

## ÉTUDE DE CAS MOOCAZ AVEC UTL

### 1. Données mises à disposition

- Document de présentation des analyses réalisées sur COURSERA
- Jeux de données MOOCAZ V1 et V2
  - Traces
  - Structure de cours
- Descriptif du format de traces propre à EDX

### 2. Préparation des données

#### *2.1 Format de la structure de cours de MOOCAZ V1*

Il s'agit d'un format CSV utilisant la virgule comme séparateur. Il est composé des colonnes suivantes : p\_display\_name, p\_start, p\_url\_name, p\_folder, e\_url\_name, p\_parent\_url, e\_id\_video, e\_display\_name.

Il dispose d'une première ligne reprenant les noms des colonnes et d'une deuxième ligne contenant uniquement « assets » dans les colonnes p\_url\_name et p\_folder.

- p\_start est une date au format suivant : « 2014-05-12T12:00:00Z »
- les urls sont cryptées
- p\_folder est un champ dont les valeurs possibles sont : chapter, course, discussion, vertical, problem, sequential et pour une ligne « MOOCAZS01\_Trimestre\_2\_2014 » que je suppose être une erreur.
- p\_parent\_url est vide la plupart du temps.

=> Le format CSV étant pris en charge par l'environnement UTL directement, il n'a pas été nécessaire de prétraiter ces données.

#### *2.2 Format des traces de MOOCAZ V1*

Il s'agit d'un fichier de logs dont la structure est composée pour chaque ligne d'une date/heure au format « Apr 22 06:49:42 », de l'adresse d'une machine « ellms22.cines.fr », du mot clé « edx\_tracking: » et de la trace formatée en JSON.

Ce fichier n'est pas exploitable directement, nous avons réalisé un programme qui supprime les premiers éléments de la ligne de façon à ne conserver que les traces JSON, cette première partie étant considérée comme inutile dans notre contexte, ensuite les lignes JSON ont été transformées en XML grâce à une librairie qui réalise cette conversion. Il a été détectée 20 lignes qui ne pouvaient pas être transformées car tronquées de manière aléatoire. En accord avec le responsable du cas d'étude, nous avons exclu ces lignes de notre base.

=> Le fichier XML obtenu étant pris en charge par l'environnement UTL directement, nous avons pu l'intégrer.

#### *2.3 Format de la structure de cours de MOOCAZ V2*

Il s'agit d'un format CSV utilisant la virgule comme séparateur. Il est composé des colonnes suivantes : p\_display\_name, p\_start, p\_url\_name, p\_folder, e\_url\_name, p\_parent\_url, e\_id\_video, e\_display\_name.

Il dispose d'une première ligne reprenant les noms des colonnes et d'une deuxième ligne contenant uniquement « assets » dans les colonnes p\_url\_name et p\_folder.

- p\_start est une date au format suivant : « 2014-05-12T12:00:00Z », j'ai trouvé une fois la valeur « vertical » dans le champ.
- les urls sont cryptées
- p\_folder est un champ dont les valeurs possibles sont : chapter, course, discussion, vertical, problem, sequential et pour une ligne « 20002S02\_Trimestre\_3\_2014\_metadonnees » que je suppose être une erreur.
- p\_parent\_url est vide la plupart du temps.
- Particularité par rapport à MOOCAZ V1, tous les chapitres ont la même date de démarrage et ils sont beaucoup plus nombreux

Une erreur a été identifiée dans la structure de cours, en effet, 2 lignes sont mal structurées :

```
"Description du parcours
""Initiation"";"";""e571a57611134d56b32b499f9bf5669e",vertical,e1554b63884546f59e29875974c15744,,,,,
"Description du parcours
""Initiation"";"";""e571a57611134d56b32b499f9bf5669e",vertical,715ec79848384a6e94fd3da3203078ed,,,,,
```

On constate ici au début un séparateur ';' au lieu de ',', ce qui fausse la génération des données UTL. La correction a été faite à la main.

=> Le format CSV étant pris en charge par l'environnement UTL directement, il n'a pas été nécessaire de prétraiter ces données.

### 2.4 Format des traces de MOOCAZ V2

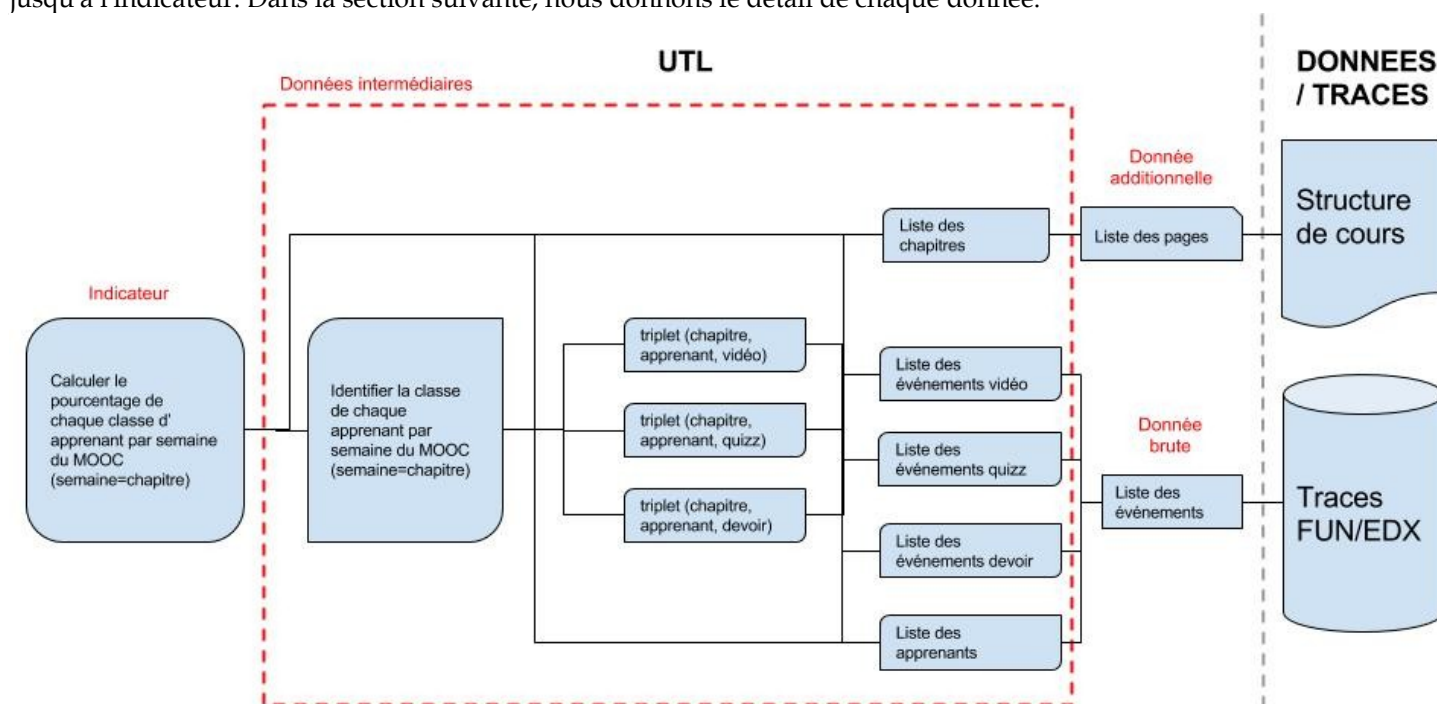
Il s'agit d'un fichier de logs dont la structure est composée pour chaque ligne d'une date/heure au format « Apr 22 06:49:42 », de l'adresse d'une machine « ellms22.cines.fr », du mot clé « edx\_tracking: » et de la trace formatée en JSON.

Ce fichier n'est pas exploitable directement, nous avons réalisé un programme qui supprime les premiers éléments de la ligne de façon à ne conserver que les traces JSON, cette première partie étant considérée comme inutile dans notre contexte, ensuite les lignes JSON ont été transformées en XML grâce à une librairie qui réalise cette conversion. Il a été détectée 20 lignes qui ne pouvaient pas être transformées car tronquées de manière aléatoire. En accord avec le responsable du cas d'étude, nous avons exclu ces lignes de notre base.

=> Le fichier XML obtenu étant pris en charge par l'environnement UTL directement, nous avons pu l'intégrer.

### 3. Description du processus d'analyse

Nous avons un graphique sur la page suivante décrivant le processus d'analyse qui a été pensé selon le fonctionnement de l'environnement UTL. Nous y retrouvons les données sources (traces ou autres), les différents types de données UTL jusqu'à l'indicateur.



### 4. Mise en œuvre dans l'environnement UTL

Travaux sur l'analyse

- Manque un opérateur distinct dans UTL permettant d'établir la liste des apprenants et des chapitres, une fonction a été ajoutée en attendant dans le projet JAVA du calculateur UTL.
- Pour l'extraction des traces liées à un type d'événement, on découvre que l'objet « event\_type » sur lequel nous devons nous baser n'est pas standardisé, en effet, il peut prendre comme valeur « problem\_check » ou « /courses/ENSCachan/20002/Trimestre\_2\_2014/modx/i4x://ENSCachan/20002/problem/cbd3067f2c7542bf9c2a28557ca3515/problem\_check »
- Est-ce que cela a du sens de prendre les 2 ou non? J'ai choisi de conserver les 2.
- Pour l'identification de la lecture de la vidéo, nous avons le type d'événement « play\_video » => OK

- Pour l'identification d'un quizz, nous avons le type d'événement « problem\_check » => OK
- Pour l'identification du chargement d'une vidéo, nous avons le type d'événement « load\_video » => OK
- Pour l'identification d'un devoir, nous avons le type d'événement « create\_submission » => OK

Nous décrivons maintenant les données conçues dans UTL.

#### 4.1 Indicateurs :

<b>I_LearnersClassification</b>	
<b>Description</b>	Indicateur permettant d'obtenir pour chaque chapitre le pourcentage et le nombre d'apprenants dans chaque catégorie.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des chapitres (ID_Chapters)</li> <li>• La liste des catégories d'apprenants par semaine (ID_ListLearnersCategorized)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre (classés par ordre croissant de date de démarrage) faire {   nbViewer = compter le nombre de « viewer » pour le chapitre courant ;   nbQuizzer = compter le nombre de « quizzer » pour le chapitre courant ;   nbActiveViewer = compter le nombre d'« active viewer » pour le chapitre courant ;   nbInactive = compter le nombre d'« inactive » pour le chapitre courant ;   total = nbViewer + nbQuizzer + nbActiveViewer + nbInactive ;   pViewer = (nbViewer * 100) / total;   pQuizzer = (nbQuizzer * 100) / total;   pActiveViewer = (nbActiveViewer * 100) / total;   pInactive = (nbInactive * 100) / total;   Enregistrer la liste des informations dans l'indicateur ; } </pre>
<b>Exemple</b>	<pre> &lt;chapter name="Introduction au cours" viewer="8.998607" cViewer="452" quizzer="0.25880948" cQuizzer="13" activeViewer="0.01990842" cActiveViewer="1" inactive="90.72268" cInactive="4557"/&gt; </pre>

#### 4.2 Données intermédiaires :

<b>ID_Chapters</b>	
<b>Description</b>	Donnée intermédiaire permettant d'obtenir la liste des chapitres distincts.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La structure du cours (AD_CourseStructure)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque nom distinct pour les éléments de type chapitre faire {   date = récupérer la date de démarrage pour l'élément chapitre courant ;   page = récupérer le code (hash) de la page pour l'élément chapitre courant ;   Enregistrer le nom, la date et le code de page dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<pre> &lt;chapter name="Introduction au cours" startAt="2014-05-12T12:00:00Z"&gt; 15ded21253eb45158b54c8840646fd96&lt;/chapter&gt; </pre>

<b>ID_ListLearnersCategorized</b>	
<b>Description</b>	Donnée intermédiaire permettant d'obtenir la liste des apprenants catégorisés (« viewer », « quizzer », « active viewer », « inactive ») par chapitre.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des chapitres (ID_Chapters)</li> <li>• La liste des apprenants (ID_Learners)</li> <li>• La liste indiquant le nombre de fois où l'on a détecté l'événement « play_video » pour un apprenant par semaine (ID_CountVideo)</li> <li>• La liste indiquant le nombre de fois où l'on a détecté l'événement « problem_check » pour un apprenant par semaine (ID_CountQuizz)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre (classés par ordre croissant de date de démarrage) faire {   Pour chaque apprenant faire {     cptv = récupérer le nombre d'événements « play_video » pour l'apprenant et le     chapitre courants ;     cptq = récupérer le nombre d'événements « problem_check » pour l'apprenant et le     chapitre courants ;     si ((cptv &gt; 0) et (cptq &gt; 0)) alors Enregistrer l'« active viewer » dans la donnée     intermédiaire ;     si ((cptv = 0) et (cptq &gt; 0)) alors Enregistrer le « quizzer » dans la donnée     intermédiaire ;     si ((cptv &gt; 0) et (cptq = 0)) alors Enregistrer le « viewer » dans la donnée     intermédiaire ;   } } </pre>

	<pre>         si ((cptv = 0) et (cptq = 0)) alors Enregistrer l'« inactive » dans la donnée intermédiaire ;     } } </pre>
<b>Exemple</b>	<pre> &lt;inactive learner="556731" chapter="Introduction au cours"&gt;556731&lt;/inactive&gt; &lt;viewer learner="384738" chapter="Introduction au cours"&gt;384738&lt;/viewer&gt; &lt;quizzier learner="373869" chapter="Introduction au cours"&gt;373869&lt;/quizzier&gt; &lt;activeViewer learner="650748" chapter="Introduction au cours"&gt;650748&lt;/activeViewer&gt; </pre>

<b>ID_Learners</b>	
<b>Description</b>	Donnée intermédiaire permettant d'obtenir la liste des apprenants distincts.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des événements du MOOC (RD_Events)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque apprenant (User) distinct dans les événements faire {     Enregistrer le code de l'apprenant dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<User>197094</User>

<b>ID_CountVideo</b>	
<b>Description</b>	Donnée intermédiaire permettant de compter le nombre de fois où l'événement « play_video » intervient par chapitre et par apprenant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des chapitres (ID_Chapters)</li> <li>La liste des apprenants (ID_Learners)</li> <li>La liste des événements « play_video » (ID_PlayVideo)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre faire {     Pour chaque apprenant faire {         cpt = compter le nombre d'événements « play_video » pour l'apprenant courant et pour lesquels la page concernée comporte le code du chapitre courant ;         Enregistrer les informations dans la donnée intermédiaire ;     } } </pre>
<b>Exemple</b>	<countVideo learner="575646" chapter="Introduction au cours">1</countVideo>

<b>ID_CountQuizz</b>	
<b>Description</b>	Donnée intermédiaire permettant de compter le nombre de fois où l'événement « problem_check » intervient par chapitre et par apprenant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des chapitres (ID_Chapters)</li> <li>La liste des apprenants (ID_Learners)</li> <li>La liste des événements « problem_check » (ID_ProblemCheck)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre faire {     Pour chaque apprenant faire {         cpt = compter le nombre d'événements « problem_check » pour l'apprenant courant et pour lesquels la page concernée comporte le code du chapitre courant ;         Enregistrer les informations dans la donnée intermédiaire ;     } } </pre>
<b>Exemple</b>	<countQuizz learner="373869" chapter="Introduction au cours">2</countQuizz>

<b>ID_PlayVideo</b>	
<b>Description</b>	Donnée intermédiaire permettant de filtrer la liste d'événements pour ne conserver que les événements « play_video », et pour chaque événement l'identifiant de l'apprenant, la date de l'événement, la page concernée ainsi que la description et le type.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des événements du MOOC (RD_Events)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque événement faire {     Si le type de l'événement contient « play_video » alors         Enregistrer les informations dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<playVideo learner="456723">

	<pre>time="2014-08-31T10:11:28.341819+00:00" page="https://www.france-universite-numerique- mooc.fr/courses/ENSCachan/20002/Trimestre_2_2014/courseware/15ded21253eb45158b54c8840646fd96/344feb3e16284e78bfe370a5f5283811/" description="{&amp;#34;id&amp;#34;:&amp;#34;i4x-ENSCachan-20002-video- 52bbacf54b764b6eb9b48e571422c1b9&amp;#34;,&amp;#34;code&amp;#34;:&amp;#34;html5&amp;#34;,&amp;#34;currentTime&amp;#34;:0}"&gt; play_video &lt;/playVideo&gt;</pre>
--	---

<b>ID_ProblemCheck</b>	
<b>Description</b>	Donnée intermédiaire permettant de filtrer la liste d'événements pour ne conserver que les événements « problem_check », et pour chaque événement l'identifiant de l'apprenant, la date de l'événement, la page concernée ainsi que la description et le type.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des événements du MOOC (RD_Events)</li> </ul>
<b>Algorithme</b>	<pre>Pour chaque événement faire {   Si le type de l'événement contient « problem_check » alors     Enregistrer les informations dans la donnée intermédiaire ; }</pre>
<b>Exemple</b>	<pre>&lt;quizz learner="745072" time="2014-08-31T16:53:36.539646+00:00" page="null" description="{&amp;#34;POST&amp;#34;: {&amp;#34;input_i4x-ENSCachan-20002-problem- 42cff20e49ae42ac9b228eab619d4d0e_2_1&amp;#34;: [&amp;#34;&amp;#34;]}, &amp;#34;GET&amp;#34;: {}}"&gt; /courses/ENSCachan/20002/Trimestre_2_2014/modx/i4x://ENSCachan/20002/problem/ 42cff20e49ae42ac9b228eab619d4d0e/problem_check &lt;/quizz&gt;</pre>

#### 4.3 Données additionnelles :

<b>AD_CourseStructure</b>	
<b>Description</b>	Donnée additionnelle permettant d'obtenir une représentation de la structure du cours sous la forme d'une liste de couples page (mère/page fille).
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>Le fichier CSV contenant la structure de cours</li> </ul>
<b>Algorithme</b>	Importation de chaque ligne du fichier CSV dans un format XML interne.
<b>Exemple</b>	<pre>&lt;Structure&gt;   &lt;ParentName&gt;Introduction au cours&lt;/ParentName&gt;   &lt;PageOpenDate&gt;2014-05-12T12:00:00Z&lt;/PageOpenDate&gt;   &lt;ParentHash&gt;15ded21253eb45158b54c8840646fd96&lt;/ParentHash&gt;   &lt;PageType&gt;chapter&lt;/PageType&gt;   &lt;ChildHash&gt;344feb3e16284e78bfe370a5f5283811&lt;/ChildHash&gt; &lt;/Structure&gt;</pre>

#### 4.4 Données brutes :

<b>RD_Events</b>	
<b>Description</b>	Donnée brute permettant d'importer les événements de la plateforme EDX/FUN en associant une sémantique qui sera exploitée par les autres données.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>Le fichier XML contenant les événements</li> </ul>
<b>Algorithme</b>	Récupération des informations suivantes pour chaque événement: Identifiant de l'apprenant (username), date/heure de l'événement (time), le serveur d'origine (host), la page concernée (page), la description de l'événement (event), le type d'événement (event_type), le navigateur utilisé (agent), l'adresse IP de l'apprenant (ip), la source de l'événement (event_source).
<b>Exemple</b>	<pre>&lt;Event&gt;   &lt;Time&gt;2014-04-22T04:49:36.858513+00:00&lt;/Time&gt;   &lt;User&gt;197094&lt;/User&gt;   &lt;Host&gt;www.france-universite-numerique-mooc.fr&lt;/Host&gt;   &lt;Page&gt;null&lt;/Page&gt;   &lt;Description&gt;{"POST": {}, "GET": {}}&lt;/Description&gt;   &lt;Type&gt;/courses/ENSCachan/20002/Trimestre_2_2014/about&lt;/Type&gt;   &lt;Agent&gt;Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)</pre>

```

Chrome/34.0.1847.116 Safari/537.36</Agent>
<IP>193.48.168.202</IP>
<Source>server</Source>
</Event>

```

### 5. Données modifiées ou ajoutées pour MOOCAZ V2

Suite au constat que tous les chapitres commençaient à la même date dans MOOCAZ V2 contrairement à MOOCAZ V1, la date n'était plus significative pour associer un événement à un chapitre, nous avons dû identifier la page directement et trouver le moyen d'associer une page à un chapitre à l'aide de la structure de cours. La structure des traces a un peu changée elle aussi car la clé « page » pouvait contenir « null » ou « x\_module » ce qui impliquait de prendre en compte la nouvelle clé « contextPath » qui contenait le code de la page au sein de sa chaîne de caractères.

#### 5.1 Indicateurs :

L'indicateur donnant la classification par chapitre a été revu afin de prendre en compte les devoirs qui étaient inexistantes dans MOOCAZ V1. Nous nous sommes basés directement sur un autre indicateur qui donne la catégorie de chaque apprenant pour chaque chapitre du MOOC (*I\_Detailed\_Classification*).

<b>I_LearnersClassification (V2)</b>	
<b>Description</b>	Indicateur permettant d'obtenir pour chaque chapitre le pourcentage et le nombre d'apprenants dans chaque catégorie.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des chapitres (ID_Chapters)</li> <li>La liste des catégories d'apprenants par semaine (I_DetailedClassification)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre (classés par ordre croissant de date de démarrage) faire {   nbViewer = compter le nombre de « viewer » pour le chapitre courant ;   nbQuizzer = compter le nombre de « quizzer » pour le chapitre courant ;   nbActiveViewer = compter le nombre d'« active viewer » pour le chapitre courant ;   nbInactive = compter le nombre d'« inactive » pour le chapitre courant ;   nbSolver = compter le nombre de « solver » pour le chapitre courant ;   nbCompleter = compter le nombre de « completer » pour le chapitre courant ;   nbLowCompleter = compter le nombre de « low completer » pour le chapitre courant ;   total = nbViewer + nbQuizzer + nbActiveViewer + nbInactive + nbSolver + nbCompleter + nbLowCompleter ;   pViewer = (nbViewer * 100) / total;   pQuizzer = (nbQuizzer * 100) / total;   pActiveViewer = (nbActiveViewer * 100) / total;   pInactive = (nbInactive * 100) / total;   pSolver = (nbSolver * 100) / total;   pCompleter = (nbCompleter * 100) / total;   pLowCompleter = (nbLowCompleter * 100) / total;   Enregistrer la liste des informations dans l'indicateur ; } </pre>
<b>Exemple</b>	<pre> &lt;chapter name="1. Cadrage du projet" viewer="0.0" cViewer="0" quizzer="9.543886" cQuizzer="498" activeViewer="0.0" cActiveViewer="0" inactive="90.456116" cInactive="4720" solver="0.0" cSolver="0" completer="0.0" cCompleter="0" lowCompleter="0.0" cLowCompleter="0"&gt;5218&lt;/chapter&gt; </pre>

Ce nouvel indicateur a été conçu pour permettre la détection des parcours de chaque apprenant, un parcours étant l'état (ou la catégorie) affecté à l'apprenant pour chacun des chapitres.

<b>I_DetailedClassification</b>	
<b>Description</b>	Indicateur permettant d'obtenir les triplets (apprenant, chapitre, état) afin d'afficher les parcours.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des chapitres (ID_Chapters)</li> <li>La liste des apprenants (ID_Learners)</li> <li>Le nombre de quizz par chapitre et par apprenant (ID_CountQuizz)</li> <li>Le nombre de devoirs par chapitre et par apprenant (ID_CountAssessment)</li> <li>Le nombre de lecture de vidéo par chapitre et par apprenant (ID_CountVideo)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre (classés par ordre croissant de nom) faire {   Pour chaque apprenant faire {     cptv = Récupérer le nombre de lectures de vidéo pour cet apprenant et ce chapitre ;     cptq = Récupérer le nombre de quizz pour cet apprenant et ce chapitre ;     cpta = Récupérer le nombre de devoirs pour cet apprenant et ce chapitre ;     Si ((cptv &gt; 0) et (cptq &gt; 0) et (cpta == 0)) alors class="activeViewer" ;     Si ((cptv &gt; 0) et (cptq &gt; 0) et (cpta &gt; 0)) alors class="completer" ; </pre>

	<pre> Si ((cptv == 0) et (cptq &gt; 0) et (cpta == 0)) alors class="quizzier" ; Si ((cptv &gt; 0) et (cptq == 0) et (cpta == 0)) alors class="viewer" ; Si ((cptv == 0) et (cptq == 0) et (cpta == 0)) alors class="inactive" ; Si ((cptv == 0) et (cptq == 0) et (cpta &gt; 0)) alors class="solver" ; Si ((cptv == 0) et (cptq &gt; 0) et (cpta &gt; 0)) alors class="lowCompleter" ; Si ((cptv &gt; 0) et (cptq == 0) et (cpta &gt; 0)) alors class="lowCompleter" ; Enregistrer les données dans l'indicateur ; } </pre>
<b>Exemple</b>	<state learner="229032" chapter="1. Cadrage du projet" class="inactive">229032</state>

## 5.2 Données intermédiaires :

Modification afin de prendre en compte la page « null » ou « x\_module ».

ID_ProblemCheck (V2)	
<b>Description</b>	Donnée intermédiaire permettant de filtrer la liste d'événements pour ne conserver que les événements « problem_check », et pour chaque événement l'identifiant de l'apprenant, la date de l'événement, la page concernée ainsi que la description et le type.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des événements du MOOC (RD_Events)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque événement faire {   Si le type de l'événement contient « problem_check » alors     Si (page == « null » ou « x_module ») alors remplacer la page par le chemin de     contexte.     Enregistrer les informations dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<quizz learner="880080" time="2014-11-05T15:22:26.340864+00:00" page="/courses/ENSCachan/20002S02/Trimestre_3_2014/xblock/i4x:;;_ENSCachan;_20002S02;_problem;_2fe31bb992854e18be463eece78bebb8/handler/xmodule_handler/problem_check" description="[Element: &lt;Description/&gt;]">problem_check</quizz>

Prise en compte du nouvel événement « create\_submission »

ID_CreateAssessment	
<b>Description</b>	Donnée intermédiaire permettant de filtrer la liste d'événements pour ne conserver que les événements « create_submission », et pour chaque événement l'identifiant de l'apprenant, la date de l'événement, la page concernée ainsi que la description et le type.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La liste des événements du MOOC (RD_Events)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque événement faire {   Si le type de l'événement contient « create_submission » alors     Si (page == « null » ou « x_module ») alors remplacer la page par le chemin de     contexte.     Enregistrer les informations dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<assessment learner="967704" time="2014-11-22T17:17:27.773743+00:00" page="/courses/ENSCachan/20002S02/Trimestre_3_2014/xblock/i4x:;;_ENSCachan;_20002S02;_openassessment;_08d800c3d7274072b2b653913583d064/handler/submit" description="[Element: &lt;Description/&gt;]">openassessmentblock.create_submission</assessment>

Prise en compte de la structure de cours pour retrouver le chapitre associé à chaque page, afin de compenser la disparition des semaines différenciées pour chaque chapitre.

ID_LinkPageChapter	
<b>Description</b>	Donnée intermédiaire permettant d'associer à chaque page le chapitre dont elle dépend.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>La structure de cours (AD_CourseStructure)</li> </ul>
<b>Algorithme</b>	Parcourir chaque chapitre de la structure de cours et de manière récursive descendre dans l'arborescence en associant à chaque fois le chapitre de départ avec la page courante.
<b>Exemple</b>	<linkStructure chapterHash="d7eca659c82f46f69ce0274f03abb990" chapterName="5. Déroulement du projet" page="c72ea6911ee148c6936cf9c51c5dd6a1"/>

<b>ID_CreateAssessmentChapter</b>	
<b>Description</b>	Donnée intermédiaire permettant de remplacer chaque page de l'événement devoir par le chapitre correspondant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des événements de création de devoir (ID_CreateAssessment)</li> <li>• La liste des associations page/chapitre (ID_LinkPageChapter)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque événement faire {   Rechercher le code de la page contenu dans la clé page, une fois la page identifiée,   retrouver le chapitre associé.   Enregistrer les informations dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<assessmentEvent chapter="Parcours Vidéo" learner="295221" time="2014-12-12T18:50:07.358948+00:00"/>

<b>ID_PlayVideoChapter</b>	
<b>Description</b>	Donnée intermédiaire permettant de remplacer chaque page de l'événement lecture vidéo par le chapitre correspondant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des événements de lecture de vidéo (ID_PlayVideo)</li> <li>• La liste des associations page/chapitre (ID_LinkPageChapter)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque événement faire {   Rechercher le code de la page contenu dans la clé page, une fois la page identifiée,   retrouver le chapitre associé.   Enregistrer les informations dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<playEvent chapter="Parcours Scénarisation" learner="566952" time="2014-11-09T00:44:26.884909+00:00"/>

<b>ID_ProblemCheckChapter</b>	
<b>Description</b>	Donnée intermédiaire permettant de remplacer chaque page de l'événement quizz par le chapitre correspondant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des événements de type quizz (ID_ProblemCheck)</li> <li>• La liste des associations page/chapitre (ID_LinkPageChapter)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque événement faire {   Rechercher le code de la page contenu dans la clé page, une fois la page identifiée,   retrouver le chapitre associé.   Enregistrer les informations dans la donnée intermédiaire ; } </pre>
<b>Exemple</b>	<quizzEvent chapter="5. Déroulement du projet" learner="723499" time="2014-11-05T23:42:17.408512+00:00"/>

Utilisation de la nouvelle donnée prenant en compte le chapitre.

<b>ID_CountVideo (V2)</b>	
<b>Description</b>	Donnée intermédiaire permettant de compter le nombre de fois où l'événement « play_video » intervient par chapitre et par apprenant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des chapitres (ID_Chapters)</li> <li>• La liste des apprenants (ID_Learners)</li> <li>• La liste des événements « play_video » (ID_PlayVideoChapter)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre faire {   Pour chaque apprenant faire {     cpt = compter le nombre d'événements « play_video » pour l'apprenant courant et pour     lesquels la page concernée comporte le code du chapitre courant ;     Enregistrer les informations dans la donnée intermédiaire ;   } } </pre>
<b>Exemple</b>	<countVideo learner="575646" chapter="Introduction au cours">1</countVideo>



ID_CountQuiz (V2)	
<b>Description</b>	Donnée intermédiaire permettant de compter le nombre de fois où l'événement « problem_check » intervient par chapitre et par apprenant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des chapitres (ID_Chapters)</li> <li>• La liste des apprenants (ID_Learners)</li> <li>• La liste des événements « problem_check » (ID_ProblemCheckChapter)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre faire {   Pour chaque apprenant faire {     cpt = compter le nombre d'événements « problem_check » pour l'apprenant courant et pour lesquels la page concernée comporte le code du chapitre courant ;     Enregistrer les informations dans la donnée intermédiaire ;   } } </pre>
<b>Exemple</b>	<countQuiz learner="373869" chapter="Introduction au cours">2</countQuiz>

Permet de compter le nombre de devoirs (nouvelle activité).

ID_CountAssessment	
<b>Description</b>	Donnée intermédiaire permettant de compter le nombre de fois où l'événement « create_submission » intervient par chapitre et par apprenant.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• La liste des chapitres (ID_Chapters)</li> <li>• La liste des apprenants (ID_Learners)</li> <li>• La liste des événements « create_submission » (ID_CreateAssessmentChapter)</li> </ul>
<b>Algorithme</b>	<pre> Pour chaque chapitre faire {   Pour chaque apprenant faire {     cpt = compter le nombre d'événements « create_submission » pour l'apprenant courant et pour lesquels la page concernée comporte le code du chapitre courant ;     Enregistrer les informations dans la donnée intermédiaire ;   } } </pre>
<b>Exemple</b>	<countAssessment learner="839614" chapter="Parcours Analyse">4</countAssessment>

### 5.3 Données brutes :

La donnée brute a été modifiée afin de prendre en compte le problème de la page « null » ou « x\_module ».

RD_Events (V2)	
<b>Description</b>	Donnée brute permettant d'importer les événements de la plateforme EDX/FUN en associant une sémantique qui sera exploitée par les autres données.
<b>Données nécessaires</b>	<ul style="list-style-type: none"> <li>• Le fichier XML contenant les événements</li> </ul>
<b>Algorithme</b>	Récupération des informations suivantes pour chaque événement: Identifiant de l'apprenant (username), date/heure de l'événement (time), le serveur d'origine (host), la page concernée (page), la description de l'événement (event), le type d'événement (event_type), le navigateur utilisé (agent), l'adresse IP de l'apprenant (ip), la source de l'événement (event_source), le chemin du contexte (context/path).
<b>Exemple</b>	<pre> &lt;Event&gt;   &lt;Time&gt;2014-12-16T20:17:48.444201+00:00&lt;/Time&gt;   &lt;User&gt;810103&lt;/User&gt;   &lt;Host&gt;www.france-universite-numerique-mooc.fr&lt;/Host&gt;   &lt;Page&gt;x_module&lt;/Page&gt;   &lt;Description/&gt;   &lt;Type&gt;problem_check&lt;/Type&gt;   &lt;Agent&gt;Mozilla/5.0 (Windows NT 6.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36&lt;/Agent&gt;   &lt;IP&gt;193.48.168.200&lt;/IP&gt;   &lt;Source&gt;server&lt;/Source&gt;   &lt;ContextPath&gt;/courses/ENSCachan/20002S02/Trimestre_3_2014/xblock/i4x::;_ENSCachan;_20 002S02;_problem;_99edc651249c4713a5dcc4e09a257fal/handler/xmodule_handler/problem_chec k&lt;/ContextPath&gt; &lt;/Event&gt; </pre>

## 6 Visualisation de l'indicateur calculé par UTL

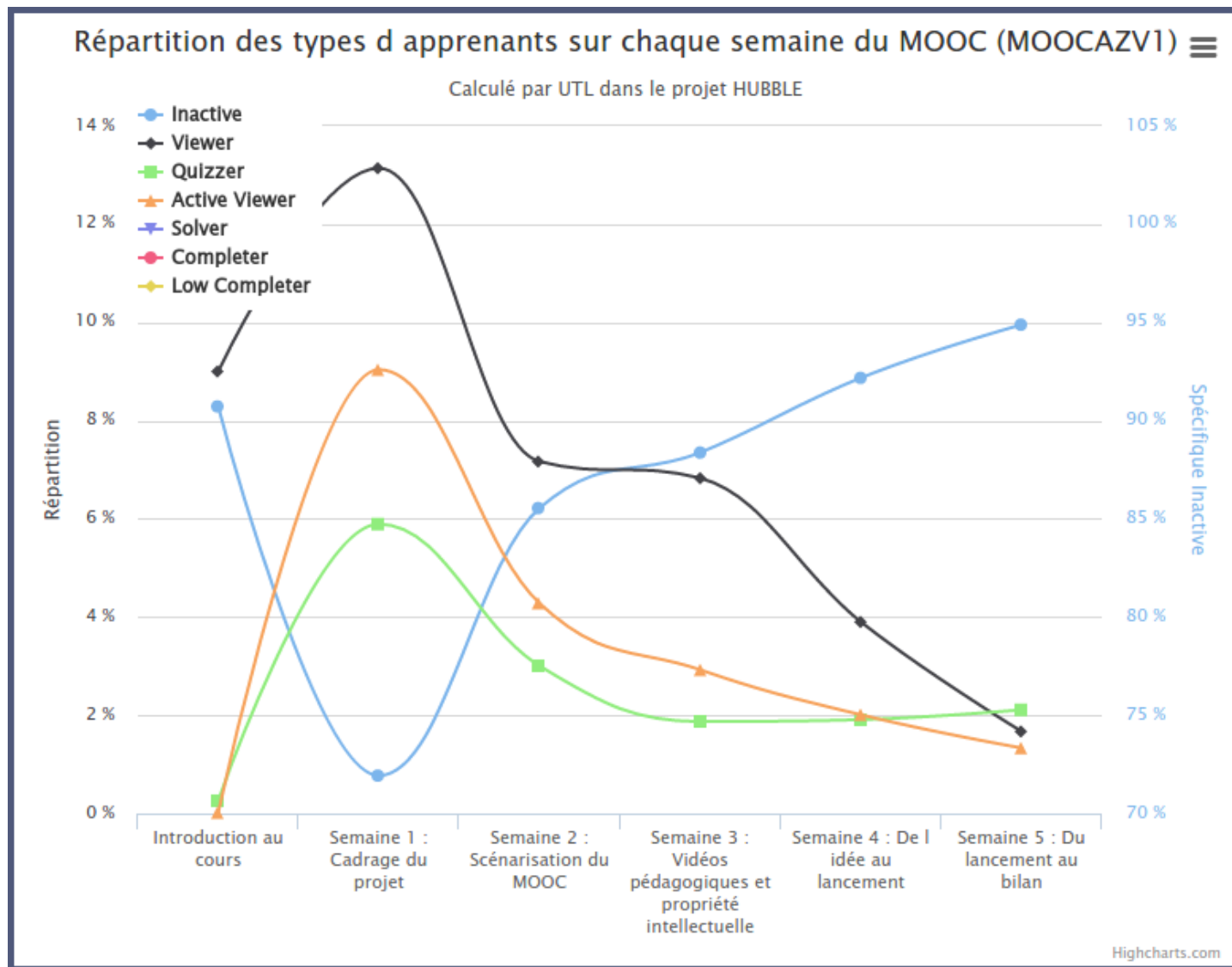
### 6.1 Technologies utilisées

Pour la visualisation, nous avons utilisé la librairie JavaScript HighCharts<sup>1</sup> ainsi que PHP et l'API de la base de données eXist permettant l'interaction entre PHP et la base de données. Nous avons choisi un type de graphique permettant d'afficher plusieurs courbes et 2 échelles de valeurs différentes car la proportion des apprenants inactifs est beaucoup plus importante que la proportion des autres apprenants.

### 6.2 Résultat

Le résultat est consultable sur <http://rech-iksal.iut-laval.univ-lemans.fr/UTL/MOOC/AZ/>.

Voici les 2 graphiques représentant le scénario 2 (types d'apprenants) sur chaque jeu de donnée MOOCAZ.



<sup>1</sup><http://www.highcharts.com>

