

# Content Personalization for Inclusive Education through Model-Driven Engineering

Christopher Power and Richard Paige

Department of Computer Science  
University of York  
York, UK, YO10 5DD  
{cpower, paige}@cs.york.ac.uk

**Abstract.** Content personalization of e-learning resources has the opportunity to encourage self-directed learning and collaborative activities between students with varying cultures and backgrounds. In the case of students with disabilities, it also has the potential to provide equality of access to learning resources that can be presented in formats that are compatible with a student's needs and preferences. In this paper, a framework is presented for doing this type of content personalization for students with disabilities using Model-Driven Engineering tools and techniques.

## 1 Introduction

Given the increased availability of education through online and blended learning environments, it is important to provide online media that students with disabilities can access in a way that takes into account their personal preferences. These preferences can include the technology that the student wishes to use to access the information, such as a Braille display or screen reader, or it could include media enhancement preferences such as captioning of audio/video for people with hearing disabilities or highlighting of text for people with specific learning difficulties. This matching of user preferences with attributes of the content can be achieved through an adaptive system that personalizes the content for each individual user.

This paper describes how this type of content personalization of digital media can be achieved through Model-Driven Engineering (MDE) languages, tools and techniques. In particular, it will demonstrate how *model comparison* (sometimes called model matching) can be used to identify and produce configurations of the digital resources that have attributes that match complex personalization requirements of an individual user. With model comparison facilities becoming available through MDE tools such as the AMMA platform and the Epsilon Model Management framework, this technique has the potential to ease the implementation and management of personalization platforms.

The paper begins with a discussion of the types of personalization that have been previously investigated in terms of adaptive systems that match users with specific types of content on the web. Following this, Model-Driven Engineering and its tools

and techniques are presented as a general framework for performing attribute and structure comparisons between different abstract models of real-world objects.

With this foundation in place, the paper presents two models, one representing user characteristics and one representing the characteristics of digital media found in e-learning environments. A sample of comparison rules for matching content with user preferences is presented. The paper closes with a discussion of future work that is planned in this area.

## 2 Personalization through Adaptive Systems

There have been a large number of initiatives to provide *adaptive systems*, be they online or traditional desktop applications, personalize aspects of the interface or other media elements based on data about the user. In many cases, a *user model* that represents characteristics about the user is used for purposes of matching features in the interface or content to the user. Examples types of personalization that have been investigated with this type of adaptive system include:

- Security and privacy settings [6,10]
- Multimedia and hypermedia content [3]
- Web search results [11]
- Internet and mobile television content [4]
- E-learning and blended learning materials [2,9]

While this list is not exhaustive, it is this last application of adaptive systems that is of interest in this paper.

Content personalization of online learning materials for students holds great promise for encouraging self-directed, collaborative learning between students with different backgrounds, cultures and abilities. In the case of students with disabilities, online education resources represent an opportunity for equality in education. With the ability to have digital resources either transformed or substituted when a student needs them, the needs and preferences of each individual student can be addressed. This type of content personalization can provide an opportunity for students with disabilities to more fully participate in online learning at all levels of education, while providing them with the opportunity to interact and collaborate with their mainstream peers over the web.

However, this idealized view of content personalization for students with disabilities in education is being disrupted by the large variety of user models and content models that are being developed by standards and guideline bodies. Currently there are a large number of initiatives to standardize the type of data that can be used to classify and categorize media; examples of these models include the upcoming ISO Access For All standard and the IMS Accessible Learner Information Profile / IMS Content Packaging [7]. In each of these models the learning materials are represented by metadata that describes their contents and the presentation options available. In the case of the IMS guidelines, the users of the media and their devices are also described in separate information models.

However, as these and other information models emerge in the marketplace there are two key problems that present themselves. Firstly, as education systems worldwide adopt different standards, how can data from one information model be transformed into another so that materials can be easily shared around the world? Secondly, how can matching between two information models be facilitated such that the appropriate content is delivered to the user, based on personalization preferences [1, 5]? It is this second problem that is addressed in this paper.

In current practice, these problems are usually addressed through bespoke solutions developed by an educational institution. This can result in poor interoperability between systems and can be difficult to alter as the information models evolve over time. In place of these bespoke solutions, Model-Driven Engineering tools and techniques can be used to provide a seamless, unified transformation and matching infrastructure for media personalization.

### 3 Model-Driven Engineering

Model-Driven Engineering (MDE) is a principled approach to system engineering that is founded on the use of abstract descriptions (models) throughout development. A model may be a description of a software artifact, a set of user requirements, an architecture, a set of interactions between users and a system, etc. What constitutes a valid model is defined in terms of a *metamodel*. A metamodel of a description language is typically used to capture the concepts and structure of that language. Metamodels can then be used as the basis for manipulating models.

MDE supports a style of development where models are constructed at the start, and are successively manipulated throughout the engineering process. A particularly important form of manipulation is *model transformation*, wherein a model (e.g., expressed in the Unified Modeling Language (UML)) is transformed into a new model expressed in a different language. Model transformation is a special operation in MDE; there are many other operations that can be applied to models, including operations to combine models, compare models, generate code from models, and validate models. Overall, *model management* is the discipline of manipulating models via precisely defined, tool-supported operations such as these.

Model comparison involves identifying *matching elements* between two or more models. For example, we might have two versions of the same model (e.g. expressed in UML). We may want to know where the differences between the two models arise so that we can reconcile those differences when combining the two versions into a single model. Several approaches to model comparison and matching have been proposed in the literature, including approaches based on model identifiers, signature-based approaches, and similarity based approaches that treat models as attributed graphs and compare elements based on the similarity of their features. Most existing approaches have limitations in terms of exploiting the semantics of the models being compared; moreover, most existing approaches can only be applied to homogeneous models, i.e., models expressed in the same language.

An approach that does not have these limitations was proposed in [8]: the Epsilon Comparison Language (ECL). ECL provides a language specifically tailored for

describing *how* models (possibly in different languages) should be compared. Using the language, programmers can specify a set of *comparison rules* that describe conditions under which model elements are considered to match. The result of executing these rules (using the Epsilon framework available at [5]) is to produce a *match trace*, which is an encoding of all matching elements in the models being compared. The match trace can then be serialized and post-processed in arbitrary ways. For example, the trace could be visualized, or could be passed to another program that uses the trace to merge models. See [8] for further examples.

We can apply model comparison, and ECL, to determine matches between user preferences and available digital media.

## 4 Content Personalization through MDE

In order to perform comparisons between user preferences and digital media, there are three components that are required:

- A metamodel describing the preferences of a user in terms of the types of media that he/she would like to have presented to them.
- A metamodel describing the attributes of different types of digital media.
- A set of comparison rules for models that conform to the above metamodels. In general, comparison rules can be expressed in many different ways; we use ECL to accomplish this, as it allows us to compare models in different languages, conforming to different metamodels.

In Figure 1, a metamodel for user preferences is presented. This metamodel has been derived from the results of requirements elicitation activities, including surveys, interviews and focus groups, conducted across Europe<sup>1</sup>. The preferences are organized by type of media, such as video or graphic media, with each subclass represents a set of preferences for one type of digital media that is available in e-learning environments. For example, in the case of video a user may wish to have an audio description accompany the video.

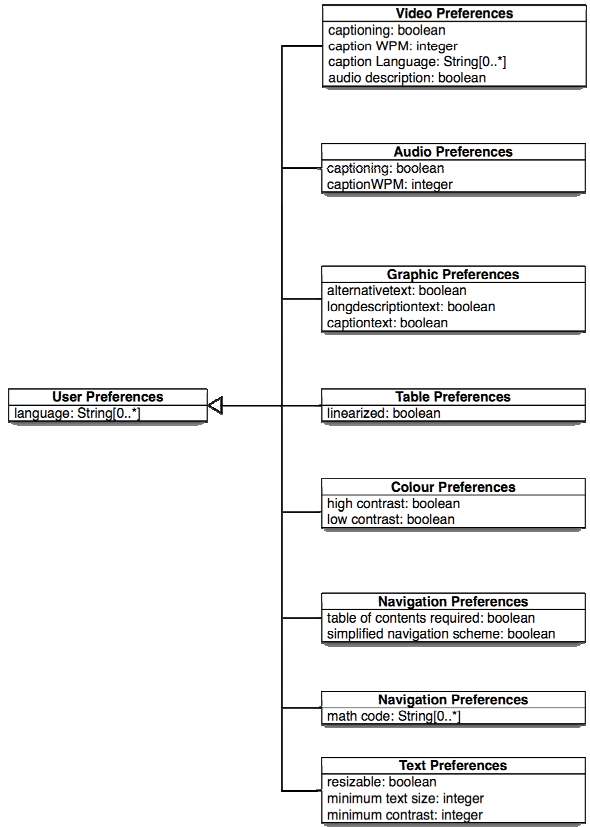
For any given user, a model of their preferences can be created for use in comparisons to the attributes of available media.

In Figure 2 a metamodel describing different types of media is presented. This classes of media types is not exhaustive, however they are the most common types of media available in current e-learning environments<sup>2</sup>. This metamodel is structured such that one type of digital media may be an aggregate of many other pieces of digital media, such as a web page having multiple graphics and videos embedded in it. Further, a piece of digital media may have an adapted version, which may have had its attributes altered in some way to make it accessible to a person with a disability. Again drawing from the example of a video, a video may originally come without any enhancements, but an adapted version of that video would have text captions or an audio description added.

---

<sup>1</sup> These elicitation activities were conducted under the auspices of the EU4ALL project, information about which can be found at [www.eu4all-project.eu](http://www.eu4all-project.eu)

<sup>2</sup> In general, these are the most common types of media on the web.



**Fig. 1.** A metamodel, named *Preferences* representing the preferences of a user for formatting of different pieces of digital media

Finally, a set of two comparison rules expressed in ECL is provided as an example. Assuming the metamodels above, the following code fragment, labeled *Rule 1*, represents one of the most basic, abstract comparison rules that can be defined. It takes a single attribute from one entity in each of the metamodels and compares them for equality.

```

% Rule 1
rule DigitalMediaWithUserPreference
  match dm: Media!DigitalMedia
  with up: Preferences!UserPreferences {
    compare: dm.language = up.language
  }

```

In this comparison rule, the *Digital Media* class of the *Media* metamodel, represented by *dm*, has its *language* attribute, representing the language of presentation (e.g. English, Spanish, German), compared to the *language* attribute of the *User Preferences* class, designated *up*, of the *Preferences* metamodel. If this comparison evaluates to *true* for a pair of model elements, these elements are said to

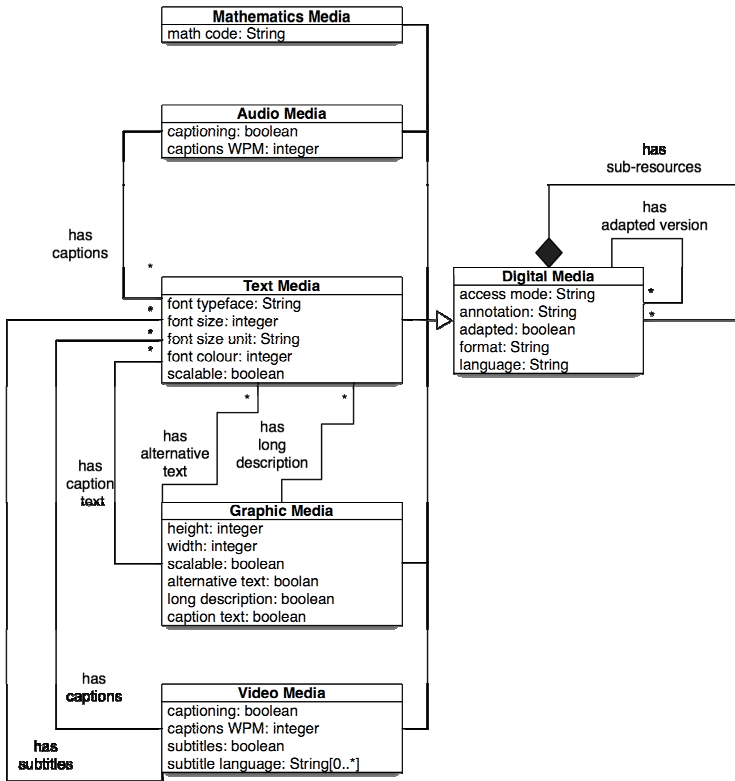


Fig. 2. A metamodel, *Media*, describing types of digital media available online

*match*, and they are recorded and distinguished from other elements that do not match (the technical details of how these elements are recorded is detailed in [8]). In general, comparison rules can be arbitrarily complicated Boolean-valued expressions, which may include quantified expressions over compound data structures like sets.

In the following example, labeled *Rule 2*, a user's preferences for captioning, the speed of the captions in words per minute, audio description and caption language are all compared to a description of a piece of media.

```

%Rule 2
rule VideoMediaWithVideoPreferences {
  match vm: Media!VideoMedia
  with vp: Preferences!VideoPreferences
  extends DigitalMediaWithUserPreference {

    compare:  vm.captioning = vp.captioning and
              vm.captionsWPM = vp.captionsWPM and
              vm.subtitles = vp.audio_description and
              vm.subtitle_language = vp.caption_language

  }
}
  
```

This rule represents a more complex set of attribute comparisons between a model of type *Media* and a model of type *Preferences*. In this case, attributes contained within the element *VideoMedia*, (described in the *Media* metamodel) are compared to the *VideoPreferences* class (defined in the *Preferences* metamodel). Of note in Rule 2 is that this rule *extends* the previous rule; ECL supports rule inheritance and extension, thus enabling reuse of rules. The meaning of this construct is similar to that in object-oriented programming: Rule 2 first executes the compare part of Rule 1, then executes its additional comparison expressions.

These rules, and others like them, can be applied to compare a user to a set of online resources. When the comparison is complete, a set of online resources that meet their user's preferences will be contained in the trace. These resources could then be displayed through a variety of online environments (e.g. virtual learning environments, websites).

## 5 Discussion

This preliminary work lays the foundation for a broader solution for performing model-based comparisons that will provide personalized media presentation to users.

The presented models and comparison rules are each subsets of a much larger system of comparison components that will be used to provide match user preferences, device characteristics and contextual information about the use of media, with a library of digital media used in e-learning courses. This work will not only do the comparatively simple model matching described in this paper, but will also examine the possibility of using model merging techniques to combine different versions digital resources, with different characteristics, to produce new accessible versions of learning materials where none currently exist.

## References

1. Baldiris, S., Santos, O.C., Baldiris, S., Barrera, C., Boticario, J.G., Velez, J., Fabregat, R.: Integration of educational specifications and standards to support adaptive learning scenarios in ADAPTAPlan. *International Journal of Computer Science and Applications (IJCSA)*. Special Issue on New Trends on AI techniques for Educational Technologies 5, 1 (2008)
2. Boticario, J.G., Gaudioso, E.: Towards personalized distance learning on the web. In: Mira, J. (ed.) *IWANN 1999*. LNCS, vol. 1607, pp. 740–749. Springer, Heidelberg (1999)
3. Bunt, A., Carenini, G., Conati, C.: Adaptive Content Presentation for the Web. In: Brusilovsky, P., Kobsa, A., Nejd, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 409–432. Springer, Heidelberg (2007)
4. Chorianopoulos, K.: Personalized and mobile digital TV applications. *Multimedia Tools and Applications* 36(1), 1–10 (2008)
5. Epsilon project web site (2009), <http://www.eclipse.org/gmt/epsilon>
6. Fink, J., Kobsa, A., Schreck, J.: Personalized hypermedia information provision through adaptive and adaptable system features: User modelling, privacy and security issues. In: *IS&N 1997: Proceedings of the Fourth International Conference on Intelligence and Services in Networks*, London, UK, pp. 459–467. Springer, Heidelberg (1997)

7. IMS Global Learning Consortium, IMS AccessForAll Metadata Specification, <http://www.imsglobal.org/accessibility/> (retrieved, 10/2008)
8. Kolovos, D.S., Paige, R.F., Polack, F.A.C.: Model Comparison: a Foundation for Model Composition and Model Transformation Testing. In: Proc. First International Workshop on Global Integrated Model Management (G@MMA) 2006, co-located with ICSE 2006, Shanghai, China, (May 2006)
9. Khribi, M., Jemni, M., Nasraoui, O.: Toward a Hybrid Recommender System for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. In: Richards, G. (ed.) Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pp. 6136–6145. AACE, Chesapeake (2007)
10. Kobsa, A.: Privacy-enhanced personalization. *Commun. ACM* 50(8), 24–33 (2007)
11. Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In: *CIKM 2007: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 525–534. ACM, New York (2007)
12. Velasco, C.A., Mohamad, Y., Gilman, A.S., Viorres, N., Vlachogiannis, E., Arnellos, A., Darzentas, J.S.: Universal access to information services—the need for user information and its relationship to device profiles. In: *Universal Access in the Information Society*. Springer, Heidelberg (2003)