

Rapport de TER : Étude du concept de profils évolutifs dans le cadre du projet PERLEA

Blandine GINON

25-01-2010

Résumé

Ce document présente l'étude du concept de profils évolutifs dans le cadre du projet PERLEA, sa définition dans le langage de description de profils PMDLe et sa mise en œuvre dans l'environnement EPROFILEA. Il présente également différentes utilisations possibles des profils évolutifs, à un niveau théorique avec l'extension du modèle de contraintes sur profils cPMDL en cPMDLe et avec les opérateurs sur profils, et d'une manière pratique avec la mise en œuvre de certaines contraintes de cPMDLe dans le module Adapte de l'environnement EPROFILEA.

Mots-clés : Profil d'apprenant, langage, formalisme, évolution.

Table des Matières

1 INTRODUCTION	2
2 PRESENTATION DU CONTEXTE	2
2.1 LE PROJET PERLEA	2
2.2 L'ENVIRONNEMENT EPROFILEA	2
2.3 LE MODELE REPRO ET LE LANGAGE PMDL.....	2
3 TRAVAIL DE RECHERCHE	3
3.1 EXTENSION DU LANGAGE DE DESCRIPTION DE PROFILS PMDL : PMDLE.....	3
3.2 CRÉATION DE NOUVEAUX OPÉRATEURS SUR PROFILS.....	4
3.3 EXTENSION DU MODÈLE DE CONTRAINTES SUR PROFILS cPMDL : cPMDLE.....	5
4 MISE EN ŒUVRE DANS L'ENVIRONNEMENT EPROFILEA	6
4.1 MODIFICATION DU MODULE DE DESCRIPTION DE STRUCTURES DE PROFILS BÂTISSEUR	6
4.2 MODIFICATION DU MODULE DE SAISIE DE PROFILS PROSE.....	7
4.3 MODIFICATION DU MODULE D'INTÉGRATION DE DONNÉES EXTERNES TORNADE.....	7
4.4 MODIFICATION DU MODULE DE PERSONNALISATION D'ACTIVITÉS PÉDAGOGIQUES ADAPTE	7
5 CONCLUSION	8
6 ANNEXES	11
6.1 ANNEXE 1 : GRAMMAIRE DE PMDLE.....	12
6.2 ANNEXE 2 : REPRÉSENTATION GRAPHIQUE DE PMDLE.....	17
6.3 ANNEXE 3 : OPÉRATEURS SUR PROFILS ÉVOLUTIFS	20
6.4 ANNEXE 4 : EXTRAIT DE cPMDLE.....	24

1 Introduction

Dans le cadre du second semestre de master 1 d'informatique et en vue d'intégrer le master 2 recherche CODE, j'ai effectué un Travail d'Etude et de Recherche, d'une durée de cinq semaines. Le sujet de mon TER est « Étude du concept de profils évolutifs dans le cadre du projet PERLEA ». Pendant ce TER, j'ai été encadrée par Madame Jean-Daubias, maître de conférences au LIRIS, Laboratoire d'InfoRmatique en Image et Systèmes d'information, et membre de l'équipe SILEX, Supporting Interaction and Learning by Experience, du département Données, Connaissances, Services, qui travaille sur le projet PERLEA, dans le domaine des EIAH, Environnements Informatiques pour l'Apprentissage Humain.

L'objectif de ce TER est de faire évoluer la définition du concept de profils d'apprenants au sein du projet PERLEA [JEL10], afin que les profils puissent être évolutifs. Cette modification devra être faite d'un point de vue théorique, notamment sur la définition du langage de modélisation de profils PMDL [Eys08]. Le langage PMDL est spécifié selon la notation BNF (Backus Naur Form) et complété par une représentation graphique, il faudra faire évoluer ces deux aspects. Cette modification sera ensuite appliquée dans l'environnement EPROFILEA, ce qui entraînera la modification de la plupart de ses modules.

2 Présentation du contexte

2.1 Le projet PERLEA

Le projet PERLEA (Profils d'Elèves Réutilisés pour L'Enseignant et l'Apprenant) concerne la personnalisation de l'apprentissage par l'exploitation de profils d'apprenants.

Il vise à proposer des modèles pour la réutilisation et l'exploitation mutualisée des profils d'apprenants hétérogènes, existants ou à venir, papier-crayon ou logiciels, dans des contextes différents et par des acteurs autres que leur auteur [JEL09].

Un profil d'apprenant est un ensemble de données informatiques ; il est conçu à partir d'informations pédagogiques (informatisées ou non) concernant un apprenant. Dans le cadre du projet PERLEA, un profil d'apprenant est hybride, certaines de ces informations peuvent par exemple être déduites de résultats d'enquêtes nationales (tests de niveau en CE2, 6e, 3e) [Min08], ou d'évaluations faites par un enseignant, et d'autres peuvent être issues de logiciels externes, tels que J'ADE, [Min07] ou MoreMaths [BMR03]. Ces informations se présentent sous diverses formes : numériques (note /20, /10...), textuelles (maîtrisé, partiellement maîtrisé, non maîtrisé...) ou encore graphiques.

2.2 Le modèle REPro et le langage PMDL

Le projet PERLEA a abouti à la création du modèle REPro, Reuse of External Profiles, un modèle qui décrit les différentes étapes du processus de gestion de profils, de la création à l'exploitation de profils d'apprenants [Eys08]. Ce modèle met en évidence la nécessité d'harmoniser la structure et les données des profils pour en proposer une représentation unifiée. Pour cela il fait appel à un formalisme de description des profils permettant de les rendre réutilisables.

Le langage PMDL, Profiles MoDeling Language, est un langage de modélisation de profils qui permet de décrire la structure d'un profil et d'exprimer différents profils hétérogènes selon un même formalisme [JEL09]. Pour cela, PMDL distingue dans un profil d'apprenant sa partie structure, pouvant être commune à plusieurs profils, et sa partie données, propre à chaque apprenant. PMDL est l'apport principal de la thèse de Carole Eyssauthier-Bavay, soutenue en 2008 [Eys08].

2.3 L'environnement EPROFILEA

Le projet PERLEA a également donné lieu au développement d'un EIAH, l'environnement Eprofilea, dans lequel sont mis en œuvre les modèles définis dans PERLEA. Eprofilea s'adresse à des enseignants de la maternelle au supérieur, il est composé de plusieurs modules (cf. figure 1 et [Jea10]).

La première phase de l'environnement Eprofilea, celle de préparation, consiste à établir la structure des profils que l'on souhaite manipuler, avant d'y intégrer les données issues des profils externes pour constituer des profils d'apprenants conformes au souhait de l'enseignant et respectant le formalisme d'Eprofilea. Cette phase correspond aux étapes d'harmonisation et de transformation des profils du modèle REPro.

Le langage PMDL est opérationnalisé dans le module Bâtitteur qui permet à l'enseignant de définir la structure des profils d'apprenants selon un formalisme unifié, à partir de données extérieures à Eprofilea, profils issus d'un autre EIAH ou profils dits « papier-crayon » (issus par exemple d'une évaluation notée faite par l'enseignant). Les structures de profils issues de Bâtitteur seront complétées dans les modules d'intégration des données : Prose et Tornade.

Le module Prose, PROfils Saisis par l'Enseignant, aide l'enseignant à saisir les données de chacun de ses

élèves selon la structure de profils définie dans Bâtitseur dans le cas des profils papier-crayon.

Pour intégrer des profils externes (issus d'un autre EIAH), le module Tornade permet de constituer puis d'utiliser des systèmes de conversions de profils, les tourbillons.

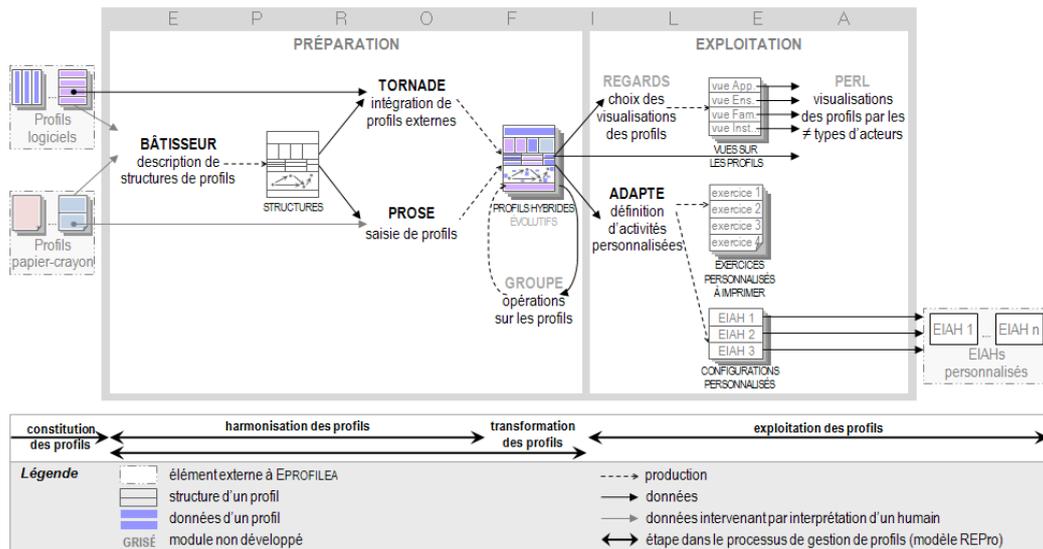


Figure 1 : Architecture d'EPROFILEA

La seconde phase de l'environnement, celle d'exploitation des profils, propose soit des activités sur les profils, avec les modules Perl et Regards, soit des activités pédagogiques définies en fonction du contenu des profils, avec le module Adapte [But06].

Le module Regards permettra de préparer la visualisation des profils. L'enseignant choisira quelles vues seront accessibles, par quel acteur de la situation pédagogique : enseignants, élèves, parents d'élèves...

Le module Perl permettra ensuite la visualisation interactive des profils par les différents acteurs selon les vues déterminées par l'enseignant dans Regards. Il propose également aux apprenants, en plus de la visualisation des profils, des activités autour des profils (reformulation, négociation des éléments du profil...) permettant à l'apprenant d'entrer dans une démarche réflexive par rapport à son apprentissage, et ainsi de mieux assimiler et exploiter les informations qui lui sont fournies [Eys08], [CAH03].

Le module Adapte, quant à lui, vise à donner à l'enseignant les moyens de personnaliser les activités pédagogiques proposées à ses apprenants selon leurs besoins en fonction de ses choix pédagogiques [Lef09]. Pour cela, il permet à l'enseignant de créer des structures d'exercices, ainsi que des stratégies pédagogiques qui affectent certaines structures d'exercices à un apprenant en fonction de son profil. Enfin, l'enseignant définit un contexte d'utilisation d'une stratégie pédagogique qui lui permettra ensuite de générer des feuilles d'exercices papier-crayon ou des séquences de travail personnalisées dans des EIAH pour chaque profil d'apprenant ou pour le profil de classe.

3 Travail de recherche

L'évolution du concept de profils d'apprenants hybrides vers le concept de profils d'apprenants hybrides et évolutifs a entraîné la modification des modèles de PERLEA. Si la modification principale concerne l'extension du langage PMDL [Eys08], cette étude a également permis la création de nouveaux opérateurs sur profils [Eys08], ainsi que l'extension du modèle de contraintes sur profils cPMDL [Lef09].

3.1 Extension du langage de description de profils PMDL : PMDLe

La première tâche de ce TER consistait à étudier et à redéfinir le concept de profils d'apprenants dans le but de rendre les profils évolutifs. Cette évolution a occasionné la modification et l'extension du langage PMDL vers le langage PMDLe, pour PMDL évolutif. Cette évolution porte aussi bien sur la notation en BNF du langage que sur sa représentation graphique.

Dans le cadre du projet PERLEA, un profil d'apprenant hybride et évolutif désigne un fichier rassemblant diverses informations sur l'apprenant. Ces informations peuvent concerner ses connaissances, compétences, conceptions, son comportement, ou encore des informations d'ordre métacognitif. Un tel profil peut comporter des informations provenant de différentes sources : on parle alors de profil hybride. Les informations qu'il contient peuvent également porter sur différentes périodes pour un même élément évalué : on parle alors de profil évolutif.

Il faut donc conserver dans le profil d'un apprenant chaque valeur et chaque commentaire, en les associant à une date et éventuellement à une source d'évaluation. C'est pourquoi certains éléments du langage PMDL ont été

modifiés. Par exemple, le contenu de l'élément *valeur_p* de PMDL dont la représentation est donnée en figure 2, a été modifié dans PMDLe (cf. figure 3). L'élément optionnel *commentaire* est remplacé par l'élément optionnel *commentaires* défini dans PMDLe. Dans PMDL, une *valeur_p* contenait un unique *commentaire*, qui est une chaîne de caractères, ainsi, chaque nouveau commentaire associé à *valeur_p* remplaçait le précédent, dont on ne gardait aucune trace. Au contraire, un *commentaires* de PMDLe contient zéro, un ou plusieurs *commentaire*. Chaque *commentaire* de PMDLe est associé à une *date*, un *texte* et éventuellement une *source*, ce qui permet d'associer un nouveau *commentaire* à *valeur_p* sans supprimer les *commentaire* précédents. L'élément *valeur_p* contient également zéro, une ou plusieurs *evaluation* définies dans PMDLe. Les *evaluation* contenues dans les éléments *valeur_p* sont associées à une *date*, une *valeur*, éventuellement une *source*, ce qui permet d'associer à *valeur_p* une nouvelle *valeur*, sans supprimer les précédentes.

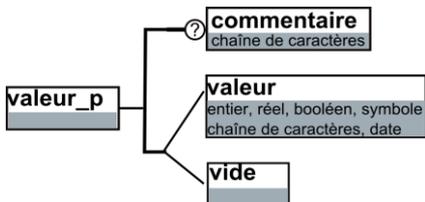


Figure 2 : Représentation graphique de *valeur_p* dans PMDL

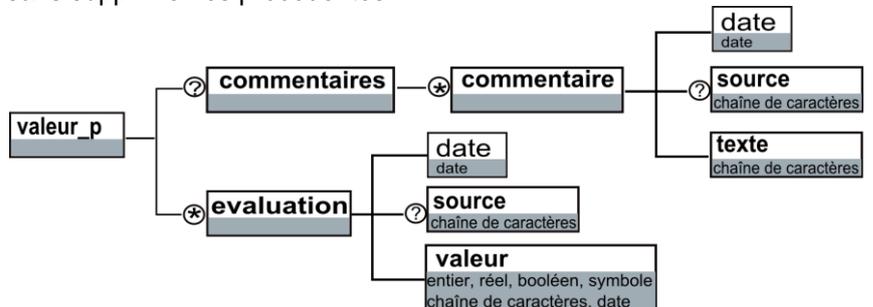


Figure 3 : Représentation graphique de *valeur_p* dans PMDLe

Le langage PMDL défini par Carole Eyssautier-Bavay dans sa thèse [Eys08], est décrit grâce à la notation BNF (Backus Naur Form). De plus, il existe une représentation graphique de PMDL qui permet de faciliter la lecture de ce langage. Les diagrammes de cette représentation graphique sont des structures d'arbres devant être lues de gauche à droite. Les éléments situés à gauche d'autres éléments et reliés à eux par un trait les contiennent, une légende détaillée est donnée en annexe 2.

En définissant PMDLe, j'ai cherché à rendre ce langage le plus complet et générique possible, en permettant par exemple d'associer à un profil autant d'informations sur l'apprenant que l'utilisateur souhaite et non plus seulement un nom et un prénom comme c'est le cas dans PMDL. J'ai également cherché à réduire les petites incohérences entre la définition de PMDL et son opérationnalisation dans le projet PERLEA. Pour cela, j'ai modifié d'autres éléments de PMDL. La notation BNF du langage PMDLe complet est donnée en annexe 1, les éléments écrits en gras sont ceux ayant été modifiés ou ajoutés par rapport à PMDL. La représentation graphique des principaux éléments de PMDLe est donnée en annexe 2.

3.2 Création de nouveaux opérateurs sur profils

L'expression des profils selon un formalisme commun à travers le langage PMDLe rend possible la définition d'opérateurs spécifiques de manipulation et de transformation de profils évolutifs. Trois familles d'opérateurs sur des profils PMDL ont déjà été définies : opérateurs de consultation, de structure de profils et de données de profils [Eys08]. Grâce à PMDLe, de nouveaux opérateurs peuvent être définis dans la famille des opérateurs de données de profils opérant sur les données individuelles des apprenants. Huit nouveaux opérateurs ont été définis. Les opérateurs sur profils respectant le formalisme PMDL sont également adaptés pour les profils évolutifs respectant le formalisme PMDLe. La figure 4 présente les nouveaux opérateurs dont les définitions semi-formelles sont données en annexe 3.

Opérateur	Description courte	Type d'élément concerné
CompM	Synthèse d'un élément par moyenne selon un intervalle	liste_composantes, graphe
CompS	Synthèse d'un élément par somme selon un intervalle	liste_repartition
FiltreE	Réduction d'un élément selon un intervalle	tous
FiltreP	Réduction d'un profil selon un intervalle	tous
ConcatE	Concaténation des commentaires d'un élément selon un intervalle	tous
ConcatP	Concaténation des commentaires d'un profil selon un intervalle	tous
InterEvalE	Intersection d'évaluations de deux profils selon un élément	tous
InterEvalP	Intersection d'évaluations de deux profils	tous

Figure 4 : PMDLe, les nouveaux opérateurs sur profils évolutifs

3.3 Extension du modèle de contraintes sur profils cPMDL : cPMDLe

Le modèle de contraintes sur profils cPMDL définit les différentes contraintes qui peuvent être associées à un profil d'apprenant défini selon le formalisme PMDL [Lef09]. Trois types de contraintes sont définis par cPMDL : les contraintes sur profils portant sur une valeur, celles portant sur un élément et celles portant sur le nombre d'occurrences. Seuls les deux premiers types de contraintes sont actuellement mis en œuvre dans l'environnement EPROFILEA.

Dans le but d'exploiter les nouvelles possibilités offertes par PMDL, j'ai défini trois nouveaux types de contraintes ajoutés à cPMDL pour étendre celui-ci en cPMDLe, pour cPMDL évolutif : contraintes d'évolution sur profils portant sur une valeur, sur un élément ou sur un nombre d'occurrences. Ces types de contraintes correspondent aux types existant dans cPMDL, lesquels étant toujours valables avec les profils évolutifs, mais portent spécifiquement sur le caractère évolutif des profils. Ainsi, avec une contrainte sur une valeur définie par cPMDL, on pourra sélectionner les profils d'apprenants pour lesquels cette valeur se situe dans un intervalle précis, par exemple une note entre 10 et 15. Avec une contrainte d'évolutivité sur une valeur définie par cPMDLe, on pourra sélectionner par exemple les profils d'apprenants dont cette valeur a augmenté d'au moins 3 points entre deux dates.

La définition formelle des contraintes d'évolution sur profils portant sur une valeur textuelle est donnée ci-après. Les autres contraintes, ainsi que des exemples sont donnés en annexe 4.

Contrainte d'évolution sur profils portant sur une valeur textuelle :

$$CPE_{V(T)} = \{Elt, E, \{D_1, b_1\}, \{D_2, b_2\}, \{\{i, b_i\}, \{s, b_s\}\}, 0 \leq i \leq s \leq \text{card}(\text{liste_valeurs}), D_1 < D_2\}$$

Définir une contrainte évolutive portant sur une valeur textuelle consiste à spécifier, pour un élément *Elt* du profil, un caractère *E* appartenant à $\{P, S, R\}$, où *P* signifie progression, *S* stabilité et *R* régression, ainsi que quatre couples $\{D_1, b_1\}$, $\{D_2, b_2\}$, $\{i, b_i\}$ et $\{s, b_s\}$, où D_1 et D_2 sont des dates, *i* et *s* sont des entiers et b_1, b_2, b_i et b_s sont des booléens. Evaluer cette contrainte pour un profil évolutif d'apprenant revient à comparer les valeurs relatives à certaines évaluations portant sur *Elt*.

Si les booléens b_1 et b_2 ont pour valeur faux, cela signifie que la contrainte ne portera que sur les deux évaluations ayant pour date D_1 et D_2 . On va alors comparer la valeur V_1 relative à l'évaluation portant sur *Elt* à la date D_1 et la valeur V_2 relative à l'évaluation portant sur *Elt* à la date D_2 .

Si *E* a pour valeur *P* ou *R*, il faut ensuite préciser un intervalle. Dans le cas d'une valeur textuelle, cet intervalle représentera l'écart de niveaux entre les valeurs relatives aux évaluations. L'intervalle sera donc inclus dans $[0, \text{card}(\text{liste_valeurs})]$, où $\text{card}(\text{liste_valeurs})$ est le nombre de niveaux de l'échelle de la valeur concernée. Par exemple, pour l'échelle (acquis, en cours d'acquisition, non acquis), un apprenant passant de non acquis à acquis a progressé de deux niveaux. Seules les échelles textuelles graduées sont concernées par ce type de contraintes, en effet on ne peut pas mesurer la progression sur une échelle non graduée telle que (bavard, actif, passif, attentif). Les booléens b_i et b_s indiquent si les bornes de l'intervalle sont incluses ou non. Si *E* a pour valeur *P*, pour Progression, on va vérifier si l'écart entre V_1 et V_2 se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si *E* a pour valeur *R*, pour Régression, on va vérifier si l'écart entre V_2 et V_1 se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si *E* a pour valeur *S*, pour Stabilité, on vérifiera si V_2 et V_1 sont identiques.

Si l'un des booléens b_1 ou b_2 a pour valeur vrai, cela signifie que la contrainte portera sur une période et non plus sur seulement deux dates. Si b_1 a pour valeur true, cela signifie que l'on va considérer toutes les évaluations dont la date est supérieure à D_1 et si b_2 a pour valeur true, cela signifie que l'on va considérer toutes les évaluations dont la date est inférieure à D_2 . Si D_1 et D_2 ont tous les deux pour valeur vrai, on ne considérera que les évaluations dont la date est comprise dans l'intervalle formé par D_1 et D_2 . Soient V_1, V_2, \dots, V_n les valeurs relatives aux évaluations dans la période considérée.

Si *E* a pour valeur *P*, pour Progression, on va vérifier si l'écart entre V_i et V_{i+1} pour tout *i* appartenant à l'intervalle $[0, n[$ se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si *E* a pour valeur *R*, pour Régression, on va vérifier si l'écart entre V_{i+1} et V_i pour tout *i* appartenant à l'intervalle $[0, n[$ se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si *E* a pour valeur *S*, pour Stabilité, on va vérifier que l'écart entre la plus petite et la plus grande des valeurs considérées se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$.

Pour que cPMDLe puisse être considéré comme complet, il faudra travailler avec des enseignants pour déterminer si cPMDLe leur permet d'exprimer tous leurs besoins, et dans le cas contraire, déterminer de quelle manière ce modèle de contraintes sur profils pourrait être enrichi. Cette proposition de cPMDLe partiellement mise en œuvre dans l'environnement EPROFILEA pourra donc faire l'objet de travaux ultérieurs.

4 Mise en œuvre dans l'environnement EPROFILEA

La mise en œuvre du concept de profils évolutifs dans le projet PERLEA, outre la modification des modèles concernés, a occasionné la modification des modules Bâtitseur, Prose, Tornade et Adapte de l'environnement EPROFILEA, cf. figure 1.

4.1 Modification du module de description de structures de profils Bâtitseur

Le module Bâtitseur permet à l'enseignant de décrire la partie *structure* d'un profil d'apprenant décrite dans le langage PMDL. Il permet ainsi de créer ou de modifier une structure de profils, qui pourra être commune à plusieurs apprenants. Dans l'environnement EPROFILEA, les structures de profils ainsi que les profils d'apprenants sont contenus dans des fichiers xml, le doctype de ces deux types de fichiers a donc été modifié pour correspondre à la définition du concept de profils d'apprenants décrit par le langage PMDL. Le module Bâtitseur a été modifié pour que les fichiers de structures de profils qui en sont issus soient conformes au nouveau doctype, cf. figures 4 et 5.

D'autres modifications ont également été apportées au module Bâtitseur, correspondant aux modifications apportées par PMDL. Par exemple, les briques de type Commentaire ont été rebaptisées Texte et contiennent des valeurs de types `texte_libre` auxquelles on peut maintenant associer des commentaires, comme pour les autres types de briques. De plus une structure de profils peut maintenant contenir plusieurs briques de type Texte, dont l'enseignant peut choisir le nom. Avant ce TER, il ne pouvait y avoir qu'une seule brique de ce type et elle s'appelait obligatoirement Commentaires.

```
<brique id="8" type="0" nom="Mathématiques" indice="2" commentaire= « très bien ① »>
  <arbre_des_composantes niveaux="2" ponderation="faux">
    <composante nom="Multiplication" commentaire= « bien, Alain est plus appliqué ② »>>
      <valeur>16 ③</valeur>
    </composante>
  </arbre_des_composantes>
</brique>
```

Figure 4 : Extrait d'un fichier de profil d'apprenant non évolutif dans EPROFILEA

```
<brique id="8" type="0" nom="Mathématiques" indice="2">
  <commentaires ④>
    <commentaire date="2009/11/23" source="contrôle de semestre">correct ⑤</ commentaire >
    <commentaire date="2010/01/12" source="interrogation">très bien ⑥</ commentaire >
  </commentaires>
  <arbre_des_composantes niveaux="2" ponderation="faux">
    <composante nom="Multiplication">
      <valeur id_echelle="0" num= "0">
        <evaluation date="2009/11/23" source="contrôle de semestre">11 ⑦</evaluation>
        <evaluation date="2010/01/12" source="interrogation">16 ⑧</evaluation>
      </valeur>
      <commentaires>
        <commentaire date="2009/11/23" source="contrôle de semestre">Des progrès à
faire ⑨</ commentaire >
        <commentaire date="2010/01/12" source="interrogation">bien, Alain est plus
appliqué ⑩</ commentaire >
      </commentaires>
    </composante>
  </arbre_des_composantes>
</brique>
```

Figure 5 : Extrait d'un fichier de profil d'apprenant évolutif dans EPROFILEA

En comparant les extraits de fichiers de profils d'apprenants, figures 4 et 5, on constate que la structure du premier profil ne peut contenir que des informations relatives à une unique évaluation pour la composante « Multiplication » et qu'en cas de nouvelle évaluation, ces informations seront perdues et remplacées par de nouvelles. Dans l'exemple de la figure 4, la valeur de cette évaluation est 16, cf. ③. Au contraire, dans le second profil qui est évolutif, les informations relatives à chaque évaluation sont conservées, modifiables et réutilisables.

La figure 5 montre en ⑦ que l'élève a obtenu la note de 11 pour l'évaluation du 23/11/2010 et en ⑧ qu'il a obtenu la note de 16 à l'évaluation du 12/01/2010. De même on voit que le second profil conserve chaque commentaire, au contraire du premier. L'attribut « commentaire » de la brique mathématiques, ①, a été remplacé dans le second profil par un nœud « commentaires », ④, pouvant lui-même contenir plusieurs nœuds « commentaire ». On voit par exemple que le 12/01/2010, l'élève a eu comme commentaire relatif à la brique « Mathématiques » ⑥, « très bien », alors que le 23/11/2009, ce commentaire était « correct », ⑤.

4.2 Modification du module de saisie de profils Prose

Le module Prose permet à l'enseignant de créer et de compléter des profils d'apprenants correspondant à une structure de profils issue du module Bâtisseur. Ce module permet ainsi de compléter la partie *donnee* d'un profil décrite dans le module Bâtisseur selon une opérationnalisation du langage PMDL et spécifique à chaque apprenant. Dans l'environnement EPROFILEA, un profil d'apprenant est contenu dans un fichier xml et respecte le même doctype que les fichiers de structure de profils. Celui-ci ayant évolué pour satisfaire le concept de profils évolutifs, le module Prose a dû être modifié en conséquence.

L'interface de Prose a également été changée pour permettre à l'enseignant de choisir entre ajouter une nouvelle évaluation aux profils sélectionnés et en modifier les évaluations précédentes.

Certains défauts de Prose ont également été corrigés pour permettre une utilisation plus pratique. Par exemple, l'enseignant peut désormais choisir entre sélectionner les profils de toute une classe et ne sélectionner que certains profils, appartenant ou non à la même classe. Seuls les profils d'élèves correspondant à la structure de profils choisie par l'enseignant lui sont proposés, afin de lui éviter des hésitations dans ses choix. Les fichiers contenant un profil d'apprenant ne sont créés que si l'enseignant saisit une information dans ce profil, ce qui évite de créer de nombreux fichiers « vides ». De plus, la sauvegarde du fichier contenant un profil d'apprenant n'est effectuée que lorsque l'enseignant change de brique, passe à un autre élève ou lorsqu'il quitte Prose et non plus à chaque petite modification comme c'était le cas au début de ce TER.

4.3 Modification du module d'intégration de données externes Tornade

Le module Tornade permet de compléter la partie *donnee* d'un profil décrite dans le module Bâtisseur selon une opérationnalisation du langage PMDL à partir de données issues de logiciels externes. De même que le module Prose, le module Tornade a été modifié afin de prendre en compte le concept de profils évolutifs. Pour cela, il a dans un premier temps fallu modifier la structure des tourbillons créés par Tornade. Un tourbillon est un système de conversion de profils permettant d'intégrer des données d'un profil externe, issu d'un autre EIAH, dans un profil respectant une structure de profil issue de Bâtisseur [JEL10]. La structure d'un tourbillon est similaire à la structure de profils à laquelle il se rapporte à la différence près que les informations que contiennent les nœuds évaluations et commentaires ne sont pas du texte ou des entiers mais des symboles permettant de repérer comment remplir correctement ces nœuds à partir d'un fichier élèves issu d'un logiciel externe. La nouvelle structure des tourbillons, basée sur celles des profils évolutifs, permet de conserver toutes les informations saisies par l'enseignant. Cela permet que celui-ci puisse facilement faire marche arrière s'il a modifié un tourbillon et souhaite revenir à une ancienne version de ce tourbillon sans avoir à saisir à nouveau les mêmes informations, ce qui peut s'avérer long et complexe.

Par ailleurs, au début de ce TER, l'utilisation de Tornade sur un profil hybride devait se faire avant l'utilisation de Prose. En effet, si l'utilisation de Prose ne modifie pas les informations issues de logiciels externes, lors de l'utilisation du module Tornade, chaque information contenue dans un profil d'apprenant était supprimée. Ensuite, l'utilisateur pouvait compléter le profil à partir de nouvelles informations issues d'un logiciel externe. Ce fonctionnement ne permettait pas l'évolutivité d'un profil et remettait également en cause son caractère hybride. Dorénavant, l'utilisation de Prose et Tornade peut se faire indifféremment dans un ordre ou dans l'autre sans perte d'information.

4.4 Modification du module de personnalisation d'activités pédagogiques Adapte

Le module Adapte permet la création de feuilles d'exercices ou de séquences d'activités personnalisées dans différents logiciels pédagogiques en fonction du profil de chaque apprenant et selon la stratégie pédagogique, ainsi que du contexte d'utilisation de cette stratégie définis par l'enseignant. Pour cela, l'étape de génération de feuilles d'exercices ou de séquences d'activités personnalisées nécessite la définition de règles d'affectation d'activités pédagogiques aux apprenants qui s'appuient sur des contraintes sur les profils, définies à l'aide de cPMDL, cf. 3.3 et [Lef09].

Le langage PMDL ayant été étendu en PMDLe, il a fallu dans un premier temps modifier les fonctions de consultation de profils qui trouvent dans le profil d'un apprenant la valeur associée à un élément du profil pour qu'elles déterminent la valeur associée à l'évaluation la plus récente concernant cet élément.

De plus, pour illustrer l'intérêt du concept de profils évolutifs et la manière dont ces derniers peuvent être utilisés par un enseignant pour personnaliser l'apprentissage de ses élèves, j'ai mis en œuvre dans Adapte trois nouvelles contraintes de cPMDL : les contraintes d'évolution sur profils portant sur une valeur textuelle, sur une valeur numérique ou sur un entier positif. Pour cela, j'ai ajouté un bouton « Nouvelle contrainte d'évolution... » à l'écran de définition d'une stratégie pédagogique, cf. figure 8 A. Il permet de créer une ou plusieurs contraintes d'évolution portant sur une valeur du profil dont l'échelle est numérique ou textuelle.

On peut voir sur la figure 8 C et D que l'enseignant est en train de créer deux nouvelles contraintes, C2 et C3, portant sur l'élément Analyse de la brique Mathématiques, dont l'échelle est numérique (note entre 0 et 20). La contrainte C2, cf. figure 8D, concernera les apprenants dont la note aura augmenté d'au moins 5 points et d'au plus 10 points entre les deux plus récentes évaluations en Analyse. La contrainte C3, cf. figure 8C, concernera les apprenants dont la note aura diminué de 0 à 8 points entre les deux plus récentes évaluations en Analyse. Avant cela, l'enseignant avait défini C1, figure 8 B, une contrainte d'évolution sur la composante Conjugaison de la brique Français, dont l'échelle est textuelle (maîtrisé, partiellement maîtrisé, non maîtrisé). La contrainte C1 concernera les apprenants ayant progressé d'au moins un niveau en Conjugaison entre les deux plus récentes évaluations dans cette discipline. Par exemple, un élève passé du niveau non maîtrisé au niveau partiellement maîtrisé ou au niveau maîtrisé entre les deux dernières évaluations sera concerné par cette contrainte, mais pas un élève resté au niveau partiellement maîtrisé. Si un profil ne contient pas deux évaluations portant sur l'élément choisi dans la période spécifiée, la contrainte ne sera pas prise en compte. C'est-à-dire que si cette contrainte est utilisée dans une règle, alors on associera à l'élève concerné les exercices qu'on associerait à un élève dont le profil vérifie les contraintes de la règle et le cas échéant ceux que l'on associerait à un élève dont le profil ne vérifie pas les contraintes de la règle. Ce fonctionnement est similaire à celui qui existait précédemment dans le module Adapte avec cPMDL.

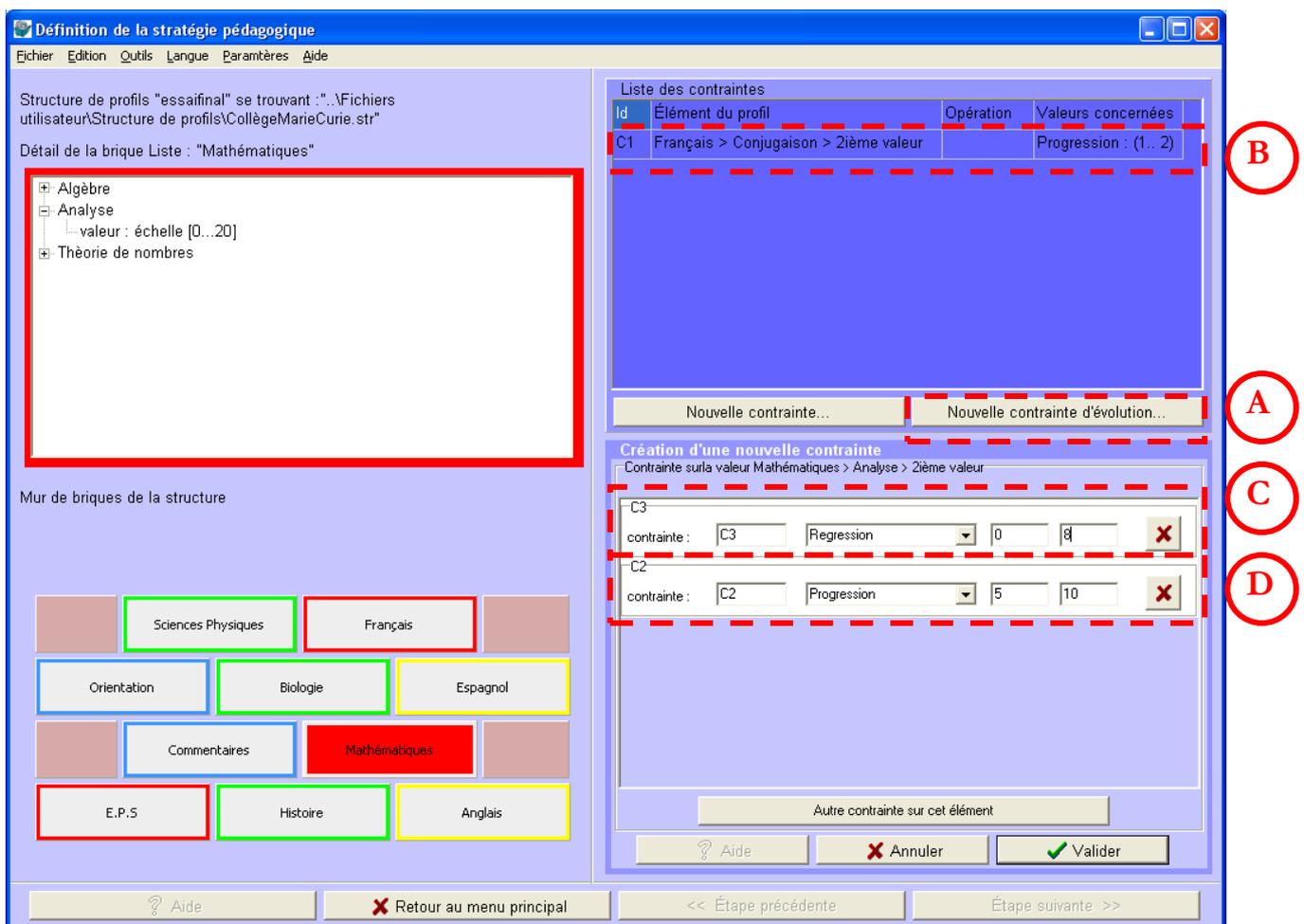


Figure 8 : Nouvel écran de définition d'une stratégie pédagogique

5 Conclusion

Durant mon TER, j'ai effectué un travail de recherche dans lequel j'ai défini et mis en œuvre dans l'environnement EPROFILEA le concept de profils d'apprenants hybrides et évolutifs. Cela m'a amenée à faire Rapport de TER de Blandine GINON

évoluer le langage de description de profils PMDL en PMDLe. Dans le but d'exploiter l'évolutivité des profils, j'ai également étendu le modèle de contraintes sur profils cPMDL en cPMDLe, puis je l'ai partiellement mis en œuvre dans le module Adapte de l'environnement EPROFILEA. Enfin, j'ai défini de manière semi formelle huit opérateurs sur profils, dont la mise en œuvre reste à faire.

Dans un premier temps, j'ai fait évoluer la notation en BNF ainsi que la représentation graphique du langage de modélisation de profils PMDL afin d'intégrer le concept de profils évolutifs d'apprenants. Ce travail a abouti à l'extension de PMDL en PMDLe. J'ai également ajouté une représentation xml, plus pratique pour la mise en œuvre du langage. On peut voir sur la figure 9 la synthèse de l'évaluation de PMDL [Eys08], avec en rouge les améliorations apportées par PMDLe.

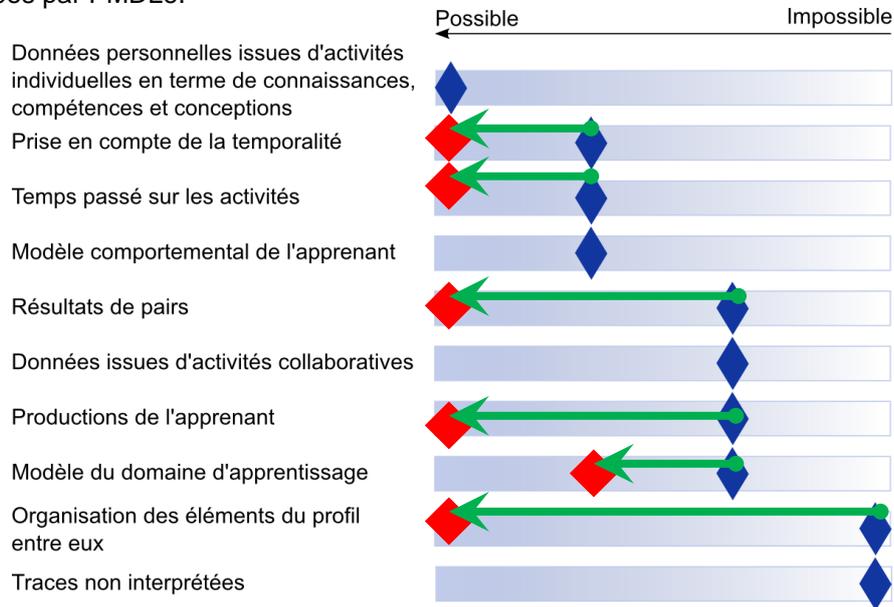


Figure 9 : évaluation de PMDL

J'ai ensuite mis en œuvre le formalisme de PMDLe dans l'environnement EPROFILEA, en modifiant les modules Bâtitseur, Prose, Tornade et Adapte. On peut voir en rouge sur l'architecture de l'environnement EPROFILEA, figure 10, les modifications apportées à cet environnement durant mon TER. La modification la plus importante tient au fait qu'Eprofilea, qui gérait jusqu'à présent uniquement des profils hybrides, gère désormais des profils hybrides et évolutifs, ce dans tous les modules existants.

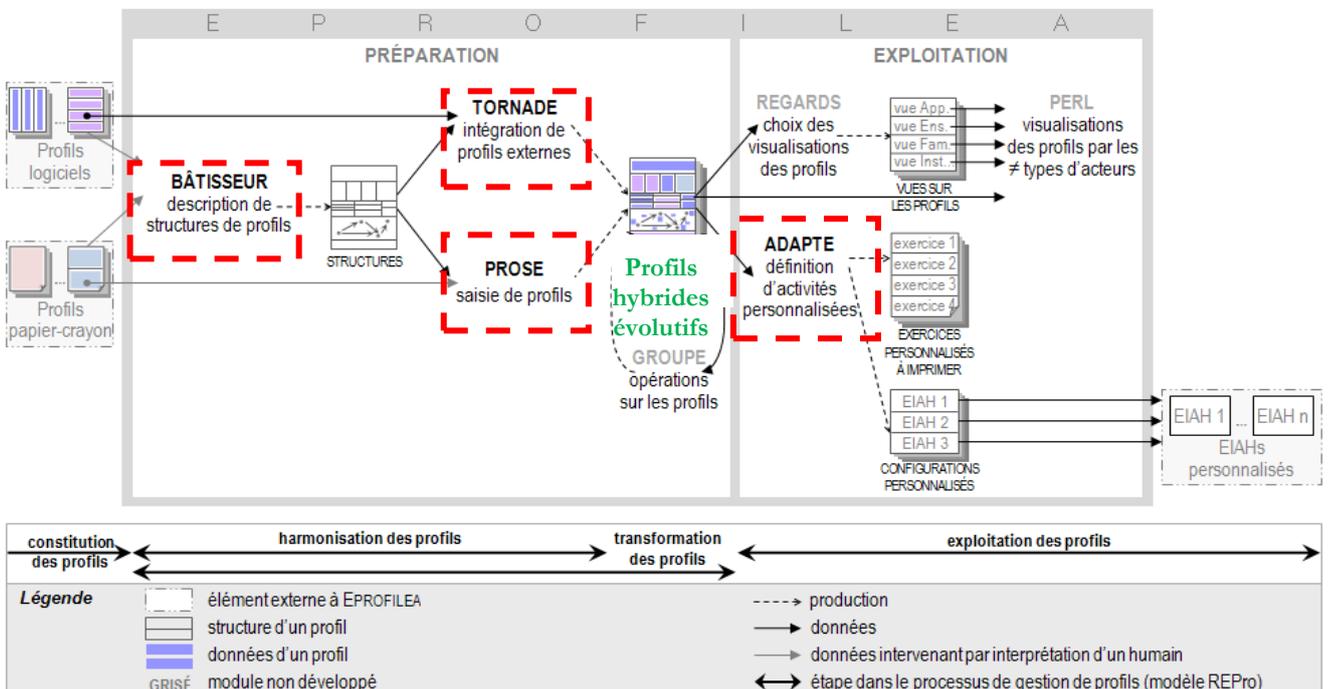


Figure 10: Architecture d'EPROFILEA

Par ailleurs, pour mettre en évidence les diverses utilisations possibles du formalisme décrit par PMDLe, j'ai défini huit nouveaux opérateurs sur profils respectant ce formalisme. Ceux-ci ne sont pas encore mis en œuvre dans l'environnement EPROFILEA mais font actuellement l'objet d'un stage de master 2 recherche, encadré par Madame Jean-Daubias.

J'ai également étendu le modèle de contraintes sur profils cPMDL en cPMDLe, qui pourra faire l'objet de travaux ultérieurs. Ainsi, j'ai défini cinq nouveaux types de contraintes permettant à l'utilisateur de contraindre l'évolutivité des profils d'apprenants. J'ai ensuite mis en œuvre trois nouvelles contraintes d'évolutivité dans le module Adapte de l'environnement EPROFILEA.

Cette première expérience de la recherche m'a particulièrement intéressée et m'a confortée dans mon projet d'études. En effet, je souhaite m'inscrire en master 2 recherche dans la spécialité CODE, dans le but de préparer par la suite une thèse dans le domaine de l'intelligence artificielle et des EIAH.

Références

- [BMR03] S. Bull, A. McEvoy, E. REID, « Learner Models to Promote Reflexion in Combined Desktop PC / Mobile Intelligent Learning Environments », AIED'03, p.199-208, 2003.
- [But06] M. Buthod, « ADAPTE - Générateurs d'exercices personnalisés dans le cadre du projet PERLEA », Mémoire de Master Recherche Connaissances et Raisonnement, Université Claude Bernard Lyon1, juin 2006.
- [Eys08] C. Eyssautier-Bavay, « Modèles, langage et outils pour la réutilisation de profils d'apprenants », thèse de doctorat, Université Joseph Fourier Grenoble1, 2008.
- [Jea03] S. Jean-Daubias, « Exploitation de profils d'apprenants » Proceedings of EIAH'03, Strasbourg, 2003.
- [Jea10] S. Jean-Daubias, <http://liris.cnrs.fr/stephanie.jean-daubias/projets/p-perlea.html>, (dernière visite : janvier 2010).
- [JEL09] S. Jean-Daubias, C. Eyssautier-Bavay, M. Lefevre « Uniformisation de la structure de profils d'apprenants issus de sources hétérogènes », EIAH 2009 - Environnements Informatique pour l'Apprentissage Humain, Le Mans, France. pp. 77-84. ISBN 978-2-7342-1149-5, 2009.
- [JEL10] S. Jean-Daubias, C. Eyssautier-Bavay, M. Lefevre « Modèles et outils pour rendre possible la réutilisation informatique de profils d'apprenants hétérogènes », STICEF, article à paraître en 2010.
- [Lef09] M. Lefevre, « Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils », thèse de doctorat, Université Claude Bernard Lyon1, 2009.
- [Min07] Ministère Français de l'éducation nationale, de l'enseignement supérieur et de la recherche « J'ADE : Guide d'installation et d'utilisation », 2007.
- [Min08] Ministère Français de l'éducation nationale, « Evaluations CE2 et 6^{ième} », <http://evace26.education.gouv.fr/>, (dernière visite : janvier 2010).

6 Annexes

Table des annexes

6 ANNEXES.....	11
6.1 ANNEXE 1 : GRAMMAIRE DE PMDLE.....	12
6.2 ANNEXE 2 : EXTRAIT DE LA REPRÉSENTATION GRAPHIQUE DE PMDLE	17
6.3 ANNEXE 3 : OPÉRATEURS SUR PROFILS ÉVOLUTIFS	20
6.4 ANNEXE 4 : EXTRAIT DE CPMDLE.....	24

6.1 Annexe 1 : Grammaire de PMDL

Dans la notation BNF, on distingue les méta-symboles, appartenant à la notation elle-même, les terminaux, appartenant au langage décrit et les non terminaux. Ainsi :

- ::= est un méta-symbole signifiant « est défini par »
- <> est un méta-symbole signifiant pour <exemple1> que exemple1 est un non-terminal
- « » est un méta-symbole signifiant pour <exemple2> que exemple2 est un terminal
- Une suite d'expressions signifie le « et » logique
- n lignes pour un même non-terminal ou des expressions à la suite séparées par | signifie le « ou » logique

```
<profil> ::= "profil" <nom> <identifiant> <datecreation> <datemodification> <structure> <donnees>
::= "profil" <nom> <identifiant> <datecreation> <datemodification> <structure> <donnees>
<commentaires>
```

```
<nom> ::= <chaîne de caractères>
```

```
<identifiant> ::= <identificateur>
```

```
<datecreation> ::= <date>
```

```
<datemodification > ::= <date>
```

```
<commentaires> ::= <vide>
::= <commentaire> <commentaires>
```

```
<commentaire> ::= <date> <source> <texte_libre>
::= <date> <texte_libre>
```

```
<texte_libre> ::= <chaîne de caractères>
```

```
<source> ::= <chaîne de caractères>
```

```
//-----partie structure du profil-----
```

```
<structure> ::= "informations_eleve" <informations_eleve> <elements>
```

```
<informations_eleve> ::= <element> <informations_eleve>
::= <element>
```

```
<intitule> ::= <chaîne de caractères>
```

```
<type> ::= "entier"
::= "date"
::= "chaîne de caractères"
::= "réel"
::= "booléen"
::= "symbole"
::= "identificateur"
```

```
<enum> ::= <liste énumérée>
```

```
<elements> ::= <element> <elements>
::= <element>
```

```
<element> ::= "element" <nom> <identifiant> <contenu>
::= "element" <nom> <identifiant> <commentaires> <contenu>
```

```

<contenu> ::= "texte"
           ::= "liste_composantes" <liste_composantes>
           ::= "liste_repartition" <liste_repartition>
           ::= "graphe" <graphe>

<liste_composantes> ::= "ponderrees" "echelles" <echelles> <liste_comp_ponderrees>
                    ::= "non_ponderrees" "echelles" <echelles> <liste_comp_non_ponderrees>

<liste_comp_ponderrees> ::= <composante_ponderree> <liste_comp_ponderrees>
                        ::= <composante_ponderree>

<composante_ponderree> ::= <niveau> "feuille" <composante> <poids>
                        ::= <niveau> <composante> <poids> <sous_composantes_ponderrees>
                        ::= <niveau> "feuille" <composante> <poids> <commentaires>
                        ::= <niveau> <composante> <poids> <commentaires> <sous_composantes_ponderrees>

<sous_composantes_ponderrees> ::= <composante_ponderree> <sous_composantes_ponderrees>
                                ::= <composante_ponderree>

<niveau> ::= <entier>

<poids> ::= <r el>
        ::= <entier>

<composante> ::= <identifiant> <intitule>

<liste_comp_non_ponderrees> ::= <composante_non_ponderree> <liste_comp_non_ponderrees>
                             ::= <composante_non_ponderree>

<composante_non_ponderree> ::= <niveau> "feuille" <composante>
                             ::= <niveau> <composante> <souscomposantes_non_ponderrees>
                             ::= <niveau> "feuille" <composante> <commentaires>
                             ::= <niveau> <composante> <commentaires> <souscomposantes_non_ponderrees>

<sous_composantes_non_ponderrees>
        ::= <composante_non_ponderree> <souscomposantes_non_ponderrees>
        ::= <composante_non_ponderree>

<echelles> ::= <echelle> <echelles>
            ::= <echelle>
            ::= <echelle> <etiquette> <echelles>
            ::= <echelle> <etiquette>

<echelle> ::= <nom> <identifiant> <ordonne> <liste_valeurs>
            ::= <nom> <identifiant> <ordonne> <liste_valeurs> <unite>
            ::= <nom> <identifiant> <intervalle_valeurs> <pas>
            ::= <nom> <identifiant> <intervalle_valeurs> <pas> <unite>

<ordonne> ::= <bool en>

<liste_valeurs> ::= <valeur> <liste_valeurs>
                ::= <valeur>

<intervalle_valeurs> ::= <valeur_num> <valeur_num>
                    ::= <date> <date>

<valeur_num> ::= <r el>
              ::= <entier>

<pas> ::= <valeur_num>

```

```

<valeur> ::= <valeur_num>
          ::= <chaîne de caractères>
          ::= <date>
          ::= <booléen>
          ::= <symbole>
          ::= <identificateur>

<unite> ::= <chaîne de caractères>

<texte> ::= <texte_libre>
         ::= <texte_libre> <commentaires>

<liste_repartition> ::= "nbre_valeurs" <nbre_valeurs> <liste_comp_non_ponderees>
                    ::= "nbre_valeurs" <nbre_valeurs> <etiquettes> <liste_comp_non_ponderees>

<etiquette> ::= "etiquette" <chaîne de caractères>

<etiquettes> ::= <numero_valeur> <etiquette> <etiquettes>
              ::= <numero_valeur> <etiquette>

<nbre_valeurs> ::= <entier>

<numero_valeur> ::= <entier>

<graphe> ::= <ens_composantes> <ens_liens>

<ens_composantes> ::= <composantes>
                   ::= "echelles_composantes" <echelles> <composantes>

<composantes> ::= <composante> <composantes>
               ::= <composante>
               ::= <composante> <commentaires> <composantes>
               ::= <composante> <commentaires>

<ens_liens> ::= <liens>
             ::= "echelles_liens" <echelles> <liens>

<liens> ::= <lien> <liens>
         ::= <lien>

<lien> ::= <identifiant> <id_composante> <id_composante>
        ::= <identifiant> <id_composante> <id_composante> <commentaires>
        ::= <identifiant> <id_composante> <id_composante> "type_lien" <type_lien>
        ::= <identifiant> <id_composante> <id_composante> "type_lien" <type_lien> <commentaires>

<type_lien> ::= <chaîne de caractères>

<id_composante> ::= <identifiant>

//-----partie données du profil-----

<donnees> ::= "informations_eleve_p" <informations_eleve_p><elements_p>

<informations_eleve_p> ::= <element_p> <informations_eleve_p>
                       ::= <element_p>

<valeur_info> ::= <valeur>
               ::= <identificateur>

<elements_p> ::= <element_p> <elements_p>
              ::= <element_p>

```

```

<element_p> ::= "element_p" <identifiant> <contenu_p>
            ::= "element_p" <identifiant> <commentaires> <contenu_p>

<contenu_p> ::= "liste_composantes_p" <liste_composantes_p>
            ::= "liste_repartition_p" <liste_repartition_p>
            ::= "graphe_p" <graphe_p>
            ::= "texte_p" <texte_p>

<liste_composantes_p> ::= <bloc_composantes_valeurs_p>

<bloc_composantes_valeurs_p> ::= <composante_valeurs_p> <bloc_composantes_valeurs_p>
                               ::= <composante_valeurs_p>

<vide> ::= "vide"
<composante_valeurs_p> ::= <identifiant> <valeurs_p>
                          ::= <identifiant>
                          ::= <identifiant> <valeurs_p> <commentaires>
                          ::= <identifiant> <commentaires>

<valeur_p> ::= <ens_valeurs>
            ::= <commentaires> <ens_valeurs>
            ::= <commentaires> <ens_valeurs> <echelle>
            ::= <ens_valeurs> <echelle>

<ens_valeurs> ::= <vide>
              ::= <evaluation> <ens_valeurs>

<evaluation> ::= <date> <source> <valeur>
               ::= <date> <valeur>
               ::= <date> <source> <texte>
               ::= <date> <texte>

<liste_repartition_p> ::= "nbre_questions" <nbre_questions> <repartition_comp_p>
                       ::= <repartition_sous_comp_p>

<nbre_questions> ::= <entier>
                  ::= <entier> <commentaires>

<repartition_comp_p> ::= <comp_valeur_p> <repartition_comp_p>
                      ::= <comp_valeur_p>

<comp_valeur_p> ::= <identifiant> <valeurs_rep>
                  ::= <identifiant> <valeurs_rep> <commentaires>

<valeurs_rep> ::= <valeur_entiere> <valeurs_rep>
               ::= <valeur_entiere>

<valeur_entiere> ::= <entier>
                  ::= <vide>
                  ::= <vide> <commentaires>
                  ::= <entier> <commentaires>

<repartition_sous_comp_p> ::= <id_composante> "nbre_questions" <nbre_questions>
                           <repartition_comp_p> <repartition_sous_comp_p>
                           ::= <id_composante> "nbre_questions" <nbre_questions> <repartition_comp_p>

<id_composante> ::= <identificateur>

<graphe_p> ::= "composantes" <bloc_composantes_valeurs_p> "liens" <liens_p>
            ::= "composantes" <bloc_composantes_valeurs_p>
            ::= "liens" <liens_p>

```

<liens_p> ::= <lien_valeurs_p> <liens_p>
::= <lien_valeurs_p>

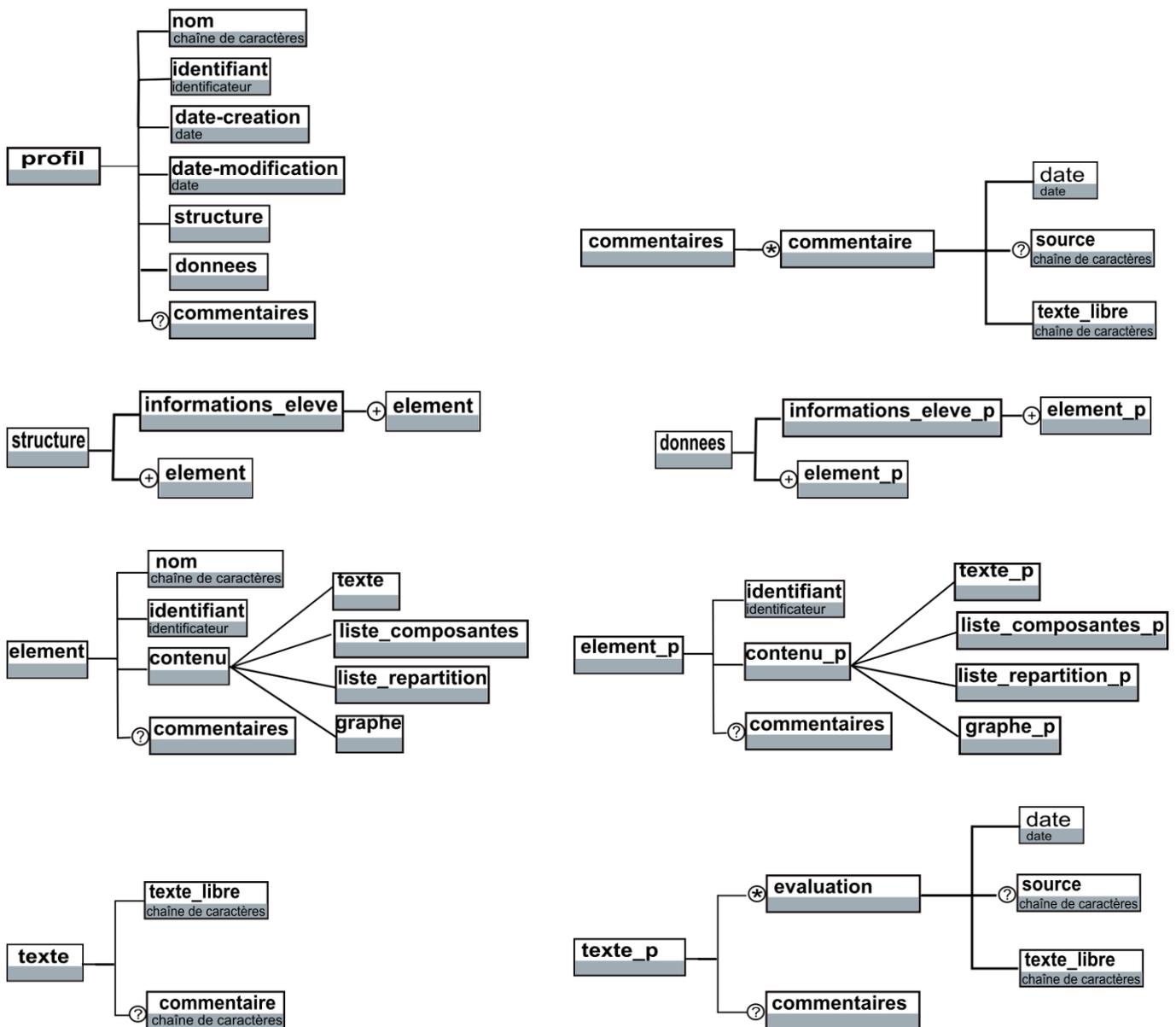
<lien_valeurs_p> ::= <identifiant>
::= <identifiant> <commentaires>
::= <identifiant> <valeurs_p>
::= <identifiant> <valeurs_p> <commentaires>

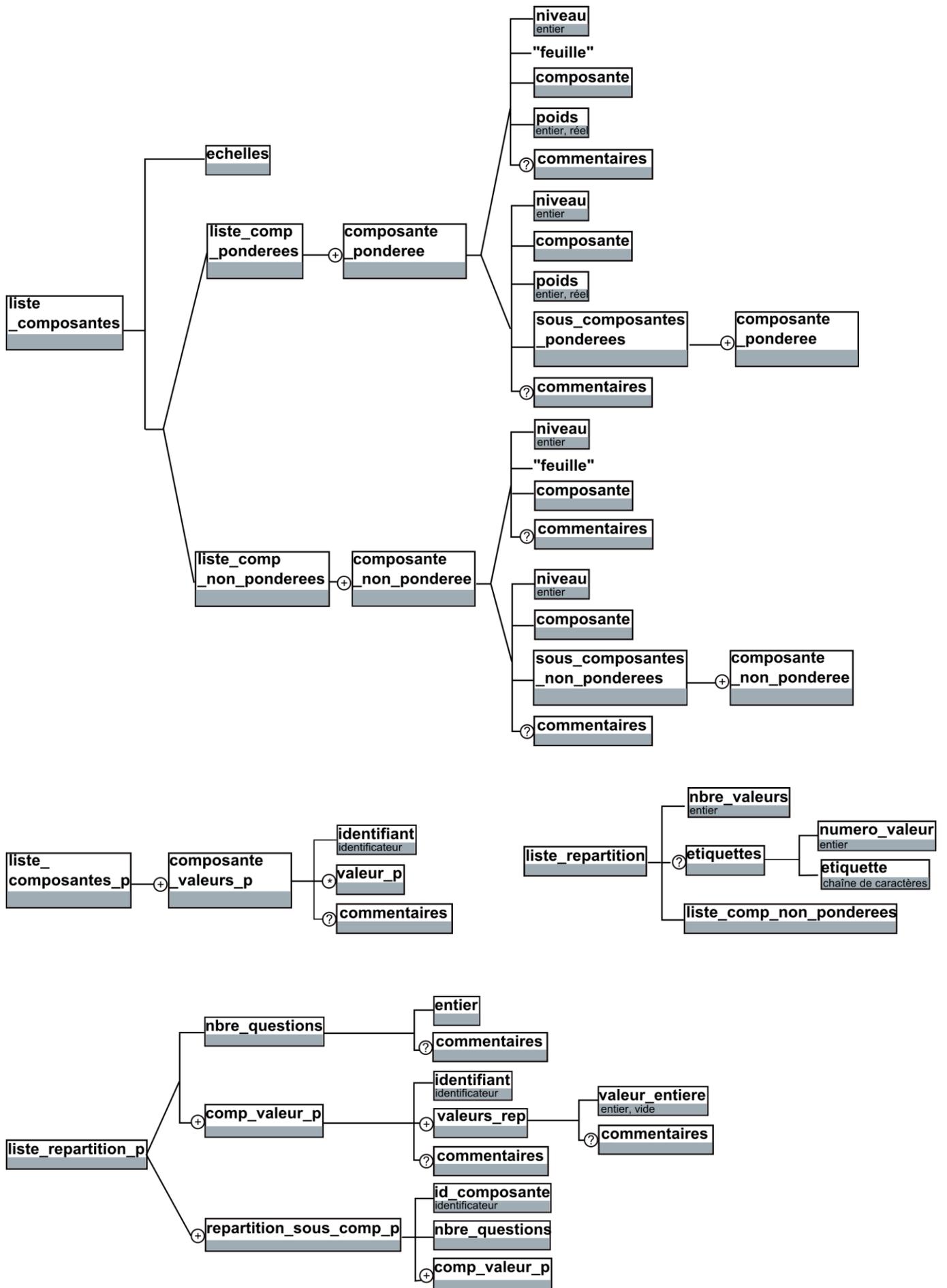
<texte_p> ::= <vide>
::= <evaluation> < texte_p >

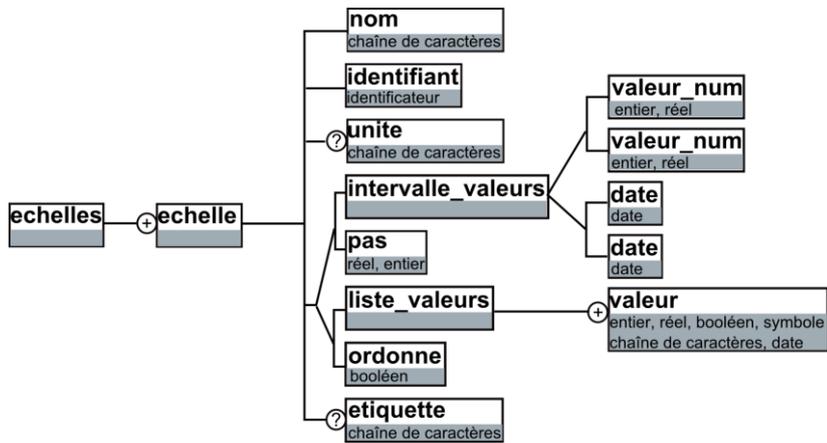
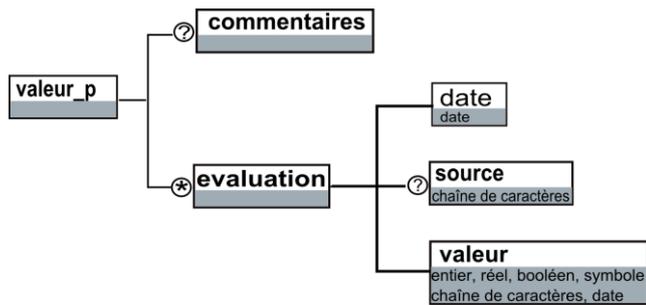
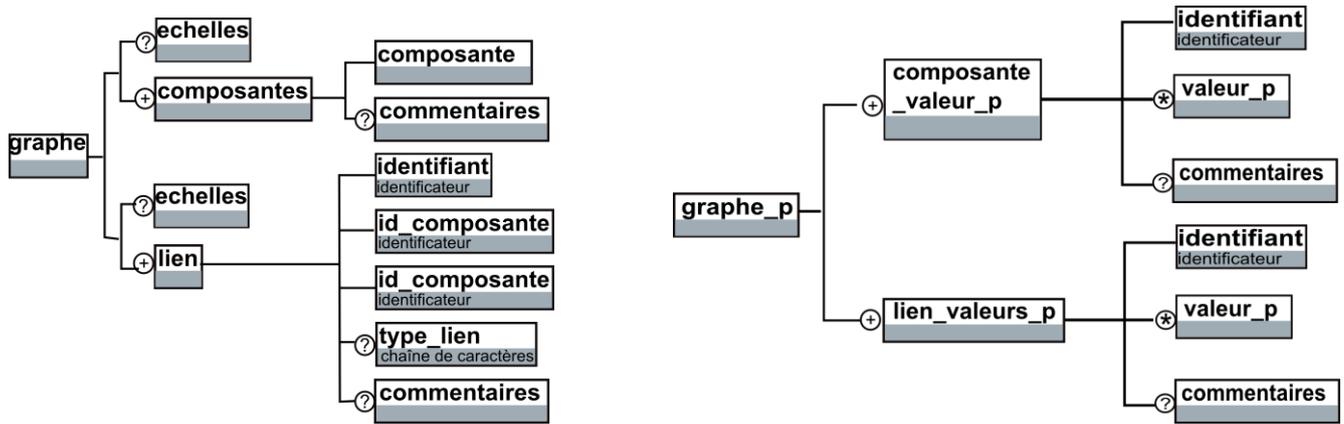
6.2 Annexe 2 : Représentation graphique de PMDL

Les diagrammes de cette représentation graphique sont des structures d'arbres devant être lues de gauche à droite. Les éléments situés à gauche d'autres éléments et reliés à eux par un trait les contiennent. Ainsi :

- < signifie une relation de « ou » logique
- [signifie une relation de « et » logique
- * signifie que l'élément peut être présent 0 ou n fois
- + signifie que l'élément peut être présent 1 ou n fois
- ? signifie que l'élément peut être présent 0 ou 1 fois, c'est-à-dire qu'il est optionnel
- aucun symbole devant un élément signifie qu'il est présent exactement une fois
- chaque élément est représenté dans un cadre possédant une partie blanche et une partie grise. La partie blanche porte le nom de l'élément. Si l'élément est un terminal, s'il n'est composé d'aucun autre élément, alors la partie grise contient le type de l'élément. Dans le cas contraire, elle est vide.







6.3 Annexe 3 : Opérateurs sur profils évolutifs

Le tableau ci-après présente les opérateurs sur profils. Les opérateurs en gras sont ceux que j'ai définis pour exploiter l'évolutivité des profils.

	Opérateur	Description	Type d'élément concerné	Opérande 1	Opérande 2	Opérande 3	Opérande 4	Résultat
Opérateurs de consultation	<i>Consult</i>	Consultation de la liste de valeurs attribuées à un objet (composante, lien ou info_eleve)	tous sauf <i>texte</i>	profil P	objet O			liste de valeurs l
	<i>ConsultText</i>	Consultation du contenu d'un élément texte	<i>texte</i>	profil P	<i>élément</i> texte			chaîne de caractère
Opérateurs de structure	<i>FiltreP</i>	Filtre du profil par <i>élément</i>	tous sauf <i>informations_eleve</i>	structure Sp	n <i>élément</i>			structure Sp1
	<i>FiltreE</i>	Filtre d'un <i>élément</i> par <i>composante</i>	<i>liste_composantes, liste_repartition, graphe</i>	structure Sp	n <i>composante</i>			structure Sp1
	<i>FiltreN</i>	Filtre du profil par <i>niveau</i>	<i>liste_composantes, liste_repartition</i>	structure Sp	<i>niveau</i> n			structure Sp1
	union « U »	Union de structures	tous	structure Sp1	structure Sp2			structure Sp3
	Intersection « ∩ »	Intersection de structures	tous	structure Sp1	structure Sp2			structure Sp3
	Concatéation « ° »	Concatéation d' <i>élément</i>	tous	<i>élément</i> E1 de Sp	<i>élément</i> E2 de Sp			structure Sp1
	ajout « + »	Ajout d' <i>élément</i> à une <i>structure</i>	tous	structure Sp	<i>élément</i> E d'une autre structure			structure Sp1
	suppression « - »	Suppression d' <i>élément</i> d'une structure	tous	structure Sp	<i>élément</i> E de Sp			structure Sp1
Opérateurs de données	<i>ValM</i>	Synthèse d'une <i>composante</i> par moyenne	<i>liste_composantes</i>	profil P	<i>composante</i> c			profil P1
	<i>ValS</i>	Synthèse d'une <i>composante</i> par comptage	<i>liste_repartition</i>	profil P	<i>composante</i> c			profil P1
	différence « - »	Différence entre deux profils	tous sauf <i>informations_eleve</i>	profil P1	profil P2			profil P3
	<i>InterE</i>	Filtrage selon un intervalle sur un <i>élément</i>	<i>liste_composantes, graphe</i>	profil P	<i>élément</i> E	<i>échelle</i> e	couple de valeur (x,y)	profil P1
	<i>InterP</i>	Filtrage selon un intervalle sur le profil	<i>liste_composantes, graphe</i>	profil P	<i>échelle</i> e	couple de valeur (x,y)		profil P1
	<i>Group</i>	Constitution de profil de groupe	tous	n profils				profil P
	<i>CompM</i>	Synthèse d'un élément par moyenne selon un intervalle	<i>liste_composantes, graphe</i>	profil P	<i>élément</i> E	couple de date (d1, d2)		profil P1
	<i>CompS</i>	Synthèse d'un élément par somme selon un intervalle	<i>liste_repartition</i>	profil P	<i>élément</i> E	couple de date (d1, d2)		profil P1
	<i>FiltreE</i>	Réduction d'un élément selon un intervalle	tous	profil P	<i>élément</i> E	couple de date (d1, d2)		profil P1
	<i>FiltreP</i>	Réduction d'un profil selon un intervalle	tous	profil P	couple de date (d1, d2)			profil P1
	<i>ConcatE</i>	Concatéation des commentaires d'un élément selon un intervalle	tous	profil P	<i>élément</i> E	couple de date (d1, d2)		profil P1
	<i>ConcatP</i>	Concatéation des commentaires d'un profil selon un intervalle	tous	profil P	couple de date (d1, d2)			profil P1
	<i>InterevalE</i>	Intersection d'évaluations de deux profils selon un élément	tous	profil P1	profil P2	<i>élément</i> E	couple de date (d1, d2)	profil P2 et profil P3
	<i>InterevalP</i>	Intersection d'évaluations de deux profils	tous	profil P1	profil P2	couple de date (d1, d2)		profil P2 et profil P3

- Opérateur de compression par moyenne selon un intervalle de dates sur un élément

Cet opérateur de profils évolutifs permet de compresser un élément de type *liste_composante* ou *graphe* en remplaçant toutes les évaluations de ses *composantes* et *sous_composantes* dont la date est comprise dans un certain intervalle par une unique évaluation qui a pour valeur la moyenne arithmétique des valeurs des évaluations remplacées. Cet opérateur prend en entrée le profil, l'élément concerné et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait compresser toutes les évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie un profil ayant la même structure que le profil initial mais où l'élément spécifié a été compressé. Cet élément ne contient plus d'évaluation dont la date est comprise dans l'intervalle considéré, mais contient pour chaque *composante* dans laquelle des évaluations ont été supprimées une seule évaluation qui en est la moyenne. Les échelles des évaluations concernées doivent être numériques ou textuelles ordonnées converties en numériques.

Cet opérateur est utile notamment pour ne pas conserver chaque petit résultat d'un trimestre ou d'une année scolaire passée, mais conserver des résultats plus généraux. Voici une définition semi-formelle de l'opérateur de

compression par moyenne selon un intervalle de dates sur un élément, respectant la notation utilisée pour la définition initiale des opérateurs [Eys08] :

On définit les notations suivantes : soient Sp une structure de profil, \mathcal{E} l'ensemble des éléments de Sp . $\forall E \in \mathcal{E}$, soit \mathcal{C} l'ensemble des composantes de C et $Type(E)$ le type de E .

Soient P un profil, $E \in \mathcal{E}$ tel que $Type(E) = (liste_composantes \vee graphe)$. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$, $C \in \mathcal{C}$.

On a : $P_1 \leftarrow CompM(P, E, \{d_1, d_2\})$, P_1 étant un profil tel que $P_1 = (P \setminus E) \cup E_1$, avec $E_1 = E \setminus \{x_1, \dots, x_n\} \cup y$ et tel que $\forall i, x_i$ est une évaluation de E dont la date est comprise dans l'intervalle $[d_1, d_2]$, et y est une évaluation dont la valeur est la moyenne arithmétique des valeurs des x_i .

- Opérateur de compression par somme selon un intervalle de dates sur un élément

Cet opérateur sur profils évolutifs permet de compresser un élément de type `liste_repartition` en remplaçant toutes les évaluations de ses composantes et sous_composantes dont la date est comprise dans un certain intervalle par une unique évaluation qui a pour valeur la somme des valeurs des évaluations remplacées. Il prend en entrée le profil, l'élément concerné et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait compresser toutes les évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie un profil ayant la même structure que le profil initial, mais où l'élément spécifié a été réduit. Cet élément ne contient plus d'évaluation dont la date est comprise dans l'intervalle considéré, mais contient pour chaque composante dans laquelle des évaluations ont été supprimées, une seule évaluation qui en est la somme. Les échelles des évaluations concernées doivent être numériques ou textuelles ordonnées converties en numériques.

Voici une définition semi-formelle de l'opérateur de compression par somme selon un intervalle de dates sur un élément :

Soit P un profil, $E \in \mathcal{E}$ tel que $Type(E) = (liste_repartition)$. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$, $C \in \mathcal{C}$.

On a : $P_1 \leftarrow CompS(P, E, \{d_1, d_2\})$, P_1 étant un profil tel que $P_1 = (P \setminus E) \cup E_1$, avec $E_1 = E \setminus \{x_1, \dots, x_n\} \cup y$ et tel que $\forall i, x_i$ est une évaluation de E dont la date est comprise dans l'intervalle $[d_1, d_2]$, et y est une évaluation dont valeur est la somme arithmétique des valeurs des x_i .

- Opérateur de filtrage selon un intervalle de dates sur un élément

Cet opérateur sur profils évolutifs permet de filtrer un élément afin de ne garder que les données relatives à une évaluation dont la date est comprise dans un certain intervalle. Il prend en entrée le profil, l'élément concerné et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait supprimer toutes les données relatives à des évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie un profil ayant la même structure que le profil initial, mais où l'élément spécifié a été filtré. Cet élément ne contient plus aucune évaluation dont la date d'évaluation est comprise dans l'intervalle considéré.

Voici une définition semi-formelle de l'opérateur de filtrage selon un intervalle de dates sur un élément :

Soient P un profil, $E_1 \in \mathcal{E}$ tel que $Type(E_1) = (liste_composantes \vee graphe \vee texte \vee liste_repartition)$. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$. On a : $P_1 \leftarrow FiltreE(P, E_1, \{d_1, d_2\})$, P_1 étant un profil tel que $\mathcal{E}_{P_1} = (\mathcal{E}_P \setminus E_1) \cup E_2$, avec $Type(E_1) = Type(E_2)$ et tel que E_2 ne contienne aucune évaluation dont la date est comprise dans l'intervalle $[d_1, d_2]$.

- Opérateur de filtrage selon un intervalle de dates sur un profil

Cet opérateur sur profils évolutifs permet de filtrer un profil afin de ne garder que les données relatives à une évaluation dont la date est comprise dans un certain intervalle. Il prend en entrée le profil et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait supprimer toutes les données relatives à des évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie un profil ayant la même structure que le profil initial, mais où celui-ci a été filtré. Le profil ne

contient plus aucune valeur, commentaire ou lien dont la date d'évaluation est comprise dans l'intervalle considéré.
Voici une définition semi-formelle de l'opérateur de filtrage selon un intervalle de dates sur un profil :

Soit P un profil. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$. On a : $P_1 \leftarrow \text{FiltreP}(P, \{d_1, d_2\})$, P_1 étant un profil tel que $\mathcal{L}_{P_1} = (\mathcal{L}_P \setminus E_1^i) \cup E_2^i$, $\forall E_1^i \in \mathcal{L}_P$ avec $Type(E_1) = Type(E_2)$ et tel que E_2^i ne contienne que les valeurs, commentaires ou liens de E_1^i dont la date n'est pas comprise dans l'intervalle $[d_1, d_2]$.

- Opérateur de concaténation des commentaires selon un intervalle de dates sur un élément

Cet opérateur sur profils évolutifs permet de réaliser un regroupement entre les commentaires d'une composante d'un profil d'apprenant, dans le but de compresser un profil sans perdre d'informations, pour une période donnée. L'opérateur prend en entrée un profil, un élément de ce profil et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait faire l'intersection entre des évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie un profil dont la structure est la même que celle du profil initial, mais dont tous les commentaires de la composante donnée dans l'intervalle de dates ont été remplacés par un seul commentaire contenant la concaténation de la date, de la source et du contenu des commentaires supprimés.

Voici une définition semi-formelle de l'opérateur de concaténation des commentaires selon un intervalle de dates sur un élément :

Soient P un profil, $E \in \mathcal{L}$. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$.
On a : $P_1 \leftarrow \text{ConcatE}(P, E, \{d_1, d_2\})$, P_1 étant un profil tel que $P_1 = (P \setminus E) \cup E_1$, avec $E_1 = E \setminus \{x_i \text{ commentaire}\} \cup X$, où $X = \text{somme}(\text{date}(x_i) + ' : + source(x_i) + ' : + x_i)$ avec $\text{date}(x_i) \in [d_1, d_2]$.

- Opérateur de concaténation des commentaires selon un intervalle de dates sur un profil

Cet opérateur sur profils évolutifs permet de réaliser un regroupement entre les commentaires d'un profil d'apprenant, dans le but de compresser un profil sans perdre d'informations, pour une période donnée. L'opérateur prend en entrée un profil et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait faire l'intersection entre des évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie un profil dont la structure est la même que celle du profil initial, mais dont tous les commentaires de chaque composante dans l'intervalle de dates ont été remplacés par un seul commentaire contenant la concaténation de la date, la source et le contenu des commentaires supprimés.

Voici une définition semi-formelle de l'opérateur de concaténation des commentaires selon un intervalle de dates sur un élément :

Soit P un profil. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$.
On a : $P_1 \leftarrow \text{ConcatP}(P, \{d_1, d_2\})$, P_1 étant un profil tel que $P_1 = (P \setminus C) \cup C_1$, $\forall C \in \mathcal{C}$ avec $C_1 = C \setminus \{x_i \text{ commentaire}\} \cup X$, où $X = \text{somme}(\text{date}(x_i) + ' : + source(x_i) + ' : + x_i)$ avec $\text{date}(x_i) \in [d_1, d_2]$.

- Opérateur d'intersection d'évaluations sur un élément

Cet opérateur sur profils évolutifs permet de réaliser une intersection entre deux profils d'apprenants, dans le but de ne conserver que les évaluations communes aux deux profils, pour une période donnée. Cela peut par exemple permettre à un enseignant de s'assurer que les élèves d'une classe ont le même nombre d'évaluations dans une composante. L'opérateur prend en entrée les deux profils, la composante de chaque profil que l'on va considérer et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait faire l'intersection entre des évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie les deux profils dont la structure sera restée la même mais dont toutes les évaluations de chaque composante dans l'intervalle de date sont communes aux deux profils.

Voici une définition semi-formelle de l'opérateur d'intersection d'évaluations sur un élément :

Soient P_1 et P_2 deux profils. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$ et $E \in \mathcal{F}$.
 On a : $P_3 \leftarrow \text{InterEvalE}(P_1, P_2, E, \{d_1, d_2\})$ et $P_4 \leftarrow \text{InterEvalE}(P_2, P_1, E, \{d_1, d_2\})$, P_3 et P_4 étant des profils tels que $P_3 = (P_1 \setminus E_1) \cup E_3$, $P_4 = (P_2 \setminus E_2) \cup E_3$, avec $\text{identifiant}(E_1) = \text{identifiant}(E_2) = \text{identifiant}(E)$ et \forall évaluation $x_i \in E_1 : x_i \in E_3 \Leftrightarrow x_i \in E_2$.

- Opérateur d'intersection d'évaluations sur un profil

Cet opérateur de profil évolutif permet de réaliser une intersection entre deux profils d'apprenants, dans le but de ne conserver que les évaluations communes aux deux apprenants, pour une période donnée. Cela peut par exemple permettre à un enseignant de s'assurer que les élèves d'une classe ont le même nombre d'évaluations dans une composante. L'opérateur prend en entrée les deux profils et un ensemble de deux dates constituant un intervalle. L'une ou l'autre de ces deux dates peut être nulle, dans le cas où l'on voudrait faire l'intersection entre des évaluations antérieures ou postérieures à une certaine date. L'opérateur renvoie les deux profils dont la structure sera restée la même mais dont toutes les évaluations de la composante donnée et dans l'intervalle de dates sont communes aux deux profils.

Voici une définition semi-formelle de l'opérateur d'intersection d'évaluations sur un élément :

Soient P_1 et P_2 deux profils. Soient d_1 et d_2 deux dates telles que $d_1 < d_2$.
 On a : $P_3 \leftarrow \text{InterEvalP}(P_1, P_2, \{d_1, d_2\})$ et $P_4 \leftarrow \text{InterEvalE}(P_2, P_1, \{d_1, d_2\})$, P_3 et P_4 étant des profils tels que $P_3 = (P_1 \setminus E_1) \cup E_3$, $P_4 = (P_2 \setminus E_2) \cup E_3$ et $\forall E_1 \in \mathcal{F}_1, E_2 \in \mathcal{F}_2$ avec $\text{identifiant}(E_1) = \text{identifiant}(E_2)$ et \forall évaluation $x_i \in E_1 : x_i \in E_3 \Leftrightarrow x_i \in E_2$.

6.4 Annexe : Extrait de cPMDLe

Le tableau ci-dessous présente toutes les contraintes sur profils définies. Les contraintes en gras sont celles que j'ai définies pour utiliser l'évolutivité des profils d'apprenants.

Nom de la contrainte	Modèle de définition
Contrainte sur profils portant sur une valeur textuelle	cPMDL
Contrainte d'évolution sur profils portant sur une valeur textuelle	cPMDLe
Contrainte sur profils portant sur une valeur numérique	cPMDL
Contrainte d'évolution sur profils portant sur une valeur numérique	cPMDLe
Contrainte sur profils portant sur un entier positif	cPMDL
Contrainte d'évolution sur profils portant sur un entier positif	cPMDLe
Contrainte sur profils portant sur un élément	cPMDL
Contrainte d'évolution sur profils portant sur un élément	cPMDLe
Contrainte sur profils portant sur un nombre d'occurrences	cPMDL
Contrainte d'évolution sur profils portant sur un nombre d'occurrences	cPMDLe

➤ Contrainte d'évolution sur profils portant sur une valeur

- contrainte d'évolution sur profils portant sur une valeur textuelle :

$$CPE_{V(T)} = \{Elt, E, \{D_1, b_1\}, \{D_2, b_2\}, \{\{i, b_i\}, \{s, b_s\}\}, 0 \leq i \leq s \leq \text{card}(\text{liste_valeurs}), D_1 < D_2\}$$

Définir une contrainte évolutive portant sur une valeur textuelle consiste à spécifier, pour un élément Elt du profil, un caractère E appartenant à $\{P, S, R\}$, où P signifie progression, S stabilité et R régression, ainsi que quatre couples $\{D_1, b_1\}$, $\{D_2, b_2\}$, $\{i, b_i\}$ et $\{s, b_s\}$, où D_1 et D_2 sont des dates, i et s sont des entiers et b_1, b_2, b_i et b_s sont des booléens. Evaluer cette contrainte pour un profil évolutif d'apprenant revient à comparer les valeurs relatives à certaines évaluations portant sur Elt .

Si les booléens b_1 et b_2 ont pour valeur *false*, cela signifie que la contrainte ne portera que sur les deux évaluations ayant pour date D_1 et D_2 . On va alors comparer la valeur V_1 relative à l'évaluation portant sur Elt à la date D_1 et la valeur V_2 relative à l'évaluation portant sur Elt à la date D_2 .

Si E a pour valeur P ou R , il faut ensuite préciser un intervalle. Dans le cas d'une valeur textuelle, cet intervalle représentera l'écart de niveaux entre les valeurs relatives aux évaluations. L'intervalle sera donc inclus dans $[0, \text{card}(\text{liste_valeurs})]$, où $\text{card}(\text{liste_valeurs})$ est le nombre de niveaux de l'échelle de la valeur concernée. Par exemple, pour l'échelle (acquis, en cours d'acquisition, non acquis), un apprenant passant de non acquis à acquis a progressé de deux niveaux. Seules les échelles textuelles graduées sont concernées par ce type de contraintes, en effet on ne peut pas mesurer la progression sur une échelle non graduée telle que (bavard, actif, passif, attentif). b_i et b_s sont des booléens indiquant si les bornes de l'intervalle sont incluses ou non. Si E a pour valeur P , pour progression, on va vérifier si l'écart entre V_1 et V_2 se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si E a pour valeur R , pour régression, on va vérifier si l'écart entre V_2 et V_1 se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si E a pour valeur S , pour Stabilité, on vérifiera si V_2 et V_1 sont identiques.

Si l'un des booléens b_1 ou b_2 a pour valeur *true*, cela signifie que la contrainte portera sur une période et non plus sur seulement deux dates. Si b_1 a pour valeur *true*, cela signifie que l'on va considérer toutes les évaluations dont la date est supérieure à D_1 et si b_2 a pour valeur *true*, cela signifie que l'on va considérer toutes les évaluations dont la date est inférieure à D_2 . Soient V_1, V_2, \dots, V_n les valeurs relatives aux évaluations dans la période considérée.

Si E a pour valeur P , pour progression, on va vérifier si l'écart entre V_i et V_{i+1} pour tout i appartenant à l'intervalle $[0, n[$ se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si E a pour valeur R , pour régression, on va vérifier si l'écart entre V_{i+1} et V_i pour tout i appartenant à l'intervalle $[0, n[$ se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$. Si

E a pour valeur S, pour Stabilité, on va vérifier que l'écart entre la plus petite et la plus grande des valeurs considérées se situe dans l'intervalle formé par $\{\{i, b_i\}, \{s, b_s\}\}$.

- contrainte d'évolution sur profils portant sur une valeur numérique :

$$CPE_{V(N)} = \{Elt, E, \{D_1, b_1\}, \{D_2, b_2\}, \{\{i, b_i\}, \{s, b_s\}\}\}, i \leq s, D_1 < D_2$$

La définition d'une contrainte d'évolution portant sur une valeur numérique, que cette valeur soit un nombre ou une date, est similaire à celle d'une contrainte d'évolution portant sur une valeur textuelle. Dans le cas d'une valeur numérique, l'intervalle défini par $\{i, b_i\}$ et $\{s, b_s\}$ devra être inclus dans l'intervalle initial défini par l'échelle. Il précisera l'écart qu'il doit y avoir entre les valeurs relatives aux évaluations considérées de la même manière que pour les valeurs textuelles.

- contrainte d'évolution sur profils portant sur un entier positif :

$$CPE_{V(E)} = \{Elt, E, \{D_1, b_1\}, \{D_2, b_2\}, \{\{i, b_i\}, \{s, b_s\}\}\}, 0 \leq i \leq s, D_1 < D_2$$

La définition d'une contrainte d'évolution portant sur un entier et celle portant sur une valeur numérique ne diffère que par les valeurs pouvant être prises par i et s . Dans le cas d'un entier positif, i et s seront des entiers positifs avec toujours s supérieur ou égal à i mais s peut cette fois être égal à l'infini. Dans ce cas, le booléen b_s prendra la valeur false.

➤ Contrainte d'évolution sur profils portant sur un élément

Le but d'une contrainte sur profils portant sur un élément est de contraindre la valeur d'un élément composé du profil des apprenants. Or cet élément n'a pas de valeur propre, celle-ci va donc être calculée en fonction de certaines ou de toutes les valeurs de ses sous-éléments. Les sous-éléments peuvent avoir plusieurs valeurs n'ayant pas forcément la même échelle, il va donc falloir les convertir dans une échelle unique.

Quel que soit l'élément du profil sur lequel est définie la contrainte d'évolution sur profils portant sur un élément, le format de la contrainte est toujours le suivant :

$$CPE_E = \{Elt, E, \{D_1, b_1\}, \{D_2, b_2\}, (echelle_1, \dots, echelle_n), Ech_R, Op, CPv\}, D_1 < D_2$$

Définir une contrainte évolutive portant sur un élément Elt du profil, consiste à spécifier un caractère E appartenant à $\{P, S, R\}$, où P signifie progression, S stabilité et R régression, ainsi que quatre couples $\{D_1, b_1\}$, $\{D_2, b_2\}$, $\{i, b_i\}$ et $\{s, b_s\}$, où D_1 et D_2 sont des dates, i et s sont des entiers et b_1, b_2, b_i et b_s sont des booléens. Evaluer cette contrainte pour un profil évolutif d'apprenant revient à comparer les valeurs relatives à certaines évaluations portant sur Elt . Comme pour les contraintes d'évolution portant sur une valeur, les valeurs concernées par la contrainte dépendent des couples $\{D_1, b_1\}$, $\{D_2, b_2\}$. Le type de l'échelle résultante Ech_R dépend du nombre de valeurs choisies et du type de l'échelle de chacune. La conversion permettant de passer des échelles d'origine à l'échelle résultante est faite automatiquement, de la même manière que pour les contraintes sur profils portant sur un élément défini dans $cPMDL$, [Lef09].

La contrainte sur profils portant sur une valeur CPE_V respectera le format associé au type de l'échelle résultante. Pour une échelle textuelle ordonnée, le format sera celui d'une contrainte sur profils portant sur une valeur textuelle $CPE_{V(T)}$. Pour une échelle numérique, le format sera celui d'une contrainte sur profils portant sur une valeur numérique $CPE_{V(N)}$. Pour un entier positif, le format sera celui d'une contrainte sur profils portant sur un entier positif $CPE_{V(E)}$.

L'opérateur Op permettant de combiner les différentes valeurs des sous-éléments dépend du type de l'échelle résultante, ce pourra être une moyenne ou une somme. Pour chaque date d'évaluation concernée par la contrainte, on va exécuter cette opération sur toutes les valeurs des sous-éléments de Elt pour obtenir une unique valeur associée à Elt et à cette date. On aura donc obtenu des valeurs V_1, V_2, \dots, V_n relatives à Elt et aux dates d_1, d_2, \dots, d_n concernées par la contrainte.

On comparera ensuite ces deux résultats de la même manière que pour une contrainte évolutive sur profils

portant sur une valeur.

Soit par exemple, le contrainte suivante : {Mathématiques, P, {17/11/2009, false}, {01/12/2009, false}, (« note de 0 à 20 », « note de 0 à 10 », « note de 0 à 20 »), moyenne, {{{(0, false), (4, true)}}} }. Ceci signifie qu'on cherche à savoir si un élève a progressé en mathématiques de 0 à 4 points sur l'échelle résultante : « note de 0 à 20 » entre l'évaluation du 17/11/2009 et celle du 01/12/2009.

Soit le profil suivant :

Dates d'évaluations :	17/11/2009	01/12/2009
Composantes de Mathématiques :		
Algèbre (« note de 0 à 20 »)	12	17
Analyse (« note de 0 à 10 »)	6.5	5.75
Arithmétique (« note de 0 à 20 »)	14	19

En moyenne, l'élève a obtenu une note de 13 sur 20 pour l'évaluation du 17/11/2009 et 15.8 pour l'évaluation du 01/12/2009. Cet élève satisfait donc la contrainte puisqu'il a progressé de 2.8 points, $0 < 2.8 \leq 4$.

➤ Contrainte d'évolution sur profils portant sur un nombre d'occurrences

Une contrainte sur profils portant sur un nombre d'occurrences peut permettre par exemple de sélectionner les apprenants maîtrisant 3 des 4 sous-éléments d'un élément donné de leur profil, ou encore ceux ayant une note comprise entre 0 et 5 pour la moitié des sous-éléments d'un élément donné de leur profil. Ce type de contraintes compte donc le nombre d'occurrences de valeurs incluses dans certains ensembles de valeurs, tels que {maîtrisé} ou $\{[0, 5]\}$.

Une contrainte d'évolution sur profils portant sur un nombre d'occurrences va donc permettre de comparer l'évolution de ce nombre d'occurrences entre certaines évaluations déterminées par les couples $\{D_1, b_1\}$ et $\{D_2, b_2\}$, de la même manière que pour les contraintes d'évolution portant sur une valeur ou sur un élément. Le format de cette contrainte est une combinaison du format d'une contrainte sur profils portant sur un nombre d'occurrences défini dans cPMDL [Lef09] et de celui d'une contrainte d'évolution sur profil portant sur une valeur :

$$CPE_0 = \{Elt, E, \{D_1, b_1\}, \{D_2, b_2\}, ((SsElt_1, CP_1), \dots, (SsElt_n, CP_n)), \{\{i, b_i\}, \{s, b_s\}\}\}, n \geq 2, i, s \in N, i \leq s \leq n, D_1 < D_2$$

$SsElt_i$ est un sous-élément de Elt pouvant être une valeur. Dans ce cas la contrainte sur profils CP_i respectera le format d'une contrainte sur profils portant sur une valeur définie par cPMDL. Si $SsElt_i$ est un élément, la contrainte sur profils CP_i respectera le format d'une contrainte sur profils portant sur un élément défini par cPMDL.

Pour chaque évaluation considérée, on va calculer le nombre d'occurrences de l'élément spécifié de la même manière que pour appliquer une contrainte sur le nombre d'occurrences défini dans cPMDL. On va donc obtenir des valeurs V_1, V_2, \dots, V_n relatives aux dates concernées par la contrainte. On pourra alors effectuer une comparaison entre ces valeurs de la même manière que pour une contrainte d'évolution portant sur une valeur.