



HAL
open science

Models and tools to personalize activities on learners profiles

Blandine Ginon, Stéphanie Jean-Daubias

► **To cite this version:**

Blandine Ginon, Stéphanie Jean-Daubias. Models and tools to personalize activities on learners profiles. ED-MEDIA 2011 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, Jun 2011, Lisbonne, Portugal. pp.3321-3330. hal-01354427

HAL Id: hal-01354427

<https://hal.science/hal-01354427v1>

Submitted on 9 Jun 2023

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

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

Models and tools to personalize activities on learners profiles

Blandine Ginon, Stéphanie Jean-Daubias
University of Lyon, CNRS
University of Lyon 1, LIRIS, UMR5205, F-69622, France
bginon@gmail.com, Stephanie.Jean-Daubias@liris.univ-lyon1.fr

Résumé: In research on the open learner model, more and more ILE offer to learners interactive activities on their profile. These activities are based on the visualization of learners profiles: in particular, it may be a simple observation of a profile, but also the negotiation of this profile content, or a comparison with another learner profile. However, the visualization of profiles proposed by existing ILE is few customizable. In this paper, we present models to personalize activities on profiles, depending on the user type (learner, teacher...), on his knowledge, skills and capacities, but also on his preferences. We implemented these models to provide tools to teachers to prepare personalized activities on profiles; and to provide to each user type a unified environment proposing personalized activities on profiles.

Introduction

A learner profile is a set of computer data gathered from information, computerized or not, about a learner (Eyssautier-Bavay & al. 2009). More specifically, the learning profile of a learner concerns specifically his knowledge and skills, such as his success rate to an exam. The capacities profile of a learner concerns his physical and mental capacities, like his hearing level or the average length of concentration that can provide on a pedagogical activity.

Learners profiles have many uses, which involve different actors of learning. They can be presented to teachers as assessment after an ILE session. They also highlight the strengths and weaknesses of a learner, and, in the case of evolutive profiles they show the progression of a learner. There are several aims for the visualization of profiles. First, visualize his own profile allows a learner to have a reflective approach (Bull & MacKay 2004), to pinpoint the knowledge and skills he should work in particular. What's more, learners become aware of what ILE perceives of their skills compared to their own perception. Some ILE propose a negotiation between the system and the learner about the content of their profile. Moreover, the visualization of learners profiles can help teachers to measure the level and the overall progression of their learners, to define remediation activities for some students in particular, or the class objectives.

Let's imagine this scenario: Alex is a teacher in a class of 5th grade, he evaluates his pupils using a profiles structure named str_5. Lucas and Olivia are pupils of this class. Alex would like to propose different activities on profiles to his pupils. He wants Leo to visualize his profile, the evolution of his profile and to compare his profile to the class profile. Alex wants Olivia to visualize her profile and to establish objectives. In addition, Alex wants a personalized visualization according to the profiles content, but also to additional information he has. For example, for Lucas, he knows that his eyesight is 7/10, he prefers textual and symbolic representations, and his favorite color is cyan. About Olivia, Alex knows that her eyesight is 10/10, she prefers graphical representations and her favorite color is orange.

State of art

Among ILE proposing a visualization of learners profiles, many have chosen to use a simple and fixed profiles structure. Some ILE represent the structure of a profile as a non-hierarchical list of values. This type of representations would be inappropriate for profiles whose structure contains varied elements. Other ILE propose a visualization of profiles with a hierarchical form, which may be appropriate to more varied profiles structures. Nevertheless, we didn't find any system that allows the teacher to determine what should be

viewable in a profile. Moreover, in many ILE, the profiles contain only elements sharing a same scale, associated with the same representation. Thus, for example, profiles of SQL-Tutor and e-Kermit (Mitrovic & Martin 2007) contain only percentages represented by progress bars, and profiles of Sam Text Editor (Kay 1997) contain only elements that can be acquired or not acquired, represented by black or white boxes. However, by adopting a generic point of view, a profile must be able to contain elements associated with different scales. Furthermore, elements that share a common scale can be represented in several ways. This multiplicity of representations allows personalization of the visualization depending on the user type. For instance, Subtraction Master, intended for children from 8 to 9 years (Bull & Mackay 2004), proposes to teachers a graphical, numerical and textual visualization of profiles, while it proposes to learners a symbolic visualization of their profile, using smileys. However, the visualization is not customizable based on information about the user (knowledge, skills, preferences...), which would offers two users of the same type a different visualization, adapted to their individual needs.

If more and more ILE propose a visualization of learners profiles, sometimes customizable according to the user type, there is no generic approach that allows teachers to prepare activities on profiles by choosing which elements of a profile will be viewable by which user type and from which ways; this by offering a personalization of these activities depending on the user knowledge, skills and preferences.

Proposition

Our work is placed in the context of PERLEA project, which concerns the personalization of learning through the exploitation of heterogeneous learners profiles. This project adopts a generic approach which notably led to the definition of PMDLe, a profiles modeling language. PMDLe profiles are evolutive: they store information on different evaluation periods.

The work we present in this paper aims to propose a personalization of activities on profiles, depending on the user type, on his knowledge, skills and preferences, and appropriately to profiles with a generic structure described in PMDLe. For this, we propose a model to personalize activities on profiles. This model allows a teacher to prepare activities on profiles by defining views respecting vPMDLe model. A vPMDLe view consists of a set of associations between elements of a profile described in PMDLe and a set of profile elements representations described by rPMDLe model. Each of these representations may be subject to a set of conditions, respecting cPMDLe model of constraints on profiles (Lefevre et al. 2008). If the teacher agrees, some representations can reflect the preferences of each user, in terms of colors and fonts among others. In addition, when visualize a profile, our personalization model takes into account user preferences, for example, to select representations that are best suited to a user, among those attributed to him by the view.

Models

A detailed study, conducted both by working with teachers and by a literature review allowed us to define a typology of reflective activities on learners profiles. These activities may concerns different actors of learning: learners, their families, teachers, researchers or institutions. Visualization of profiles must be personalized to these users type. Several categories of activities on learners profiles have been identified. Some activities concern the visualization of the profiles content, like the visualization of his profile by a learner, or the visualization of the class' profile by a teacher. Some activities aim at positioning a learner compared with others. For example, the family of a learner may compare his profile with the class' profile, or institutions may compare two groups' profiles from different regions. Some activities concerns the taken of stand on the content profiles. For instance, a learner can negotiate some element of his profile with his teacher.

Models of profile elements representations: rPMDLe

A learner profile respecting PMDLe formalism consists of a set of elements specified into components. Except for elements of comments type, all components are associated with a scale. This scale can be numeric (a note of 10 or 20, a percentage...), textual graduated (acquired / not acquired, systematically / sometimes / never, ABCDEF...) or textual not graduated (talkative / active / quiet...). In a PMDLe profile, only terminal elements,

that is to say elements not containing another element, are associated with a value expressed in the scale corresponding to this element. It is nevertheless possible to obtain a value for a non-terminal element, performing an operation, sum or average, on values of the elements it contains. To enable personalized visualization of such profiles, we defined in rPMDLe representations for each type of scale, for all elements, whether terminal or not, and for comments.

To represent the value associated with an element, we offer three categories of representation: a representation for all types of elements, representations for elements associated with a numerical scale and representations for elements associated with a textual scale. Thereafter, *elt* is an element of the profile and *s* is a scale, *f* is a font associated with a name, a style, a size and a color and *dev* is a device of another type, for example a sound file or a device for taking account of disability (such as a Braille keyboard). An asterisk after an element indicates it is optional. Examples of each representations type are given in the next section.

To represent all types of elements, we propose a general representation, textual or numerical, noted Rgen (cf. Figure 1). This type of representations is used to associate to an element, and possibly to its scale, the formatting instructions that will be used to represent the value of this element, or the text of the comment. *f* represents the font and *col* the background color. The boolean *disp* controls whether or not we want to display the scale associated with the element; *disp* must be false if no scale is specified. As for all representations type, it is also possible to associate a set of *dev* to the representation, to allow for example a reading of the commentary or value of this element by a suitable device.

$$\text{Rgen}(elt, s^*) = (\text{disp}, f, \text{col}, \{\text{dev}_0, \dots, \text{dev}_m\}^*)$$

Figure 1: Formal definition of general representations

To represent elements of numerical scale, we offer three representations types (cf. Figure 2): numerical representation RnSn, graphical representation RgSn, and symbolic representation RsSn. In each case, the representation allows defining any number of numerical intervals $[a_i, b_i]$ or $[a_i, b_i[$ covering the definition interval $[\text{min}, \text{max}]$ of the scale *s* associated with *elt*, with $a_0 = \text{min}$, $b_n = \text{max}$, $b_i = a_{i+1}$, $a_i < b_i$ and $0 < i \leq k$, where *k* is an integer.

For an element of numerical scale, a numerical representation RnSn allows to define a layout for each interval. A graphical representation RgSn allows associating to each interval a color *col*, a boolean *grad* indicating whether or not this color will be gradated, and a graphic object *graph* (progress bar, thermometer, counter...). A symbolic representation RsSn allows associating to each interval one or more images *Img* (smiley, arrow, medal...) associated with characteristics: number, size, and color (that can be the default color of the image: def).

$$\begin{aligned} \text{RnSn}(elt, s) &= (([a_0, b_0[, \text{disp}_0, f_0, \text{col}_0, \{\text{dev}_{00}, \dots, \text{dev}_{0j}\}^*], \dots, ([a_k, b_k], \text{disp}_k, f_k, \text{col}_k, \{\text{dev}_{k0}, \dots, \text{dev}_{km}\}^*))) \\ \text{RgSn}(elt, s) &= (([a_0, b_0[, \text{col}_0, \text{grad}_0, \text{graph}_0, \{\text{dev}_{00}, \dots, \text{dev}_{0j}\}^*], \dots, ([a_k, b_k], \text{col}_k, \text{grad}_k, \text{graph}_k, \{\text{dev}_{k0}, \dots, \text{dev}_{km}\}^*))) \\ \text{RsSn}(elt, s) &= (([a_0, b_0[, \{\text{img}_{00}, \dots, \text{img}_{0j}\}, \{\text{dev}_{00}, \dots, \text{dev}_{0j}\}^*], \dots, ([a_k, b_k], \{\text{img}_{k0}, \dots, \text{img}_{kj}\}, \{\text{dev}_{k0}, \dots, \text{dev}_{km}\}^*))) \end{aligned}$$

Figure 2: Formal definition of representations of elements associated with a numerical scale

To represent elements of textual scale, we offer three types of representations (cf. Figure 3), similar to the three representations types for elements of numerical scale: textual representation RtSt, graphical representation RgSt, and symbolic representation RsSt. The difference between the representations for elements of numerical scale and those of textual scale is that there are *n* scale levels *lev*, and not intervals. In addition, to represent graphically the level of a textual scale, it is necessary to specify an integer that corresponds to this level, which is equivalent to convert a textual scale to numerical scale.

$$\begin{aligned} \text{RtSt}(elt, s) &= ((\text{lev}_0, \text{disp}_0, f_0, \text{col}_0, \{\text{dev}_{00}, \dots, \text{dev}_{0j}\}^*), \dots, (\text{lev}_n, \text{disp}_n, f_n, \text{col}_n, \{\text{dev}_{n0}, \dots, \text{dev}_{nm}\}^*)) \\ \text{RgSt}(elt, s) &= ((\text{lev}_0, \text{col}_0, \text{grad}_0, \text{graph}_0, \{\text{dev}_{00}, \dots, \text{dev}_{0j}\}^*), \dots, (\text{lev}_n, \text{col}_n, \text{grad}_n, \text{graph}_n, \{\text{dev}_{n0}, \dots, \text{dev}_{nm}\}^*)) \\ \text{RsSt}(elt, s) &= ((\text{lev}_0, \{\text{img}_{00}, \dots, \text{img}_{0j}\}, \{\text{dev}_{00}, \dots, \text{dev}_{0j}\}^*), \dots, (\text{lev}_n, \{\text{img}_{n0}, \dots, \text{img}_{nj}\}, \{\text{dev}_{n0}, \dots, \text{dev}_{nm}\}^*)) \end{aligned}$$

Figure 3: Formal definition of representations of elements associated with a textual scale

For non terminal elements, it is also possible to represent the elements they contain. For this, we have defined in rPMDLe a representations type, noted Rcomp (cf. Figure 4), allowing for example to represent a non-terminal element of a profile by a chart (bar chart, spider web chart, pie chart, areas chart...), examples of such representations are given in Figure 5. For these representations, *type* is the type of composed

representation, dim is an integer representing the number of dimensions (2 or 3), $eff3D$ is a boolean that indicates for a 2D chart if it has a relief effect, bar represents the type of bars (blocks or cylinders) and $orient$ indicates for a bar chart whether the bars will be horizontal or vertical. In addition, tit_i is a string that specifies the name of the subtitle i ; col_j is the color of the serial j , $rbgcol$ is the representation background color, $cbgcol$ is the chart background color, $gcol$ is the grid representation color, $pref$ is a boolean indicating whether or not we allow the taken into account of user preferences for colors, f_i is the font of the subtitle i .

$$Rcomp(elt, s)=(type, dim^*, eff3D^*, bar^*, orient^*, (tit_0, \dots, tit_k)^*, (col_0, \dots, col_l), rbgcol, cbgcol, gcol^*, pref, (f_0, \dots, f_m)^*)$$

Figure 4: Formal definition of representations for non terminal element

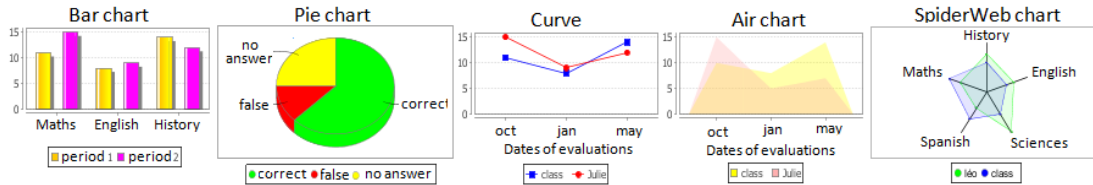


Figure 5: Examples of composed representations

Formal definition of conditions on learners using cPMDLe model

The models we have presented allow the creation of many varied representations for each element of a profile, terminal or not. However, to personalize the visualization of profiles for each user, it is necessary to associate conditions to these representations in order not to offer the same representations to all users regardless of their knowledge and capacities. Thus, we will not offer the same representation to a learner who can read and a learner who cannot read, or to a blind user and a sighted user. What's more, representations should allow for example to highlight the positive points of a profile to encourage a learner in difficulty, but also to highlight the points that can still be improved for a learner with excellent results. To allow such personalization, we have identified three types of conditions that may be associated with a representation: conditions on a learners list, on the learning profile and on the user capacities profile. For conditions on learning or capacities profiles, we chose to use the model of constraints on profiles cPMDLe (Lefevre 2009). This model allows constraining PMDLe profiles. In the case of a numerical scale, we indicate an interval which the value of the element must belong to. In the case of a textual scale, we specify what should be the level of this element. For example, cPMDLe allows selecting the learners whose score in history is between 65% and 80%, and those who master the skill poetry. Using cPMDLe on a capacities profile will also for example selecting the profiles of users whose hearing is greater than 6/10 and those whose level of use of the mouse is "rarely".

We completed cPMDLe by conditions on a learners list l_{list} (cf. Figure 6), by associating to a set of students l_i a boolean $inout$ indicating whether it is a condition of exclusion ($inout = false$) or a particular condition ($inout = true$). It is possible to define the conditions ((Julie Thomas), false) which means "all students except Julie and Thomas", and ((Alice), true) which means "only Alice".

$$Cl(l_{list})=((l_0, \dots, l_n), inout)$$

Figure 6: Formal definition of conditions on a learners list

Model of views on profile: vPMDLe

To propose to every users type personalized activities on profiles, we need to determine which activities will be proposed to which user type. Moreover, in order to personalize these activities based on visualization of profiles for each user, it is necessary to determine which elements of a profiles structure will be viewable and how. For this, we need to associate to these elements zero, one or several representations, themselves possibly associated with a list of conditions. It should also be possible to rename an element so it appears to the user with another title. For this, we propose a model of views definition, vPMDLe (cf. Figure 7), for a learning profiles structure $strLP$ described in PMDLe, a capacity profile structure $strCP$ described in

PMDLe, a user type $userT$, a learners list l and an activity act . This model associates to elements elt of the profiles structure one or more representations R , respecting rPMDLe. Each representation R may be associated with a set of conditions C on the learner list, or on learning or capacities profile. An example of view is given in the next section.

$$\text{View}(\text{strLP}, \text{strCP}, \text{userT}, l, \text{act}) = ((elt_0, R_{00}\{C_{000}, \dots, C_{00k}\}, \dots, R_{0l}\{C_{0l0}, \dots, C_{0lm}\}), \dots, (elt_n, R_{n0}\{C_{n00}, \dots, C_{n0p}\}, \dots, R_{nq}\{C_{nq0}, \dots, C_{nqs}\}))$$

Figure 7: Formal definition of views

Model to personalize activities on profiles

To allow a teacher to propose personalized activities on the profiles for each user, we have defined vPMDLe model (cf. Figure 8). On one hand, vPMDLe allows the teacher to express his preferences for the visualization of a profiles structure described in PMDLe for a users type. These choices represent a view that defines which elements of the profile will be viewable by which users and how (cf. ① Figure 8). On the other hand, vPMDLe allows the taken into account of the user choices and preferences. Indeed, according to the view defined by the teacher, zero, one or several representations of an element elt of the profile are enabled for a user. These representations are those defined by the teacher with rPMDLe for this element, and where the potential conditions are verified. Each of these enabled representations is then associated with a priority degree deg reflecting the user preferences (cf. ② Figure 8). Finally, the n representations of highest degree are used to represent the element, n is an integer expressed in the user preferences (cf. ③ Figure 8).

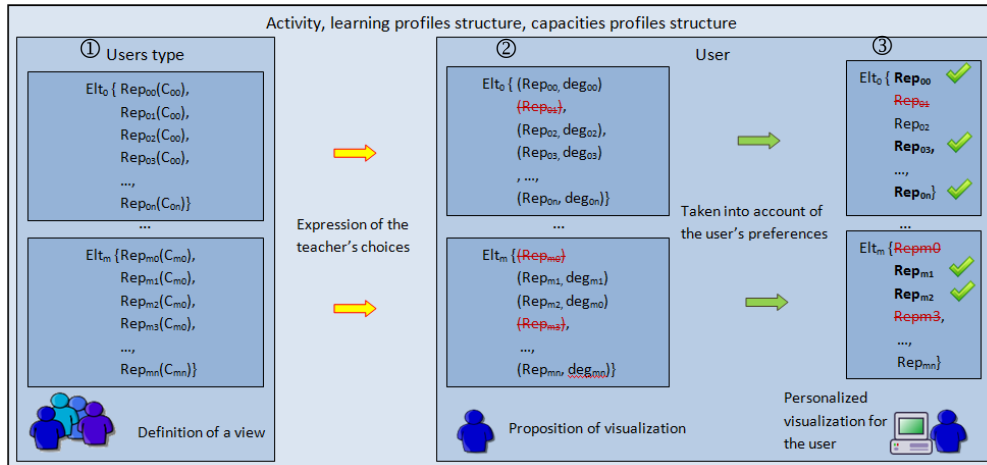


Figure 8: Model to personalize activities on profiles

Implementation

EPROFILEA environment

We implemented these different models in EPROFILEA environment. EPROFILEA concerns the different actors of learning, even if it is intended primarily for teachers. It contains three parts, composed of several modules (cf. Figure 9).

The first part of EPROFILEA, the profiles preparation, consists to establish the structure of the profiles that we want to handle, before integrating data from external profiles to build learners profiles in accordance with the teacher's wishes and respecting EPROFILEA formalism. PMDLe language is operationalized in Bâtitseur module that allows defining learners profiles structures in a unified formalism. Profiles structures from Bâtitseur will be completed in data integration modules: Prose and Tornade. Prose module helps teachers to enter data from paper and pencil profiles of each of their pupils according to the profiles structure defined in

Bâtisseur. To integrate external software profiles (from outside EPROFILEA), Tornade module allows experts to set up parsers, then to teachers to use this parsers to integrate data from corresponding profiles to profiles respecting EPROFILEA formalism.

The second part of EPROFILEA is dedicated to profiles transformations. These transformations may relate to operations on their structure or their data (Group module allows among others establishment of a group profile from the profiles of learners in the group) or prepare the visualization of profiles (Regards module enables a teacher to define different views on profiles).

The third part of the environment, the profiles exploitation, offers personalized activities on the profiles, with Perl, and personalized pedagogical activities with Adapte. The objective of Perl is to enable interactive visualization of profiles by the different actors according to the views defined by the teacher in Regards. It also proposes to learners, in addition to visualization of profiles, activities on their profiles (rewording, negotiating elements of the profile...) allowing the learner to enter into a reflexive approach in relation to his learning, and so better assimilate and exploit the information provided to him. Adapte module aims to give teachers the means to personalize the pedagogical activities proposed to their students according to their needs based on his pedagogical choices (Lefevre & al., 2008).

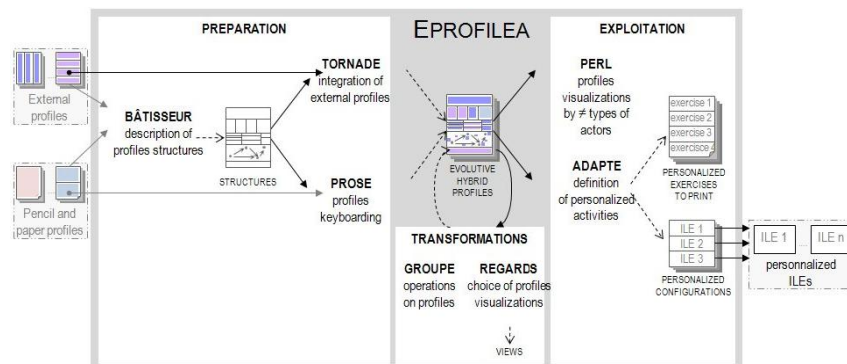


Figure 9: Architecture of EPROFILEA

Regards module

Regards module allows the teacher to define views on the profiles for each activity and each user type. For this, Regards implements vPMDLe view model, which assigns to each element of a given profiles structure for an activity, some representations, possibly associated with a list of conditions which may relate to the learners list, to their learning profile or to the user capacities profile. In addition, Regards allows the teacher to define personalized representations respecting rPMDLe.

Let's come back to our example. With Regards, Alex defines several representations. He defines two general representations: Rgen1 and Rgen2, whose definitions are given in Figure 10. These representations are very similar: they make it possible to write text in Arial black on a white background; only the font size varies from one representation to another, 12 for Rgen1 and 16 for Rgen2. Figure 11 gives an example of their use.

Rgen1(elt)=(false, (Arial, normal, 12, black), white) Rgen2(elt)=((Arial, normal, 16, black), white)

Figure 10: Formal definition of representations Rgen1 and Rgen2

Textual or numeric representation Rgen1	Textual or numeric representation Rgen2
This is an example	This is an example

Figure 11: Example of general representations

Alex also defines three representations adapted to elements of scale *mastery(three levels)*, formally defined in Figure 12, and illustrated in Figure 13. The first representation, RtSt, is textual. It can textually represent the level of mastery of a learner for an element as text formatted as specified: the level will be displayed in green if the element represented is mastered or partially mastered by the learner, in red if not.

Moreover, the level mastered will be in bold type. The second representation, RgSt1, is graphic. It can represent the level of mastery of a learner for an element with a blue progress bar. This progress bar will be filled to 100% if the element is mastered, to 50% if it is partially mastered and to 0% if it is not mastered. The third representation, RsSt1, is symbolic. It allows associating to an element a traffic light, green if the element is mastered, orange if it is partially mastered and red if it is not mastered.

RtSt1 (elt, mastery(three levels)) = ((mastered, false, (Arial, bold, 14, green), white), (partially mastered, false, (Arial, normal, 12, green), white), (not mastered, false, (Arial, normal, 12, red), white))
RgSt1 (elt, mastery(three levels)) = ((mastered, 100, blue, false, progress bar), (partially mastered, 50, blue, false, progress bar), (not mastered, 0, blue, false, progress bar))
RsSt1 (elt, mastery(three levels)) = ((mastered, traffic_light 1(1, 1, def)), (partially mastered, traffic_light 2(1, 1, def)), (not mastered, traffic_light 3(1, 1, def)))

Figure 12: Formal definition of representations RtSt1, RgSt1 and RsSt1







Scale : mastery(three levels)	Textual representation RtSt1	Graphic representation RgSt1	Symbolic representation RsSt1
mastered	mastered		
Partially mastered	Partially mastered		
Non mastered	Non mastered		

Figure 13: Examples of representations applying RtSt1, RgSt1 and RsSt1

In addition, Alex defines four representations adapted to elements of scale $note(0, 20)$, formally defined in Figure 14 and illustrated in Figure 15. The two first representations, RnSn1 and RnSn2, are numerical. RnSn1 focus on points to improve, representing in red bold notes below 10, while RnSn2 highlight positive points, representing in green bold notes above 10. The third representation RgSn1 is graphic. It is used to associate to each note a progress bar, filled in proportion to the note and which color depends on the interval in which the note is. The fourth representation, RsSn1 is symbolic. It allows associating to an element a smiley or two. It will be a smiling smiley if the note of the learner for this element is greater than 10, a sad smiley if not.

RnSn1 (elt, note(0...20) = ([[0, 10[, true, (Arial, bold, 12, red), white), ([10, 20], true, (Arial, normal, 12, black), white))
RnSn2 (elt, note(0...20) = ([[0, 10[, true, (Times, normal, 16, black), white), ([10, 20], true, (Times, bold, 16, green), white))
RgSn1 (elt, note(0...20) = ([[0, 6[, red, false, progress bar), ([6, 10[, orange, false, progress bar), ([10, 14[, light green, false, progress bar), ([14, 20[, green, progress bar))
RsSn1 (elt, note(0...20) = ([[0, 6[, smiley1(2, 1, def)), ([6, 10[, smiley1(1, 1, def)), ([10, 14[, smiley2(1, 1, def)), ([14, 20[, smiley2(2, 1, def))

Figure 14: Formal definition of representations RnSn1, RnSn2, RgSn1 and RsSn1







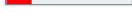

Scale : Note(0 ... 20)	Numeric representation RnSn1	Numeric representation RnSn2	Graphic representation RgSn1	Symbolic representation RsSn1
[14, 20]	16/20	16/20		
[10, 14[11/20	11/20		
[6, 10[9/20	9/20		
[0, 6[4/20	4/20		

Figure 15: Examples of representations applying RnSn1, RnSn2, RgSn1 and RsSn1

Finally, Alex defines a composed representation Rcomp1, whose formal definition is given in Figure 16. Rcomp1 allows representing a set of elements with a 3D bar chart with horizontal cylinders bars. For colors, Rcomp1 takes into account user preferences. If the user has no preferences, default colors will be used.


```
Rcomp1 = (bar chart, 3, null, cylinder, vertical, ("",""), (green), white, white, light gray, true, (Arial, normal, 16, black))
```

Figure 16: Formal definition of representation Rcomp1

Alex also defines two conditions, formally specified using cPMDLe (Figure 17). The condition Clp1 concerns the learning profile and selects learners whose average for the component *Mathematics* is equal or greater than 10. Ccp1 is a condition on the capacities profile that selects users which eyesight is greater than 8.

```
Clp1 = {Mathematics, ((10, true), (20, true))}           Ccp1 = {Eyesight, ((8, false), (10, true))}
```

Figure 17: Formal definition of conditions Clp1 and Ccp1

After defining personalized representations and conditions he will use, Alex defines a view for learners for the activities he wants to propose to them. For example, for the activity of visualization of their profiles by learners, Alex defines with Regards the view View5, formally represented on Figure 18 and synthetically on Figure 19.

```
View5(Str_5, StrC, learners, visualisation)={
(English>Grammar, RtSt1, RgSt1, RsSt1),
(English >Conjugation, RtSt1, RgSt1, RsSt1),
(English >Spelling, RtSt1, RgSt1, RgSt1),
(Mathematics, Rcomp1),
(Mathematics>Comments, Rgen1{Ccp1}, Rgen2{non Ccp1}),
(Mathematics>Calculation, RsSn1),
(Mathematics>Calculation>Addition,
  RnSn1{Clp1}, RnSn2{non Clp1}, RgSn1),
(Mathematics>Calculation >Multiplication,
  RnSn1{Clp1}, RnSn2{non Clp1}, RgSn1),
(Mathematics>Algebraic exercises, « Problems »,
  RnSn1{Clp1}, RnSn2{non Clp1}, RgSn1),
```

Figure 18: Formal definition of View5

```
English
Comments
Grammar      RtSt1; RgSt1; RsSt1
Conjugation  RtSt1; RgSt1; RsSt1
  groupe1
  groupe2
  groupe3
Spelling     RtSt1; RgSt1; RsSt1
Mathematics
Comments     if Ccp1 then Rgen1, else Rgen2
Calculation  NRS1
  Additions  if Clp1 then RnSn1 else RnSn2; RgSn1
  Multiplications if Clp1 then RnSn1 else RnSn2; RgSn1
Algebraic exercises « Problems »;
  if Clp1 then RnSn1 else RnSn2; RgSn1
```

Figure 19: Synthetic representation of View5

For the component *English*, View5 associates to *Grammar*, *Conjugation* and *Spelling* the representations RtSt1, RgSt1 and RsSt1. *Conjugation* is composed of two elements (*regular* and *irregular*), so the value associated with *Conjugation* is the average of the values of these elements. Let's note that *Comments*, *regular* and *irregular* are not associated with any representation, so they will not be visible by learners during the activity of visualization. For the component *Mathematics*, View5 associates to the whole component the composed representation Rcomp1. It will represent its constituent elements, *Calculation* and *Algebraic exercises* (element of *Comments* type cannot be represented by a composed representation). View5 associates with *Comments* the representation Rgen1 if the user's eyesight is greater than 8/10, and the representation Rgen2 else. *Calculation* is associated with RsSn1, as it is a non terminal element, its value will be the average of the values of its sons *Additions* and *Multiplications*. *Additions*, *Multiplications* and *Algebraic exercises* are associated with RgSn1 and with RnSn1 if the average of the learner for *Mathematics* is equal or greater than 10, with RnSn2 else. What's more, *Algebraic exercises* will be presented to learners with the name *Problems*.

With Regards, Alex defined personalized representations, conditions, and a view for a profiles structure, a user type and an activity. This view can now be used in Perl to provide each user with personalized activities on profiles, taking into account the choices of teachers and the user knowledge, skills and preferences.

Perl module

Perl uses views defined in Regards to propose to a user personalized activities, among those identified in our typology of activities. Depending on the teacher's choices expressed through a view, an activity is proposed or not to the user. Thus, the proposed activities will depend on the user type, but may also vary between two users of the same type. Each activity proposed by Perl is based on the visualization of profile. This visualization is personalized for the user using the view defined by the teacher in Regards. For each viewable element of the profile, Perl generates a list of representations associated with this element and for which the conditions are verified by the user. Then these representations are ordered according to the user's preferences, to suggest representations that suit him the most among those available for him. However, the user is informed

that others representations are available for this element, allowing him to choose others representations if he wishes.

Let's come back to the example of Alex's class of 5th grade, for which he created several views. Perl module provides each pupil of the class activities on the visualization of profile. For example, Lucas can view his profile, the evolution of his profile or compare his profile with the profile of the class; Olivia can view her profile and define objectives. Let's look at the visualization of their profile by Lucas and Olivia (Figure 20).

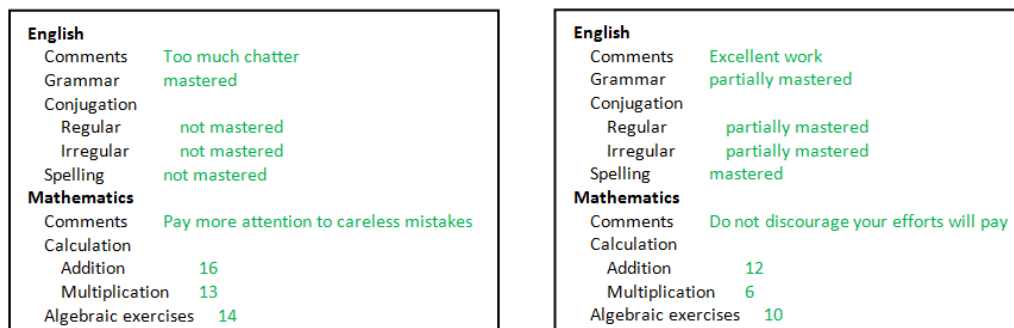


Figure 20: Lucas' and Olivia's profiles

For the component *English*, *Grammar*, *Conjugation* and *Spelling* are viewable. For these three elements, the available representations for both Lucas and Olivia are RtSt1, RgSt1 and RsSt1. Lucas prefers textual or symbolic representations, so RtSt1 and RsSt1 are proposed to him (cf. left part of Figure 21). Olivia prefers graphical representations, so RgSt1 is proposed to her (cf. right part of Figure 21). However, both Lucas and Olivia can choose themselves another combination of the three available representations if they wish.

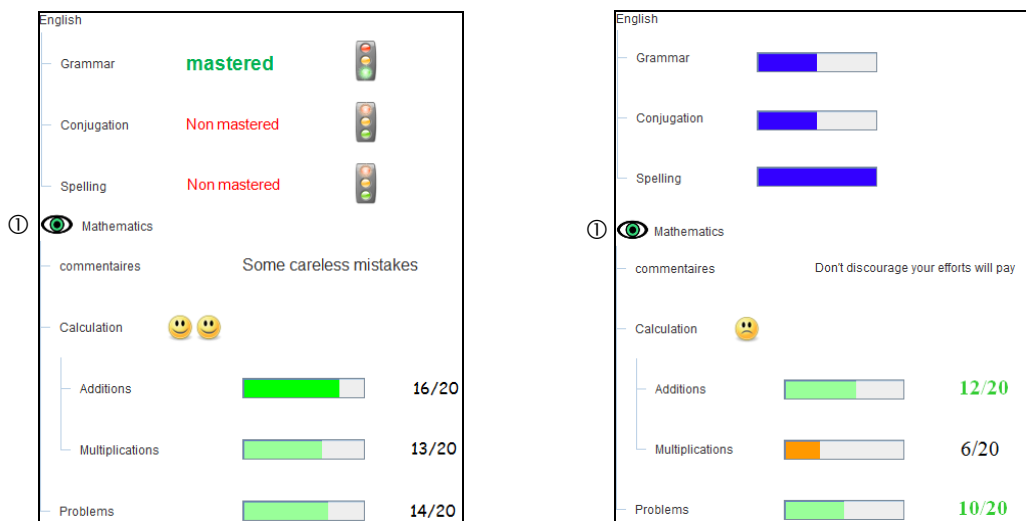


Figure 21: Visualization of their profile by Lucas (left) and Olivia (right)

For the element *Comments* of *Mathematics*, a single representation is available for Lucas whose eyesight is below 8: RtSn2 (cf. left part of Figure 21). For Olivia whose eyesight is over 8, the only available representation is RnSn1 (cf. right part of Figure 21). For *Calculation*, a single representation is available for each pupil: RsSn1. For *Additions*, *Multiplications* and *Algebraic problems*, the representations proposed for Lucas are RgSn1 and RnSn1 because he has an average of 14.25 in Mathematics, while RgSn1 and RnSn2 are proposed to Olivia who has 9.5. Let's note that *Algebraic exercises* is visible under the title *Problems*, according to the choice of Alex expressed in the view. For *Mathematics*, the composed representation Rcomp1 is also proposed for Lucas and Olivia. They can click on an icon (cf. ① Figure 21) to see a bar chart that represents the elements that compose *Mathematics*. This representation takes into account user preferences for colors. Therefore, the bars of the chart are cyan for Lucas (cf. left part of Figure 22) while they are orange for Olivia (cf. right part of Figure 22).

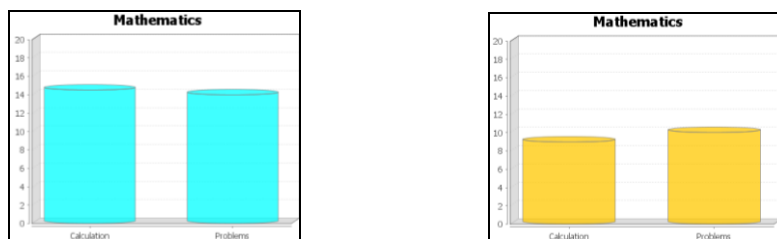


Figure 22: Composed representation associated to *Mathematics* for Lucas (left) and Olivia (right)

Conclusion

In this paper, we have presented a model of personalization of interactive activities on learners profiles. This model allows proposing to each user personalized activities on his profiles according to the teacher's choices, and to his own knowledge, skills, capacities and preferences. This model uses vPMDLe views model. vPMDLe allows associating to an element of a PMDLe profile a list of representations respecting rPMDLe representations model and possibly depending on a list of conditions respecting cPMDLe constraints on profiles model.

In addition, we have implemented this generic model in EPROFILEA environment. Regards module allows the teacher to define personalized representations and views, in order to prepare activities on the profiles. Then, Perl module allows each user of each type accessing available activities on the profiles, on a fully personalized manner, taking into account of his knowledge, his skills, his capacities and also his preferences.

We have illustrated this process by a running example. Our example shows the needs of a teacher concerning preparation of activities on learners profiles. This example also demonstrates how our models, implemented in EPROFILEA environment, satisfy this need, while proposing personalized activities on profiles, respecting the teacher's choices and each user's preferences.

Experiments of Regards and Perl modules are planned soon, in order to verify firstly that Regards effectively enables teachers to express each of their choices, and secondly that Perl allows offering to all users activities truly adapted to them.

Our approach allows using these activities on profiles with various kinds of profiles, from different sources, paper and pencil or from ILE. But these activities have been created to be done after learning activities. They cannot be done during the activity of the learner, for instance by teachers to monitor them.

Références

- Bull, S., McKay, M. (2004). *An open learner model for children and teachers: inspecting knowledge level of individuals and peers*. ITS'04, Maceio, Brazil, p. 646-655.
- Eyssautier-Bavay, C., Jean-Daubias, S. & Pernin, J.-P. (2009). *A model of learners profile management process*. AIED'09, Brighton, Great Britain. p. 265-272.
- Lefevre, M., Guin, N. & Jean-Daubias, S. (2008). *Adapte, a Tool for the Teacher to Personalize Activities*. Research report RR-LIRIS-2008-008.
- Lefevre, M., (2009). *Processus unifiés pour la personnalisation des activités pédagogiques: méta-modèles, modèles et outils*. Ph. D (in French), University Claude Bernard Lyon1.
- Mitrovic, A., Martin, B. (2007). *Evaluating the Effect of Open Student Models on SelfAssessment*. IJAIED, vol. 17, p. 121-144.

Kay, J. (1997). *Learner Know Thyself: Student Models to give Learner Control and Responsibility*. ICCE'97, Kuching, Malaysia, p. 17-24.