



**Projet de fin d'études**  
**Approche logicielle pour l'évaluation de stratégies**  
**de testing pédagogique à l'Université**

**Diplôme d'ingénieur ENSEEIHT filière SN**  
**Master PSMSC**  
**ISAE-Supaero – IDEA**  
**Université Toulouse - Jean-Jaurès – CLLE**

**Tuteur entreprise : Bruno Ilponse**

**Tuteur école : Joseph Gergaud**

**Nassim Bennouar**

**01 septembre 2023**

## Remerciements

Je tiens tout d'abord à exprimer ma gratitude à Bruno Ilponse pour son soutien et ses compétences qui ont grandement contribué à enrichir mon parcours durant ces trois années d'apprentissage.

Je souhaite également remercier Laurent Dairaine ainsi que toute l'équipe IDEA pour leur accueil chaleureux et leur professionnalisme, qui ont su instaurer un environnement propice à mon épanouissement professionnel.

En outre, mes remerciements vont à mon tuteur de l'école, Joseph Gergaud, ainsi qu'à l'ensemble de l'équipe pédagogique de l'ENSEEIH pour leur encadrement et la qualité de l'enseignement dispensé.

## Résumé

Ce rapport présente une approche logicielle pour l'évaluation de stratégies de testing pédagogique à l'université et conclue un apprentissage de trois ans à l'ISAE-Supaero. Les réalisations logicielles ainsi que les perspectives et les recherches autour du sujet seront abordées. Le testing est une méthode pédagogique qui consiste à évaluer régulièrement les apprenants durant la phase d'apprentissage afin d'améliorer l'ancrage des connaissances. Ce processus, favorisant une mémorisation active, est devenu un sujet de recherche important dans le domaine de l'éducation. Ainsi, la communauté scientifique se consacre à l'élaboration de nouvelles stratégies de testing, comme celle de l'intercalage présentée par Sana & Yan dans une étude publiée en 2022. L'intercalage est une technique qui consiste à entrelacer les concepts évalués dans un test, plutôt que de les évaluer de manière séquentielle comme c'est le cas dans un test traditionnellement bloqué. Cette étude a démontré que l'intercalage a un impact notable sur la rétention des connaissances un mois après la période d'apprentissage. Cependant, pour valider cette étude, il est nécessaire de reproduire ses effets à plus grande échelle, en faisant varier les conditions expérimentales. L'objectif de ce projet est la conception et le développement d'une solution logicielle qui rend ces expériences plus facilement réalisables, de façon générique. Cela est réalisé en automatisant la mise en œuvre des tests pédagogiques, en fournissant une interface utilisateur accessible, et en assurant une manipulation appropriée des données de mesure.

## Abstract

This report presents a software approach for the evaluation of testing strategies at a university scale, marking the conclusion of a three-year apprenticeship at ISAE-Supaero. It will discuss the software implementations, as well as the perspectives and surrounding research on the subject. Testing is a pedagogical method that involves the regular assessment of learners during the learning phase with the aim of enhancing knowledge anchoring. This process, which promotes active memorization, has emerged as a critical research topic in the field of education. Consequently, the scientific community is committed to developing new testing strategies, such as the interleaving technique introduced in a study by Sana & Yan in 2022. Interleaving involves interspersing the concepts evaluated in a test, rather than assessing them sequentially as in traditional blocked tests. The study demonstrated that interleaving has a significant impact on knowledge retention one month after the learning phase. However, to validate this study, it is necessary to reproduce its effects on a larger scale, varying the experimental conditions. The aim of this project is the design and development of a software solution that makes these experiments affordable, in a generic manner. This is achieved by automating the implementation of the tests, providing an accessible user interface, and ensuring the appropriate handling of measurement data.

## Table des matières

<b>Remerciements.....</b>	<b>2</b>
<b>Résumé .....</b>	<b>3</b>
<b>Abstract.....</b>	<b>4</b>
<b>1 Introduction.....</b>	<b>7</b>
<b>1.1 Acteurs .....</b>	<b>7</b>
1.1.1 IDEA - ISAE-Supaero .....	7
1.1.2 CLLE - Université Toulouse II Jean Jaurès .....	8
1.1.3 TICEA – Université Toulouse Capitole .....	8
<b>1.2 Contexte du projet.....</b>	<b>9</b>
1.2.1 Effet d’intercalage de Sana & Yan.....	9
1.2.2 Reproduction de l’expérience par Fanny Verkampt .....	11
1.2.3 Logiciel à concevoir .....	12
<b>1.3 Technologies utilisées .....</b>	<b>15</b>
1.3.1 Moodle .....	15
1.3.2 Vue.js .....	15
1.3.3 Django .....	16
<b>2 Formalisation et architecture.....</b>	<b>16</b>
<b>2.1 Séquence pédagogique .....</b>	<b>16</b>
<b>2.2 Architecture globale.....</b>	<b>21</b>
<b>2.3 Générateur de séquences .....</b>	<b>23</b>
2.3.1 Découpage du processus de génération.....	23
2.3.2 Structures des données entrantes .....	24
<b>2.4 Gestion des données d’expérience.....</b>	<b>28</b>
2.4.1 Architecture de l’envoi des données.....	28
2.4.2 Structure des données d’expérience.....	29
2.4.3 Protocole d’envoi des données de résultat .....	30
<b>3 Mise en oeuvre .....</b>	<b>31</b>
<b>3.1 Générateur de séquences .....</b>	<b>31</b>
3.1.1 Pre-solve factory et post-solve factory .....	31
3.1.2 Gestions des contraintes.....	32
<b>3.2 Plug-in moodle .....</b>	<b>32</b>
3.2.1 Le formulaire.....	32

3.2.2	Génération des activités.....	34
<b>4</b>	<b>Recherche : intégration de la pratique espacée .....</b>	<b>36</b>
4.1	Motivation.....	37
4.2	Paramétrage de la récupération espacée .....	37
4.2.1	Variables .....	38
4.3	Recherche de solutions techniques .....	39
4.3.1	Hard vs soft constraints .....	40
4.3.2	Approches pour résoudre le problème.....	41
<b>5</b>	<b>Réflexion sur mon développement personnel et professionnel</b>	
	<b>42</b>	
5.1	L'écueil de la sur-conception.....	42
5.2	La communication dans le cadre d'un nouveau projet....	42
5.3	Apprentissage technique.....	43
<b>6</b>	<b>Bibliographie .....</b>	<b>43</b>
<b>7</b>	<b>Annexes.....</b>	<b>45</b>

# 1 Introduction

J'ai réalisé ce projet de fin d'études dans le cadre de mon alternance à l'ISAE-Supaero, en collaboration avec des chercheurs en psychologie cognitives de l'Université de Toulouse Jean Jaurès et avec l'université Toulouse Capitole.

## 1.1 Acteurs

Ce projet est le fruit d'une collaboration entre trois entités distinctes que je vais présenter succinctement dans cette section.

### 1.1.1 IDEA - ISAE-Supaero

- J'ai effectué mon alternance au sein de l'unité IDEA ("Innovations Digitales pour l'Enseignement Aéronautique") de l'ISAE-Supaero, dont la mission est d'assurer la transformation numérique de l'institut – en particulier de son cœur de métier : la formation. Elle est structurée autour de trois fonctions majeures :
  - 
  - OPS (Opérations) : l'unité se consacre aux aspects opérationnels de soutien à l'enseignement, la formation pédagogique des enseignants, l'accompagnement pédagogique, etc.
  - FAB (Fabrication) : il s'agit de la création de contenus multimédias (e.g cours vidéo, micro-contenu, reproduction d'expériences scientifiques...), de formations et programmes totalement ou partiellement en ligne, par exemple des MOOCs ou un programme de Mastère Spécialisé® 100% en ligne.
  - LAB (Laboratoire) : centrée sur la recherche et le développement dans le secteur des EdTech, cette unité développe et assure la maintenance de diverses technologies. Parmi ses réalisations notables, on cite en particulier IQUIZ<sup>1</sup>, une plateforme de quiz interactifs et agiles en amphi, IREAL<sup>2</sup>, une plateforme d'expérimentation scientifiques digitalisées pour les enseignement en travaux pratiques à distance, ou encore

---

<sup>1</sup> Voir <https://iquiz.univ-toulouse.fr>

<sup>2</sup> Voir <https://ireal.isae-supero.fr>

l'écosystème NaaS<sup>3</sup> (Nugget as a Service), un environnement dédié à la réutilisation de micro-contenus éducatifs et de Microlearning<sup>4</sup>.

Le projet présenté sur ce rapport entre dans la fonction LAB, et a été tutoré par Bruno Ilponse, responsable EdTech de l'institut. L'unité IDEA est quant à elle pilotée par Laurent Dairaine, Chief Digital Officer de l'institut et ex-enseignant-chercheur en informatique et réseaux.

### **1.1.2 CLLE - Université Toulouse II Jean Jaurès**

Le projet est réalisé en collaboration avec Patrice Terrier et Fanny Verkampt, deux chercheurs affiliés au Laboratoire Cognition, Langues, Langage, Ergonomie (CLLE) de l'Université Toulouse II Jean Jaurès.

Patrice Terrier se consacre à l'étude des phénomènes de mémoire et de conscience, en particulier dans le contexte de l'interaction entre l'individu et le système. Fanny Verkampt, quant à elle, étudie l'utilisation de l'entretien cognitif auprès des mineurs pour recueillir des témoignages en contexte judiciaire.

Ils sont tous deux intéressés par l'étude des stratégies de testing pédagogiques pour améliorer la rétention des connaissances.

### **1.1.3 TICEA – Université Toulouse Capitole**

L'équipe TICEA (Technologies de l'Information et la Communication pour les Enseignements et l'Apprentissage) a pour mission de développer et accompagner l'utilisation de méthodes et outils pédagogiques transversaux favorisant la réussite des étudiants, auprès des enseignants et formateurs de l'Université Toulouse Capitole.

Nous avons collaboré avec Eleonore Falière, ingénieure pédagogique au sein de l'équipe TICEA, qui est intéressée par la démocratisation de la pratique du testing à l'université.

---

<sup>3</sup> Voir <https://online.isae-supaero.fr/naas>

<sup>4</sup> Voir <https://microlearning.groupe-isae.fr/>



## 1.2 Contexte du projet

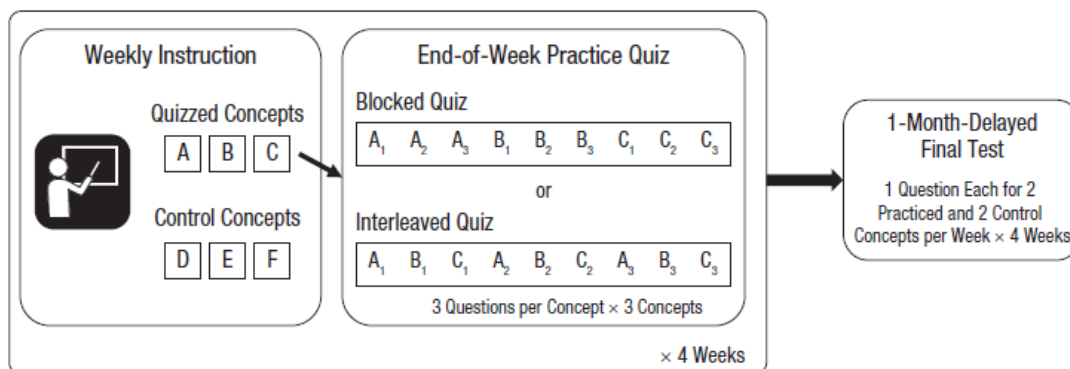
Une étude publiée en 2022 par Sana & Yan démontre les effets positifs d'une certaine stratégie de testing. Fanny Verkampt, pour tenter de répliquer cet effet, a reproduit l'expérience à la main, et s'est rendue compte que le travail de reproduction est long et exigeant. Notre projet consiste en l'automatisation de ce travail. Dans cette section, nous présenterons succinctement l'étude de Sana & Yan, le travail de Fanny Verkampt, et présenterons notre problématique d'ingénierie.

### 1.2.1 Effet d'intercalage de Sana & Yan

L'étude motivant le projet est intitulée *“Interleaving Retrieval Practice Promotes Science Learning”*<sup>[14]</sup>, est menée par Sana & Yan, et s'est penchée sur l'efficacité de la pratique de récupération entrelacée dans le contexte scolaire.

Sur une période de quatre semaines, 155 élèves canadiens de la 9ème à la 12ème année, équivalent de la 3ème à la Terminale en France, ont été soumis chaque semaine à un quiz portant sur la moitié des concepts enseignés durant cette semaine.

Ces quiz étaient soit regroupés par concept (bloqués) soit entrelacés avec différents concepts (intercalés). On y voit que, chaque semaine, trois concepts A, B et C sont testés sur les six concepts A, B, C, D, E, F enseignés. Pour chaque concept il y a trois questions, qui dans le cas des quiz bloqués sont ordonnées de cette façon (A1,A2,A3,B1,B2,B3,C1,C2,C3), et dans le cas des quiz intercalés sont ordonnées de cette façon (A1,B1,C1,A2,B2,C2,A3,B3,C3). Un mois après le dernier quiz, une évaluation a été réalisée sur ces mêmes concepts. Le schéma ci-dessous récapitule l'expérience.



**Figure 1 : General design of the study.**

Les résultats montrent un avantage notable pour les quiz entrelacés : les élèves ont obtenu des scores moyens de 63% pour les concepts testés de cette manière, contre 54% pour les concepts testés via des quiz bloqués.

Le  $d$  de Cohen, représentant la taille de l'effet, indique que l'impact de l'entrelacement est substantiel avec une valeur de 0,35 comparativement à l'effet de la pratique de récupération seule, qui est de 0,30. Ces résultats suggèrent que l'entrelacement est une stratégie pédagogique efficace pour améliorer la rétention des connaissances des apprenants sur le long terme. Il est nécessaire de répliquer l'effet pour pouvoir le valider.

Dans le domaine de l'éducation, un défi majeur pour les enseignants de sciences est de favoriser un apprentissage conceptuel profond plutôt que la simple mémorisation. Pour illustrer ce point, considérez l'apprentissage en physique des différents types de circuits. Il ne suffit pas de connaître les circuits en série ou en parallèle, mais de comprendre comment et pourquoi ils diffèrent, et de savoir quand appliquer chaque concept dans des situations variées.

C'est pour adresser ce défi que l'approche du testing est adoptée, en utilisant notamment des méthodes comme l'intercalage et les quiz bloqués. Toutes deux sont des formes de pratique de récupération, mais l'entrelacement pourrait particulièrement favoriser la mise en perspective des concepts en les mélangeant pendant l'apprentissage. Cette technique pourrait encourager les étudiants à comparer activement et à distinguer les concepts d'avantage que le testing

bloqué, renforçant ainsi leur compréhension conceptuelle à travers une pratique dite “de récupération”.

### **1.2.2 Reproduction de l’expérience par Fanny Verkampt**

Afin de tenter de répliquer l’effet d’intercalage, Fanny Verkampt a appliqué la stratégie d'entrelacement à un cours d'épistémologie de L1 à fort effectif. Le processus nécessitait une préparation minutieuse : il s'agissait tout d'abord de formuler les questions, puis de les classer et de les ordonner pour enfin les implémenter sous forme de tests sur la plateforme Moodle.

Le cours se divisait en deux chapitres majeurs. Le premier, de nature plus théorique, comportait quatre tests intermédiaires. Chacun de ces tests explorait trois concepts différents, représentés par trois questions à choix multiples (QCM), suivis d'un QCM de contrôle. Trois semaines après le quatrième test intermédiaire, un test final était administré. Le second chapitre, orienté vers la pratique, suivait la même structure que le premier. À noter qu'aucun des tests, qu'ils soient intermédiaires ou finaux, n'était noté.

Sur le plan de la gestion avec Moodle, les étudiants inscrits à l'Unité d'Enseignement (UE) étaient répartis en deux groupes distincts : le groupe 1 et le groupe 2. Ces tests étaient alors transposés sur Moodle, avec une duplication pour créer deux versions distinctes chaque semaine. Après avoir intégré les questions, l'ordonnancement des QCM était effectué soit de manière bloquée soit de manière alternée. Pour éviter que les étudiants ne détectent la manipulation expérimentale en discutant entre eux, les versions n'étaient pas explicitement nommées "1 et 2" ou "A et B". Une attention particulière a été portée au contrebalancement pour le test suivant. Dès le deuxième test, une version était ouverte seulement aux étudiants ayant passé le test initial, et une autre version destinée au reste du groupe. Cette distinction permettait d'identifier les étudiants ayant participé à tous les tests intermédiaires lors de l'analyse des données du test final. Enfin, pour répartir les étudiants entre les deux versions de tests, le système de restrictions Moodle était utilisé.

Le test final, lui, contenait à chaque fois une question pour deux concepts pratiqués et deux concepts testés par semaine, ceci pour les 4 semaines. Le

document ci-dessous présente la structure du test final tel que Fanny Verkampt l'a représenté.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1			Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
2	Test 1	concept A	testé				control								control			
3		concept B									testé							
4		concept C																
5	Test 2	concept A		control				testé										
6		concept B										control						
7		concept C																
8	test 3	concept A		testé					control									
9		concept B																
10		concept C																
11	test 4	concept A			control					testé								
12		concept B												control				
13		concept C																testé

**Figure 2 : structure des tests finaux, Qn étant la question en position n dans le test, “testé” signifiant une question sur le concept testé de même ordonnée, et “control” sur un concept de contrôle.**

Par ailleurs, l'extraction des données de l'expérience s'est avérée être une étape particulièrement ardue. Il était impératif d'obtenir des résultats sous une forme qui soient à la fois exploitable et intelligible. L'objectif était d'extraire deux matrices : l'une reflétant le score de chaque étudiant pour chaque question de chaque test (initiaux et finaux), et l'autre permettant de quelle façon chaque étudiant a été testé sur chaque concept. L'obtention de la deuxième matrice peut se traduire par la mise en place de matrices associatives, l'une reliant un étudiant à un groupe spécifique, l'autre illustrant la configuration de test pour chaque concept selon le groupe. Ainsi, on peut en déduire les performances de chaque étudiant sur les différents concepts testés par quiz bloqués ou intercalés.

L'exploitation des logs pour obtenir ces matrices s'avère être une tâche complexe, requérant une bonne connaissance de la plateforme Moodle, des connaissances en ingénierie de la donnée, et un travail de vérification pour s'assurer que des étudiants n'aient pas été laissés de côté. Cette tâche incombe généralement à un ingénieur pédagogique, est relativement chronophage, et demeure sensiblement la même peu importe le scénario.

### 1.2.3 Logiciel à concevoir

Face à cet enjeu, nous avons déduit la problématique suivante : le développement d'un logiciel capable, à partir de la description formelle du protocole et des données d'expérience (questions étiquetées, contraintes

temporelles), de générer une séquence pédagogique correcte.

La description formelle du protocole doit contenir :

- **La configuration des tests**, c'est-à-dire leur nature et leurs caractéristiques propres.

Dans la littérature sur le testing pédagogique, un test est généralement conçu comme un instrument visant à activer la capacité d'un individu à récupérer activement et à appliquer des informations précédemment apprises. Un test peut-être un quiz, mais également une évaluation du travail de ses pairs, un travail de rédaction, des cartes mémoires - ou "flashcards" – qui sur le recto présentent le nom d'un concept, et sur le verso les notions principales à retenir sur ledit contexte, etc.

Par exemple, « *The effects of interleaving versus blocking on foreign language pronunciation learning* »<sup>[3]</sup> par Shana K. Carpenter & Frank E. Mueller étudie l'effet intercalé vs l'effet bloqué dans l'apprentissage de la prononciation des langues étrangères. L'activité n'est pas un quiz mais des mots apparaissent à l'écran et sont prononcés, quatre mots conformes à huit règles de présentation.

- **Les informations d'ordonnement des tests**, qui permettent de déterminer quand un test doit être programmé, et qui est concerné par le test.

Lors de mes recherches sur les protocoles d'évaluation de stratégies de testing, j'ai engagé une discussion avec les chercheurs concernant leur structure temporelle. Le plus petit dénominateur commun des protocoles est la notion de cycle. Concrètement, un cycle fait référence à une séquence d'étapes répétitives et structurées dans un protocole.

Dans l'expérience de Sana & Yan, il y a deux cycles, chaque cycle dure deux semaines : lors de la première semaine du cycle, le premier groupe a un test bloqué et le deuxième groupe a un test intercalé, puis vice-versa lors de la deuxième semaine. Ce cycle de deux semaines est répété deux fois, et le test final est effectué un mois plus tard.

Chaque cycle permet de mesurer la configuration du test (intercalé vs bloqué dans le cas de Sana & Yan) en variable intra-groupe, c'est-à-dire

qu'on mesure la variabilité des scores observés au sein d'un même groupe pour les concepts testés par une configuration ou l'autre. Cela implique qu'un cycle dure au minimum deux périodes, une période étant essentiellement le laps de temps pendant lequel une configuration spécifique est mise en œuvre pour chaque groupe.

Une configuration de test peut ne contenir qu'un test (e.g l'expérience de Sana & Yan), mais peut également contenir plusieurs tests (e.g un quiz et des rappels), donc une période contient au moins deux tests (un par groupe), mais possiblement plus.

Ce modèle cyclique assure une évaluation équitable et rigoureuse des différentes conditions expérimentales, réduisant ainsi les risques de biais.

- **Les informations de groupe**, sous quelles conditions sont formés les groupes, sont-ils formés aléatoirement ? ou sont-ils formés par l'enseignant ? Le concepteur de l'expérience pourrait configurer ceci. Il serait également possible de choisir aléatoirement l'assignation d'un groupe.
- Dans l'expérience de Sana & Yan, les groupes étaient les classes et étaient donc déjà formés, mais ils ont aléatoirement été assignés pour commencer avec un test bloqué ou intercalé.
- La conception des questions, les potentielles incitations financières (e.g les étudiants ont reçu un bon de 3\$ pour un café en échange de leur consentement à participer à l'expérience), le choix des concepts, sont en dehors de la portée de notre logiciel mais font partie du protocole scientifique.

Les données d'expériences sont les objets du test (les questions dans le cas d'un quiz), les concepts, la liste des étudiants (ou les groupes déjà formés), ainsi que le planning et les contraintes temporelles.

Une séquence pédagogique est un ensemble organisé d'activités d'enseignement et d'apprentissage visant à atteindre des objectifs éducatifs précis. Nous définirons la notion de séquence pédagogique à la section 2.1.

Le logiciel est par ailleurs conçu dans une double optique : évaluer des stratégies de testing et déployer ces stratégies en pratique. Eléonore Falière de l'UT1 a mis en avant l'importance de démocratiser le testing auprès des enseignants. Ainsi, au-delà de sa capacité à évaluer et comparer différentes méthodologies de testing, le logiciel doit également être accessible et intuitif pour les enseignants qui souhaitent simplement mettre en œuvre des stratégies de testing sans s'embarrasser de complexités d'un protocole scientifique. C'est dans cet esprit que nous avons envisagé un mode de fonctionnement simplifié pour répondre à cet impératif.

### **1.3 Technologies utilisées**

#### **1.3.1 Moodle**

Moodle, acronyme de "Modular Object-Oriented Dynamic Learning Environment", est un système de gestion de l'apprentissage (Learning Management System ou LMS). Fondé en 2001, il est développé en PHP et est maintenu sur le modèle open source. Les fonctionnalités de Moodle peuvent être étendues et personnalisées grâce à l'ajout de modules complémentaires, appelés plugins. En France, Moodle est largement utilisé pour assurer la gestion de l'apprentissage des étudiants, notamment à l'Université de Toulouse.

Moodle occupe une place centrale dans notre projet. C'est sur cette plateforme que nous avons implémenté la partie client de notre logiciel, sous forme de plugin.

#### **1.3.2 Vue.js**

Vue.js est un framework JavaScript progressif conçu pour la création d'interfaces utilisateur et d'applications monopages. L'un des principaux atouts de Vue.js est son approche basée sur les composants. Chaque composant représente une unité indépendante avec sa propre logique, ses données et son rendu, permettant ainsi une organisation claire et une réutilisation aisée du code. Cette structure par composants favorise le développement rapide d'interfaces utilisateur réactives.

De plus, la vaste communauté autour de Vue.js a contribué à la mise à disposition de nombreux composants préconçus. Ces composants, souvent disponibles sous

forme de packages ou de bibliothèques, nous permettent d'économiser du temps en épargnant le développement ex-nihilo des fonctionnalités de base. Cela permet de développer rapidement des expériences utilisateur modernes et performantes.

Vue.js a été utilisé dans le développement du plugin client sur Moodle, du "front-end".

### **1.3.3 Django**

Django est un framework de développement web haute performance écrit en Python. Favorisant le développement rapide, il se caractérise par une architecture propre et pragmatique. L'une de ses principales caractéristiques est son système "ORM" (Object-Relational Mapping) qui simplifie grandement les interactions avec les bases de données en permettant d'utiliser le langage Python au lieu de requêtes SQL directes. Pour notre projet, nous avons adopté Django, complété par le Django REST framework, afin de développer une API robuste et évolutive. Cette dernière est essentiellement conçue pour déployer la logique de génération de séquences pédagogiques et fait office d'intermédiaire entre la base de données et le front-end. Notre choix s'est porté sur cette combinaison principalement pour sa facilité et sa rapidité de développement, tout en tenant compte de la nécessité d'une évolutivité future.

## **2 Formalisation et architecture**

Dans cette section, je vais procéder à une formalisation précise des termes employés. De plus, j'exposerai l'architecture du logiciel, offrant ainsi une vue d'ensemble de sa structure et de ses composants principaux.

### **2.1 Séquence pédagogique**

Le but du logiciel est de générer et d'implémenter une séquence pédagogique en fonction d'un protocole précis et du contenu fourni. Il faut donc définir ce qu'est une séquence pédagogique.

Pour cela, je me suis inspiré du travail de Pierre Dillenbourg, chercheur en technologies de l'éducation, dans "Orchestration Graphs", un ouvrage qui

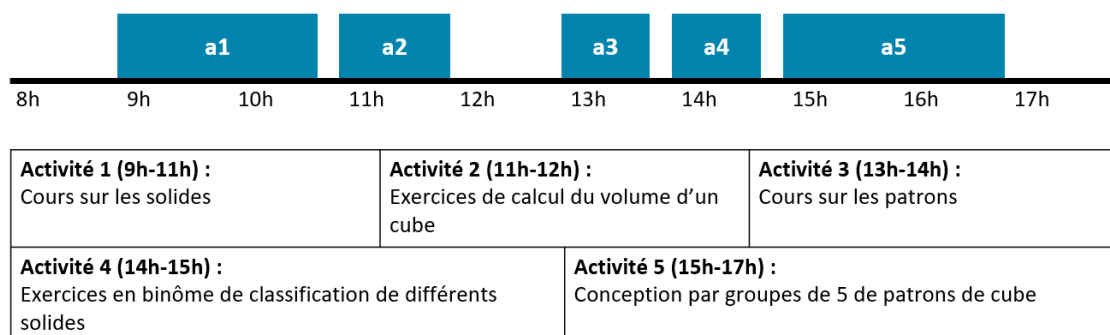


propose un modèle formel pour structurer des activités pédagogiques. Je ne vais pas utiliser son modèle de graphe, notre besoin étant trivial, mais je vais utiliser sa définition d'une séquence pédagogique.

Pierre Dillenbourg définit une séquence (ou scenario) pédagogique comme suit :

A pedagogical scenario is a sequence  $(a_1, a_2, \dots, a_i, \dots, a_n)$  where  $a_i$  is the  $i^{\text{th}}$  learning activity.

Voici deux exemples de séquences pédagogiques, le premier (Figure 3) est la séquence pédagogique associée à un cours de géométrie de niveau élémentaire. Les activités se déroulent de façon synchrone, durent une heure ou deux, et la séquence prend fin à la fin de la journée.



**Figure 3 : Séquence pédagogique d'un cours de géométrie**

La deuxième séquence (Figure 4) est beaucoup plus longue, peut être suivie de façon asynchrone, et concerne un MOOC en ligne.

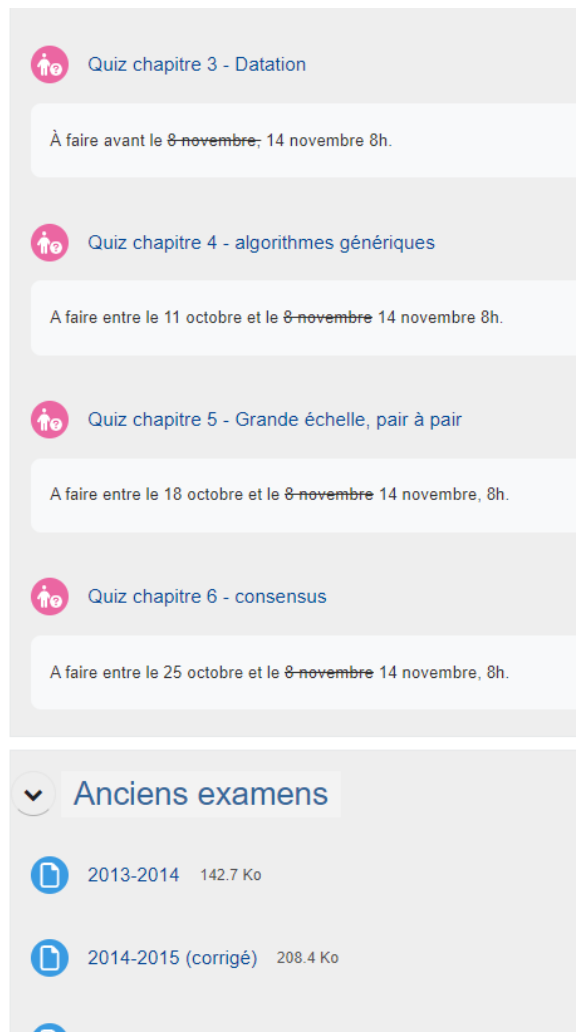


<b>Activité 1 (2 semaines) :</b> Les étudiants suivent cinq vidéos d'introductions avec deux QCM	<b>Activité 2 (1 semaine) :</b> Les étudiants ont une étude de cas à faire et rendre en PDF	<b>Activité 3 (1 semaine) :</b> Les étudiants ont un TP technique présenté par cinq vidéos
<b>Activité 4 (1 semaine) :</b> Les étudiants sont automatiquement groupés par quatre dans une salle de vidéoconférence et doivent corriger la réalisation des uns et des autres.		<b>Activité 5 (2 semaines) :</b> Les quatre étudiants doivent étoffer leur TP pour en faire un gros projet par équipe. Une session de vidéo conférence est prévue avec un professeur encadrant

**Figure 4 : Séquence pédagogique d'un MOOC**

Ces deux séquences sont correctes selon la définition, mais présentent deux formats très différents, il est donc nécessaire de définir ce qu'est une activité.

Dans notre cas, une activité est une activité au sens de Moodle, il peut s'agir d'un quiz, d'un PDF, d'une vidéo interactive, ou de tout autre type d'élément pédagogique que l'on peut inclure dans un cours du LMS. L'activité dure le temps durant laquelle l'étudiant est censé la consulter.



**Figure 5 : Extrait d'un cours sur la plateforme Moodle de l'ENSEEIH**

La figure 5 présente un exemple d'espace de cours Moodle, où les différentes activités sont les quiz et les annales d'examen. Cet extrait de cours est celui que voit un étudiant, chaque étudiant a donc sa propre séquence pédagogique.

Voici comment Pierre Dillenbourg inscrit une activité dans le temps :

An activity  $a_i$  runs from starting time  $t_i^s$  to the ending time  $t_i^e$   
 The duration of  $a_i$  is  $d_i = t_i^e - t_i^s$   
 Time is counted as the number of time units (e.g., seconds, weeks) from the  
 start of the scenario and is denoted  $t_0$ ,  $t_0 = 0$

The time lag between the end of activity  $a_i$  and the beginning of  $a_j$  is denoted  
by  $l_{ij} = t_j^s - t_i^e$

Dans notre cas, l'unité temporelle est la milliseconde, afin de faciliter le traitement par timestamp et d'éviter de limiter le système inutilement.

Nous distinguons deux types d'activités, les tests et le contenu d'enseignement. Notre logiciel a pour vocation première de générer des tests, nous nous concentrerons donc dessus.

Pour un test,  $t_i^s$  correspond au moment où le test est rendu disponible à l'étudiant, et  $t_i^e$  correspond au moment où il n'est plus possible d'entrer dans le test et où l'étudiant n'est plus en train d'effectuer le test.

Pour un contenu d'enseignement, nous nous contenterons pour l'instant de dire que  $t_i^s$  correspond au début la période associée au contenu (dans le modèle cyclique présenté à la section 1.2.3), et  $t_i^e$  à la fin de la période associée au contenu. Cela peut être raffiné pour des besoins futurs.

A noter que les contenus d'enseignement peuvent être disponibles à la consultation avant et après la période liée. On distingue alors le temps de l'activité et le temps de disponibilité de cette activité sur la plateforme. Cela peut aussi être raffiné si besoin.

La notion de "time lag" dont parle Pierre Dillenbourg n'est pertinente que pour les activités qui se suivent temporellement et ne superposent pas dans le temps. Seules les activités dépendantes entre elles pédagogiquement, c'est-à-dire que le suivi par l'étudiant de la première activité influe sur la qualité de suivi de la seconde activité, doivent nécessairement avoir un lag positif.

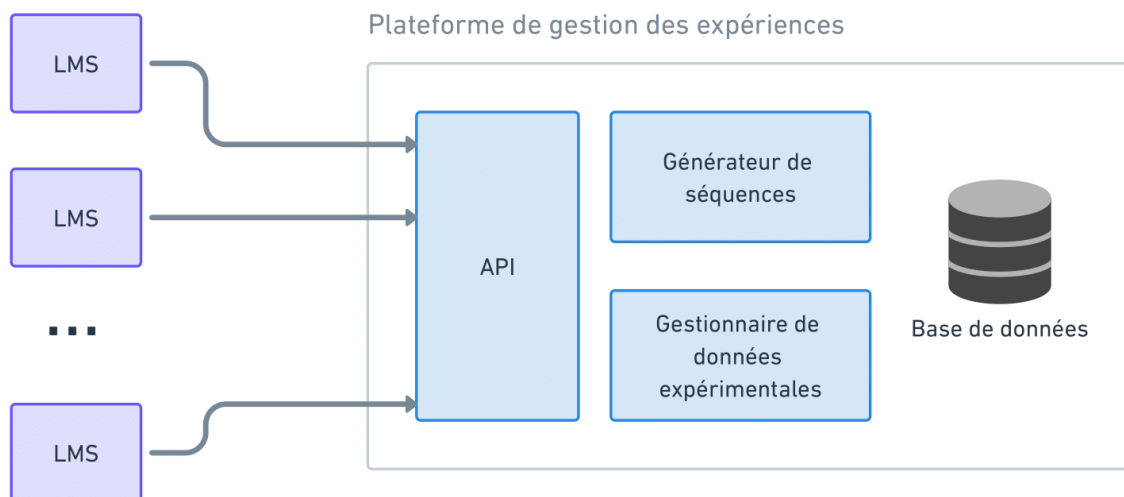
Un test dépend pédagogiquement de l'enseignement des concepts testés, et du test précédent.

Concrètement, notre générateur de séquences pédagogiques génère une liste d'activités, chaque activité étant une table de hachage comprenant :

- le timestamp qui correspond au moment où elle est disponible à la consultation, ie. à la somme de  $t_i^s$  et du timestamp de départ de la séquence.
- le timestamp correspondant au moment elle n'est plus disponible à la consultation, ie. à la somme de  $t_i^e$  et du timestamp de départ de la séquence.
- le type d'activité, et les données associées (nom, id des questions dans le cas d'un quiz, paramètres divers...)
- si l'activité n'est pas disponible pour tout le monde, le groupe concerné par l'activité est également indiqué. Les étudiants non concernés par l'activité ne verront pas apparaître cette activité dans leur espace de cours, afin de ne pas biaiser l'expérience.

De cette liste, on peut déduire une séquence pédagogique pour chaque groupe.

## 2.2 Architecture globale



**Figure 6 : Représentation globale du système**

La figure 6 représente une représentation globale du système, on y voit plusieurs plateformes LMS et une « plateforme de gestion des expériences ».

La plateforme de gestion des expériences est chargée de fournir aux LMS la fonction de génération de séquences pédagogiques, mais également le stocker et de traitement des données expérimentales. Les plateformes LMS accèdent aux services offerts par la plateforme via une API REST.

Nous avons choisi de séparer la logique de génération et des données expérimentales du client sur le LMS pour quatre raisons :

- **La centralisation des données expérimentales et des traitements sur la donnée** garantit une gestion simplifiée et uniforme. En regroupant toutes les informations en un seul lieu, on élimine la complexité liée à la coordination de multiples intervenants et à la préparation disparate des données. C'est ainsi que l'on assure la qualité des données et la fiabilité de l'évaluation des stratégies de testing, qui est le cœur de ce projet. Ainsi, l'accord devrait stipuler que l'établissement s'engage à fournir ses données expérimentales sous une forme pseudonymisée à des fins de recherche, en échange d'un accès gratuit au logiciel.
- **La performance des plateformes LMS** : ces plateformes peuvent ne pas être adaptées pour gérer des traitements d'envergure, particulièrement le traitement des données expérimentales pour des cours à fort effectif. Il est donc préférable d'effectuer ces traitements sur un serveur dédié et correctement dimensionné.
- **L'agnosticité vis-à-vis des LMS** : Le logiciel doit être conçu pour intégrer diverses plateformes, et pas uniquement Moodle. Des systèmes comme Canvas et Blackboard, très répandus, devraient pouvoir être intégrés sans problème. Il ne devrait pas être nécessaire de refondre toute la logique du logiciel à chaque intégration, mais simplement d'adapter l'interface et la logique d'incorporation des activités générées dans les espaces de cours propres à chaque LMS.
- **Développement communautaire et modularité** : Certaines fonctionnalités de génération, dont le détail sera fourni en section 2.3, sont conçues pour permettre des extensions développées par la communauté et pour être aisément intégrées sur la plateforme. En délocalisant cette logique du côté serveur plutôt que du côté client, on évite la nécessité de mettre à

jour constamment le plugin à chaque contribution de la communauté, même minime. Cela non seulement préserve la stabilité de la plateforme LMS, mais allège également la charge de gestion pour les administrateurs.

## 2.3 Générateur de séquences

Le générateur de séquence génère une séquence pédagogique à partir du protocole décrit et du contenu fourni.

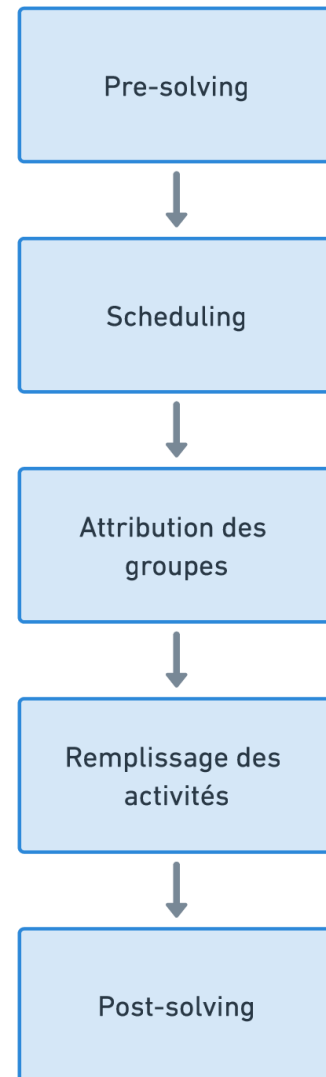
### 2.3.1 Découpage du processus de génération

Ce processus se déroule en cinq étapes, comme illustré par la figure 7

- **Préparation initiale (Pre-solving) :**  
Cette phase concerne les traitements préalables effectués sur les données en amont du processus de génération. Ces traitements, qui peuvent être combinés et séquencés, incluent par exemple la randomisation de l'ordre des groupes ou des questions au sein des quiz.

Dans l'expérience de Sana & Yan, le premier groupe à être évalué de façon intercalé, et vice-versa, est tiré aléatoirement. Il faut donc que l'ordre des groupes dans la liste des groupes soit tiré aléatoirement par le générateur.

- **Planification (Scheduling) :** À cette étape, la séquence pédagogique est définie, et chaque activité est planifiée. On détermine ainsi les moments



**Figure 7 : Représentation des étapes du processus de génération de séquences pédagogiques**

d'ouverture et de fermeture des activités en tenant compte des contraintes temporelles du protocole. Par exemple, pour reproduire un effet d'intercalage, un test unique est prévu par semaine et par groupe, fixé à un moment précis, comme le vendredi à 17h, pour une durée définie, telle que trente minutes. De plus, des informations essentielles pour les étapes suivantes sont préchargées durant cette phase, à l'instar des passes d'un compilateur.

- **Paramétrage des activités selon le groupe:** Lors de cette phase, chaque activité est adaptée en fonction de l'indice du groupe auquel est affecté l'activité, de l'indice de la période à laquelle fait partie l'activité et d'une matrice d'attribution. Dans le cadre de la reproduction de l'effet d'intercalage, cela pourrait consister à définir si les questions doivent être présentées de manière intercalée ou bloquée.
- **Remplissage des activités:** Ici, la logique intrinsèque de chaque activité est mise en œuvre. Par exemple, les questions sont ajoutées aux quiz selon l'ordre déterminé à l'étape précédente (bloqué ou intercalé dans le cas d'usage courant). Il peut exister d'autre type de contraintes comme nous allons le voir dans la section 3.1.2.
- **Post-traitement (Post-solving) :** Enfin, cette phase concerne les ajustements et traitements finaux appliqués aux données générées. Par exemple, une question pourrait être ajoutée à tous les quiz pour recueillir le ressenti des étudiants sur le questionnaire. Ces traitements, comme ceux du pré-solving, peuvent être combinés et appliqués séquentiellement.

### 2.3.2 Structures des données entrantes

Pour faire fonctionner le générateur, il faut fournir un certain nombre d'informations, que voici :

- **La méthode d'implémentation choisie :** il peut s'agir d'une implémentation sans mesure d'effet, avec un seul groupe et une seule configuration de test, ou bien d'une implémentation avec mesure d'effet.



- **Les traitements en amont (pre-solving)** : ces traitements consistent en une série d'opérations préliminaires. Chaque traitement est représenté par une table de hachage qui spécifie son nom et les paramètres correspondants. L'ordre des traitements dans la liste détermine la séquence d'exécution par le générateur.
  
- **Les traitements en aval (post-solving)** : Ces traitements interviennent après la génération principale et permettent d'affiner ou d'ajouter des éléments aux données générées. Tout comme pour le pre-solving, chaque traitement est décrit par une table de hachage contenant son nom et les paramètres associés. La succession des traitements dans la liste indique l'ordre dans lequel ils seront appliqués par le générateur.
  
- **Liste des groupes** : elle joue un rôle fondamental dans la configuration du générateur. Chaque groupe est identifié soit par son nom, soit par son identifiant, en fonction des spécificités de la plateforme utilisée. L'ordre dans lequel ces groupes sont listés détermine directement le type de traitement ou de configuration de test qu'ils recevront. Ainsi, par exemple, dans le cadre de l'étude menée par Sana & Yan, le premier groupe pourrait démarrer avec un quiz intercalé, tandis que le groupe suivant débiterait avec un quiz bloqué. C'est pour cette raison que les étapes de pre-solving, comme la randomisation de l'ordre des groupes, sont cruciales pour garantir le respect du protocole que l'on souhaite implémenter.
  
- **Paramétrage des tests** : Ce point concerne la personnalisation des tests via une table de hachage. Parmi les réglages possibles :
  - **Ordre des questions** : Par exemple, l'option d'un agencement intercalé ou bloqué, particulièrement dans le contexte de reproduction de l'effet d'intercalage.
  - **Feedback aux utilisateurs** : Ce feedback, le retour fait à l'utilisateur sur sa réponse, peut être instantané après chaque réponse ou être délivré à la fin du test.
  - **Quantité de questions par page** : Un paramètre à la fois scientifique et pratique. Dans des établissements comme l'université du Mirail, pour des cours à large effectif, limiter le

nombre de questions par page est essentiel pour éviter de surcharger la plateforme, surtout lorsque des milliers d'étudiants accèdent simultanément. Ce paramètre peut aussi biaiser le résultat.

- Options de navigation : Offrir ou restreindre la possibilité pour les utilisateurs de revenir sur des questions précédentes. Si, pour l'effet d'intercalage, il est nécessaire de restreindre la navigation vers l'avant, d'autres protocoles pourraient nécessiter un comportement différent.

Le générateur peut par ailleurs prendre en charge des paramètres qui ne font pas partie de cette liste si le besoin venait à émerger.

- **La matrice d'attribution inter-cycle** : elle représente une structure essentielle pour déterminer la configuration de test attribuée à chaque groupe durant chaque période. Sa finalité principale est d'orienter la manière dont les configurations de tests sont réparties à travers les groupes et les périodes, permettant ainsi de mieux organiser et contrôler l'expérimentation.

Formellement, la matrice  $M$  est définie comme suit :

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k1} & m_{k2} & \dots & m_{kn} \end{bmatrix}$$

où :

- $m_{ij}$  est l'élément de la matrice situé à la  $i$ -ème ligne et  $j$ -ème colonne. Sa valeur indique la configuration de test attribuée au  $i$ -ème groupe durant la  $j$ -ème période. Si elle est nulle, alors aucun test n'est prévu.
- $n$  représente le nombre total de périodes du cycle. Généralement deux, mais possiblement plus.
- $k$  représente le nombre total de groupes. Nous supposons que sa valeur est de deux dans le code de l'interface, mais ne limitons pas notre modèle à deux.

Prenons un exemple basé sur l'effet d'intercalage de Sana & Yan

$$M = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Cela signifie que, lors de la première période, le premier groupe reçoit la configuration de test 1, tandis que lors de la deuxième période, il reçoit la configuration de test 2. Symétriquement, le second groupe reçoit la configuration de test 2 durant la première période, et la configuration de test 1 durant la seconde.

Un exemple supplémentaire serait :

$$M = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \end{bmatrix}$$

Ici, aucune configuration de test n'est attribuée à l'un ou l'autre des groupes lors de la troisième période, indiquée par la valeur 0.

En conclusion, la matrice d'attribution inter-cycle sert de guide pour la distribution des configurations de tests à travers les périodes et les groupes, permettant une expérimentation systématique et organisée.

- **Le contenu des cycles** : chaque période au sein d'un cycle est définie par des spécificités claires. Il s'agit tout d'abord de fournir les contraintes temporelles, incluant la date de début et la durée de la période. Ensuite, on précise les activités programmées pour cette période. Par exemple, pour l'étude sur l'effet d'intercalage, cela impliquerait de mentionner le jour de la semaine et l'heure précise à laquelle le quiz est prévu lors de la dernière semaine.

Au-delà des contraintes temporelles et des activités, le contenu des cycles inclut également la définition des concepts qui seront abordés pendant cette période. Chaque concept est lié à un contenu précis, comme des questions spécifiques. Pour l'effet d'intercalage, cela se traduit par la liste des identifiants des questions associées à chaque concept, ordonnés dans l'ordre souhaité (ordre intra-concept) par l'équipe pédagogique. Ce niveau de détail assure une parfaite adéquation entre le contenu pédagogique et la méthodologie de l'étude.

Ces informations sont consolidées dans une table de hachage, l'objectif étant de pouvoir les représenter sous format JSON. Ceci vise à promouvoir

l'interopérabilité et la modularité du code. L'intention principale est d'éviter une rigidité superflue du système.

## **2.4 Gestion des données d'expérience**

### **2.4.1 Architecture de l'envoi des données**

Dans le cadre des expériences menées sur la plateforme, un enjeu crucial se présente une fois ces dernières achevées. Il s'agit de la nécessité de transférer, de manière fiable, les données d'expérience accumulées sur la plateforme LMS vers la plateforme dédiée à leur analyse et leur traitement. Cette opération de transfert est primordiale pour garantir l'intégrité des données, leur exploitation optimale et la pérennité des résultats de l'expérience.

Lors de la mise en œuvre d'une expérience sur la plateforme LMS, la préparation initiale et la configuration de l'expérience ne sont donc que les premières étapes d'un processus plus global.

Deux propriétés fondamentales doivent être scrupuleusement respectées par le système en charge de la réception des données : la vivacité et la correction.

La vivacité garantit que chaque demande de transfert des données finit par être exécutée. Cela assure que tous les utilisateurs de la plateforme bénéficient d'un service égal.

Quant à la correction, elle s'assure que chaque traitement est soit mené à bien dans son intégralité, soit, en cas de défaillance, reprogrammé pour une nouvelle tentative.

Dans la pratique, ces propriétés sont garanties par l'utilisation d'un mécanisme de file d'attente et d'un mécanisme d'accusés de réception, dont le dernier sera décrit section 2.4.3.

Nous avons anticipé cette logique en amont, bien qu'à l'état de preuve de concept cette logique d'industrialisation ne fasse pas encore de sens.

Le serveur fournirait trois procédures aux plateformes LMS pour initier un transfert :

- **L'inscription et la désinscription à la liste d'attente** : une liste d'attente doit organiser les expériences qui souhaitent transmettre leurs données expérimentales.
- **La procédure de polling** : une fois inscrit en liste d'attente, le client doit régulièrement demander au serveur s'il est prêt à recevoir le transfert.

Le serveur répond en fonction de ses disponibilités (possibilité de traiter N transferts à la fois) et de la liste d'attente (les premiers arrivants sont traités en premier).

Un besoin existant pourrait cependant mener à une congestion : les administrateurs Moodle pourraient vouloir limiter le transfert des données aux plages horaires creuses pour éviter de dégrader les performances de la plateforme lors des heures de cours, c'est pourquoi la liste d'attente ne suffit pas, il faut associer à cette liste d'attente un ensemble avec les clients « actifs », qui ont appelé la fonction de polling il y a moins de n minutes.

Ainsi le serveur ne prend en compte dans la liste d'attente que les serveurs actifs, afin d'éviter qu'un serveur inactif n'empêche les opérations de se dérouler.

Le pseudo-code associé à cette logique est en annexe. Celui-ci n'a pas implémenté, le projet étant encore à l'état de preuve de concept, mais une première version de l'algorithme a été conçu dans une optique de préparation au passage en production.

#### 2.4.2 Structure des données d'expérience

Lors de la conduite d'une expérience impliquant diverses configurations de tests et plusieurs groupes, l'obtention de résultats à la fois exploitables et compréhensibles est cruciale. L'ambition est d'avoir de pouvoir analyser clairement la performance des étudiants dans une configuration de test ou une autre en variable intra-groupe. Il faut aussi pouvoir filtrer les étudiants, c'est-à-dire les exclure du calcul en cas de comportement biaisé, par exemple un manque d'assiduité sur les tests.

Il est nécessaire de disposer de plusieurs matrices pour analyser efficacement les données. La première matrice doit représenter le score individuel de chaque étudiant pour chaque question et chaque test, sans tenir compte spécifiquement

de la configuration ou du groupe. La seconde matrice devrait démontrer comment chaque étudiant a été évalué sur chaque concept, identifiant la configuration spécifique utilisée.

D'autres matrices peuvent compléter cet ensemble pour affiner l'analyse, notamment celles portant sur l'engagement des étudiants dans les activités pédagogiques : leur présence en classe, l'interaction avec les contenus pédagogiques sur la plateforme LMS, etc.

Ces matrices doivent être conçues pour s'adapter à N configurations de tests et K groupes différents.

En fusionnant ces différentes matrices, nous pourrions collecter toutes les informations requises pour comprendre l'impact d'une configuration de test donnée.

#### **2.4.3 Protocole d'envoi des données de résultat**

Le calcul des matrices et leur envoi au serveur est à la charge de la plateforme LMS.

Cependant, une expérience de deux cycles de 4 semaines avec un effectif de 1000 étudiants atteint plus de 2 Mo, dépassant la limite de Django si toutes les matrices sont envoyées dans une même requête POST.

Moodle ne prévoyant pas la possibilité d'ouvrir un canal bidirectionnel type websocket, nous avons imaginé un protocole très basique fonctionnant uniquement en HTTP.

Celui-ci est très simple :

1. Le client génère les matrices et les enregistre dans différents fichiers qui sont ensuite compressées en un fichier. Ensuite il fait du polling jusqu'à obtenir l'accès à la téléversion des données.
2. Une fois que le serveur indique être prêt, le client doit créer un lien de téléchargement unique du fichier compressé, et le communiquer dans une requête POST qui indique également le checksum du fichier compressé, la liste des matrices et le checksum de chaque matrice.
3. LE

Il est important de noter qu'en utilisant HTTP, la connexion est stateless, donc chaque requête doit contenir suffisamment d'informations pour que le serveur sache à quelle session elle appartient. Cet identifiant de session est par ailleurs l'identifiant utilisé pour identifier l'expérience dans la file d'attente des traitements du serveur.

Notons également qu'en cas d'attente trop longue, le serveur peut décider de reporter le téléversement et de mettre la session en dernière place de la liste d'attente : le client devra réobtenir un accès.

### 3 Mise en oeuvre

Cette section fournira un aperçu concis de la mise en application pratique du logiciel.

#### 3.1 Générateur de séquences

Le générateur de séquences a été développé en langage de Python. À l'intérieur d'un fichier nommé "engine.py", on trouve une classe dédiée, la SequenceEngine. Cette classe encapsule l'ensemble de la logique associée à la génération de séquences.

Lorsque l'API reçoit une demande, elle instancie la classe SequenceEngine.

##### 3.1.1 Pre-solve factory et post-solve factory

Afin de garantir une exécution séquentielle et ordonnée des étapes de pré-traitement (pre-solving) et de post-traitement (post-solving), deux classes centrales ont été conçues : PresolveFactory et PostsolveFactory. Ces "factories" (usines, en français) assurent la création et la gestion des différentes instances de traitements. Tous les traitements de pré-traitement implémentent une interface commune, tout comme le font les traitements de post-traitement, garantissant ainsi une uniformité et une interchangeabilité entre eux.

Cette approche s'inspire du design pattern "Factory Method" (Méthode de Fabrication). Ce modèle de conception permet de définir une interface pour créer un objet, mais c'est aux sous-classes de décider quelles classes instancier, offrant ainsi une flexibilité dans le choix des classes à créer : on peut ainsi enchaîner plusieurs classes pour donner le comportement souhaité.

### 3.1.2 Gestions des contraintes

Comme mentionné précédemment, l'ajout de nouvelles contraintes est conçu pour être simple et fluide. Nous envisageons également d'ouvrir cette capacité à la communauté en lui permettant de publier ses propres contraintes directement sur le serveur.

Ces contraintes sont enregistrées en base de données et chaque contrainte est définie par :

- **Sa catégorie** : Par exemple, "questions\_order" concerne l'intercalage et les quiz verrouillés. Une autre catégorie pourrait être "scheduling\_constraint", pouvant accueillir la pratique espacée. Cette dernière stipule qu'un concept doit être abordé une première fois, puis repratiqué après un certain délai.
- **Son identification** : Chaque contrainte possède un nom court, et un nom long traduit dans les langues prises en charge par la plateforme, actuellement le français et l'anglais.
- **Son implémentation** : Il s'agit du nom de la classe qui matérialise et met en œuvre la logique associée à cette contrainte.

## 3.2 Plug-in moodle

La seule plateforme LMS pour l'instant supportée (pour laquelle on a un besoin) est Moodle, le client n'est donc implémenté que sur cette plateforme. Il est développé sous forme d'une activité de cours, d'un "module" dans le langage Moodle.

### 3.2.1 Le formulaire

L'exigence de flexibilité et d'interactivité était au cœur du design du formulaire. Deux modes distincts étaient nécessaires : une "simple implémentation" et une "implémentation avec mesure des effets". Chacun de ces modes requérait une série de champs différents, tout en conservant les informations précédemment renseignées lors du basculement d'un mode à l'autre.

Par ailleurs, la possibilité d'ajouter ou de retirer des cycles devait être fluide et intuitive. Bien que Moodle soit une plateforme robuste, elle n'offrait pas



d'option correspondant à nos critères en matière d'expérience utilisateur. Les formulaires standard de Moodle sont, de notre point de vue, alambiqués, moins intuitifs et manquent de réactivité.

Ainsi, nous avons opté pour une approche innovante en intégrant une interface Vue.js directement au sein d'un formulaire Moodle traditionnel. Cette stratégie impliquait l'usage d'un champ caché dans le formulaire Moodle. Ainsi, chaque modification apportée au formulaire depuis l'interface Vue.js est traduite en une chaîne de caractères JSON, qui est ensuite assignée comme valeur du champ caché.

L'utilisation de Vue.js impliquait la nécessité de transpiler le code javascript du format ECMAScript 2015+ vers un format compatible avec le navigateur. Cela est réalisé en utilisant webpack, un « bundler » javascript dont le rôle est de transpiler l'arborescence de fichiers vers un script Javascript unique exécutable par le navigateur, et nécessite d'inclure cette logique dans le processus de déploiement continu déjà en place.

Moodle fournit aux plug-ins des modules AMD (Asynchronous Module Definition), qui sont des spécifications permettant le chargement asynchrone de modules JavaScript, assurant l'accès via JavaScript à des fonctionnalités cœur de la plateforme tel que l'accès aux chaînes de caractères linguistiques ou au système de notifications par exemple.

Le fichier de configuration webpack doit donc prévoir des liens symboliques vers ces modules externes, ce qui implique par ailleurs que le script ne peut fonctionner que dans le contexte d'une plateforme Moodle. Ce fichier sera ajouté en annexe.

Ce fonctionnement n'est pas officiellement prévu par les équipes Moodle, mais il existe au moins un plug-in, nommé « Differentiator local plugin », qui utilise Vue.js et qui a reçu la validation des équipes Moodle pour apparaître dans le répertoire officiel des plug-ins : créant ainsi une jurisprudence.

Concernant le design, mon premier prototype s'écartait de l'apparence traditionnelle de Moodle. Avec l'aide de la librairie primevue, qui propose des éléments de formulaire fonctionnels, personnalisables, et stylisés. J'ai d'abord conçu un encadré (délimité par un fond légèrement plus sombre que le fond de la page) au sein du formulaire Moodle dans lequel était présentée une interface moderne, esthétique et ergonomique. Cependant, j'ai rapidement constaté que ce

design, bien qu'élégant, pouvait désorienter les utilisateurs habitués à l'interface de Moodle. Face à ce constat, nous avons décidé d'adapter notre interface au style du formulaire de manière à ce qu'il s'harmonise mieux avec le thème de Moodle.

Ce parti pris est risqué, il est en effet difficile de prévoir les effets de bords de chaque thème Moodle en circulation : peut-être que l'un d'entre eux pourrait déformer notre formulaire chez un utilisateur. De plus, l'encadré personnalisé et moderne était plus facilement réutilisable dans l'optique d'un développement pour un LMS non-Moodle.

#### ☑ **Adding a new Experience**

[Expand all](#)

Experience name

Experience type

Real experience

Cycle template

**Quiz config 1**

Name of quiz

Quiz bloqué

Quiz type

Blocked concepts

How questions behave

Deferred feedback

Questions per page

Every 10 questions

**Quiz config 2**

Name of quiz

Quiz intercalé

**Figure 8 : Capture d'écran extraite du formulaire de création d'une expérience**

### 3.2.2 Génération des activités

La procédure de génération débute une fois l'expérience configurée. Une fois cette étape initiée, l'utilisateur doit se rendre sur la page dédiée à l'expérience en question, où un résumé de celle-ci est présenté. C'est sur cette même page que se trouve le bouton spécifique pour déclencher la création de la séquence pédagogique (ou sa suppression).

Suite à cette action, les activités correspondantes sont automatiquement générées et incorporées à l'espace de cours. Cependant, elles sont placées initialement

dans une section du cours qui demeure invisible pour les étudiants. Seuls l'enseignant et l'administrateur peuvent accéder à cette section. C'est alors à l'enseignant qu'incombe la tâche de réorganiser le contenu : il doit manuellement déplacer ces activités fraîchement générées vers les sections appropriées du cours, afin qu'elles soient accessibles aux étudiants au moment voulu.

Quiz bloqué

Questions order	Blocked concepts
Questions per page	Every 10 questions
How questions behave	Deferred feedback

Quiz intercalé

Questions order	Interleaved concepts
Questions per page	Every 10 questions
How questions behave	Deferred feedback

Cycle 1

Période n°1 du 21/08/2023

La période démarre le 21/08/2023 et dure une semaine.

Le quiz aura lieu le friday à 17:00 et durera 30 minutes.

Group A : testés en configuration Quiz bloqué.

Group B : testés en configuration Quiz intercalé.

Concepts étudiés

aire

cube

Période n°2 du 28/08/2023

La période démarre le 28/08/2023 et dure une semaine.

Le quiz aura lieu le friday à 17:00 et durera 30 minutes.

Group A : testés en configuration Quiz intercalé.

Group B : testés en configuration Quiz bloqué.

Concepts étudiés

cylindre

polygone

Cycle 2

Figure 9 : Capture d'écran extraite de la page dédiée à l'expérience

## ▼ Generated quizzes for démo Hidden from students

Quiz de la période 1

Available but not shown on course page

Opened: Friday, 25 August 2023, 7:00 PM  
Closed: Friday, 25 August 2023, 7:30 PM

Not available unless: You belong to Group A

Quiz de la période 1

Available but not shown on course page

Opened: Friday, 25 August 2023, 7:00 PM  
Closed: Friday, 25 August 2023, 7:30 PM

Not available unless: You belong to Group B

Figure 10 : Capture d'écran extraite de l'espace de cours lié à l'expérience, représentant une partie de la section de cours où les activités sont initialement disposées.

## 4 Recherche : intégration de la pratique espacée

Diverses stratégies ont été élaborées pour améliorer la rétention des connaissances à long terme. Parmi elles, la "pratique espacée" se distingue comme une méthode efficace. Elle consiste à répartir les sessions d'apprentissage ou les éléments à étudier sur des intervalles de temps, favorisant ainsi une meilleure mémorisation par rapport à des sessions ou éléments groupés. Cette technique met en lumière l'importance non seulement de la structure des sessions d'apprentissage, mais également de la manière dont on organise l'apprentissage dans le temps.

Dans le cadre de mon master, je souhaite trouver une solution pour intégrer les diverses contraintes basées sur cette stratégie dans notre logiciel.

#### 4.1 Motivation

Afin de comprendre l'effet de la pratique espacée, je me suis appuyé sur la méta-analyse intitulée " *The Effects of Spaced Practice on Second Language Learning: A Meta-Analysis*"<sup>[7]</sup>, rédigée par Su Kyung Kim et Stuart Webb de l'Université du Western Ontario. Cette analyse offre une vue d'ensemble rigoureuse des recherches menées sur la pratique espacée, spécifiquement dans le contexte de l'apprentissage d'une seconde langue (L2).

La recherche entreprise par Kim et Webb a permis d'identifier 37 études répondant à tous les critères d'inclusion, offrant ainsi une vision panoramique de la question. Le matériel étudié dans ces recherches était divers, allant des contenus verbaux tels que des paires de mots ou des faits, aux contenus visuels comme des images ou des vidéos, sans oublier des contenus éducatifs variés tels que des conférences ou des formules mathématiques. Les avantages de la pratique espacée ont été constatés dans un large éventail de ces supports, et dans différentes activités d'apprentissage de la L2, comme les tâches associatives par paires ou les activités d'écoute et de lecture.

Il est important de souligner que l'efficacité notable de la pratique espacée a été démontrée, surtout en ce qui concerne l'apprentissage du vocabulaire en L2. Toutefois, si la majorité des études se sont penchées sur cet aspect, un besoin de recherche supplémentaire se fait sentir concernant ses effets sur la grammaire et la prononciation en L2 (Page 51).

Faciliter le déploiement de stratégies de pratique espacée faciliterait la découverte de configurations efficaces pour des disciplines telles la grammaire, la prononciation, ou dans tout autre domaine non linguistique.

#### 4.2 Paramétrage de la récupération espacée

Nous allons étudier la conception d'un protocole par récupération espacée pour comprendre le défi technique que l'implémentation d'une telle stratégie représente.

Afin de n'étudier que les variables dont on a besoin pour faire évoluer le logiciel, je vais sélectionner, parmi les 37 études de la meta-analyse, les études qui répondent aux critères suivants :

- L'expérience est compatible avec le modèle cyclique décrit à la section 1.2.3. Notons que ce modèle ne rend pas nécessaire que les configurations soit testées en variable intra-groupe, elles peuvent très bien être testées en variable inter-groupes, ce qui est le cas de nombreuses études parmi celles qui ont été sélectionnées.
- L'expérience se déroule sur une période correspondant à une unité d'enseignement, et non pas sur le long court (plusieurs années par exemple)
- L'expérience peut être incarnée sur un espace de cours virtuel

11 études ont correspondues à nos critères, nous allons donc nous en servir comme de références.

#### 4.2.1 Variables

- **Intervalle entre les sessions (ISI) :** Dans le cadre de l'apprentissage, l'ISI fait référence au temps qui s'écoule entre deux sessions d'étude ou de pratique d'un même contenu. Cet intervalle peut varier de quelques secondes à plusieurs jours ou semaines. Les participants peuvent être exposés à différents ISI : soit un jour, soit une semaine. Les participants peuvent être divisés en deux groupes avec des ISI courts (intervalle de 3,3 jours) et des ISI longs (intervalle de 7 jours). Ces variations d'ISI permettent d'examiner comment la fréquence des sessions d'apprentissage influence la rétention et la maîtrise du contenu.

Dans un espacement égal, les intervalles entre les rencontres d'un élément donné sont constants (par exemple, 2, 2 et 2 minutes). Dans un espacement croissant (également connu sous le nom de répétition croissante), les intervalles entre les rencontres augmentent progressivement (par exemple, 1, 2 et 3 minutes).

L'apprentissage groupé (ou massé) fait référence à des sessions d'apprentissage concentrées en un seul bloc de temps, sans intervalles significatifs entre elles. En revanche, l'apprentissage espacé divise ces sessions sur des périodes plus longues, avec des pauses ou des intervalles entre elles. Par exemple, dans l'étude sur l'effet des distributions espacées et massées sur l'apprentissage du vocabulaire EFL, le groupe massé a reçu une session intensive d'apprentissage, tandis que le groupe espacé a eu plusieurs sessions réparties sur une semaine.

- **Intervalle de rétention (RI) :** L'intervalle de rétention se réfère à la période qui s'écoule entre la dernière session d'apprentissage et le moment où l'apprenant est testé sur le contenu étudié. Cet intervalle est crucial pour évaluer la durabilité de la mémorisation. Dans "*Distribution of Practice Effects in the Acquisition and Retention of L2 Mandarin Tonal Word Production*"<sup>[9]</sup>, les participants ont été soumis à des tests de rétention après des intervalles d'une semaine ou de quatre semaines, permettant d'évaluer la rétention à court et à long terme. De même, dans "*The lag effect in secondary school classrooms: Enhancing students' memory for vocabulary*", les élèves ont été testés soit 7 jours soit 35 jours après la dernière session d'apprentissage, offrant une perspective sur la rétention à une semaine par rapport à un mois après l'apprentissage.
- **Taille du bloc:** Dans l'étude "*OPTIMISING SECOND LANGUAGE VOCABULARY LEARNING FROM FLASHCARDS*", la "block size" désigne le nombre de mots à apprendre en une seule session. Par exemple, si 20 mots cibles sont répétés en cinq blocs de quatre mots chacun, la taille du bloc est de quatre. Si les 20 mots cibles sont répétés en un seul bloc de 20 mots, la taille du bloc est de 20. Ce paramètre examine comment la quantité de contenu présenté en une fois influence l'apprentissage.
- **L'intercalage :** L'intercalage introduit un espacement du même concept à l'intérieur d'un quiz. Cette variable fait donc partie des recherches sur le sujet. L'intercalage peut par ailleurs être d'ordre aléatoire. Dans « *The Synergistic Benefits of Systematic and Random Interleaving for Second Language Grammar Learning* » par exemple l'entrelacement aléatoire (ordre non régulier) est comparé à l'entrelacement systématique (ordre régulier).

#### 4.3 Recherche de solutions techniques

Du point de vue technique, et pour s'intégrer à un emploi du temps universitaire à charge réelle, mener de telles études sur les stratégies par récupération, très exigeantes concernant la planification, peuvent nécessiter de bouleverser l'emploi du temps d'une promotion ou d'un groupe.

En effet, le respect de l'intervalle entre les sessions et de l'intervalle de rétention est primordial pour mener à bien ces études.

Pour anticiper ce changement, il est nécessaire d'exprimer l'expérience sous forme de contraintes claires, et d'explorer les différentes possibilités de replanification automatique de l'emploi du temps. En se reposant sur « *A survey of approaches for university course timetabling problem* »<sup>[1]</sup>, nous allons explorer les possibilités qui nous sont offertes.

#### **4.3.1 Hard vs soft constraints**

Les contraintes d'emploi du temps sont classées en deux catégories : "soft constraints" (contraintes souples) et "hard constraints" (contraintes strictes).

Les "hard constraints" sont des exigences impératives qui doivent être absolument respectées pour qu'une solution soit considérée comme valide. Le non-respect de ces contraintes rend la solution inapplicable. Par exemple, s'assurer qu'un enseignant ne donne pas deux cours en même temps est une "hard constraint".

En revanche, les "soft constraints" sont des préférences ou des recommandations qui, bien que souhaitables, ne sont pas strictement nécessaires. Leur respect améliore la qualité de la solution, mais leur non-respect ne la rend pas invalide. Un exemple pourrait être la préférence d'un enseignant pour un créneau horaire spécifique.

Dans notre contexte d'étude, les "hard constraints" (contraintes strictes) sont essentielles pour garantir la précision et la pertinence de ce que nous cherchons à mesurer.

Par exemple, si notre objectif est d'avoir un Intervalle de Stimulation (ISI) fixe de  $n$  jours, ou si nous souhaitons un ISI croissant selon un motif spécifique, ce sont les "hard constraints" qui définissent ces critères, indépendamment des valeurs absolues.

Ces contraintes sont donc incontournables pour assurer la validité de l'étude. En revanche, les "soft constraints" (contraintes souples) sont des éléments flexibles qui, bien qu'ils puissent faciliter ou optimiser l'étude, ne sont pas indispensables à sa réalisation. Cela pourrait inclure des préférences telles que le choix d'une date précise pour commencer l'étude ou la détermination de la valeur



du premier ISI dans une étude à ISI relatif. Ces contraintes sont utiles, mais leur non-respect n'invalide pas l'étude.

#### **4.3.2 Approches pour résoudre le problème**

Selon le survey, plusieurs approches sont utilisées pour résoudre le problème de planification des cours universitaires (UCTTP). Voici un aperçu des différentes solutions :

- Méthodes de recherche opérationnelle : Ces méthodes comprennent des techniques basées sur la théorie de la coloration des graphes, la méthode IP/LP et les techniques basées sur les contraintes (CSPs). Ces méthodes sont particulièrement utiles pour des problèmes où une solution optimale est recherchée, mais elles peuvent devenir inefficaces lorsque la taille du problème augmente, car leur complexité augmente avec le nombre d'étudiants et d'universités.
- Méthodes métaheuristiques : Ces approches comprennent des méthodes basées sur une population, comme les algorithmes génétiques, et des méthodes basées sur une seule solution, comme la recherche tabou. Elles sont utiles pour explorer un grand espace de solutions et trouver une solution approximative en un temps raisonnable.
- Méthodes intelligentes et novatrices : Ces méthodes englobent des approches hybrides, basées sur l'intelligence artificielle, la théorie floue et les algorithmes de clustering. Elles sont utiles lorsque des connaissances spécifiques ou des adaptations sont nécessaires pour résoudre le problème.
- Approches multi-critères et multi-objectifs : Ces approches sont utiles lorsque plusieurs objectifs ou critères doivent être pris en compte simultanément pour obtenir une solution équilibrée.
- Approche basée sur les systèmes multi-agents distribués : Cette approche est particulièrement utile pour la planification d'événements communs entre départements en raison de sa scalabilité.

Chaque solution a ses propres avantages et est adaptée à des situations spécifiques en fonction des besoins et des contraintes exactes du protocole.

## **5 Réflexion sur mon développement personnel et professionnel**

### **5.1 L'écueil de la sur-conception**

Durant ce projet de fin d'études, j'ai eu l'opportunité de développer un logiciel de manière autonome. L'un des principaux écueils que j'ai rencontrés fut le piège de la "sur-conception". Cette tendance à élaborer des fonctionnalités non essentielles ou excessivement complexes par rapport aux exigences réelles du projet résulte souvent d'une aspiration à concevoir une solution universelle adaptée à tous les scénarios imaginables. Dans ma quête de généralisation, j'ai parfois perdu de vue les impératifs fondamentaux qui méritaient une priorisation.

Il m'a été difficile de cerner précisément les besoins spécifiques de l'UT1 et de l'UT2. Cela se manifeste par exemple par ma première approche qui m'a conduit à développer une interface d'apprentissage hors de Moodle, avant même de focaliser sur le générateur de séquences.

Cependant, grâce à de nombreux échanges, j'ai réussi à équilibrer ma vision à long terme du projet avec une démarche de développement itérative et alignée sur les besoins immédiats.

### **5.2 La communication dans le cadre d'un nouveau projet**

Lorsqu'on entame un nouveau projet, il n'est pas rare que le besoin initial soit flou. Face à cette incertitude, la communication devient primordiale. Il s'agit d'établir un dialogue efficace entre tous les intervenants du projet qui, en raison de leurs responsabilités et préoccupations diverses, peuvent avoir des vues différentes sur la tâche à accomplir.

J'ai ainsi compris l'importance d'ajuster ma communication selon les interlocuteurs et le stade d'avancement du projet. Cette adaptation m'a aidé à cerner avec précision ce qui est primordial à réaliser par rapport à ce qui est secondaire. J'ai également acquis la compétence de progresser malgré un horizon parfois obscur, en me concentrant sur la réalisation d'une fonction à la fois. Cela

a grandement facilité les échanges avec les acteurs métiers, mon tuteur et mon responsable de service.

### 5.3 Apprentissage technique

Sur le plan technique, ce projet m'a offert l'opportunité d'affiner mes compétences en développement avec Vue.js, une aptitude que j'avais initialement développée à l'ENSEEIH dans le cadre des enseignements de Jean-Cristophe Buisson en technologies web. De plus, j'ai approfondi ma compréhension de la plateforme Moodle. Étant donné le manque de documentation de Moodle, j'ai dû plonger dans son vaste code source pour progresser. Je suis convaincu que cette immersion enrichira considérablement mon parcours professionnel.

Mon maître d'apprentissage, Bruno Ilponse, a été un pilier essentiel de mon évolution. En plus de revisiter mon code et mon architecture, il m'a instruit sur l'utilisation optimale de l'outil de versioning, git. J'ai notamment découvert l'importance des "commit atomiques", qui consistent à segmenter mes changements en petits blocs logiques, chaque commit incarnant une fonctionnalité ou une amélioration particulière.

Je suis enthousiaste à l'idée de continuer à contribuer à ce projet en tant qu'acteur open-source, du moins jusqu'à ce que l'outil soit testé à l'Université de Toulouse et affiche ses premiers retours.

## 6 Bibliographie

1. Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86, 43-59. <https://doi.org/10.1016/j.cie.2014.11.010>.
2. Bloom, K. C., & Shuell, T. J. (1981). Effects of massed and distributed practice on the learning and retention of secondlanguage vocabulary. *Journal of Educational Research* 74(4), 245–248. <http://doi.org/10.1080/00220671.1981.10885317>
3. Carpenter, S. K., & Mueller, F. E. (2013). The effects of interleaving versus blocking on foreign language pronunciation learning. *Memory & Cognition*, 41, 671–682. <http://doi.org/10.3758/s13421-012-0291-4>

4. Kanayama, K., & Kasahara, K. (2016). The effects of expanding and equally-spaced retrieval practice on long-term L2 vocabulary retention. *ARELE: Annual Review of English Language Education in Japan*, 27, 217–232. [http://doi.org/10.20581/arele.27.0\\_217](http://doi.org/10.20581/arele.27.0_217)
5. Karpicke, J. D., & Bauernschmidt, A. (2011). Spaced retrieval: Absolute spacing enhances learning regardless of relative spacing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(5), 1250–1257. <http://doi.org/10.1037/a0023436>
6. Khoii, R., & Abed, K. F. (2017). Effects of equal spacing, expanding spacing, and massed condition on EFL learners' receptive and productive vocabulary retrieval. In Pixel, *Proceedings of ICT for language learning (19th Ed.)* (pp. 500–504). Florence, Italy: ICT for Language Learning. Retrieved from <https://conference.pixelonline.net/ICT4LL/files/ict4ll/ed0010/FP/0960-SLA2580-FP-ICT4LL10.pdf>
7. Kim, S.K. and Webb, S. (2022), The Effects of Spaced Practice on Second Language Learning: A Meta-Analysis. *Language Learning*, 72: 269-319. <https://doi.org/10.1111/lang.12479>
8. Lee, E., & Choe, M-H. (2014). The effect of spaced repetitions on Korean elementary students' L2 English vocabulary learning. *Studies in English Education*, 19(1), 55–75.
9. . Li, M., & DeKeyser, R. (2019). Distribution of practice effects in the acquisition and retention of L2 Mandarin tonal word production. *The Modern Language Journal*, 103(3), 607–628. <http://doi.org/10.1111/modl.12580>
10. Miles, S. W. (2014). Spaced vs. massed distribution instruction for L2 grammar learning. *System*, 42, 412–428. <http://doi.org/10.1016/j.system.2014.01.014>
11. Namaziandost, E., Nasri, M., Rahimi Esfahani, F., & Keshmirshekan, M. H. (2019). The impacts of spaced and massed distribution instruction on EFL learners' vocabulary learning. *Cogent Education*, 6:1661131, 1–10. <http://doi.org/10.1080/2331186X.2019.1661131>
12. Nakata, T. (2013). Optimising second language vocabulary learning from flashcards (Unpublished doctoral dissertation). Victoria University of Wellington, New Zealand

13. Pan, S. C., Tajran, J., Lovelett, J., Osuna, J., & Rickard, T. C. (2019). Does interleaved practice enhance foreign language learning? The effects of training schedule on Spanish verb conjugation skills. *Journal of Educational Psychology*, 111, 1172–1188. <http://doi.org/10.1037/edu0000336>
14. Sana, F., & Yan, V. X. (2022). Interleaving Retrieval Practice Promotes Science Learning. *Psychological Science*, 33(5), 782–788. <https://doi.org/10.1177/09567976211057507>

## 7 Annexes

Procédures d'accès au transfert des données expérimentales :

### **Variables :**

```

num[N] = {0, ...}    // Pour l'exclusion mutuelle
choosing[N] = {false, ...} // Pour l'exclusion
mutuelle
condition[N]          // Un tableau de conditions
ou de sémaphores pour l'exclusion mutuelle
LA = [] : FIFO          // Liste d'attente
pour le transfert
SD = [] : Set           // Sessions en demande
depuis moins de 5 min
SA = [] : Set           // Sessions actuellement
en train de transférer ses données
SLU = {} : HashMap      // Timestamp du dernier
polling de chaque expérience
N : Entier              // Nombre possible de
traitements simultanés

```

### **Procédure entrer\_exclusion\_mutuelle(int ID) :**

```

// Étape 1: Choix du numéro
choosing[ID] = vrai
num[ID] = max(num[0], num[1], ..., num[N-1]) + 1
choosing[ID] = faux

// Étape 2: Vérification

```

```

    pour j de 0 à N-1:
        si j == ID:
            continuer

        // Attendre que le processus j finisse de
        choisir un numéro
        si choosing[j]:
            attendre(condition[j])

        // Attendre que les autres processus aient
        un droit d'entrée
        tant que num[j] != 0 et (num[j], j) <
        (num[ID], ID):
            attendre(condition[j])

Procédure sortir_exclusion_mutuelle(int ID):
    num[ID] = 0
    pour j de 0 à N-1:
        signaler(condition[j])    // Réveillez tous
    les processus qui attendent ce processus

Procédure inscrire(expérience):
    entrer_exclusion_mutuelle(expérience)
    si ¬(expérience ∈ LA):
        LA := LA ∪ expérience
    sortir_exclusion_mutuelle(expérience)

Procédure désinscrire(expérience):
    entrer_exclusion_mutuelle(expérience)
    si expérience ∈ LA:
        LA := LA \ { expérience }
    si expérience ∈ SD:
        SD := SD \ { expérience }
    sortir_exclusion_mutuelle(expérience)

Fonction vérifierPrêtPourTransfert(expérience):

```

```

    si  $\neg$ (expérience  $\in$  LA):
        retourner
    entrer_exclusion_mutuelle(expérience)
    si  $\neg$ (expérience  $\in$  SD):
        SD := SD  $\cup$  expérience
        SLU[expérience] = maintenant()
        LA_SD := {e|e $\in$ LA et e $\in$ SD}
        si (longueur(SA) < N) et expérience  $\in$ 
LA_SD[0..(longueur(SA)-1)]:
            SA := SA  $\cup$  expérience
            LA := LA  $\setminus$  expérience
            SD := SD  $\setminus$  expérience
            sortir_exclusion_mutuelle(expérience)
        retourner PRÊT
    sortir_exclusion_mutuelle(expérience)
    retourner PAS_ENCORE_PRÊT

Procédure planificateur:
    toutes les minutes:
        entrer_exclusion_mutuelle(ID_ANY) //
Utiliser n'importe quel ID pour bloquer l'accès
        pour chaque expérience dans SD:
            si maintenant() - SLU[expérience] >= 5
min:
            SD := SD  $\setminus$  expérience
            sortir_exclusion_mutuelle(ID_ANY)

```

Fichier webpack.conf.js utilisé pour le plug-in Moodle :

```

const path = require('path');
const webpack = require('webpack');
const { VueLoaderPlugin } = require('vue-loader')

module.exports = {
    mode: process.env.NODE_ENV === 'production' ?
'production' : 'development',
    entry: './main.js',
    devtool : 'source-map',

```

```

output: {
  path: path.resolve(__dirname, "../amd/build"),
  filename: 'forms.min.js',
  chunkFilename: "[id].forms.min.js?v=[hash]",
  libraryTarget: 'amd',
},
module: {
  rules: [
    {
      test: /\.css$/,
      use: [
        'vue-style-loader',
        'css-loader'
      ],
    },
    {
      test: /\.scss$/,
      use: [
        'style-loader',
        'css-loader',
        'sass-loader',
      ],
    },
    {
      test: /\.vue$/,
      loader: 'vue-loader',
    },
    {
      test: /\.js$/,
      loader: 'babel-loader',
      exclude: /node-modules/,
    },
  ],
},
watchOptions: {
  ignored: /node_modules/
},
performance: {
  hints: false
},
externals: {
  'core/ajax': {
    amd: 'core/ajax'
  }
}

```



```

    },
    'core/str': {
      amd: 'core/str'
    },
    'core/modal_factory': {
      amd: 'core/modal_factory'
    },
    'core/modal_events': {
      amd: 'core/modal_events'
    },
    'core/fragment': {
      amd: 'core/fragment'
    },
    'core/yui': {
      amd: 'core/yui'
    },
    'core/localstorage': {
      amd: 'core/localstorage'
    },
    'core/notification': {
      amd: 'core/notification'
    },
    'jquery': {
      amd: 'jquery'
    }
  },
  performance: {
    hints: false
  },
  plugins: [
    new VueLoaderPlugin(),
    new webpack.DefinePlugin({
      __VUE_OPTIONS_API__ : true,
      __VUE_PROD_DEVTOOLS__ : false,
    })
  ],
};

```