous venons d'expliquer dans la sous-section précédente, ont joué un rôle crucial dans différents contextes, particulièrement pour améliorer la qualité des technologies à offrir à l'humanité. Parmi ces modèles, il y'a pratiquement quelques-uns qui ayant marqués des évolutions importantes de recherches à savoir : le modèle de l'acceptation de la technologie (en anglais *TAM : Technology Acceptance Model*) proposé par [Davis 89], le modèle des 3P (Pouvoir- Performance-Perceptions) introduit par Dillon & Morris (1999) et le modèle de la théorie unifiée d'acceptation et d'utilisation des technologies (en anglais *UTAUT : Unified Theory of Acceptance and Use of Technology*) de Venkatesh et al. (2003). Nous nous intéresserons, dans le cadre de cette thèse, au dernier modèle afin de tirer les explications pertinentes à propos de l'intention comportementale des apprenants à l'égard de l'usage de notre *Groupware PBPCLG*.

5.2.1.1. Modèle TAM

Le principe de ce modèle repose sur la théorie TAR en dégageant deux facteurs essentiels qui sont : l'utilité perçue et la facilité d'usage perçue. Ces dernières ayant une influence importante sur l'attitude de l'être humain à l'égard de l'utilisation de la technologie en question. C.-à.-d. Afin de sortir avec les prédictions utiles à propos du comportement humain vis-à-vis l'utilisation d'une technologie quelconque, il faut considérer les deux facteurs précédents. En réalité, ces deux facteurs (*utilité perçue* et *facilité perçue*) sont liés par un lien d'influence d'où la *facilité d'utilisation perçue* impacte significativement l'*utilité perçue* ainsi que les deux sont incités par des *stimulus*. D'autre part, l'*intention d'adoption* est aussi influencée par l'*utilité perçue* (figure 5.3).

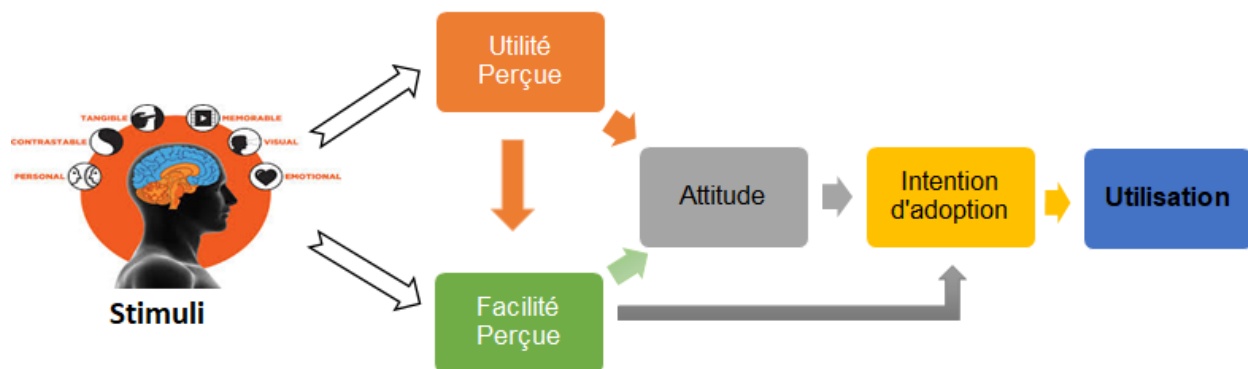


Figure 5. 3 : Schéma du modèle TAM [Davis 89].

5.2.1.2. Modèle des 3P

Le modèle des 3P, qui est l'acronyme de (*Pouvoir- Performance-Perceptions*), s'appuie sur le principe stipulant que la décision d'usage d'une technologie quelconque est influencée directement ou indirectement par les trois facteurs suivants : *pouvoir*, *performance* et *perceptions*. Les 3P ont chacun une influence directe sur la décision

d'adoption d'une technologie. Toutefois, le *pouvoir* et la *performance* ont encore un effet indirect en influençant les perceptions qui influencent à leur tour ladite décision (figure 5.4).

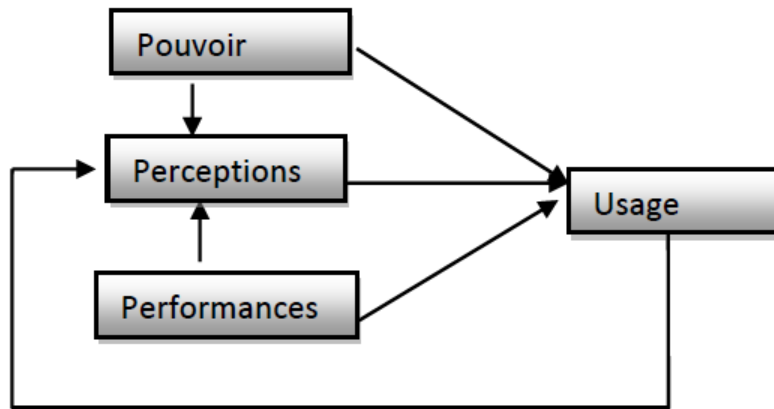


Figure 5. 4 : Schéma du modèle des 3P [Dillon et Morris 99].

Par ailleurs, les perceptions que l'être humain peut avoir en adoptant une technologie donnée renvoient à l'évaluation subjective de cette dernière. Ainsi, la performance, qui tient en compte de la dimension fonctionnelle de la technologie, renvoie en effet à la performance dont un être humain peut bénéficier en utilisant de telles technologies. Quant au *pouvoir*, qui fait référence à l'utilité et à la fonctionnalité de la technologie, représente les aptitudes techniques qu'une technologie dispose pour accomplir une tâche à réaliser par un être humain.

5.2.1.3. Modèle UTAUT

UTAUT est un modèle qui a été conçu par Venkatesh et al. (2003) afin de créer une version unifiée de la théorie permettant d'expliquer l'intention d'adoption d'usage, chez l'être humain, à l'égard d'une technologie innovante. D'autre part, le principe de cette unification peut être résumé en ce qui suit :

- La notion de performance espérée et l'effort attendu que UTAUT référence dans son modèle sont directement priés du Modèle TAM ;
- L'influence sociale et les conditions qui facilitent l'usage sont tirées des théories de l'apprentissage social ou collaboratif ;
- Quatre variables (sexe, âge, expérience, volontariat) sont par la suite utilisées comme facteurs modérateurs influençant l'intention de comportement.
- L'effet des conditions facilitatrices sur l'usage est significatif en fonction de la présence des modérateurs « âge » et « expérience ».

Partant de ce principe, le modèle UTAUT peut être schématiquement résumé par la figure 5.5.

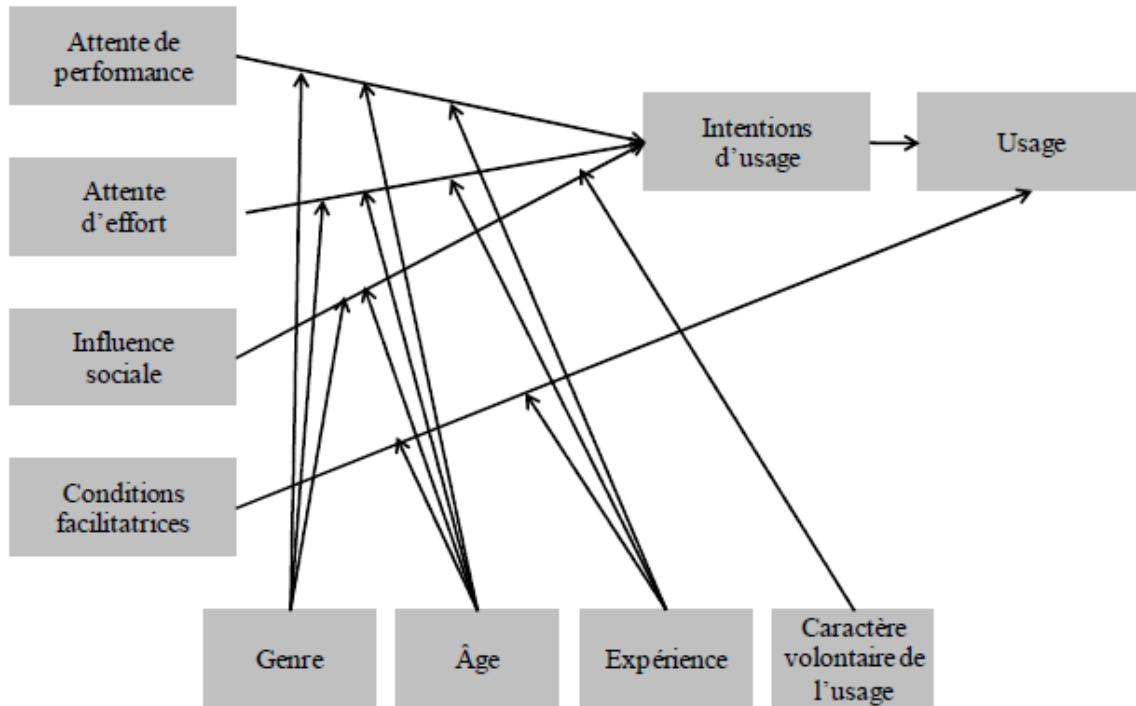


Figure 5. 5 : Schéma du modèle UTAUT [Venkatesh et al. 03].

Le modèle *UTAUT* constitue donc un outil efficace et utile pour les gestionnaires, notamment pour leur permettre d'évaluer les possibilités de succès de l'adoption des technologies dans leurs travaux quotidiens. Quant aux développeurs, *UTAUT* peut jouer un rôle primordial, particulièrement pour comprendre les facteurs influençant l'intention d'adoption de leurs applications informatiques développées.

5.3. Notre étude expérimentale

L'adoption, par l'Université Algérienne, d'un support technologique assistant l'apprentissage collaboratif à distance de la programmation informatique doit être motivée. Parmi les points de motivation que nous qualifions d'importants, dans ce contexte, est celui de l'efficacité de la technologie, notamment en termes d'accompagnement des apprenants dans leur cursus éducatif. D'autre part, l'efficacité d'accompagnement peut être considérée comme un des facteurs clés permettant d'influencer, chez l'apprenant, l'intention d'adopter la technologie *Groupware* lors de son apprentissage. En effet, l'étude des effets ayant une influence positive sur l'intention d'acceptation du *PBPCLG* apparaît utile dans cette thèse. Cette section vise ainsi à présenter notre étude expérimentale menée, à l'Université d'Oum El bouaghi, pour mesurer, chez les apprenants en informatique, l'intention d'adopter *PBPCLG* pour l'apprentissage de la programmation informatique. Nous avons également opté pour le modèle *UTAUT* pour mener de telle étude.

5.3.1. Développement des hypothèses

UTAUT est initialement concentré sur sept constructions totalement indépendantes comme une *Espérance de Performance (PE)*, *Espérance d'Effort (EE)*, *Influence Sociale (SI)* et les *Conditions de Facilitation (FC)*, *l'auto-efficacité de l'ordinateur*, *l'anxiété* et *l'attitude face à l'utilisation de la technologie* [Venkatesh et al. 03]. Néanmoins, les trois derniers concepts ont été considérés comme négligeables, alors que les quatre autres sont qualifiés d'importants du fait qu'ils ont été largement utilisés dans de nombreuses études [Boonsiritomachai et Pitchayadejanant 17] [Tosuntaş et al. 15] [Ghalandari 12] [Marchewka et al. 07].

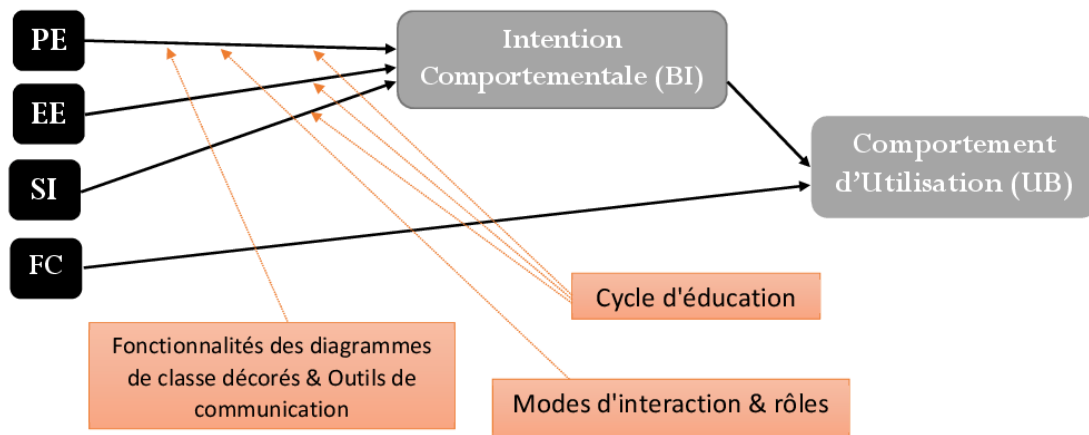


Figure 5. 6 : Modèle de recherche UTAUT modifié.

Les hypothèses privilégiant le modèle UTAUT sont généralement adaptées pour mieux correspondre au contexte de l'étude [Curtis et al. 10] [Duyck et al. 08] [Venkatesh et al. 11]. Par conséquent, nous avons modifié le modèle initial d'UTAUT pour qu'il soit bien adapté à notre contexte d'étude (figure 5.6.). Ainsi, les hypothèses que nous avons formulées sont bien détaillées dans la suite de cette section.

5.3.1.1. Performance Espérée (PE)

Selon Davis (1989), la Performance Espérée (PE) (*en anglais PE : Performance Expectancy*) est « le degré auquel une personne croit que l'utilisation d'un système particulier pourrait renforcer sa performance au travail ». PE tient en compte du degré de croyance d'un individu quant au gain de performance remporté en adoptant une technologie. En d'autres termes, plus l'individu en est convaincu de performances obtenues par la technologie, plus forte sera l'intention de l'adopter.

En fait, notre *Groupware* ne peut être accepté par les apprenants au sein des Universités Algériennes sauf si ces derniers perçoivent ses performances par rapport aux besoins de leur apprentissage, et s'ils sont convaincus que ce *Groupware* va vraiment les assister pour

atteindre un apprentissage efficace. Nous prévoyons un effet direct et positif de *PE* sur l'intention d'adoption du *PBPCLG* pour assister l'apprentissage de la programmation informatique. C'est ainsi que nous avons formulé l'hypothèse suivante :

H1 : *PE a une relation significative et positive avec l'intention d'accepter le PBPCLG pour l'apprentissage de la programmation.*

Les différents rôles adoptés par *PBPCLG* pour être attribués aux apprenants pendant le déroulement des sessions d'apprentissage auraient aussi une influence positive sur le comportement d'intention. De plus, les modes d'interaction intégrés dans *PBPCLG* peuvent constituer un autre moyen important favorisant la reconstruction de l'interaction naturelle à travers des mécanismes virtuels. Ces modes auraient, sans aucun doute, des influences positives sur le comportement d'intention d'utilisation du *PBPCLG*. En fait, deux autres hypothèses sont formulées comme suit :

H1.1 : *Les modes d'interaction intégrés modèreront positivement l'influence du PE sur le comportement d'intention dans l'apprentissage de la programmation en utilisant PBPCLG.*

H1.2 : *Les rôles définis modèreront positivement l'influence de PE sur le comportement d'intention dans l'apprentissage de la programmation en utilisant PBPCLG.*

Les outils de communication ainsi que les fonctionnalités liées au *DCD* peuvent jouer un rôle primordial dans la coordination des activités des différents apprenants. Par conséquent, l'utilisation de ces outils et ces fonctionnalités auraient des effets positifs sur le comportement d'intention. De ce fait, nous avons considéré les deux hypothèses suivantes :

H1.3 : *les outils de communication incorporés dans PBPCLG modèreront positivement l'influence de PE sur le comportement d'intention dans l'apprentissage de la programmation en utilisant notre Groupware.*

H1.4 : *Les fonctionnalités associées à la gestion du DCD modèreront positivement l'influence du PE sur le comportement d'intention dans l'apprentissage de la programmation en utilisant PBPCLG.*

5.3.1.2. Effort Espéré (EE)

Selon *Venkatesh et al.* (2003), l'effort espéré (en anglais *EE* : *Effort Expectancy*) est une construction liée la simplicité d'utilisation d'une technologie donnée. Nous indiquons *EE* en tant que critère de mesure dans lequel un apprenant ou un tuteur pense que l'utilisation d'un *Groupware* dans l'apprentissage est facile. Dans le contexte de l'apprentissage de la programmation informatique, nous supposons que les participants analysent leurs attentes quant au degré de simplicité ou de complexité de l'utilisation de notre *Groupware*, et que cette simplicité observée influence leur intention d'apprendre en

utilisant le système *PBPCLG*. Ceci nous a incités de développer l'hypothèse de recherche suivante H2 :

H2 : *EE a un lien significatif et positif avec l'intention d'adopter l'apprentissage de la programmation via PBPCLG.*

5.3.1.3. *Influence Sociale (SI)*

L'influence sociale (en anglais *SI* : *Social Influence*) regroupe les normes subjectives, les facteurs sociaux et l'attente de valorisation. Selon *Venkatesh et al.* (2003), les facteurs sociaux, notamment les personnes ayant une importance significative à l'individu, exercent une certaine influence sur son comportement d'intention. Dans ce sens, il est désirable d'apprécier l'influence de certaines relations sociales de façon distinctive, en tenant compte des supérieurs hiérarchiques et l'attente de valorisation.

En fait, nous supposons que *SI* a un impact sur l'intention d'utilisation du *PBPCLG*. Ceci est valable du fait que les enseignants, les gestionnaires des facultés, les responsables du ministère de l'enseignement supérieur et de la recherche scientifique (*MESRS*) et d'autres personnes peuvent influencer le comportement d'intention de l'apprenant par rapport à l'utilisation de notre *Groupware* lors de l'apprentissage de la programmation informatique. Par conséquent, les hypothèses suivantes sont posées :

H3 : *SI a une relation positive et significative avec le comportement d'intention d'adoption du PBPCLG dans l'apprentissage de la programmation informatique.*

H3.1 : *L'influence des enseignants a un impact positif sur le comportement d'intention d'adoption du PBPCLG dans l'apprentissage de la programmation informatique.*

H3.2 : *L'influence des administrateurs de faculté a un impact positif sur le comportement d'intention d'adoption du PBPCLG dans l'apprentissage de la programmation informatique.*

H3.3 : *L'influence des décideurs du MESRS a un impact positif sur le comportement d'intention d'adoption du PBPCLG dans l'apprentissage de la programmation informatique.*

5.3.1.4. *Conditions Facilitantes (FC)*

Les conditions Facilitantes (en anglais *FC* : *Facilitating conditions*) représentent le degré auquel une personne pense que les ressources disponibles comme les infrastructures organisationnelles et techniques sont indispensables et encouragent l'adoption d'une technologie innovante [Venkatesh et al. 03][Teo 09][Terzis et Economides 11]. Précisément, cette construction comprend des facteurs de ressources (par exemple, le temps de formation) ou des facteurs technologiques (par exemple, l'efficacité du système) [Lu et al. 08]. *FC* est

un facteur essentiel déterminant le comportement d'intention d'adoption d'une technologie spécifique, car lorsque les utilisateurs n'ont pas assez de temps pour se former, ou lorsqu'ils rencontrent des problèmes de fonctionnement de ladite technologie, ils auront moins l'intention d'y utiliser [Huang 15]. Dans cette étude, nous avons anticipé que FC aurait des influences positives sur le comportement d'intention d'adoption du PBPCLG dans l'apprentissage de la programmation informatique. Ainsi, l'hypothèse suivante est formulée :

H4 : *Les FC ont un impact sur le comportement d'intention dans l'apprentissage de la programmation informatique via PBPCLG.*

Étant donné que le contexte de notre étude, les apprenants possèdent souvent une bonne expérience en termes d'utilisation de l'outil informatique, le modérateur nommé "expérience" du modèle d'UTAUT semble moins important. Cependant, nous pouvons définir la variable modératrice : *le cycle d'enseignement* (Licence ou Master).

Ainsi, l'hypothèse suivante est formulée comme suit :

H4-1 : *le cycle d'éducation a un impact positif sur le comportement d'intention d'adoption du PBPCLG dans l'apprentissage de la programmation informatique.*

5.3.2. Méthodologie

Rappelant que l'objectif principal de ce chapitre est de présenter notre étude descriptive qui se base sur les expérimentations, dans un contexte réel, de l'apprentissage de la programmation informatique. Cette étude est également menée, à l'Université d'Oum El bouaghi, afin de mesurer les éléments clés qui influencent le comportement d'intention des apprenants par rapport à l'adoption de notre *Groupware PBPCLG* dans l'apprentissage de ladite programmation informatique. Dans cette perspective, Il semble utile de choisir tout d'abord une méthodologie de recherche adéquate. En effet, avant d'analyser les données correspondantes à nos résultats obtenus, il est important de considérer au préalable la méthode de collecte de données, la population et la procédure expérimentale. Les détails de cette démarche sont bien discutés dans le reste de section.

5.3.2.1. Collecte des données

Les données pertinentes constituent l'ensemble le plus important dans toute étude expérimentale analysant et mesurant les performances de n'importe quelle application informatique. De plus, les applications de type *Groupware* représentent l'une des applications les plus concernées par des études expérimentales. Ces dernières se basent principalement sur la phase d'analyse de données collectées. Notant aussi que sans données pertinentes, cette phase ne pouvait pas établir de bons rapports entre un *Groupware* et ses résultats. C'est ainsi que dans notre contexte de travail, l'adoption d'une

méthode de collecte de données semble assez importante. Plusieurs méthodes peuvent être utilisées à ces fins parmi lesquelles nous distinguons celle qui s'appuie sur l'élaboration de questionnaires. Ainsi, l'investigation par questionnaire apparaît être le moyen privilégié qui nous permettrait la collecte des données utiles.

	Valeur	Fréquence	%
Sexe	Male	56	47
	Femelle	63	53
Cycle d'éducation	License	65	55
	Master	54	45
Age	19-21 ans	54	45
	22-25 ans	60	51
	>25 ans	05	04
Expérience en programmation	débutant	28	24
	Moyen	68	57
	Expérimenté	23	19
Duré de sessions	<1 h	12	52
	1-2 h	08	35
	2-3 h	03	13
Complexité du problème	facile	06	26
	Moyen	10	44
	difficile	07	30
Problème résolu	entièrement	20	87
	non entièrement	03	13
Université	Oum el Bouaghi	119	100

Tableau 5. 1 : Questionnaire quantitative adopté par notre étude.

De nombreuses formes peuvent être utilisées afin de classer les données d'un questionnaire. Les chercheurs optent souvent à distinguer les données *quantitatives* des données *qualitatives*. Les données quantitatives sont des observations numériques, alors que les données qualitatives sont des interrogations correspondant à des catégories (p. ex., pour l'apprenant, *Licence ou Master*, ou pour le sexe des participants, *Masculin ou Féminin*). En effet, le tableau 5.1 résume les données qualitatives de notre étude.

Quant aux données quantitatives, nous avons conçu un questionnaire qui se compose de plusieurs questions organisées, selon les constructions que nous avons sélectionnées du modèle *UTAUT*, en plusieurs catégories (Tableau 5.2).

D'autre part, les apprenants sont invités, après chaque session d'apprentissage, à préciser sur une échelle de *Likert* à cinq points (allant de "1 : pas du tout d'accord" à "5 : tout à fait d'accord") dans quelle mesure ils étaient d'accord avec l'annonce.

Le tableau 5.2 décrit clairement le questionnaire que nous avons élaboré pour collecter les données pertinentes à notre étude. En outre, afin de bien percevoir mieux l'utilité du *PBPCLG*, et d'en tirer les conclusions appropriées, d'autres données jugées utiles ont été recueillies en utilisant des observations.

<i>Const</i>	Élément de Questionnaire
PE	<p>PE1: L'utilisation de PBPCLG m'aide.</p> <p>PE2: L'utilisation de PBPCLG me permettra d'apprendre la programmation informatique plus rapidement.</p> <p>PE3: L'utilisation de PBPCLG dans son intégralité me permettra de participer davantage à la résolution des problèmes de programmation informatique.</p> <p>PE4: L'utilisation des modes d'interaction incorporés augmentera ma productivité dans la résolution de problèmes de programmation informatique.</p> <p>PE5: L'utilisation des rôles définis augmentera ma productivité dans la résolution des problèmes de programmation informatique.</p> <p>PE6: L'utilisation des fonctionnalités associées à la gestion des diagrammes de classes décorés augmentera ma productivité dans la résolution des problèmes de programmation informatique.</p> <p>PE7: L'utilisation des outils de communication augmentera ma productivité dans la résolution des problèmes de programmation informatique.</p>
EE	<p>EE1: Je pense que le PBPCLG est facile à utiliser.</p> <p>EE2: Apprendre à utiliser le PBPCLG est facile pour moi.</p> <p>EE3: Je pense que les outils de collaboration incorporés dans PBPCLG sont clairs et compréhensibles.</p> <p>EE4: Il sera facile pour moi de devenir compétent pour utiliser PBPCLG.</p>
SI	<p>SI1: Les personnes qui influencent mon comportement (par exemple, les enseignants, les administrateurs de la faculté) pensent que je devrais utiliser PBPCLG.</p> <p>SI2: Les personnes qui sont importantes pour moi (par exemple, des collègues) pensent que je devrais utiliser PBPCLG.</p> <p>SI3: les décideurs du ministère algérien de l'enseignement supérieur pensent que je devrais utiliser PBPCLG.</p>
FC	<p>FC1: Je dispose des ressources nécessaires pour utiliser le PBPCLG.</p> <p>FC2: J'ai les connaissances nécessaires pour utiliser PBPCLG.</p> <p>FC3: <i>PBPCLG est compatible avec les autres systèmes que j'utilise.</i></p> <p>FC4: Un administrateur de PBPCLG est disponible pour m'aider en cas de difficultés d'utilisation.</p>
BI	<p>BI1: J'ai l'intention d'utiliser PBPCLG avec mes collègues pour l'apprentissage collaboratif de la programmation informatique.</p> <p>BI2: Je prévois d'utiliser PBPCLG avec mes collègues pour l'apprentissage collaboratif de la programmation informatique.</p> <p>BI3: Je prévois d'utiliser PBPCLG pour acquérir des compétences en programmation informatique.</p>
UB	<p>UB1: J'utilise le PBPCLG lorsque j'apprends la programmation informatique.</p> <p>UB2: J'utilise le PBPCLG lorsque j'apprends la programmation informatique avec mes collègues.</p> <p>UB3: J'utilise le PBPCLG lorsque j'apprends la programmation informatique avec des professeurs.</p>

Tableau 5. 2 : Aperçu des constructions de recherche utilisées dans le questionnaire.

5.3.2.2. Échantillonnage et procédure expérimentale

Avant de préciser la nature des individus que nous avons sélectionnés pour mener nos expérimentations, il semble utile de définir au préalable le concept d'échantillon. Un échantillon tel qu'il est défini par Savard (1978) représente un groupe proportionnellement petit et sélectionné scientifiquement de sorte à représenter le plus fidèlement possible une population donnée. Donc, au lieu de considérer toute la

population, nous étudions uniquement un sous-ensemble de cette dernière qui est considérée comme un élément représentatif. D'autre part, plusieurs méthodes d'échantillonnage peuvent être utilisées pour sélectionner l'échantillon sur lequel l'étude expérimentale va se dérouler afin de dégager les bonnes conclusions à généraliser par la suite, avec un certain degré d'incertitude, sur toute la population. Dans cette étude, afin d'avoir un échantillon plus représentatif de la population, nous avons opté pour la méthode d'échantillonnage aléatoire stratifiée. Le principe d'une telle méthode stipule qu'une population doit être au préalable subdivisée en sous-ensembles homogènes baptisés *strates*. Ces derniers sont spécifiés selon un ou plusieurs critères. Ensuite, dans chaque strate, on choisit, de manière aléatoire, des individus qui vont constituer des sous-échantillons aléatoires simples.

En effet, notre étude a considéré les étudiants en informatique à l'Université d'Oum El Bouaghi. Dans le Département d'informatique d'une telle université, nous considérons deux catégories d'étudiants représentant les *strates* à savoir : les *Licences* et les *Masters*. Ainsi, notre enquête a été menée auprès de 119 étudiants dont 55% sont des étudiants en Licence et 45% en Master dont environ 47% sont de sexe *Masculin*. Selon la classification décrite sur le tableau 5.1, nous remarquons que la majorité des interrogés appartiennent aux classes d'âge (19-21 et 22-25) ans, tandis que le reste de notre échantillon (04%) sont âgés plus de 25 ans.

Au début, cet échantillon a pris part à une formation accélérée sur le principe de fonctionnement de notre *Groupware*. Par ailleurs, du fait que *PBPCLG* permet d'assister l'apprentissage collaboratif de la programmation informatique, les apprenants (étudiants) de chaque *strate* sont subdivisés en groupes de trois ou quatre individus. Chaque groupe est assigné à une session d'apprentissage, est tuteuré par un enseignant spécialiste. Le tuteur prépare, au préalable, sa session en proposant un problème de programmation autour duquel les apprenants vont interagir afin d'apprendre les techniques de la programmation en y proposant une solution.

A la fin de chaque session, les apprenants participants sont invités à répondre au questionnaire (tableau 5.2).

5.3.2.3. Analyse des résultats

Nous discutons, dans cette section, la partie analyse de données de l'étude. Par conséquent, nous avons opté pour une approche dite *PLS* (*Projection to Latent Structure*). Cette approche a été largement utilisée pour analyse des applications très variées, en marketing, en sociologie, en management des systèmes d'information, en modèle d'acceptation de la technologie ou en chimie.

PLS permet d'analyser les résultats selon deux modèles : *Modèle de mesure* et *Modèle structurel*. Les données recueillies ont fait l'objet de traitements statistiques via les logiciels SPSS et AMOS.

5.3.2.3.1. *Modèle de mesure*

Un modèle de mesure (autrement dit *modèle externe*) permet de relier les variables observées aux variables latentes qui leur sont associées. En statistique, une variable latente (VL) est une variable non observable (ou un construit) qui peut être déduite d'autres variables observées appelées variables manifestes (VM).

Notre modèle de mesure décrit par la figure 5.1 doit être vérifié et validé. Pour ce faire, nous avons considéré deux paramètres : les *charges factorielles* des items et la *mesure de fiabilité*. Ainsi, un item est fiable si sa *charge factorielle* (*Charge Fact.*) est supérieure à 0,5. Comme le montre le tableau 5.3, toutes les valeurs des charges factorielles sont supérieures à 0,7 et toutes les valeurs de l'écart type (*Std. D*) sont petits, ce qui signifie qu'il y a peu d'écart dans les scores de chaque item.

Constructions	Item	Charge Fact.	Std. D
PE	PE1	0.727	0.751
	PE2	0.708	0.659
	PE3	0.820	0.691
	PE4	0.845	0.731
	PE5	0.621	0.881
	PE6	0.613	0.756
	PE7	0.699	0.717
EE	EE1	0.787	0.661
	EE2	0.878	0.732
	EE3	0.747	0.688
	EE4	0.600	0.818
SI	SI1	0.628	0.792
	SI2	0.816	0.940
	SI3	0.850	0.769
FC	FC1	0.714	0.699
	FC2	0.789	0.757
	FC3	0.783	0.766
	FC4	0.862	0.732
BI	BI1	0.928	0.751
	BI2	0.860	0.779
	BI3	0.754	0.789
UB	UB1	0.735	0.834
	UB2	0.774	0.887
	UB3	0.865	0.838

Tableau 5. 3 : Charges factorielles des items.

D'autre part, la fiabilité de cohérence interne peut être vérifiée par deux autres mesures qui sont l'*Alpha de Cronbach* et la *fiabilité composite* [Chin 98]. Selon *Hair et al. (2006)*, les résultats, présentés dans le tableau 5.4, montrent que toutes les valeurs de la fiabilité composite et de l'*Alpha de Cronbach* sont supérieures aux seuils minimums qui sont respectivement de 0,6 et 0,7.

Fiabilité des mesures		
Constructions	Fiabilité Composite	Alpha de Cronbach
PE	0.884	0.877
EE	0.843	0.826
SI	0.812	0.800
FC	0.868	0.866
BI	0.886	0.879
UB	0.835	0.840

Tableau 5. 4 : Validité convergente et la fiabilité des mesures.

Quant à l'évaluation de la validité convergente, nous avons calculé pour chaque construction la *Variance Moyenne Extraite* (en anglais *AVE : Average Variance Extracted*), qui devait être supérieure au seuil conventionnel fixé à la valeur 0,5 [Hair et al. 2006]. En outre, l'évaluation de la validité discriminante a été réalisée en examinant que la racine carrée d'*AVE* de chaque construction. Elle devrait être supérieure à la corrélation partagée entre une construction et les autres constructions [Fornell et Lacker 81]. Comme la montre le tableau 5.5, toutes les valeurs d'*AVE* sont comprises entre 0,524 et 0,723 (supérieures à 0,5) et les racines carrées des différentes valeurs d'*AVE* sont également supérieures aux coefficients de corrélation. Tous nos résultats satisfont les critères de *Fornell & Larcker (1981)*. Ils sont en conséquence conformes à la mesure de la validité discriminante.

Construction	Validité convergente	Validité discriminante					
	AVE	PE	EE	SI	FC	BI	UB
PE	0.524	0.724					
EE	0.577	0.124	0.760				
SI	0.594	0.221*	0.118	0.770			
FC	0.622	0.32**	0.215*	0.128	0.789		
BI	0.723	0.295**	0.201*	0.054	0.145	0.850	
UB	0.629	0.204	0.144	0.085	0.305***	0.321***	0.793

Note. ***p < 0,001; ** p < 0,01 ; * p < 0,05 ; La racine carrée de AVE se trouve sur la diagonale.

Tableau 5. 5 : Validité discriminante des mesures.

Par ailleurs, pour étudier les modérateurs définis dans notre modèle (*sexe de l'apprenant & cycle d'éducation*), nous avons utilisé la méthode de *test de Student* (autrement dite *test t*)

proposé par *William Gosset* qui, en 1908. Un *test t* est adopté pour examiner si les ensembles de données *Masculines* et *féminines* étaient significativement différents les uns des autres, ainsi que pour étudier s'il y avait des différences significatives entre les moyennes des personnes interviewées des deux cycles d'éducation (*Licence* et *Master*).

5.3.2.3.2. *Modèle structurel*

Un modèle structurel (autrement dit *modèle interne*) est un modèle qui sert à relier des variables latentes dites endogènes à d'autres variables latentes. Cette liaison est effectuée en utilisant des équations linéaires.

Quant à notre modèle, nous avons tout d'abord débuté par tester l'adéquation des données collectées à notre modèle théorique proposé (figure 5.1). Dans cette perspective, nous avons opté pour une variété d'indices d'ajustement comme : *GFI* (*Goodness of Fit Index*) et *AGFI* (*Adjusted GFI*). Indicateur de la part relative de la covariance expliquée par le modèle. Ces indicateurs varient théoriquement entre 0 et 1 et selon *Seyal et al.* (2002) et *Scott* (1995) *GFI* et *AGFI* devraient être supérieurs à 0,80 ; *CFI* (*Bentler comparative fit index*) selon *Bentler & Bonett* (1980) devrait être supérieur 0,9 ; *RMSEA* (*Root Mean Square Error of Approximation*), qui est un indice d'ajustement dont le meilleur ajustement est celui qui correspond à une valeur de zéro pour *RMSEA*. Selon *Browne & Cudeck* (1993) cet indice devrait être inférieur ou égal à la valeur 0.08. D'autre part, la majorité des travaux de recherche sont d'accord pour stipuler qu'un *RMSEA* supérieur ou égal à 0,1 indique un mauvais ajustement. D'autres indices sont aussi calculés afin de bien mener ce type de test à savoir : *RMR*, *TLI*, *NFI*, et X^2/df . Le tableau 5.6 présente clairement les valeurs obtenues en termes de ces indices ainsi que leurs valeurs souhaitées. Parmi les huit indices calculés, nous avons cinq qui indiquent un bon ajustement ce qui démontre la qualité de l'ajustement global de notre modèle théorique. Les trois autres indices peuvent être considérés acceptables dans la mesure où leurs valeurs sont très proches des valeurs recommandées. Dans cette vision, nous pouvons juger que l'ajustement du modèle de mesure est satisfaisant.

Indices	Résultats	Valeurs recommandées	Références
GFI	0.803	>0.8	(<i>Seyal et al., 2002</i>)
AGFI	0.755	>0.8	(<i>Scott, 1995</i>)
CFI	0.900	>0.9	(<i>Bentler&Bonett, 1980</i>)
NFI	0.773	>0.9	(<i>Bentler&Bonett, 1980</i>)
TLI	0.882	>0.9	(<i>Hu &Bentler, 1999</i>)
RMR	0.051	<0.08	(<i>Hair et al., 2006</i>)
RMSEA	0.072	<0.08	(<i>Loehlin, 2004</i>), (<i>Browne&Cudeck, 1993</i>)
X^2/df	1.615	<3	(<i>Hair et al., 2009</i>), (<i>Bagozzi& Yi, 1988</i>)

Tableau 5. 6 : *Indices d'ajustement.*

Par ailleurs, l'évaluation du modèle structurel est d'une importance extrême dans cette étude, notamment pour évaluer aussi bien l'impact statistique des hypothèses que l'aptitude du modèle à expliquer la variance des variables dépendantes. De plus, en s'appuyant sur la suggestion de Chin (1998) stipulant que « l'estimation des coefficients de détermination R^2 et l'appréciation des coefficients de chemin (en anglais path coefficients) constituent deux critères importants parmi ceux qui peuvent être utilisés pour évaluer un modèle structurel », nous avons opté pour ces deux types de coefficients afin mener l'évaluation de notre modèle.

	Estimate	S.E.	C.R.	P	P < 0.05
BI ← PE	0.376	0.131	2.879	0.004	significativement attendu
BI ← SI	0.049	0.093	0.527	0.598	statistiquement insignifiant
BI ← EE	0.270	0.133	2.027	0.043	significativement attendu
UB ← BI	0.328	0.101	3.249	0.001	significativement attendu
UB ← FC	0.345	0.114	3.029	0.002	significativement attendu

Tableau 5.7 : Relation entre les variables exogènes et endogènes.

Le R^2 est un indice qui permet de mesurer le pouvoir explicatif d'un modèle. Donc, il permet de rendre compte de la contribution de chaque *variable explicative (indépendante)*³ dans la prévision de la *variable expliquée (dépendante)*⁴. D'autre part, les *coefficients de chemin*⁵ sont estimés dans cette étude afin d'évaluer l'impact statistique des hypothèses. Dans cette dialectique, les résultats de cette évaluation sont bien présentés sur aussi bien le tableau 5.7 que la figure 5.7. En fait, PE a influencé significativement BI (0,29, $p < 0,05$), EE a influencé significativement BI (0,20, $p < 0,05$), et BI et FC ont influencé significativement sur UB (0,32, $p < 0,05$) et (0,31, $p < 0,05$). Le reste des relations étaient statistiquement insignifiantes. Tous ces résultats montrent que si on améliore PE, EE et FC le comportement d'intention d'adoption du PBPCLG augmente de manière significative.

³ Les *variables explicatives*, appelées également *variables indépendantes*, sont celles que nous utilisons dans le but d'expliquer, de décrire ou de prédire la ou les *variable(s) dépendante(s)*.

⁴ La *variable dépendante*, c'est la *variable* qui subit l'effet de la *variable indépendante*. Si on compare les hommes et les femmes quant à leur satisfaction au travail dans une usine, la *variable indépendante* c'est le sexe, tandis que la *variable dépendante*, c'est la satisfaction au travail.

⁵ Les *coefficients des chemins* baptisés aussi coefficients de régression multiple standardisés ou coefficients structurels.

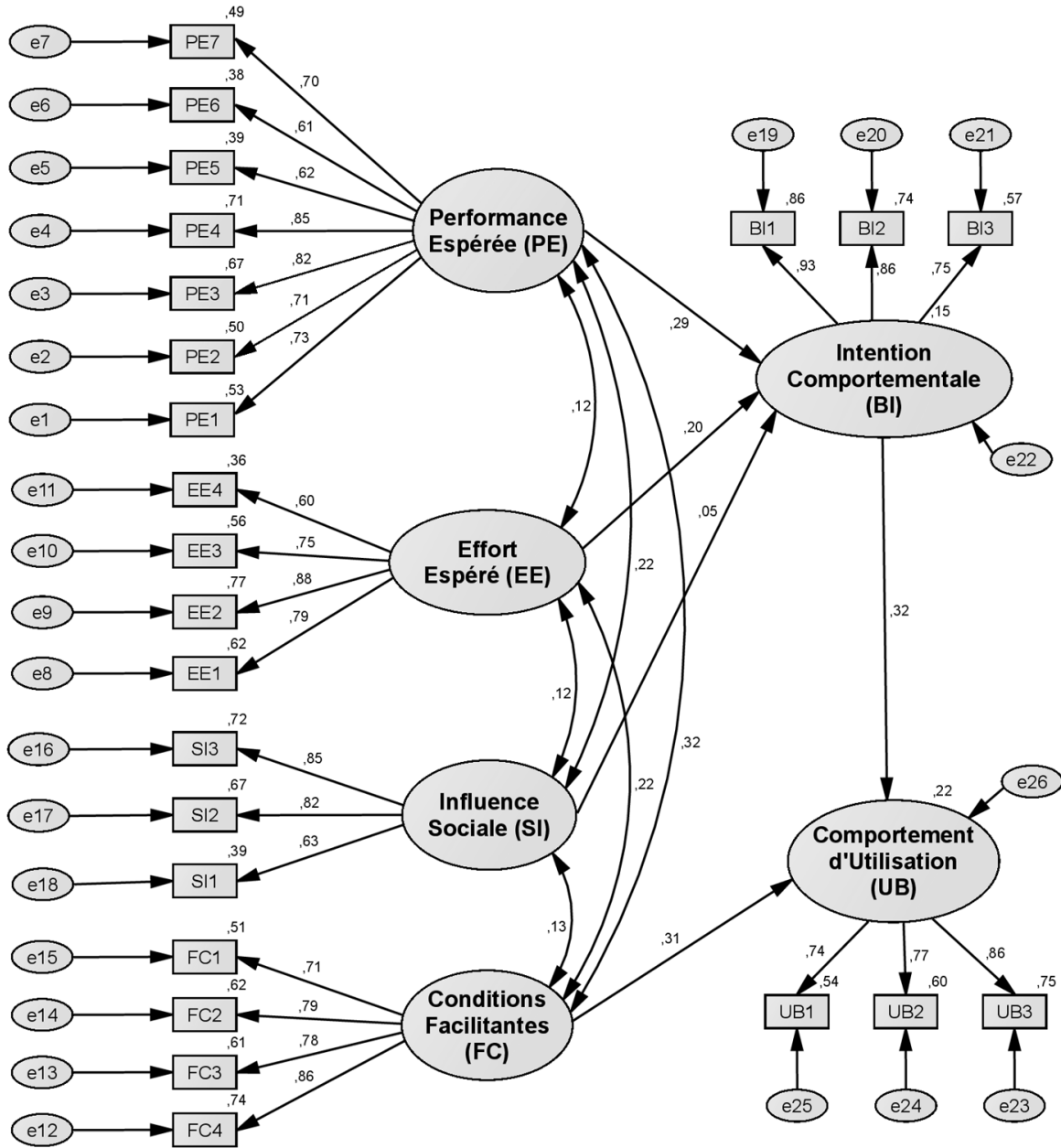


Figure 5. 7 : Résultats de l'analyse de la modélisation structurelle.

5.3.3. Discussion

Selon le modèle *UTAUT*, l'intention comportementale (*BI*) pour adopter une technologie de type *Groupware* comme *PBPLG* lors du déroulement du processus d'apprentissage de la programmation informatique est une tâche de perception qui est utile (*PE*), qui présente une facilité d'utilisation (*EE*), que des personnes significatives ont supposé que les apprenants devraient y utiliser dans leurs séances d'apprentissage de la

programmation (*SI*), et que les moyens disponibles encouragent les apprenants à l'adopter (*FC*).

Les résultats obtenus montrent que parmi les quatre facteurs cités auparavant, *PE* constitue le facteur le plus influent pour prévoir l'intention d'utiliser *PBPCLG*. Les hypothèses *H1.1*, *H1.2*, *H1.3* et *H1.4* présument que le *PE* dépend étroitement de l'amélioration des modes d'interaction de groupe, des supports de communication et des mécanismes de conscience. Par conséquent, nos résultats sont conformes avec ceux de l'étude de *Barron & Sears* qui a confirmé que la recherche dans le domaine d'*ACAO* peut être avancée en continuant à améliorer les caractéristiques de l'interaction de groupe [Barron et Sears 02]. C'est ainsi que ce travail de recherche constitue, bien entendu, une réflexion significative dans ce contexte. Ainsi, nos résultats sont également conformes avec ceux de nombreuses études antérieures où *PE* a plus d'impact que *EE*, *SI* et *FC* [Venkatesh et al. 03] [Phichitchaisopa & Naenna 13].

D'autre part, dans cette étude *PE* est déclarée significative dans la mesure où les apprenants trouvent dans *PBPCLG* un moyen efficace permettant d'assister les interactions de groupe lors du déroulement de l'apprentissage de la programmation informatique, notamment lorsqu'ils surmontent l'obstacle de distance en suivant des séances d'apprentissage indépendamment de leurs localisations géographiques. De plus, ils constatent que *PBPCLG* présente de nombreuses fonctionnalités simples et faciles à utiliser, ce qui explique l'effet significatif d'*EE*. Enfin, nos résultats exposent aussi que *FC* possède un effet sur le comportement d'utilisation (*UB*) du *PBPCLG*. Dans ce sens, nous pouvons juger que pour pouvoir encourager les apprenants et augmenter leur motivation à utiliser une technologie de type *Groupware* comme *PBPCLG*, il est extrêmement important de mettre à leur disposition toutes les ressources humaines et matérielles nécessaires par exemple :

- *Les administrateurs du Groupware doivent être disponibles à tout moment afin de décharger les apprenants et les tuteurs des tâches ardues d'administration du système comme par exemple la gestion des comptes utilisateurs.*
- *Les formations sur l'utilisation du Groupware.*
- *Les responsables de maintenance.*
- ...etc.

Dans cette dialectique, nous pouvons confirmer que le manque de motivation, chez les apprenants, en termes de partage des connaissances peut influencer négativement leur comportement d'intention pour adopter *PBPCLG* dans l'apprentissage de la programmation. Cette affirmation a été appuyée par *Fallery et al.* (2014) qui stipule que « ...si le partage des connaissances n'est pas récompensé, les employés ne sont pas incités à s'y engager ».

Finalement, à travers cette étude on a non seulement pu prouver les impacts directs de *PE*, *EE*, *SI* et *FC* sur *BI*, qui représentent le cœur du modèle *UTAUT*, mais on a aussi pu tirer de bonnes conclusions sur l'impact des variables modératrices comme *Cycle d'éducation* et le *Sexe* sur le comportement d'intention. Bien que les résultats montrent que ces variables n'affectent pas le comportement d'intention d'adoption, ils fournissent quand-même une vision claire en termes de compréhension du comportement des apprenants à l'égard d'une technologie innovante comme *PBPCLG*.

5.4. Conclusion

Dans ce chapitre, nous avons présenté de manière détaillée notre étude expérimentale menée, à l'Université d'Oum El bouaghi, pour mesurer objectivement les effets d'une technologie innovante dite *Groupware* sur le comportement d'intention, chez les apprenants en informatique. En effet, le modèle théorique *UTAUT*, proposé par *Venkatesh et al.* (2003), a été également adopté en ajustant certaines hypothèses pour qu'elles correspondent mieux au contexte de travail de cette thèse.

En fait, nous avons également tenu à apporter toutes les clarifications requises à propos de : la formulation des hypothèses, la méthodologie de recherche adoptée et discussion des résultats. D'autre part, la méthodologie de recherche, qui occupe une place importante dans ce chapitre, a été particulièrement bien détaillée dans la mesure qu'elle spécifie l'approche de collecte de données, l'échantillonnage, la procédure expérimentale, et l'analyse des résultats.

Conclusion générale et perspectives

Dans cette thèse, nous avons proposé une nouvelle approche permettant de soutenir le télé-apprentissage collaboratif de la programmation informatique. Cette approche, qui combine les fondements de l'ACAO avec le principe de la méthode pédagogique dite APP, offre aux apprenants l'opportunité de suivre, à tout moment et indépendamment de leurs localisations géographiques, des séances d'apprentissage collaboratif de la programmation informatique. En effet, un prototype de système *Groupware* baptisé PBPCLG est conçu et mise en œuvre pour concrétiser les idées dégagées par une telle approche. Ainsi, un espace de travail partagé est bien conçu pour permettre aux apprenants avec leur tuteur d'interagir autour d'un problème de programmation commun en proposant des solutions sous forme de programmes sources. L'accent est particulièrement mis sur les aspects « *interaction de groupe* » et « *conscience de groupe* », dans la mesure qu'ils représentent les facteurs clés de la réussite de la méthode pédagogique APP. D'autre part, en s'appuyant sur l'idée d'Isohätälä et al. (2017) stipulant que " *l'interaction de groupe encourage largement les apprenants à partager et à exprimer une variété de points de vue et en conséquence à élargir leur réflexion au-delà des compétences personnelles*", nous avons intégré dans PBPCLG une variété de fonctionnalités favorisant l'interaction de groupe et la conscience de groupe. Parmi ces fonctionnalités, nous citons à titre indicatif : *les mécanismes régissant le fonctionnement de l'espace de travail partagé, les modes d'interaction collaborative et les rôles et leur attribution, les outils de conversation, les mécanismes de notification, etc.*

Au cours d'une session de télé-apprentissage, les apprenants peuvent changer des rôles, échanger dynamiquement des fragments de code source et interagir à travers les quatre modes d'interaction proposés. Cette flexibilité est prescrite par le besoin de rendre *PBPCLG* plus dynamique possible en obéissant aux spécificités de l'apprentissage collaboratif en présentiel de la programmation informatique. De plus, l'intégration, dans *PBPCLG*, des mécanismes d'échange d'informations vocales et textuelles permet, bien entendu, aux apprenants de paralléliser les communications avec la réalisation des tâches. Ceci optimise de manière considérable les durées des sessions d'apprentissage.

Dans le but de permettre aux apprenants de suivre, à tout moment, l'état d'avancement de la session d'apprentissage, notamment pour apercevoir une vision globale sur le code source en cours de développement, nous avons inclus dans l'espace partagé un espace de modélisation graphique nommé *DCD*. Ainsi, nous avons largement motivé l'utilité de ce *DCD*, dans la mesure où il permet d'offrir aux apprenants certaines opportunités, notamment pour augmenter leurs consciences à propos de l'activité partagée. Au niveau de l'*IHM*, l'association, à des fragments de code source, de certaines informations graphiques complémentaires comme *les étiquettes, les couleurs, les icônes, ...* facilite la tâche aux apprenants, tout en leur permettant de comprendre intuitivement le statut de chaque fragment ainsi que le mode d'interaction qui lui est appliqué.

Par ailleurs, dans l'approche proposée, le tuteur ne joue qu'un simple rôle de *facilitateur* de l'apprentissage. Ceci crée, chez les apprenants, un certain sentiment d'autonomie, notamment en termes d'acquisition des connaissances en bénéficiant de la synergie des compétences du groupe.

Enfin, l'aspect expérimental de l'approche proposée semble assez primordial dans ce contexte ce qui nous a incité à consacrer une partie importante de la thèse à l'étude expérimentale d'évaluation de l'intention des apprenants à l'égard de l'adoption d'une technologie *Groupware* comme *PBPCLG*. En fait, nous avons utilisé le modèle *UTAUT* afin de mener plusieurs expérimentations faisant participer des étudiants de *Master* et *Licence* de l'Université *d'Oum el Bouaghi*- Algérie. Les bonnes conclusions sont également bien discutées à la fin du dernier chapitre.

Bien que cette recherche nous a permis d'enrichir nos connaissances, surtout en termes de compréhension du comportement d'intention des apprenants à l'égard d'une technologie innovante comme *PBPCLG*, certaines améliorations peuvent être envisagées en perspective. Par exemple, des recherches supplémentaires peuvent être planifiées afin d'étudier les impacts de l'approche *Groupware* sur la qualité de l'apprentissage de la programmation informatique.

Références

[Abdallah 09]	Abdallah, F. (2009). Meta-modélisation pour décrire et instrumenter une situation d'apprentissage de pédagogie par projet collectif . <i>Thèse de doctorat en informatique. Université du Maine, Le Mans.</i>
[Ajzen 91]	Ajzen, I. (1991). The theory of planned behavior . <i>Organizational Behavior and human Decision Processes</i> , 50, 179-211.
[Akgün et Akkoyunlu 13]	Akgün, E. & Akkoyunlu, B. (2013). Instruction of computer supported collaborative learning environment and students' contribution quality . <i>Procedia-Social and Behavioral Sciences</i> , 93, 1489-1492.
[Allen et al. 02]	Allen, E. Cartwright, R. and Stoler, B. (2002). Drjava: A lightweight pedagogic environment for java . <i>In Pro-ceedings of ACM SIGCSE Symposium.</i>
[Anzieu et Martin 68]	Anzieu, D. Martin, J.-Y. (1968). La dynamique des groupes restreints, le psychologue . <i>Presses Universitaires de France, première édition en mai 1968, dernière édition en décembre 1982.</i>
[Ardoino 66]	Ardoino, Jacques. (1966). Communications et relations humaines , Paris, <i>Institut d'administration des entreprises de l'Université de Bordeaux.</i>
[Atmani et Stainier 00]	Atmani, N. et Stainier, N. (2000). ABC : Du Problem-Based Learning (PBL) ou apprentissage par problèmes (APP) . Chaire de pédagogie et didactique de l'École polytechnique fédérale de Lausanne, <i>app.cegep-steyoy.qc.ca/fileadmin/user_upload/APP/pdf/ABC__Lausanne.pdf</i> , consulté le 20 mars 2016.
[Balacheff et al. 97]	Balacheff, N. Baron, M. Desmoulin, C. Grandbastien, M. Vivet, V. (1997). Conception d'environnements interactifs d'apprentissage avec ordinateur - Tendances et perspectives . <i>PRC-GDR Intelligence Artificielle</i> , p. 315-337.
[Bannan 02]	Bannan-Ritland, B. (2002). Computer-Mediated Communication (CMC), e-Learning and Interactivity: A Review of the Research . <i>Quarterly Review of Distance Education</i> 3 (2): 161-179.
[Bannon 94]	Bannon, L. (1994). Issues in Computer Supported Collaborative Learning . In C. O'Malley (ed.) <i>Computer Supported Collaborative Learning. Proceedings of NATO Advanced Research Workshop on Computer Supported Collaborative Learning, Aquafredda di Maratea, Italy, Sept. 24-28, 1989. NATO ASI Series, SERS F Vol.128. Berlin: SpringerVerlag. ISBN 3-540-57740-8.</i>
[Bannon et al. 91]	Bannon, L. Robinson, M. m Schmidt, K. (1991). <i>ECSCW'91 : Proceedings of the second European conference on computer-supported cooperative work. Kluwer Academic, Dordrecht, the Netherlands.</i>

[Bany et Johnson 71]	Bany, Mary A. & Johnson, Lois V. (1971). Dynamique des groupes et éducation (le groupe-classe) , Paris, Dunod.
[Baron et al. 91]	Monique, B. Gras, R. Nicaud, J-F. (1991). Actes des deuxièmes journées EIAO de Cachan . <i>Les Éditions de l'École Normale Supérieure de Cachan</i> .
[Barron et Sears 02]	Barron, B., Sears, D. (2002). Advancing Understanding of Learning in Interaction: How Ways of Participating can Influence Joint Performance and Learning . In <i>Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community</i> (pp. 593–594). <i>International Society of the Learning Sciences</i> .
[Bawamohiddin et Razali 14]	Bawamohiddin, A. B. and Razali, R. (2014), January. The Suitability of Adopting Problem-based Learning in Computer Science Education . In <i>International Conference on Infocomm Technologies in Competitive Strategies (ICT). Proceedings</i> (p. 64). <i>Global Science and Technology Forum</i> .
[Bawamohiddin et Razali 17]	Bawamohiddin, A. B. and Razali, R. (2017). Problem-based Learning for Programming Education . <i>International Journal on Advanced Science, Engineering and Information Technology</i> 7 (6): 2035–2050. doi:10.18517/ijaseit.7.6.2232
[Beaudouin-Lafon 92]	Beaudouin-Lafon, M. Karsenty, A. (1992), December. Transparency and awareness in a real-time groupware system . <i>UIST '92: Proceedings of the 5th annual ACM symposium on User interface software and technology</i> , Pages 171–180. https://doi.org/10.1145/142621.142646
[Bellström et Kilbrink 09]	Bellström, P. and Kilbrink, N. (2009). Problem-Based Learning in a Programming Context—Planning and Executing a Pilot Survey on Database Access in a Programming Language . In <i>Information Systems Development</i> , edited by G. Papadopoulos, W. Wojtkowski, G. Wojtkowski, S. Wrycza, and J. Zupancic, 867–875. Boston, MA: Springer.
[Ben Henda 17]	Ben Henda, M. (2017). L'enseignement du code informatique à l'école : Prémices d'un humanisme numérique congénital . <i>Chaire Unesco-ITEN. Actes de la 5e rencontre annuelle d'ORBICOM, Les éditions de l'immatériel</i> .
[Benadi 04]	Benadi, S.A. (2004). Structuration des données et des services pour le télé-enseignement . <i>Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, France</i> .
[Ben-Ari 08]	Ben-Ari, M. (2008), March. The Effect of the Jeliot Animation System on Learning Elementary Programming . In <i>4th Panhellenic Conference on Didactics of Informatics</i> (pp. 28–30).
[Bentler et Bonett 80]	Bentler, P. M. & Bonett, D. G. (1980). Significance tests and goodness of fit in the analysis of covariance structures . <i>Psychological Bulletin</i> , 88(3), 588–606. https://doi.org/10.1037/0033-2909.88.3.588
[Boonsiritomachai et Pitchayadejanant 17]	Boonsiritomachai, W. Pitchayadejanant, K. (2017). Determinants Affecting Mobile Banking Adoption by Generation Y Based on the Unified Theory of Acceptance and Use of Technology Model Modified by the Technology

	Acceptance Model Concept. <i>Kasetsart Journal of Social Sciences</i> 40 (2): 349–358.
[Bordallo et Ginestet 93]	Bordallo, I. & Ginestet, J. P. (1993). <i>Pour une pédagogie du projet.</i> Paris : Hachette-Education.
[Boufaïda 95]	Boufaïda, M. (1995), <i>Mars. Méthodes et outils pour la génération d'interfaces, la conduite du dialogue pédagogique et la mise en oeuvre de diagnostics.</i> Thèse de doctorat.
[Bouton et al. 20]	Bouton, E. Tal, S. B. & Asterhan, C. S. (2020). Students, social network technology and learning in higher education: Visions of collaborative knowledge construction vs. the reality of knowledge sharing. <i>The Internet and Higher Education</i> , 100787.
[Bravo et al. 13]	Bravo, C. Duque, R. Gallardo, J. (2013). A Groupware System to Support Collaborative Programming: Design and Experiences. <i>Journal of Systems and Software</i> 86 (7): 1759–1771.
[Browne et Cudeck 93]	Browne, M. W. and Cudeck, R. 1993. Alternative ways of assessing model fit. <i>In Testing structural equation models, Edited by: Bollen, K. A. and Long, J. S. 136–62. Newbury Park, CA: Sage.</i>
[Bruner 98]	Bruner, J. L'approche psycho-culturelle de l'éducation. (1998). <i>4ème biennale de l'éducation et de la formation, La Sorbonne, Paris, France.</i>
[Bulu 12]	Bulu, S. T. (2012). Place presence, social presence, co-presence, and satisfaction in virtual worlds. <i>Computers & Education</i> , 58(1), 154-161.
[Capra et Arpin 02]	Capra, L. & Arpin, L. (2002). La médiation pédagogique de l'enseignant. <i>Québec français</i> , (126), 67-71.
[Chaguiboff 85]	Chaguiboff, J. (1985). Informatique et apprentissages. <i>Enfance</i> , 38(1), 31-42.
[Charlier et al. 98]	Charlier, B. Deschryver, N. Daele, A. (1998). Apprendre en collaborant à distance : ouvrons la boîte noire. <i>In Guir, R. (Ed.) TIC et formation des enseignants, Bruxelles, De Boeck.</i>
[Chiang 17]	Chiang, T.H. C. (2017). Analysis of Learning Behavior in a Flipped Programing Classroom Adopting Problem-Solving Strategies. <i>Interactive Learning Environments</i> , 25 (2): 189–202.
[Chin 98]	Chin, W.W. (1998). The Partial Least Squares Approach to Structural Equation Modeling. <i>Modern Methods for Business Research</i> , N°2, pp. 295-336.
[Chis et al. 18]	Chis, A. E. Moldovan, A.-N. Murphy, L. Pathak, P. and Muntean, C. H. (2018). Investigating Flipped Classroom and Problem-Based Learning in a Programming Module for Computing Conversion Course. <i>Educational Technology & Society</i> 21 (4): 232–247.
[Chorfi et al. 20]	Chorfi, A. Hedjazi, D. Aouag, S. & Boubiche, D. (2020). Problem-based collaborative learning groupware to improve computer programming skills. <i>Behaviour & Information Technology</i> , 1-20.

[Cockburn et Bryant 98]	Cockburn, A. Bryant, A.(1998), Juillet. Cloego: Collaborative and Multi-Metaphor Programming for Kids , Actes du 3 ^{ème} colloque Asian Pacific Computer and Human Interaction, Shonan Village Center, Japon, 15-17, IEEE Computer Society: Washington DC USA.
[Coutaz et Salber 95]	Coutaz, J. Salber, D. (1995). Etude de cas: Contrôle d'accès . Rapport GT SCOOP, GDR-PRC CHM, 11 pages.
[Cress et al. 15]	Cress, U. Stahl, G. Ludvigsen, S. Law, N. (2015). The core features of CSCL: Social situation, collaborative knowledge processes and their design . <i>International Journal of Computer-Supported Collaborative Learning</i> , 10(2), pp. 109-116.
[Curtis et al. 10]	Curtis, L. Edwards, C. Fraser, K. L. Gudelsky, S. Holmquist, J. Thornton, K. and Sweetser, K. D. (2010). Adoption of Social Media for Public Relations by Nonpro Fit Organizations . <i>Public Relations Review</i> 36 (1): 90–92.
[da Silva et Prikladnicki 15]	da Silva Estácio, B. J. & Prikladnicki, R. (2015). Distributed pair programming: A systematic literature review . <i>Information and Software Technology</i> , 63, 1-10.
[David et al. 03a]	David, B. Chalon, R. Vaisman, G. Delotte, O. (2003a). Capillary CSCW . In Stephanidis C., Jacko J., (Eds.) <i>Human-Computer Interaction Theory and Practice Vol. 2</i> , LEA, London, pp. 879-883.
[David et al. 03b]	David, B. Chalon, R. Delotte, O. Ros, J. Boutros, N. (2003b), Octobre. Travail coopératif capillaire en dépannage, maintenance et interventions de crise . In Actes du 5ème Congrès International de Génie Industriel, Québec, Canada.
[Davis 89]	Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology . <i>MIS Quarterly</i> , 13, 318-340.
[Davis et al. 89]	Davis, F. D. & Bagozzi, R. P. and Warshaw, P. R. (1989). User Acceptance of Computer Technology: A Comparison of Two Theoretical Models . <i>Management Science</i> (35:8), pp. 982-1002.
[De Hei et al. 15]	De Hei, M. S. A. Strijbos, J. W. Sjoer, E. and Admiraal, W. (2015). Collaborative Learning in Higher Education : Lecturers' Practices and Beliefs . <i>Research Papers in Education</i> , 30 (2): 232–247.
[Devide et al. 08]	Devide, J.A. Meneely, C.-w. Ho, Williams, L. and Devetisikiotis, M. (2008). Jazz Sangam: A Real-Time Tool for Distributed Pair Programming of a Team Development Platform . In <i>Infrastructure for Research on Collaborative Software Engineering (IReCoSE) Workshop at ACM SIGSOFT Foundations of Software Engineering (FSE)</i> , Atlanta. GA.
[Dewan 99]	Dewan, P. (1999). Architecture for Collaborative Applications . <i>Computer Supported Cooperative Work</i> . John Wiley & Sons Ltd,
[Dewey 68]	Dewey, J. (1899). The school and society . <i>The University of Chicago Press</i> , 1968.

[Dillenbourg 99]	Dillenbourg, P. (1999). What do you mean by collaborative learning?. <i>Collaborative-learning : Cognitive and Computational Approaches. Oxford: Elsevier. 1-19.</i>
[Dillenbourg et al. 09]	Dillenbourg, P. Jarvela, S. Fischer, F. (2009). The evolution of research on computer supported collaborative learning. <i>In N. Balacheff (Ed.), Technology-enhance learning, NewYork : Springer.</i>
[Dillon et Morris 99]	Dillon, A., & Morris, M. G. (1999). Power, Perception and Performance: From usability Engineering to technology acceptance with the P3 model of user response. <i>Paper presented at the 43rd Annual Conference of the Human Factors and Ergonomics Society, Santa Monica, CA: HFES.</i>
[Dolmans et Schmidt 96]	Dolmans, D. H. J. M. & Schmidt, H. (1996). The advantages of problem-based curricula. <i>Postgraduate Medical Journal, 72(851), 535-538.</i>
[Dourish et Bellotti 92]	Dourish, P. Bellotti, V. (1992). Awareness and Coordination in Shared Workspaces. <i>Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW-92), Toronto, Ontario, ACM Press, p. 107-114.</i>
[Duque et Bravo 08]	Duque, R. and Bravo, C. (2008). Analyzing Work Productivity and Program Quality in Collaborative Programming. <i>In the Third International Conference on Software Engineering Advances (ICSEA'2008) (pp. 270–276). IEEE Computer Society Press.</i>
[Duyck et al. 08]	Duyck, P. Pynoo, B. Devolder, P. Voet, T. Adang, L. Vercruyssen, J. (2008). User Acceptance of a Picture Archiving and Communication System. Applying the Unified Theory of Acceptance and Use of Technology in a Radiological Setting. <i>Methods of Information in Medicine 47 (2): 149–156.</i>
[El Hassan et Lazrek 07]	El Hassan, A. & Lazrek, A. (2007). Des ontologies pour la description des ressources pédagogiques et des profils des apprenants dans l'e-learning. <i>1ere Journées Francophones sur les ontologies.</i>
[Ellis 91]	Ellis, C.A. Gibbs, S.G. Rein, G.L. (1991). Groupware, Some Issues and Experiences. <i>Communication ACM, vol 34, n° 1, (January 1991), p. 38-58.</i>
[Engeström 01]	Engeström, Y. (2001). Expansive learning at work : Toward an activity theoretical reconceptualization. <i>Journal of Education, 14(1).</i>
[Engeström 97]	Engeström, Y. (1997). Activity theory and individual and social transformation. <i>In Y. Engeström, R. Miettinen & P. R.-L. (Eds.), Perspectives on activity theory (pp. 19-38). Cambridge England: Cambridge University Press.</i>
[Erkens et Bodemer 19]	Erkens, M. and Bodemer, D. (2019). Improving Collaborative Learning: Guiding Knowledge Exchange Through the Provision of Information About Learning Partners and Learning Contents. <i>Computers & Education 128: 452–472.</i>
[Faerber 02]	Faerber, R. (2002). Le groupe d'apprentissage en formation à distance : ses caractéristiques dans un environnement virtuel. <i>In Larose F & Karsenti T. (Ed.), La place des TICE en formation initiale et continue à l'enseignement :</i>

	<i>bilan et perspectives. Sherbrooke. Editions du CRP, Université de Sherbrooke, 99-128.</i>
[Fallery et al. 14]	Fallery, B. Ologeanu-Taddei, R. Oiry, E. et Tchobanian, R. (2014). Usages des outils collaboratifs : le rôle des formes organisationnelles et des politiques de ressources humaines. <i>Management & Avenir, vol.1 n° 67, pp.1-15.</i>
[Ferradji 17]	FERRADJI, M. A. (2017). Analyse, conception et implémentation d'un environnement pour le support de l'apprentissage collaboratif du diagnostic médical. <i>Thèse de Doctorat, Université de Batna 2.</i>
[Ferradji et Hedjazi 17]	Ferradji, M. A. and Hedjazi, D. (2017), October. Towards Virtual Collaborative Learning of Clinical Skills. <i>International Conference on Internet of Things, Embedded Systems and Communications (IINTEC) (pp. 26–31). IEEE.</i>
[Fishbein et Ajzen 75]	Fishbein, M. & Ajzen, I. (1975). Belief, attitude, intention, and behavior: An introduction to theory and research. <i>Reading, MA: Addison-Wesley.</i>
[Fornell et Lacker 81]	Fornell, C. and Lacker, D. F. (1981). Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. <i>Journal of Marketing Research 18 (1): 39–50.</i>
[Gašević et al. 18]	Gašević, D., S. Joksimović, B. R. Eagan, and D. W. Shaffer. (2018). SENS: Network Analytics to Combine Social and Cognitive Perspectives of Collaborative Learning. <i>Computers in Human Behavior, 92: 562–577.</i>
[George 01]	George, S. (2001). Apprentissage collectif à distance. SPLACH : un environnement informatique support d'une pédagogie par projet. <i>Thèse de doctorat, Université du Maine, France.</i>
[Gergen 99]	Gergen, K. J. (1999). An Invitation to Social Construction. <i>Thousand Oaks, CA: Sage Publications.</i>
[Ghalandari 12]	Ghalandari, K. (2012). The Effect of Performance Expectancy, Effort Expectancy, Social Influence and Facilitating Conditions on Acceptance of e-Banking Services in Iran: The Moderating Role of age and Gender. <i>Middle-East Journal of Scientific Research 12 (6): 801–807.</i>
[Gholami et al. 16]	Gholami, M. Moghadam, P. K. Mohammadipoor, F. Tarahi, M. J. Sak, M. Toulabi, T. & Pour, A. H. H. (2016). Comparing the effects of problem-based learning and the traditional lecture method on critical thinking skills and metacognitive awareness in nursing students in a critical care nursing course. <i>Nurse Education Today, 45, 16-21.</i>
[Goldman et al. 11]	Goldman, M. Little, G. and Miller, R. C. (2011). Collabode: Collaborative Coding in the Browser. <i>In Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering (pp. 65–68). ACM.</i>
[Grappin 04]	Grappin, A.M. (2004). Itinéraires de découverte le goût d'apprendre. <i>Grenoble: CRDP de l'académie de Grenoble.</i>

[Greenberg et al. 96]	Greenberg, S. Gutwin, C. and Cockburn, A. (1996). Awareness through Fisheye Views in Relaxed-WYSIWIS groupware. In <i>Graphics Interface (Vol. 96)</i> , 28–38. Toronto, Canada: Morgan Kaufmann.
[Gutwin et al. 96]	Gutwin, C. Greenberg, S. and Roseman, M. (1996). Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets and Evaluation. In <i>HCI'96. Proceedings of the Conference on People and Computers</i> . New York: Springer-Verlag, pp. 281–298.
[Hair et al. 2006]	Hair, J. F. Black, W. C. Babin, B. J. Anderson, R. E. and Tatham, R. L. (2006). Multivariate Data Analysis. 6th ed. Upper Saddle River, NJ: Prentice-Hall.
[Hedjazi 11]	Hedjazi, D. (2011). Conception d'un modèle coopératif de support de la télémaintenance industrielle. (Doctoral dissertation, Université de Batna 2).
[Hedjazi 18]	Hedjazi, D. (2018). Constructing collective competence : a new CSCW-based approach. <i>International Journal of Information and Communication Technology</i> , 12(3-4), 393-416. https://doi.org/10.1504/IJICT.2018.090418
[Hedjazi et Zidani 11]	Hedjazi, D. & Zidani, A. (2011). Development of an industrial e-maintenance system integrating groupware techniques. <i>International Journal of Industrial and Systems Engineering</i> , 9(2), 227-247.
[Hedjazi et Zidani 14]	Hedjazi, D. & Zidani, A. (2014). Group awareness evaluation in e-maintenance Groupware system. <i>The 1st International conference on artificial intelligence and information technology, Ouargla, Algeria.</i>
[Henri et al. 01]	Henri, F. & Lundgren-Cayrol, K. (2001). Apprentissage collaboratif à distance. Sainte-Foy : Presses de l'Université du Québec.
[Henry et Vandeput 18]	Henry, J. & Vandeput, E. (2018). Apprendre à programmer : comment les enseignants justifient-ils le choix d'un outil didactique?. In <i>De 0 à 1 ou l'heure de l'informatique à l'école : Actes du colloque Didapro 7–DidaSTIC</i> (pp. 325-342). Peter lang.
[Hernández-Sellés et al. 19]	Hernández-Sellés, N. Muñoz-Carril, P. C. & González-Sanmamed, M. (2019). Computer-supported collaborative learning: An analysis of the relationship between interaction, emotional support and online collaborative tools. <i>Computers & Education</i> , 138, 1-12.
[Herrera-Pavo 21]	Herrera-Pavo, M. Á. (2021). Collaborative learning for virtual higher education. <i>Learning, Culture and Social Interaction</i> , 28, 100437. https://doi.org/10.1016/j.lcsi.2020.100437
[Homans 50]	HOMANS, G. C. (1950). The human group. New York, Harcourt Brace.
[Homans 61]	HOMANS, G. C. (1961). Social behavior : its elementary forms. New York, Harcourt Brace.
[Huang 15]	Huang, Y. M. (2015). Exploring the Factors That Affect the Intention to Use Collaborative Technologies: The Differing Perspectives of Sequential/Global Learners. <i>Australasian Journal of EducationalTechnology</i> 31: 3.

[Huber 05]	Huber, M. (2005). Conduire un projet-élèves . Hachette Éducation.
[Hurst et al. 13]	Hurst, B. Wallace, R. and Nixon, S. B. (2013). The Impact of Social Interaction on Student Learning . <i>Reading Horizons</i> , 52 (4): 5.
[Hwang et al. 12]	Hwang, W. Y. Shadiev, R. Wang, C. Y. and Huang, Z. H. (2012). <i>A Pilot Study of Cooperative Programming Learning Behavior and Its Relationship with Students' Learning Performance</i> . <i>Computers & Education</i> 58 (4): 1267–1281.
[Isohätälä et al. 17]	Isohätälä, J. Järvenoja, H. and Järvelä, S. (2017). Socially Shared Regulation of Learning and Participation in Social Interaction in Collaborative Learning . <i>International Journal of Educational Research</i> , 81: 11–24.
[Jeong et al. 19]	Jeong, H. Hmelo-Silver, C. E. & Jo, K. (2019). Ten years of computer-supported collaborative learning: A meta-analysis of CSCL in STEM education during 2005–2014 . <i>Educational Research Review</i> , 28, 100284.
[Johnson 98]	Johnson, P. M. (1998). Reengineering Inspection: The Future of Formal Technical Review . <i>Communications of the ACM</i> 41:49–52.
[Jung et al. 02]	Jung, I. Choi, S. Lim, C. and Leem, J. (2002). Effects of Different Types of Interaction on Learning Achievement, Satisfaction and Participation in Web-Based Instruction . <i>Innovations in Education and Teaching International</i> , 39(2): 153–162.
[Kay et al. 20]	Kay, J. Barg, M. Fekete, A. Greening, T. Hollands, O. Kingston, J. H. and Crawford, K. (2000). Problem-Based Learning for Foundation Computer Science Courses . <i>Computer Science Education</i> 10 (2): 109–128.
[Kelleher et al. 02]	Kelleher, C. Cosgrove, D. Culyba, D. Forlines, C. Pratt, J. Pausch, R. (2002), <i>Octobre</i> . Alice2: Programming without Syntax Errors . <i>Actes du 15^{ème} colloque annuel User Interface Software & Technology, Paris, France, 27-30</i> .
[Klopfer et Yoon 05]	Klopfer, E. Yoon, S. (2005). Developing Games and Simulations for Today and Tomorrow's Tech Savvy Youth . in <i>TechTrends: Linking Research & Practice to Improve Learning</i> , n°49, vol. III, pages 33-41.
[Koivuniemi et al. 18]	Koivuniemi, M. Järvenoja, H. and Järvelä, S. (2018). Teacher Education Students' Strategic Activities in Challenging Collaborative Learning Situations . <i>Learning, Culture and Social Interaction</i> , 19: 109–123.
[Kwon et al. 14]	Kwon, K. & Liu, Y. H. & Johnson, L. P. (2014). Group regulation and social-emotional interactions observed in computer supported collaborative learning: Comparison between good vs. poor collaborators . <i>Computers & Education</i> , 78, 185-200.
[Lafifi 07]	Lafifi, Y. (2007). SACA : un système d'apprentissage collaboratif . <i>Doctoral dissertation, PhD Thesis, Computer science department, University of Annaba, Algeria</i> .
[Larmer 14]	Larmer, J. (2014), <i>January</i> . Project-Based Learning vs. Problem-Based Learning vs. X-BL .

[Lave et Wenger 91]	Lave, J. et Wenger, E. (1991). Situated learning: legitimate peripheral participation. <i>Cambridge : Cambridge University Pr</i>
[Lebrun 07]	Lebrun, M. (2007). Théories et méthodes pédagogiques pour enseigner et apprendre : Quelle place pour les TIC dans l'éducation ? (2ème édition.) . <i>Bruxelles : De Boeck Supérieur.</i>
[Leclercq et Denis 01]	Leclercq, D. & Denis, B. (2001). Auto-observation des modalités d'apprentissage en situation de projet. <i>Métacognition mathématique au cours de PARMs. Revue des sciences de l'éducation, 27 (2), 421-440.</i> https://doi.org/10.7202/009940ar
[Lee et al. 11]	Lee, S. J. Srinivasan, S. Trail, T. Lewis, D. and Lopez, S. (2011). Examining the Relationship among Student Perception of Support, Course Satisfaction, and Learning Outcomes in Online Learning. <i>The Internet and Higher Education 14 (3): 158-163.</i>
[Legendre 93]	Legendre, R. (1993). Distionnaire actuel de l'éducation. Montréal : Guérin.
[Legros et al. 02]	Legros, D. Maître de Pembroke, E. & Talbi, A. (2002). Les théories de l'apprentissage et les systèmes multimédias. <i>In Legros, D. & Crinon, J. (dir.) Psychologie des apprentissages et multimédia. Paris : Armand Colin. pp. 22-39.</i>
[Liccardi et al. 07]	Liccardi, I. Davis H. C. and White. (2007). CAWS: a wiki system to improve workspace awareness to advance effectiveness of co-authoring activities. <i>CHI EA '07: CHI '07 Extended Abstracts on Human Factors in Computing Systems, San Jose CA USA , April 28- May 3.</i> https://doi.org/10.1145/1240866.1241040.
[Linard 01]	Linard, M. (2001). Concevoir des environnements pour apprendre : l'activité humaine, cadre organisateur de l'interactivité technique. <i>Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation Année 2001 8-3-4 pp. 211-238</i>
[Liu et al. 18]	Liu, M. Liu, L. and Liu, L. (2018). Group Awareness Increases Student Engagement in Online Collaborative Writing. <i>The Internet and Higher Education 38: 1-8.</i>
[Loehlin 04]	Loehlin, J.C. (2004). Latent variable models : an introduction to factor, path, and strutural equation analysis. <i>ISBN 0-8058-4909-2.</i>
[Lu et al. 08]	Lu, J. Liu, C. Yu, C. S. and Wang, K. (2008). Determinants of Accepting Wireless Mobile Data Services in China. <i>Information & Management 45 (1): 52-64. doi:10.1016/j.im.2007.11.002.</i>
[Maloney et al. 04]	Maloney, J. Burd, L. Kafai, Y. Rusk, N. Silverman, B. Resnick, M. (2004), Janvier. Scratch: A Sneak Preview. Actes du 2ème colloque international Creating Connecting, and Collaborating through Computing. <i>Keihanna-Plaza, Kyoto, Japon, 29-30. IEEE Computer Society: Washington DC USA.</i>

[Marchewka et al. 07]	Marchewka, J. T. Liu, C. and Kostiwa, K. (2007). An Application of the UTAUT Model for Understanding Student Perceptions Using Course Management Software. <i>Communications of the IIMA</i> 7 (2): 93–104.
[Marot et Darnige 96]	Marot, J-C. Darnige, A. (1996). <i>La téléformation, Que sais-je ?</i> . (Presses Universitaires de France, Paris,).
[Maureen et Mildred 81]	Pope Maureen, L. Shaw Mildred, L.G. (1981). Personal construct psychology in education and learning. <i>Int. J. Man-Machine Studies</i> , 14, (223-232).
[Mayer 81]	Mayer, Richard. E. (1981). The psychology of how novices learn computer programming. <i>ACM Computing Surveys</i> 13 (1) (121-141).
[Michard 82]	Michard, A. (1982). Didactique de l'informatique. <i>In : Giordan et Martinand.</i>
[Molinillo et al. 18]	Molinillo, S. Aguilar-Illescas, R. Anaya-Sánchez, R. and Vallespín-Arán, M. (2018). Exploring the Impacts of Interactions, Social Presence and Emotional Engagement on Active Collaborative Learning in a Social web-Based Environment. <i>Computers & Education</i> , 123: 41–52.
[Moore 93]	Moore, M. G. (1993). Three Types of Interaction. <i>In Distance Education: New Perspectives</i> , edited by K. Harry, M. John, and D. Keegan, 19–24. New York: Routledge.
[Moreno et al. 04]	Moreno, A. Myller, N. and Sutinen, E. (2004). JeCo, a Collaborative Learning Tool for Programming. <i>In Proceedings of the 2004 IEEE Symposium on Visual Languages-Human Centric Computing (VLHCC'04), Washington, DC, USA</i> , 261–263.
[Morval 67]	MORVAL, Jean. (1967), <i>juin-juillet</i> . Apport de la dynamique de groupe à l'approfondissement des relations humaines. <i>Bulletin social des industriels</i> , 338, pp. 508-513.
[Morval 71]	MORVAL, J. (1971). Approches du comportement interpersonnel dans les groupes restreints. <i>thèse de doctorat, Louvain, Université catholique.</i>
[Nuutila et al. 08]	Nuutila, E. Törmä, S. Kinnunen, P. and Malmi, L. (2008). Learning Programming with the PBL Method—Experiences on PBL Cases and Tutoring. <i>In Reflections on the Teaching of Programming</i> , 47–67. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-77934-6_5 .
[Papert 81]	Papert, S. (1981). Jaillissement de l'esprit – Ordinateurs et apprentissage. (traduit par R.-M. Vassallo-Villaneau). Paris : Flammarion.
[Paquette 76]	Paquette, Claude. (1976). Vers une pratique de la pédagogie ouverte. <i>Victoriaville, Québec, Les Éditions NHP.</i>
[Paquette 79]	Paquette, Claude, (1979), décembre. Quelques fondements d'une pédagogie ouverte. <i>Québec français</i> , n° 36, pp. 18-19.
[Paré 77]	PARÉ, André. (1977). Créativité et pédagogie ouverte. <i>Volume I, Victoriaville, Québec, Les Éditions NHP.</i>

[Park et al. 17]	Park, J. Park, Y. H. Kim, S. and Oh, A. (2017), February. Eliph: Effective Visualization of Code History for Peer Assessment in Programming Education. <i>In Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing (pp. 458–467).</i> ACM.
[Pedersen et al. 09]	Pedersen, R. U. Nørbjerg, J. and Scholz, M. P. (2009). Embedded Programming Education with Lego Mindstorms NXT using Java (leJOS), Eclipse (XPairtise), and Python (PyMite). <i>Proceedings of the WESS'09, Oct 11–16, p. 50–55, Grenoble, France.</i>
[Peng 10]	Peng, W. (2010). Practice and experience in the application of problem-based learning in computer programming course. <i>In 2010 International Conference on Educational and Information Technology (Vol. 1, pp. VI-170).</i> IEEE, September.
[Pérez 06]	Pérez, J. R. Paule, M. P. and Cueva, J. M. (2006). SICODE: A collaborative tool for learning of software development. <i>In IV International Conference on Multimedia and Information & Communication Technologies in, Education (m- ICTE2006).</i>
[Perriault 96]	Perriault, J. (1996). <i>La communication du savoir à distance. Éducation et formation (L'Harmattan, Paris).</i>
[Phichitchaisopa & Naenna 13]	Phichitchaisopa, N. Naenna, T. (2013). Factors affecting the adoption of healthcare information technology. <i>EXCLI Journal, 12, 413-436.</i> DOI: 10.17877/DE290R-5602
[Piaget 35]	Jean Piaget, J. (1935). Psychologie et pédagogie. (<i>Folio essais, Paris, ed. 1991, 1935</i>).
[Posner et Baeker 92]	Posner, I. and Baeker, R. (1992). How People Write Together ?. <i>Proceedings of the 25th Annual Hawaiian International Conference on System Sciences: 127–138.</i>
[Proulx 04]	Proulx, J. (2004). L'Apprentissage Par Projet. PUQ.
[Raby et al. 07]	Raby, C. Beaudry, N. & Viola, S. (2007). Modèles d'enseignement et théories d'apprentissage : de la pratique à la théorie. <i>Anjou : Éditions CEC.</i>
[Raclet et al. 20]	Raclet, J. B. Silvestre, F. & Pons, M. (2020), February. Mise en oeuvre d'approches pédagogiques fondées sur des pratiques de l'industrie du logiciel pour l'apprentissage de la programmation. <i>In 2020 Colloque Didapro 8-DidaSTIC: L'informatique, objets d'enseignements (pp. 1-12).</i>
[Reis et al. 18]	Reis, R. C. D. Isotani, S. Rodriguez, C. L. Lyra, K. T. Jaques, P. A. and Bittencourt, I. I. (2018). Affective States in Computer-Supported Collaborative Learning: Studying the Past to Drive the Future. <i>Computers & Education 120: 29–50.</i>
[Rogalski 88]	Rogalski, Janine. (1988). Enseignement de méthodes de programmation dans l'initiation à l'informatique. <i>Colloque francophone sur la didactique de l'informatique, Sep 1988, Paris, France. pp.61-72. edutice-00359382.</i>

[Rovers et al. 18]	Rovers, S. F. Clarebout, G. Savelberg, H. H. and Merrienboer J. J. van. (2018). Improving Student Expectations of Learning in a Problem-Based Environment. <i>Computers in Human Behavior</i> 87: 416–423. doi:10.1016/j.chb.2018.02.016.
[Salinger et al. 10]	Salinger, S. Oezbek, C. Beecher, K. and J. Schenk. (2010). Saros: An Eclipse Plug-In for Distributed Party Programming. <i>Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering</i> , p.48-55, May 02-02, 2010, Cape Town, South Africa. collabedit. Online: http://collabedit.com .
[Savard 78]	Savard, J.G. (1978). Statistiques. Montréal, éd. HRW, traduit et adapté de Gilbert (N.) 384 p.
[Savery 06]	Savery, J. R. (2006). Overview of problem based learning: Definition and distinctions. <i>Interdisciplinary Journal of Problem-based Learning</i> , 1(1), 3.
[Scager et al. 16]	Scager, K. Boonstra, J. Peeters, T. Vulperhorst, J. and Wiegant, F. (2016). Collaborative Learning in Higher Education : Evoking Positive Interdependence. <i>CBE Life Sciences Education</i> , 15 (4): 1–9.
[Scherer et al. 20]	Scherer, R. Siddiq, F. & Viveros, B. S. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. <i>Computers in Human Behavior</i> , 106349.
[Schlemminger 03]	Schlemminger, G. (2003). La pédagogie Freinet et le travail en projet : Plus de manuels scolaires. <i>Les langues modernes</i> , n° 2. pp. 47-57.
[Serrano et al. 14]	Serrano-Cámara, L. M. & Paredes-Velasco, M. Alcover, C. M. and Velazquez-Iturbide, J. Á. (2014). An Evaluation of Students' Motivation in Computer-Supported Collaborative Learning of Programming Concepts. <i>Computers in Human Behavior</i> 31: 499–508.
[Seyal et al. 02]	Seyal, A. Rahman, M.N. Mahbubur,R. (2002). Determinants of academic use of the Internet: A structural equation model. <i>Behaviour and Information Technology</i> 21(1):71-86. DOI: 10.1080/01449290210123354
[Shute et al. 17]	Shute, V. J. Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. <i>Educational Research Review</i> , 22, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003 .
[Skinner 54]	Skinner,B.F.(1954). The Science of Learning and the Art of Teaching. <i>Harvard Education Revue</i> , vol. 24, p. 86-97, (1954).
[Sohlenkamp 98]	Sohlenkamp, M. (1998). Supporting group awareness in multi-user environments through percetualization. <i>Dissertation Fachbereich Mathematik-Informatik der Universität- Gesamthochschule-Paderborn.</i>
[Spears 12]	Spears, L. R. (2012). Social Presence, Social Interaction, Collaborative Learning, and Satisfaction in Online and Face-to-Face Courses. Graduate Theses and Dissertations. 12976. https://lib.dr.iastate.edu/etd/12976

[Stahl et al. 06]	Stahl, G. & Koschmann, T. Suthers, D. (2006). Computer-Supported Collaborative Learning: An Historical Perspective. In Cambridge Handbook of the Learning Sciences. edited by R. K. Sawyer, 409–426. Cambridge, UK: Cambridge University Press.
[St-Arnaud 79]	St-Arnaud, Yves. (1979). La psychologie (modèle systémique). Montréal, Les Presses de l'Université de Montréal et les Editions du CIM.
[Talhi 07]	Talhi, S. (2007). Intégration des technologies de coopération et d'intelligence dans les environnements d'apprentissage à distance. <i>Doctoral dissertation, Université de Batna 2.</i>
[Tardif 98]	Tardif, J. (1998). Intégrer les nouvelles technologies de l'information / Quel cadre pédagogique ?. Paris. ESF. pp.54-56.
[Teo 09]	Teo, T. (2009). Modelling Technology Acceptance in Education: A Study of Pre-Service Teachers. <i>Computers & Education 52 (1): 302–312.</i> doi:10.1016/j.compedu.2008.08.006.
[Terzis et Economides 11]	Terzis, V. and Economides, A. A. (2011). Computer Based Assessment: Gender Differences in Perceptions and Acceptance. <i>Computers in Human Behavior 27 (6): 2108–2122.</i> doi:10.1016/j.chb.2011.06.005.
[Thien 14]	Thien, N. V. (2014). Pédagogie par projet : une étude de cas dans l'enseignement supérieur technologique. <i>Thèse de doctorat en Science de l'Education, Ecole doctorale EPIC, UNIVERSITE LUMIERE LYON 2, France.</i>
[Tosuntaş et al. 15]	Tosuntaş, ŞB. Karadağ, E. and Orhan, S. (2015). The Factors Affecting Acceptance and use of Interactive Whiteboard Within the Scope of FATIİH Project: A Structural Equation Model Based on the Unified Theory of Acceptance and Use of Technology. <i>Computers & Education 81: 169–178.</i>
[Triandis 80]	Triandis, H. C. (1980). Values, attitudes and personal behaviour. Beliefs, attitudes and values. <i>University of Nebraska Press.</i>
[Venkatesh et al. 03]	Venkatesh, V. Morris, M. G. Davis, G. B. and Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. <i>MIS Quarterly 27 (3): 425–478.</i>
[Venkatesh et al. 11]	Venkatesh, V. & Thong, J. Y. L. & Chan, F. K. Y. & Hu, P. J.-H. and Brown, S. A. (2011). Extending the Two-Stage Information Systems Continuance Model: Incorporating UTAUT Predictors and the Role of Context. <i>Information Systems Journal 21 (6): 527–555.</i>
[Verbruggen et al. 20]	Verbruggen, S. Depaepe, F. & Torbeyns, J. (2020). Effectiveness of educational technology in early mathematics education: A systematic literature review. <i>International Journal of Child-Computer Interaction, 100220.</i>
[Vygotski 34]	Vygotski, L.S. (1934). Pensée et langage. (MIT press, ed. 1988, 1934).

[Vygotsky 78]	Vygotsky, L. (1978). Mind in Society: The Development of Higher Psychological Processes. <i>Cambridge: Harvard University Press.</i>
[Weisband 02]	Weisband, S. (2002). Maintaining Awareness in Distributed Team Collaboration: Implication for Leadership and performance. <i>In P. Hinds & S. Kiesler, Distributed Work, pp. 311-333. Cambridge, MA: MIT Press.</i>
[Westland et Clark 00]	Westland, J. C. and Clark, T. H. K. (2000). Global Electronic Commerce : Theory and Case Studies. <i>MIT Press, Cambridge, MA.</i>
[Wing 06]	Wing, J. M. (2006). Computational thinking. <i>Communications of the ACM, 49, 33–35.</i>
[Woo et Reeves 07]	Woo, Y. and Reeves, T.C. (2007). Meaningful Interaction in Web-Based Learning: A Social Constructivist Interpretation. <i>The Internet and Higher Education 10 (1): 15–25.</i>
[Yadav et al. 17]	Yadav, A. Good, J. Voogt, J. & Fisser, P. (2017). Computational thinking as an emerging competence domain. <i>In M. Mulder (Ed.), Competence-based vocational and professional education: Bridging the worlds of work and education (pp. 1051–1067). Cham: Springer International Publishing.</i>
[Zhou et al. 20]	Zhou, N. Kisselburgh, L. Chandrasegaran, S. Badam, S. K. Elmqvist, N. & Ramani, K. (2020). Using social interaction trace data and context to predict collaboration quality and creative fluency in collaborative design learning environments. <i>International Journal of Human-Computer Studies, 136, 102378.</i>

Annexe 1. Notre Production Scientifique

Publications dans des Revues Internationales classe-A :

Chorfi, A., Hedjazi, D., Aouag, S., & Boubiche, D. (2020). Problem-based collaborative learning groupware to improve computer programming skills. *Behaviour & Information Technology*, 1-20.

Publication dans les Actes de Conférences Internationales :

Chorfi, A., Zidani, A., & Lezzar, F. (2012, July). Building a shared editing workspace for learners' cooperative programming. In *International Conference on Education and e-Learning Innovations* (pp. 1-9). IEEE.

Chorfi, A., Zidani, A., & Lezzar, F. (2012, November). Project-Based learning with a shared editor. In *2012 Second International Workshop on Advanced Information Systems for Enterprises* (pp. 48-56). IEEE.

Chorfi, A., Zidani, A., & Lezzar, F. (2012, May). A shared workspace supporting students programming projects. In *2nd International Symposium on Modelling and Implementation of Complex Systems*, Constantine, Algeria.