

²PMDL: a modeling language to harmonize heterogeneous learners profiles

Carole Eyssautier-Bavay^{1,2}, Stéphanie Jean-Daubias¹

¹ Université de Lyon, CNRS

Université Lyon 1, LIRIS, UMR5205, F-69622, France

² Université de Grenoble, LIG, BP 53, F-38041 Grenoble Cedex 9, France

Stephanie.Jean-Daubias@liris.univ-lyon1.fr

Abstract: Our work aim at proposing models and tools facilitating reuse and integration in a same environment and same profiles of heterogeneous profiles. In the research presented in this paper, we address the issue of heterogeneous profiles reuse by actors (human or software) different from their creators. These profiles can be pencil and paper or coming from software, they can be existing or to be created, and we don't know their structure. We adopt a generic approach that consists in a posteriori rewriting profiles to integrate in a same formalism: PMDL profiles modeling language. Throughout this paper, the example of an existing profile is used to illustrate our propositions.

Introduction

This research concerns the reuse of existing heterogeneous learners profiles. In this context we identify three major sources of profiles. First, national and international educational institutions are looking for common representations of data concerning learning activities. This is shown through numerous initiatives around skills frameworks, standardization works or portfolio. Then, some ILE (Intelligent Learning Environments) collect learning information in order to customize the system, to help the teacher or tutor in his follow-up task or to deliver a reflexive learning report to the learner. Finally, teachers collect information, often on pencil and paper, to evaluate learners, to propose individualized activities or to provide a view of the learning process to the learning situation actors. These various needs lead to different models of the individual information related to the learning that we call “learners profiles”.

ILE designers and teachers build heterogeneous learners profiles according to each other needs. Currently, there is no generic method allowing to reuse various rich existing profiles – created by other systems, human or not – in different contexts or practices – mixing pencil and paper and computerized profiles. Our research aims to achieve this issue.

This paper first gives an overview on the issue of reusing heterogeneous profiles. Then we present our approach and show the profiles modeling language that we propose.

Heterogeneous Learners Profiles Reuse

We define a learner profile as information concerning a learner or a group of learners, collected or deduced from one or several pedagogical activities, computerized or not. Information contained in the learner profile can concern his knowledge, meta-knowledge, abilities, conceptions or his behaviour.

The heterogeneity of existing profiles, concerning both their structure and their content, makes difficult their reuse in various contexts. The profiles reuse requires harmonizing their structure: it consists in writing the various profiles according to a common formalism.

One way to solve the heterogeneity problem consists in defining a priori a set of information about a given learner. Normalization works have chosen this approach, like PAPI, IMS-RDCEO or IMS-LIP. However, these standards do not provide enough details to describe fine information about learner's knowledge. Moreover, existing standards are focused on storage and exchange of data in order to help in managing educational institutions. This explains why standards are not precise enough to describe learner profiles. Finally, this information is stored as free text making difficult data reuse by a computer system. This is a major limitation of this approach in our context of profiles reuse. Approaches based on ontologies (Heckmann et al. 2005) or on the portfolio (Grant & Hubner, 1998) are also possible, but, as other a priori approaches, they require that learner profiles are written in a specified formalism.

Another approach (as proposed in this paper) consists in reusing external profiles within a common framework by rewriting them according to an internal formalism. ViSMod (Zapata-Rivera and Greer 2004) and DynMap systems (Rueda et al. 2006) allow the reuse of learners profiles built by an ILE. They rewrite on one hand the domain-centric data and on the other hand the specific learner data. (Ramandalahy et al. 2009) propose a model of learners profile, that can be extended if necessary. But taking into account a new item in a profile requires modification in the model and the system. Other approaches are more generic as interoperability system for user models (Celik et al. 2008), schema conversion (Rahm and Bernstein 2001) or user modeling services (Conlan et al. 2002).

But these different systems are not totally satisfying in the context of various existing learners profiles reuse. First, in order to encourage the integration of learners profiles in teacher's practices, we claim that it is necessary for him to be the major contributor in the whole profiles reuse process. This requires that he can be able to associate profiles he has created with other ones built by other teachers or by ILEs. Moreover, these systems do not adopt a generic enough approach to enable flexibility necessary to treat various rich profiles, as pencil and paper profiles. Finally, by reproducing domain specific data, previous systems do not allow to represent all information existing in learner profiles, such as free text (e.g. a comment), graphs with values suited to the links, or distribution lists.

Our approach consists in rewriting a posteriori the profiles in a common formalism, based on PMDL, the profiles modeling language that we present in this paper, thanks to semi-automatic processes. The harmonized profiles are then exploited in EPROFILEA, the environment we develop in two ways: profiles visualization and personalization of learners' activities (Jean-Daubias et al. 2009). We operationalized PMDL language in EPROFILEA environment, but it can be implemented in other systems or used in other contexts.

General Structure of PMDL Language

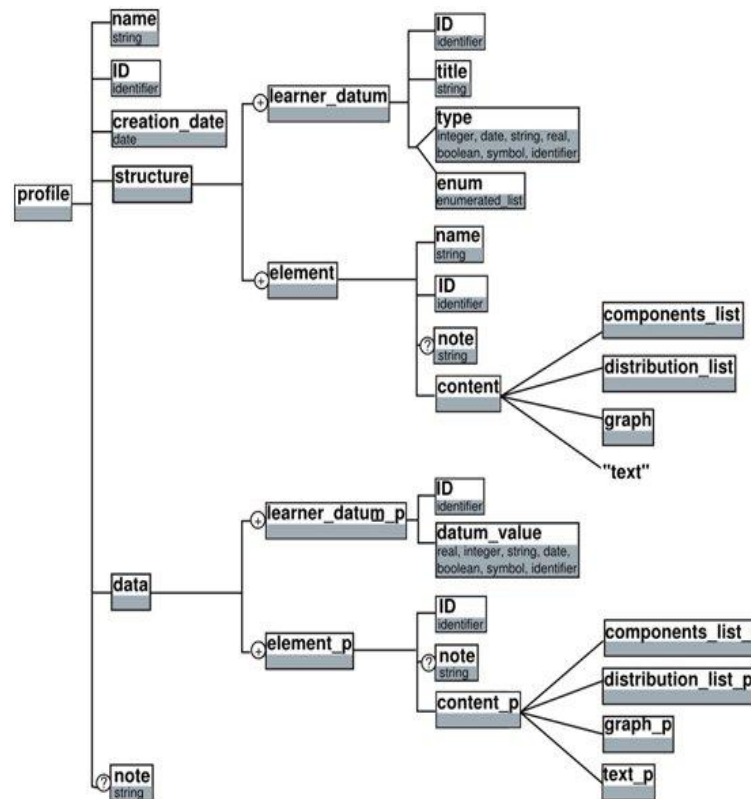


Figure 1. General structure of PMDL.

In order to propose PMDL language, we first carried out a state of the art (Eyssautier-Bavay 2008). Studied profiles come from research works, marketplace and teachers practices. We have analyzed in depth 24 profiles, selected for their richness and variety. We proposed to classify contained data in five main categories from which we based PMDL profiles modeling language. For this study, we worked with seven teachers, coming from primary school to university and continuing education. The aim of this collaboration was to collect teachers and institutional practices about learners profiles.

Specifications of PMDL language are described in BNF (Backus Naur Form) formal notation. In this paper, in order to help understanding, a graphical notation¹ is also given, based on the notation used in some IMS specifications. Figure 1 presents the general structure of PMDL according to the graphical representation.

In PMDL, a *profile* consists of a *name*, an *ID*, a *creation-date*, a *structure* part, a *data* part and an optional *note*. In a learner profile written in PMDL, its *structure* and the learner *data* are described separately. Several learners profiles can then have the same *structure*, but different *data*.

structure part consists of one or more general information about a learner: *learner-datum*. A *learner-datum* contains an *ID* (e.g. D1), a *title* (e.g. “living city”) and either a *type* (in the previous example “string”) or an enumerate list, *enum* (e.g. “Lyon, London, Geneva”). *structure* part consists also of one or more *element*. An *element* is made of a *name* (e.g. “algebra”), an *ID*, an optional *note* and a *content*. The content can be of four types: *components_list*, *distribution_list*, *graph* and *text*. These, plus *learner datum*, corresponds to the five categories identified in the state of the art. PMDL defined specifications for each one of these categories. In this paper, as an example, *distribution_list* specifications will be presented.

data part consists of a set of data concerning a given learner, as well as one or more *element_p*. A *learner_datum_p* contains an *ID* (in the previous example: D1) and a *datum_value* (e.g. “London”). There must be as many *learner_datum_p* (respectively *element_p*) as *learner_datum* (respectively *element*) previously declared. In addition, for each *ID* of a *learner_datum_p*, a *learner_datum* having the same *ID* must be previously declared. Identically, for each *ID* of an *element_p*, an *element* having the same *ID* must be declared in the *structure* part. The type of *content_p* must also be the same than the *content* type previously declared for this *ID*.

An Example of a Profile Rewritten in PMDL

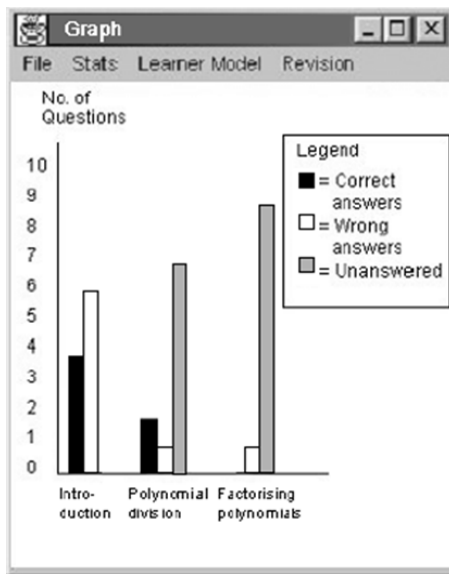


Figure 2. A part of MoreMaths profile (Bull et al. 2003).

```

element « MoreMaths » E1
distribution_list
number_of_values 1①
1, E1_C1 « Introduction »
2, leaf, E1_C2 « Correct answers »
2, leaf, E1_C3 « Wrong answers »
2, leaf, E1_C4 « Unanswered »
1, E1_C5 « Polynomial division »
2, leaf, E1_C6 « Correct answers »
2, leaf, E1_C7 « Wrong answers »
2, leaf, E1_C8 « Unanswered »
1, E1_C9 « Factorising Polynomials »
2, leaf, E1_C10 « Correct answers »
2, leaf, E1_C11 « Wrong answers »
2, leaf, E1_C12 « Unanswered »

```

Figure 3. Structure of MoreMaths profile in PMDL.

```

element_p E1
distribution_list_p
E1_C1 number_of_questions 10③
E1_C2 (4)
E1_C3 (6)
E1_C4 (0)
E1_C5 number_of_questions 10③
E1_C6 (2)
E1_C7 (1)
E1_C8 (7)
E1_C9 number_of_questions 10③
E1_C10 (0)
E1_C11 (1)
E1_C12 (9)

```

Figure 4. Data of MoreMaths profile in PMDL.

To illustrate PMDL specifications, this section presents an example of an existing profile rewritten according to PMDL. Figure 2 is part of MoreMaths profile (Bull et al. 2003) that represents the distribution of learner answers between correct, wrong and unanswered items for components: “Introduction”, “Polynomial division” and “Factorising polynomials”. Here, each component has been assessed by 10 questions. For “Introduction”, the learner had 4 correct answers, 6 wrong answers and 0 unanswered.

To rewrite this part of MoreMaths profile according to PMDL, *distribution_list* element have to be used. Indeed, it allows representing the distribution of learner answers between some components: both for *structure* (figure 3) and *data* (figure 4).

Figure 3 shows the *structure* of MoreMaths profile rewritten according to PMDL. The two first lines give the *element name* (MoreMaths) with its *ID* (E1) and its type of *content* (*distribution_list*). In the MoreMaths example (figure 2), there is only one value linked to each subcomponent (“correct answers”, “wrong answers” and “unanswered”). In accordance with

¹ In our case, graphical representation is less accurate than BNF notation, but easier for reading and understanding the concepts used.

PMDL specifications, the tag “number_of_values” precedes an integer, 1 in example ①. Then, the structure of the components tree ② is rewritten according to PMDL specifications. At the *depth* 1, the components are described with an *ID* (e.g. E1_C1) and a *title* (e.g. “Introduction”). At the *depth* 2, subcomponents are described. *leaf* tags indicate leaf nodes in the tree. Following these tags, *ID* (e.g. E1_C4) and *title* (e.g. “Unanswered”) can be found.

Figure 4 presents the **data** part of MoreMaths profile rewritten in PMDL. First line gives the *ID* of one element previously declared in the *structure* part of the profile (in this example E1, i.e. the MoreMaths element). Then, *ID* of each components declared in the structure part are listed. *ID* of components which have a depth 1 declared in the structure part (e.g. E1_C5) are followed by the tag “number_of_questions” and an integer. This indicates the number of questions in which the distribution is realized, 10 here ③. *ID* of components which have a depth 2 declared in the structure part (e.g. E1_C6) are followed by the value linked to the component in the profile ④. In the MoreMaths example, for subcomponents of “Introduction”, value attributed to E1_C2 (“correct answers”) is 4, E1_C3 (“Wrong answers”) is 6 and E1_C4 (“Unanswered”) is 0, and so on for the two other components. This **data** part gives the results obtained by the learner in example figure 2.

Specifications of the *distribution_list* Element

This section presents the PMDL specifications of the *distribution_list* element used in the previous example.

Structure part of the *distribution_list* element is shown graphically in Figure 5. Due to a lack of space, we don't detail graphical representation of the last element, *non_weighted_components_list*, but we give its details in BNF below.

```
<distribution_list>
  ::= "number_of_values" <number_of_values> sep
  <non_weighted_components_list>
  ::= "number_of_values" <number_of_values> sep <labels>
  <non_weighted_components_list>

<number_of_values> ::= <integer>

<labels> ::= <value_number> <label> sep <labels>
          ::= <value_number> <label> sep

<value_number> ::= <integer>

<label> ::= "label" <string>
```

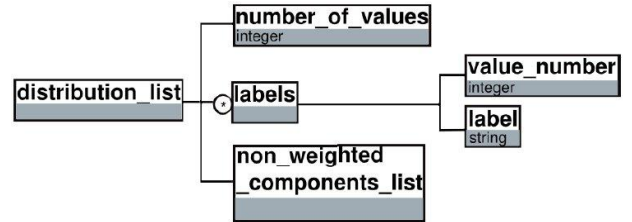


Figure 5. Structure of *distribution_list* element.

The first frame of BNF below gives the formal specifications corresponding to the figure 5. A tag “number_of_values” precedes an integer giving the number of values attributed to each component (① in figure 3). *sep* is a separator to distinguish the different elements. It can be for example instantiated in “;”, or “crlf” (Carriage Return Line Feed). In the previous example, it was instantiated in crlf. An optional *labels* precedes *non_weighted_components_list*. There are no *labels* in MoreMaths example. *labels* is a set of pair <value_number> <label> where *value_number* is an integer and *label* consists of a tag “label” followed by a string. A label is used to give sense to the diverse values linked to a component: label “distribution value”, label “success rate”. This example means that the first value linked to a component is the distribution value and the second one is the success rate of the learner to this component.

```
<non_weighted_components_list>
  ::= <non_weighted_component> sep
  <non_weighted_components_list>
  ::= <non_weighted_component> sep
  <non_weighted_component>
  ::= <depth> "," "leaf," <component>
  ::= <depth> "," <component> sep
  <non_weighted_subcomponents>
  ::= <depth> "," "leaf," <component> "," <note>
  ::= <depth> "," <component> "," <note> sep
  <non_weighted_subcomponents>
  <non_weighted_subcomponents>
  ::= <non_weighted_component> sep
  <non_weighted_subcomponents>
  ::= <non_weighted_component>

<depth> ::= <integer>
<component> ::= <ID> <title>
<ID> ::= <identifier>
<title> ::= <string>
```

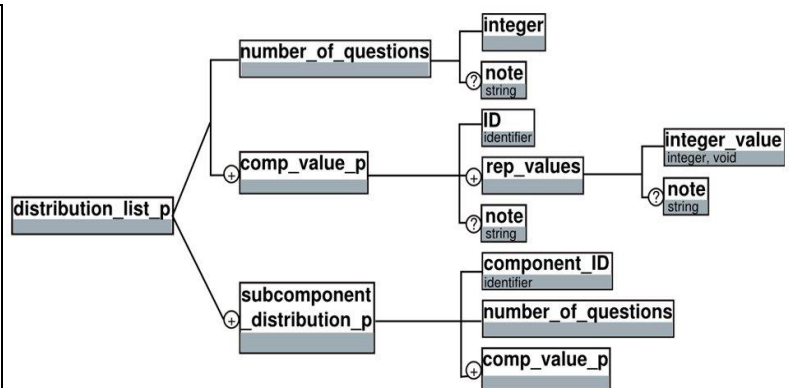


Figure 6. A part of *distribution_list* element in Bâtisseur.

The frame below gives the BNF specifications for the element *non_weighted_components_list* which is a list of *non_weighted_component* (② in figure 3). This last one first gives the depth of the component in the tree of components. If the component is a leaf of the tree of components, a tag “leaf” precedes the *component*. If it is not a leaf node of the tree, there is no tag and the *component* is followed by its child nodes, *non_weighted_subcomponents*. An optional *note* can be linked to each *component*. A component is defined by an *ID* (e.g. E1_C3) followed by a *title* (e.g. “Wrong answers”).

The *structure* part of the *distribution_list* element has been presented previously. Its *data* part will be presented in this section. First, figure 6 shows the graphical representation of distribution list *data* part: *distribution_list_p*.

```

<distribution_list_p> ::= "number_of_questions" <number_of_questions> sep <component_distribution_p>
// case of a distribution on the components.
      ::= <subcomponent_distribution_p>
// case of a distribution on the subcomponents

<number_of_questions>      ::= <integer>
                          ::= <integer> <note>

<component_distribution_p> ::= <comp_value_p> sep <component_distribution_p>
                          ::= <comp_value_p> sep

<comp_value_p> ::= <ID> "(" <rep_values> ")"
                  ::= <ID> "(" <rep_values> ")" <note>

<rep_values>      ::= <integer_value> "," <rep_values>
                  ::= <integer_value>

<integer_value>   ::= <integer>
                  ::= <void>
                  ::= <void> <note>
                  ::= <integer> <note>

<subcomponent_distribution_p>
      ::= <component_ID> "number_of_questions" <number_of_questions> sep <component_distribution_p> <subcomponent_distribution_p>
      ::= <component_ID> "number_of_questions" <number_of_questions> sep <component_distribution_p>

<component_ID> ::= <identifier>

```

A *distribution_list_p* element may be of two types. The first one corresponds to the case of the values distribution on the components (depth 1), and the second one corresponds to the case of the distribution on the subcomponents (depth 2) (like the MoreMaths example figure 2). In the first case, the tag “number_of_questions” is followed by an integer in which the distribution is realized and by the list of components with their distribution values.

In the second case the number of questions (used to realized the distribution of the learner answers) is linked to the components which have been declared at a depth 1 in the *structure* part (③ in figure 4) and the values are associated to the components which have a depth 2 (④ in figure 4). *comp_value_p* is an *ID* of a *component* declared in the *structure* part followed by a list of values, *rep_values*. These values are void or integer (a distribution value is a count value so it is an integer). They can be linked to a *note*.

Concluding remarks

An application framework of PMDL language has been defined. It qualifies which type of information can be described with PMDL: for example, raw data (as log files) cannot be described with PMDL, while analyzed data about learner knowledge can. Ten criteria have thus been defined (*raw data*, *temporality*, *time spent on activities*, *learner’s behavioral model*, *peer’s data*, *data from collaborative activities*, *learner’s productions*, *domain model*, *relations between the profile items*, *analyzed data about learner*). Then, PMDL language expressive power has been evaluated within this application framework, for the 24 studied profiles, as MoreMaths example presented in this paper.

All of these profiles consisted of at least one *learner_datum* element in order to identify the learner. *components_list* element is used in all studied profiles coming from teachers practices or from the marketplace. Indeed these profiles are based on skills frameworks which are lists of components. *text* element is used in teachers practices to add comments about a learner as well as in several profiles coming from research works (for example to describe misconceptions like in C-Polmile). *graph* element is used in a few profiles, mostly to represent Bayesian networks. Finally, *distribution_list* element is seldom used (only 2 profiles among 24). These last two elements only exist in studied profiles coming from research works.

The only data existing in the studied profiles which cannot be described with PMDL are set outside of our application framework. For example, profiles using Bayesian networks (like ViSMod), cannot be rewritten entirely with PMDL: indeed, PMDL allows describing learner data about knowledge, but not about the knowledge model which created them.

Many actors, human and systems, produce information about learning. But reusing and integrating in a same profile these heterogeneous data is currently difficult, particularly when learners profiles that have a rich content, a complicated structure and are different from each other. The platform-independent Profiles Modeling Language (PMDL) we propose aims to solve this problem. Based on a large state of the art, we use a classification of learners profiles data in five main categories to define PMDL. This language allows rewriting various learners profiles created by any systems, human or not. Existing profiles will now be able to be reused in transformations or exploitations purposes. In this paper, we illustrate the rewriting principle with an example of an existing learner profile.

First results of PMDL evaluations are promising for language expressive power point of view. From an operational point of view, we have implemented this language within a profile management system named EPROFILEA (Jean-Daubias et al. 2009). This implementation shows the feasibility of our approach as it allows the reuse of various heterogeneous existing profiles for profiles visualization and personalization of learning activities purposes. Experimentations prove that teachers can use the main concepts of the operationalized language in order to represent, combine and exploit effective data, digital or not, constituting learner profiles. The BNF formal notation used for the specifications of PMDL makes it usable in various contexts by different software. A potential use of this language is to format profiles as soon as they are created. Indeed, PMDL could also be used as a first proposition for future works on profiles standardization. But it should be extended to take into account evolutivity of profiles (Ginon et al. 2011)

References

- Bull S., McEvoy A.T. & Reid E. (2003). *Learner Models to Promote Reflection in Combined Desktop PC / Mobile Intelligent Learning Environments*. In workshop LeMoRe, AIED'03, p. 199-208, Australia.
- Celik I., Ashman H., Blanchfield P. & Brailsford T. (2008). *An Interoperability System for User Models in Adaptive Educational Hypermedia*. In workshop User Model Integration, Adaptive Hypermedia and Adaptive Web-Based Systems 2008, p. 33-37, Germany.
- Conlan, O., Wade, V., Bruen, C. & Gargan M. (2002). *Multi-model, metadata driven approach to adaptive hypermedia services for personalized e-learning*. In Adaptive Hypermedia and Adaptive Web-Based Systems 2002, p 100-111, Spain.
- Eyssautier-Bavay C. (2008). *Modèles, langage et outils pour la réutilisation de profils d'apprenants*. PhD of Joseph Fourier University, France.
- Ginon, B., Jean-Daubias, S. & Lefevre, M. (2011). *Evolutive learners profiles*. In Ed-Media 2011, Lisbon, Portugal.
- Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M. & Von Wilamowitz-Moellendorff, M. (2005). *GUMO - the general user model ontology*. User Modeling 2005, p. 428-432, UK.
- Jean-Daubias S., Eyssautier-Bavay C. & Pernin J.-P. (2009). *A model of learners profiles management process*. In AIED'09, p. 265-272, UK.
- Rahm, E. & Bernstein, P. A. (2001). *A survey of approaches to automatic schema matching*. The International Journal on Very Large Data Bases, Vol 10, Issue 4, p 334 – 350.
- Ramandalahy M. T., Vidal P. & Broisin J. (2009). *An abstract modeling of learning environments to ensure tracking of learners*. In AIED'09, UK.
- Rueda U., Larrañaga M., Arruarte A. & Elorriaga J.A. (2006). *DynMap+: A Concept Mapping Approach to Visualize Group Student Models*, In EC-TEL'06, W. Nejdl & K. Tochtermann (Eds), p. 383-397, Crete, Greece.
- Zapata-Rivera J.D., Greer J. (2004). *Interacting with Inspectable Bayesian Student Models*. In IJAIED, Vol. 14, p. 1-37.