Vers la génération automatique de tests d'évaluations différenciés et équitables en contexte universitaire

Richardson CIGUENE[1], Céline JOIRON[1], Gilles DEQUEN[1]
[1] Laboratoire Mis, Université de Picardie Jules Verne, 33 rue Saint-Leu - Amiens
richardson.ciguene@u-picardie.fr, celine.joiron@u-picardie.fr , gilles.dequen@u-picardie.fr

Résumé. Ce travail de recherche s'intéresse à la génération automatique de sujets d'évaluations en contexte universitaire. L'objectif est d'étudier la possibilité de générer automatiquement des séries de sujets d'examens portant sur un thème ou une discipline commune, ayant pour caractéristiques, d'une part d'être différents deux à deux, et d'autre part de garantir une certaine équité de l'évaluation. Par équité, nous entendons en particulier la garantie d'une équivalence dans les niveaux de difficulté de l'ensemble des sujets différenciés générés. Cet article présente les bases et la genèse de ce travail de recherche. Nous présentons l'approche que nous nous proposons d'adopter pour une mise en œuvre dans le cadre d'évaluations sommatives universitaires de types Questionnaires à Choix Multiples.

Mots-clés. Evaluations sommatives, génération automatique, QCM

1 Introduction

En contexte universitaire, le processus d'évaluation sommative des apprentissages comporte classiquement trois phases : la conception par l'enseignant d'un sujet d'évaluation, la composition par les apprenants (étudiants) sur des copies et enfin la correction de ces dernières par l'enseignant. L'usage de supports de composition numériques offre des facilités d'automatisation de tout ou partie de ce processus, en particulier dans le cadre d'enseignements destinés à de larges cohortes d'apprenants, qu'ils soient dispensés en ligne, en présentiel ou de façon plus hybridée.

Notre travail de recherche se focalise plus spécifiquement sur la question de la génération automatique de sujets d'examens, permettant d'assister l'enseignant dans l'élaboration de ses sujets, et ce notamment dans les situations suivantes :

- Différents sujets d'évaluations à produire pour différentes cohortes d'apprenants sur un même enseignement. Ce même enseignement peut être dispensé dans différentes filières ou encore d'une année sur l'autre.
- Différents sujets d'évaluations à produire pour une même cohorte d'apprenants et pour un même enseignement. Ce cas de figure s'illustre par exemple dans les formations relevant du contrôle continu ou encore lors d'examens multisessions.
- Différents sujets d'évaluation à produire pour une unique session d'examen. L'objectif affiché dans ce cas est d'être en mesure de limiter la fraude aussi bien dans le cadre d'une composition unique en amphithéâtre par exemple, que dans le

cadre d'une composition par sous-groupes en horaires décalés comme par exemple dans le cas de certaines épreuves du Certificat Informatique et Internet (C2I niveau 1).

L'usage de sujets d'examens différenciés peut engendrer des disparités dans les niveaux de difficulté associés. Ainsi, même si l'on considère un thème d'évaluation dans sa globalité, comment sommes-nous en mesure de garantir que chaque individu sera évalué avec un niveau d'exigence sensiblement équivalent ? Cette contrainte s'applique aussi bien d'une session à l'autre ou encore d'une cohorte à l'autre.

Lorsque l'on s'intéresse à un sujet d'examen de type Questionnaire à Choix Multiples construit à partir d'une base de questions candidates, si le nombre de questions et le nombre de choix associés à chaque question dans la base source est strictement égal à ceux devant figurer sur chaque sujet, un simple tirage aléatoire dans l'ensemble des arrangements peut suffire à garantir à la fois la différenciation et un niveau acceptable d'équité (si l'on met de coté l'influence que peut avoir l'ordre d'une série de question sur la perception de sa difficulté par un apprenant). En revanche, lorsque l'ensemble des questions candidates à la sélection est strictement plus grand que celui des questions figurant sur chaque sujet, garantir des niveaux de difficulté proches n'est plus si trivial. Si, pour une question donnée, seule les données d'entrée changent d'un sujet à l'autre, la solution peut rester relativement simple à mettre en oeuvre. En revanche, être en mesure d'offrir les mêmes garanties alors que l'énoncé change est plus ardu. Dans ce cadre, il convient de caractériser ce qui différencie le niveau de difficulté d'une question par rapport à une autre et d'un sujet par rapport à un autre.

Pour apporter des éléments de réponse à cette question, nous proposons de concevoir, développer et expérimenter un environnement informatique permettant la génération de sujets d'examens différenciés et garantissant une homogénéité des niveaux de difficulté, et ce indépendamment du support de composition. Notre approche consiste en une construction itérative de sujets "à la volée", en partant d'une base d'exercices source, construite au préalable par l'enseignant.

Cet article présente la conception d'un générateur automatique sur la base d'exercices de type Questions à Choix Multiples appelé DIFAIRT-G (DIfferent FAIR Tests Generator). Ce dernier s'inspire d'un générateur de sujets différenciés par tirage aléatoire et mélange de questions intégré dans un prototype de gestion des évaluations papier QCM appelé YMCQ (whY Multiple Choice Questionnaire). Aussi la section 2 présente la génese de DIFAIRT-G en lien avec YMCQ. La section 3 détaille l'approche adoptée pour DIFAIRG. Enfin nous concluons cet article sur les perspectives de nos travaux à court et moyen terme.

2 Genèse des travaux de recherche : YMCQ

Nos travaux de recherche trouvent leur origine dans le développement d'un prototype permettant la génération et la correction automatique de questionnaires à choix multiples papiers (YMCQ), développé par des enseignants chercheurs du laboratoire MIS pour un usage dans le cadre de leurs enseignements [1].

La chaine de processus de YMCQ s'appuie sur quatre étapes :

1. La conception d'une base de questions source et le paramétrage de la génération des sujets, étape réalisée par un enseignant ;
2. La génération automatique des sujets d'évaluation ;
3. La numérisation des copies (si besoin) ;
4. La correction automatique des compositions et la vérification de la correction par l'enseignant.

Dans YMCQ, la génération automatique permet la construction d'une série de sujets d'évaluation différenciés. Pour cela il s'appuie sur un ensemble de "questions sources", constituant des "modèles" à partir desquels sont construites des séries de questions à choix multiples. Chaque question source est composée d'un énoncé de question, puis de toutes les réponses possibles (bonnes ou mauvaises) envisagées par l'enseignant pour cet énoncé. Le nombre de réponses possibles associées à un énoncé de question n'est pas borné. Pour faciliter le travail de l'enseignant, et lui permettre de constituer sa base de questions sources, de naviguer dedans, et de lancer la génération automatique de ses sujets, une interface en php a été développée, comme l'illustre la figure 1 ci-dessous.

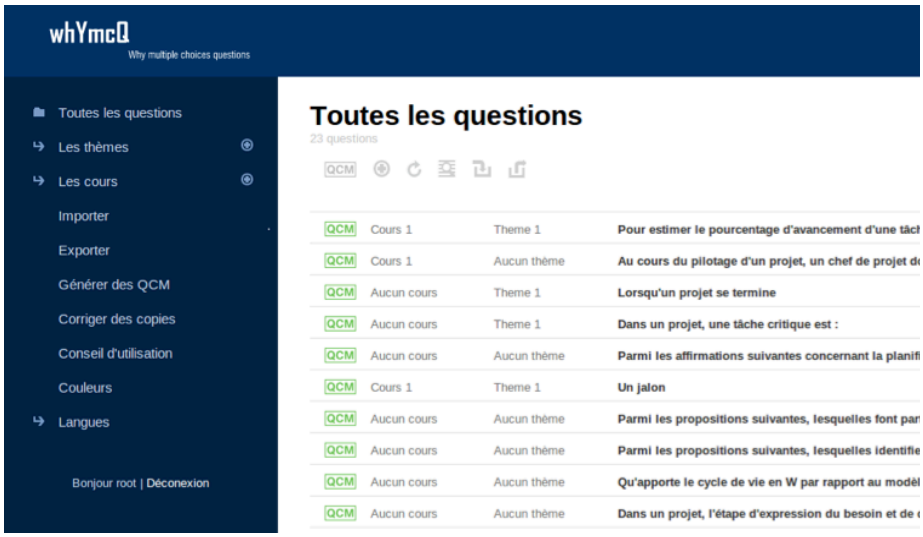

Figure 2. Interface de YMCQ - navigation dans les questions sources

Il est à noter qu'à cette phase de constitution de la base de questions sources s'associe naturellement une capitalisation des questions sources pour une réutilisation lors de différentes sessions d'examens.

La génération automatique des sujets d'évaluation débute ensuite par une phase de *paramétrage* permettant de dimensionner le futur examen. Ce dimensionnement consiste à positionner trois paramètres associés à l'évaluation d'un groupe d'apprenants à savoir : le nombre de sujets générés , le nombre de questions à choix multiples (noté ***n***) par sujet et le nombre de choix possibles par question.

La *génération* automatique suit, pour chacun des sujets, le processus suivant:

1. A partir de la base de questions sources, YMCQ procède à un tirage aléatoire sans remise de n énoncés de questions. L'ordre du tirage détermine l'ordre des questions. Pour chaque question q_i ($1 \leq i \leq n$) issue de ce tirage, la sélection de l'ensemble des choix multiples suit un processus quasi similaire.
2. Un premier tirage aléatoire et sans remise du choix noté x_{ref} est réalisé sur l'ensemble des réponses justes associées à q_i. Ce premier tirage garantit qu'au moins une réponse correcte est associée à q_i.
3. Au terme de cette première étape on procède à un tirage aléatoire sans remise dans l'ensemble des autres réponses associées à q_i afin de compléter l'ensemble de choix multiples associé.
4. A l'issue de cette sélection de choix multiples parmi lesquels au moins l'un d'entre est valide, YMCQ procède à un mélange aléatoire suivant l'algorithme de Fisher-Yates popularisé par Knuth [2].

Ce mode de génération offre une garantie quasi-certaine d'unicité des sujets. En effet, il est intéressant de noter que même pour une base de questions sources constituée d'exactement *20* questions auxquels sont associés *4* choix possibles de réponses, dont exactement 1 seule est correcte (ce qui constitue le minimum nécessaire à la construction d'une évaluation), le dénombrement des sujets possibles correspond, à titre d'exemple, à $1{,}46.10^{19}$ (si ces derniers nécessitent 20 questions à 4 choix de réponse), ce qui tend à montrer que l'individualisation est envisageable même lorsque le nombre de questions sources est réduit à son strict minimum. Remarquons toutefois que dans le cas présent, il n'est envisagé aucune mesure de 'différence' entre les sujets.

Le prototype YMCQ a été utilisé depuis trois ans sur plusieurs séries de sujets d'évaluation, et ce à différents niveaux d'enseignement universitaire. Aujourd'hui se pose la question d'aller plus loin dans l'approche de génération automatique et par là même d'évoluer d'une approche purement développement d'outil, vers une problématique scientifique relevant des EIAH et de la combinatoire : :

1 - Offrir davantage de garanties de différenciation : en effet, le seul tirage aléatoire ne permet pas de garantir une distance moyenne cohérente entre tous les sujets générés. A titre d'exemple, il est tout à fait possible d'obtenir deux sujets différents et qui auraient pour seul différence la place d'une seule des questions, ou encore le même jeu de questions, positionnées dans le même ordre, avec simplement un des choix de réponse qui est différent.

2 - Permettre une homogénéisation des niveaux de difficulté : en effet, la question de l'équité étant centrale dans les épreuves certificatives universitaires, nous souhaitons disposer d'un générateur automatique qui puisse tenir compte de la difficulté des sujets générés et notamment qui puisse générer une série consécutive de sujets présentant des niveaux de difficultés équivalents.

3 - Etre générique : en effet, le principe global de génération doit intégrer une partie "générique", entendons par là indépendante du type des exercices qui composent les sujets d'évaluation, et de ce fait ne pas limiter les principes de génération automatique aux seules questions à choix multiples.

La section suivante présente l'approche générale de DIFAIRT-G.

3 Vers la différenciation et l'équité : DIFAIRT-G

Générer des sujets d'évaluation différenciés en garantissant une homogénéité dans les niveaux de difficulté est l'objectif de DIFAIRT-G. Pour atteindre cet objectif, notre approche vise, d'une part, à définir des mesures structurelles. D'autre part, la caractérisation sémantique de certains éléments constituant les questions sources et les sujets générés est nécessaire. Il s'agit notamment des différentes dimensions qui permettront de caractériser la difficulté d'une question ou la difficulté d'un sujet d'évaluation dans son ensemble. Le problème de génération se fait alors sous contraintes.

Depuis les années 2000, un effort de normalisation des éléments pouvant constituer un sujet d'évaluation a été réalisé par le consortium international IMS-GLC (Instructional Management System – Global Learning Consortium) sous le nom de IMS-QTI (Question and Tests Interoperability). IMS-QTI représente un Test (un sujet d'évaluation) comme un ensemble pouvant contenir des Testparts, qui se décomposent en Sections, qui à leur tour contiennent des Items. Un item constitue alors la plus petite granularité d'un Test [3]. L'Item est un bloc contenant basiquement un énoncé à destination de l'apprenant et les réponses possibles ou proposées pour cet énoncé. Des outils auteurs peuvent être définis pour créer des items respectant le format QTI, des banques d'items peuvent être utilisées et échangées, et des outils permettant la construction automatique de tests d'évaluation (TestConstructionTools) peuvent être développés en respect avec cette norme [4].

Par ailleurs, IMS-QTI permet de faciliter l'échange des éléments constitutifs d'une évaluation entre différents environnements d'apprentissage, comme la plateforme Moodle par exemple [5]. Bien évidemment, certains chercheurs identifient des failles à IMS-QTI ou des faiblesses [6] [7] et certains en proposent des extensions [8]. Pourtant d'autres, pensent que QTI représente le meilleur point de départ pour tout système d'évaluation [9]. C'est le parti pris que nous avons choisi de suivre, en nous inspirant de l'approche de QTI en vue de faciliter l'interopérabilité de notre prototype avec différents systèmes. L'outil DIFAIRT-G peut ainsi être vu comme un "TestConstructionTool", qui a la spécificité de générer des sujets différenciés et équitables, constitués d'Items de type Questions à Choix Multiples. Cet outil est bien évidemment compatible avec le standard QTI. Il permet notamment d'utiliser des Items construits à partir de ce standard et en permettant également l'exportation d'items dans ce standard. Notons également qu'une autre de nos contraintes est que notre outil puisse également réutiliser les questions sources issus du logiciel YMCQ.

Ainsi, le cycle général de fonctionnement de DIFAIRT-G est décrit dans la figure 2. Nous retrouvons une base composée de tous les modèles d'Items de Test (TestItemPattern) pouvant être utilisés par le générateur. Dans le cas des QCM, les patterns d'items sont principalement constitués d'énoncés de questions, d'ensembles de réponses justes et d'ensembles de réponses fausses (tout comme les questions sources de YMCQ). De plus, dans le but de contribuer au calcul de mesures structurelles, d'autres paramètres sont venus enrichir les modèles d'items, tels que un identifiant unique, ou enfin un degré de pertinence de chaque réponse par rapport à l'énoncé de la question ou encore le thème du domaine d'enseignement dont relève le modèle d'item.

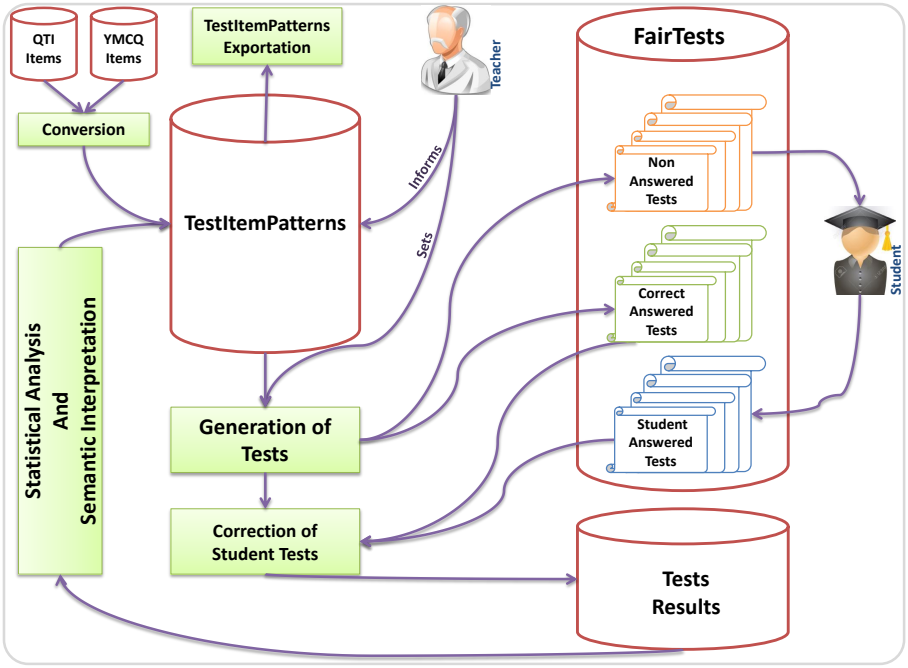

Figure 2. Cycle général de fonctionnement du système

Ainsi, le cycle de fonctionnement de DIFAIRT-G débute par la constitution de cette base de TestItemPatterns. Cette dernière peut être construite directement par l'enseignant, importée depuis des questions sources établies dans YMCQ ou encore depuis des items respectant le standard IMS-QTI. Vient ensuite la génération automatique des FairTests (tests équitables), paramétrée également par l'enseignant qui définit les éléments tels que le nombre de questions souhaité par sujet généré, le nombre de sujets différents souhaité ou enfin le niveau moyen de difficulté attendu. A partir de là sont créés en parallèle les sujet d'examen (NonAnsweredTest) et les corrigés de chacun des sujets générés (CorrectAnsweredTest). Une fois que l'étudiant a composé, les copies rendues par l'étudiant (StudentAnsweredTest) peuvent ensuite être réintégrés dans le système en vue d'une correction automatique[1]. Les résultats sont conservés dans la base des "TestResults". Cette dernière doit pouvoir être par la suite analysée sémantiquement et statistiquement en vue d'enrichir la base des TestItemPatterns d'indicateurs visant à affiner la caractérisation de la difficulté des questions construites à partir de ces TestItemPatterns.

Concernant plus précisément la génération automatique, elle s'appuie sur deux éléments constitutifs : la distance structurelle entre deux sujets, c'est-à-dire les écarts entre les séquences d'ItemTests composant 2 sujets différents générés; l'estimation du

[1] Ici le correcteur automatique intégré dans YMCQ peut être utilisé dans le cadre d'examens composés sur papier.

niveau de difficulté de chaque sujet généré. L'objectif de l'algorithme de génération est alors double : faire en sorte d'avoir une distance structurelle satisfaisante entre tous les sujets générés pris deux à deux; minimiser l'écart-type entre les mesures des niveaux de difficulté de sujets pris deux à deux (ce dernier point garantit une homogénéité de la difficulté). Bien évidemment, caractériser le niveau de difficulté d'une question, et à fortiori le niveau de difficulté d'un sujet, constitue un des enjeux majeurs de ce travail de recherche. Il nécessite de définir précisément et sémantiquement ce que représente une difficulté pour un apprenant, puis d'arriver à estimer la valeur de chacun des paramètres qui permettent de la caractériser, ce qui implique notamment de prendre en compte le contexte de l'évaluation. En effet, « *Un exercice n'a sans doute pas une difficulté intrinsèque, mais une difficulté statistique pour un certain type d'élèves à un certain moment de leur parcours* » [10]. Ce calcul reposer alors sur une caractérisation sémantique des TestItemPattern, de la notion de sujets d'examens et une modélisation de la notion de sujet d'examen.

5 Conclusion et perspectives

Dans cet article nous avons présenté le cadre d'un travail de recherche prospectif, concernant la génération automatique de sujets d'évaluations différenciées et équitables. Le générateur DIFAIRT-G combine une dimension sémantique pour la caractérisation du niveau de difficulté d'un sujet d'évaluation, et une approche combinatoire pour guider le processus de construction des sujets d'évaluation et garantir les contraintes de différenciation et d'équité. Les critères de différentiation des sujets d'évaluation générés, et le calcul d'une distance pondérée entre les sujets est en cours de construction. A court terme, il convient alors : d'approfondir la caractérisation de la notion de difficulté d'une question, de difficulté d'un sujet ; identifier les critères de différentiation de sujets d'évaluation ; étudier la combinatoire inhérente au processus de génération automatique et proposer un algorithme de génération intégrant la/les contrainte(s) d'équivalence de difficulté des sujets générés en fonction du nombre de questions candidates existantes, des informations sémantiques associées à chaque question. A moyen terme un premier prototype sera alors implémenté et expérimenté auprès de plusieurs enseignants en France et en Haïti.

Références

1. Joiron, C., Rosselle, M., Dequen, G., Le Mahec, G. : Automatiser la génération et la correction d'évaluations individualisées en contexte universitaire présentiel. In Environnements Informatiques pour l'Apprentissage Humain, EIAH2013, page poster, Toulouse, France, 29-31 mai 2013.
2. Knuth. : The Art of Computer Programming vol. 2 (3rd ed.). Boston: Addison–Wesley. pp. 145–146. 1998 ISBN 0-201-89684-2. OCLC 38207978
3. IMS-QTI – Question and Test Interoperability - http://www.imsglobal.org/question/qtiv2p1/imsqti_oviewv2p1.html (2015-03-27)

4. IMS-GLC – Spécifications – http://www.imsglobal.org/specifications.html (2015-03-30)
5. Kaustar, I. A. , Kubota, S., Musashi, Y., Sugitani, K. : Moodle XML to IMS-QTI Assessment Test Portability on Learning Management Systems. In The proceedings of The 7th ICTS, Bali, May 15th-16th 2013. (ISSN : 9772338185001)
6. Piotrowski, M. : QTI : A Failed E-learning Standard ? In F. Lazarinis, S. Green, and E. Pearson (eds.). Handbook of Research on E-Learning Standards and Interoperability : Framework and issues, p. 59-82. Hershey. PA, USA : IGI Global. 2011.
7. Durand. G. : Vers une scénarisation de l'évaluation en EIAH : L'évaluation scénarisable dans un dispositif de scénarisation pédagogique. In Rencontre Jeunes Chercheurs en EIAH de l'ATIEF. (2006)
8. Auzende, O., Giroire, H., Le Calvez, F. : Propositions d'extensions à IMS-QTI 2.1 pour l'expression de contraintes sur les variables d'exercices mathématiques. Jun 2007, INRP. <hal-001611375>
9. Radenkovic, S., Krdzavac, N., Devedzic, V. : A QTI Metamodel. In Proceedings of the Multiconference on Computer Science and Information Technology, pp. 1123 – 1132. (2007).
10. Auzende, O., Giroire, H., Le Calvez, F. : Quelles Caractéristiques utiliser pour stocker et rechercher des exercices ? Une réalisation. In Environnements Informatiques pour l'Apprentissage Humain, EIAH2009, Le Mans, France, 23-26 Juin 2009.