Adaptive Personalization of Pedagogical Sequences using Machine Learning

Benjamin Clément

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

**DOCTEUR DE
L'UNIVERSITÉ DE BORDEAUX**

ECOLE DOCTORALE DE
MATHÉMATIQUE ET INFORMATIQUE

SPÉCIALITÉ : INFORMATIQUE

Par Benjamin CLEMENT

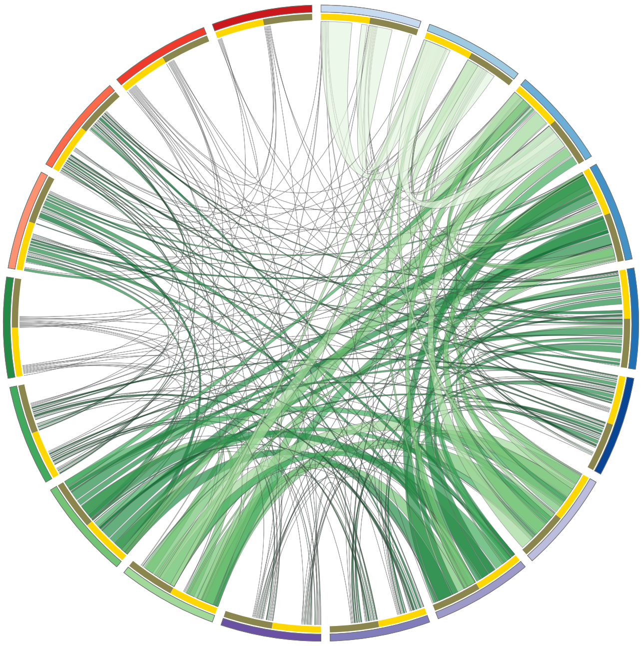
Adaptive Personalization of Pedagogical Sequences using Machine Learning

sous la direction de : Pierre-Yves OUDEYER

Date de soutenance : 12 décembre 2018

Membres du jury :

Vanda Luengo, Présidente du jury
Stéphane Magnenat, Rapporteur
Emma Brunskill, Examinatrice
Hélène Sauzeon, Examinatrice
Fabien Lotte, Examinateur



Thesis presented in order to obtain the title of
Docteur de l'Université de Bordeaux
Mathématiques et Informatique
Spécialité Informatique

by Benjamin D. N. Clément

Adaptive Personalization of Pedagogical Sequences using Machine Learning

under the direction of Pierre-Yves Oudeyer, Research Director,
Flowers Team – Inria Bordeaux Sud-Ouest, ENSTA Paristech

Vanda Luengo	Professor, *MoCAH team, LIP6 Paris*	Reviewer
Stéphane Magnenat	Deputy Scientific Director, *Game Technology Center, ETH Zurich*	Reviewer
Emma Brunskill	Professor, *AI for Human Impact Lab, Stanford University*	Examiner
Hélène Sauzeon	Professor, *University of Bordeaux; Flowers Team, Inria*	Examiner
Fabien Lotte	Research Scientist, *Potioc Team, Inria*	Examiner

Defended the 12th of December, 2018

Remerciements

Cette thèse, et les années durant lesquelles j'ai travaillé dans l'équipe Flowers, font partie des périodes les plus enrichissantes et tumultueuses de ma vie à la fois sur le plan personnel et professionnel, et je ne serais pas la personne que je suis aujourd'hui sans les gens qui m'ont entourés pendant ces années.

Je tiens tout d'abord à remercier Pierre-Yves Oudeyer pour la bienveillance, le soutien et la confiance qu'il m'a donné durant ces années, ce fut un réel plaisir de travailler à ses côtés et un enrichissement d'échanger avec lui. Je remercie également Manuel Lopes, pour son encadrement durant les première années de ma thèse, qui m'a permis d'apprendre énormément de choses autant sur mon travail que sur moi-même, et sans qui je n'aurais certainement pas l'expérience que j'ai aujourd'hui. J'aimerais également remercier Hélène Sauzéon pour son aide et les échanges que nous avons eu, pour l'expérience qu'elle m'a apportée sur les questions de psychologie et la disponibilités qu'elle m'a témoignée.

Il est évident que mes collègues, avec qui j'ai passé d'excellents moments, ont également jouer un rôle très important et ont fait de ces années de thèse, de très bon souvenirs. Tout d'abord mes remerciement vont à ceux avec qui j'ai partagé un bureau, Didier Roy, pour son amitié, son soutien tout au long de ma thèse, nos discussions animés qui n'ont pas toujours fait l'unanimité du voisinage et pour sa collaborations durant sur ce travail de thèse; Baptiste Busch et Thibault Munzer pour leur présence et leur aide dans les moments de doute et leur propension à ne jamais dire non à un café. Je remercie aussi, William Schueller pour ses traits d'esprit, son humour infaillible et les nuits de rédactions passé au bureau et Fabien Benureau qui a été une réelle inspiration dans mon travaille de chercheur par sa fougue sans limite, son approche de la science sans concession et le soutien qu'il m'a toujours apporté. Je remercie également Josias Levi Alvares pour son aide durant la dernière expérimentation et pour m'avoir permis d'encadrer un étudiant intéressé, motivé et entreprenant.

Merci également à tout les membres de l'équipe Flowers et d'Inria que je n'ai pas cité et avec qui j'ai passé de très bon moments, que ce soit dans les bureaux, à l'escalade, au baby-foot ou autour d'une bière.

Je tiens également à remercier tout particulièrement les enseignants qui ont accepté de travailler avec nous durant nos travaux et qui m'ont permis de travailler avec leurs élèves.

Enfin je remercie mes parents, ma mère et mon père, pour leur amour, le soutien qu'ils m'ont toujours apporté et qui m'ont permis d'arriver là où j'en suis aujourd'hui.

Et, puisqu'il est est de coutume de garder le meilleur pour la fin, je remercie très profondément mon grand frère sans qui je ne serais pas l'homme que je suis aujourd'hui. Je te remercie pour ton amour, pour toute ces années où nous avons grandi ensemble, pour tout ce que tu m'as appris, pour ta patience et ton ouverture d'esprit et, pour le temps que tu as consacré et la minutie dont tu as fait preuve dans ta relecture de ce manuscrit, tu as mon infinie gratitude. Merci, frangin.

Acknowledgments

This PhD, and the years I worked in the Flowers team are parts of the most enriching and tumultuous periods of my life, both personally and professionally, and I would not be the person I am today without the people who surrounded me during those years.

First of all, I would like to thank Pierre-Yves Oudeyer for the benevolence, the support and the trust he showed me over the years, it has been a real pleasure to work and exchange with him. I would also like to thank Manuel Lopes for his supervision during the first years of my PhD, which has allowed me to learn a lot about my work and myself, and without whom I would certainly not have the experience I have today. I would also like to thank Hélène Sauzéon to have always been available for my questions, for the help and the feedbacks she gave to me and for the exchanges we had and the experience she brought me on psychological issues.

It is obvious that my colleagues, with whom I spent really good times, also played a very important role and made these years of PhD, a lot of very good memories. First of all, my thanks go to those with whom I shared an office, Didier Roy, for his friendship, his support throughout my PhD, our lively discussions which were not always welcomed by the office neighbourhood and for his collaboration during this PhD work; Baptiste Busch and Thibault Munzer for their support and help in times of doubt and their propensity to never say no to a coffee. I would also like to thank William Schueller for his witty traits, his infallible humour and the nights of writing spent at the office, and Fabien Benureau who has been a real inspiration in my work as a researcher through his boundless enthusiasm, his uncompromising approach to science and the support he always provided me. I also thank Josias Levi Alvares for his help during the last experiment and for allowing me to supervise an interested, motivated and engaged student.

Thanks also, to all the members of the Flowers team and Inria that I didn't mention and with whom I had great times, whether in the offices, at climbing, playing foosball or while sipping a beer.

I would also like to particularly thank the teachers who agreed to work with us during the PhD and who allowed me to work with their students.

Finally, I thank my parents, my mother and my father, for their love, for the support they have always given me and which has allowed me to get to where I am today.

And, since it is the custom to keep the best for the end, I thank very deeply my big brother without whom I would not be the man I am today. I thank you for your love, for all the years we have grown up together, for all the things you have taught me, for your patience and open-mindedness, and, for the time you have devoted and the meticulous review you made about this manuscript, you have my eternal gratitude.
Thanks, bro'.

Résumé

Les ordinateurs peuvent-ils enseigner ? Pour répondre à cette question, la recherche dans les Systèmes Tuteurs Intelligents est en pleine expansion parmi la communauté travaillant sur les Technologies de l'Information et de la Communication pour l'Enseignement (TICE). C'est un domaine qui rassemble différentes problématiques et réunit des chercheurs venant de domaines variés, tels que la psychologie, la didactique, les neurosciences et, plus particulièrement, le machine learning.

Les technologies numériques deviennent de plus en plus présentes dans la vie quotidienne avec le développement des tablettes et des smartphones. Il semble naturel d'utiliser ces technologies dans un but éducatif. Cela amène de nombreuses problématiques, telles que comment faire des interfaces accessibles à tous, comment rendre des contenus pédagogiques motivants ou encore comment personnaliser les activités afin d'adapter le contenu à chacun.

Au cours de cette thèse, nous avons développé des méthodes, regroupées dans un framework nommé HMABITS, afin d'adapter des séquences d'activités pédagogiques en fonction des performances et des préférences des apprenants, dans le but de maximiser leur vitesse d'apprentissage et leur motivation. Ces méthodes utilisent des modèles computationnels de motivation intrinsèque pour identifier les activités offrant les plus grands progrès d'apprentissage, et utilisent des algorithmes de Bandits Multi-Bras pour gérer le compromis exploration/exploitation à l'intérieur d'un espace d'activité. Les activités présentant un intérêt optimal sont ainsi privilégiées afin de maintenir l'apprenant dans un état de Flow ou dans sa Zone de Développement Proximal. De plus, certaines de nos méthodes permettent à l'apprenant de faire des choix sur des caractéristiques contextuelles ou le contenu pédagogique de l'application, ce qui est un vecteur d'autodétermination et de motivation. Afin d'évaluer l'efficacité et la pertinence de nos algorithmes, nous avons mené plusieurs types d'expérimentation.

Nos méthodes ont d'abord été testées en simulation afin d'évaluer leur fonctionnement avant de les utiliser dans d'actuelles applications d'apprentissage. Pour ce faire, nous avons développé différents modèles d'apprenants, afin de pouvoir éprouver nos méthodes selon différentes approches, un modèle d'apprenant virtuel ne reflétant jamais le comportement d'un apprenant réel. Les résultats des simulations montrent que le framework HMABITS permet d'obtenir des résultats d'apprentissage comparables et, dans certains cas, meilleurs qu'une solution optimale ou qu'une séquence experte.

Nous avons ensuite développé notre propre scénario pédagogique et notre propre serious game afin de tester nos algorithmes en situation réelle avec de vrais élèves. Nous avons donc développé un jeu sur la thématique de la décomposition des nombres, au travers de la manipulation de la monnaie, pour les enfants de 6 à 8 ans. Nous avons ensuite travaillé avec le rectorat et différentes écoles de l'académie de bordeaux. Sur l'ensemble des expérimentations, environ 1000 élèves ont travaillé sur l'application sur tablette.

Les résultats des études en situation réelle montrent que le framework HMABITS permet aux élèves d'accéder à des activités plus diverses et plus difficiles, d'avoir un meilleure apprentissage et d'être plus motivés qu'avec une séquence experte. Les résultats montrent même que ces effets sont encore plus marqués lorsque les élèves ont la possibilité de faire des choix.

Mots-Clés: Système Tuteur Intelligent, Enseignement Adaptatif, Théorie du Flow, Motivation Intrinsèque, Algorithme de Bandit Multi Bras, Modèle d'apprenant, Serious Game.

Abstract

Can computers teach people? To answer this question, Intelligent Tutoring Systems are a rapidly expanding field of research among the Information and Communication Technologies for the Education community. This subject brings together different issues and researchers from various fields, such as psychology, didactics, neurosciences and, particularly, machine learning.

Digital technologies are becoming more and more a part of everyday life with the development of tablets and smartphones. It seems natural to consider using these technologies for educational purposes. This raises several questions, such as how to make user interfaces accessible to everyone, how to make educational content motivating and how to customize it to individual learners.

In this PhD, we developed methods, grouped in the aptly-named HMABITS framework, to adapt pedagogical activity sequences based on learners' performances and preferences to maximize their learning speed and motivation. These methods use computational models of intrinsic motivation and curiosity-driven learning to identify the activities providing the highest learning progress and use Multi-Armed Bandit algorithms to manage the exploration/exploitation trade-off inside an activity space. Activities of optimal interest are thus privileged with the target to keep the learner in a state of Flow or in his or her Zone of Proximal Development. Moreover, some of our methods allow the student to make choices about contextual features or pedagogical content, which is a vector of self-determination and motivation. To evaluate the effectiveness and relevance of our algorithms, we carried out several types of experiments.

We first evaluated these methods with numerical simulations before applying them to real teaching conditions. To do this, we developed multiple models of learners, since a single model never exactly replicates the behavior of a real learner. The simulation results show the HMABITS framework achieves comparable, and in some cases better, learning results than an optimal solution or an expert sequence.

We then developed our own pedagogical scenario and serious game to test our algorithms in classrooms with real students. We developed a game on the theme of number decomposition, through the manipulation of money, for children aged 6 to 8. We then worked with the educational institutions and several schools in the Bordeaux school district. Overall, about 1000 students participated in trial lessons using the tablet application.

The results of the real-world studies show that the HMABITS framework allows the students to do more diverse and difficult activities, to achieve better learning and to be more motivated than with an Expert Sequence. The results show that this effect is even greater when the students have the possibility to make choices.

Keywords: Intelligent Tutoring System, Adaptive Teaching, Flow Theory, Intrinsic Motivation, Multi-Armed Bandit, Learner Model, Serious Game.

Contents

List of Figures

List of Tables

List of Algorithms

I do not know what I may appear to the world,
but to myself I seem to have been only like a boy
playing on the seashore, and diverting myself in now and then
finding a smoother pebble or a prettier shell than ordinary,
whilst the great ocean of truth lay all undiscovered before me.

Isaac Newton

List of Publications

Journals

1. Alexandra Delmas, Benjamin Clément, Pierre-Yves Oudeyer, and Hélène Sauzeon (2018). "Fostering health education with a serious game in children with asthma: pilot studies for assessing learning efficacy and automatized learning personalization". In: *Frontiers in Education*. DOI: 10.3389/feduc.2018.00099

2. Benjamin Clément, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2015). "Multi-Armed Bandits for Intelligent Tutoring Systems". In: *Journal of Educational Data Mining (JEDM)* 7.2, pp. 20–48. URL: https://hal.inria.fr/hal-00913669

Conference Papers

1. Benjamin Clément, Pierre-Yves Oudeyer, and Manuel Lopes (2016). "A Comparison of Automatic Teaching Strategies for Heterogeneous Student Populations". In: *EDM 16 - 9th International Conference on Educational Data Mining*. Proceedings of the 9th International Conference on Educational Data Mining. Raleigh, United States. URL: https://hal.inria.fr/hal-01360338

2. Benjamin Clément, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2014b). "Online Optimization of Teaching Sequences with Multi-Armed Bandits". In: *7th International Conference on Educational Data Mining*. London, United Kingdom. URL: https://hal.inria.fr/hal-01016428

3. Benjamin Clément, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2014a). "Developmental Learning for Intelligent Tutoring Systems". In: *IEEE ICDL-Epirob - The Fourth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*. Genoa, Italy. URL: https://hal.inria.fr/hal-01061195

4. Benjamin Clément, Didier Roy, Manuel Lopes, and Pierre-Yves Oudeyer (2014). "Online Optimization and Personalization of Teaching Sequences". In: *DI : Digital Intelligence - 1st International conference on digital cultures*. Nantes, France. URL: https://hal.inria.fr/hal-01061211

Poster

1. Benjamin Clément, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2014c). *Optimisation et Personnalisation automatiques des parcours d'apprentissage dans les Systèmes Tutoriels Intelligents*. TICE. Poster. URL: https://hal.inria.fr/hal-01090900

Enseigner à vivre n'est pas seulement enseigner à lire,
écrire, compter, ni seulement enseigner
les connaissances basiques utiles de l'histoire, de la géographie,
des sciences sociales, des sciences naturelles.
Ce n'est pas se concentrer sur les savoirs quantitatifs,
ni privilégier les formations professionnelles spécialisée,
c'est introduire une culture de base qui comporte
la connaissance de la connaissance.

Edgar Morin

Résumé en Français

Le domaine des Systèmes Tuteurs Intelligents (STI) a vu son essor se renforcer ces dernières années, avec l'apparition des MOOCs, des serious games, et l'utilisation de systèmes digitaux dans les écoles et à la maison via le support des tablettes et smartphones. Une des problématiques inhérentes à l'éducation est :

> Comment peut-on fournit un enseignement adapté et personnalisé à chaque apprenant afin de permettre une expérience d'apprentissage la plus optimale et la plus motivante possible ?

Cette problématique est d'autant plus vrai et prend d'autant plus de sens avec les systèmes numériques où il est possible de concevoir des programmes qui peuvent s'adapter automatiquement aux profils des élèves, à leur progression et à leur caractéristiques d'apprentissage propres.

Dans une étude pionnière, Thomas W Malone (1980) a utilisé les théories de motivation intrinsèque proposées par Berlyne (1960) and White (1959) pour évaluer quelles propriétés des jeux vidéo pourraient les rendre intrinsèquement motivants pour les joueurs, et pour étudier comment ce type de caractéristique pourrait être utilisé pour distiller plus efficacement le contenu éducatif aux étudiants. En particulier, il a montré que les jeux vidéos sont intrinsèquement plus motivants lorsqu'ils incluent des objectifs clairs et de plus en plus complexes[1].

[1] On peut connecté ce phénomène à la théorie du Flow (M. Csikszentmihalyi and I. Csikszentmihalyi, 1975) qui postule qu'une personne est profondément immergé et engagé dans une activité lorsque celle-ci présente un challenge optimale.

Dans la lignée à ces travaux, l'amélioration de l'apprentissage des élèves est au centre des systèmes éducatifs (Dunlosky et al., 2013) et le progrès des technologies d'apprentissage et d'enseignement a le potentiel d'améliorer l'accessibilité et l'efficacité de l'enseignement à grande échelle. En effet, étant donné que les apprenants suivent généralement des cours dont la structure est similaire de manière linéaire, des systèmes de tutorat intelligents (STI) ont été proposés. Ils fournissent des environnements et une rétroaction adaptés aux besoins uniques de l'apprenant et des mesures objectives et utiles sur l'apprentissage qui seront utilisés dans la recherche en éducation (Anderson et al., 1995; Koedinger, Anderson, et al., 1997; Nkambou et al., 2010).

Une grande variété de sujets d'apprentissage ont été mis en œuvre dans des STI tels que : l'aide en géométrie (Roll et al., 2011), la programmation (Vaessen et al., 2014), la classification visuelle pour la résolution de problèmes (Crowley and Medvedeva, 2006), la dynamique des véhicules (Huertas and Juárez-Ramirez, 2013) ou l'apprentissage des langues (Mahmoud and El-Hamayed, 2016).

Comment des méthodes inspirées de la communauté STI pourraient-elles être conçues pour exploiter l'apprentissage machine et les sciences cognitives afin de personnaliser automatiquement les activités STI et de garder les élèves motivés ?

De plus, comment ces méthodes pourraient-elles être indépendantes de l'environnement d'apprentissage STI pour devenir utilisables à diverses fins éducatives ?

Ces questions sont au cœur du travail développé dans cette thèse de doctorat, qui fait partie du projet Kidlearn. Ce projet vise à développer des méthodologies et des logiciels qui personnalisent des séquences d'activités éducatives pour chaque apprenant afin de maximiser son apprentissage et sa motivation. En plus de contribuer à l'apprentissage et à la motivation, l'approche vise également à réduire le temps nécessaire à la conception des systèmes STI et à proposer des méthodes générales qui peuvent être utilisées indépendamment du domaine éducatif pour lequel les systèmes sont mis en œuvre.

Pour répondre à la dernière question, un formalisme a été développé pour décrire un ensemble d'activités pédagogiques définies pour un STI particulier comme un espace d'activités paramétré. Des méthodes appropriées peuvent alors explorer et exploiter cet espace d'activité pour calculer les séquences d'activités. On peut imaginer des apprenants se former pour acquérir de nombreuses compétences différentes. Un enseignant peut aider les apprenants en leur proposant différentes activités, telles que des questions à choix multiples, des opérations abstraites à calculer au crayon, des jeux où les objets doivent être comptés par manipulation, des vidéos ou autres. Ce formalisme, présenté dans le chapitre 2.1, permet de définir ces différentes possibilités. C'est la première contribution de ce travail. Le défi consiste alors à trouver la séquence d'activités qui maximise le niveau de compétence moyen sur l'ensemble des compétences.

L'objectif principal de cette thèse est le modèle de tutorat, qui choisit les activités de l'Espace d'activités à présenter à l'apprenant. Comme mentionné dans la section 1.1, la promotion de la motivation intrinsèque est un aspect très important à prendre en compte pour offrir une expérience d'apprentissage efficace et agréable. Les progrès en apprentissage ont été identifiés comme une mesure pertinente de la qualité des activités. Les algorithmes proposés dans le chapitre 2 sont basés sur l'hypothèse du progrès en apprentissage

(Oudeyer, Gottlieb, et al., 2016) présentée dans Sec. 1.1, mais présentent de légères différences avec ces travaux.

Le système qui sélectionne l'activité n'est plus un agent d'apprentissage mais utilise le progrès d'apprentissage (Oudeyer, Gottlieb, et al., 2016) pour devenir un "enseignant". Il sélectionne les activités pour l'étudiant sans lui donner la possibilité de choisir. C'est un point de vue différent de celui qui consiste à considérer la motivation intrinsèque comme un moteur de l'autogestion des agents/organismes. Cela soulève deux questions : la première est de savoir si le fait de forcer un apprenant à faire des activités basées sur les progrès de l'apprentissage peut réellement produire une motivation intrinsèque. Cependant, pour se rapprocher du modèle original, le chapitre 6 présente une expérimentation qui introduit des versions algorithmiques qui incluent la possibilité pour les étudiants d'avoir un choix dans la composante contextuelle ou pédagogique. La deuxième question est de savoir comment évaluer les progrès d'apprentissage d'un élève humain dans un STI.

Différentes possibilités sont explorées pour répondre à ces questions dans le chapitre 2, conduisant à des méthodes combinant le progrès en apprentissage avec des algorithmes MAB (Multi-Armed Bandit algorithms) pour amener les apprenants dans leur Zone de Développement Proximal (ZPD)[2] et les maintenir dans l'état de Flow par des activités adaptées.

[2] Le concept de ZPD a été introduit par Vygotsky (1930-1934/1978) et représente l'ensemble des activités qu'un apprenant peut faire avec de l'aide et ne peut pas faire sans et suggère que cet ensemble d'activités présente une valeur pédagogique particulière.

Plusieurs auteurs se sont penchés sur la conception des STI basée sur l'utilisation de la ZPD (Luckin, 2001; Murray and Arroyo, 2002; D. Wood and H. Wood, 1996). L'approche proposée dans cette thèse diffère en ce que la ZPD est défini approximativement par un expert, puis les algorithmes l'ajustent en fonction des réponses et des progrès de l'apprenant. Si plusieurs activités sont possibles, il pourrait être nécessaire de les explorer toutes afin d'estimer leur impact sur chaque composante du savoir (*KC*). Une telle exploration prend beaucoup de temps et fournirait des séquences d'apprentissage peu performantes. Au lieu de cela, les algorithmes sont initialisés avec une structure graphique, réalisée par un expert pédagogique, qui sert de base à l'exploration et à l'exploitation des activités.

Ces algorithmes permettent une expérience d'apprentissage personnalisée, s'appuyant sur une connaissance réduite du domaine... Deux variantes principales ont été développées : RiARiT et ZPDES. RiARiT utilise les relations entre les activités et *KCs* pour déduire le niveau de l'élève dans chaque *KC* et proposer ensuite des activités qui maximisent les progrès d'apprentissage de l'élève sur tous les *KCs* du ZPD de l'élève. ZPDES est plus simple, puisqu'il est basé sur des relations de base entre les activités qui définissent un graphique. Il utilise ensuite le succès empirique de l'élève et calcule les progrès d'apprentissage pour sélectionner les activités qui offrent les progrès d'apprentissage les plus élevés dans le ZPD de l'élève. Ces algorithmes sont regroupés dans le framework HMABITS (Chap. 2), qui est la principale contribution de cette thèse.

Pour évaluer ce framework, plusieurs outils ont été développés. Des environnements de simulation ont été mis en place, qui incluent des modèles d'élèves et de populations et permettent de tester des scénarios virtuels. Une interface utilisateur et un scénario d'enseignement[3] ont été développés pour tester les algorithmes avec de vrais apprenants. Un ensemble de mesures qualitatives et quantitatives a été conçu pour appuyer chaque expérimentation. Ces mesures portent soit sur les performances et l'apprentissage de l'agent virtuel et de l'apprenant humain, soit sur le profil de l'apprenant humain et ses caractéristiques psychologiques.

[3] Le scénario d'enseignement est un jeu marchand, où l'étudiant est soit un client, soit un marchand et doit effectuer des sommes d'argent pour acheter un objet ou rendre la monnaie. (Chap. 4.2)

Trois études ont été réalisées et leurs paramètres sont présentés dans leurs chapitres respectifs. De plus, certains points de la littérature spécifiques à certaines parties de la thèse sont présentés dans leurs chapitres respectifs.

La première étude (Chap. 3) a été publiée et présentée à la conférence : Educational Data Mining Conference à Raleigh en 2016 (Clément, Oudeyer, et al., 2016). Elle présente une comparaison entre POMDP[4] et ZPDES[5] dans un environnement virtuel. La contribution de cette étude est de montrer que le cadre HMABITS est une alternative viable au modèle POMDP. Les résultats montrent que le cadre HMABITS permet d'obtenir des résultats d'apprentissage comparables et, dans certains cas, meilleurs que la solution optimale mise en œuvre comme solveur POMDP.

[4] Partial Observable Markov Decision Process traduit par processus de décision markovien partiellement observable est une généralisation d'un processus de décision markoviens.

[5] On ne compare que l'algorithme ZPDES en raison d'une perte d'intérêt pour l'algorithme RiARiT résultant de la détection de limites critiques dans l'étude présentée dans le chapitre 5.

La deuxième étude (Chap. 5) a été publiée dans la revue : Journal of Educational Data Mining en 2015 (Clément, Roy, Oudeyer, et al., 2015). Elle présente une comparaison entre RiARiT, ZPDES et un algorithme qui met en œuvre une " séquence d'experts " basée sur la conception d'un expert pédagogique. Le but de cette étude est d'analyser les différences entre les comportements des algorithmes et d'évaluer leur impact sur l'apprentissage des élèves. Les expérimentations ont été réalisées dans un environnement virtuel et dans le cadre d'une étude utilisateur. Les résultats montrent que le cadre HMABITS permet d'obtenir de meilleures performances d'apprentissage en simulation que la Séquence Expert et permet à l'étudiant d'effectuer des activités plus diverses et plus difficiles dans l'étude des utilisateurs.

La troisième étude (Chap. 6) présente une étude utilisateurs avec cinq conditions expérimentales, qui comprennent une séquence experte comme référence, l'algorithme ZPDES, une variante de ZPDES qui n'exploite pas les progrès de l'apprentissage mais tire des activités au hasard dans la ZPD, et deux versions de ZPDES qui proposent deux types de choix différents à l'élève. La première condition avec choix propose un choix contextuel sans impact sur les paramètres pédagogiques de l'activité[6], tandis que l'autre propose un choix pédagogique et non un choix contextuel[7].

[6] L'objet avec lequel ils vont jouer, mais les paramètres de l'activité sont entièrement gérés par l'algorithme.

[7] Les élèves peuvent choisir parmi deux activités que l'algorithme a sélectionnées mais l'objet avec lequel ils vont jouer est aléatoire.

Le but de l'étude est d'évaluer l'impact de chaque condition sur les performances des apprenants et sur leur motivation. Un protocole expérimental particulier est proposé, avec le développement de plusieurs métriques psychologiques. Les données de cette étude sont encore en cours d'analyse. Les résultats préliminaires montrent que la version ZPDES avec choix contextuel présente le meilleur impact d'apprentissage et de motivation.

EN CONCLUSION, les observations faites durant les expérimentations semblent montrer que le framework HMABITS présente un fort intérêt pour la gestion et l'adaptation des séquences d'activités pour les apprenants. En plus de sa capacité à personnaliser efficacement les séquences d'activités, il a une faible complexité de calcul, ne nécessite pas de formation sur un ensemble de données, et il a beaucoup moins d'hypothèses par rapport aux modèles cognitifs et étudiants que d'autres systèmes. Cependant, pour obtenir ces résultats, la mise en œuvre d'un algorithme dans le cadre des HMABITS nécessite une définition bien structurée de l'espace d'activité et un calcul judicieux des progrès de l'apprentissage et nécessite d'évaluer empiriquement l'impact pédagogique de chaque paramètre d'activité.

It is, in fact, nothing short of a miracle
that the modern methods of instruction
have not yet entirely strangled
the holy curiosity of inquiry;
for this delicate little plant, aside from stimulation,
stands mainly in need of freedom;
without this it goes to wrack and ruin without fail.
It is a very grave mistake to think
that the enjoyment of seeing and searching
can be promoted by means of coercion
and a sense of duty.

Albert Einstein

1

From learning theories to digital systems for education

Humankind is a very social species. We are characterized by an extreme dependence on culturally transmitted information. We interact and exchange information with each other on a daily basis. These interactions allow us to learn and evolve. The evolution of humans has favoured the emergence of complex social behaviors along with the development of brain architectures managing these mechanisms (Bjorklund and Harnishfeger, 1995). Humans have (proportionally to body size) the largest neocortex of the animal kingdom. It is a brain zone mostly involved in the highest cognitive functions such as language acquisition, conscious thought and social learning. These functions allow us to attribute mental states as emotions, intents, beliefs or knowledge to oneself and to others as defined by the Theory of Mind by Premack and Woodruff (1978). It is a key function in human cognition and social interaction such as communication, collaboration or teaching (C. D. Frith and U. Frith, 2012).

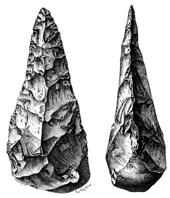


Figure 1.1: Drawing of a biface, considered as one of the first tools humans crafted. Drawing by A. de Mortillet.

Human culture and history are based on knowledge transmission through successive generations. From the first tools crafted by humans, such as the biface learned from imitation and social learning methods (Stade, 2017), to the most recent technologies, such as rockets or Artificial Intelligence (AI), results of modern science, humans have passed down knowledge to their peers leading humankind to evolve and progress. Human societies created schools to give access to fundamental knowledge and provide basic education such as literacy and numeracy to every people, in order to build a more enlightened and equal society. Moreover, with the emergence of the Internet at the dawn of the 21st century, the access to knowledge has skyrocketed to a level never reached in known human history.

Figure 1.2: SpaceX Falcon Heavy rocket launch at Kennedy Space Center, the 6th of february 2018. Photo by SpaceX on Unsplash.

Therefore, the understanding of the mechanisms involved in human learning presents a deep interest to improve human society and address the multiple challenges of education, such as keeping people motivated and engaged while learning and efficiently transmit knowledge. These functions are a central scientific question, explored during the last century and still the object of multiple research.

At the beginning of the 20th century, the main paradigm to study human behavior was behaviorism. It viewed mental events such as thoughts, ideas, attention, and consciousness as unobservable (Skinner, 1974) and could not be part of a science of psychology. It assumes that observable behaviors are either shaped or conditioned through the consequences they have along individual history (Watson, 1930) or mainly conditioned by reflex mechanisms responding to external stimuli (e.g. Pavlov's dogs[1]). However, it was highly criticized (Chomsky, 1959) because it did not consider internal mental states and only studied "primary" motivations as security need or survival while neglecting social and psychological motivation.

[1] Pavlov's dogs is a classical conditioning experiment made by the Russian physiologist Pavlov. He presented a stimulus (e.g. the sound of a metronome) to a dog and then gave him food, which makes gods salivate. After a few repetitions, the dog started to salivate in response to the stimulus before the food was presented.

During the same period, psychologists used several approaches to explain individual exploration processes and intrinsically motivated behavior. (Hull, 1943) described behavior as driven by the need of an organism to reduce specific deficits, like hunger or pain. Harlow (1950) proposed that learning is motivated by a manipulation drive and Montgomery (1954) proposed the exploration drive to have a role in learning. Moreover, White (1959) observed that organisms engage in exploratory and playful behaviors even in the absence of reinforcement or reward. These spontaneous behaviors appear to be done for the positive experiences associated with exercising and expanding one's capabilities.

In parallel, some studies tried to asses the impact of open-ended learning environments on each particular child development (Froebel, 1885; Montessori, 1948/2004). These approaches let the learners be active and allow them to explore their environment. The teacher's role is to scaffold challenges of increasing complexity and provide feedback to the students, rather than considering the learner as an empty box to fill with knowledge. These approaches were in accordance with constructivist[2] theories of learning developed by Dewey, Vygotsky, Piaget, or Bruner. They explained the importance of fostering curiosity and allowing free play in the classroom to keep the students engaged and motivated and improve learning efficiency.

[2] "Constructivism is a psychological theory that construes learning as an interpretive, recursive, nonlinear building process by active learners interacting with their surround – the physical and social world."
Fosnot and Perry (1996)

It was followed by the development of computer science and AI, which drew parallels between human thinking and computer processes (Newell, Simon, et al., 1972). It brought a conceptualization of mental functions based on the way computers handle activities such as memory storage and retrieval or problem solving (Anderson, 1985). It opened an important doorway to computational approaches of psychology.

These evolutions and developments led to the "cognitive revolution" (Gardner, 1993; Mandler, 2002) and to the birth of cognitive science. In particular, cognitive psychology emerged as the main paradigm to study human mind through several mental mechanisms involved in learning such as memory, attention, perception, creativity, problem solving and other human psychological function.

From this point, a lot of theoretical approaches have been proposed to study learning and used for educational purposes (Burton et al., 2013; Duffy and Jonassen, 2013; Mergel, 1998). In particular, Freeman et al. (2014) described the stimulation of intrinsic motivation as a fundamental component to consider for efficient learning and therefore for education. What stimulates the student's intrinsic motivation and how to integrate it into instructional design to make educational activities attractive and motivating is then a fundamental question to explore to address the challenge of providing an efficient teaching.

1.1 *Intrinsic motivation, a motor for learning*

People can have different motivations for committing themselves to an activity. Their motivation may come from the pleasure that an activity provides in itself or may rather come from external effects such as positive or negative outcome e.g, a reward system, promotion, evaluation or fear. The intrinsic motivation concept was proposed to explain the spontaneous human behavior to explore their environment (Berlyne, 1960) and Deci and Ryan (1985) developed the self-determination theory to describe the role of different kinds of motivation. They are classified according to different degrees of self-determination in the social and cognitive. Ryan and Deci (2000) distinguish two major kinds of of motivation, intrinsic and extrinsic motivation[3], which they define as :

[3] Self-determination theory also defines amotivation as the absence of motivation. People are amotivated when they do not apprehend the relation between their actions and the outcome. It causes a perception of the situation as the result of factors out of their control and may lead to the abandon of the activity. (Deci and Ryan, 1985)

> Intrinsic motivation is defined as the doing of an activity for its inherent satisfaction rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external products, pressures or rewards. Ryan and Deci (2000)

Indeed, from birth onward, humans are active, inquisitive, curious, and playful creatures. They display an ubiquitous readiness to learn and explore and they do not require external incentives to do so. This natural motivational tendency is a critical element in cognitive, social, and physical development because it is through acting on one's inherent interests that one grows in knowledge and skills. However, intrinsic motivation is not the only form of motivation in humans even if it is an important one. External motivation mechanisms are also at stake in human behavior regulation.

> Extrinsic motivation is a construct that pertains whenever an activity is done in order to attain some separable outcome. Extrinsic motivation thus contrasts with intrinsic motivation, which refers to doing an activity simply for the enjoyment of the activity itself, rather than its instrumental value. Ryan and Deci (2000)

Humans are often motivated by external factors. For example, a student who does his homework because he expects a parental reward (or sanction) is extrinsically motivated, since he is doing the work to attain the separate outcome of receiving a reward (or avoiding a sanction). Similarly, a student who studies to obtain a good job at the end of the study process, is extrinsically motivated. The instrumental value of work is at stake rather than the student self-interest in studying.

This distinction between intrinsic and extrinsic motivation has led psychologists and computational scientists to study the properties of activities that make them intrinsically motivating. Berlyne (1960) observed that the most rewarding situations were those considered to be between already familiar and completely new and developed the idea of "intermediate novelty" as a vector of intrinsic motivation. Hunt (1965) and Dember and Earl (1957) had a similar approach, observing that people prefer situations between completely certain and completely uncertain, and described intrinsic motivation as fostered by the difference between the people expectations and the real outcome of a situation.

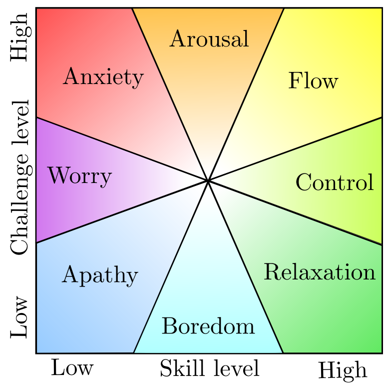


Figure 1.3: Mental state in terms of challenge level and skill level, according to the theory of Flow. Adapted from M. Csikszentmihalyi (1997)

These considerations connect with the self-determination theory, which describes someone's motivation to be linked to the degree of control they can have on their environment and themselves. M. Csikszentmihalyi and I. Csikszentmihalyi (1975) developed the analogous concept that, "beyond boredom and anxiety", people are intrinsically motivated by activities that present an optimal challenge allowing them to enter the state of Flow. It is described as the mental state where someone is fully immersed in a feeling of energized focus, full involvement, and enjoyment in the process of an activity leading to an optimal experience (M. Csikszentmihalyi, 1991). To enter this state of Flow, the skill of the person must match the challenge of the activity, i. e. the activity that presents an optimal challenge.

This theory can be correlated to the concept of Zone of Proximal Development (ZPD) introduced by Vygotsky (1930-1934/1978)[4]. The ZPD represents the set of activities (considered as a "Zone") a learner can do with some help and can't do without it and suggests that this set of activities presents a particular learning value. The initial concept of ZPD has been developed with the integration of the notion of scaffolding, a structure of "support points" for performing activities (Obukhova and Korepanova, 2009). According to Wass and Golding (2014), giving students the hardest activity they can do with scaffolding leads to the greatest learning gains.

[4] The book "Mind in Society" is meant to represent the last phase of Vygotsky's thinking and work. Although all of the base material was written in Russian, it has all been translated. Moreover, it has been significantly edited; and more importantly, the content was selected from four or more sources never intended by Vygotsky to comprise a book.

Inspired by the concepts of "intermediate novelty" (Berlyne, 1960) and "optimal challenge" (M. Csikszentmihalyi, 1991), which allow an intuitive explanation of behavioral manifestations of intrinsic motivation, computational learning theory has explored an alternative mechanism. The concept of *learning progress* has been hypothesized to be part of humans and animals innate mechanisms to generate intrinsic motivation (Kaplan and Oudeyer, 2007; Oudeyer and Smith, 2016) and and is used in computational models to generate intrinsic reward (Oudeyer and Kaplan, 2007):

> This hypothesis proposes that the brain, seen as a predictive machine constantly trying to anticipate what will happen next, is intrinsically motivated to pursue activities in which predictions are improving, ie, where uncertainty is decreasing and learning is actually happening. This means that the organism loses interest in activities that are too easy or too difficult to predict (ie, where uncertainty is low or where uncertainty is high but not reducible) and focuses specifically on learnable activities that are just beyond its current predictive capacities.
>
> Oudeyer, Gottlieb, et al. (2016)

Several works discussed that learning in itself does not have consequences on curiosity and motivation (Kang et al., 2009; Stahl and Feigenson, 2015) but on the contrary, the learning progress hypothesis proposes that experiencing learning in a given activity (rather than just intermediate novelty) triggers an intrinsic reward, and thus that learning in itself causally influences state curiosity and intrinsic motivation. Thus, this hypothesis argues that there is a closed self-reinforcing feedback loop between learning and curiosity-driven intrinsic motivation. Here, the learner becomes fundamentally active, searching for niches of learning progress and in turn memory retention is facilitated.

This kind of mechanisms was shown to be intrinsically motivating for people (Gottlieb et al., 2013) and can be used in active learning algorithms to select tasks to learn efficient models of the world (Baranes and Oudeyer, 2013; Chentanez et al., 2005; Lopes, Lang, et al., 2012; Oudeyer, Kaplan, and Hafner, 2007; Schmidhuber, 1991). In particular, Lopes, Lang, et al. (2012) showed that maximizing the empirical learning progress can drive intrinsically motivated model-based reinforcement learning and allows to learn non-stationary dynamics models of the world. Moreover Lopes and Oudeyer (2012) showed that when a learner must choose tasks to work on, selecting the ones maximizing learning progress is very efficient for a large variety of learner models. These results comfort the hypothesis that learning progress can be a good metric to assess the motivational and instructional quality of an activity.

The theory of Flow, the ZPD and the learning progress are three key concepts used in the methods developed during this PhD, whose goal is to combine these breakthroughs made in psychology with the possibilities offered by digital systems to improve education (Collins and Halverson, 2018).

1.2 *Digital technologies for education*

THE DIGITAL REVOLUTION of the end of the 20th century allowed progress in several domains and particularly in education and AI.

In a pioneer study, Thomas W Malone (1980) used theories of intrinsic motivation as proposed by Berlyne (1960) and White (1959) to evaluate which properties of video games could make them intrinsically motivating for players, and to study how this kind of features could be used to more efficiently distill educational content to students. In particular, he showed that video games are more intrinsically motivating when they include clear goals of progressively increasing complexity[5], when the system provides clear feedback on the performance of users[6], and when outcomes are uncertain, fostering curiosity.[7]. For example, he showed how arithmetic concepts could be taught in an intrinsically motivating scenarized dart video game. As an outcome of their studies, they could generate a set of guidelines for the design of education-oriented video games.

[5] This phenomenon connects to the theory of Flow (M. Csikszentmihalyi and I. Csikszentmihalyi, 1975)

[6] This connects to the self-determination theory (Deci and Ryan, 1985) which states that people's motivation is linked to the degree of control thay can have on their environment and themselves

[7] This connects to the concept of "intermediate novelty" as a motor of curiosity and motivation (Berlyne, 1960)

SIXTEEN YEARS LATER, Cordova and Lepper (1996) presented a study where a population of elementary schoolchildren trained on a game targeting the acquisition of arithmetic knowledge, with a scenario developed around a "space quest" story. Before the children trained, they were given a questionnaire where they could express preferences about school subjects, hobbies, television shows, books, and magazine. They observed that the addition of personalization in the math exercises, based on each child preferences, significantly improved their intrinsic motivation, engagement and learning. Moreover, these effects were even stronger if the software also offered the possibility for the children to personalize visual displays.

More recently, Walkington (2013) made a study where ninth grade students were given algebra story problems personalized to their out-of-school interests in areas such as sports, music and movies. Their results showed that personalization leads students to a better accuracy and a faster problem solving than without personalization.

IN LINE WITH THESE WORKS, improvements of student's learning are at the center of educational systems (Dunlosky et al., 2013) and making progress in learning and teaching technologies has the potential to improve education accessibility and learning efficiency on a large scale. Indeed, as learners generally study structurally-similar courses in a linear manner, Intelligent Tutoring Systems (ITS) have been proposed. They provide environments and feedback tailored to the learner's unique needs and useful objective metrics on learning to be used in education research (Anderson et al., 1995; Koedinger, Anderson, et al., 1997; Nkambou et al., 2010).

A large variety of educational topics have been implemented in these ITS such as : help in geometry (Roll et al., 2011), programming (Vaessen et al., 2014), visual classification for problem solving

(Crowley and Medvedeva, 2006), vehicle dynamics topics (Huertas and Juárez-Ramirez, 2013) or learning languages (Mahmoud and El-Hamayed, 2016).

Among the ITS field, there has been several approaches to optimize, personalize and adapt ITS properties to learners. Vandewaetere et al. (2011) reviewed the field of adaptation learning technologies and analyzed adaptive instruction to obtain a tripartie structure with three main components (Fig. 1.4). The first is the Source of adaptive instruction, i.e. to what it will be adapted, such as the learner learning style (Bunderson and Martinez, 2000; Sun et al., 2007), knowledge (Koedinger and Anderson, 1993) or preferences (Ray and Belden, 2007). The second component is the Target of the adaptive instruction, i.e. what will be adapted, such as the content (Sun et al., 2007) or the presentation (Milne et al., 1997). The third one corresponds to the Pathway between the two first components, i.e., how to adapt a Target to a Source, such as rule-based systems (Sun et al., 2007) or Bayesian-networks (Conati et al., 2002).

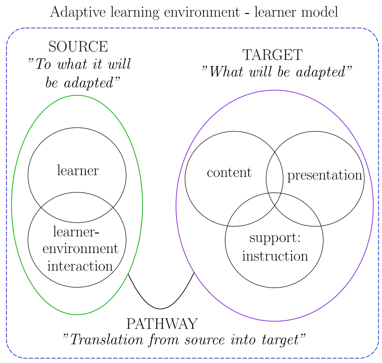


Figure 1.4: The tripartite structure of adaptive instruction. Adapted from Vandewaetere et al. (2011)

Monterrat et al. (2017) proposed a way to group several different aspects of adaptation learning technologies into one generic model, but a majority of approaches are limited to one or very few aspects of adaptation. This could be due to the large diversity of existing adaptation techniques (Naik and Kamat, 2015), but it could also be due to the difficulty to evaluate the impact of each different aspect if several are implemented in an adaptive system.

Let's take the example of a system which adapts, at same time, the content and the interface presentation based on the learner's profile, behavior and performance through the use of rules, learning analytics and neural networks. When this system is tested, if there is a control group that does not use this system compared to a group that uses it, and the groups that used the system present better results, it means the system has a positive impact. But what is the impact of each part of the system ? Does the content adaptation have the greatest impact ? or the profiles data ? or the tool used to adapt ? To answer these questions, several groups must be formed, corresponding to different possible combinations, as done in Monterrat et al. (2017). However, it can be difficult to gather the conditions required to perform such an experiments (e.g. large experimental population, rich content, adequate learning environment, ...).

Thereby, several approaches use data-driven strategies, hand-made rules strategies, priors knowledge about the student, or hybrid strategies to adapt the pedagogical content. The approaches that are the most relevant to the work presented in this PhD thesis are those where the personalization is made automatically, without particular prior assumptions about the students.

This is a very active line of work, and approaches vary in their assumptions about the knowledge domain, goals in terms of personalization and availability of student's data as discussed by Koedinger, Anderson, et al. (1997), Koedinger, Brunskill, et al. (2013) and Nkambou et al. (2010).

The framework of Partial Observable Markov Decision Process (POMDP) was proposed to select the optimal activities offered to the students based on the estimation of their level of acquisition of each skill (Rafferty et al., 2011). In general, the solution to a POMDP is difficult to obtain and approximate solutions have been proposed using the concept of envelope states (Brunskill and Russell, 2010) that, instead of tracking the full knowledge units, considers groups of units. In most cases the tutoring model incorporates the student model inside. For instance, in approaches based on POMDPs, the optimization of teaching sequences is made by assuming that all students learn in the same way. These approaches are potentially optimal, but require good student and cognitive models.

To build and manage a student model, many approaches rely on Knowledge Tracing methods (Corbett and Anderson, 1994) or their variants. Typically, these models have many parameters, and identifying all such parameters for a single student is a very hard problem due to the lack of data, the intractability of the problem, and the lack of identifiability of many parameters (Beck and Chang, 2007; Beck and Xiong, 2013). This often results in models which are inaccurate in practice. Some methods try to estimate those parameters from data (Baker et al., 2008; Dhanani et al., 2014; González-Brenes, Huang, et al., 2014; González-Brenes and Mostow, 2012), but these planning methods are for a population of students and not for a particular student, which was proven to be suboptimal (J. Lee and Brunskill, 2012).

Other approaches used reinforcement learning to provide hints during problem solving (Barnes et al., 2011) and to improve the adaptation of pedagogical strategies (Chi et al., 2011), or used Bayesian networks to model and decide how to help students (Gertner et al., 1998). Other approaches consider a global optimization of the pedagogical sequence based on data from all the students using ant colony optimization algorithms (Semet et al., 2003), but cannot provide a personalized sequence.

Then, how could methods, inspired from the ITS community, be designed to exploit machine learning and cognitive sciences to automatically personalize activities in ITS and keep student motivated ?

Moreover, how could these methods be independent of the ITS learning environment to become usable for various educational purposes ?

1.3 *The Kidlearn Project*

These questions are at the core of the work developed in this PhD thesis, which is part of the Kidlearn project. This project aims to develop methodologies and software which personalize sequences of educational activities for each individual learners, to maximize their learning and motivation. In addition to contributing to learning and motivation, the approach also aims to reduce the time needed to design ITS systems and propose general methods that can be used independently of the educational field for which the systems are implemented.

To answer the last question, a formalism was developed to describe a set of pedagogical activities defined for a particular ITS as a parametrized Activity Space. Suitable methods can then explore and exploit this Activity Space to compute activity sequences. We can imagine learners training to acquire many different skills. A teacher can help the learners by proposing different activities, such as: multiple-choice questions, abstract operations to compute with a pencil, games where items need to be counted through manipulation, videos, or others. This formalism, presented in chapter 2.1, allows to define these different possibilities. It is the first contribution of this work. The challenge is then to find the sequence of activities that maximizes the average competence level over all skills.

The main focus of this thesis is the tutoring model, which chooses the activities from the Activity Space to present to the learner. As mentioned in section 1.1, fostering intrinsic motivation is a really important aspect to take into account to provide an efficient and enjoyable learning experience. The learning progress was identified as a relevant measure of the activities quality. The algorithms proposed in chapter 2 are based on the learning progress hypothesis (Oudeyer, Gottlieb, et al., 2016) presented in Sec. 1.1, but present slight differences with these works.

The system that is selecting the activity is no more a learning agent but use the learning progress (Oudeyer, Gottlieb, et al., 2016) to become a "teacher". It selects the activities for the student without giving him the possibility to choose. This is a different view from seeing intrinsic motivation as a motor for agents/organisms self-management. This raises two question: the first question is whether forcing a learner to do activities based on the learning progress can actually produce intrinsic motivation. However, to get closer to the original model, chapter 6 presents an experiment that introduces algorithmic versions which include the possibility for the students to have a choice in either contextual or pedagogical component. The second question is how to evaluate the learning progress of a human student in an ITS.

Different possibilities are explored to answer these questions in chapter 2, leading to methods combining the learning progress with Multi-Armed Bandit algorithms (MAB) to bring learners inside their ZPD and keep them in the state of Flow through appropriate activities.

Several authors considered the design of ITS based on the use of the ZPD (Luckin, 2001; Murray and Arroyo, 2002; D. Wood and H. Wood, 1996). The approach proposed in this thesis differs in that the ZPD is defined approximately by an expert and then the algorithms adjust it based on the answers and learning progress of the learner. If many activities are possible, exploring them all could be necessary in order to estimate their impact on each Knowledge Component (*KC*). Such exploration is very time consuming and would provide under-performing learning sequences. Instead, the algorithms are initialized with a graph structure, made by a pedagogical expert, as a basis for activity exploration and exploitation.

These algorithms allow a personalized learning experience, relying on a reduced domain knowledge.. Two main variants were developed: RiARiT and ZPDES. RiARiT uses relations between activities and *KCs* to infer the level of the student in each *KC* and then propose activities that maximize the student's learning progress over all *KCs* in the learner's ZPD. ZPDES is simpler, since it is based on basic relations between activities which define a graph. It then uses the student's empirical success and computes the learning progress to select the activities which provide the highest learning progress inside the student's ZPD. These algorithms are grouped in the HMABITS framework (Chap. 2), which is the main contribution of this PhD.

To evaluate this framework, several tools were developed. Simulation environments were implemented, which include models of students and populations and allow to test virtual scenarios. A user interface and a teaching scenario[8] were developed to test the algorithms with real learners. A set of qualitative and quantitative metrics were designed to support each experiment. These metrics are either about virtual agent and human learner performances and learning, or about human learner profile and psychological characteristics. Three studies were carried out, and their metrics are presented in their respective chapters. Also, some points of the literature specific to certain parts of the thesis are presented in their respective chapters.

[8] The teaching scenario is a merchant game, where the student is either a client or a merchant and has to perform money sums to buy an object or give change. (Chap. 4.2)

The first study (Chap. 3) was published and presented at the Educational Data Mining Conference in Raleigh in 2016 (Clément, Oudeyer, et al., 2016). It presents a comparison between POMDP and ZPDES [9] in a virtual environment. The contribution of this study is to show that the HMABITS framework is a viable alternative to the POMDP model. The results show the HMABITS framework achieves comparable, and in some cases better, learning results than the optimal solution implemented as a POMDP solver.

[9] Only the ZPDES algorithm is compared due to a lose of interest for the RiARiT algorithm resulting from the detection of critical limits in the study presented in Chap 5.

The second study (Chap. 5) was published in Journal of Educational Data Mining in 2015 (Clément, Roy, Oudeyer, et al., 2015). It presents a comparison between RiARiT, ZPDES and an algorithm which implements an "expert sequence" based on the design of a pedagogical expert. The goal of this study is to analyze the differences between the algorithms behavior and to evaluate their impact on student's learning. The experimentations were carried out in a virtual environment and in a user study. The results show that the HMABITS framework achieves better learning performance in simulation than the Expert Sequence and allows student to do more diverse and difficult activities in the user study.

The third study (Chap. 6) presents a user study with five experimental conditions, which include an Expert Sequence as a baseline, the ZPDES algorithm, a variant of ZPDES which does not exploit the learning progress but, instead, draws activities randomly inside the ZPD, and two versions of ZPDES which propose two different kinds of choices to the student. The first condition with choice proposes a contextual choice without impact on the pedagogical parameters of the activity[10], while the other proposes a pedagogical choice and not a contextual one[11]. The goal of the study is to evaluate the impact of each condition on the learners performances and on their motivation. A particular experimental protocol is proposed, with the development of several psychological metrics. The data of this study are still being analyzed. Preliminary results show that the HMABITS version with contextual choice presents the best learning and motivational impact.

[10] the object with which they will play, but the parameters of the activity are managed fully by the algorithm

[11] The students can choose from two activities the algorithm has selected but the object they will play with is random

On a last note, parts of this PhD thesis are adapted from Clément, Roy, Oudeyer, et al. (2015) and Clément, Oudeyer, et al. (2016). Also, the images and pictures, used to illustrate the discussions, are either extracted from previous publications, made specifically for the manuscript, under creative comment licensing or free to be used by others.

Through education comes understanding.
Through understanding comes true appreciation.
All children are artists. The problem is
how to remain an artist once he grows up.

Pablo Picasso

2

Pedagogical Activity Sequence Manager

Contents

Let's imagine a teacher making his or her students go through a series of mathematical activities on a computer. Each student has a computer. The sequence of activities is not fixed: as a student goes through the activities, the computer tries to determine which activities to propose next to best help the student progress. Each student will go through a personalized sequence of activities which content is adapted depending on the student knowledge, preferences or performances. The hypothesis is that this system, an Intelligent Tutoring Sytem (ITS), helps students progress faster than following a standardized sequence of activities, such as using a textbook. In this chapter, such computerized system will be formally introduced and several methods of creating personalized activity sequences will be described.

A crucial part of any ITS is to decide which sequences of activities to propose to the student. In the framework presented here, this is the role of the aptly-named Activity Sequence Manager.

This Activity Manager requires a definition of an activity space in which the student will progress, acquiring skills or Knowledge Components (*KCs*). Some information about the domain or the student's knowledge state can be incorporated in the definition of the activity space to guide the Activity Manager more efficiently during the studying process. The expert knowledge provided to or required by the framework algorithms varies.

The goal of an ITS is to give students the activities which are most efficient at increasing their competence level at each point in time. It is assumed that a human expert in educational design first identifies a target set of skills to be acquired or improved and a corresponding target population of students. Then, he or she identifies the relevant study activities. For example, in the teaching scenario developed here, the set of skills/*KCs* includes the mastery of addition, subtraction and decomposition of integers and real numbers and their application to money calculation for children between 7-8 years. The associated activities space will span activity families whose parameters determine properties such as the kind of number involved (integer, decimals), the way they are visually represented (e.g. ab, cd€ or ab€cd), or whether the activity uses abstract number tokens or actual money coins and banknotes.

This chapter aims to introduce the fundamental concepts of this thesis. It first presents a definition of an Activity Space. The Activity Manager will explore this space to propose activity sequences. How Multi-Armed Bandits (MAB) can be used to manage in real time such activity sequences is then described. Next, several algorithms based on MAB are introduced. The first one is RiARiT, which explicitly estimates the level of the student's proficiency for different *KCs* to base its choice of activities. The second is ZPDES, which uses a "graph" of activities and tracks progress to choose the next activity; it does not use a student model based on *KCs*. Finally, the ExpSeq algorithm is presented. It implements an expert sequence without using any machine learning technology and is used as a baseline during several study presented in later chapters.

2.1 Activity Space

A pedagogical Activity Space is considered to be a set of activities that a learner can practice to acquire skills or knowledge components. An activity or exercise is characterized by multiple parameters a_i (difficulty, shape, type, ...) which can take different values v_j. For example, to work on mathematical skills, an exercise may have a type that works the addition and another type that works subtraction. "Addition" and "subtraction" are then two possible values for the parameter "type of exercise".

These parameters and their respective values define all the possible activities that can be instantiated inside the activity space. Depending on their nature and meaning, these parameters can be organized in different groups. Such group of parameters is noted as $H_x = a_1, \ldots, a_{n_x}$. In addition, these parameter groups can be structured hierarchically, since some parameters depend on others to be used in an activity.

Different types of exercises can require different skills, so the first group of pedagogical parameters (which will be the first level in the hierarchy) determines which type of exercise is selected. Several difficulty levels exist for each type of exercise, so different groups of parameters will determine which difficulty is chosen depending on the type of exercise (second level in the hierarchy). In this case, when one exercise type is selected, the parameter groups that determine the difficulty for the other types are not involved in the parametrization of the activity. Thus, not all parameter groups are necessarily used to define all the activities in the activity space. Therefore, an Activity Space is defined as a set of n_H hierarchical groups of parameters, $A = H_1, \ldots, H_{n_H}$.

An activity/exercise e is characterized as a particular combination of parameter values inside an activity space where values were selected for each hierarchical group of parameters involved. All the parameters needed to define an activity are instantiated to produce a unique combination of parameter values. The index u_i corresponds a selected value v_{u_i}, for a parameter a_i, used to generate an activity. To simplify the notation, u_i is noted as a given parameter selected value to generate an exercise to differentiate it with v_j which defines any values of a parameter. For a group H_x with m parameters, the selection of each parameter value produce a combination leading to a singular instantiation of this group $h_x = u_1, \ldots, u_m$.

After the selection process, a certain number of groups was instantiated, each producing an activity $e = h_1, \ldots, h_{n_e}$, which groups all parameter values that were selected to produce a unique combination. An activity space groups all possible distinct combinations of parameter values that can define an activity in this space. An illustration of a simple example of an Activity Space with the instantiation of an exercise is shown in figure 2.1.

How can this activity space be managed to propose relevant and personalized activities and offer a motivating and enriching experience to the learners ? Several methods are proposed below to answer this question.

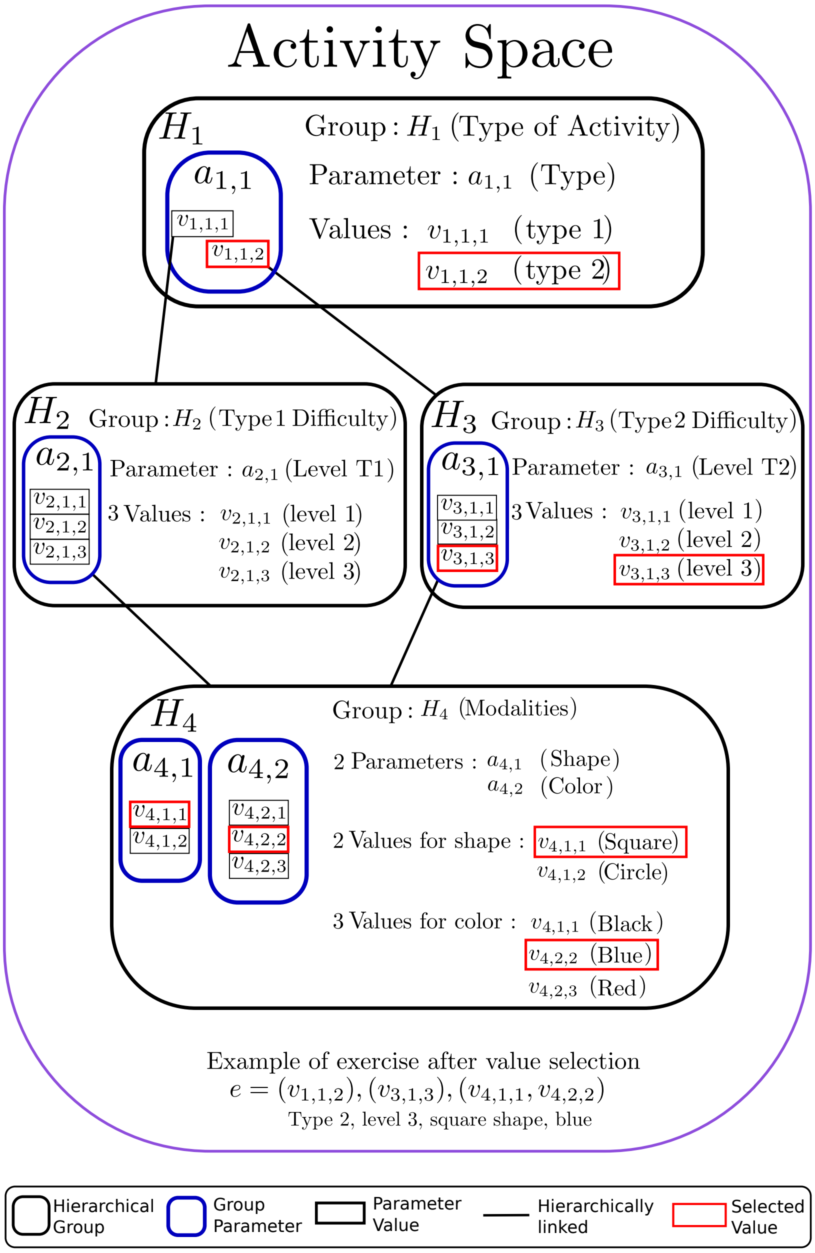


Figure 2.1: Illustration of an Activity Space with 4 groups of parameters, and a selection of values which lead to an example of an activity. A group is noted H_x, a parameter $a_{x,i}$ and a value $v_{x,i,j}$.

To compute an exercise, the values are first selected in the primary group H_1. Here 2 values are possible for 1 parameter. $v_{1,1,2}$ is selected thus $u_{1,1} = 2$ and the first group instantiation gives $h_1 = v_{1,1,2}$, meaning the exercise type is 2. This value is hierarchically linked to the group H_3. (H_2 is not used in this case).

Then, values in the group H_3 are selected. Here, $v_{3,1,3}$ is selected so $h_3 = v_{3,1,3}$, i.e. a difficulty level of 3. All levels are linked to the group of modalities H_4. Next, values are selected in the group H_4. For example $v_{4,1,1}$ and $v_{4,2,2}$ leading to $h_4 = v_{4,1,1}, v_{4,2,2}$ meaning a square shape and a blue colour.

So the final exercise resulting from the values selection is: $e = h_1, h_3, h_4 \leftrightarrow e = (v_{1,1,2}), (v_{3,1,3}), (v_{4,1,1}, v_{4,2,2})$

This means the exercise is of Type 2, level 3 with square shape and blue modalities. The Activity Space groups all the activities that can be generated in this way.

2.2 *Multi-Armed Bandit and Intrinsic Motivation theories to manage Teaching Sequences*

To address the challenge of managing activities in an Intelligent Tutoring System, the methods proposed below rely on state-of-the-art Multi-Armed Bandit techniques (MAB) (Auer et al., 2003; Bubeck and Cesa-Bianchi, 2012) and exploit the learning progress hypothesis. To use a casino analogy, multi-armed bandits describe the problem of finding the slot machine that provides the maximum reward, initially unknown, in a set of many different machines. To find the best machine it is needed to spend money exploring each one before being able to always bet on the best one. This boils down to what is called the "exploration/exploitation" trade-off[1] in machine learning and learning processes generally. Here, these approaches are adapted to ITS where the gambler is replaced by the Activity Manager, the choice of machine is replaced by a choice of activity parameter, and the reward is replaced by the student learning progress. Several ways of computing the learning progress are described in the next sections. It is assumed that activities which are currently estimated to provide a good learning progress must be selected more often as described in section 1.1.

Figure 2.2: A raw of slot machines at a casino in Las Vegas. License CC BY-SA 3.0

A particularity here is the reward (learning progress) which is non-stationary. This requires specific mechanisms to track its evolution. Indeed, a given activity will stop providing a reward, or learning progress, after the student reaches a certain mastery level of the skill or of the activity. Also, it cannot be assumed that the rewards are independent and identically distributed as different students will have different preferences, sensibilities or human factors. They may be distracted or make mistakes when using the system which can create spurious effects. Thus, the framework introduced here rely on a variant of the EXP4 algorithm, proposed initially by (Auer et al., 2003), which considers a set of experts[2] and make a choice based on the proposals of each expert. In case presented here, the experts are a set of variables that track how much reward each activity is providing (Lopes and Oudeyer, 2012). These bandit experts are used to evaluate the quality of each activity parameter value during the learner's working session.

[1] The exploration-exploitation trade-off is a fundamental dilemma whenever you learn about the world by trying things out. The dilemma is between choosing what you know and getting something close to what you expect ("exploitation") and choosing something you aren't sure about and possibly learning more ("exploration"). (Stafford et al., 2012)

[2] The general term "expert" is borrowed from Cesa-Bianchi et al. (1997). They use it to refer to strategies used in algorithms for"prediction with expert advice", "by combining the predictions of several prediction strategies".

Due to the combinatorial explosion of parameter values, only one MAB is not used for each possible combination of parameters values in the activity space but a set of simultaneous MAB is used for each group of parameters. The first alternative of considering a given arm for each activity would increase the number of arms. That would increase the number of parameters and the number of trials required to estimate learning progress and thus the learning time. Also, the approach presented here allows the algorithm to identify which features benefit some students more than others.

For example, to learn a particular skill, the same information may be presented in a written text, a video, a game, an audio track or another format. The knowledge the learner must acquire is the same in each case, but the format of the information differs and individual learners may be more receptive to a particular format[3].

[3] "Everybody is a genius. But if you judge a fish by its ability to climb a tree, it will live its whole life believing that it is stupid. The question I have for you at this part of our journey together is, What is your genius?" (Kelly, 2004)

A case can be imagined where a student works to learn mathematics; different activities are presented to him in a written format, and he almost never answers correctly. But when activities are presented to him in an audio format, he begins to succeed and progress. In this case, the problem is not about the mathematics skills he could learn, but rather his skills in reading. As another example, if an audio format is presented to a student with a hearing impairment, he will not perform and progress as well as with a written format. In light of this, the introduced method evaluates the relevance of and gives meaning to each feature and detects weaknesses and preferences of each student. The propositions it makes are more customized than the ones from an approach where particular combinations would be evaluated, but where features are not taken into account.

Each simultaneous MAB, used to sample each group of parameter, uses a bandit algorithm derived from EXP4 as presented in Lopes and Oudeyer (2012). The following process is described in Alg. 1.

For each parameter a_i inside a group, the quality of its values is evaluated by a bandit expert w_i. An expert track the reward provided by each value v_j on the last several sampling to compute its quality noted $w_i(v_j)$. At any given time, the value to use for each parameter is sampled according to the probabilities given by:

$$p_i = \tilde{w}_i(1-\gamma) + \gamma\xi_u \tag{2.1}$$

where $\tilde{w}_i$ are the normalized w_i values to ensure a correct probability distribution, ξ_u is a uniform distribution that ensures sufficient parameter exploration and γ is the exploration rate, tuned to make the exploration wide or narrow. This sampling methodology leads to stochastically select a value, proportionally based on its quality and γ. For low values of γ, the parameter value is chosen mostly based on its quality, whereas for high values of γ, low quality parameter values have a higher probability of being picked, which means a high exploration rate. The set of experts correlated to H_x is noted $W_x = w_1, \dots, w_{n_x}$. From now on, a Stochastic Activity Space A^S is considered to be a set of tuples (H_x, W_x).

To generate an activity, this process is done recursively on the hierarchical groups that are involved in the activity generation, in accordance with the hierarchical dependencies between the groups of parameters. As describe in Alg. 2, it starts by the instantiation of the primary group of parameter H_1 and is followed by the instantiation of the groups that are iteratively selected according to their dependencies. This leads to a stochastic draw of activity, resulting from the combination of each parameter value sampled depending on the evaluation of their quality by each expert. An abstract illustration of an activity generation is presented in figure 2.3.

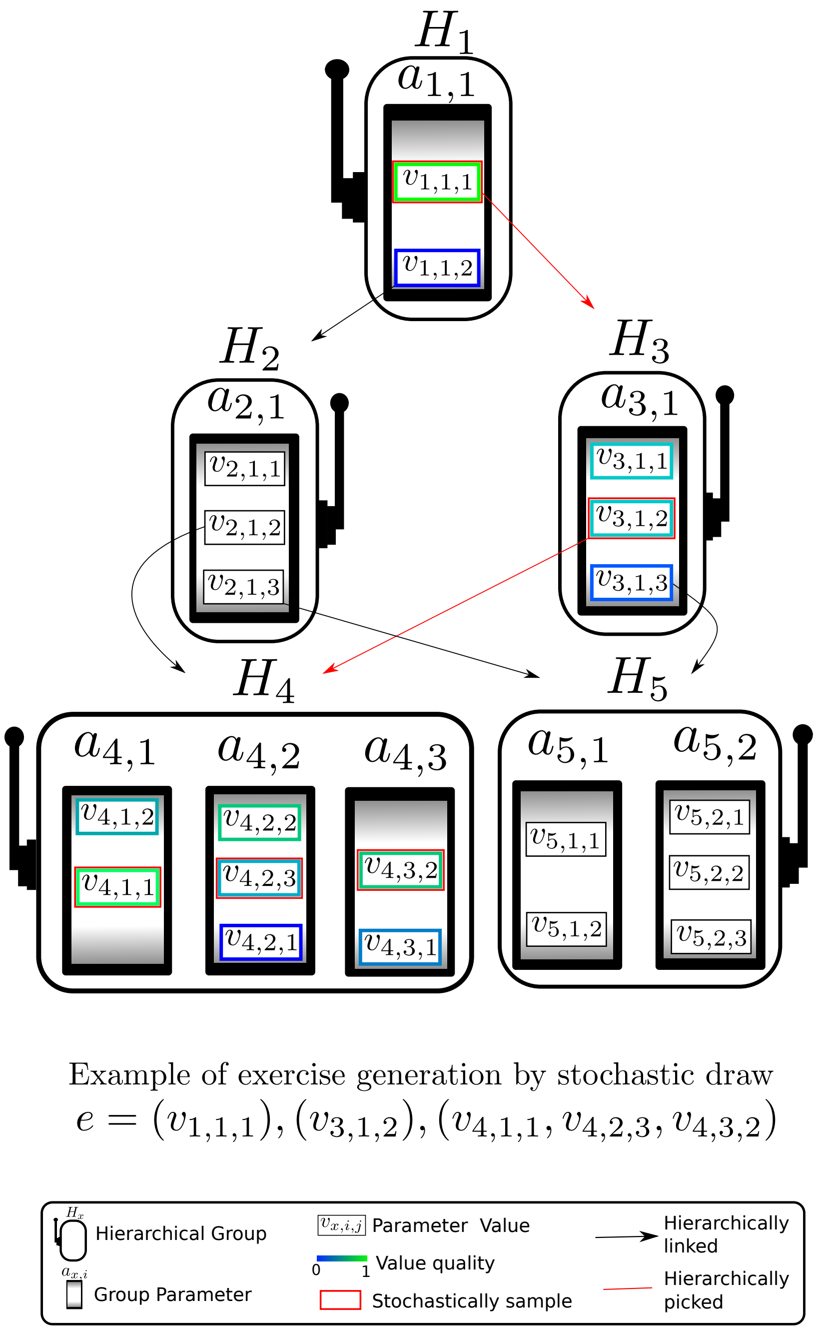


Figure 2.3: Hierarchical Multi-Armed Bandit with 5 groups of parameters and a selection, by stochastic draw, of an example of activity. A group is noted H_x, a parameter $a_{x,i}$ and a value $v_{x,i,j}$.

The primary group H_1 has one parameter $a_{1,1}$, for this parameter, the first values is evaluated to have a higher quality than the second one: $w_{1,1,1} \geq w_{1,1,2}$. There are then more chances for the first value to be sampled. Here, the result of the stochastic sample is that $v_{1,1,1}$ is drawn.

$v_{1,1,1}$ is linked hierarchically to H_3, which is the next to be instantiated. The two first values have the same medium quality (same chance to be drawn), and the last one has a low quality (less chance of being drawn). $v_{3,1,2}$ is drawn, which has a dependency with the group H4.

H4 is then instantiated and has three parameters. The first parameter $a_{4,1}$ has its first value evaluated to be more interesting than its second one. The quality for the second parameter is ordered as $w_{4,2,2} \geq w_{4,2,3} \geq w_{4,2,1}$ and for the third parameter $w_{4,3,2} \geq w_{4,3,1}$.

The three parameters are sampled simultaneously, and $v_{4,1,1}$, $v_{4,2,3}$ and $v_{4,3,2}$ are drawn. Even though $w_{4,2,2} \geq w_{4,2,3}$, $v_{4,2,3}$ had a chance to be drawn, following the process of exploration. The result of the activity generation is: $e = (v_{1,1,1}), (v_{3,1,2}), (v_{4,1,1}, v_{4,2,3}, v_{4,3,2})$.

Algorithm 1: Procedure to stochastically sample group parameter values according to their quality evaluation.

Require: Group H_x of m parameters a_i with their n_i values v_j
Require: Set W_x of m experts w_i for each parameter
Require: γ rate of exploration
Require: distribution for parameter exploration ξ_u

procedure sampleValues(H_x, W_x)
  for $i = 1 \dots m$ **do**
    $\tilde{w}_i \leftarrow \frac{w_i}{\sum_{j=0}^{n_i} w_i(v_j)}$
    $p_i \leftarrow \tilde{w}_i(1-\gamma) + \gamma\xi_u$ (Eq. 2.1)
    $u_i \leftarrow$ value sampled from a_i proportionally to p_i
  $h_x \leftarrow \{u_1, \dots, u_{n_x}\}$
  return h_x

Algorithm 2: Activity generation procedure based on an Activity Space and Hierarchical Multi Armed-Bandit mechanisms.

Require: A Stochastic Activity Space A^S, set of tuples (H_x, W_x)

procedure genActivity(A^S)
  {Initialize}
  Instantiate primary group $h_1 \leftarrow sampleValues(H_1, W_1)$
  $i \leftarrow 1$
  {Recursive sample}
  while h_i require to instantiate a group H_x **do**
    $i \leftarrow x$
    $h_i \leftarrow sampleValues(H_x, W_x)$
return $e = h_1, ..., h_i$

Once an activity is generated, this activity is proposed to a learner to work on and answer to. After answering, the algorithm retrieves his answer. Each time an exercise is given and answered, the expert of each parameter value u_i used in the activity is updated:

$$w_i(u_i) \leftarrow \beta w_i(u_i) + \eta r \tag{2.2}$$

where r is a reward that measures the benefit the activity gives to the learner in terms of progress. The variables β and η define the tracking dynamics of this estimation, which is the compromise between the old rewards and the new ones brought by the last activity. This mechanism allows the experts to assess and update the quality of each parameter value, used over time, based on the student learning. Ways to compute this reward are proposed in sections 2.3 and 2.4.

A pure selection, based solely on the previous considerations, would explore all possible activities that could be generated in the activity space from the start of the work process. This would have two drawbacks. First, the type and difficulty of the exercises proposed could change too often and reduce the learners' motivation and engagement. This could lead to a reduction in learners' motivation and engagement. Second, it might not be possible to explore all activity parameters to estimate their learning progress providing. To ensure that learners remain in challenging but possible to achieve areas and to be able to assess the quality of each parameter, a mechanism to limit exploration is introduced.

Inspired by the Zone of Proximal Development theory (Vygotsky, 1930-1934/1978) and the concept of Flow (M. Csikszentmihalyi and I. Csikszentmihalyi, 1975), a pedagogical expert has the possibility to specify rules that define an evolving set of possible/activated activities, judged relevant for the student. These activities keep the student in the zone of Flow or in the Zone of Proximal Develpment (ZPD) based on his successive results. The goal is to propose activities that are neither too easy nor too difficult, without having to try all possible activities. The different possible activities proposed by the algorithm are then the active ones which are inside the ZPD. The use of the ZPD offers three advantages: it helps to improve motivation as discussed before, it further reduces the need of quantitative metrics for the educational design expert and it provides a more predictive choice of activities.

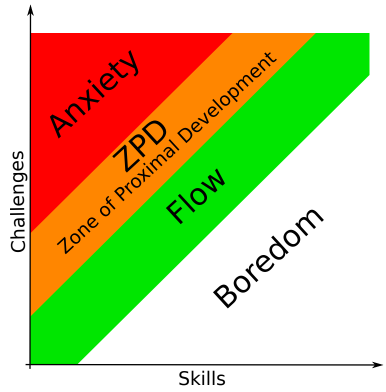


Figure 2.4: Basawapatna et al. (2013) propose the concept of Zones of Proximal Flow where they come up with the idea of the Zone of Proximal Development being located in between regions of Flow and anxiety.

The implementation of these principles is applied to the algorithm by the definition of rules that guide the bandits experts and restrict the exploration of the activity parameters. Theses rules define activation/deactivation mechanisms which allow the algorithm to activate and deactivate parameters values, depending on the evaluation of their relevance and the quality of the students learning process. As a consequence, the active parameters values generate a subset of all possible activities inside the activity space. The ZPD is defined here as a particular subset of active activities with its corresponding parameter values. The resulting algorithm, Hierarchical Multi-Armed Bandit for Intelligent Tutoring System (HMABITS), is shown in Algorithm 3.

Algorithm 3: HMABITS : Hierarchical Multi-Armed Bandit for Intelligent Tutoring System

Require: A Stochastic Activity Space A^S, set of tuples (H_x, W_x)
Require: Rules R^{ZPD} to define and update the ZPD
Initialize value of experts uniformly according to R^{ZPD}.
while *learning* **do**
  Generate activity $e \leftarrow genActivity(A^S)$ (Alg. 2)
  Get learner answer C
  Compute reward $r \leftarrow computeReward(e, C)$
  {Update greedy expert}
  for h_x in e **do**
    for u_i in h_x **do**
      $w_i(u_i) \leftarrow \beta w_i(u_i) + \eta r$ (Eq. 2.2)
  Update ZPD based on R^{ZPD}

In the following section, two algorithms are introduced: RiARiT, which has complex computational considerations and practical limits, and ZPDES which is more simple and the main studied algorithm. They vary according to three hypotheses: the learner's model, the rules that define the evolution mechanisms of the ZPD and the way to compute the learning progress. This leads to different ways of computing the internal reward and restricting the exploration of the HMABITS algorithm described above.

2.3 RiARiT: the Right Activity at the Right Time

A particular adaptation of HMABITS algorithm is proposed. It is strongly informed about the activities domain and the student. This information will be used to build a cognitive model of the student by estimating the knowledge level of the learner depending on his answer to each activity and the definition of a set of Knowledge Component (*KC*) related to each activity. This model will thus be used to manage the Zone of Proximal Development where the learner will train and evaluate the quality of each activity parameter inside it. First the modelling of the relation between *KCs* and pedagogical activities is presented. Then, the way to estimate the activities impact over the learner competence level is developed. The integration of these mechanisms into HMABITS to compute the reward and manage the Zone of Proximal Development is explained leading to the algorithm RiARiT, the Right Activity at the Right Time.

2.3.1 Relation between Knowledge Component and Activity Space

In general, activities can differ along several dimensions and take several forms (e.g. video lectures, interactive games, exercises of various types) as discussed in section 2.1. Each activity can provide an opportunity to acquire different skills, and may contribute differently to the improvement of several *KCs* (e.g. one activity may help a lot in progressing in KC_1 and only little in KC_2). Vice versa, succeeding in an activity may require to leverage differentially various *KCs*. While some parts of this relation can be similar across individuals, its details will differ for every student. Still, an intelligent learning environment might use this relation in order to estimate the level of each student. Several approaches have been introduced to describe such relation between activities and *KCs* and Desmarais (2011) made a comparison of such approaches.

For an activity e and a knowledge component KC_k, a quantity $q_k(e)$ is defined. It can be seen as the encoding of how well the activity e can help to learn the competence KC_k or as the required competence level in KC_k to master the activity e. To clarify the method description, this kind of quantity will always be referred to as the q-value[4]. Since activities are parameterized, there can be a huge number of activities and estimating empirically the impact of each combination of parameters over competence levels for each *KC* may be impossible to estimate for the expert. To address this issue, the parameterized activity space factorization described in section 2.1 is used and q-values are defined for each sub-element of this space.

[4] The q-values notion take its inspiration from the work of Barnes (2005) where she defines a Q-matrix or "attribute by item incidence matrix" to define relations between questions (activities in our case) and concepts (*KCs* in our case) and contains a one if a question is related to a concept and a zero if not.

Relations between *KCs* and hierarchical group sub-element are first defined. For a parameter a_i and a value v_j of group H_x, the q-value $q_k(v_j)$ represents the competence level a learner can acquire about a knowledge component KC_k with the value v_j related to a_i.

This q-value is represented as a continuous number between 0 and 1, where 1 means the parameter value provides all the competence level the learner can acquire with the parameter a_i for this KC, 0.5 means the value provide 50% of this competence level and 0 means the value does not allow to acquire this KC. A vector regrouping information about each value of a parameter can be defined as $q_k(a_i) = q_k(v_1), \ldots, q_k(v_{n_i})$.

To illustrate this, let's take the example of an activity, which allows to learn KC_1, with a parameter a_1 managing difficulty with 4 different level/values. The last level allows the learner to acquire 100% of the competence the activity can provide, the third level allows 75%, the second 50% and the first 25%. The corresponding vector is then $q_1 = \{0.25, 0.5, 0.75, 1\}$.

As described before, each hierarchical group H_x has several possible instantiations depending on the values selected during the activity generation process. From these considerations, $q_k(h_x)$ is defined as the q-value relating to the group of m parameters H_x and the Knowledge Component KC_k. The instantiation of H_x with particular selected values u_i is $h_x = u_1, \ldots, u_m$. Then, $q_k(u_i)$ represents the q-value of a selected value for the parameter a_i used in h_x. The $q_k(h_x)$ is computed as the product of each $q_k(u_i)$:

$$q_k(h_x) = \prod_{i=1}^{m} q_k(u_i). \tag{2.3}$$

However, for a group H_x, parameter values can be hierarchically linked to other groups of parameters. This induces a computational dependency of the q-value $q_k(u_i)$. Let's take the example of a parameter group which manages the type of exercise and groups that manages the difficulty level of each type. The first type of exercise allows to acquire 50% of the competence KC_1 and the second 100%. Then if there are 4 difficulty levels, the first level of type 1 would allow to acquire 25% of the competence level a learner can acquire with type 1. The type 1 allows to acquire 50% of the competence activities can provide. So, an activity of type 1 and level 1 would allow to acquire 12, 5% (0.25×0.5) of competence provided by the activity.

The parameter group H_{v_j} is the group linked to the value v_j one level below H_x in the hierarchical structure and h_{v_j} its instantiation. Then $q_k^H(u_i) = q_k(h_{u_i})$ is the hierarchical component of the q-value $q_k(u_i)$, relating to KC_k, depending on the hierarchical group below the selected value u_i. This leads to the final expression of $q^k(u_i)$:

$$q_k(u_i) = q_k^H(u_i) q_k(v_{u_i}) \leftrightarrow q_k(h_{v_{u_i}}) q_k(v_{u_i}) \tag{2.4}$$

Where $q_k(v_{u_i})$ is the q-value of the selected value v_{u_i} related to his parameter a_i. If u_i has no link with other group, $q_k^G(u_i) = 1$ and the $q_k(u_i)$ only depend on the q-value defined for the selected parameter value. Then $q_k(h_x)$ can be expressed for each instantiated group according to Eq. 2.3 and Eq. 2.4 as:

$$q_k(h_x) = \prod_{i=1}^{m} q_k^H(u_i) q_k(v_{u_i}) \tag{2.5}$$

A set of R Tables is established to define these q-values. The pedagogical expert will be able to complete these tables to describe the impact of each parameter value on the opportunity to acquire different skills. The general description of such table is shown in Table 2.1.

Table 2.1: R Table indicating the q-values related to n_c Knowledge Components for a group H_x of m parameters.

group H_x		Knowledge Components			
Parameter	Values	KC_1	$\dots$	KC_k	$\dots$
	$v_{1,1}$	$q_{1,1,1}$	$\cdots$	$q_{k,1,1}$	$\cdots$
a_1	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	v_{1,n_1}	$q_{1,1,n_1}$	$\cdots$	$q_{k,1,n_1}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$v_{m,1}$	$q_{1,m,1}$	$\cdots$	$q_{k,m,1}$	$\cdots$
a_m	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	v_{m,n_m}	q_{1,m,n_m}	$\cdots$	q_{k,m,n_m}	$\cdots$

The set of q-values, related to each KC_k and the group instantiation h_x is noted $q(h_x) = q_1(h_x), \dots, q_{n_c}(h_x)$. The q-values related to a particular activity e are then noted $q_k(e) = q_k(h_1), \dots, q_k(h_{n_e})$ and $q(e) = q(h_1), \dots, q(h_{n_e})$. The actual estimated competence level a learner can acquire over KC_k with this activity is given by the q-value of the primary group instantiation $q_k(h_1)$ which is the final result of the computational process and contains all the information about the activity estimation level for KC_k. Alg. 1 and Alg. 2 are adapted to include the q-values computing process leading to Alg. 4.

Algorithm 4: Specific valueSample procedure for RiARiT including q-values computation recursive process.

Require: Tuple (H_x, W_x)
Require: Set of n_c Knowledge Components KC_k
Require: R Table for H_x
Require: γ,ξ_u
procedure sampleValues(H_x, W_x)
  for $i = 1 \dots m$ **do**
    $\tilde{w}_i = \frac{w_i}{\sum_{j=0}^{n_i} w_i(v_j)}$
    $p_i = \tilde{w}_i(1-\gamma) + \gamma\xi_u$
    $u_i \leftarrow$ value sampled from a_i proportionally to p_i
    for $k = 0 \dots n_c$ **do**
      if u_i is hierarchically linked to a group H_{u_i} **then**
        $(h_{u_i}, q_k^H(u_i) \leftarrow sampleValues(H_{u_i}, W_{u_i})$ (Alg. 4)
      else
        $q_k^H(u_i) \leftarrow 1$
      $q_k(u_i) \leftarrow \prod_{i=1}^{n_{a_x}} q_k^H(u_i) q_k(v_{u_i})$ (Eq. 2.5)
  return $h_x, q(h_x)), (h_{u_1}, q(h_1)), \dots, (h_{u_m}, q(h_m)$

Algorithm 5: Specific genActivity procedure for RiARiT with h_x q-values computation. The recursive process of hierarchically sample the parameter values is now done in the *sampleValues* procedure.

Require: A^S, set of tuples (H_x, W_x) with related R Tables
procedure genActivity(A^S)
  $e, q(e) \leftarrow sampleValues(H_1, W_1)$ (Alg. 4)
  return $e, q(e)$

In the next part, the mechanisms to model the estimation of the learner competence level is presented. It is based on the computational process of the q-values described above and the performance of the student during the working session.

2.3.2 *Estimating activities impact on student model*

Similar to an extension to Knowledge Tracing (Wang and Heffernan, 2013), the competence level $l_{x,k}$ of a learner in a given KC, relating to a particular parameter group, is modelled as a continuous number between 0 and 1 where 0 means not acquired at all, 0.6 means acquired at 60% and 1 means entirely acquired.

Key to the approach is the estimation of the impact of each parameter over the student's competence level in each knowledge unit for each hierarchical group. This requires an estimation of the current competence level of the student for each KC_k. The introduction of regular tests, outside activities, that would be specific to evaluate each KC_k has to be avoided. It would have a sizeable probability to negatively interfere with the learning experience of the student. Thus, competence levels need to be inferred through stealth assessment (Shute, 2011) that uses indirect information coming from the combination of performances in activities and the R Table specified above. Also, no explicit assessment phases are made but any teaching activity is used for learning and assessment simultaneously, following stealth assessment principles (Shute, 2011; Shute et al., 2008).

The tracking of the competence levels l_x could have been achieved using Knowledge Tracing (Corbett and Anderson, 1994). The case presented here relies on a simplified version based on the previously defined relation between activities and KCs. Let us consider a given Knowledge Component KC_k and a given parameters group H_x for which the student has an estimated competence level of $l_{x,k}$. When doing an exercise e, the student can either succeed or fail. In the case of success, if the estimated competence level $l_{x,k}$ is lower than $q_k(h_x)$, the competence level of the student in KC_k for H_x is underestimated and should be increased. If the student fails and $q_k(h_x) < l_{x,k}$, then the competence level of the student is overestimated and should be decreased. Other cases provide little information, and thus $l_{x,k}$ is not updated and the reward is not used. For these two first cases a particular reward can be defined for each group of parameter:

$$r_{x,k} = q_k(h_x) - l_{x,k} \tag{2.6}$$

and use it to update the estimated competence level of the student according to:

$$l_{x,k} = l_{x,k} + \alpha r_{x,k} \tag{2.7}$$

where α is a tunable parameter that allows to adjust the confidence in each new piece of information. Accordingly, this also encodes that being always successful with a given instantiated group h_x cannot increase the estimated competence level $l_{x,k}$ above $q_k(h_x)$.

A crucial point is that the quantity $r_{x,k} = q_k(h_x) - l_{x,k}$ is not only used to update $l_{x,k}$, but is also used to generate an internal reward to update the bandit experts defined in section 2.2. This reward is computed by calculating the average reward over all KC_k for the instantiated group h_x of the group H_x:

$$r_x = \sum_{i=1}^{n_c} \frac{r_{x,i}^{used}}{n_c} \tag{2.8}$$

Indeed, it is assumed the result from doing an activity e allows to compute r_x which is a good indicator of the learning progress over all KC for the values used in h_x. The intuition behind is if you have repeated successes in an activity for which the required competence level is higher than your current estimated competence level, this means you are probably progressing and then the used values in the activity e are interesting. The RiARiT specific reward computing process is presented in Alg. 6.

Algorithm 6: Specific reward computation procedure for RiARiT

Require: Activity e
Require: Set of n_c q-values $q_k(e)$
Require: Set L^M of competence level estimation $l_{x,k}$
Require: Student answer C

procedure computeReward(e, $q(e)$, C)
    for h_x in e **do**
        for $k = 1, \ldots, n_c$ **do**
            $r_{x,k} = q_k(h_x) - l_{x,k}$ (Eq. 2.6)
            if correct(S) and $r_{x,k} > 0$ or wrong(S) and $r_{x,k} < 0$ **then**
                $l_{x,k} = l_{x,k} + \alpha r_{x,k}$ (Eq. 2.7)
                $r_{x,k}^{used}.insert(r_{x,k})$
        $r_x = \sum_{i=1}^{n_c} \frac{r_{x,i}^{used}}{n_c}$ (Eq. 2.8)
    return Total reward $r \leftarrow r_1, \ldots, r_{n_e}$

Other expert knowledge can be incorporated as a set of global rules R^{ZPD} on the algorithm as described in sec. 2.2. Indeed, for example the pedagogical expert knows that for most students it will be useless to propose exercises about decomposition of real numbers if they do not know how to add simple integers. Here the evolution of the ZPD can rely on explicit values of the estimated competence level of the leaner. Thus, the expert can specify minimal competence levels in given KC_k that are required to allow the algorithm to try a given parameter values and activate its related bandit expert. Each parameter value is only explored if the learner is already above this minimum threshold. The teaching experts are allowed to define threshold for which a given parameter is removed from the exploration, its bandit expert is deactivated.

This follows well known instructional design methodologies (Gagne and Briggs, 1974) and concords with theories of intrinsic motivation which clearly suggest motivation and learning improve if exercises are proposed at levels that are only slightly higher than the current level (Engeser and Rheinberg, 2008; Habgood and Ainsworth, 2011). The final algorithm, RiARiT, proceeds as presented in Alg. 7.

Algorithm 7: RiARiT algorithm is a particular adaptation of HMABITS. Its specificity lies in the KC requirements, the R Tables leading to the the internal learner and activity model, the reward computation and how ZPD is updated.

Require: A Stochastic Activity Space A^S
Require: Set of n_c Knowledge Componets KC_k
Require: Set L^M of competence level estimation $l_{x,k}$
Require: R Tables for each H_x and R^{ZPD} rules
Initialize bandit experts uniformly according to R^{ZPD}.
Initialize estimated competence levels $l_{x,k}$
while *learning* **do**
    Generate activity $(e, q(e)) \leftarrow genActivity(A^S)$ (Alg. 5)
    Get learner answer C
    Compute reward $r \leftarrow computeReward(e, q(e), C)$ (Alg. 6)
    {Update greedy expert}
    for (h_x, r_x) in (e, r) **do**
        for u_i in h_x **do**
            $w_i(u_i) \leftarrow \beta w_i(u_i) + \eta r_x$ (Eq. 2.2)
        Update ZPD: (de) activate w_i based on L^M and R^{ZPD}

More complex rules could be used to model more accurately the probability of succeeding depending on the differences between the required and estimated competence level, e.g., similar to the ELO system in chess (Elo, 1978) or any of its variants. The ELO system provides a statistical interpretation of the probability of winning or losing based on player levels. Similarly here the probability of correctly solving an exercise based on the difference of the required competence level and the estimated one could be considered.

RiARiT algorithm uses a lot of information about the domain. The teaching expert defines tables with the relation between the parameter values and the KC, and also a set of minimum competence levels to activate a new parameter value. The relation between the success of an exercise, the estimated competence level and the required competence level of an exercise allows two things: a) to estimate the level of the student; and b) to compute a reward for that activity. The information required for this algorithm is more online than a lot of other ITS systems but this amount of information might be really difficult to give for a teaching expert when the number of activities, or KC, is high. Automatic methods to fill such knowledge already exist and is an area of active research (Baker et al., 2008; Dhanani et al., 2014; González-Brenes, Huang, et al., 2014; González-Brenes and Mostow, 2012).

2.4 ZPDES: Zone of Proximal Development and Empirical Success

Another HMABITS adaptation is presented here. It aims at reducing even further the dependency on variable to model the cognitive and student models like R Tables which leads to a lot of work for the teaching expert and can be impractical in practice. The table is used to compute the reward for the bandit system and it is only domain dependent and not student dependent. Two sources of inspiration are used to simplify the algorithm: Zone of Proximal Development (C. D. Lee, 2005) and the empirical estimation of learning progress (Oudeyer and Kaplan, 2007). First, the reward computation in this particular case is presented and then the mechanisms built to manage the exploration of the activities through the use of ZPD concept are developed.

2.4.1 Compute reward using Empirical Success

As discussed before, focusing on activities that are providing more learning progress can act as a strong motivational cue (Gottlieb et al., 2013). Estimating explicitly how the success rate on each parameter group is improving will remove the dependency on the cognitive student model. For this, Eq. 2.8 is replaced with:

$$r_x = \sum_{t=T-d/2}^{t} \frac{C_t}{d/2} - \sum_{t=T-d}^{T-d/2} \frac{C_t}{d-d/2} \tag{2.9}$$

where $C_t = 1$ if the activity at time t was solved correctly. At the time T, the equation compares the success of the last $d/2$ samples with the $d/2$ previous samples, providing an empirical measure of the time evolution of the success rate. This reward allows to compute a measure of the quality of each activity parameter value, measuring how much progress it provided in a recent time window. Both extreme cases, when an activity is already mastered or when it is impossible to solve, will have a reward of zero. Moreover, parameter values providing a faster progress are assumed to be better than others. The algorithm to compute the reward is presented in Alg. 8.

Algorithm 8: Specific computeReward procedure for ZPDES

Require: Activity e
Require: Student answer C
Require: parameter d
procedure computeReward(e, C)
  for h_x in e **do**
    $r_x = \sum_{t=T-d/2}^{t} \frac{C_t}{d/2} - \sum_{t=T-d}^{T-d/2} \frac{C_t}{d-d/2}$ (Eq. 2.9)
  $r \leftarrow r_1, \dots, r_{n_e}$
  return r

Under this sole mechanism there are still too many activities to explore and there is no knowledges about the level of the students to guide exploration. To restrict the exploration, ordered relations between activity parameter values can be defined, leading to a "graph" governing the activity space, and a set of rules that define and manage the ZPD using activation/deactivation mechanisms, as described in section 2.2. Rules and ordered relations are not always defined for each parameter: there is a distinction between subsets of activity parameters that have a clear difficulty progression, and subsets that don't. For the example used from Sec 2.1, the difficulty levels have a clear ordering while the modalities don't. In practice, the management of the ZPD proceeds as follows. For activity parameters with no difficulty level relations between their values, a free exploration is allowed and so all of their values are always active. While for parameters that have a clear progression in difficulty, the values will be activated and deactivated depending on the success rate over all active values.

2.4.2 *Activity Space exploration*

The following mechanism is proposed to generally manage the ZPD. When the recent learner success rate over all active parameter values $\delta_{i,ZPD}$ reaches a value λ_{ZPD}, the ZPD is expanded to explore another parameter value $v_{i,j}$ by initializing its expert as : $w_i(v_{i,j}) = \min w_i(v^{ZPD})$. When the recent success rate for a particular value $\delta_{v_{i,j}}$ is higher than a threshold λ_d, this activity can be deactivated and removed from the active list of values. These two threshold allow to configure the general exploration behaviour of the algorithm inside the activity space and is illustrated in figure 2.5.

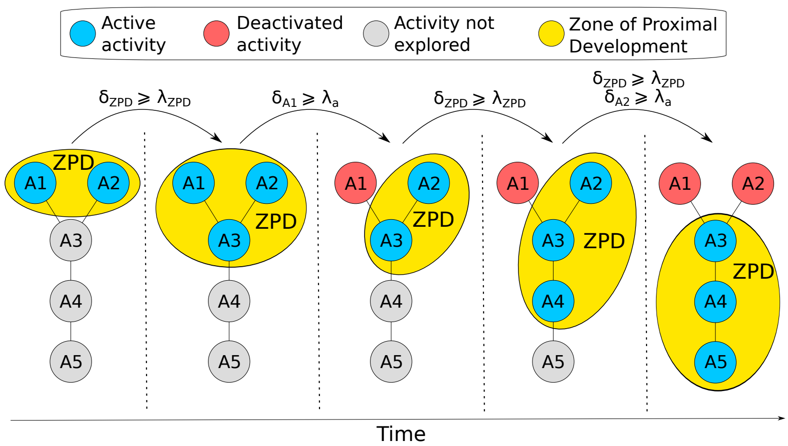


Figure 2.5: ZPDES exploration of an activity space, with δ_{ZDP} the success rate over all active activities, λ_{ZPD} the threshold to expand the ZPD, δ_{Ax} the success rate for the activity Ax, and λ_a the threshold to reach to deactivate an activity.

The main intuition of this process is that when there are some activities whose difficulty grows, the ZPD will have to grow at the same rate. When activities do not have a clear order of difficulty, or when the order might change from person to person, then it is necessary to allow wider exploration of the activities to accommodate individual differences.

Another kind of mechanism is added to allow a more precise and specific parametrization of the ZPD. Indeed, the algorithm needs to be able to activate and deactivate values when the conditions of exploration for an activity parameter depends on another set of parameters. If the value $v_{g,i,j}$ of parameter a_i of goup H_g requires a certain mastery level of value $v_{x,y,z}$, a threshold $\lambda_{v_{x,y,z}}$ is defined corresponding to the success rate a learner must reach with activities using $v_{x,y,z}$ to activate $v_{g,i,j}$. The requirements can be multiple, meaning a values activation can depend on multiple other values, parameters or group to be activated.

For example, the difficulty level for a particular type of exercise can require only the previous level to be mastered, or it can require various previous levels, or it can even require other types of exercises in different levels to be mastered. In a mathematics analogy, if a student works on a simple subtraction activity, he needs to master simple addition activities to be able to succeed. And if he works on hard subtractions with basic decimal number, he needs to master hard addition and basic decimal numbers first to succeed.

But this mechanism can lead to blockages in the exploration. If a type A of exercise is easy during 3 levels for a student, leading to a 100% success rate, the quality of this type will be very low. The algorithm will then select other types of excercises more often. But if the level 2 of type B needs a higher level of type A to be mastered, the algorithm will continue to propose more type B without being able to activate the level 2 until the required level type A is mastered.

To address this issue, a quality upgrading mechanism is added for values that are required. If ZPDES tries to activate a value parameter but is unable to do so due to a requirement, the qualities of required parameter values are increased. This way, ZPDES will exploit values needed to expanded the graph as a priority.

Basically, the first mechanism introduced to manage the ZPD is a simplification of the mechanism presented above. It is integrated to reduce the information needed to define ZPD rules and to allow a freer exploration of the activity space by the algorithm.

Several mechanisms were presented to compute the reward and manage the ZPD. They lead to the final adaptation of HMABITS called ZPDES presented in Alg. 9. One of the main advantages of these principles is the consideration of an empirical estimation of the learning progress. It has been proposed in artificial curiosity and intrinsic motivation systems by Oudeyer and Kaplan (2007). Instead of relying on a precise model of the learning system, with all limitations in terms of parameter identification and computational complexity, it is possible to create surrogate functions of the learning progress. These estimators are simpler, robust, and, even if not optimal, more flexible and adapt better to model errors and situations where the model assumptions are violated.

ZPDES algorithm is a simple implementation of HMABITS and can use very little information about the domain knowledge. The pedagogical expert only has to define an Activity Space with a "graph" and some rules to parametrize exploration/exploitation mechanisms. Even though a more complicated scenario requires more information form the teaching expert, it is still much less information consuming than RiARiT or other ITS systems.

Algorithm 9: ZPDES algorithm is a particular adaptation of HMABITS. Its specificity lies in how the reward is computed and how the ZPD is updated.

Require: A Stochastic Activity Space A^S
Require: R^{ZPD} rules
Initialize bandit experts uniformly according to R^{ZPD}.
while *learning* **do**
    Generate activity $e \leftarrow genActivity(A^S)$ (Alg. 2)
    Get learner answer C
    Compute reward $r \leftarrow computeReward(e, C)$ (Alg. 8)
    Update greedy expert
    for (h_x, r_x) in (e, r) **do**
        for u_i in h_x **do**
            $w_i(u_i) \leftarrow \beta w_i(u_i) + \eta r_x$
        Update ZPD: activate/deactivate w_i based on R^{ZPD}

2.5 ExpSeq: Expert Sequence of Activity

To have a baseline to study HMABITS framework, a simple algorithm has been designed. It is inspired by mastery learning strategy (Bloom, 1968) and instructional design whose reliability has been validated through several user studies (Roy, 2012) and it does not use any machine learning technology.

The algorithm manages a sequence of n_g groups of precomputed activities $g_i = e_{i,1}, ..., e_{i,m_i}$, leading to the sequence $S^G = g_1, ..., g_{n_g}$. In the sequence, the groups and activities inside each group are hierarchically organized by level of difficulty. Prerequisites, noted R^{ExpSeq}, can be added to restrain the group access, similar to the requirements defined for ZPDES. For example, a learner must reach a certain success rate for a particular activity to be able to access another activity or another group, following the same principle than ZPDES presented in section 2.4. A prerequisite can be defined by multiple restrictions over different activities or group of activities.

When a learner trains, ExpSeq registers the success rate for each exercise of the learner. To do so, it proposes consecutively n_E examples of the current exercise of the current group to which the learner answers. ExpSeq estimates an exercise $e_{i,j}$ to be mastered when the success rate δ_e over the n_E examples is above a certain threshold λ_M. After the n_E examples have been answered, different possibilities can occur. If the exercise is "mastered", ExpSeq updates the mastery level of the current group and proposes n_E examples of the next exercise.

If there is no higher level exercise in the group, then ExpSeq changes group. If the exercise is not mastered and there is no other available group (requirements are not satisfied), then ExpSeq proposes examples of the same exercise. Otherwise, if other groups are available, it changes group.

When ExpSeq changes the group, it proposes examples of the lowest non mastered exercise of the next group, if all the exercises of the group are mastered then it propose the highest exercise. If the current group is the last one available (due to the requirements or not), then it loops and changes to the first group. The global algorithm is shown in Alg. 10.

Algorithm 10: ExpSeq algorithm : Expert Sequence of Activity

Require: Set of n_g Group g_i of m_i exercises
Require: Student level for each activity group $l_1, \dots, l_{n_g}$
Require: Prerequisites R^{ExpSeq}
Require: Number of examples to propose n_E
Require: Threshold of mastery λ_M

Initialize group levels $l_1 = 0, \dots, l_{n_g} = 0$
Initialize current group index: $i = 1$
while *learning* **do**
    for $k = 1 \dots n_E$ **do**
        Propose example of exercise e_{i,l_i}
        Get Student Answer C_k
    Compute Success Rate $\delta_e \leftarrow \frac{\sum C_k}{n_E}$
    if $\delta_e \geq \lambda_M$ and $l_i < m_i$ **then**
        {Level up}
        $l_i \leftarrow l_i + 1$
    else
        {Change group}
        if Requirements allows it and $i \leq n_g$ **then**
            $i \leftarrow i + 1$
        else
            $i \leftarrow 1$

ExpSeq algorithm presents a simple implementation which allows to simply manage sequences of activities. It does not exploit any machine learning methods but is able to provide a poorly adapted sequence of exercises, similar to a list of exercises a teacher could provide in class to his students. The big difference here is that there is no teacher to help them when they are stucked. This algorithm is used as a baseline to be compared with the HMABITS framework to evalutate its quality.

2.6 Discussion

This chapter presented formalism considerations. The Activity Space, which identifies and includes different objects manipulated to generate activities, was first defined. Then an adaptation of the Multi-Armed Bandit framework, named HMABITS, was introduced. It manages an Activity Space and computes activity sequences adapted to the learner. Finally, particular cases of this algorithm which use different amounts and types of information to model the learner to adapt to each one was presented. An algorithm used as a baseline to be compared with these two methods was also presented.

However, although the current version of the different methods and algorithms developed was described, they are the product of a natural evolution over the years. They come from the reflections, results and limitations encountered during the development and the experimentations that were done. Different studies were carried out and the main ones are described in the following Chapters 3, 5 and 6. This way, the last experiment presented in Chapter 6 uses the current versions presented in this chapter, but the previous ones were made with older versions of the algorithms. The differences between the current versions and the old ones will be described in the preface of each related chapter.

Also, even if these are the most advanced versions of the algorithms, they still have some limitations. Some are critical and led to put some methods aside. Others are less serious and lead to further developments.

It must be kept in mind that these methods are only usable for problems that can be formalised as an Activity Space, as defined in Sec. 2.1. But not all pedagogical problems can be formalised like that and thus, these methods can't be used for all pedagogical situations. For example, it would be really hard to use these methods to make people train on dissertation.

One major and critical limitation concerns RiARiT. As said in the end of section 2.3, RiARiT needs a lot of information to define the different q-values used to model the learner and the impact of the different activities on student progress. Indeed, when used in practice, the teacher was able to provide the amount of work required to fill the different tables with the information relative to the teaching scenario but it was very time-consuming. This leads us to put further development of this algorithm aside after the related experiment presented in Chap. 5.

Another limitation is about ZPDES. After the experiments were carried out, the way to compute the reward for certain activity parameters and assess their quality does not seem to be very efficient. Indeed, the assumption that computing the success rate for a given parameter value regardless of the dependencies with other parameters can be a useful simplification in some cases but not always.

An example of usefulness is when there are several types of exercise with several difficulty levels. All student answers for the last considered steps are used to compute the reward relative to each exercise independently of the level. When a student answers right to a lower level after having answered wrong to a higher level, the quality of the exercise type will improve even if the success rate of the low or high level parameter is not improving. This can be seen as a weakness of the ZPDES model. But the fact that the type of activity allows the student to success and fails (creating challenge for the student but not too much), can keep him motivated to continue to work and improve with this exercise type. With this consideration, this simplification can be useful and harmless in the quality evaluation process.

On the other hand, for some types of parameters, this simplification can be a hindrance to accurately evaluate a parameters quality and can bring a loss of information about the quality of this parameter. Let's take the example of modalities. If a modality allows a learner to progress with all kinds of activities due to the learner's preferences and sensibilities, but some activities are too difficult for the learner to progress, the evaluation of the modality's quality can be disturbed. The computation will be done sequentially, interpreting each error the same way without considering other parameters. This can lead to a reduction of the quality of the parameter because of the last trial, since the student did not progress, even though it was due to another parameter.

Different mechanisms are being developed to counter this phenomenon. For example, the success rate can be computed relatively to each other parameter values; the quality of a modality will thus be evaluated for each level and activity type and not generally anymore. This new mechanism could allow a better appreciation of the modality's pertinence for each student.

3

A comparison between ZPDES and POMDP

Contents

This chapter presents a study published in Educational Data Mining conference in Raleigh in 2016 (Clément, Oudeyer, et al., 2016). The goal of this study is to compare ZPDES with the POMDP framework. RiARiT is not integrated in this study, which came after it was put aside due to the reasons explained in sections 2.6 and 5.4.

A major aspect of personalized education is to be able to identify the current level of students and how to address particular difficulties in the student learning process. The goal is to be able to choose online the activity that better addresses the challenges encountered by each particular student. Even two students with the same knowledge will require different activities to progress due to their previous experience, cognitive skills or preferences. This is a difficult challenge because as ITS are meeting the students for the first time, it is difficult to know the impact of each activity on their progress. A commonly used method is to exploit a population-wide model on how students learn and assume that they are all similar. The personalization in such an approach is limited to adapting to the student's knowledge levels but assumes the exercise's impact to be the same for all students with similar knowledge levels.

Different methods were proposed to handle this problem. One popular and well-known method is the Partially Observable Markov Decision Process (POMDP) framework which was proposed to select the optimal activities to propose to a learner (Rafferty et al., 2011).

This framework can find the optimal teaching trajectories for a given teaching scenario model if an accurate student model is provided which is not always possible in practice. The main drawback is the high computational complexity and as a consequence, only the simplest cases can be solved exactly. The goal of this study is to compare these methods with the HMABITS framework and specifically the algorithm ZPDES, which optimize learning in the short term (rather than in the long-term) and relies on simpler student models while being computationally efficient.

Some algorithmic considerations about the version of ZPDES used here and a small definition of POMDP are presented first. Then a student model used to compare the different algorithms is introduced. Next, some ways are proposed to model the heterogeneity in student population by considering that different students will not only have different learning parameters but also might have different dependencies between knowledge components. The experiments that follow evaluate how well a HMABITS can approach the optimal solution of a POMDP, and how the different algorithms behave when encountering a heterogeneous group of students.

3.1 Algorithmic considerations

The version of ZPDES used in the study presented in this chapter is an equivalent of the current version presented in the Sec. 2.4. The second mechanism described in Sec. 2.4.2 was not implemented yet but, because a very simple scenario is studied (there is only one group of parameter) only the general mechanism to manage the ZPD was needed. There is then no need for a mechanism to manage evolution restricted by group dependencies.

POMDP is a markovian decision process where the state is hidden and can only be inferred indirectly from the observations. A POMDP consists of a tuple $\langle S, A, Z, T, R, O, \gamma \rangle$ with S the state space, A the action space and Z the observation space. T is the transition model, it gives the probabilities $p(s'|s, a)$ of transitioning from state s to state s' with the action a. O is the observation model, it gives the probabilities $p(z|s, a)$ of having the observation z when action a is made in state s. R the cost model, it specifies the cost $r(s, a)$ of choosing action a in state s, and the discount factor γ gives the relation between immediate costs and delayed costs. With all these components, the solution of a POMDP is a policy that optimizes total discounted future reward.

This framework has already been used in the context of ITS (Rafferty et al., 2011). The learner's mastery is the hidden state s, learning is the transition between states, the probabilities that the learner gives a good answer are given by the observation model of the observation {correct, incorrect}. Perseus (Spaan and Vlassis, 2005) is used as solver to find the optimal policy for the POMDP problem.

3.2 Models

To compare ZPDES and POMDP in a virtual environment, a student model is needed to train with these algorithms. The goal is to test their robustness and adaptability in the case of students being different and not learning the same way. Means to model heterogeneity in a student population are then needed.

3.2.1 Student Model

Here is presented the student model used in the experiment, also called learner model in the literature. The goal is to have a generative model that can simultaneously be used to predict students behaviour, model their knowledge acquisition and track their mastery level. For this, a student model is built, shown in figure 3.1 similar to the Knowledge Tracing framework (Koedinger, Anderson, et al., 1997) and its variants. Similarly to (González-Brenes, Huang, et al., 2014), extra features are included in the model. The more realistic cases where each KC might depend on other KCs is particularly interesting. In most cases it is assumed that each exercise just depends on one KC and that they are independent, this is not realistic most of the time, and such dependencies have a strong impact on the learning sequences generated by the different algorithms.

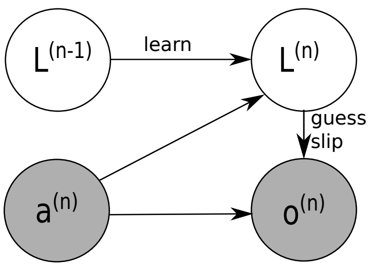


Figure 3.1: Graphical model of the Student model, with $L^{(n)}$ the hidden state of the student at step n, $a^{(n)}$ activity proposed, and $o^{(n)}$ the result obtained by the student.

The considered situation is a student who has a set of m KCs K_i to learn. A student's state at step n is represented by the state of each KC, $L^{(n)} = K_1^{(n)}, \ldots, K_m^{(n)}$, the global model is described in figure 3.1. Each KC is defined by its state, mastered ($K_i = 1$) or not mastered ($K_i = 0$). For each KC, there is an initial probability of mastering it $p(K_i^{(0)} = 1)$ which is always null in the experiments to make students learn all the KCs through activities. The emission probabilities are defined by the guess probability, i.e performing correctly without mastering the skill, and the slip probability, i.e performing incorrectly despite mastering the knowledge. Theses probabilities are constant. Finally $p(K_i^{(n)} = 1|L^{(n-1)}, a^{(n)})$ defines the probability of transition from not mastered to mastered K_i while doing activity a at step n and depending on the constraints between KCs and their states. An activity can be represented as a vector $a = \alpha_1, \ldots, \alpha_m$ where $\alpha_i = 1$ if the activity allows to acquire K_i, $\alpha_i = 0$ else.

The transition probability to learn a given KC K_i at step n is given by the following formula:

$$p(K_i^{(n)} = 1|L^{(n-1)}, a^{(n)}) = \alpha_i(\beta_{i,i} + \sum_{j \neq i}^{m} \beta_{i,j} K_j^{(n-1)}) \tag{3.1}$$

Where $\beta_{i,i}$ represent the probability to learn K_i without considering other KCs and $\beta_{i,j}$ represent the impact of the KC K_j on the probability to learn K_i. If a given KC does not need other KCs to be learned, the term $\sum_{j \neq i}^{m} \beta_{i,j} K_j$ is null.

For more simplicity, in the experiments, an activity a can provide an opportunity to acquire only one KC which induces an isomorphism between the knowledge space and the activity space.

3.2.2 Population model

The previous model can be used to describe a single student or an average model of a population. The goal is to understand the impact that the diversity of students has when the given sequence of activity is optimized considering the same parameters for all students. Such goal will be achieved by considering a canonical model and then make two types of disruptions: i) change the probabilities between the variables; ii) change the knowledge graph.

The first way is to disrupt the parameters in the model, i.e. the probability of transition, guess, and slip. To do that, each parameter is sampled from a gaussian distribution. The variance can be modified to increase the heterogeneity of the population. With a null variance, the whole population has the same parameters. The second way is to change the knowledge graph that changes the dependencies between the different knowledge. This type of disruption can be small like adding or removing a dependency, or it can be as critical as rearranging completely the organization of the dependencies. These two types of disruption are combined in the experiments.

$$LM_0 : \xrightarrow{0.2} K_1 \xrightarrow{0.2} K_2 \xrightarrow{0.2} K_3 \xrightarrow{0.2} K_4 \xrightarrow{0.2} K_5 \xrightarrow{0.2} K_6$$

$$LM_1 : \left.\begin{array}{l} \xrightarrow{0.2} K_1 \\ \xrightarrow{0.2} K_2 \end{array}\right\} \xrightarrow[0.1]{0.1} K_3 \xrightarrow{0.2} K_4 \xrightarrow{0.2} K_5 \xrightarrow{0.2} K_6$$

$$LM_2 : \begin{array}{l} \xrightarrow{0.2} K_1 \xrightarrow{0.2} K_2 \xrightarrow{0.2} K_4 \xrightarrow{0.2} K_6 \\ \xrightarrow{0.2} K_3 \xrightarrow{0.2} K_5 \end{array}$$

$$LM_3 : \xrightarrow{0.2} K_1 \xrightarrow{0.2} K_2 \xrightarrow{0.2} K_3 \xrightarrow{0.2} K_5 \xrightarrow{0.2} K_4 \xrightarrow{0.2} K_6$$

$$LM_4 : \xrightarrow{0.2} K_1 \xrightarrow{0.2} K_2 \xrightarrow{0.2} K_4 \xrightarrow{0.2} K_3 \xrightarrow{0.2} K_5 \xrightarrow{0.2} K_6$$

Figure 3.2: Knowledge graphs used in the simulations. LM_0 is the nominal knowledge graph, with LM_1 and LM_2 introducing small disruptions in the pre-requirements between KCs. LM_3 and LM_4 represent more critical disruptions that change the overall order of KCs.

Multiple knowledge graphs were used, shown in figure 3.2. The arrows represent the dependencies between KCs. For example, LM_0 represents a graph where the constraints between the different KC are ordered in a linear way. Here, $\beta_{1,1} = \beta_{2,1} = \beta_{3,2} = \beta_{4,3} = \beta_{5,4} = \beta_{6,5} = 0.2$ and all the others values of $\beta_{i,j}$ are null. Several different transformations and variants were created to model different needs of the students in terms of the order of the different KC.

LM_1 and LM_2 follow approximately the same overall sequence of KC, but considering two initial branches for the different KC. LM_1 considers that KC_1 and KC_2 are independent and any of them allows to learn KC_3. An expectation is that optimizing algorithms for a particular knowledge graph will also work for the other as the overall sequence of KC is respected, even if the strategy is no longer optimal. More critical disruptions in the knowledge graph was also created. LM_3 and LM_4 present an inversion between two KCs. For LM_3, KC_4 and KC_5 are inverted, what radically change the overall sequence of KCs. For LM_4, it is K_3 and K_4 that are inverted.

3.3 Experiments

The goal of the experiments is to compare the impact of the knowledge about the students on the online algorithms for exercises selection, namely POMDP and ZPDES. The heterogeneity of the student populations is changed to see how much disruption each algorithm is able to adapt. The comparative metric of performance is the average skill level overall knowledge and over time, for all the students in the population.

The results obtained are compared between two algorithms: POMDP and ZPDES. Each algorithm have different variants based on the knowledge included on each of them. POMDP relies on a knowledge graph and the parameters of such graph. Each variant of $POMDP_x$ is characterized by a specific student model used to find the optimal policy. ZPDES receives as information the knowledge graph, and some parameters describing how to traverse this graph, no particular assumption is made about the probabilities of knowledge acquisition. $ZPDES_x^H$ is a variant of ZPDES with the corresponding graph x and using the parameters that were used in another experiment in a real world situation mostly hand-tuned with the help of a pedagogical expert. $ZPDES_x^*$ also uses the graph x but the parameters to traverse the graph are optimized for that particular graph using a greedy search. During the optimization, the majority of parameters presents average results and only extreme parameters gave critical results.

3.3.1 Single model results

The first experiment does a sanity check to evaluate each algorithm in conditions where each student is the same in the population and each algorithm is configured for this model of student. POMDP is expected to have the best results and a goal is to see how far ZPDES is from the optimal solution. A Random strategy which selects one activity randomly among all possible is also presented in this first experiment to see the gain of the algorithms.

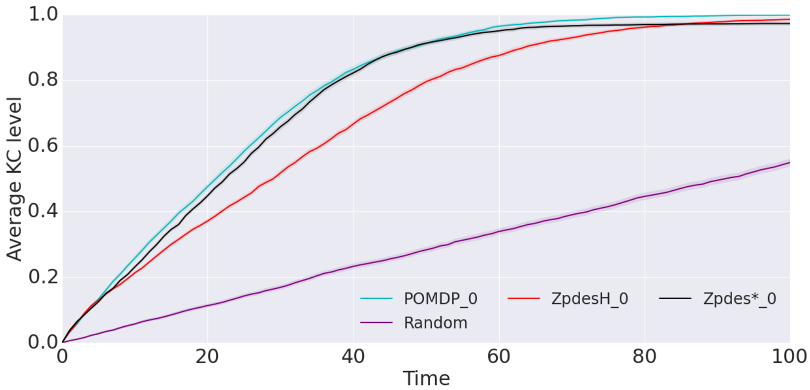


Figure 3.3: Evolution of the average skill level for 600 students modelled with LM_0 which activity are managed by POMDP, ZPDES*, ZPDES^H configured for LM_0. Shaded area represents the standard error of the mean.

Figure 3.3 shows the comparison of POMDP, ZPDES*, ZPDES^H and Random with a population of 600 students modelled with the knowledge graphs LM 0. POMDP is the best for all the models, closely followed by ZPDES*. ZPDES^H give a slower learning than the two others. Unsurprisingly, for one particular model, POMDP has the best performance. The optimized ZPDES is very close in performance to POMDP. The results are similar for models 1, 2, 3 and 4, the curves are not presented here for space reason.

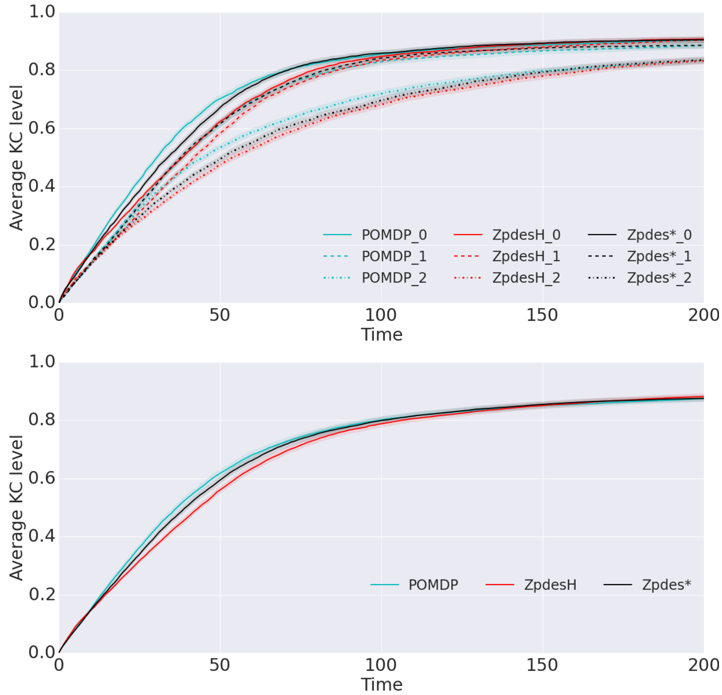
The combination of knowledge graphs and the activity exploration rules in ZPDES can be verified to provide a space of policies that is close to the optimal POMDP one. ZPDES^H presents the slowest population learning among the algorithms but, as its configuration was not optimized for any particular model, such result is expected.

These results show the algorithms behaving as expected and ZPDES having the potential to be close to optimal POMDP solution.

3.3.2 *Multi model results*

The main results of this study are presented below. POMDP, ZPDES* and ZPDES^H are compared when confronted with heterogeneous populations of students. The protocol of the experiments is as follows. First the information about a specific population of students is provided to each algorithm. Then the capability of each algorithm to address to different and diverse population of students is tested.

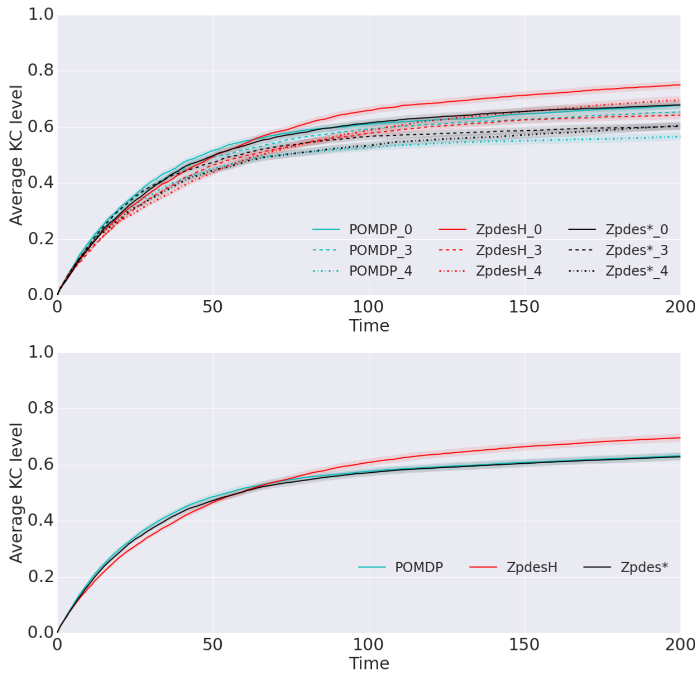
As described earlier, each algorithm is given information about a particular student model x, POMDP$_x$ receives the graph and the student model parameters, ZPDES$^*_x$ receives the graph and exploration parameters optimized for that same graph, ZPDES^H_x receives the graph and standard parameters for the graph exploration. Different versions of each algorithm are tested with a population composed of 600 students following 3 different knowledge graphs (200 per graphs). The probabilistic parameters of the student models in the population follow a Gaussian distribution.



Students 0,1,2 / Alg config 0,1,2

	Algorithm configurations rank			
	t 50		t 200	
Algorithm	Per conf	Over All	Per conf	Over All
$POMDP_0$	1		1	
$POMDP_1$	3	1^*	2	2
$POMDP_2$	4		3	
$ZPDES_0^H$	3		1	
$ZPDES_1^H$	3	3^*_{**}	1	1
$ZPDES_2^H$	6		3	
$ZPDES_0^*$	2		1	
$ZPDES_1^*$	3	2_{**}	2	2
$ZPDES_2^*$	5		3	

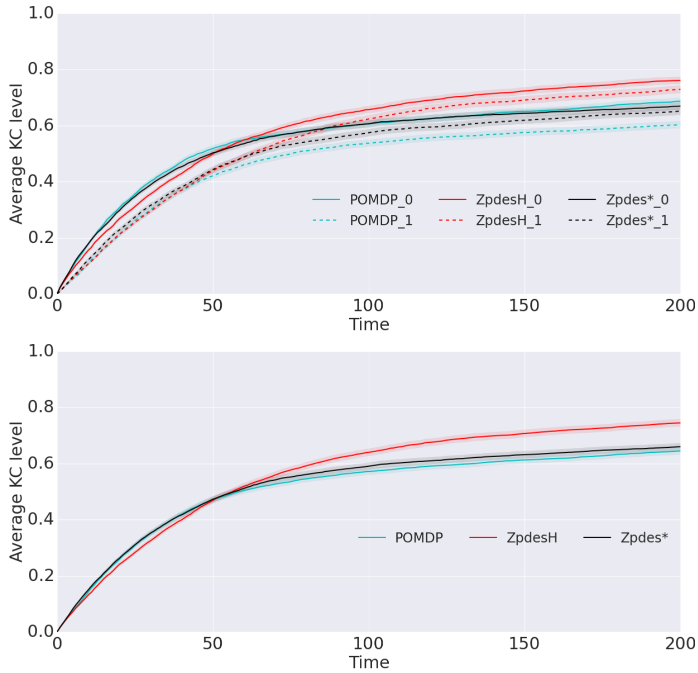
Figure 3.4: Evolution of the average skill level for 600 students ($LM_{0,1,2}$) with POMDP, ZPDES*, $ZPDES^H$ configured with graphs 0, 1 and 2 (indexed number). Each curve shows the average KC level of student populations over time for each algorithm configuration. There is not that much differences between each algorithms apart from $ZPDES^H$ being less efficient in early due to his wider exploration.



Students 0,3,4 / Alg config 0,3,4

	Algorithm configurations rank			
	t 50		t 200	
Algorithm	Per conf	Average	Per conf	Average
$POMDP_0$	1		2	
$POMDP_3$	2	1	3	2^*
$POMDP_4$	4		5	
$ZPDES_0^H$	2		1	
$ZPDES_3^H$	3	2	2	1^*_{**}
$ZPDES_4^H$	4		3	
$ZPDES_0^*$	2		2	
$ZPDES_3^*$	3	2	4	2_{**}
$ZPDES_4^*$	4		4	

Figure 3.5: Evolution of the average skill level for 600 students ($LM_{0,3,4}$) with POMDP, ZPDES*, $ZPDES^H$ configured with graphs 0, 3 and 4 (indexed number). Each curve shows the average KC level of student populations over time for each algorithm configuration. There is not that much differences between each algorithms in early but $ZPDES^H$ presents better results with time.



Students 2,3,4 / Alg config 0,1

Algorithm	Algorithm configurations rank: t 50, Per conf	t 50, Average	t 200, Per conf	t 200, Average
$POMDP_0$	1	1	3	2*
$POMDP_1$	4		6	
$ZPDES^H_0$	2	1	1	1^*_{**}
$ZPDES^H_1$	3		2	
$ZPDES^*_0$	2	1	4	2_{**}
$ZPDES^*_1$	3		5	

Figure 3.6: Evolution of the average skill level for 600 students ($LM_{2,3,4}$) with POMDP, ZPDES*, $ZPDES^H$ configured with graphs 0 and 1. Each curve shows the average KC level of student populations over time for each algorithm configuration. There is not that much differences between each algorithms in early but $ZPDES^H$ presents better results with time

Figures 3.4, 3.5 and 3.6 show the evolution of the average mastery level over all KCs for each set-up. The tables next to each couple of curves presents the ranking of each version of the algorithms and the average ranking of each algorithm at step 50 and 200 according to the curves comparison for each set-up $LM_{0,1,2}$, $LM_{0,3,4}$, and $LM_{2,3,4}$. The table 3.1 presents the statistical significance tests at step 50 and 200 for each set-up and what is the best methods if the results are statistically significant.

By comparing the different p-values, the differences between POMDP and ZPDES* are never significant, but it's not the case for $ZPDES^H$. For the models $LM_{0,1,2}$, at step 50, $ZPDES^H$ shows worse performances than the two others, this can be explained by the wider exploration of the graph. But it catches up rapidly and present the same results at step 200. So for models which are close to each other, the 3 algorithms present almost the same result.

LM	P/Z*: t 50	P/Z*: t 200	P/Z^H: t 50	P/Z^H: t 200	Z^*/Z^H: t 50	Z^*/Z^H: t 200
0,1,2	.075	.95	10^{-6} **(P)**	.82	.003 **(Z*)**	.87
0,3,4	.24	.90	.17	10^{-5} **(Z^H)**	.89	10^{-4} **(Z^H)**
2,3,4	.31	.30	.18	10^{-5} **(Z^H)**	.77	10^{-7} **(Z^H)**

Table 3.1: ANOVA p-values for each set-up to verify if the differences in the KC level distribution according to each algorithm are statistically significant with the best algorithms in parenthesis when it is significant. POMDP is noted P and ZPDES is noted Z

For the models $LM_{0,3,4}$, observations are different. At step 50, all the algorithms seem to have approximately the same performance, even if $ZPDES^H$ seems a bit behind but it's not significant (p-values at 0.17 and 0.89). But with time, it takes the lead and achieves the best performance at 200 steps. So when there are two models critically different from another, ZPDES^H presents the best results. For the last case, the population is constituted of students following $LM_{2,3,4}$ models, and the algorithms are configured for models $LM_{0,1}$. As for the previous case there is no differences at step 50 but ZPDES^H presents the best results at step 200 showing his better adaptation to populations that differs of his configured model.

ZPDES^H provides the best result because its exploration parameters were not optimized for any particular knowledge graph, giving it higher adaptability and less constrains in the exploration. For a particular type of student model it presents worse performance than POMDP or ZPDES*, but for a heterogeneous population, ZPDES^H, being more adaptable, has the best performance.

3.4 Discussion

In this study, the student models considered present knowledge components which can have constraints among each other, allowing to model some kind of pre-requisites. Under different student models, an optimal teaching sequence can be found using POMDP. Another alternative is the use of the ZPDES algorithm that is computationally more efficient but without optimality guarantees. The goal was to test how robust each of these methods is, in relation with ill-estimated parameters of the models, or even wrongly estimated relations between KCs. This corresponds to the more realistic case of heterogeneous classes of students.

For the trivial situation where the students are perfectly modeled with the student model, ZPDES can achieve the same performance as POMDP. For heterogeneous populations, again, ZPDES can achieve solutions similar to POMDP and can present even better performances. The best algorithm was the ZPDES version which uses parameters that are not optimized for a particular population. By having more flexibility in the exploration it becomes more robust to changes in the population.

In conclusion, when combined with an activity graph, the HMABITS framework can be a better choice in comparison with POMDP due to its computational efficiency and reliance on simpler student models.

Problems in theory are all alike;
every application is different.

Tor Lattimore and Csaba Szepesvari.

4
Teaching Scenarios

This chapter presents two different scenarios of application to implement the HMABITS framework. One was implemented in the Kidbreath project, carried out by Alexandra Delmas, which proposed a health educational ITS to teach ill children about asthma. One of their study study used the HMABITS framework to manage their activities. The second scenario developed during this PhD, as part of the Kidlearn project, was implemented in a serious game about money operations to train 7-8 year old children in mathematics.

4.1 The Kidbreath Project

In recent years, digital health is an expanding domain in both prevention, care and health education. Asthma is the first chronic disease among children and has severe medical outcomes (hospitalization, death). One of the key issues is the personalization of therapeutic education to improve disease self-management skills, which can have a strong impact on the disease evolution. In this context, the general purpose of the Kidbreath project aimed to develop and to assess a serious game for the education of children with asthma.

Figure 4.1: Kidbreath connexion Interface. http://www.kidbreath.fr/

During this project, a user-centered method of Participatory Design was formalized to develop a framework for therapeutic education-related serious games targeting children with chronic disease. A web platform, KidBreath (Fig. 4.1), was designed to propose an edutainment therapeutic program for asthmatic children. During the first study of the project, the device's efficiency was successfully tested by a group of control children (without asthma).

In a second study, KidBreath was used by children with asthma over a two-month period in a real-life setting. The main results supported pedagogical efficacy and efficiency of KidBreath for target audience at short term (1 month) and mid-term (2 month). However, the increase of asthma knowledge after using KidBreath failed to modify the children perceptions about disease self-management.

In a third study, the implementation of a system that automates the personalization of learning path was assessed in similar conditions than the previous one. The HMABITS framework – ZPDES – was used during this last study to personalize each student's learning path. The activity space used during this study and the corresponding HMABITS structure are summarized in figure 4.2. An activity has three different parameters. The primary parameter is a Theme : Biomedical (B), Symptoms (S), General Knowledge (C), Treatments (T). The second parameter is a difficulty level : (1, 2, 3). The last one is a Type (Quiz, QCM, Games). During this study, the pedagogical efficiency of the KidBreath application was also demonstrated. The amount of contents and the time spent in the personalized condition with ZPDES were diminished compared to the non-personalized condition. The results of these studies have been published in Frontiers (Delmas et al., 2018). This study is not presented in this report because it is not a central topic of this PhD.

4.2 *Kidlearn scenario*

Kidlearn is the core project of this PhD thesis. A teaching scenario was developed and tested, with the results presented in Chap. 5 and 6. This scenario uses money to teach children how to decompose numbers, typically targeting 7-8 year old students. This scenario was chosen for its simplicity, while remaining rich enough to offer different learning/teaching trajectories to impact individual students differently.

Figure 4.3: Kidlearn on tablet

Furthermore, combining number and money manipulation is a way to instantiate abstract knowledge into a practical, useful real-world scenario. This scenario is instantiated in a browser environment. The web application development is an important part of the thesis and was built using the Django [1] framework. The application proposes exercises to students in the form of money games (see Figure 4.4). For each exercise type, one object is presented with a given tagged price, and the learner has to choose which combination of bank notes, coins or abstract tokens need to be taken from the wallet to buy the object, with various constraints depending on the exercise parameters.

[1] "Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source." https://www.djangoproject.com/

Seven Knowledge Components (*KC*) are targeted in this senario: a) **KnowMoney**: Global skill characterizing the capability to handle money to buy objects in an autonomous manner; b) **SumInteger**: Capability to add integers; c) **SubInteger**: Capability to subtract integers; d) **DecomposeInteger**: Capability to decompose integers into groups of ten and units; e) **SumCents:** Capability to add decimal numbers (cents); f) **SubCents:** Capability to subtract decimal numbers (cents); g) **DecomposeCents**: Capability to decompose decimal numbers (cents).

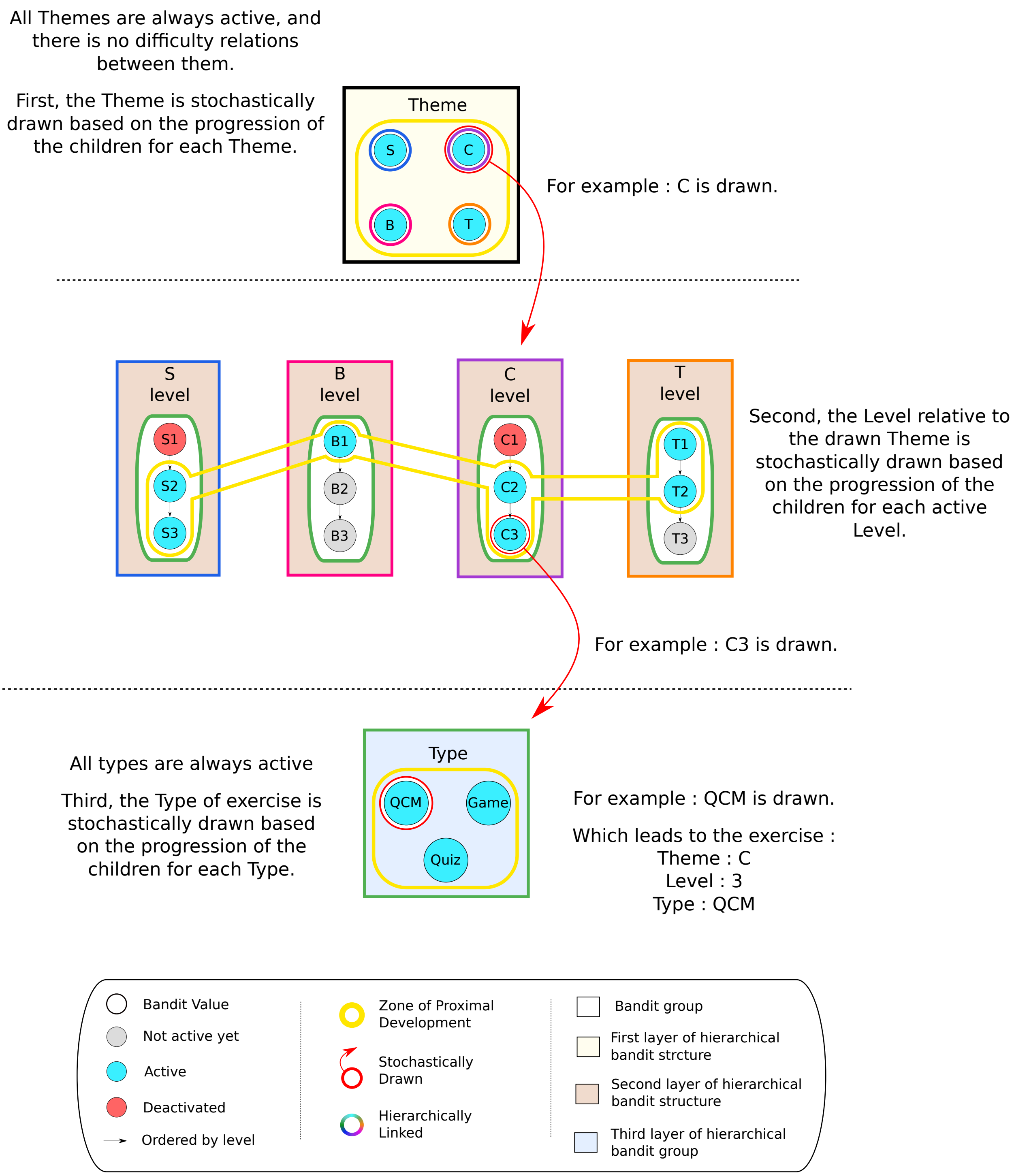


Figure 4.2: A representation of the HMABITS framework used for a pedagogical scenario to teach kids about asthma. A virtual example of activity is sampled at a particular state of the HMABITS restulting from activities already made by an hypothetical learner.

The various activities are parameterized in order to allow students to acquire a greater flexibility in using money. The activity space used during the different studies and the corresponding HMABITS structure are summarized in figure 4.5. There are 5 parameters organized hierarchically. First, the **Exercise Type** is chosen: the student can be the costumer or the merchant and buy or give change with one or two objects, which leads to four different possibilities. For each type of exercise, the difficulty is chosen based on the difficulty **Level** of decomposing a number. A number can be easy to decompose if there is a direct relation with a real bill/coin $a = (1, 2, 5)$ and hard to decompose if it requires more than one item $b = (3, 4, 6, 7, 8, 9)$. The exercises will be generated by choosing prices with these properties and picking an object that is priced realistically. A dimension related to the difficulty is the presence of **Carried Numbers** in the operation, when there are two objects. It is managed by a different parameter because it is not related to a particular exercise type. **Price Presentation** varies due to the different practices in stores and countries, which do not always follow the standardized rule. Finally, different **Money Shapes** are used: Real Euro or poker tokens, which can reduce the visual ambiguity. Another parameter that controls the **Price Modality** is added: in written form and/or using a speech synthesizer, but technical problems appeared with sound control in the tablet web browser: the sound was delayed or was simply not produced, which required to remove this parameter.

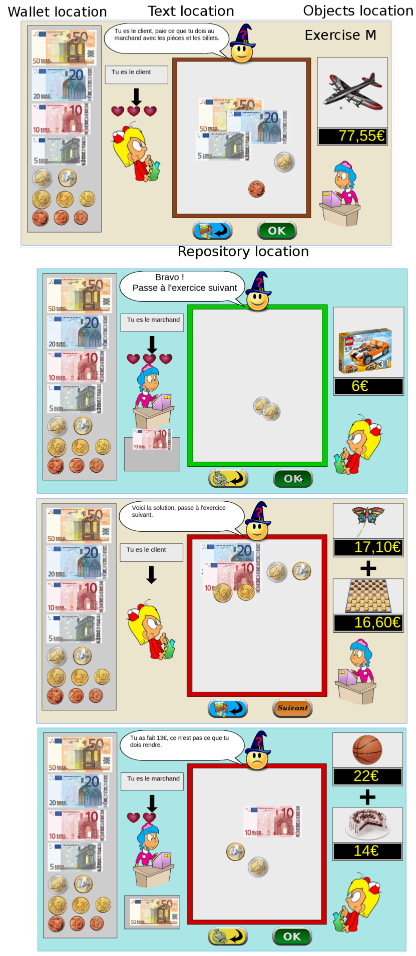


Figure 4.4: Four principal regions are defined in the graphic interface. The first is the wallet location, where users can pick and drag the money items and drop them on the repository location to compose the correct price. The object and the price are present in the object location. Four different types of exercises exist: M : customer/one object, R : merchant/one object, MM : customer/two objects, RM : merchant/two objects. (Full page picture in appendix C)

Graphical interfaces in ITS can have unwanted side effects. For this reason, the interface was entirely designed with the help of a didactician, with several specific design choices motivated by instructional design principles, and motivational and attentional requirements. For example, the interface, shown in Figure 4.4, is such that: a) display is as clear and simple as possible; b) there is no chronometer, so that students are not put under time pressure; c) coins and banknotes have realistic visual appearance, and their relative sizes are respected; d) costumer and merchant are represented to indicate clearly the role of the student; e) text quantity is kept to minimum.

When the student begins the activity, one or two objects with their corresponding prices are shown. To complete the exercise, the student has to drag and drop the money that she/he wants to use from the wallet location to the repository location. It is possible to request extra cues, by clicking on the face. They have to click on the "OK" button to submit the answer leading to a feedback. If the answer is correct, the feedback is "Congratulation you can move on to the next exercise". The experience must provide the most pedagogical gains and so, the student has 3 opportunities to solve the exercise and extra cues are provided each time the student makes a try. If after 3 trials the answer is still wrong, then a feedback with the correct solution is given and the system moves on to the next exercise.

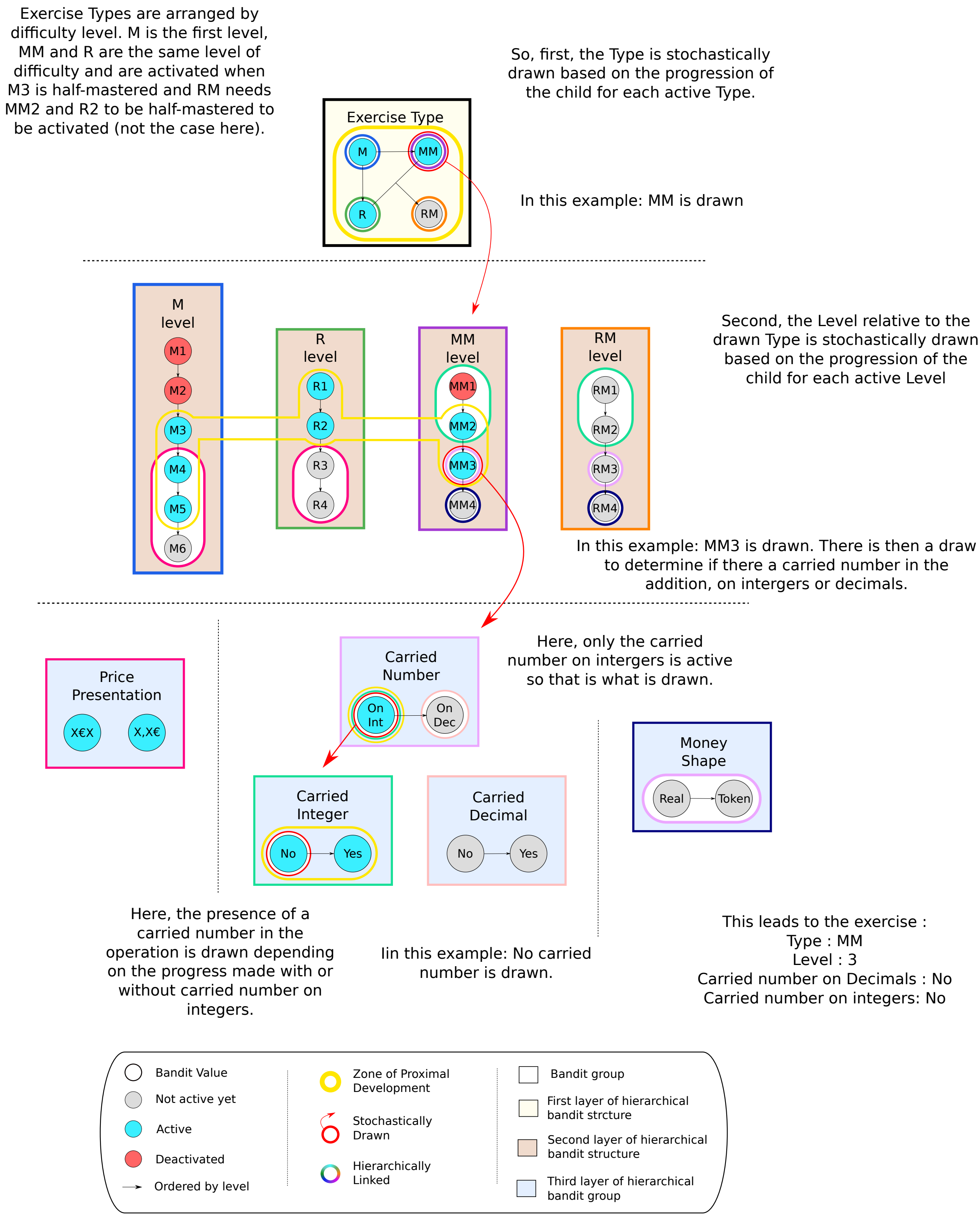


Figure 4.5: A representation of the HMABITS framework used for a pedagogical scenario to teach kid about mathematics by making them manipulate money. A virtual example of activity is sampled at a particular state of the HMABITS resulting from activities already made by an hypothetical student.

Our children need to be treated as human beings - exquisite, complex and elegant in their diversity.

Lloyd Dennis

5

Kidlearn impact on learning

Contents

This chapter presents a study published in the Journal of Educational Data Mining in 2015 (Clément, Roy, Oudeyer, et al., 2015). Two experiments are showed to evaluate HMABITS framework impact on student learning. In a first experiment, a systematic statistical studies of the impact of HMABITS framework was conducted over a population of simulated students. Then a real-world experiment is presented where the framework is implemented as a tablet application used for teaching number decomposition while using money. The experiment involves 400 children (7-8 year old) from 11 schools. The effectiveness of the framework is measured by the comparison of their output to a teaching sequence hand-crafted and validated by an expert implemented as a ExpSeq algorithm.

5.1 Algorithmic considerations

Three algorithms are studied, RiARiT, ZPDES and ExpSeq. The version of RiARiT is the one presented in Sec. 2.3, it came after the evolution of a primary implementation presented in a previous published study (Clément, Roy, Oudeyer, et al., 2014b). The information provided by the pedagogical expert to configure RiARiT are presented in appendix A.1.

A VERSION OF ZPDES previous to the one presented in Sec. 2.4 was used. The difference lies in the ZPD management. In this case, the evolution of the ZPD is different. For activities at the same difficulty level, a free exploration is allowed as in the current version. But for activity parameters that have a clear progression in values difficult, such as $v_1 < v_2 < \ldots$ for the parameter i, there are two ways to activate the upper values.

First: if there is no parameters group hierarchically linked below this parameter. Then, the number of active values n_{ZPD} is capped at I. This window of values represent the ZPD. When the success rate is higher than a pre-defined threshold : $\sum_{t=1}^{T} \frac{C_t(j)}{T} > \omega$, the parameter value $j + n_{ZPD}$ are activated with the following rule $w_i(j + n_{ZPD}) = w_i(j + n_{ZPD} - 1)$. If n_{ZPD} is equal to I then the activation take place only when the bandit level of the parameter value j is below the level of the more complex parameter value : $w_i(j) < \theta w_i(j+1)$. Then v_{j+I} is activated as described before and $v_(j)$ is deactivated ($w_i(j) = 0$).

Second: if there are hierarchical groups below the parameter values. When the average of the success rate over all the values of the parameter l_i hierarchically below the parameters value j is higher than a pre-defined threshold $\sum_{l=1}^{n_v} \left(\frac{\sum_{t=1}^{T} \frac{C_t(l)}{T}}{n_v} \right) > \zeta$, the parameter value $j + 1$ is activated with the following rule $w_i(j+1) = \frac{w_i(j)}{2}$ (if the student does not make an exercise with a value, his success rate for this value is considered null).

FOR EXPSEQ, the difference between the current version (sec. 2.5) and the one used in this study lies in the requirement mechanism. Indeed in this study, no requirement was introduced to allow access to particular activities, only the group structure, and the advancement depending on the success rate for each exercise group, was implemented. ExpSeq is used as a baseline to evaluate the algorithms, as described in Sec. 2.5. This baseline sequence grows in terms of complexity of the problem and simultaneously in terms of the difficulty of interaction. The prices produced, as seen before, become more complex in terms of the difficulty of decomposing a number and not on its absolute value. That is, the prices presented can be directly matched with the corresponding items, while the others require the composition of several items. Also the introduction of cents increases the complexity in several dimensions, requiring understanding of the concept of decimal and also on how to represent them. The introduction of tokens allows students to train with decimal numbers directly. Using cents is easier with real money as the items for integers (bills) and cents (coins) are different. The full details of this sequence are presented in Section A.2.

Also, a random policy is not used as a baseline because this leads to a lot of errors, and changes on types of exercises which is disturbing for many students and not acceptable for the teachers.

5.2 Simulation

5.2.1 Cognitive Student Model

A set of simulations is first presented to systematically test different properties of the algorithms. Two different virtual populations of students are defined to see how well the algorithm is able to select exercises adequate for each particular student and the impact of different properties of students. A population "Q" is considered where all the students are able to use all the activities to learn, even if at different learning rates and with different maximum comprehension levels. Another population "P" aims at representing even more heterogeneous populations where each student might have a limitation for the comprehension of a particular type of activity. A concrete example is the case of a student that is not yet able to read will not be able to use exercises in written form to learn about mathematics, but if the exercise is presented in the spoken form it might be used for learning. Another example would be a student with hearing problems not able to solve an exercise that is presented with audio.

A personalization is expected to provide small gains with population "Q" because all students are able to use all exercises to progress. On the other hand, the population "P" will require that the algorithm finds a specific teaching sequence for each particular student. Both models follow a standard Item Response Theory (Hambleton, 1991), where the probability of solving an exercise is given by:

$$p(success) = \frac{\gamma(a)}{1 + e^{-(\beta(c^{Q} - c(a) + \alpha))}} \tag{5.1}$$

where β and α are constants that allow to change the shape of the probability distribution and that can be chosen to provide different learning rates of a population. For model "Q", $\gamma(a) = 1$ means all activities can be solved. For the model "P" some of the activities have $0 \leq \gamma(a) << 1$. This implies that for "P", some activities cannot be solved regardless of the competence level. The students have a probability of learning based on the difference between their levels and the level of the exercise.

5.2.2 Results

The results presented here show how fast and efficiently the algorithms estimate and propose exercises at the correct level of the students. Each experiment considers a population of 1000 students generated using the previous method and lets each student solve 100 exercises. For all populations the different initial, maximum final level of understanding of each KC is sampled from a truncated Gaussian distribution. For the population "P" the values of parameter's understanding are sampled from four different distributions that include different levels of understanding (γ) for each parameter.

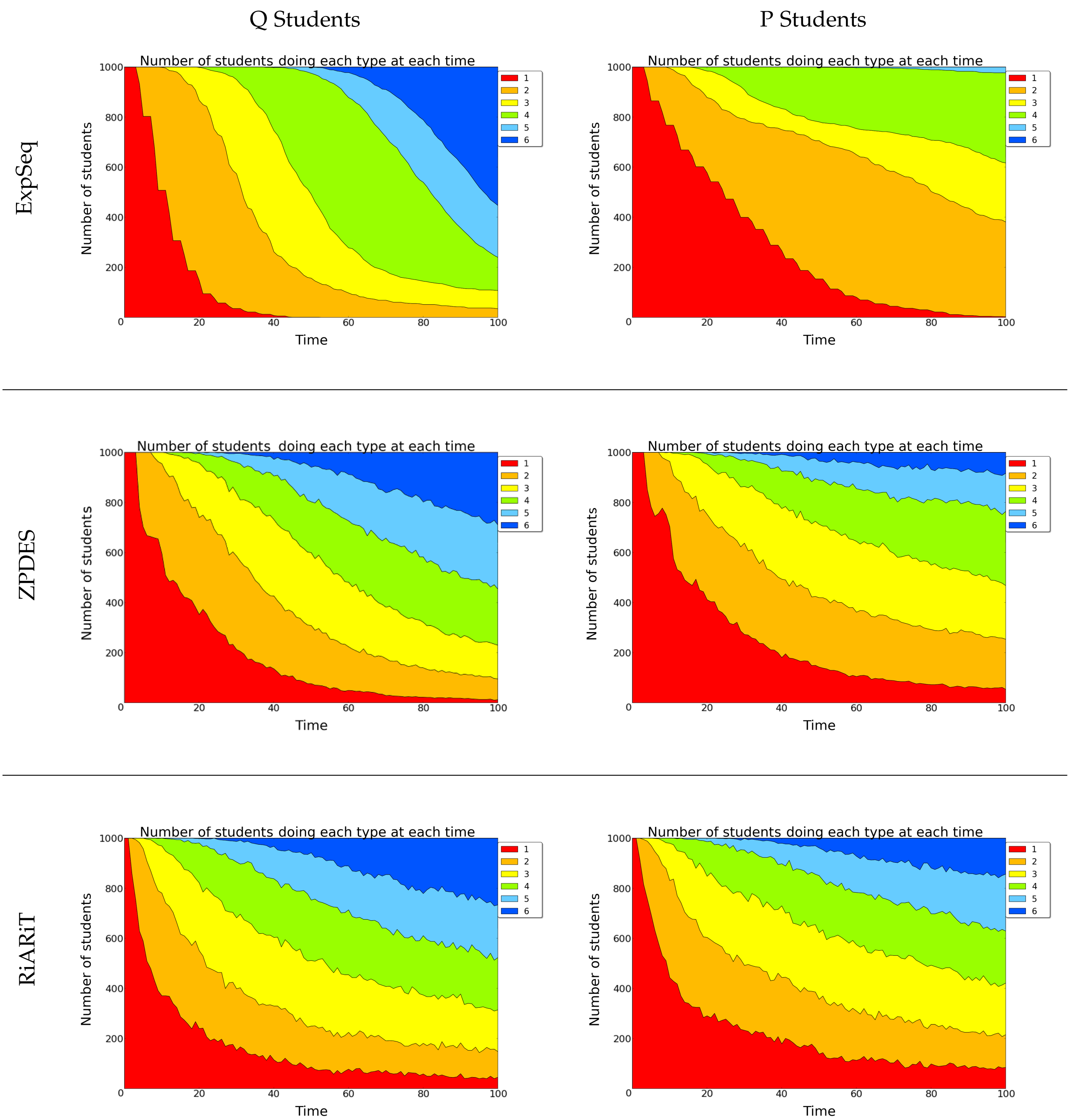


Figure 5.1: For each time instant, the curves show the total number of students being proposed each level of Exercise Difficulty. ExpSeq is not able to propose more difficult exercises as early. RiARiT and ZPDES can thus propose more difficult exercises sooner and keep proposing easier exercises longer. This shows the personalization properties of the algorithm.

Figure 5.1 shows the number of students that are being proposed each type of exercise (only showing the parameter Difficulty for exercise Type M), independently if they succeed or fail the exercise. The actual student's levels are shown in Figure 5.2. In general, RiARiT and ZPDES start proposing more difficult exercises earlier while at the same time keep proposing the basic exercises much longer. This shows a clear adaptation to the actual level of the students.

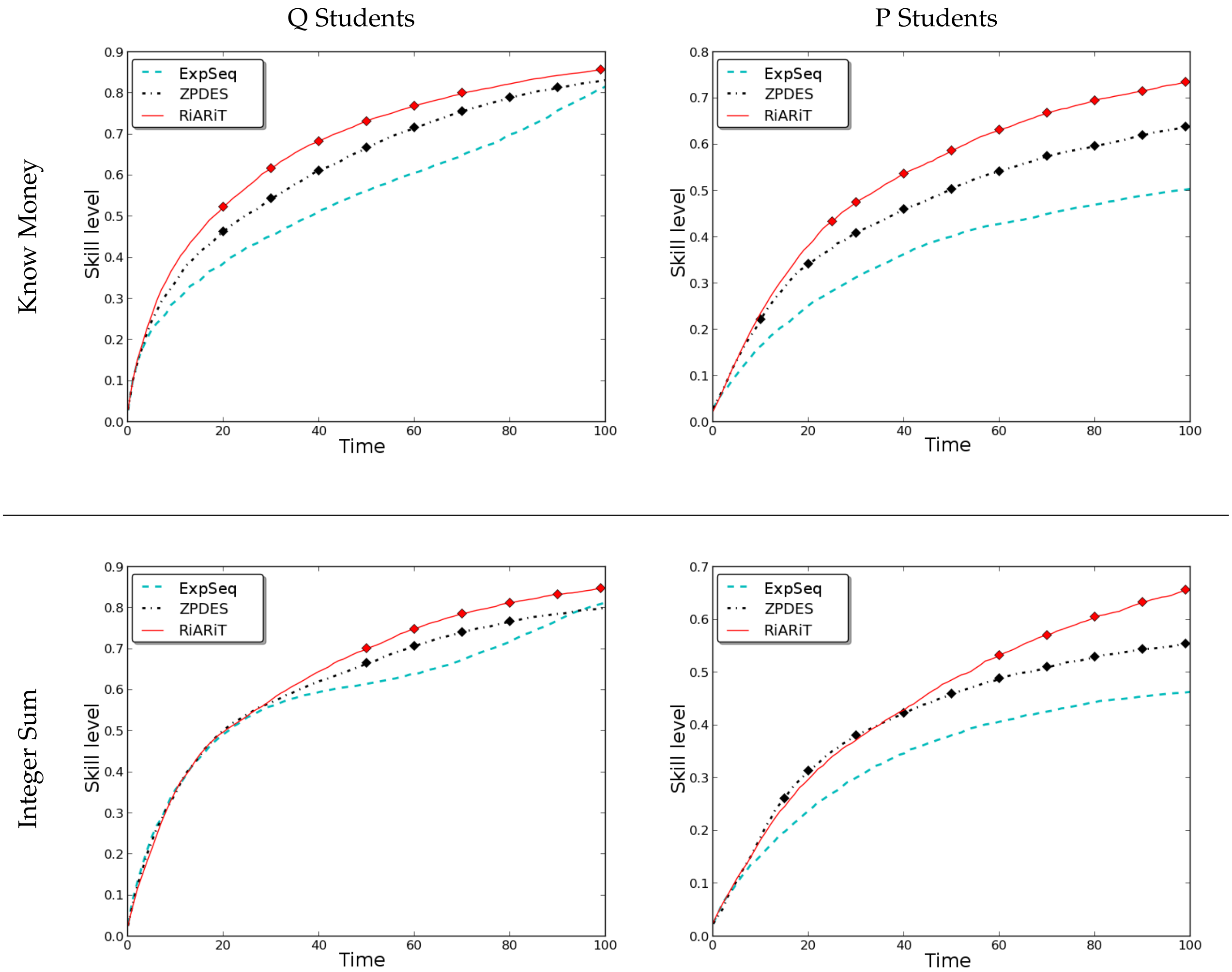


Figure 5.2: The evolution of the skill's levels of two KC with time for population "Q" and "P". Markers on the curve mean that the difference is statistical significant (red : RiARiT/ZPDES, black : ZPDES/ExpSeq). Both algorithms are able to improve upon the Expert Sequence.

Figure 5.2 shows the skill's levels evolution during 100 steps. For Q students, learning with RiARiT and ZPDES is faster than with the expert sequence. For P students, as they might not understand particular activities, they block on certain stages due to the lack of adaptability of the expert sequence. On the other hand, ZDPES by estimating learning progress, and RiARiT, by considering the estimated level on all KC and parameter's impact, are able to propose better adapted exercises.

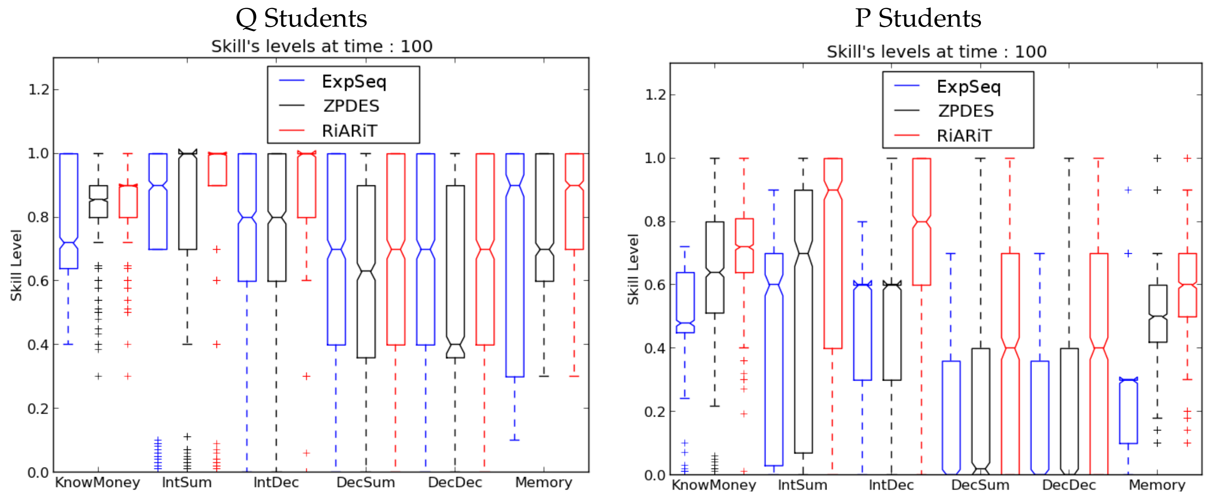


Figure 5.3: Distribution of the acquired competence levels after 100 steps represented as a boxplot indicating median and the 4 quartiles. A statistical significant difference exists if the notches do not overlap. Overall, the automatic methods allowed a better understanding of all KC with a stronger gain in the case of P students.

Figure 5.3 shows the competence level of the students after 100 steps, represented as a standard boxplot. For "Q" and "P" students, differences are statistically significative for almost all KCs. RiARiT gives better results than ExpSeq due to its greater adaptation to the students' levels. No difference can be distinguish between ExpSeq and ZPDES. In the case of students of type "P", RiARiT and ZPDES are both better than ExpSeq. This is because when the students are not able to understand a specific activity, an hand-designed sequence cannot adapt to all possible variants of the students' learning.

The errors made by students during learning can also be analysed. If the exercises are too difficult to solve there will be many errors and this can be a source of frustration. Figure 5.4 shows that for both types of students, at the beginning, the number of errors is equal among methods but with time, expert sequence gives rise to more errors than when using RiARiT or ZPDES, in particular for "P" students. And for "P" simulation, students have less errors with RiARiT, showing it adapt better.

5.3 *User Studies*

As the final goal of an ITS is to provide a more efficient teaching experience to students, a user study was performed and aimed at validating the software infrastructure, the interface and the algorithms themselves. The goal is to evaluate principally the learning improvement, the personalization, and the impact of the use of a model. The set-up was deployed in 11 schools in Bordeaux school district with 15 tablets and a portable server to avoid school material constraint. A total of 400 students between 7 and 8 years old participated in this experiment. the population is divided into 4 experimental condition groups, a control group where student do not use the software and 3 groups where exercises are proposed using the Expert Sequence of Activity, ZPDES or RiARiT. For this experiment, the class are divided into groups of 15 or less students.

Figure 5.5: A photo taken during a study session in a Bordeaux classroom.

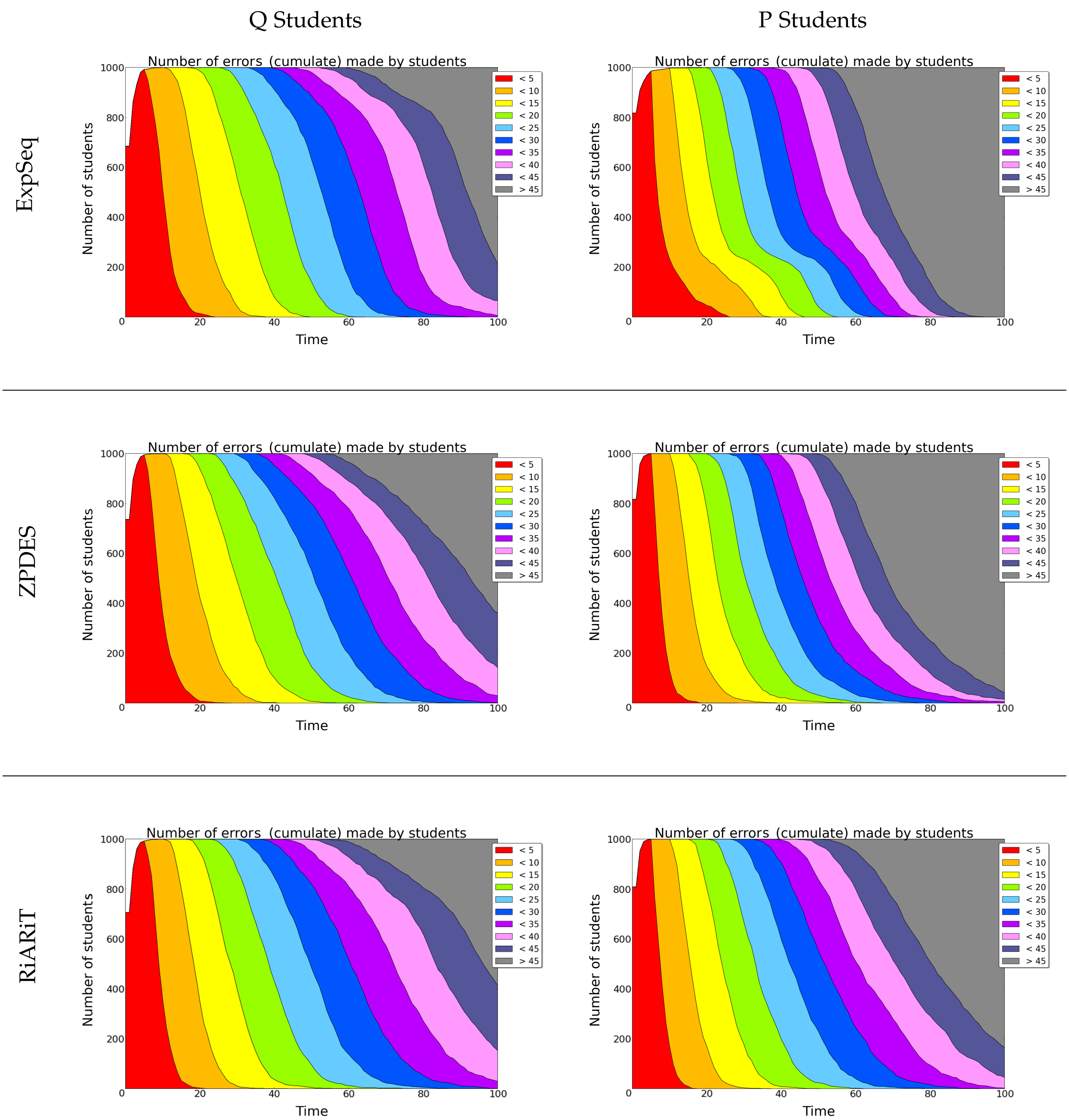


Figure 5.4: This figure shows the number of errors made by the students. For each time instant, and for each number of cumulative errors (indicated in the colors), the curves shows the total number of students that made that number of errors.

For each group, the students pass a paper pre-test just before using the ITS with a time limit of 20 minutes. The students work then on tablets during 40 minutes due to schools constraints and personal time restrictions. This led to a different number of exercises completed by each student. This makes the comparison of results between the different algorithms harder but, on the other hand, it is a real constraint when using class time. In the following results, the axis "Time" represents the succession of exercises. For example, "Time 1" is the first exercise for all students, but if at time 30, some students have already finished, they don't do the exercises at time 30, so with time the cumulative number of students decreases.

Students make a post-test, given to the teacher, a few days after. This post-test is used to measure student learning. The control group passed the pre- and post-test at similar time frame but without using the game. During the working sessions and tests, Students were allowed and encouraged to use a draft.

5.3.1 Maximum level achieved

Figure 5.6 shows the percentage of students who succeeded each level and type of exercise. The graphic is not cumulative, so students are taken into account only for the maximum level they reach for each type of exercise. Globally, there is much more students who succeed higher levels of R, MM and RM exercises with the RiARiT and ZPDES algorithms than with the Expert Sequence. To know if

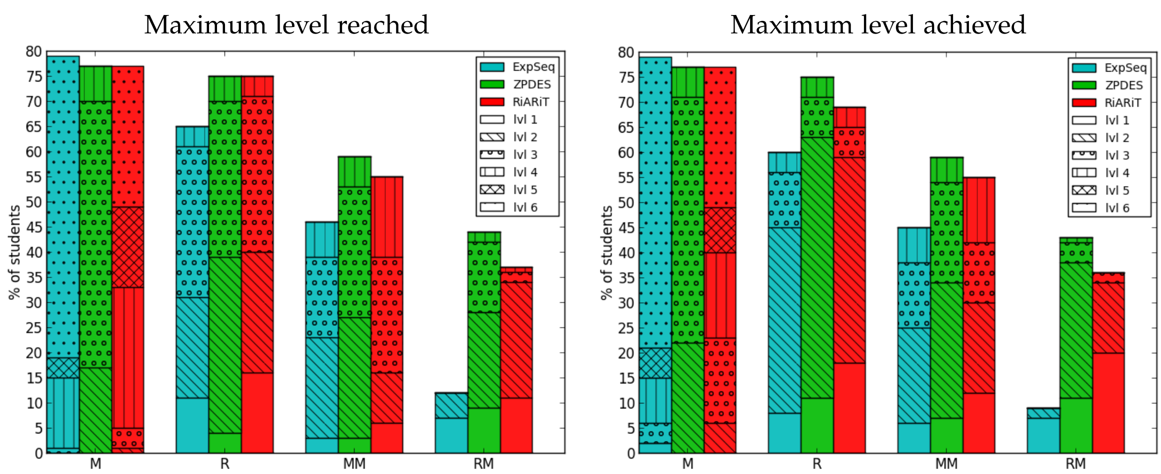


Figure 5.6: The figures show the proportion of highest level reached (left) or achieved (right). A level can be reached, yet not achieved. ZPDES and RIARIT allowed students to reached and succeed the most challenging types of exercises (MM and RM). By combining this with the information in Fig. 5.1, students seem to reach their level of competence earlier when using the automatic algorithms.

the type of sequence management have a significant impact on the maximum level succeeded by students, a χ^2 test was done to test the dependence and an *ANOVA* to test if the differences are significant. Tests results are summarized in Table 5.1. The first part shows the student medium level for each group, students have succeeded highest level exercises with ZPDES and RiARiT than with the Expert sequence except for M type. This is not surprising as the M exercises are the first ones to be proposed and the Expert Sequence spent more time there. The second part of the table shows the p-value of χ^2 test for independence. For the majority of exercises type (M, R, RM), the p-value is lower than 5%, the null hypothesis of independence can be rejected. Then to improve the analysis, an *ANOVA* was made to ensure differences between groups to be significant. In majority, the *ANOVA* allows to say that the differences are significant.

So even if there are much more students who reach and succeed the highest exercise of M type with the Expert Sequence (75% versus 0% for ZPDES and 35% for RiARiT), there is much more students who reach and achieve the other types. ZPDES and RiARiT proposed exercises of other types that, in the end, results in a better acquisition of the KCs. For R type exercise : 95% for ZPDES and 90% for RiARiT of students succeed at least one exercise versus 75% for Expert Seq. And the difference increase with MM and RM exercises.

level average

	M	R	MM	RM
ExpSeq	**5.42**	1.66	1.41	0.14
RiARiT	4.47	1.74	1.77	0.70
ZPDES	2.79	**2.01**	**1.83**	**1.05**

test χ^2 : p-values

	M	R	MM	RM	All type
ExpSeq/RiARiT	$\ll$ **.001**	**.04**	.17	$\ll$ **.001**	$\ll$ **.001**
RiARiT/ZPDES	$\ll$ **.001**	.14	.059	$\ll$ **.001**	$\ll$ **.001**
ZPDES/ExpSeq	$\ll$ **.001**	$\ll$ **.001**	.085	$\ll$ **.001**	$\ll$ **.001**

ANOVA : p-values

	M	R	MM	RM
ExpSeq/RiARiT	$\ll$ **.001**	.88	.11	$\ll$ **.001**
RiARiT/ZPDES	$\ll$ **.001**	**.04**	.70	$\ll$ **.001**
ZPDES/ExpSeq	$\ll$ **.001**	.07	**.04**	$\ll$ **.001**

Table 5.1: Statistical test on the results of the user study. The top table shows the average difficulty level reached for each type of exercise. Then two statistical tests are presented to verify if the difference in the means and in the distributions are statistically significant. From the results the conclusion is for most cases ZPDES is better than the Expert Sequence.

5.3.2 *Personalized Learning Sequences*

Do the different algorithms provide qualitatively different learning sequences or do they only adapt the speed of progression ? Figure 5.7 shows two different things. On the left, the figure shows the evolution of the estimation of the students' competence level, corresponding to the exercise that is being proposed to the learners (only showing the parameter Exercise type and level). On the right side, circular design made using Circos (Krzywinski et al., 2009) are displayed. On this figure, the transitions between exercises made by students along time are represented by the colored curved lines (blue for ExpSeq, green for ZPDES and red for RiARiT). A transition starts on an exercise (start), situated on the yellow part of an exercise, and finish on another (target), situated on the brown part of an exercise and represented by an arrow. The line thickness represent the number of students who did that transition. The time is represented by the color shade, light colors correspond to early exercises, darker colors to later ones.

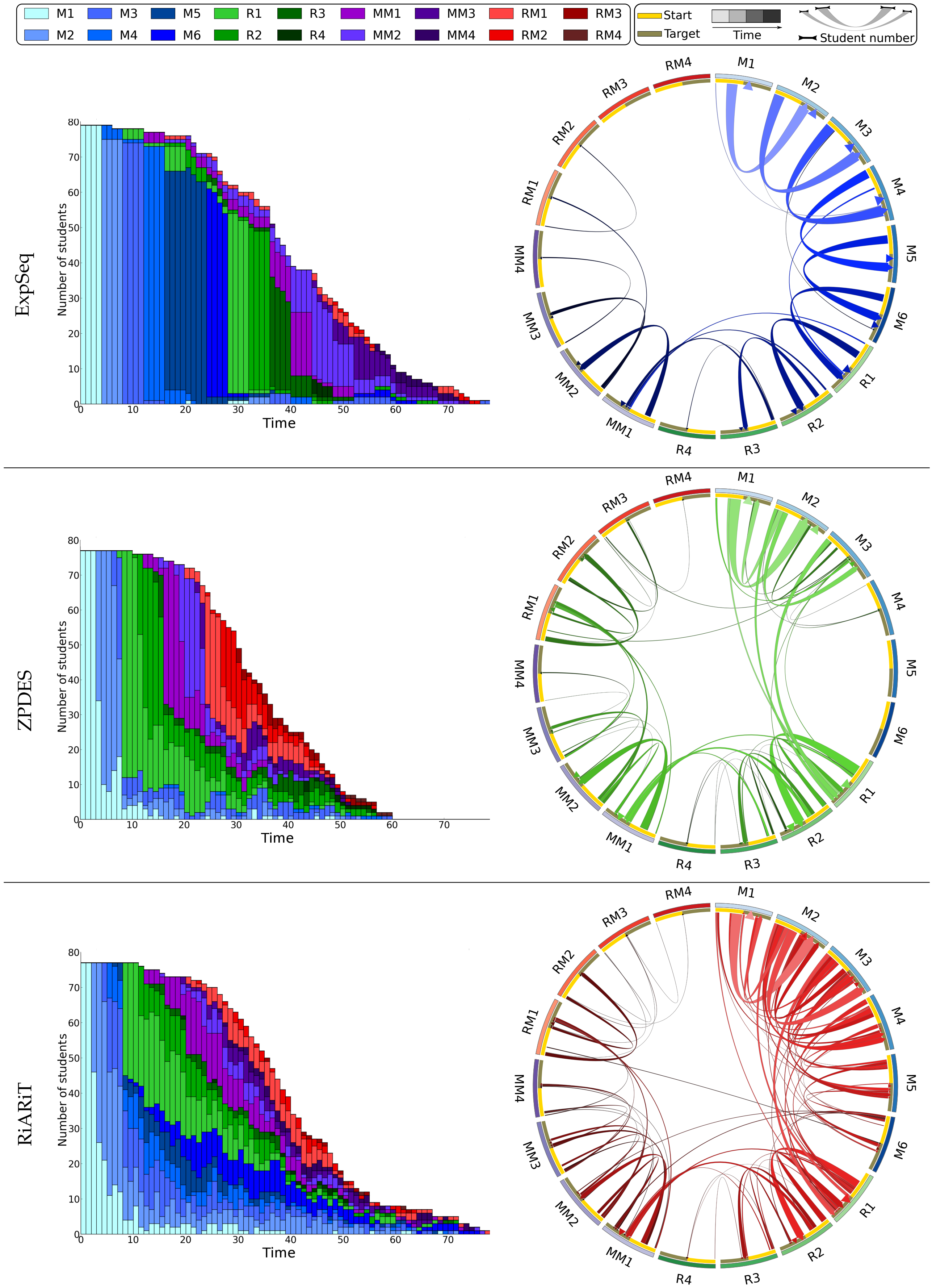


Figure 5.7: For each type and level of exercises : (left) the histograms show the number of students doing it at each time and (right) Circos drawing show the number of students did transitions between activities where students transitioning from one activity (start) to another (target). The line thickness is proportional to the number of students. Light intensity corresponds to time. A stronger variety of paths is proposed by HMABITS algorithms resulting from the online personalization. Only main transitions are displayed (frequency $\geqslant 5$).

This figure allows to see the main paths followed by the students for each algorithm. ZPDES even ignored the more difficult exercises of Type M, as it has found that the other types of exercises were providing greater learning progress. This can be seen as a good sign of adaptability from ZPDES but it also reveals a weakness in the Activity Space definition. Indeed, student could work on R, MM and RM exercises with decimals while they did not train with decimals for M activities. This led to a difficulty spike for some students. This was also observed with the Expert Sequence where student could also work on R, MM and RM activities with decimal while they do not mastered decimals with M. In general, RiARiT and ZPDES propose a large diversity of type and difficulty of exercises earlier that the Expert Sequence. The same phenomena is visible in figure 5.6, where there are more students who reach MM exercises and RM exercises with the algorithms than with the Expert Sequence. The circos figures show that RiARiT and ZPDES proposed more different activities and paths, which reveals an adaptive behavior, where the Expert Sequence proposes almost always the same path.

5.3.3 *Differences in pre- and post- tests*

The pre- and post- tests enable to test student knowledge on some KC, buying one object (M) or two (MM) and exercises of giving change (R). To give change with two objects (RM) is not tested because it is not part of the official program for that grade. Figure 5.8 shows the evolution of results between pre and post-tests for the control group which has not used the ITS and the normal group.

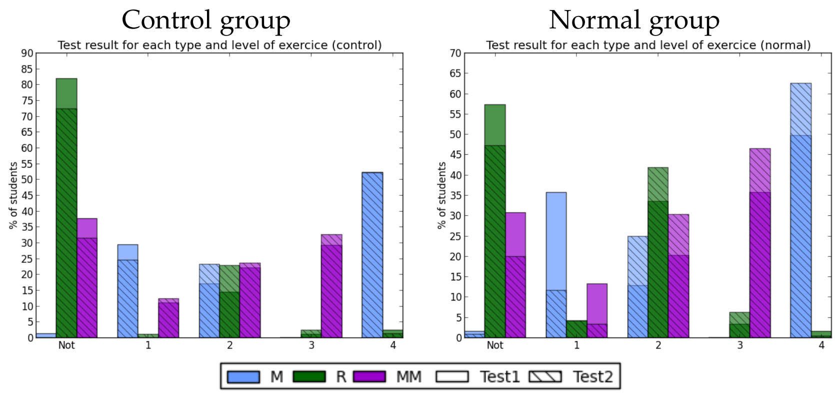


Figure 5.8: For each type and level of exercises, the histograms show the percentage of students who have achieved it for the first and the second test. For each bar the shaded area corresponds to Test 2. If the shaded goes higher is means that an higher number of students answered correctly in Test 2. Control group is on the left and the normal group is on the right.

The normal group improved their results between the pre-test and the post-test, about 65% of students who were at level 1 for M type are moved to a higher level. Likewise for R and MM types, there are respectively 20% and 40% of students who were at level 0 and 1 who have increased their level. For the control group, their learning is much lower than those who worked on the application. Only 15% of the students who were at level 0 or 1 for all type of exercises are moved to a higher level. An *ANOVA* is made to test the significance of theses differences. The null hypothesis is the control and normal group learned the same amount. The null hypothesis can be rejected ($p - value < 5\%$), the students learned more using the application.

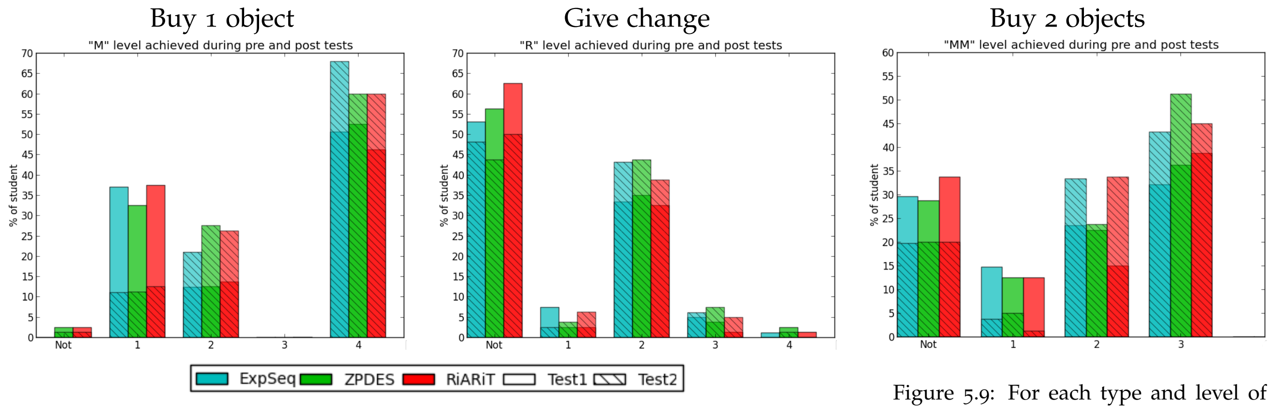


Figure 5.9: For each type and level of exercices, the histograms show the percentage of students who have achieved it for the first and the second test.

Also, as can be seen in figure. 5.9, the analysis of results of written tests differentiated according to customized techniques of learning activities sequences shows that the learning differences among students for each techniques are not significant (same hypothesis as before and a p-value of $ANOVA > 5\%$). These differences do not allow us to say whether a technique is better than another for the limited amount of KC evaluated in the pre- and post- test. However, significant differences exist in the results obtained in the application as presented above.

5.3.4 *Observation in classroom*

No validated motivation or psychological metric was used during this experiment but a questionnaire was made with 12 questions to try to evaluate algorithm impact on student psychological state.

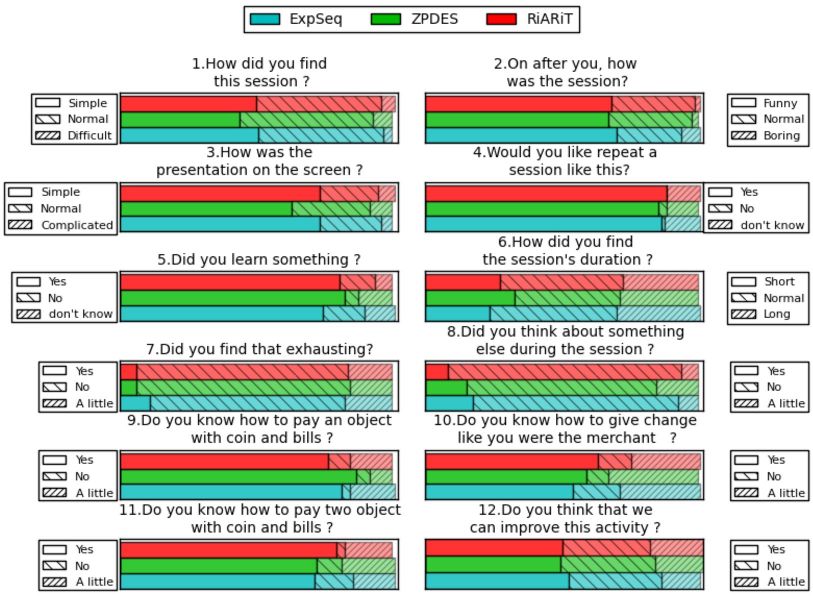


Figure 5.10: Questionnaire results showing the percentage of student answer trained with each algorithm.

Student answers are shown in figure 5.10. Some tendencies seem to emerge through the result. There are more student who think they learned with the activity with HMABITS framework (question 5). There are less student who were distracted with HMABITS algorithms (question 8). More students found the activity exhausting with the Expert Sequences (question 7) and more students think to know how to return money and pay 2 objects with the algorithms (question 10 - 11). These tendencies seem to show that students have a better experience and seem to think they learned better with the algorithm than with the Expert Sequence. The differences are not much significant but it can be correlated with some observations made during sessions.

Students working with the Expert sequence were observed to be more frustrated during the sessions because they could be blocked on activities they do not understand during four steps. Then they had to make three errors to pass to the next exercise. Indeed, the design choice to deny student opportunity to pass exercises was made. The goal was to make them work on each proposed exercise. But this led to other escape strategies – doing three errors on purpose to pass the exercise. This shows some weakness in the activity space as discussed in section 5.3.2. This behaviour was less happening with HMABITS framework which can alternate difficulty levels between two steps.

5.4 Discussion and limits

In this study, HMABITS framework was tested in an intelligent tutoring systems. Due to their efficiency, these algorithms allow a true personalized learning experience by relying on more or less knowledge of the field and by doing an adaptation in real time based on student results. ZPDES and RiARiT were studied. ZPDES relies only on the measure of success and failure on exercises and on a coarse predefined exploration graph to provide a personalized teaching sequence. RiARiT is able to exploit more information about the domain to estimate the level of the students and to personalize the teaching sequence. These two algorithms were compared to an Expert Sequence. One perhaps not surprising fact is that in simulation RiARiT, with its extra information provided better results, while in the user studies ZPDES provided better results.

The goal is not so much to provide better teaching sequences than expert teachers but, instead, provide a tool that can deliver exercises to the students at their competence level. Nevertheless, the results show that HMABITS algorithms can achieve an efficiency of learning comparable to a sequence provided by an expert teacher, even without using much information about the students, and without much information to be provided by the teacher for ZPDES.

The HMABITS framework was showed to be able to achieve comparable results for homogeneous populations of students, but showed a great gain in learning for populations of students with larger variety and stronger difficulties.

Even in cases where there is no gain in learning speed, a formulation of the problem based on the KC is already useful as it identifies more clearly the problems of each particular student, as was observed in the results. One problem is the amount of information the pedagogical expert has to provide to the system. The tables in appendix A.1 needed a great amount of work which, in practice, may be impractical. This led to put the study of RiARiT aside. Looking for ways to reduce this amount of information could be interesting to study.

The results from the user studies show a significant increase in progression for several activities, and a much better personalization teaching sequence. The algorithm ZPDES is the most promising for a real use as it requires very little information and much less parameters. But it can also present some weaknesses as discussed in Sec. 5.3.2 and 5.3.4. It did not explore M activity with decimals, judging this activity was not relevant because over M activity with integer, student had close to 100% success leading to a very low quality of the M type. This behaviour was taken into account and the ZPD updating and exploring mechanism was modified.

No significant learning difference was measured between each algorithms from the pre/post-test comparison. This can be due to different things. There may be no significant learning advantage between each methods but it can also come from the set-up. Indeed, the tests made was short and did not cover all the different possible level of competences. This was to avoid cognitive surcharge to the student, when doing the pre-test before training in the game. Student may also have difficulties to transfer skills they learnt in video game situation to a paper test. Another possibility is that students only train for 40 minutes which may be not enough time to allow us to see significant learning differences. These considerations was taken into account, and the set-up was adapted in the last study presented in chapter 6.

A more practical limit was the usage of paper test on a study of this scale. Indeed, each of the 400 students had two paper test and one paper questionnaire. The tests correction and data collection of these measures was very redundant, time consuming and obviously led to errors in the dataset that could not be spotted. A way to counter this practical limit would be to digitalize these questionnaires and tests. Even if students are more accustomed to paper and can use it as a draft, providing a lot of information about what students did to answer test questions and to more accurately assess their skills, it may not be worth it.

Another interesting point would have been to evaluate the students motivation depending on the experimental condition. Indeed, one of the main assumption is that the methods used to build the HMABITS framework, the learning progress hypothesis, the theory of Flow and the concept of ZPD, increase the engagement of the students and keep them motivated. This consideration will also be incorporated in the study present in chapter 6.

The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.

Sir William Bragg

6

Kidlearn impact on motivation alongside learning

Contents

This chapter presents the last experimental study which has not been published yet. It continues the work presented in chapter 5. As discussed then, this previous experimentation did not allow us to measure accurately student motivation and to observe significant differences in learning between the different experimental conditions. Several new components are also introduced in this study. One of the goals is to evaluate the relevance of using learning progress in ZPDES. Then, a variant of ZPDES named ZPDRD (ZPD and Random Draw) is introduced, where the parameter values available in the ZPD are chosen randomly and not according to their quality. Following intrinsic motivation theory, another goal is to integrate choice to the framework and study its impact on motivation and learning.

An experiment was designed to compare 5 experimental conditions. ExpSeq is still used as a baseline and is implemented as an ExpSeq algorithm. The others are ZPDES, ZPDRD and two conditions with choice named ZCO (object choice) ad ZCA (activity choice). Another goal of this study is to evaluate the impact of these conditions on students motivation and learning. This experiment involves 581 children (7-8 year old) in 24 classes of the Bordeaux school district during 4 month (Mars-June 2018). Algorithmic considerations are first presented, then the experimental protocol is described and finally, preliminary results of the study are displayed.

6.1 *Algorithmic and experimental considerations*

THE VERSIONS OF ZPDES AND EXPSEQ used in this study are the current ones described in chapter 2. The Activity Space used is the same as in user experiment in Chap. 5 with the addition of requirements to access some activities. As discussed in the previous chapter (Sec. 5.3 and 5.4), some weaknesses in the activity space were identified. Restrictions, identical for ZPDES and ExpSeq for fair comparison (Appx. B), are introduced to force the verification of particular mastery levels before allowing access to certain activities.

Moreover, several ZPDES variants are introduced. ZPDRD uses a random draw in place of a stochastic selection based on empirical learning progress. Choice is part of Lopes and Oudeyer (2012) model used to develop the HMABITS framework and has a positive motivational impact and an efficient vector of performance (Cordova and Lepper, 1996; Leotti and Delgado, 2011; Murayama et al., 2013). Two conditions are introduced to study its impact. ZCO does not change ZPDES algorithm but introduces contextual choice on the objects the students train with. Its goal is to increase intrinsic motivation by adding a preference depending on the student personality and thus introducing an emotional valence on the object (Carstensen et al., 2003). Another interesting aspect is to study the impact of providing pedagogical and cognitive choice to learners (Dweck and Leggett, 1988). ZCA has been developed in this optic by giving choice to the student between different activities that ZPDES selects.

To limit the number of experimental condition, no version of ExpSeq was made with choice, it is only used as a baseline. This leads to 5 experimental conditions presented in figure 6.1. No control group is made due to the evaluation of the system efficiency in the previous study (Sec. 5.3.3).

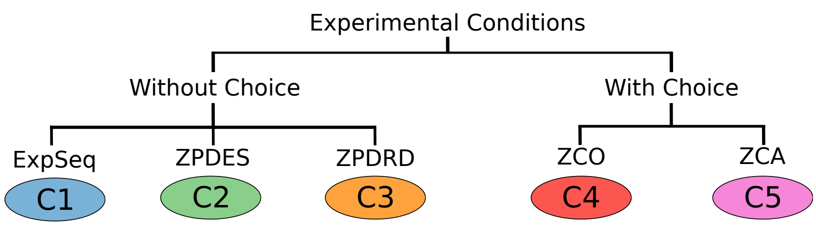


Figure 6.1: Experimental Conditions in Study 3. Three conditions without choice for the students and two conditions with choice for the student. There was around 100 students per condition at the beginning of the study.

6.1.1 ZPDRD : ZPD and Random Draw

ZPDRD is a variant of ZPDES. It uses the same mechanisms to manage the ZPD (Sec. 2.4) which are an exploration of an Activity Space with activating/deactivating mechanism depending on the success rate of the student. The different thresholds constraining the exploration depend on the activities mastery. However, active activities in the ZPD are no longer proposed according to the evaluation of their quality based on the learning progress or their necessity to activate a new parameter, they are drawn according to a uniform law. This means all possible active activities in the ZPD have the same probability of being proposed. ZPDRD is introduced to evaluate the relevance of using the learning progress to compute the quality of parameter values.

6.1.2 ZCO : ZPDES with Choice of Object

As explained before, studying the impact of choice as a motivational and learning tool without link to pedagogical content is interesting. ZCO is an experimental condition where the student has choice over a contextual parameter: the objects presented on the screen. In the money game scenario (Sec. 4.2), students compose sums corresponding to the item prices or the change to be given when a customer purchases an item. The choice given to the student is between two different objects, but the activity parametrisation is the same. The activity is still selected by a ZPDES algorithm and the choice has no impact on the ZPDES operation. The interface used to implement this experimental condition is shown in figure 6.2. Only the type of exercise and the objects are presented to simplify the interface and and reduce the perturbation of the student to a minimum. The position of the choice icon on the screen is determined randomly to avoid presentation bias.

Figure 6.2: Object choice interface. The student choose the object(s) he wants to train with by typing on it. The bubble indicate instruction "Choose what you want". (Full page in appendix C)

6.1.3 ZCA : ZPDES with Choice of Activity

The goal of ZCA is to give the opportunity to the learner to be active in the learning process by choosing the activity he wants to work on. ZCA will propose, depending on the pedagogical and population context, n_C different activities, that are active in the ZPD and drawn according to their quality, for the learner to choose. If the number of active activities is smaller than n_C, then the same activity can be drawn multiple times. An example of interface is shown in Figure 6.3, where the student can choose between the R1 and M4 exercises. In this case, only two choices are given to limit the cognitive surcharge a larger number of choices could induce in students. Also, the fact that students can be more attracted by some objects than others can bias their choice of activity. Thus, the objects are not displayed to avoid perturbing the student's selection process and allow them to concentrate on the activity they want to work on. The choice's position on the screen is randomly selected to avoid bias.

Figure 6.3: Activity choice interface. The student choose what he wants to do by typing on the corresponding activity. Here the choice is between R1 and M4 exercise. The bubble indicate instruction "Choose what you want". (Full page in appendix C)

6.2 Experimental set-up

THE EXPERIMENTAL SET-UP includes several main components. It uses the Kidlearn game[1]. Several tools are introduced to measure psychological and profile characteristics of the students. A planning was designed to articulate the different tools around the Kidlearn game inside each session. Moreover, a pilot study was carried out to test the set-up.

[1] Described in Sec. 4.2, it mainly contains four types of exercises M, R, MM and RM. There are six difficulty levels for type M and four levels for types R, MM and RM.

6.2.1 Measurement toolkit

A variety of tools was developed during this study to measure student features and their motivational and psychological state through working sessions. These metrics are mainly defined as questionnaires presented in appendix B.2. These tools were developed in collaboration with Hélène Sauzéon, researcher at HACS laboratory (EA4136) at Bordeaux University and associated researcher in Flowers Team at INRIA Bordeaux, Alexandra Delmas who used similar tools during her PhD (Delmas, 2018) and Josias Levi Alvares, an internship student supervised during this study.

6.2.1.1 Profile metrics

Several measures are done to build profiles. The General Profile (GP) questionnaire (appendix B.2) groups questions related to student information such as gender, familiarity with technologies, his perception and habits to use money. Another set of questions concerns the habits of choosing in life-related decision making such as clothes or food choices: it is used to probe the student's self-determination trait[2]. Also, each teacher received a questionnaire to provide information such as the student's age, literacy and mathematical level.

[2] When a behaviour is self-determined, the regulatory process is choice, but when it is controlled, the regulatory process is compliance (or in some cases defiance). (Deci, Vallerand, et al., 1991).

A questionnaire to evaluate the psychological profile of the student at school is also used to build the initial student profile. It brings together two questionnaires based on two different works. The first evaluates the quality of school life (Lazar, 1999) with a 70% prediction of disengagement behaviours in school. It is multi-component including items on satisfaction at school, the student's interest in academic learning, and the nature of the student-teacher interactions / students' attitude towards the teacher. The second questionnaire is based on the one developed by Weber et al. (2005), which measures the learner's empowerment with questions such as "This course will help me achieve my future goals" and "I have the qualifications to succeed in this class". The two questionnaires were combined and reworked to produce the School Profile questionnaire (SP), which contains 10 items on a 5-point Likert scale (appendix B.2); 5 items are related to school while the others to relationships.

The goal of these information is to establish an initial profile for each student to compare and correlate motivation and learning measures for all student features.

6.2.1.2 *Motivation metrics*

Two questionnaires measure motivation. The first one is the Intrinsic Motivation (IM) questionnaire presented by Corbett and Anderson (1994). It was used in Kidbreath studies (Delmas, 2018) and adapted for Kidlearn with the addition of two items. It measures the intrinsic motivation of students playing with Kidlearn. Students answer to this questionnaire after each working session to allow a tracking of of the student's motivation through successive sessions. It is the only questionnaire which students answer several times. It contains 10 items on a 5-point Likert scale.

The second one is a questionnaire of Type of Motivation (TM) used to measure what kind of motivation a student experiences when working on Kidlearn. Vallerand's questionnaire (Vallerand, Blais, et al., 1989; Vallerand, Pelletier, et al., 1992) was selected. It is based on Self-Determination Theory of Deci and Ryan (1985) and is commonly used to ascertain the elicitation of intrinsic and extrinsic motivation (e.g. Desrochers et al. (2006)). Once adapted to a public of children users playing a serious game, children had to answer a 21 items questionnaire on their experience playing during the last session. Items were visualized by groups of three representing one of the following type of motivation subcomponent:

- Amotivation (AM) appears when a student doesn't see how the results obtained are related to his or her actions. This creates a lack of motivation.

- Extrinsic Motivation (EM) is divided into tree main subcomponents: External regulation, where behaviour is regulated through rewards, e.g. passing a grade or a level with honors; Introjection, where the learner begins to internalize external constraints, e.g. shows he is smart to others; Identification, where the learner chooses ways to achieve external purposes, such as helping a friend.

- Intrinsic Motivation (IM) includes three subcomponents: To Know, the epistemic need to know and understand, satisfaction of learning for its own sake; Accomplishments, interacting with the environment to feel competent, focus on the process of achieving rather than on the outcome; Feelings, engagement in learning to experience stimulating feelings, e.g. sensory pleasure, excitement, fun.

Motivation scores span from 0 to 3 in each category of items, presented randomly to children as to avoid learning effects. Only yes or no answers are proposed to avoid children cognitive overload.

6.2.1.3 *Ancillary metrics*

Two ancillary metrics are added. A well-being scale is created to investigate further the psychological impact of Kidlearn on the students during the working sessions. It is presented in figure 6.4.

The student must position a cursor between "the best time of my life" and "the worst time of my life" depending on how he/she feels right now. They have to answer to this scale at the beginning, middle and end of each session. This allows us to measure approximately the evolution of the well-being of the children during each session.

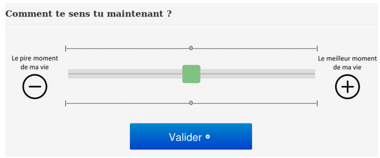


Figure 6.4: well-being scale. The student is asked a question "How do you feel ?" and moves towards + (best time) or - (worst time) depending on how he/she feels right now.

The Game Interface questionnaire (GI, appendix B.2), is based on Murchland et al. (2011) work. It is a user satisfaction questionnaire for the interface. It was used to have a full review of the students feelings about the system during this study. It contains 8 items on a 5-point Likert scale (except for one question).

These different tools are used to obtain an evaluation of the student's profile, feelings and level of motivation at the beginning, throughout and at the end of the study.

6.2.1.4 *Design process*

The design process followed several steps. A first step was to choose the content of the questionnaires and scales based on the literature and integrate it in the interface of the web application used in the experiment. This first version of the tools was tested in a pilot study with 59 children in two classes. It was modified related to the pilot study observations. As discussed in chapter 5.4, no paper was used, all interactions were digitalized to simplify the analysis of the data.

Figure 6.5: Fun Toolkit. Smileyometer which groups a set of smiley to illustrate questionnaire items.

The Fun Toolkit (Read, MacFarlane, and Casey, 2002) presented in figure 6.5 can be used to illustrate questionnaire/scale items and make answers more understandable by students. According to Read and MacFarlane (2006) the Smileyometer is not very useful for children younger than 10 years old (7-8 years in this case) because variability of the responses is very low. Young children tend to choose the highest (most positive) score. A goal of the pilot study was to test this phenomenon. One version presented a Smileyometer, while another version did not and used "+" and "-" sign to illustrate the negative or positive answers (Fig. 6.4). The pilot study observations led to the same conclusion as Read and MacFarlane (2006). The Smileyometer was even counter productive by inducing biases in students' answers. Looking at the data was not even necessary, it was clear by just observing them interact and answer. They were discussing and declaring "I chose the happy face" while no such behaviour was observed when there was no Smileyometer. Even if children tend to choose the most positive score anyway, it was less prominent. The Smileyometer was thus not used during the actual experiment.

Observations about the game interface (not observed in the previous experiment) were made. Some children were reluctant to use their three attempts to answer an exercise because they did not want to loose all their "life" represented by three hearts above the head of the character they were playing (client or merchant). These hearts were removed from the interface, and the students were just informed to have three trials to succeed for each exercise.

6.2.2 *Organization of the sessions*

Two people managed the experimental sessions. The equipment was composed of a set of 30 tablets, two computers and two wifi routers. This equipment was carried in every school to be independent from school equipment constraints and limit equipment bias. Four experimental sessions are organized over two weeks where the game phases, questionnaires and scales follow one another in particular orders. The organisation of each session is presented in table 6.1.

Session 1 (1h20)	Session 2 (50 min)	Session 3 (50 min)	Session 4 (40min + bonus)
1. Project explanation	1. Well-being scale 1	1. Well-being scale 1	1. Well-being scale 1
2. Well-being scale 1	2. Game phase (30 min)	2. SP questionnaire	2. TM questionnaire
3. GP questionnaire	3. Well-being scale 2	3. Well-being scale 2	3. Well-being scale 2
4. Pre-test (20 min)	4. IM questionnaire	4. Game phase (30 min)	4. Post-test (20 min)
5. Well-being scale 2	5. GI questionnaire	5. IM Questionnaire	5. Well-being scale 3
6. Game phase (30 min)	6. Well-being scale 3	6. Well-being scale 3	6. Bonus game phase
7. Well-being scale 3			
8. IM questionnaire			

Table 6.1: Sessions planning. The table shows the sequence of steps for each session. There are 8 steps in the first session and 6 steps in the other sessions.

There are some points to consider in particular. Three well-being scales are done per session to track student feeling throughout each session. A game phase is always followed by a well-being scale and an IM questionnaire (except for the bonus phase). The questionnaires are distributed over the 4 sessions so as not to ask too many questions at a time. In the first session, the General Profile (GP) questionnaire is done first. They have all the time they want to answer. It allows students to acclimatize to the tablet, the web site interface and to be confident in the tablet usage.

The Game Interface (GI) is done after a game phase to collect fresh student impressions. The School Profile (SP) questionnaire is done before a game phase to collect the student's feeling without the bias the game session could bring. The Type of Motivation (TM) is done before the post-test to collect impression about the game sessions without the bias the post-test could bring. A bonus game phase is proposed at the end of the experiment for the students who want to do it. Its aim is to avoid frustration after the post-test. This bonus phase is not part of the experiment, the data collected are not used and there is no well-being neither IM questionnaire after.

The original plan (elaborated before the pilot study) was to do each session as a whole class. This was changed after the first session because the students were strongly distracted. Having 30 students working side by side on tablets in the same room appears to not provide a good environment for the experiment. The class was then divided in two for the rest of the experiment. Each group was doing the same session in parallel in different rooms supervised by a researcher.

Also, each student was not finishing each step at the same time leading to even more distraction and logistical issues. So a process was developed to conduct the sessions and manage the behaviour of the students. Figure 6.6 shows part of supervisor interface (top picture) and student waiting page (bottom picture).

At each step of the session, finishing a step leads the student to the waiting page. While waiting for the others to finish the step, students can draw on their draft or if waiting time may be long (as for tests, some finished what they can do in 10 minutes), they can read a book present in the classroom or discuss with other that also finished without disturbing the classroom. This process allowed to considerably reduce distraction and provided a better and quieter working environment for the students.

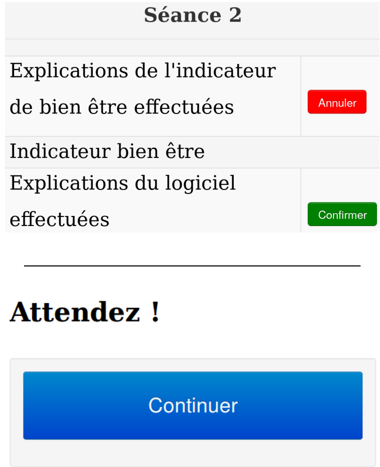


Figure 6.6: Supervisor interface is presented on top picture. Explanation of well-being scale is done (red button), and game explanation is in progress (green button). During this time, the students only have access to the waiting page which says "Wait !" with a continue button. As long as the researcher does not authorize the students to continue, clicking on the "Continue" button has no effect. When the explanation is over, the research clicks on the green button and the students can click on "Continue" to pass to the game phase.

6.3 *Experimental population*

This experiment was carried out in 24 classes of the Bordeaux urban area, with a total of 581 students.. It was done in partnership with the educational institutions of Bordeaux. The pilot experiment was made with 2 classes (59 students), but the data cannot be used like the ones of the actual experiment. Moreover, some data cannot be used because of the absence of some students to the sessions. Due to time and space constraints, these students could not compensate for the missed sessions and their data cannot be taken into account in the same way as the others. This reduces the final number of student with exploitable data to 428. This is a huge lose of 18% exploitable data. This shows a limit in this kind of experimental set-up where external constraints (time, space and people) can't be easily managed or adapted and can lead to unusable data.

In spite of some data being unusable, a huge amount of data was still gathered during the experiment. There is a lot of analysis and correlation that can be done. The time between this thesis writing and the end of this experiment was a bit short, so the results presented in this section are preliminary.

The population is divided into 5 experimental conditions corresponding to each algorithm. A condition was assigned randomly for each experimental group supervised by a researcher. All the students of a group have the same condition to avoid questions and frustration from students seeing differences on their mate's screens. Even with the loss of data, final groups have almost the same number of students : ExpSeq (86), ZPDES (85), ZPDRD (86), ZCO (83) and ZCA (88). Several student characteristics were measured to evaluate initial differences between the condition populations. Comparisons between experimental conditions (and thus between algorithms) are harder to make if the characteristics of their populatoins vary significantly.

6.3.1 *Initial population state*

Student profiles Some teachers did not provide information about the student's level and date of birth, making the data unexploitable. However, several other characteristics were measured through the profile questionnaire. The main ones are the gender, the frequency of screen device use (tablet, computer, smartphone), the habit to use money, enjoyment doing calculations and the habit to make choice. A χ^2 test was used to test the null hypothesis of independence between each condition. The significant differences (p-vales < 0.05) are shown on table 6.2.

	Enjoy doing calculation		Device use frequency				Gender	
	No	Yes	Never	Week-end	During week	Every day	Girls	Boys
ExpSeq	8	78	2	20	53	11	40	46
ZCO	18	65	11	15	42	15	54	29
ZCA	19	69						
χ^2 test p-values								
ExpSeq/ZCO	0.0436		0.0323				0.0231	
ExpSeq/ZCA	0.0425							

Table 6.2: χ^2 test which are significant to determine student profile differences between each group. Students in ExpSeq condition seem to appreciate calculation more than in ZCO and ZCA. And they also seem to be more used to using screen devices than in ZCO condition.

From this test, there are significantly more girls in ZCO condition than in ExpSeq condition. Students in ExpSeq condition seem to appreciate calculation more than in ZCO and ZCA, and they also seem to be more used to using screen devices than in ZCO condition. This shows these conditions do not present homogeneous population. Thus, particular precautions must be taken during analysis between ExpSeq results with ZCO and ZCA results.

Pre-test The pre-test is used to evaluate the students' initial skill level. Its goal is to evaluate evaluate the progress made by the student by comparing it to a post-test done at the end of the sessions. The pre-test is also used to compare the skill level of each condition population and verify there are no significant differences between their initial skill distribution. The distribution of notes in each condition is presented on figure 6.7 and the significant p-values are presented on table 6.3. The students in ExpSeq are significantly better at the beginning of the study, making it impactical to analyse the raw data from the experiment. The population has to be reduced by selecting adequate students to be able to exploit the data.

	ExpSeq	ZPDRD	ZPDES	ZCO	ZCA
Pre-test mean	7.7	5.9	6.4	5.8	6.3
Pre-test median	8.0	5.5	6.0	5.0	5.0
ANOVA p-values to ExpSeq		0.025	0.003	0.002	0.014

Table 6.3: Mean and median in each group with ANOVA post-hoc pairwise comparisons p-values to compare each condition to ExpSeq ($\leqslant 0.05$). The differences with ExpSeq are significant. Differences between other groups are not significant.

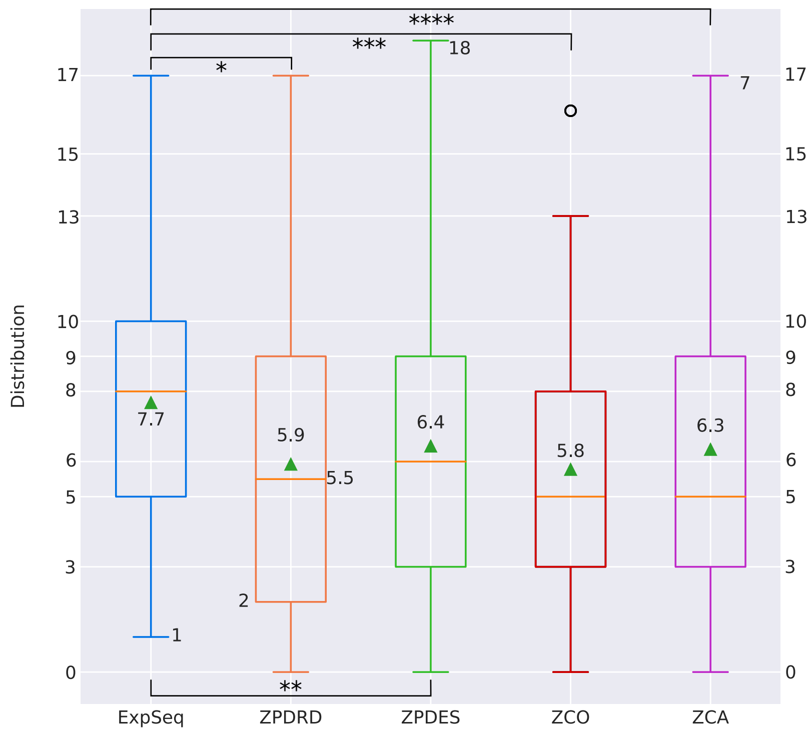


Figure 6.7: Pre-test result distribution for each condition. the median value is represented by an orange line and the mean value by a green triangle. Students in expert sequence condition have a higher initial level than students in other conditions.

6.3.2 *Population selection*

The selection of the students whose data are analysed is based on their results at the pre-test. The student's pre-test ratings are between 0 and 19. Each condition population is reduced to have the same number of student for each rating leading to the exact same pre-test rating distribution in each condition. This leads to 45 student per group with the distribution per rating described in table 6.4. To have a fair student selection among the condition, the same rule is used to remove students from the different conditions, based on their post-test ratings.

For example, if 10 students obtained a rating of 5 in a condition, 5 students with this rating have to be removed to obtain the necessary number of 5 shown in the table. The student with the lowest rating at the post-test will be first removed, then the student with the highest post-test rating will be removed and so on. This leads to the selection of medium-performing students. The final population is composed of 225 students with 45 students per condition following the same distribution over pre-test rating as described in table 6.4. The analysis made in subsection 6.3.1 has also been done on this population. There are no significant differences between the initial profile measures, so these condition groups seem to be comparable. The results presented in the following section come from the analysis of this particular population.

Rating	1	2	3	4	5
Number	4	5	4	5	5
Rating	6	7	8	9	10
Number	6	2	4	4	3
Rating	11	12	16		
Number	1	1	1		

Table 6.4: Number of students per pre-test rating for each condition, to build the final population. The total number of students over all ratings is 45.

6.4 Results

The conditions are compared through the students activities, their results in the post-test and the psychological measures obtained with the measurement toolkit. Some preliminary results are presented, which open the path to further and deeper data analysis.

6.4.1 Game activities

Figure 6.8 presents histograms showing the number of students doing each activity (only showing the parameters "Exercise type" and "Difficulty level") at each step and for each condition. As in the experiment presented in chapter 5, ExpSeq graphic presents a particular shape, where the students with the Expert Sequence start to do more diverse exercises later than students with algorithms. Students in ExpSeq start MM (purple) activities after 12 steps, when some students started MM type between step 7 and 8 with algorithms. The same kind of observation can be done with R (green) and RM (red).

ZPDRD, ZPDES, ZCO and ZCA present graphics with similar shape. But there is a small difference in the ZPDRD graphic. In these four conditions, a majority of students have started to do activities with type R and MM around step 15 with a progressive decrease of number of students that work on type M (blue). But for ZPDES, ZCO and ZCA, the number of type M activity increases again after around 30 steps, which is not the case with ZPDRD. This may come from ZPDES ability to increase parameter quality when they are needed to activate a new parameter which is not the case of ZPDRD. A more detailed analysis of ZPDES, ZCO and ZCA graphics shows a majority of students doing R and MM types between around step 15 and 30. M is evaluated with a low quality due to the student's mastery. But the next level of the R and MM exercises require M5 mastery (decimals). Then, the algorithm increases M quality. This does not happen with ZPDRD, because it proposes each type of exercise in the same proportions.

Students with ZCA seem to have done less activities. 25 students finished before doing 75 activities with ZCA, 95 activities with ExpSeq, 93 with ZPDRD, 105 with ZPDES and 105 with ZCO. This phenomenon may be due to students not wanting to do some activities and thus no making a choice during some time. This behaviour was spotted during class session. This could also be explained by the students taking time to choose what activity to do (and more time than student choosing object). This may be investigated by checking the logs about choosing time of the students. For the next part of the results, only the 100 first steps will be analyzed to try to reduce this bias. Students with ZCA seem to have made less activity of type RM than in other conditions. Two reasons can be considered: the students did not choose this activity, or they did not have the opportunity to do so because they did less activities than others beforehand (as described above).

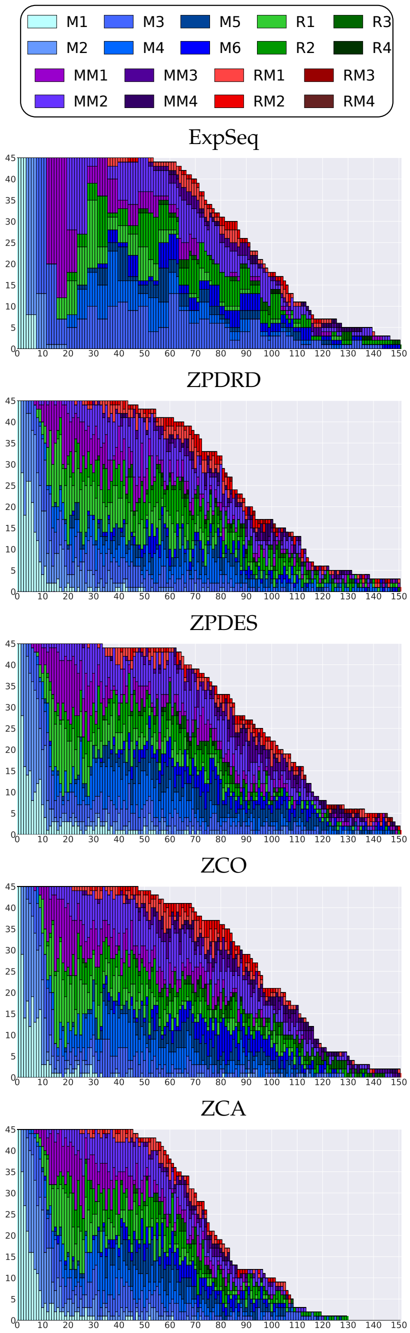


Figure 6.8: For each condition, the histograms show the number of students (ordinate) doing each type and level of exercises at each time (abscissa). There are 4 types of activity M, MM, R and RM with their related levels (M: 6, MM: 4, R: 4, RM: 4).

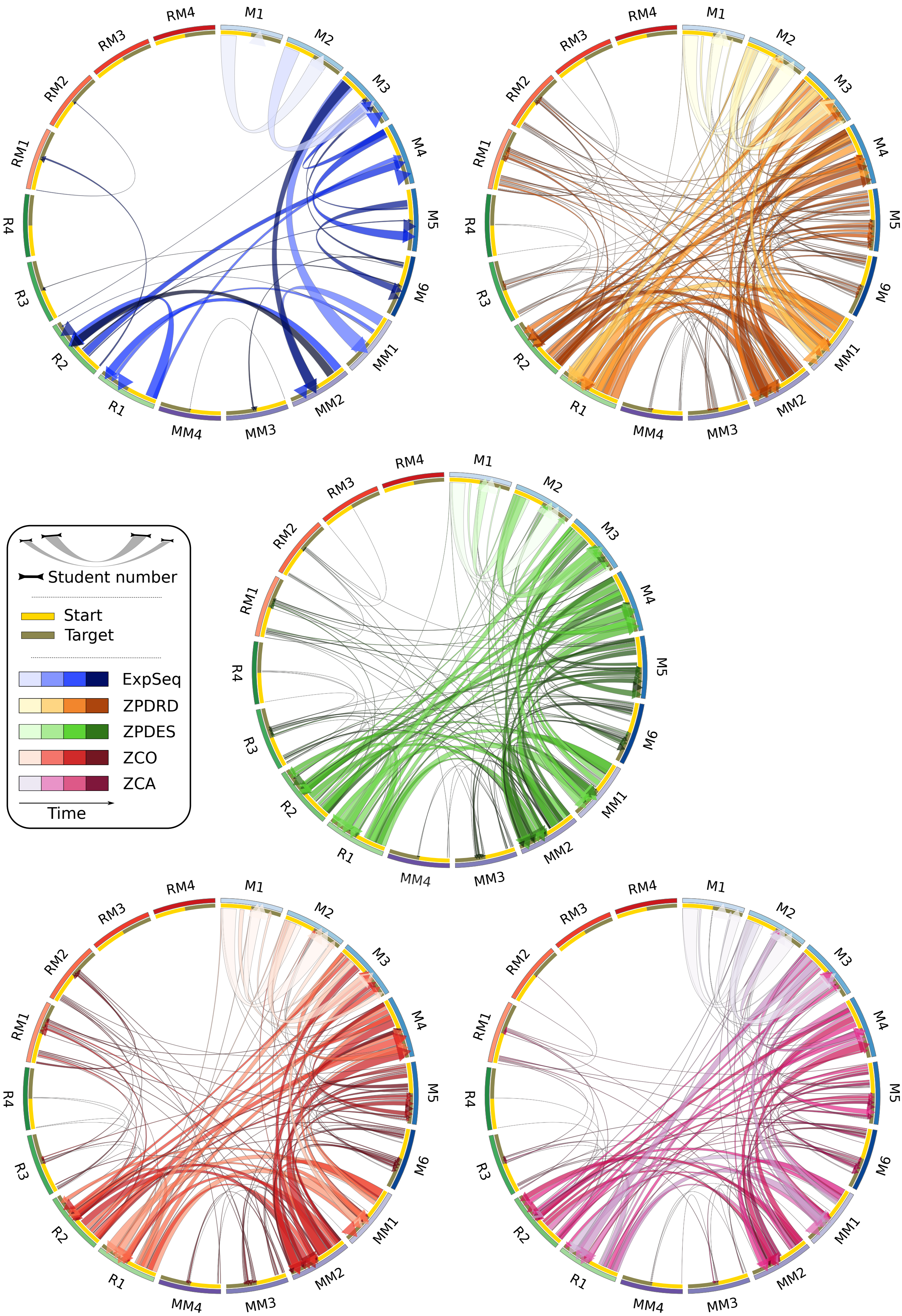


Figure 6.9: For each condition, a Circos graph shows the number of students that did transitions between activities where students transitioning from one activity (start) to another (target). The line thickness is proportional to the number of students. Light intensity corresponds to time. A larger variety of paths is proposed by ZPDES versions resulting from the online personalization. Only main transitions are displayed (frequency $\geqslant 5$).

Figure 6.9 shows circular designs made using Circos (Krzywinski et al., 2009). On this figure, the transitions between exercises made by students along time are represented by the colored curved lines (one color per sequence manager). A transition starts on an exercise (start), situated on the yellow part of an exercise, and finishes on another (target), situated on the brown part of an exercise and represented by an arrow. The line thickness represents the number of students who did that transition. The time is represented by the color shade, light colors correspond to early exercises, darker colors to later ones. Students present a smaller diversity of transitions with ExpSeq than with other algorithms, because of the more rigid management of activity provided by ExpSeq. As described before, students working with ExpSeq do fewer high level activities like $R3-4$, $MM3-4$ and RM. This confirms observations made in chapter 5. Students working with ZCA present less variety of transitions than with other algorithms. This may be due to the students having their own strategy of choice[3], which can lead to a smaller diversity of transition. It could also be related to what was described previously – that students with ZCA do less activities and if students do less activities they present less transitions. On the contrary, students working with ZPDRD seem to present a larger diversity of transitions, which may come from the random selection policy. Further analysis will provide more information about these phenomena.

[3] For example, some students were always selectionning the most difficults exercices while others were always selecting the simplest ones.

To compare more quantitatively the activities made by the students, two scores are built. The first one represents the activities reached by a student in the activity space, and the second one represents the success rate over these reached activities. For a student, these scores are defined as follows:

$$score^{reached}(t) = \sum_{i=1}^{4} max(\{l^{i,j}(t) \mid j \in L^i\})f^i \tag{6.1}$$

$$score^{success}(t) = \sum_{i=1}^{4} max(\{\delta_{i,j}(t)l^{i,j}(t) \mid j \in L^i\})f^i \tag{6.2}$$

	M	MM	R	RM
Index i	1	2	3	4
Factor f^i	1	2	3	4
Levels L^i	1-6	0-4	0-4	0-4

Table 6.5: Table of factor, index and the number of levels for each type of activity to compute scores in equations 6.1 and 6.2. The level 0 represents the fact a student has not made any exercise of this type yet. Students start with exercises of type M, so there is no level 0

where i is the index corresponding to each type of activity, f^i is the factor related to the activity type i as described in table 6.5. If the level j for type i has been reached at time t, then $l^{i,j}(t) = j$, or else $l^{i,j}(t) = 0$. $\delta_{i,j}(t)$ is the student's success rate over the 4 last steps activity type i, level j. For example, at time t, if a student has reached M4, MM2, R1 and did not reach RM, his $score^{reached}$ is equal to : $4 \times 1 + 2 \times 2 + 1 \times 3 + 0 \times 4 = 11$.

Figure 6.10 shows the time evolution of the average score of the students for each condition. As expected, the criteria $score^{reached}$ is much lower for ExpSeq than for the others. For the $score^{success}$ criteria, it is significantly behind mostly for the 35 first steps. ZCO seems to present the best performance, it is significantly better than ExpSeq from step 80 to the end, while ZCA, ZPDES and ZPDRD seems equivalent when compared with this metric.

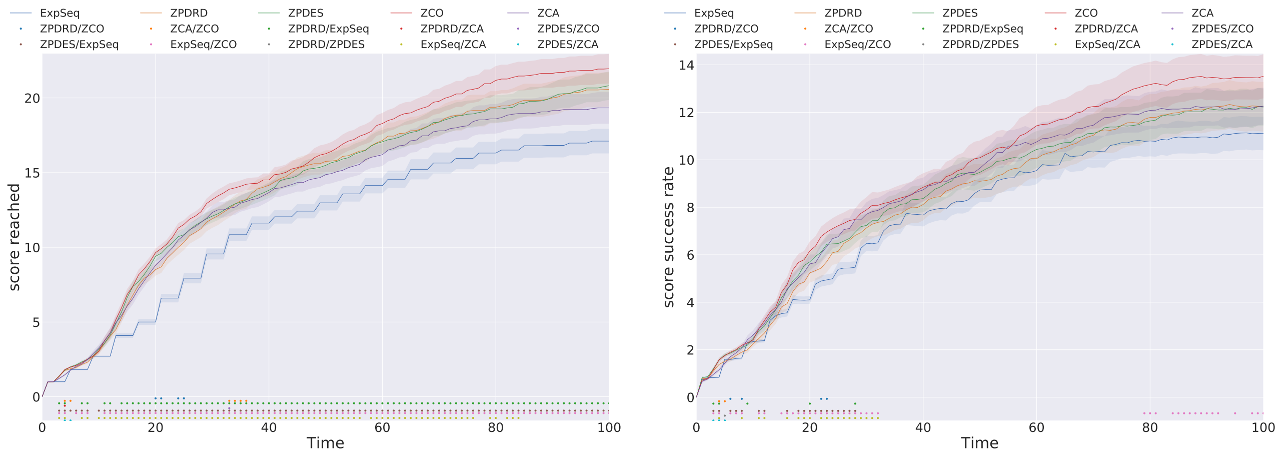


Figure 6.10: Evolution of $score^{reached}$ and $score^{success}$ over time for each condition. The curves represent the average score over all student for one condition. The shaded area represent the standard error of the mean. Colored points indicate if the score differences are significant two by two.

These observations confirm that the algorithms provide a better variety of exercises and allow students to perform better on the activities. The difference between algorithms seems less obvious. ZCO seems to present the best performances, student made more activities and qualitatively performed better which could indicate a stronger motivation. Also, even if ZPDRD presents a different behaviour from other ZPDES versions, it does not seem to impact the performances of the students based on the above-mentioned scores.

6.4.2 *Motivational measures (detailed statistics in B.3)*

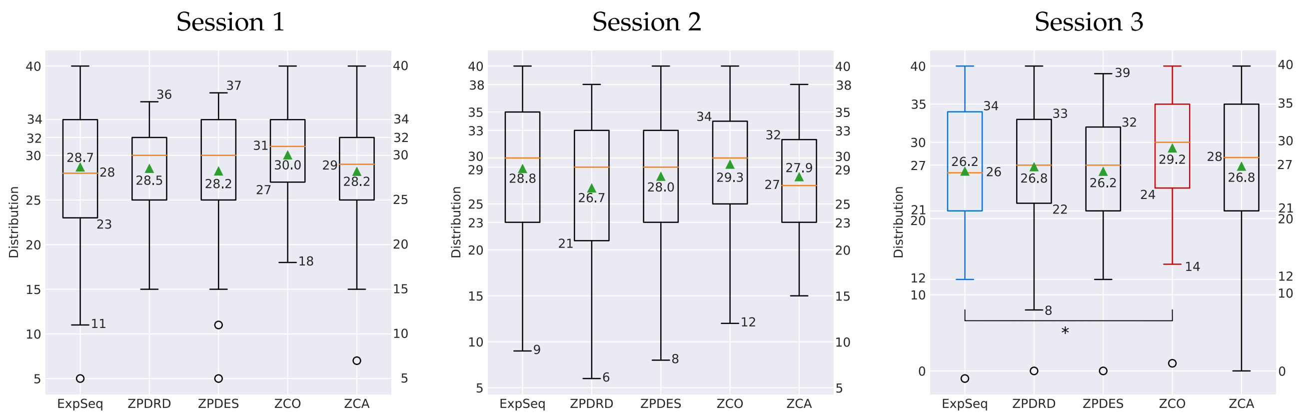


Figure 6.11: Results on IM questionnaires for sessions S1, S2 and S3. Students seem more motivated with ZCO. Conditions with significant differences are colored.

IM questionnaire Figure 6.11 shows measures of the IM questionnaire. There are no significant differences for the two first sessions, even if the students seem a bit more motivated with ZCO and less motivated with ZPDRD and ZCA at the end of session 2. For the third session, ZCO students are significantly more motivated than the ExpSeq ones with a mean of 29.2 and a median of 30 (p-value $= 0.048$). The students alsoe seems more motivated with ZCO than with the other ZPDES versions at the end of session 3 but these differences are not significant (all p-values > 0.05). However, the general trend shows a motivation that fades over time (particularly between S1 and S3) over time and reflects the students weariness. This decrease in motivation is clearly less important for students working with ZCO than with other conditions, as shown in table 6.6.

Condition	Average score difference
ExpSeq	2.5
ZPDRD	1.7
ZPDES	2
ZCO	**0.8**
ZCA	1.4

Table 6.6: IM questionnaire score differences between session S3 and session S1. The difference is less important with ZCO than with other conditions

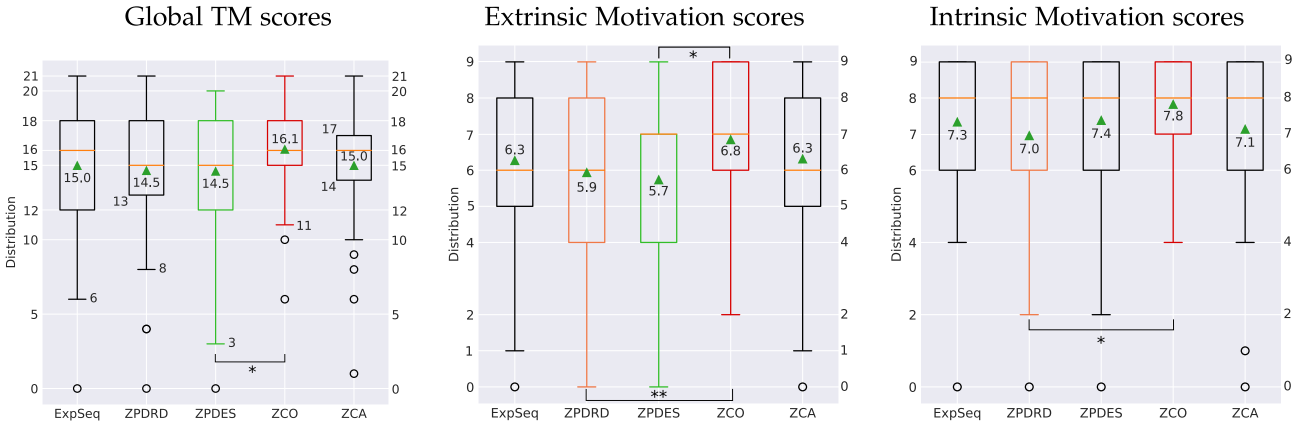


Figure 6.12: Results on TM (Type of Motivation) questionnaires. The global score is presented left, extrinsic score in the middle and intrinsic score on the right. Students seem more motivated with ZCO, globally, extrinsically and intrinsically. Conditions with significant differences are colored.

TM questionnaire Figure 6.12 shows the scores obtained with the TM questionnaire. The global score is presented on the left, the extrinsic score in the middle and the intrinsic score on the right. ZCO students seem globally more motivated than students with other conditions, with higher mean and medians values and a smaller variance. This difference is significant with ZPDES (p-value $= 0.041$) and almost significant with ZPDRD (p-value $= 0.059$). More precisely, students seem more extrinsically and intrinsically motivated with ZCO. This effect is significant between ZCO and ZPDES for extrinsic score (p-value = 0.013) and between ZCO and ZPDRD for extrinsic (p-value = 0.043) and intrinsic (p-value = 0.0323) motivation.

6.4.3 *Pre-test / Post-test comparison (details in B.3)*

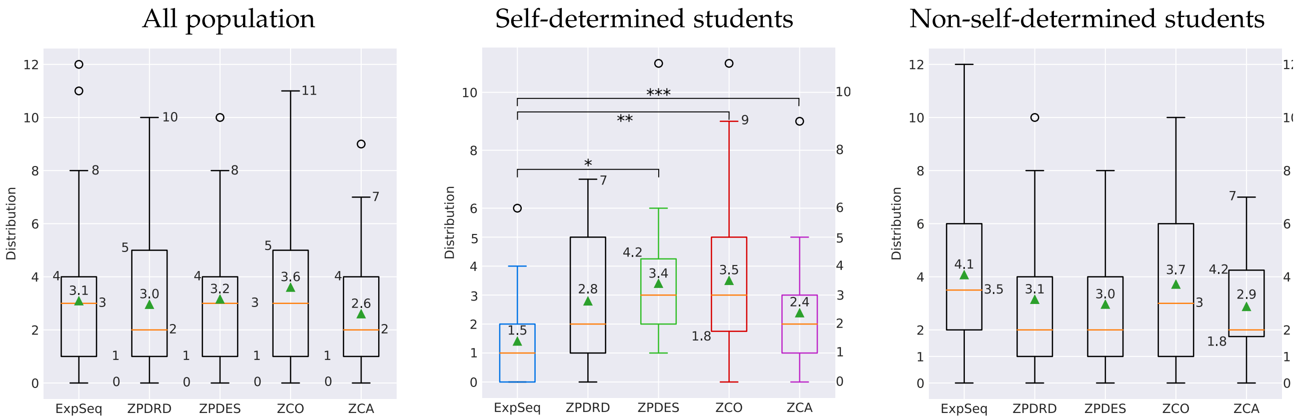


Figure 6.13: Differences between post-test and pre-test results over the whole population student, self-determined (SD) students and non-self-determined (NSD) students. SD students have learned significantly better with ZPDES, ZCO and ZCA than with ExpSeq, but not with ZPDRD. NSD students seem to have learned better with ExpSeq but not significantly. In both case, ZCO students seem to have learned well. Conditions with significant differences are colored.

Figure 6.13 shows the differences between pre and post-test which is a metric to measure learning. This measure is made in three cases. The first case, on left, presents this measure for the general population. Students with ZCO seem to have learned more than in other conditions, but differences are not significant (p-values > 0.1). The population has been divided in two according to their self-determination score in the General Profile questionnaire (Sec. 6.2.1.1). "Self-determined students" have a self-determination score higher than the median, and "Non-self-determined students" have a score below the median. This characteristic of the students has a clear effect on the differences between pre and post-test.

For the self-determined group, students have learned significantly better with ZPDES, ZCO and ZCA than with ExpSeq (p-values $<$ 0.05). This also seems true for ZPDRD, but the result is less significant (p-value $=$ 0.056). Non-self-determined students seem to have learned better with ExpSeq, but this difference is not statistically significant (p-values $>$ 0.2). One conclusion is that when students work with ExpSeq, they present a clear and significant difference in learning efficiency depending on their self-determination. This phenomenon is no more true with the algorithms. These observations lead to a possible correlation between self-determination and dependence on the method for learning efficiency.

Let's take a look at the interactions between self-determination, motivation and condition through TM questionnaire scores.

6.4.4 *Self-determined (or not) population TM scores (details in B.3)*

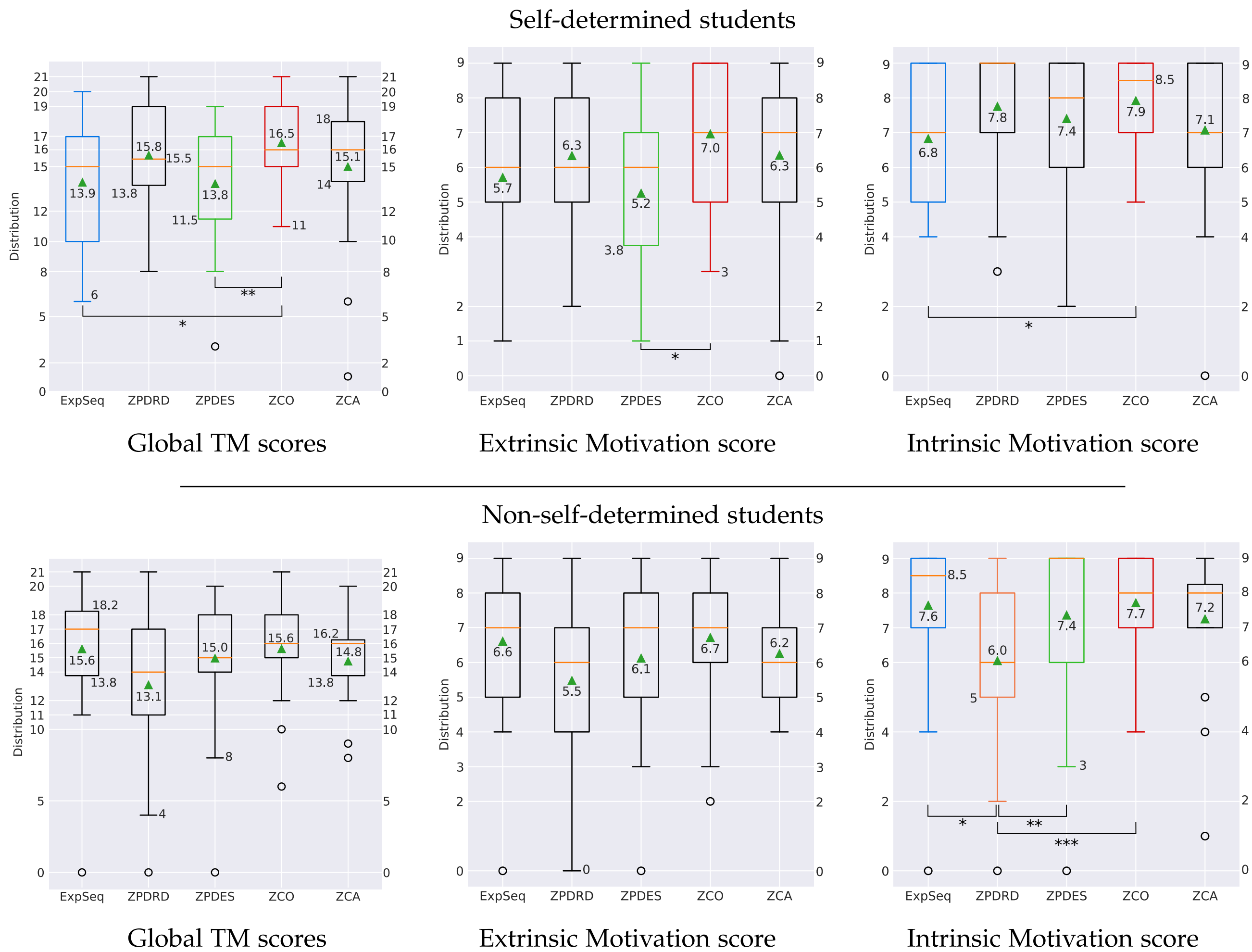


Figure 6.14: Results of TM questionnaires with self-determined (SD) and non-self-determined (NSD) students. The global score is presented on the left, extrinsic score in the middle and intrinsic score on the right. Conditions with significant differences are colored.

Figure 6.14 shows TM scores for self-determined (top) and non-self-determined students (bottom). For the self-determined population, ExpSeq and ZPDES students seem to be the less motivated ones, whereas students working with ZCO are still the most motivated. This difference is globally significant between ZCO and ZPDES (p-value $=$ 0.017) and between ZCO and ExpSeq (p-value $=$ 0.028).

More precisely, students are significantly less extrinsically motivated with ZPDES than with ZCO (p-value = 0.013) and less intrinsically motivated with ExpSeq than with ZCO (p-value = 0.047).

For the non-self-determined population, students working with ZPDRD seem extrinsically and intrinsically less motivated than the other ones. This difference is significant for intrinsic motivation, where ExpSeq, ZPDES and ZCO students are significantly more motivated than the ZPDRD ones (p-values: 0.006, 0.024, 0.020).

These last observations tend to show that ExpSeq and ZPDES are less efficient for self-determined students, while ZPDRD seems to be less efficient for non-self-determined students.

6.5 Results interpretation and discussion

This study had several goals. The first goal was to evaluate the impact of the Kidlearn framework on motivation and learning compared to an Expert Sequence without machine learning. The second goal was to observe the impact of the ZPDES activity quality evaluation policy compared to a random policy implemented in ZPDRD. The third goal was to observe the impact of different kind of choice implemented in ZPDES. The last goal was to use the psychological and contextual data measures to see if correlation can be observed between the students psychological state evolution, their profile, their motivation and their learning.

The different observations show that generally, ExpSeq provides a weaker learning experience that the other algorithms. In particular, it provides a less motivating and enriching experience to self-determined students. This consolidates the results of the experiments presented in chapter 5 and answers the first goal.

At first, ZPDES and ZPDRD do not show major differences in terms of performance and motivation, except for the qualitative difference in the spectrum of activities proposed to the student, as shown in Sec. 6.4.1. However, the last analysis showed that ZPDES is significantly more intrinsically motivating for non-self-determined students and tends to be less extrinsically motivating for self-determined students than ZPDRD.

There is a strong learning and motivation difference between self-determined and non-self-determined students with ExpSeq which offers activities with a lower challenge and diversity to the students. From this observation, non-self-determined students seem to prefer less challenging and diverse activities whereas self-determined students prefer the opposite. ZPDRD proposes more challenging activities than ZPDES due to its randomness: it proposes equiproportionally very hard, normal and easy activities. On the opposite, ZPDES focuses on activities that maximize the learning progress of the student, which are generally not the hardest ones.

Thus, ZPDES seems generally adapted to both population while ZPDRD seems better for only one. This could be a sign of the good impact of the ZPDES mechanisms to propose activities based on the learning progress of the student. This result is particularly interesting and partially answers the second goal of the study. Further analysis should be done to consolidate the interpretation of this phenomenon.

Offering choice allows a higher integration of the intrinsic motivation concepts and the model from Lopes and Oudeyer (2012) into the HMABITS framework. Its goal is to increase the learner's sense of control and self-determination during the activity sequence (Deci and Ryan, 1985). Motivation and performance measures show the students working with ZCO are more motivated and efficient when training on the Kidlearn game than students working with other conditions. Giving contextual choices to the students makes them more motivated and more efficient. This observation is in accordance with the results of Cordova and Lepper (1996), Leotti and Delgado (2011) and Murayama et al. (2013) who describe choice as a vector of motivation and performance.

A big difference with Cordova and Lepper (1996) is the possibility for students to make choices that have a pedagogical impact on the activities (ZCA), when this previous work only proposed contextual choices to avoid the risk of pedagogically poor choices from students[4] due to their lack of knowledge about the activity (Thomas W. Malone and Lepper, 1987; Steinberg, 1989). This risk should be less important here, because choices proposed are over activities evaluated to be relevant by ZPDES. Then, students are not required to perform an appropriate pedagogical choice but have the possibility to do the activity they prefer. But the effects observed with the contextual choice (ZCO) do not appear with activity choice.

[4] "We offered students choices only over instructionally irrelevant aspects of the learning activity. In this way, we sought to take advantage of the demonstrated motivational benefits of the provision of choice, without running the risk that students might make pedagogically poor choices if allowed to determine instructionally crucial aspects of the activity."
Cordova and Lepper (1996)

ZCA seem to be as efficient as ZPDES or ZPDRD. However, it presents particular behaviours as the small number of activities completed by the students. This could be due to too much complexity for students to make a choice on activities and induce a certain cognitive load (Sweller, 1994) or, if a student does not want to do any of the proposed activities (due to a fear to no succeed or other reasons), maybe he can stay blocked without making a choice. This effect could be due to the student age and the nature of the activity, since the students may not be able to apprehend or understand the consequences and changes induced by their choices. This could cancel the positive properties of offering choices (Thomas W. Malone and Lepper, 1987). These observations partially fulfil the third goal, although further analysis is required, in particular to analyse in more details the effect of pedagogical choice.

A lot of different metrics were used to interpret and find correlations in the data. Several correlations appeared between self-determination, game motivation and performance. This was the last goal of the experiment and its pertinence was highlighted. Other effects may still wait to be discovered and further analysis will be done to exploit this huge amount of data.

These results could also have implications in instructional design. Indeed ZCO, which combines Zone of Proximal Development concepts, activity quality evaluation with empirical success and contextual choice, seems to be the most efficient method in terms of learning performances and motivation. In particular, it seems to adapt well to different cognitive profiles that start existing at an early age, such as the self-determination trait. This could inspire pedagogical design to develop methods as Universal Design for Learning (Rose, 2000).

However, the majority of these results come from small groups of students (45 per condition and around 20 for each self-determination condition sub-group). This must be kept in mind when interpreting the results and drawing conclusions. Building larger comparable students populations is one of the next steps to analyse this study. Other selecting strategies can also be made to analyse the population. For example, selecting post-test high-performing or low-performing students could be a rule to analyse the experiment data instead of selecting average-performing ones.

In addition to that, an experiment was conducted remotely by teachers in other Aquitaine schools. This interface, described in Sec. 6.2.2, greatly simplified their management of the experiment. However, neither their working environment nor their material was under control, so their data are not mixed with the Bordeaux experimental data. They may be included in the future.

*If you can see the light at the end of the tunnel,
you are looking the wrong way.*

Barry Commoner

7
Learning has no ends

THROUGHOUT THIS PHD, several concepts and theories coming from the fields of psychology, computational cognitive science, machine learning and instructional technology were combined to produce the HMABITS framework. This framework aims to adaptively personalize sequences of activities in ITS to keep the students motivated and maximize their learning.

To attain this goal, an Activity Space parametrization is proposed in Chapter 2.1. To manage this Activity Space, the framework uses several mechanisms to keep the student in his ZPD (Vygotsky, 1930-1934/1978). The learning progress hypothesis (Oudeyer, Gottlieb, et al., 2016) is used to assess the quality of the activities and Multi-Armed Bandit algorithms are used to find a balance between exploiting the learning progress of efficient activities and exploring the Activity Space to find new relevant ones.

This framework groups two main algorithms, RiARiT and ZPDES, presented in Chapter 2. RiARiT is able to exploit information about the domain to estimate the level of the students and to personalize the teaching sequences. ZPDES is a simpler algorithm that relies only on the measure of success and failure on exercises and on a coarse predefined exploration graph to provide a personalized teaching sequence.

THE FRAMEWORK WAS TESTED and evaluated through virtual environments with learner population models (Chap. 3 and 5). A real teaching scenario, presented in Section 4.2, was implemented to test the framework with students at school (Chap. 5 and 6). Several tools, metrics and questionnaires were implemented in order to carry out these experimentations and analyse the impact of the algorithms on learners' progress and motivation.

In virtual environments, the HMABITS framework has the same efficiency on virtual populations' learning than a POMDP model (Chap. 3) and than an Expert Sequence (Chap. 5), and even presents better performances in some interesting cases. The HMABITS framework was particularly efficient when the population of virtual students was heterogeneous and presented several different learning characteristics. This shows its capability to adapt to the particular characteristics of a learner.

During the classroom experimentations based on the Kidlearn teaching scenario, the HMABITS framework allowed students to reach and succeed more diverse and difficult activities than with an Expert Sequence (Chap. 5 and 6). The experimentation presented in Chapter 6 shows that particular students, the "self-determined" ones, learn more with the framework and tend to be more motivated than with an Expert Sequence. Moreover, compared to a random selection policy, the use of the learning progress to measure the quality of activities seems pertinent to adapt to each student's particularities. To finish, including choice[1] in the framework makes students more motivated and more efficient. This result is in accordance with the results of Cordova and Lepper (1996), Leotti and Delgado (2011) and Murayama et al. (2013), who describe choice as a vector of motivation and performance.

[1] Contextual choice has no pedagogical impact on the activity sequence.

The observations lead to the conclusion that the HMABITS framework presents a strong interest to manage and adapt sequences of activities for learners. In addition to its capability to efficiently personalize activity sequences, it has a low computational complexity, do not require training on a data-set, and it has much less assumption in relation to the cognitive and student models than other systems. However, to obtain these results, the implementation of an algorithm in the HMABITS framework requires a well structured definition of the Activity Space and a judicious computation of the learning progress and requires to empirically evaluate the teaching impact of each activity parameter.

RiARiT presents the best performances in simulations (Sec. 5.2.1) due its precise definition of the relations between activities and knowledge acquisition. But this precise definition can require a lot of work from pedagogical experts and could be impracticable on a large scale teaching scenario.

ZPDES, on the contrary, uses no information about the relations between activities and knowledge components and requires a structured and hierarchical definition of the Activity Space. Moreover, the performance gains of RiARiT in simulation do not appear in the classroom study (Sec. 5.3), where ZPDES shows results as good as RiARiT. Even if some weaknesses in ZPDES graphs and mechanisms led to skip activities and revealed some directions to improve ZPDES, the requirements for RiARiT are huge and led to put it aside. The following studies then focused on the ZPDES algorithm.

ZPDES seems very promising to adapt content in ITS but, as discussed in Section 2.6, there are still a few issues that need to be addressed. The way to compute the reward for some types of activity parameters and assess their quality does not seem very efficient. For parameters that could be used in combination with several different other parameters, such as the context or support of the activity, or the way to present it, computing their quality regardless of the other parameters can bring a loss of information about the quality of this parameter.

Calculating the quality of a parameter is done sequentially over the last activities made with this parameter. If a parameter allows a learner to progress with all kinds of activities due to the learner's preferences and sensibilities, but some activities are too difficult for the learner to progress, the evaluation of the parameter's quality can be disturbed. The computation will be done sequentially, interpreting each error the same way without considering other parameters. This can lead to a reduction of the quality of the parameter because of the last trial, since the student did not progress, even though it was due to another parameter.

Some mechanisms are being developed to counter this phenomenon. For example, the success rate can be computed relatively to each other parameter values; the quality of a parameter will thus be evaluated for each level and activity type and not generally anymore. This new mechanism could allow a better appreciation of the parameter's pertinence for each student.

Also, as the experiment presented in Section 5.3 showed, ZPDES is very dependant of the activity graph and can have unexpected behaviours if the activity graph is not well designed in amount. This requires a good expertise of the Activity Space and the pedagogical field to integrate ZPDES in an ITS.

Some critics can also be made about the metrics, protocols and measurement tools used to analyse the experiments, evaluate these methods and interpret the results. For example, the absence of differences between pre and post-tests in some cases can be interpreted in different ways.

As discussed in Section 5.4, the tests can be too short to evaluate accurately the impact of the game sessions[2] or the student may have difficulties to transfer the knowledge they acquired during the sessions to the post-test. Finally, student can be less motivated to answer correctly to a test, when it has less impact for them than playing a game.

[2] This was taken into account in the experiment presented in Chap. 6.

These considerations can lead to the exploration of other tools to measure the learning of the student. A possibility could be to add game-like tests without the possibility to have hints, and where a specific set of game activities would be proposed to evaluate the knowledge of the student in the same environment as the game sessions.

Another example of pertinent critics is about the analysis of the sequences. Extracting a meaning from the sequences of activities generated for each student is challenging and questionable. This aspect is still under study: alternative methods include the definition of distances between the student's sequences or a measure of correlation between a student's profile and the generated sequences.

Perspectives

In parallel to this PhD, the ZPDES algorithm was used in several other contexts. Ropelato et al. (2017) proposed a system for a virtual reality driving simulator including an ITS to to train the learner's driving skill. The simulator includes a lot of components, such as a model of a city or physical driving engine and they adapted the ZPDES algorithm to manage the driving activities. Due to the limited time of their experiment, they were not able to detect any significant improvement in the measured skills related to ZPDES, but this approach shows how easily ZPDES can be used and adapted to other educational fields.

During the PhD of Alexandra Delmas, ZPDES was used to personalize the student's learning path in a serious game for health education of children with asthma (presented in Sec. 4.1). The results of this study were published in Frontiers (Delmas et al., 2018) and showed that the number of contents and the time spent in the personalized condition with ZPDES were diminished compared to the non-personalized condition.

In her master thesis, Woon (2017) adapted the ZPDES algorithm and combined it with an automatic story generation system to produce an adaptive tutorial about the Thymio robot. This goal of the system is to guide a user to learn how to program the robot. The adaptation of ZPDES was validated using a variety of simulated students. These simulations showed that the system was working, at least in a simulated environment. In their user study, the constraints of the story made it difficult to exploit in depth the dynamics of the ZPDES algorithms, but there is a great potential to explore.

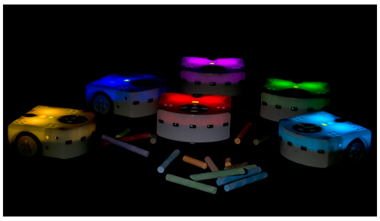


Figure 7.1: . "Thymio is a small robot which will allow you to discover the universe of robotics and learn a robot's language. You will be able to program it and carry out numerous experiments. With Thymio, the basics of robotics and programming become notions everyone can discover, whatever their age." Picture and text come from `https://www.thymio.org`.

Mu et al. (2017) combined ZPDES with automatic curriculum generation from execution traces ideas and a novel approach to determine the initial knowledge state of the student within the curriculum. They tested their approach with simulated students. The results show that their approach better tailors the number and kinds of problems proposed to students in order to master all associated topics compared to non-adaptive, hand designed curricula.

This last approaches shows an interesting opening to improve the HMABITS framework. The integration of methods to assess the initial knowledge of the students would allow the framework to propose activities directly inside the student's ZPD. This would reduce the time required by the algorithm to compute the student ZPD.

Even if the results of the experiments show that using a model might not be optimal when the goal is personalization, the use of accurate methods for learning models, such as the ones presented by González-Brenes, Huang, et al. (2014) and Dhanani et al. (2014), still needs to be evaluated. A promising approach could consist in using models to bootstrap teaching strategies and then using the HMABITS framework to personalize to each individual student.

Moreover, integrating methods to exploit the different student's traces to build the Activity Space and rearrange it dynamically could reduce even further the work needed from the pedagogical expert to build the resources of ZPDES. This information could also be transferred from one student to another based on similarities detected at runtime. Methods were recently introduced (Azar et al., 2013) to exploit transfer in multi-armed-bandits.

The exploration of further MAB and machine learning techniques must also be considered[3]. Among the possibilities are the use of contextual bandits to take into account the current state of the student and the possible parameters available, and linear bandits to consider more complicated relations between the parameters.

[3] Bubeck and Cesa-Bianchi (2012) propose a survey of Multi-Armed Bandits algorithms. Frenoy (2016) discussed the use of several MAB and reinforcement learning algorithms for education

Open Learner Models show a great potential to support student self-regulated learning in ITS (Long and Aleven, 2017). Thus, developing ways to formalize HMABITS student informations to provide feedbacks to the student about his progression and skills could help self-regulation and choice decisions processes. This could be a promising way to upgrade the framework.

A lot of other aspects could also be interesting to explore: integrating new ways to compute the reward and evaluate the learning progress; taking into account the errors of the students; using other kinds of measures such as the clicks of the students, the time passed on activities, eye tracking or heartbeat; integrating collaboration between students and studying how the ZPDs of different students can interact. The range of possibilities is vast.

However, the fact that ZPDES was used and adapted in various fields of application shows how the general structure of the HMABITS framework allows to exploit it easily. Because its mechanisms are simple and rely on general concepts and ideas which can be implemented in several different ways, it is a robust framework, independent to external factors. To improve and develop the framework, these aspects must be kept in mind because they are the key that open the possibilities to use it on a large scale. Some methods, approaches or aspects can be interesting to include in the framework, but must remain modular if they reduce its adaptability.

For example, integrating curriculum of student or dynamic reorganisation of the Activity Space should be modular options that can be used, or not, depending on the system. Even if they can have a great impact on the framework efficiency, making them mandatory could be a hindrance in the framework usage.

To all the researchers that came before me,
and graciously lent me their shoulders to stand on,
and without who this work could be very different.

Fabien Benureau

Bibliography

Anderson, John R. (1985). *Cognitive psychology and its implications*. WH Freeman/Times Books/Henry Holt & Co.

Anderson, John R., Albert T. Corbett, Kenneth R. Koedinger, and Ray Pelletier (1995). "Cognitive tutors: Lessons learned". In: *The journal of the learning sciences* 4.2, pp. 167–207.

Auer, Peter, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire (2003). "The nonstochastic multiarmed bandit problem". In: *SIAM Journal on Computing* 32.1, pp. 48–77.

Azar, Mohammad Gheshlaghi, Alessandro Lazaric, and Emma Brunskill (2013). "Sequential Transfer in Multi-armed Bandit with Finite Set of Models". In: *NIPS*, pp. 2220–2228.

Baker, Ryan S. Jd, Albert T. Corbett, and Vincent Aleven (2008). "More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing". In: *Intelligent Tutoring Systems*, pp. 406–415.

Baranes, Adrien and Pierre-Yves Oudeyer (2013). "Active learning of inverse models with intrinsically motivated goal exploration in robots". In: *Robotics and Autonomous Systems* 61.1, pp. 49–73.

Barnes, Tiffany (2005). "The q-matrix method: Mining student response data for knowledge". In: *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pp. 1–8.

Barnes, Tiffany, John Stamper, and Marvin Croy (2011). "Using Markov Decision Processes for Automatic Hint Generation". In: *Handbook of Educational Data Mining*, p. 467.

Basawapatna, Ashok R., Alexander Repenning, Kyu Han Koh, and Hilarie Nickerson (2013). "The zones of proximal flow: guiding students through a space of computational thinking skills and challenges". In: *Proceedings of the ninth annual international ACM conference on International computing education research*. ACM, pp. 67–74.

Beck, Joseph E and Kai-min Chang (2007). "Identifiability: A fundamental problem of student modeling". In: *User Modeling 2007*. Springer, pp. 137–146.

Beck, Joseph E and Xiaolu Xiong (2013). "Limits to Accuracy: How Well Can We Do at Student Modeling?" In: *Educational Data Mining*.

Berlyne, Daniel E. (1960). *Conflict, arousal, and curiosity.* McGraw-Hill Book Company.

Bjorklund, David F. and Katherine K. Harnishfeger (1995). "The evolution of inhibition mechanisms and their role in human cognition and behavior." In:

Bloom, Benjamin S. (1968). "Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1." In: *Evaluation comment* 1.2, n2.

Bruner, Jerome S. (1965). "The process of education". In: *The Physics Teacher* 3, pp. 369–370.

Brunskill, Emma and Stuart Russell (2010). "RAPID: A reachable anytime planner for imprecisely-sensed domains". In: *UAI*.

Bubeck, Sébastien and Nicolò Cesa-Bianchi (2012). "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems". In: *Foundations and Trends® in Stochastic Systems* 1.4.

Bunderson, C. Victor and Margaret Martinez (2000). "Building interactive World Wide Web (Web) learning environments to match and support individual learning differences". In: *Journal of Interactive Learning Research* 11.2, pp. 163–195.

Burton, John K., David M. Mike Moore, and Susan G. Magliaro (2013). "Behaviorism and instructional technology". In: *Handbook of research on educational communications and technology*. Routledge, pp. 15–48.

Carstensen, Laura L., Helene H. Fung, and Susan T. Charles (2003). "Socioemotional selectivity theory and the regulation of emotion in the second half of life". In: *Motivation and emotion* 27.2, pp. 103–123.

Cesa-Bianchi, Nicolò, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth (1997). "How to use expert advice". In: *Journal of the ACM (JACM)* 44.3, pp. 427–485.

Chentanez, Nuttapong, Andrew G. Barto, and Satinder Singh (2005). "Intrinsically motivated reinforcement learning". In: *Advances in neural information processing systems*, pp. 1281–1288.

Chi, Min, Kurt VanLehn, Diane Litman, and Pamela Jordan (2011). "Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies". In: *User Modeling and User-Adapted Interaction* 21.1-2, pp. 137–180.

Chomsky, Noam (1959). *Verbal behavior*.

Clément, Benjamin, Pierre-Yves Oudeyer, and Manuel Lopes (2016). "A Comparison of Automatic Teaching Strategies for Heterogeneous Student Populations". In: *EDM 16 - 9th International Conference on Educational Data Mining*. Proceedings of the 9th International Conference on Educational Data Mining. Raleigh, United States. URL: https://hal.inria.fr/hal-01360338.

Clément, Benjamin, Didier Roy, Manuel Lopes, and Pierre-Yves Oudeyer (2014). "Online Optimization and Personalization of Teaching Sequences". In: *DI : Digital Intelligence - 1st International conference on digital cultures*. Nantes, France. URL: https://hal.inria.fr/hal-01061211.

Clément, Benjamin, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2014a). "Developmental Learning for Intelligent Tutoring Systems". In: *IEEE ICDL-Epirob - The Fourth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*. Genoa, Italy. URL: https://hal.inria.fr/hal-01061195.

– (2014b). "Online Optimization of Teaching Sequences with Multi-Armed Bandits". In: *7th International Conference on Educational Data Mining*. London, United Kingdom. URL: https://hal.inria.fr/hal-01016428.

– (2014c). *Optimisation et Personnalisation automatiques des parcours d'apprentissage dans les Systèmes Tutoriels Intelligents*. TICE. Poster. URL: https://hal.inria.fr/hal-01090900.

– (2015). "Multi-Armed Bandits for Intelligent Tutoring Systems". In: *Journal of Educational Data Mining (JEDM)* 7.2, pp. 20–48. URL: https://hal.inria.fr/hal-00913669.

Collins, Allan and Richard Halverson (2018). *Rethinking education in the age of technology: The digital revolution and schooling in America*. Teachers College Press.

Conati, Cristina, Abigail Gertner, and Kurt Vanlehn (2002). "Using Bayesian networks to manage uncertainty in student modeling". In: *User modeling and user-adapted interaction* 12.4, pp. 371–417.

Corbett, Albert T. and John R. Anderson (1994). "Knowledge tracing: Modeling the acquisition of procedural knowledge". In: *User modeling and user-adapted interaction* 4.4, pp. 253–278.

Cordova, Diana I. and Mark R. Lepper (1996). "Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice." In: *Journal of educational psychology* 88.4, p. 715.

Crowley, Rebecca S. and Olga Medvedeva (2006). "An intelligent tutoring system for visual classification problem solving". In: *Artificial intelligence in medicine* 36.1, pp. 85–117.

Csikszentmihalyi, Mihaly (1991). *Flow: The Psychology of Optimal Experience*. Harper Perennial.

– (1997). *Finding flow: The psychology of engagement with everyday life.* New York: BasicBooks.

Csikszentmihalyi, Mihaly and Isabella Csikszentmihalyi (1975). *Beyond boredom and anxiety*. Vol. 721. Jossey-Bass San Francisco.

Deci, Edward L. and Richard M. Ryan (1985). "The general causality orientations scale: Self-determination in personality". In: *Journal of research in personality* 19.2, pp. 109–134.

Deci, Edward L., Robert J. Vallerand, Luc G. Pelletier, and Richard M. Ryan (1991). "Motivation and education: The self-determination perspective". In: *Educational psychologist* 26.3-4, pp. 325–346.

Delmas, Alexandra (2018). "Conception et validation d'un jeu d'auto-apprentissage de connaissances sur l'asthme pour le jeune enfant: Rôle de la motivation intrinsèque". PhD thesis. Ecole Doctorale Sociétés, Politique, Santé Publique de Bordeaux.

Delmas, Alexandra, Benjamin Clément, Pierre-Yves Oudeyer, and Hélène Sauzeon (2018). "Fostering health education with a serious game in children with asthma: pilot studies for assessing learning efficacy and automatized learning personalization". In: *Frontiers in Education*. DOI: 10.3389/feduc.2018.00099.

Dember, William N. and Robert W. Earl (1957). "Analysis of exploratory, manipulatory, and curiosity behaviors." In: *Psychological review* 64.2, p. 91.

Desmarais, Michel C. (2011). "Performance comparison of item-to-item skills models with the IRT single latent trait model". In: *User Modeling, Adaption and Personalization*. Springer, pp. 75–86.

Desrochers, Alain, Gilles Comeau, Nisreen Jardaneh, and Isabelle Green-Demers (2006). "L'élaboration d'une échelle pour mesurer la motivation chez les jeunes élèves en piano". In: *Recherche en éducation musicale* 24, pp. 13–33.

Dewey, John (1910). "How we think. Boston". In: *MA: DC Heath*.

Dhanani, Asif, Seung Yeon Lee, Phitchaya Phothilimthana, and Zachary Pardos (2014). "A Comparison of Error Metrics for Learning Model Parameters in Bayesian Knowledge Tracing". In: *Inter. Conf. on Educational Data Mining Workshops*.

Duffy, Thomas M. and David H. Jonassen (2013). *Constructivism and the technology of instruction: A conversation*. Routledge.

Dunlosky, John, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, and Daniel T. Willingham (2013). "Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology". In: *Psychological Science in the Public Interest* 14.1, pp. 4–58.

Dweck, Carol S. and Ellen L. Leggett (1988). "A social-cognitive approach to motivation and personality." In: *Psychological review* 95.2, p. 256.

Elo, Arpad E. (1978). *The rating of chessplayers, past and present*. Arco Pub.

Engeser, Stefan and Falko Rheinberg (2008). "Flow, performance and moderators of challenge-skill balance". In: *Motivation and Emotion* 32.3, pp. 158–172.

Fosnot, Catherine Twomey and Randall Stewart Perry (1996). "Constructivism: A psychological theory of learning". In: *Constructivism: Theory, perspectives, and practice* 2, pp. 8–33.

Freeman, Scott, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth (2014). "Active learning increases student performance in science, engineering, and mathematics". In: *Proceedings of the National Academy of Sciences* 111.23, pp. 8410–8415.

Frenoy, Rémy (2016). "Modélisation de l'activité gestuelle et sélection automatique de feedback pour des environnements interactifs d'apprentissage: application à la calligraphie". PhD thesis. Compiègne.

Frith, Chris D. and Uta Frith (2012). "Mechanisms of social cognition". In: *Annual review of psychology* 63, pp. 287–313.

Froebel, Friedrich (1885). *The education of man*. A. Lovell & Company.

Gagne, Robert M. and Leslie J. Briggs (1974). *Principles of instructional design.* Holt, Rinehart & Winston.

Gardner, Howard (1993). "Histoire de la révolution cognitive". In:

Gertner, Abigail S., Cristina Conati, and Kurt VanLehn (1998). "Procedural help in Andes: Generating hints using a Bayesian network student model". In: *AAAI/IAAI* 1998, pp. 106–11.

González-Brenes, J.P., Yun Huang, and Peter Brusilovsky (2014). "General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge". In: *Inter. Conf. on Educational Data Mining*.

González-Brenes, J.P. and Jack Mostow (2012). "Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models." In: *EDM*, pp. 49–56.

Gottlieb, Jacqueline, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes (2013). "Information-seeking, curiosity, and attention: computational and neural mechanisms". In: *Trends in Cognitive Sciences* 17.11, pp. 585–593. doi: 10.1016/j.tics.2013.09.001.

Habgood, M.P. Jacob and Shaaron E. Ainsworth (2011). "Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games". In: *The Journal of the Learning Sciences* 20.2, pp. 169–206.

Hambleton, Ronald K. (1991). *Fundamentals of item response theory*. Vol. 2. Sage publications.

Harlow, Harry F. (1950). "Learning and satiation of response in intrinsically motivated complex puzzle performance by monkeys." In: *Journal of comparative and physiological psychology* 43.4, p. 289.

Huertas, Carlos and Reyes Juárez-Ramirez (2013). "Developing an intelligent tutoring system for vehicle dynamics". In: *Procedia-Social and Behavioral Sciences* 106, pp. 838–847.

Hull, Clark Leonard (1943). "Principles of behavior: An introduction to behavior theory." In:

Hunt, JMcVLevine (1965). "Intrinsic motivation and its role in psychological development". In: *Nebraska symposium on motivation*. Vol. 13. University of Nebraska Press, pp. 189–282.

Kang, Min Jeong, Ming Hsu, Ian M. Krajbich, George Loewenstein, Samuel M. McClure, Joseph Tao-yi Wang, and Colin F. Camerer (2009). "The wick in the candle of learning: Epistemic curiosity activates reward circuitry and enhances memory". In: *Psychological Science* 20.8, pp. 963–973.

Kaplan, Frederic and Pierre-Yves Oudeyer (2007). "In search of the neural circuits of intrinsic motivation". In: *Frontiers in Neuroscience* 1.1, p. 225.

Kelly, Mathew (2004). *The rhythm of life: Living every day with passion and purpose*. Simon and Schuster.

Koedinger, Kenneth R. and John R. Anderson (1993). "Effective Use of Intelligent Software in High School Math". In: *Proceedings of AI-ED 93, World conference of Artificial Intelligence in Education*, pp. 241–248.

Koedinger, Kenneth R., John R. Anderson, William H. Hadley, and Mary A. Mark (1997). "Intelligent tutoring goes to school in the big city". In:

Koedinger, Kenneth R., Emma Brunskill, Ryan S. Jd Baker, Elizabeth A. McLaughlin, and John Stamper (2013). "New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization". In: *AI Magazine*.

Krzywinski, Martin, Jacqueline Schein, İnanç Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J. Jones, and Marco A. Marra (2009). "Circos: an information aesthetic for comparative genomics". In: *Genome research* 19.9, pp. 1639–1645.

Lazar, Sandra M (1999). "The Quality of School Life Scale as a predictive indicator of student disengagement from school". In:

Lee, Carol D (2005). "Signifying in the zone of proximal development". In: *An introduction to Vygotsky* 2, pp. 253–284.

Lee, J.I. and Emma Brunskill (2012). "The Impact on Individualizing Student Models on Necessary Practice Opportunities". In: *Inter. Conf. on Educational Data Mining (EDM)*.

Leotti, Lauren A. and Mauricio R. Delgado (2011). "The inherent reward of choice". In: *Psychological science* 22.10, pp. 1310–1318.

Long, Yanjin and Vincent Aleven (2017). "Enhancing learning outcomes through self-regulated learning support with an Open Learner Model". In: *User Modeling and User-Adapted Interaction* 27.1, pp. 55–88.

Lopes, Manuel, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer (2012). "Exploration in model-based reinforcement learning by empirically estimating learning progress". In: *Advances in Neural Information Processing Systems*, pp. 206–214.

Lopes, Manuel and Pierre-Yves Oudeyer (2012). "The strategic student approach for life-long exploration and learning". In: *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, pp. 1–8.

Luckin, Rosemary (2001). "Designing Children's Software to Ensure Productive Interactivity through Collaboration in the Zone of Proximal Development (ZPD)". In: *Information Technology in Childhood Education Annual* 2001.1, pp. 57–85. ISSN: 1522-8185. URL: http://www.editlib.org/p/8501.

Mahmoud, Mona H. and Sanaa H. Abo El-Hamayed (2016). "An intelligent tutoring system for teaching the grammar of the Arabic language". In: *Journal of Electrical Systems and Information Technology* 3.2, pp. 282–294.

Malone, Thomas W (1980). "What makes things fun to learn? A study of intrinsically motivating computer games." In:

Malone, Thomas W. and Mark R. Lepper (1987). "Making learning fun: A taxonomy of intrinsic motivations for learning". In: *Aptitude, learning and instruction* 3, pp. 223–253.

Mandler, George (2002). "Origins of the cognitive (r) evolution". In: *Journal of the History of the Behavioral Sciences* 38.4, pp. 339–353.

Mergel, Brenda (1998). *Instructional design and learning theory*.

Milne, Sue, Jean Cook, Edward Shiu, and Angus McFadyen (1997). "Adapting to learner attributes: Experiments using an adaptive tutoring system". In: *Educational psychology* 17.1-2, pp. 141–155.

Monterrat, Baptiste, Amel Yessad, François Bouchet, Elise Lavoué, and Vanda Luengo (2017). "MAGAM : Un modèle générique pour l'adaptation multi-aspects dans les EIAH". In: *Environnements Informatiques pour l'Apprentissage Humain*. EIAH 2017. Strasbourg, France, pp. 29–40. URL: https://hal.archives-ouvertes.fr/hal-01517137.

Montessori, Maria (1948/2004). *The discovery of the child*. Aakar books, Delhi.

Montgomery, Kay C. (1954). "The role of the exploratory drive in learning." In: *Journal of Comparative and Physiological Psychology* 47.1, p. 60.

Mu, Tong, Karan Goel, and Emma Brunskill (2017). "Program2Tutor: Combining Automatic Curriculum Generation with Multi-Armed Bandits for Intelligent Tutoring Systems". In:

Murayama, Kou, Madoka Matsumoto, Keise Izuma, Ayaka Sugiura, Richard M. Ryan, Edward L. Deci, and Kenji Matsumoto (2013). "How self-determined choice facilitates performance: A key role of the ventromedial prefrontal cortex". In: *Cerebral Cortex* 25.5, pp. 1241–1251.

Murchland, Sonya, Jocelyn Kernot, and Helen Parkyn (2011). "Children's satisfaction with assistive technology solutions for schoolwork using the QUEST 2.1: Children's Version". In: *Assistive Technology* 23.3, pp. 162–176.

Murray, Tom and Ivon Arroyo (2002). "Toward measuring and maintaining the zone of proximal development in adaptive instructional systems". In: *International Conference on Intelligent Tutoring Systems*. Springer, pp. 749–758.

Naik, Vandana and Venkatesh Kamat (2015). "Adaptive and gamified learning environment (AGLE)". In: *Technology for Education (T4E), 2015 IEEE Seventh International Conference on*. IEEE, pp. 7–14.

Newell, Allen, Herbert Alexander Simon, et al. (1972). *Human problem solving*. Vol. 104. 9. Prentice-Hall Englewood Cliffs, NJ.

Nkambou, Roger, Riichiro Mizoguchi, and Jacqueline Bourdeau (2010). *Advances in intelligent tutoring systems*. Vol. 308. Springer.

Obukhova, L.F. and I.A. Korepanova (2009). "The zone of proximal development: A spatiotemporal model". In: *Journal of Russian & East European Psychology* 47.6, pp. 25–47.

Oudeyer, Pierre-Yves, Jacqueline Gottlieb, and Manuel Lopes (2016). "Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies". In: *Progress in brain research*. Vol. 229. Elsevier, pp. 257–284.

Oudeyer, Pierre-Yves and Frederic Kaplan (2007). "What is intrinsic motivation? a typology of computational approaches". In: *Frontiers in neurorobotics* 1.

Oudeyer, Pierre-Yves, Frederic Kaplan, and Verena V. Hafner (2007). "Intrinsic motivation systems for autonomous mental devel-

opment”. In: *IEEE transactions on evolutionary computation* 11.2, pp. 265–286.
Oudeyer, Pierre-Yves and Linda B. Smith (2016). “How evolution may work through curiosity-driven developmental process”. In: *Topics in Cognitive Science* 8.2, pp. 492–502.
Pavlov, Ivan Petrovich (1927). “Conditional reflexes: an investigation of the physiological activity of the cerebral cortex.” In:
Piaget, Jean (1937). “La construction du réel chez l’enfant.” In:
Premack, David and Guy Woodruff (1978). “Does the Chimpanzee Have a Theory of Mind?” In: *Behavioral and brain sciences* 1.4, pp. 515–526.
Rafferty, Anna N., Emma Brunskill, Thomas L. Griffiths, and Patrick Shafto (2011). “Faster Teaching by POMDP Planning.” In: *AIED*. Springer, pp. 280–287.
Ray, Roger D. and Noelle Belden (2007). “Teaching college level content and reading comprehension skills simultaneously via an artificially intelligent adaptive computerized instructional system”. In: *The Psychological Record* 57.2, pp. 201–218.
Read, Janet C. and Stuart J. MacFarlane (2006). “Using the Fun Toolkit and Other Survey Methods to Gather Opinions in Child Computer Interaction”. In: *Proceedings of the 2006 Conference on Interaction Design and Children*. IDC ’06. Tampere, Finland: ACM, pp. 81–88. DOI: 10.1145/1139073.1139096. URL: http://doi.acm.org/10.1145/1139073.1139096.
Read, Janet C., Stuart J. MacFarlane, and Chris Casey (2002). “Endurability, engagement and expectations: Measuring children’s fun”. In: *Interaction design and children*. Vol. 2. Shaker Publishing Eindhoven, pp. 1–23.
Roll, Ido, Vincent Aleven, Bruce M. McLaren, and Kenneth R. Koedinger (2011). “Improving students’ help-seeking skills using metacognitive feedback in an intelligent tutoring system”. In: *Learning and Instruction* 21.2, pp. 267–280.
Ropelato, Sandro, Fabio Zünd, Stéphane Magnenat, Marino Menozzi, and Robert Sumner (2017). “Adaptive Tutoring on a Virtual Reality Driving Simulator”. In: *1st Workshop on Artificial Intelligence Meets Virtual and Augmented Worlds (AIVRAR) in conjunction with SIGGRAPH Asia 2017*. ETH Zurich.
Rose, David (2000). “Universal design for learning”. In: *Journal of Special Education Technology* 15.3, pp. 45–49.
Roy, Didier (2012). “Usage d’un robot pour la remédiation en mathématiques”. Français. MA thesis. Université de Bordeaux. URL: http://hal.archives-ouvertes.fr/hal-00921444.
Ryan, Richard M. and Edward L. Deci (2000). “Intrinsic and extrinsic motivations: Classic definitions and new directions”. In: *Contemporary educational psychology* 25.1, pp. 54–67.
Schmidhuber, Jürgen (1991). “Curious model-building control systems”. In: *Neural Networks, 1991. 1991 IEEE International Joint Conference on*. IEEE, pp. 1458–1463.

Semet, Y., Y. Yamont, R. Biojout, E. Luton, and P. Collet (2003). "Artificial ant colonies and e-learning: An optimisation of pedagogical paths". In: *International Conference on Human-Computer Interaction*.

Shute, Valerie J. (2011). "Stealth assessment in computer-based games to support learning". In: *Computer games and instruction* 55.2, pp. 503–524.

Shute, Valerie J., Eric G. Hansen, and Russell G. Almond (2008). "You Can't Fatten A Hog by Weighing It–Or Can You? Evaluating an Assessment for Learning System Called ACED". In: *International Journal of Artificial Intelligence in Education* 18.4, pp. 289–316.

Skinner, Burrhus Frederic (1974). *About behaviorism*. Alfred A. Knopf; Book Club Edition edition.

Spaan, Matthijs T. J. and Nikos Vlassis (2005). "Perseus: Randomized Point-based Value Iteration for POMDPs". In: *Journal of Artificial Intelligence Research* 24, pp. 195–220.

Stade, Cory Marie (2017). "Lithic morphological variability as a proxy for Palaeolithic linguistic ability: a knapping training study exploring cultural transmission, theory of mind and language". PhD thesis. University of Southampton.

Stafford, Tom, Martin Thirkettle, Tom Walton, Nicolas Vautrelle, Len Hetherington, Michael Port, Kevin Gurney, and Pete Redgrave (2012). "A novel task for the investigation of action acquisition". In: *PloS one* 7.6, e37749.

Stahl, Aimee E and Lisa Feigenson (2015). "Observing the unexpected enhances infants' learning and exploration". In: *Science* 348.6230, pp. 91–94.

Steinberg, Esther R. (1989). "Cognition and learner control: A literature review, 1977–1988." In: *Journal of computer-based instruction*.

Sun, Shanghua, Mike Joy, and Nathan Griffiths (2007). "The use of learning objects and learning styles in a multi-agent education system". In: *Journal of Interactive Learning Research* 18.3, pp. 381–398.

Sweller, John (1994). "Cognitive load theory, learning difficulty, and instructional design". In: *Learning and instruction* 4.4, pp. 295–312.

Vaessen, Bram E, Frans J. Prins, and Johan Jeuring (2014). "University students' achievement goals and help-seeking strategies in an intelligent tutoring system". In: *Computers & Education* 72, pp. 196–208.

Vallerand, Robert J., Marc R. Blais, Nathalie M. Brière, and Luc G. Pelletier (1989). "Construction et validation de l'échelle de motivation en éducation (EME)." In: *Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement* 21.3, p. 323.

Vallerand, Robert J., Luc G. Pelletier, Marc R. Blais, Nathalie M. Briere, Caroline Senecal, and Evelyne F. Vallieres (1992). "The Academic Motivation Scale: A measure of intrinsic, extrinsic, and amotivation in education". In: *Educational and psychological measurement* 52.4, pp. 1003–1017.

Vandewaetere, Mieke, Piet Desmet, and Geraldine Clarebout (2011). "The contribution of learner characteristics in the development of computer-based adaptive learning environments". In: *Computers in Human Behavior* 27.1, pp. 118–130.

Vygotsky, Lev Semenovich (1930-1934/1978). *Mind in society: The development of higher psychological processes*. Harvard university press.

Walkington, Candace A (2013). "Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes." In: *Journal of Educational Psychology* 105.4, p. 932.

Wang, Yutao and Neil Heffernan (2013). "Extending knowledge tracing to allow partial credit: using continuous versus binary nodes". In: *Artificial Intelligence in Education*. Springer, pp. 181–188.

Wass, Rob and Clinton Golding (2014). "Sharpening a tool for teaching: the zone of proximal development". In: *Teaching in Higher Education* 19.6, pp. 671–684.

Watson, John Broadus (1930). "Behaviorism, rev". In:

Weber, Keith, Mathew M. Martin, and Jacob L. Cayanus (2005). "Student interest: A two-study re-examination of the concept". In: *Communication Quarterly* 53.1, pp. 71–86.

White, Robert W. (1959). "Motivation reconsidered: The concept of competence." In: *Psychological review* 66.5, p. 297.

Wood, David and Heather Wood (1996). "Vygotsky, tutoring and learning". In: *Oxford review of Education* 22.1, pp. 5–16.

Woon, Marie (2017). "Combining Intelligent Tutoring Systems and Automatic Story Generation for Personalized Tutoring with Mobile Robots". In:

Appendix A
Study 2 configuration (Chap. 5)

A.1 RiARiT configuration

The following tables (provide all the parameters during the user studies when using the algorithm RiARiT.

Table A.1: This table shows another type of pedagogical restrictions that are enforced into the RiARiT algorithm in study 2. A given exercise parameter can be deactivated if the pre-condition is achieved, usually in the form of maximum skill levels for one or many KC.

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
M difficulty	1	0	0.6	0	0	0	0	0
	2	0	0	0	0.7	0	0	0
	3	0	0	0	0	0	0	0.8
	4	0	0	0	0	0	0	0.7
	5	0	0	0	0	0	0	0.8
	6	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
R difficulty	1	0	0	0	0.7	0	0	0
	2	0	0	0	0	0.7	0	0
	3	0	0	0	0	0	0	0.8
	4	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
MM difficulty	1	0	0.8	0	0	0	0	0
	2	0	0.9	0	0	0	0	0
	3	0	0	0	0	0	0.9	0
	4	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
RM difficulty	1	0	0	0.8	0	0	0	0
	2	0	0	0	0	0	0.8	0
	3	0	0	0	0	0	0.9	0
	4	1	1	1	1	1	1	1

Table A.2: R Tables that was used in the study 2 for algorithm RiARiT. It shows the relation of the parameters values and the minimum required competence level, for each KC, to be able to solve that exercise.

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
Exercise Type	M	1	0.5	0.3	0.7	0.3	0.2	0.7
	R	1	0.5	0.8	0.7	0.3	0.7	0.7
	MM	1	1	0.4	1	1	0.3	1
	RM	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
M difficulty	1	0.3	0.2	0	0	0	0	0
	2	0.5	0.5	0.3	0.5	0	0	0
	3	0.5	0.6	0.5	0.7	0	0	0
	4	0.7	0.4	0	0	0	0	0.3
	5	0.9	0.8	0.7	0.7	0.5	0.6	0.6
	6	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
R difficulty	1	0.3	0.6	0.4	0.6	0	0	0
	2	0.5	1	0.7	1	0	0	0
	3	0.8	0.8	0.9	0.8	0.5	0.5	0.5
	4	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
MM difficulty	1	0.5	0.6	1	1	0	0	0
	2	0.5	0.7	1	1	0	0	0
	3	0.8	1	1	1	0.7	1	0.8
	4	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
RM difficulty	1	0.5	0.6	0.7	1	0	0	0
	2	0.5	0.7	0.7	1	0	0	0
	3	0.8	1	0.8	1	0.7	0.7	0.7
	4	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
M/R modality	x€x	0.8	1	1	1	0.6	1	0.7
	x.x€	1	1	1	1	1	1	1

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
Remainder	No	1	0.7	1	1	0.7	1	1
	Unit	1	1	1	1	0.7	1	1
	Decimal	1	1	1	1	1	1	1
Money Type	Real	1	1	1	1	1	1	0.8
	Token	0.9	1	1	1	1	1	1

Table A.3: This table shows the pedagogical restrictions that are enforced when using the RiARiT algorithm in study 2. A given exercise parameter can only be used if the pre-condition is achieved, usually in the form of a minimum skill level for a given KC.

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
Exercise Type	M	0	0	0	0	0	0	0
	R	0.3	0.2	0	0.3	0	0	0
	MM	0.3	0.3	0.3	0.3	0	0	0
	RM	0.3	0.5	0.3	0.3	0	0	0

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
M difficulty	1	0	0	0	0	0	0	0
	2	0.1	0.1	0	0	0	0	0
	3	0	0	0	0.3	0	0	0
	4	0.3	0.3	0.2	0	0	0	0
	5	0	0	0	0	0	0	0.1
	6	0	0	0	0	0	0	0.4

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
R difficulty	1	0	0	0	0	0	0	0
	2	0	0	0.3	0	0	0	0
	3	0.4	0	0.5	0	0	0	0
	4	0.6	0	0.6	0	0	0.3	0.3

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
MM difficulty	1	0	0	0	0	0	0	0
	2	0	0.3	0	0	0	0	0
	3	0	0.4	0	0	0	0	0
	4	0	0	0	0	0	0	0.5

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
RM difficulty	1	0	0	0	0	0	0	0
	2	0	0	0.4	0	0	0	0
	3	0	0.6	0	0	0	0	0
	4	0	0.7	0	0	0	0.4	0

		KnowMoney	IntSum	IntSub	IntDec	DecSum	DecSub	DecDec
Remainder	No	0	0	0	0	0	0	0
	Unit	0	0.4	0	0	0	0	0
	Decimal	0	0	0	0	0	0	0

A.2 ExpSeq configurations

In order to evaluate our algorithm, we use as baseline an optimized sequence created based on instructional design theory, whose reliability has been validated through several user studies, see Roy, 2012. This sequence has the following characteristics:

- there is 5 groups for a total of 28 exercises:
  - M exercise with integer price : 3 exercises
  - M exercise with decimal price : 4 exercises
  - R exercise with one object : 5 exercises
  - MM exercise : 8 exercises
  - RM exercise : 8 exercises
- propose 4 times the same parameterized exercise :
  - after 3 correct answers, pass to the next group of exercises. If it's the last exercise group, change the exercise group
  - else change directly of exercise group to work on another type of exercise
  - when changing groups, begin from the highest exercise succeed

Table A.2 summarizes the 28 stages of progression for the students. Following the parameters previously defined, the table also shows how the different parameters evolve.

Table A.4: Expert sequence including 28 different stages of different parameters for the proposed activities to configure ExpSeq in study 2.

	G1.1	G1.2	G1.3	G2.1	G2.2	G2.3	G2.4	G3.1	G3.2	G3.3	G3.4	G3.5
Ex Type	M	M	M	M	M	M	M	R	R	R	R	R
Difficulty	1	2	3	4	5	5	6	1	2	3	3	4
Cents Not	-	-	-	x€x	x€x	x,x€	x,x€	-	-	x€x	x€x	x,x€

	G4.1	G4.2	G4.3	G4.4	G4.5	G4.6	G4.7	G4.8
Ex Type	MM	MM	MM	MM	MM	MM	MM	MM
Difficulty	1	1	2	2	3	3	4	4
Remainder	-	Int	-	Int	-	Int	-	Dec
Money Type	Real	Real	Real	Real	Real	Real	Real	Token

	G5.1	G5.2	G5.3	G5.4	G5.5	G5.6	G5.7	G5.8
Ex Type	RM	RM	RM	RM	RM	RM	RM	RM
Difficulty	1	1	2	2	3	3	4	4
Remainder	-	Int	-	Int	-	Int	-	Dec
Money Type	Real	Real	Real	Real	Real	Real	Real	Token

Appendix B
Study 3 metrics and configuration (Chap. 6)

B.1 Activities Restrictions

Here are presented the activity restrictions ZPDES and ExpSeq in study 3 presented in Chap. 6.

To be able to access MM and R activities, the students need 50% success rate on M3 over the last four steps.

To be able to access MM3 and R3 activities, the students need 50% success rate on M5 over the last four steps.

To be able to access RM activities, the students need 50% success rate on MM2 and R2 over the last four steps.

To be able to access RM3 activities, the students need 50% success rate on MM4 and R4 over the last four steps.

B.2 Questionnaires

Profile questionnaire

These elements are presented in Section 6.2.1.1.

- Informations provided by the teacher about each student:

  1. Student birth date
  2. Numeracy level and Literacy level
     (a) Does not master the notions at all (b) Does not master the notions enough
     (c) Good mastery of the notions (d) Very good mastery of the notions

- Questionnaire given to the students:

  1. (About his/her gender)
     (a) Je suis un garçon (b) Je suis une fille
  2. (About use of money)
     (a) J'ai déjà utilisé de la monnaie et j'aime le faire
     (b) J'ai déjà utilisé de la monnaie mais je n'aime pas le faire
     (c) Je n'ai jamais utilisé de la monnaie et j'aimerai apprendre à le faire
     (d) Je n'ai jamais utilisé de la monnaie mais je n'ai pas envie d'apprendre à le faire
  3. Aimes-tu faire des calculs ?
     (a) Non (b) Oui
  4. Quels sont les écrans que tu utilises à la maison ?
     (a) Un ordinateur (b) Une tablette (c) Un smartphone
     (d) Une console portable (e) Une console de salon
  5. À quelle fréquence utilises-tu ces écrans ?
     (a) Jamais (b) Uniquement le week-end (c) Quelques fois dans la semaine (d) Tous les jours
  6. Chez moi, je choisis les activités que je fais
     (a) Pas du tout (b) Pas souvent (c) Des fois oui, des fois non (d) Souvent (e) Tout le temps
  7. En général, je choisis la nourriture que je mange
     (a) Pas du tout (b) Pas souvent (c) Des fois oui, des fois non (d) Souvent (e) Tout le temps
  8. Je choisis qui sont mes copains et mes copines
     (a) Pas du tout (b) Pas souvent (c) Des fois oui, des fois non (d) Souvent (e) Tout le temps
  9. C'est moi qui choisis les vêtements que je porte
     (a) Pas du tout (b) Pas souvent (c) Des fois oui, des fois non (d) Souvent (e) Tout le temps

School Profile questionnaire (SP)

This questionnaire is presented in Section 6.2.1.1.

1. Je fais des choses intéressantes en classe
   (a) Jamais (b) Presque jamais (c) Quelques fois (d) Souvent (e) Tout le temps

2. À l'école je m'entends bien avec...
   (a) Personne (b) Une ou deux personnes (c) Moins de la moitié (d) Plus de la moitié (e) Tout le monde

3. J'aime bien mon maître/ma maîtresse
   (a) Pas du tout (b) Pas beaucoup (c) Un peu (d) Beaucoup (e) Vraiment beaucoup

4. À l'école je m'ennuie...
   (a) Tout le temps (b) Souvent (c) Quelques fois (d) Presque jamais (e) Jamais

5. Si l'école était une personne, elle serait...
   (a) Mon ennemi(e) (b) Un inconnu (c) Quelqu'un que je connais (d) Un copain / une copine (e) Mon meilleur ami / ma meilleure amie

6. Quand tu reçois tes évaluations, tu es...
   (a) Très triste (b) Triste (c) Comme d'habitude (d) Content(e) (e) Très content(e)

7. En classe, tu es. . . (créé)
   (a) Très triste (b) Triste (c) Comme d'habitude (d) Content(e) (e) Très content(e)

8. Quand tu fais tes devoirs à la maison, tu es...
   (a) Très triste (b) Triste (c) Comme d'habitude (d) Content(e) (e) Très content(e)

9. À la récréation, tu es...
   (a) Très triste (b) Triste (c) Comme d'habitude (d) Content(e) (e) Très content(e)

10. Quand tes copains et copines parlent de toi, tu es...
    (a) Très triste (b) Triste (c) Comme d'habitude (d) Content(e) (e) Très content(e)

Intrinsic Motivation questionnaire (IM)

This questionnaire is presented in Section 6.2.1.2.

1. Je pense que j'ai appris de nouvelles choses en jouant au jeu (ajout)
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

2. J'ai aimé jouer sur le jeu
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

3. Je voudrais rester après l'école un jour pour jouer au jeu
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

4. Je pense que le jeu est utile pour apprendre à manipuler de la monnaie
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

5. Je conseillerais à un copain d'essayer le jeu
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

6. Je pense que j'ai été bon sur le jeu
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

7. Je voudrais que le jeu ait un niveau plus dur la prochaine fois
   (a) Pas du tout (b) Non (c) Un peu (d) Oui (e) Oui beaucoup

8. Par rapport à mon jeu préféré, j'ai trouvé que le jeu était...
   (a) Beaucoup moins cool (b) Un peu moins cool (c) Aussi cool (d) Un peu plus cool (e) Beaucoup plus cool

9. Par rapport à ma matière préférée à l'école, j'ai trouvé que le jeu était...
   (a) Beaucoup moins cool (b) Un peu moins cool (c) Aussi cool (d) Un peu plus cool (e) Beaucoup plus cool

10. Est-ce que jouer au jeu t'a demandé beaucoup d'efforts ?
    (a) Vraiment beaucoup (b) Beaucoup (c) Un peu (d) Presque pas (e) Pas du tout

Type of Motivation questionnaire (TM)

This questionnaire is presented in Section 6.2.1.2. The children can answer by yes or no to the items.

1. J'ai joué sur le jeu parce que j'aurais eu honte si je n'avais pas essayé.
2. J'ai joué sur le jeu parce que j'aime bien réussir à un jeu.
3. J'ai joué sur le jeu mais je ne sais pas pourquoi.
4. J'ai joué sur le jeu parce que ça me rend heureux quand je réponds correctement à une activité du jeu qui est compliquée.
5. J'ai joué sur le jeu pour éviter de faire un exercice que me donnerait le professeur.
6. J'ai joué sur le jeu parce que je suis content d'apprendre à rendre la monnaie.
7. J'ai joué sur le jeu parce que ça me détend quand j'y joue.
8. J'ai joué sur le jeu pour pouvoir utiliser la tablette/l'ordinateur.
9. J'ai joué sur le jeu parce que j'ai fais ce qu'on m'a dit de faire.
10. J'ai joué sur le jeu parce que je suis heureux quand j'y joue.
11. J'ai joué sur le jeu pour montrer que je sais faire des choses.
12. J'ai joué sur le jeu parce que je pense que si je sais mieux rendre la monnaie je pourrais aider un copain ou mes parents avec ça.
13. J'ai joué sur le jeu parce que je ne m'ennuie jamais quand j'y joue.
14. J'ai joué sur le jeu parce que j'aime bien apprendre de nouvelles choses.
15. J'ai joué sur le jeu parce que c'est une manière pour moi de m'améliorer en calcul.
16. Je ne sais pas pourquoi j'ai joué sur le jeu, j'ai l'impression de m'être ennuyé.
17. J'ai joué sur le jeu parce que j'apprends plein de choses qui m'intéressent.
18. J'ai joué sur le jeu parce que j'ai toujours bien réussi les jeux et je veux le prouver.
19. J'ai joué sur le jeu pour avoir un bon score au jeu.
20. J'ai joué sur le jeu parce que ça me met de bonne humeur si je réussis dans ce jeu.
21. J'ai joué sur le jeu parce que je suis content quand je sens que je deviens meilleur à un jeu.

GI questionnaire

This questionnaire is presented in Section 6.2.1.3.

1. Est-ce que c'est facile d'utiliser le jeu ?
   (a) Non, pas facile du tout (b) Non, pas facile (c) Ça va (d) Oui, facile (e) Oui, super facile

2. Est-ce que le jeu est joli ?
   (a) Non, vraiment pas joli (b) Non, pas très beau (c) Ça va (d) Oui, c'était joli (e) Oui, c'était super joli

3. Est-ce que les séances de jeu étaient longues ?
   (a) Oui, beaucoup trop longues (b) Oui, trop longues (c) Ça va (d) Non, presque pas (e) Non, beaucoup trop courtes

4. Est-ce qu'il y a eu des problèmes pendant que tu jouais (bugs, . . .) ?
   (a) Oui, vraiment beaucoup (b) Oui, beaucoup (c) Ça va (d) Non, presque pas (e) Non, jamais

5. Est-ce que tu as utilisé les indices du petit bonhomme ?
   (a) Oui, trop souvent (b) Oui, souvent (c) Oui, un peu (d) Non, presque pas (e) Non, jamais

6. Si tu as utilisé les indices, est-ce que tu les as compris ?
   (a) Non, pas du tout (b) Non, pas trop (c) Ça va (d) Oui, c'était clair (e) Oui, c'était super clair

7. Combien de temps as-tu mis à comprendre le fonctionnement du jeu la première fois ?
   (a) Beaucoup trop longtemps (b) Longtemps (c) Pas trop longtemps (d) Rapidement (e) Très rapidement

8. Coche les 3 choses que tu considères les plus importantes dans un jeu en général
   (a) C'est facile de l'utiliser (b) Il est joli (c) Il n'y a pas beaucoup de bugs (d) Les aides sont faciles à comprendre (e) On comprend vite le jeu

B.3 Detailed statistics

The statistical design was the same for each analysis presented in 6.4.2, 6.4.3 and 6.4.3. At first, the statistics which are presented in the core text were made with t-test for pairwise comparisons with a p-value threshold $\alpha = 0.05$.

To produce a more rigorous statistical analysis, we ran an additional statistical procedure. The procedure consist of three-way mixed ANOVA (group x algorithm x measure) on our two targeted dependent measures (learning score and the two motivation scores), with the measure factor as within-subject factor. The group factor includes self-determined (SD) and non self-determined students (NSD) conditions. The measure factor includes pre-test and post-tests conditions or motivational questionnaires condition. The algorithm factor includes the five experimental conditions (Expseq, ZPDES, ZP-

DRD, ZCO, ZCA). The p-value threshold is $\alpha = 0.05$. Pairwise comparisons are carried out with the Least Significant Difference (LSD) and Bonferroni procedure for corrected comparisons.

Only significant results or tendencies are reported below.

Analyse on learning scores

Three-way ANOVA revealed a significant interaction effect (SD x pre/post x algorithm) :
$[F(4, 225) = 2.446,\ p - value = 0.047,\ \eta^2 = 0.020]$

Partial two-way (algorithm x measure) ANOVAs have been conducted for each group condition for detailing the previous three-way interaction effect. They reveal a significant (algorithm x measure) interaction effect in SD children.
$[F(4, 114) = 2.579,\ p - value = 0.041,\ \eta^2 = 0.086]$

while this interaction was not significant for the NSD children.
$[F(4, 114) = 0.821,\ p - value = 0.514,\ \eta^2 = 0.030]$.

Pairwise comparison across algorithm conditions for all children and for each children group (SD or NSD) are presented in table B.1 .

	All ZCO		SD ExpSeq		NSD ExpSeq	
	LSD	Bonferroni	LSD	Bonferroni	LSD	Bonferroni
Expseq	0.346	1				
ZPDES	0.412	1	**0.011**	0.108	0.149	1
ZPDRD	0.235	1	0.067	0.673	0.249	1
ZCO			**0.005**	**0.054**	0.657	1
ZCA	0.066	0.659	0.159	1	0.172	1

Table B.1: Summary of pairewise comparisons across algorithms, for learning scores, for all children and for each children group algorithm (SD and NSD) with the LSD and Bonferroni procedures.

Pairewise comparisons based on LSD revealed that significant differences between algorithm conditions only for SD children, notably for Expseq vs ZPDES, Expseq vs ZCO. Note that only Expseq vs ZCO remained significant after Bonferroni correction.

IM questionnaire (Motivational score)

ANOVA reveals no significant effect, but pairwise comparisons exhibits very tenuous tendency regarding algorithm differences for the IM 3 questionnaire that was significant with t-test, cf table B.2.

	ZCO	
	LSD	Bonferroni
Expseq	0.103	1
ZPDES	0.100	1
ZPDRD	0.186	1
ZCA	0.198	1

Table B.2: Summary of pairewise comparisons across algorithms, for IM3 score, for all children with the LSD and Bonferroni procedures.

TM questionnaire (motivational score)

Whole population

Three-way ANOVA reveals no significant effect, although LSD pairwise comparisons reveal algorithm differences for the two motivation condition. ZCO vs ZPDES for extrinsic score while ZCO vs ZPDRD are significant for extrinsic and intrinsic motivation condition. These differences disappear with Bonferroni correction. These results fit those reported with t-test procedure.

	ZCO					
	Global		Extrinsic		Intrinsic	
	LSD	Bonferroni	LSD	Bonferroni	LSD	Bonferroni
Expseq	0.205	1	0.204	1	0.257	1
ZPDES	0.063	0.633	**0.015**	0.151	0.302	1
ZPDRD	0.071	0.709	**0.046**	0.457	**0.045**	0.450
ZCA	0.205	1	0.241	1	0.110	1

Table B.3: Summary of pairewise comparisons across algorithm algorithms, for TM scores, for all children with the LSD and Bonferroni procedures.

The two partial two-way ANOVA for each sub-group condition reveal no significant effect although post-HOC LSD comparison reveal for SD population significant differences for ZCO vs Expseq and ZCO vs ZPDES for global motivation score. Also, ZCO vs ZPDES is significant for extrinsic motivational score. These differences disappear with Bonferroni correction, cf table B.4.

For NSD population, post-HOC LSD comparisons reveal significant differences for ZPDRD vs Expseq, ZPDRD vs ZPDES and ZPDRD vs ZCO. for intrinsic motivation score. These differences disappear with Bonferroni correction, cf table B.5.

SD population

	ZCO					
	Global		Extrinsic		Intrinsic	
	LSD	Bonferroni	LSD	Bonferroni	LSD	Bonferroni
Expseq	**0.043**	0.435	0.078	0.776	0.064	0.645
ZPDES	**0.029**	0.287	**0.012**	0.124	0.357	1
ZPDRD	0.529	1	0.331	1	0.755	1
ZCA	0.209	1	0.318	1	0.099	0.990

Table B.4: Summary of pairewise comparisons across algorithms, for TM scores, for SD children with the LSD and Bonferroni procedures.

NSD population

	ZPDRD					
	Global		Extrinsic		Intrinsic	
	LSD	Bonferroni	LSD	Bonferroni	LSD	Bonferroni
Expseq	**0.040**	0.402	0.062	0.616	**0.013**	0.126
ZPDES	0.136	1	0.297	1	**0.044**	0.442
ZCO	0.054	0.535	0.056	0.557	**0.015**	0.147
ZCA	0.236	1	0.263	1	0.099	0.990

Table B.5: Summary of pairewise comparisons across algorithms, for TM scores, for NSD children with the LSD and Bonferroni procedures.

Appendix C
Kidlearn Interface

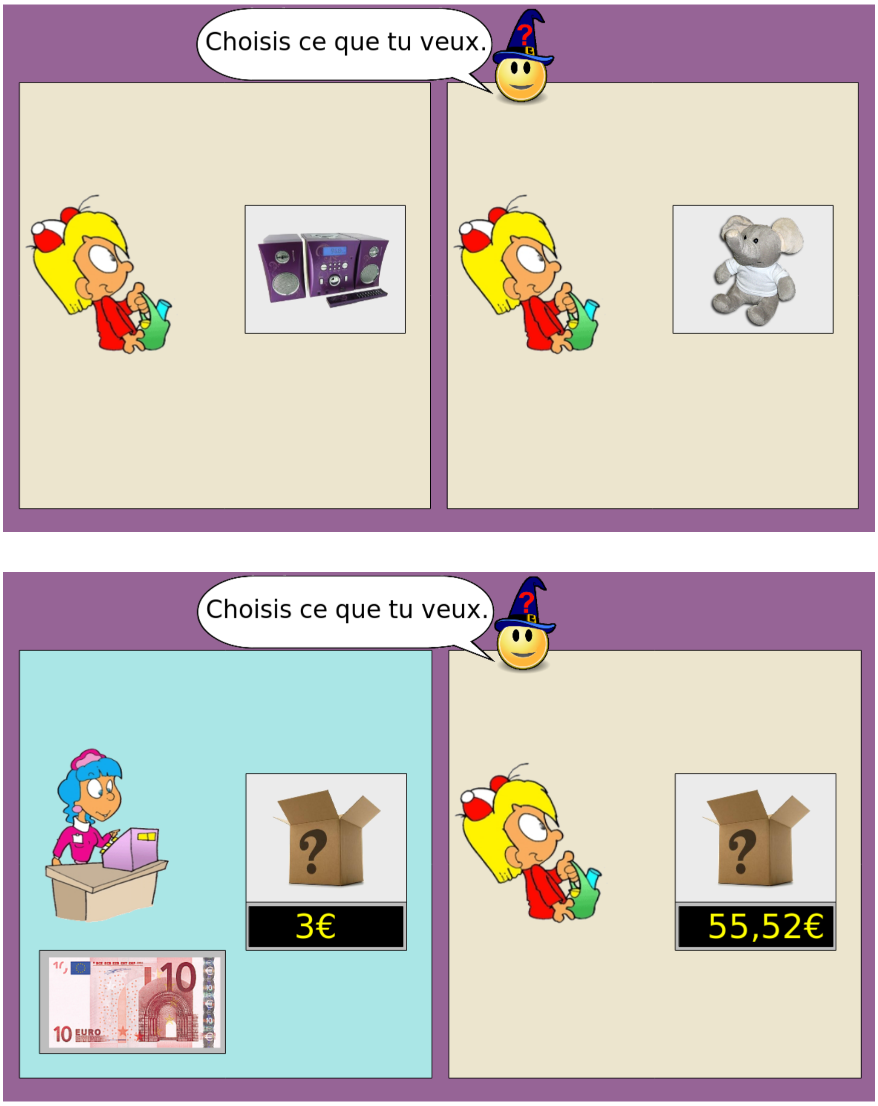


Figure C.1: Object and Activity choice interface. The student choose the object(s) he wants to train with by typing on it. The bubble indicate instruction “Choose what you want”.

Figure C.2: Four principal regions are defined in the graphical interface. The first is the wallet location where users can pick and drag the money items and drop them on the repository location to compose the correct price. The object and the price are present in the object location. Four different types of exercises exist: M : customer/one object, R : merchant/one object, MM : customer/two objects, RM : merchant/two objects.