



HAL
open science

Adaptation du Feedback dans les plateformes d'apprentissage de la programmation : quelle politique de décision ?

Badmavasan Kirouchenassamy

► **To cite this version:**

Badmavasan Kirouchenassamy. Adaptation du Feedback dans les plateformes d'apprentissage de la programmation : quelle politique de décision ?. RJC-EIAH 2024, Le Mans Université, Jun 2024, Laval, France. pp.124-134. hal-04637101

HAL Id: hal-04637101

<https://hal.science/hal-04637101v1>

Submitted on 5 Jul 2024

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

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

Adaptation du Feedback dans les plateformes d'apprentissage de la programmation : quelle politique de décision ?

Badmavasan Kirouchenassamy^[0009–0003–6502–154X], 1^{re} année de thèse

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
badmavasan.kirouchenassamy@lip6.fr

Abstract. Dans cet article, notre attention se porte sur la prise de décision relative aux feedback adaptatifs dans le cadre de l'apprentissage de la programmation en ligne. Dans un premier temps, nous nous concentrons sur l'examen de l'état actuel des modèles de feedback et des techniques de décision. Par la suite, nous positionnons notre sujet de thèse en relation avec les recherches antérieures et actuelles.

Keywords: feedback adaptatif, plateforme d'apprentissage de la programmation, politique de décision des feedbacks

1 Introduction

Des plateformes d'apprentissages sont disponibles dans des domaines variés tels que les langues, la médecine, les arts et les mathématiques. Ces plateformes informatiques visent à améliorer l'environnement de travail des étudiants et à favoriser l'apprentissage de différentes thématiques. Cependant, il est fréquent que ces plateformes ne considèrent pas les différences entre les apprenants et donc ne leur adaptent pas les parcours et les contenus fonctionnant selon le principe du “one size fits all” [4].

L'article, et la thèse associée, se focalisent spécifiquement sur les plateformes dédiées à l'apprentissage de la programmation, et plus spécifiquement AlgoPython¹. Cette plateforme résulte d'un partenariat entre un enseignant de mathématiques au lycée et la maison d'édition Génération 5².

AlgoPython est un EIAH d'apprentissage de la programmation utilisé par plus de 8000 lycéens. Actuellement, l'apprenant ne bénéficie que d'une quantité limitée de feedbacks, qui portent essentiellement sur des aspects syntaxiques et le nombre optimum de lignes de code que l'apprenant peut utiliser.

Cette plateforme propose un large éventail d'exercices de programmation en Python, structurés en parcours thématiques (une présentation plus complète de l'environnement est proposée dans [12]). Chaque section est composée de

¹ <https://www.algopython.fr/>

² <https://www.generation5.fr/>

plusieurs exercices de niveaux variés, faisant de cette plateforme un environnement d'apprentissage axé sur la résolution d'exercices. Ainsi, dans notre contexte de travail, les feedbacks font référence aux feedbacks fournis aux apprenants lors d'une activité de résolution d'exercices.

Cet article présente le début de notre travail de thèse qui a démarré en novembre 2023, dans le cadre d'un contrat CIFRE entre l'équipe MOCAH du laboratoire LIP6³ et l'entreprise Génération 5. L'objectif de cette thèse est de concevoir et d'implémenter une stratégie de décision automatique de feedbacks épistémiques, adaptés à l'apprenant, au sein de la plateforme AlgoPython.

La décision de feedback peut se définir comme le choix par un système du feedback le plus approprié à un apprenant dans un contexte donné.

Nous poursuivons trois objectifs dans cet article : (1) Expliciter le concept de feedback dans les plateformes d'apprentissage et son utilité dans l'apprentissage (**Sect. 2**) (2) fournir un aperçu des différents modèles et stratégies utilisés pour dispenser des feedbacks aux apprenants dans les plateformes d'apprentissage de la programmation et (3) identifier les apports et les limites de ces modèles et stratégies (**Sect. 2**). Ces éléments nous permettent de positionner plus précisément le travail de thèse dans la **Sect. 3**. En conclusion, nous évoquons les perspectives du travail de thèse.

2 Etat de l'art, feedback, feedbacks dans des EIAH, décision

Dans l'état de l'art relatif à l'adaptation dans les EIAH, on identifie deux types d'adaptation : en boucle longue (*outer loop* ou *task loop*) et en boucle courte (*inner loop* ou *step loop*) [1, 28, 27, 13]. L'adaptation en boucle longue concerne l'enchaînement des activités d'apprentissage (Instructional sequencing), c'est-à-dire, la décision de la prochaine activité à recommander à un apprenant alors que l'adaptation en boucle courte concerne les messages (ou plus particulièrement les feedbacks) qu'on décide d'afficher à un apprenant sur la base de ses actions au sein de l'activité en cours.

Dans le cadre de notre travail de thèse, nous nous intéressons à l'adaptation en boucle courte, c'est à dire, à celle des feedbacks affichés à l'apprenant lors de la résolution d'un exercice de programmation.

Dans la suite nous présentons un état de l'art autour de la question de l'adaptation des feedbacks. Tout d'abord nous présentons quelques éléments généraux relatifs aux feedbacks. Puis nous nous intéressons plus particulièrement à la modélisation et à la décision des feedbacks au sein de plateformes d'apprentissage.

³ <https://www.lip6.fr/recherche/team.php?acronyme=MOCAH>

2.1 Éléments généraux sur les feedbacks et leur décision dans des plateformes d'apprentissage

Selon Hattie [11], l'objectif du feedback est de mettre en évidence les points forts de l'apprenant et de fournir des commentaires sur les éléments à améliorer et à développer, dans le but de susciter une amélioration entre la performance actuelle de l'apprenant et la performance attendue. Un feedback adapté constitue un moyen efficace de stimuler l'apprentissage de l'apprenant [29].

Des feedbacks sont fréquemment dispensés dans les salles de classe ; cependant, la plupart des informations sont mal perçues, car elles sont présentées à des groupes, et souvent, les apprenants ne considèrent pas que le feedback apporté par l'enseignant est spécifiquement pertinent pour eux [10]. Ainsi, il semble nécessaire de fournir des feedbacks adaptés aux apprenants pour améliorer l'apprentissage. Le feedback adapté, contrairement au feedback générique, est dynamique, les apprenants travaillant sur des exercices où chaque apprenant recevra des informations différentes [14]. Ainsi, dans cet article et tout au long de la thèse, l'objectif est de fournir un feedback adapté à l'apprenant.

Nous distinguons la nature du feedback de son type. En effet, plusieurs natures de feedback existent, parmi lesquelles figurent les feedbacks épistémiques, métacognitifs ou motivationnels [19]. Comme mentionné dans l'introduction, l'objectif est de fournir des feedbacks épistémiques dans la plateforme Algorpython. Les feedbacks épistémiques, selon la définition proposée par [16], concernent le savoir en jeu dans l'apprentissage. Dans les feedbacks de nature épistémique, nous pouvons distinguer différents types de feedbacks suivant les informations qu'ils contiennent. Ces informations font référence aux connaissances mobilisées dans la tâche, aux erreurs, à un modèle de solution possible, à un rappel de cours, à des indices sur la prochaine action à faire, etc.

La décision d'un feedback adapté est conditionnée, a minima, par la réalisation de deux enjeux importants : le premier est de comprendre la production de l'apprenant notamment en termes d'erreurs et le deuxième est de disposer d'une modélisation de l'apprenant. La modélisation de l'apprenant implique par exemple d'avoir un système qui permet d'inférer dynamiquement (en temps réel) le niveau de maîtrise par l'apprenant des différents concepts de programmation travaillés au sein de la plateforme d'apprentissage.

La mise en oeuvre dans une plateforme d'apprentissage d'un système de feedbacks adaptatifs implique l'élaboration d'un processus allant de la modélisation des feedbacks à leur évaluation, en passant par leur production puis leur décision. Le diagramme 1 présente ainsi une vision globale du projet dont l'objectif est de doter un EIAH de feedbacks épistémiques et adaptatifs. Pour donner une vision globale du travail, nous présentons dans cette contribution des éléments relatifs à différents aspects du diagramme, cependant le cœur de la thèse est l'élaboration d'un système de décision de feedback, composant numéro 3 du diagramme 1. Nous considérons, à l'instar d'autres chercheurs, que la décision d'un feedback adapté est un enjeu majeur [2, 22, 24, 17] pour soutenir l'apprentissage des apprenants.

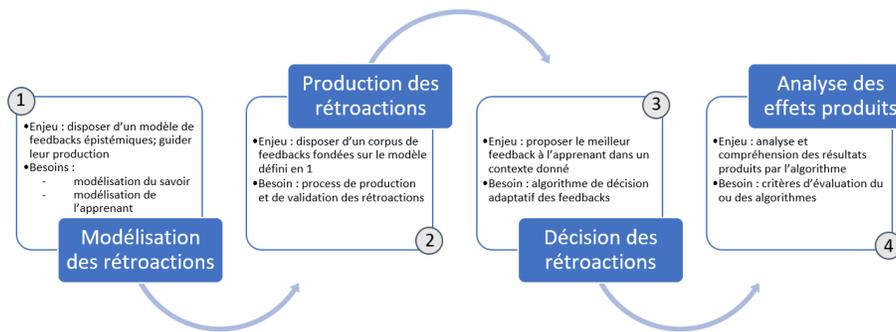


Fig. 1: Processus de mise en oeuvre de feedbacks épistémiques et adaptatifs dans une plateforme d'apprentissage

La mise en place d'un système de décision de feedback adaptatif nécessite la disponibilité d'un ensemble diversifié de feedbacks parmi lesquels il convient de sélectionner celui qui est le plus adapté à l'apprenant (référence 2 **Fig. 1**). Nous voulons explorer l'hypothèse suivante : pour une même situation (même tentative de solution), des feedbacks de types différents et présentés dans des ordres différents sont plus ou moins adaptés à des apprenants différents (référence 3 **Fig. 1**). Pour vérifier cette hypothèse, un système de décision apprend une politique de décision de feedbacks adaptatifs à partir des traces laissées par les apprenants sur la plateforme d'apprentissage de la programmation et d'autres facteurs (**Sect. 3.1**).

Selon la revue de littérature réalisée dans [5], on constate qu'il y a peu d'études et recherches sur des dispositifs proposant des feedbacks de types différents (tels que la référence aux connaissances mobilisées dans la tâche, aux erreurs, à des indices ou à la prochaine action à faire) et encore moins ordonnés différemment au sein d'un même exercice. Même si le feedback est adapté à l'apprenant, son type demeure souvent uniforme.

L'état de l'art autour des feedbacks dans les plateformes d'apprentissage a été mené pour tenter de répondre aux questions suivantes :

1. Quels sont les types de feedbacks épistémiques dans les plateformes d'apprentissage de la programmation ?
2. Quels éléments sont considérés pour générer un feedback ?
3. Quelles techniques sont utilisées pour générer un feedback ?
4. Quels systèmes de décision des feedbacks ?

2.2 Types de feedbacks épistémiques dans les plateformes d'apprentissage de la programmation

Au sein des plateformes d'apprentissage de la programmation, les recherches s'intéressent le plus souvent à l'adaptation en boucle longue ou "Task Loop" qui peut être considérée comme une forme de feedback qui n'est pas affiché à

l'apprenant pendant la réalisation de la tâche mais en enchaînant de manière adaptative des tâches d'apprentissage [23, 20, 18]

Une autre forme de feedback fréquemment utilisée est la génération d'indices [2, 22, 24], réalisée à travers différentes méthodes. Contrairement à l'enchaînement adaptatif de tâches, la génération d'indices agit à court terme, fournissant rapidement une compréhension supplémentaire à l'apprenant pour surmonter ses difficultés à résoudre la tâche. Dans ce contexte, une question pertinente émerge : en cas de co-occurrence de multiples erreurs dans le même code, quelle erreur faut-il prioriser pour générer un indice ? Cette approche soulève la question de la connaissance qu'a le système de l'élève et du contexte de la tâche pour décider le feedback le plus adapté. Il est également pertinent de se demander s'il ne serait pas parfois préférable d'encourager l'élève à se concentrer davantage sur la résolution plutôt que de générer un indice à sa demande.

Une autre modalité de feedback consiste à proposer des cas de test pour lesquels le code a échoué. L'approche consiste à générer automatiquement les cas de test [8] et faire un feedback à l'apprenant lorsque son code ne passe pas complètement ou partiellement des cas de test. Cette méthode, bien qu'elle permet de signaler des anomalies dans le code de l'apprenant, ne fournit pas de connaissances approfondies sur les concepts de programmation mal appliqués ni sur l'erreur. L'indication de l'emplacement de l'erreur telle que suggérée par Malik dans son approche [17] présente les mêmes limites. A noter que ces feedbacks s'inscrivent davantage dans le "Step Loop", étant donné qu'ils sont fournis pour chaque soumission de l'apprenant.

On observe que la plupart des recherches se concentrent sur la proposition d'un seul type de feedback et ne varient pas leurs types en fonction de l'apprenant et de la situation. Nous voulons explorer dans ce travail de thèse la possibilité de décider automatiquement des feedbacks épistémiques de types différents. Dans cette perspective, notre question de recherche peut être formulée comme suit : Dans le cas où on dispose de plusieurs types de feedbacks pour une même situation, **quel système mettre en place pour décider le type de feedback le plus adapté ?**

2.3 Éléments guidant la génération des feedbacks

Selon deux revues de la littérature [5, 3], la génération des feedbacks s'appuie sur des éléments issus des experts (expert-driven), des données des apprenants (data-driven) ou un mixte des deux (hybride).

Les modèles dirigés par des experts (expert-driven) impliquent l'intervention humaine pour par exemple établir des règles [25], telles que des cas de tests, des solutions expertes aux problèmes, des bibliothèques de réponses correctes ou des ensembles d'erreurs typiques dans la discipline.

Les modèles dirigés par les données (data-driven) reposent sur l'utilisation exclusive de données, notamment celles de l'apprenant. Dans la plupart des cas, on s'appuie sur les codes soumis par les apprenants pour générer un feedback adapté [5, 7, 8, 22, 2]. Mais il y a également l'utilisation des données de log, des données contextuelles capturant l'environnement dans lequel l'apprenant

résout l'exercice, ou encore, plus rarement, des données comme l'EEG (électroencéphalogramme) ou des réponses à des questionnaires.

Enfin, le modèle hybride constitue une combinaison des éléments issus de l'expert et des données de l'apprenant. Par exemple, [24] utilise à la fois les solutions soumises par l'apprenant et les solutions correctes fournies par l'expert pour construire un espace de solutions permettant la génération de feedbacks adaptés.

2.4 Techniques de génération de feedbacks

Dans l'état de l'art, plusieurs techniques ont été employées pour générer des feedbacks [5]. La technique proposée par Malik et ses collègues [17] est dirigée par les données et s'appuie sur un modèle génératif qui permet à partir du code fourni par un apprenant de remonter son "processus de résolution". Ce modèle génératif vise à approcher le plus possible le processus d'évaluation des enseignants et donc fournir un feedback le plus précis possible à l'apprenant.

Une autre technique est celle basée sur des contraintes [25], où le code d'un apprenant est utilisé en conjonction avec un ensemble de contraintes pour fournir des indices vers la prochaine étape. Cette technique s'inscrit dans les modèles dirigés par l'expertise humaine, visant à proposer les modifications nécessaires pour atteindre le code correct. Néanmoins, cette approche comporte certaines limites, notamment en ce qui concerne l'incapacité à différencier deux codes distincts vérifiant chacun les mêmes contraintes. Ainsi, le feedback ne prend pas en compte des variations fines dans les codes des apprenants. De plus, un inconvénient majeur réside dans le temps de latence nécessaire pour fournir un feedback. Le feedback est reçu par mail (asynchrone à la réalisation de la tâche par l'apprenant) ce qui limite son impact et son efficacité.

Deux autres techniques de génération d'indices ont été proposées, l'une présentée par Tiffany Barnes et ses collègues [2] et l'autre par Rivers et ses collègues [24]. Dans la première technique, le modèle se concentre sur la génération automatique d'indices en utilisant les données historiques des apprenants et en utilisant un processus de décision Markovien. L'approche décisionnelle du modèle consiste à construire un graphe contenant toutes les solutions et les transitions entre ces solutions. En utilisant les données existantes, le modèle examine les passages entre deux solutions différentes. À l'aide de quatre systèmes de matching, le modèle tente de déterminer si une solution similaire à celle proposée par l'apprenant existe dans le graphe. En cas de correspondance, le modèle propose un "hint" pour guider l'apprenant vers la prochaine étape menant à la solution correcte. Cette approche permet une génération rapide de feedback, et la construction du graphe s'appuie sur les données antérieures des apprenants pour représenter un ensemble diversifié de solutions et de transitions entre celles-ci.

La deuxième technique propose la génération d'indices basée sur l'historique des codes des apprenants. L'idée sous-jacente est de construire un espace de solutions avec des liens représentant le passage d'un nœud à un autre en effectuant des opérations telles que l'addition, la suppression et la modification d'un élément dans la structure abstraite du code (Abstract Syntactic Tree ou AST).

Si la solution de l'élève réside dans cet espace, des indices sont suggérés vers la solution correcte la plus proche. Il convient de noter que cette méthode ne semble pas adaptée à un contexte "zero shot" (démarrage à froid), car elle requiert certains codes, et l'espace de solutions demeure étendu.

La technique employé dans [7] consiste à proposer des indices en "zero shot" en utilisant l'apprentissage par renforcement. Que l'on utilise le bootstrapping ⁴ ou non, l'agent de renforcement est capable de générer l'indice pour la prochaine étape.

Enfin, d'autres types de techniques émergent grâce au développement des modèles de langage (Large Scale Language Models, LLM) et leur utilisation dans le domaine éducatif. Des modèles tels que GPT-3, Gemini, LLMA2 affirment leur capacité à résoudre des problèmes de programmation de haut niveau selon divers benchmarks tels que MMLU ⁵, et d'autres, ainsi qu'à fournir des instructions et des feedbacks. Cependant, leur pertinence et efficacité demeurent encore insuffisamment testées dans le milieu éducatif. Dans [21], l'auteur examine la possibilité d'intégrer un GPT à une plateforme de résolution d'exercices en C# afin de fournir des indices aux apprenants. Ces indices sont générés par le modèle GPT-3.5. Cependant, l'évaluation de l'efficacité du modèle s'appuie uniquement sur les perceptions des apprenants à travers un questionnaire, ce qui constitue une limite à l'évaluation présentée. En effet, la question de l'impact des feedbacks générés par les modèles de langage reste pour l'instant sans réponse.

2.5 Méthodes de décision de feedbacks

La plupart des techniques étudiées dans l'état de l'art s'intéressent au processus de génération des feedbacks, souvent de même type. En particulier, elles s'intéressent à définir les données en entrée et aux traitements sur ces données pour permettre la génération d'un feedback adapté. Peu de recherches, à notre connaissance, s'intéressent à la décision des feedbacks, c'est à dire, au processus de choix entre plusieurs feedbacks possibles et de types différents pour un apprenant dans une situation donnée. Dans le contexte de notre travail de thèse nous formalisons le problème de décision des feedbacks comme un problème de décision dans l'incertain car il existe une incertitude sur le type et le contenu du feedback le plus adapté, sur la façon d'ordonner des feedbacks entre eux mais aussi sur les conséquences de ces feedbacks décidés sur l'apprenant, notamment sur leur apprentissage.

Parmi les différentes techniques étudiées dans l'état de l'art, la décision des feedbacks a été étudiée lors de la génération d'indices, et deux méthodes se distinguent particulièrement à cet égard. Tout d'abord, l'approche basée sur l'apprentissage par renforcement qui est une famille d'algorithmes d'apprentissage automatique où un agent apprend à prendre des décisions

⁴ Le bootstrapping en apprentissage par renforcement implique l'utilisation de méthodes itératives où un agent améliore continuellement son modèle et sa politique à partir des expériences passées.

⁵ <https://paperswithcode.com/dataset/mmlu>

séquentielles en interagissant avec un environnement (incluant généralement l'apprenant) pour maximiser l'espérance d'une récompense sur le long terme [26]. Selon ce principe, une action (le choix d'un indice parmi plusieurs possibles) est sélectionnée pour un apprenant dans une situation donnée. Ensuite, la deuxième méthode s'appuie sur la construction d'un espace de solutions [24]. Le processus décisionnel persiste dans le choix de l'indice menant à la prochaine étape générée. Étant donné qu'il existe plusieurs solutions correctes et, par conséquent, plusieurs itinéraires à partir de la solution de l'apprenant pour atteindre ces solutions, le système prend une décision sur l'indice à fournir en se basant sur le parcours le plus court pour atteindre une solution correcte.

L'état de l'art étudié montre clairement un fort intérêt pour la génération des feedbacks (les composants 1 et 2 du diagramme présenté à la figure 1) dans des plateformes d'apprentissage de la programmation. Par contre, nous constatons que le problème de décision de feedbacks de types différents est peu abordé dans la littérature étudiée. Ceci s'explique probablement par la contrainte de produire suffisamment de feedbacks différents permettant l'adaptation.

3 Objectifs de la recherche

Dans cette section, nous présentons nos deux objectifs de recherche (OR) envisagés pour la thèse.

3.1 OR1 : Identification de la politique de décision pour le feedback en fonction de l'apprenant

Comme évoqué dans la (Sect. 2.2), notre attention se porte sur la politique de décision du feedback.

Dans le domaine de la programmation, les informations incluses dans un feedback individuel peuvent englober divers éléments tels que l'emplacement du code erroné, des explications sur l'application incorrecte d'un concept de programmation, ou des conseils spécifiques visant à améliorer la solution de l'étudiant, par exemple "le programme est erroné parce que la ligne de code X ne respecte pas le concept Y" [15].

Il est à souligner que pour fournir un feedback adaptatif de manière générale, un système doit être capable d'analyser les erreurs présentes dans les solutions des apprenants. Cependant, la simple identification des erreurs ne suffit pas pour qu'un feedback soit véritablement adapté. Étant donné que les apprenants n'ont pas le même niveau de maîtrise des concepts de programmation et n'apprennent pas au même rythme, l'efficacité de leur apprentissage est probablement étroitement liée à la qualité de l'adaptation des feedbacks [9, 6].

Dans notre travail, nous formulons l'hypothèse suivante : la prise en compte du code de l'apprenant, des erreurs qui y sont détectées ainsi que des informations sur l'apprenant obtenues à travers son modèle de l'apprenant permettent de décider un feedback adapté impactant fortement l'apprentissage de l'apprenant.

Ces trois éléments (codes, erreurs et modèle de l'apprenant) constituent les entrées du système de décision que nous envisageons de proposer dans le cadre de cette thèse.

Ensuite, pour qu'un système puisse décider un feedback, il est nécessaire de disposer d'un ensemble diversifié de feedbacks pour un même type d'erreur, répondant aux caractéristiques cognitives variées des différents apprenants. Comme nous l'avons pointé à la section (**Sect.** 2.2), la plupart des travaux travaillant sur la génération de feedbacks, se focalisent sur un type unique de feedbacks tel que des indices vers la prochaine étape à faire ou la référence à des cas de tests qui ne passent pas. Cependant, notre objectif est de décider des feedbacks qui sont issus d'un processus de génération qui produit des feedbacks épistémique de différents types.

De manière plus formelle, nous cherchons à définir une fonction π (pour politique de décision) qui associe à un apprenant a et une situation s un feedback f . Étant donné que chaque élève est unique, et que le feedback peut varier pour une même tentative de solution, nous modélisons notre système comme un environnement stochastique.

La fonction π peut alors être définie comme une probabilité conditionnelle, notée $P(f|a, s)$, représentant la probabilité de décider un certain feedback f pour un apprenant a dans une situation s . Dans cette formalisation, notre objectif de thèse consiste à automatiser le calcul de la probabilité conditionnelle après chaque tentative de l'apprenant.

Cette formalisation du problème de décision des feedbacks permet de modéliser le système de décision comme un processus de décision markovien. L'utilisation de l'apprentissage par renforcement constitue ainsi une piste sérieuse pour décider les feedbacks. Étant déjà appliqué dans l'adaptation en "task loop", l'apprentissage par renforcement est une approche intéressante à explorer pour l'adaptation en "step loop".

En conclusion, dans le cadre de notre recherche, nous nous concentrons sur la "step loop", qui consiste à afficher des informations aux apprenants, généralement sous forme de feedback, afin de les guider dans la résolution de l'activité en cours et de favoriser la compréhension des concepts mobilisés.

4 Conclusion

Dans cet article, nous avons souligné l'intérêt de disposer de feedbacks adaptés, en explorant diverses méthodes utilisées pour décider différents types de feedback de nature épistémique à travers des cas pratiques pertinents. Dans la poursuite de nos recherches, une attention particulière sera accordée à la prise de décision concernant le type de feedback à dispenser. Une considération importante sera accordée à une approche hybride, combinant l'expertise humaine, l'IA générative et l'apprentissage par renforcement pour guider la décision sur le feedback à fournir. Cette approche vise à doter des plateformes d'apprentissage de la programmation d'un système de décision de feedbacks épistémiques et adaptatifs permettant de soutenir l'apprentissage des apprenants.

References

1. Alevin, V., McLaughlin, E.A., Glenn, R.A., Koedinger, K.R.: Instruction based on adaptive learning technologies. *Handbook of research on learning and instruction* **2**, 522–560 (2016)
2. Barnes, T., Stamper, J.: Toward automatic hint generation for logic proof tutoring using historical student data. pp. 373–382 (2008). https://doi.org/10.1007/978-3-540-69132-7_41
3. Bimba, A., Idris, N., Al-Hunaiyyan, A., Mahmud, R., Shuib, L.: Adaptive feedback in computer-based learning environments: a review. *Adaptive Behavior* **25**, 105971231772759 (2017). <https://doi.org/10.1177/1059712317727590>
4. Brusilovsky, P.: *Adaptive Hypermedia for Education and Training*, vol. 6, pp. 46–68 (01 2012). <https://doi.org/10.1017/CBO9781139049580.006>
5. Deeva, G., Bogdanova, D., Serral, E., Snoeck, M., De Weerd, J.: A review of automated feedback systems for learners: Classification framework, challenges and opportunities. *Computers & Education* **162**, 104094 (2021)
6. Dunlosky, J., Rawson, K.A., Marsh, E.J., Nathan, M.J., Willingham, D.T.: Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest* **14**(1), 4–58 (2013). <https://doi.org/10.1177/1529100612453266>
7. Efremov, A., Ghosh, A., Singla, A.: Zero-shot learning of hint policy via reinforcement learning and program synthesis. In: *International Educational Data Mining Society* (2020)
8. Gao, J., Pang, B., Lumetta, S.: Automated feedback framework for introductory programming courses. pp. 53–58 (2016). <https://doi.org/10.1145/2899415.2899440>
9. Gouli, E., Gogoulou, A., Papanikolaou, K., Grigoriadou, M.: An adaptive feedback framework to support reflection, guiding and tutoring. In: Magoulas, G., Chen, S. (eds.) *Advances in Web-based Education: Personalized Learning Environments*, p. 178–202. Idea Group Inc., Calgary, AB, Canada (2006)
10. Hattie, J., Gan, M.: Instruction based on feedback. In: *Handbook of Research on Learning and Instruction*, pp. 249–271 (2011)
11. Hattie, J., Timperley, H.: *The power of feedback*. Review of educational research **77**(1), 81–112 (2007), publisher: Sage Publications Sage CA: Thousand Oaks, CA
12. Jolivet, S., Dechaux, E., Gobard, A.C., Wang, P.: Description et analyse de trois EIAH d'apprentissage de Python. In: *Actes de l'atelier APIMU : 11ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain*. Brest (2023)
13. Koedinger, K.R., Brunskill, E., Baker, R.S., McLaughlin, E.A., Stamper, J.: New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine* **34**(3), 27–41 (2013)
14. Le, N.T.: A classification of adaptive feedback in educational systems for programming. *Systems* **4**(2), 22 (2016)
15. Le, N.T.: A classification of adaptive feedback in educational systems for programming. *Systems* **4**, 22 (2016). <https://doi.org/10.3390/systems4020022>
16. Luengo, V.: *Les rétroactions épistémiques dans les environnements informatiques pour l'apprentissage humain* (2009)
17. Malik, A., Wu, M., Vasavada, V., Song, J., Coots, M., Mitchell, J., Goodman, N., Piech, C.: Generative grading: Near human-level accuracy for automated feedback on richly structured problems (2019), <https://arxiv.org/abs/1905.09916>
18. Na Nongkhai, L., Wang, J., Mendori, T.: A framework of exercise recommendation for novice learners in computer programming. In: In M. Mercedes, T. Rodrigo, S. Iyer, & A. Mitrovic (Eds.) (2021)

19. Narciss, S.: Feedback strategies for interactive learning tasks. In: Handbook of research on educational communications and technology, pp. 125–143. Routledge (2008)
20. Nwana, H.S.: Intelligent tutoring systems: an overview. *Artificial Intelligence Review* **4**(4), 251–277 (1990)
21. Pankiewicz, M., Baker, R.S.: Large language models (GPT) for automating feedback on programming assignments. In: Shih, J.L., Kashihara, W., Ogata, H. (eds.) Proceedings of the 31st International Conference on Computers in Education. pp. 68–77. Asia-Pacific Society for Computers in Education, <https://eds.let.media.kyoto-u.ac.jp/ICCE2023/wp-content/uploads/2023/12/ICCE2023-Proceedings-V1-1214-final.pdf>
22. Piech, C., Huang, J., Sahami, M., Guibas, L.: Autonomously generating hints by inferring problem solving policies. pp. 195–204 (2015). <https://doi.org/10.1145/2724660.2724668>
23. Reddy, S., Levine, S., Dragan, A.D.: Accelerating human learning with deep reinforcement learning (2017)
24. Rivers, K., Koedinger, K.: Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education* (2015). <https://doi.org/10.1007/s40593-015-0070-z>
25. Singh, R., Gulwani, S., Solar-Lezama, A.: Automated feedback generation for introductory programming assignments. *ACM SIGPLAN Notices* **48** (2012). <https://doi.org/10.1145/2491956.2462195>
26. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
27. VanLehn, K.: The behavior of tutoring systems. *International journal of artificial intelligence in education* **16**(3), 227–265 (2006)
28. VanLehn, K.: Regulative loops, step loops and task loops. *International Journal of Artificial Intelligence in Education* **26**, 107–112 (2016)
29. Wisniewski, B., Zierer, K., Hattie, J.: The power of feedback revisited: A meta-analysis of educational feedback research. *Frontiers in Psychology* **10**, 3087 (2020). <https://doi.org/https://doi.org/10.3389/fpsyg.2019.03087>, publisher: Frontiers