Abstract: The aim of this paper is presenting an AContent extension called BEAT (Bologna E-learning Authoring Tool) which has been designed and developed to support the use of templates during authoring and editing of e-learning content. Such templates have been added to AContent without the need of integrating a template engine, by using existing data structures and e-learning standards. Three levels of templates have been introduced to support (i) the control of graphical aspects, (ii) the structure of each single page, and, (iii) the structure of the whole learning content.

Introduction

The production of effective e-learning materials clearly relies on the availability of tools that support authors both in designing and building content. The choice of a specific authoring tool has a significant impact on the final quality of e-learning products (Di Iorio et al. 2006) and this is particularly true whenever it is used by large organizations to produce content on a wide scale. Main e-learning authoring tools provide support to (Guralnick & Levy 2009):

- compliance to e-learning formats and standards, to ensure portability through different platforms/systems;
- creation of new content as well as editing of existing ones, without requiring specific technological skills (Di Iorio et al. 2008);
- reduce the learning curve to learn how to use the tool.

Just few of e-learning authoring tools encourage and support good design, both in terms of educational aspects and technological issues. This feature is mainly addressed by the use of templates or specific frameworks provided to the author through wizard or other interactive GUIs. Usually, Web templates are run-time applied by means of a template engine embedded in the platform that delivers and manages content. Differently from this approach, templates for e-learning have to be compliant with the well-known mechanism of packetized standard content, that are self-contained and platform independent. Consequently, e-learning templates cannot rely on specific engines provided by LCMS or repositories and are used just during the design and the editing phases to create the standard packetized learning content (Hui & Liu 2005).

Main advantages obtained by introducing templates in e-learning authoring tools are:

- the support of authors in designing (by suggesting predefined paths to follow) and creating (by providing examples of content) an e-learning content. Templates augment the usability of authoring tools and provide new, more effective ways to produce learning content.
- The availability of a set of rules to be followed. This is generally useful but is actually necessary whenever content are produced on a large scale and some qualitative/quantitative standards have to be met by all authors. Again, simple characteristics can be in charge of the graphical aspects of content, others, more
complex, to didactical proprieties of the content itself (e.g. methodology, goals description, assessment).

We are currently working on many activities that require different sets of given standards and this can be done by providing authors with several templates (Aust & Meyen 2005).

Different existing tools have been analyzed, including:
- Xerte (http://www.nottingham.ac.uk/xerte/);
- EXeleARNING (http://exelearning.org/);
- MyUDUTU (http://www.myudutu.com/);
- HTMLArea (http://moodle.org/plugins/view.php?id=9);

We finally decided to work on AContent (http://tutorial.ca/acontent/), an open source learning content authoring system used to create interoperable, accessible, adaptive Web-based learning content. This choice has been driven by different factors:
- it is open source;
- it produces content in conformance to main e-learning standards;
- it is accessible and it supports the creation of accessible content.

The remainder of this paper is organized as follows: section Design issues synthesizes the system design while section Layers of template presents the three levels of templates managed by BEAT. Section On designing lesson structure describes the supported design proposed in BEAT with the aim to drive the author during the creation process. The paper ends with a section devoted to present conclusions and future work.

Design issues

The whole design of the BEAT extension has been driven by the need of defining some sorts of standardization assist to ensure technical and didactical quality in creating e-learning content. Templates are contemporaneously a support to a flexible and easy content creation and a tool to provide authors with suggestions/methodologies in designing and producing e-learning content.

The idea of using a problematic didactic model in e-learning training paths echoes the topic of pedagogical problematicism and defines a complex hypothesis which can emphasize the integrated coexistence of different didactic strategies referring back to a problematicist matrix (Bertin 1968, Guerra 2010). The theoretical foundation of this model explicitly recalls the main learning theories and their critical interaction.

The starting point of the models described here is the possibility to define three main learning paths, focusing respectively on the object, the process and the subject of learning (Guerra et al. 2008).

These models aim to support authors for didactical aspects in creating the e-learning content using the BEAT authoring tool. We proposed a structure template providing a predefined organization of the lesson as for goals, content, assessments, tools, references, etc. This approach can support the authors to create their content paying attention to pedagogical and didactical aspects but, at the same time, it is not binding and allows the author to move freely in the methodological choices.

In order to address both these issues, we worked by coupling template philosophy with different mechanisms:

A. three layers templates architecture that is top-down organized as follows:
- each lesson can be created on the basis of a structure template that provides a predefined organization of the lesson as for goals, content, assessments, tools, references and so on;
- each page of the structure is associated to a page template, that schematizes the page content by splitting it into sub-parts. Page templates have been designed on the basis of the well-known “slide layout” used by main presentation programs, i.e. Open Office Presentation editor (http://www.openoffice.org/);
- each page content is associated to a page layout for the graphic aspects of the page.

Section Layers of template focuses on the template structure by introducing it with a bottom-up view.

B. Editable/uneditable elements. The above mentioned templates are coupled with specific ways/tools to edit them. We have considered:
- editable elements: for example, each page template is editable, the page structure can be associated to a page template and then changed. In this case, the main goal of the template is giving support to
authors, without introducing any constrains in their design.

- Uneditable elements: for example some items in a lesson structure can be mandatory. This means the author has to fill in a specific page (e.g. to define the lesson goals) and cannot remove it from the structure. In this case the template is used to constrain to authors, by forcing their design to a specific didactical standard. Similar examples can be done to force some technical standards (e.g. imposing an alternative content for accessibility related purposes). Other constrains can be expressed in terms of minimum and maximum number of items in a specific container, e.g. to be sure there will be from 2 to 5 final assessments.

C. Wizards: we are currently working on designing and implementing wizards to support an easy step-by-step creation of content. This aspect has been considered from the beginning in BEAT design, but it is not completely addressed and will be the object of future work.

D. Authoring permissions: to modulate authoring capabilities on editable/uneditable items, we introduced some editing permissions in the author profile which is provided by AContent.

Layers of Template

To support authors during content design and creation, BEAT provides three different levels of templates, which can be bottom-up listed as follows.

A. Layout template: to select and apply a graphic design to display the content.
B. Page template: to support the author in composing pages (inspired by slide models).
C. Structure template: to provide the author with some predefined learning object structures (with/without assessment, overview, goals, etc.).

The remainder of this section is devoted to present each type of templates and to introduce some implementation issues.

Layout Template

Layout templates are designed to select and apply graphic issues to a page. They are basically implemented as CSS to be added to the page and they use the cascading applying philosophy to merge with the LCMS graphic design. Layout templates can be applied to (i) a whole content (so as to define a general graphical standard for it) or to (ii) a single page. Figure 1 shows the Layout templates panel. Authors can choose the layout from a list of available graphic designs and then decide to apply it to the whole lesson (as AContent calls an e-learning content) or to the single edited page.

![Layout templates panel](image)

These two options relate to two different authoring requirements:

1. the need to give a consistent graphic design to a whole content. Figure 2 and 3 show respectively a page without any layout template and the same page with a specific layout template applied to it;
2. the possibility to show a specific page with some definite graphic details, such as different way to underline some parts of the content (i.e. with or without a border) or different layout settings (i.e. images alignment). Figure 4 and 5 show the same page with a different alignment for images.
It’s worth noting that the content editor provided by AContent permits the author to locally modify some layout aspects without the need of applying any layout template. In this case templates are provided to support the creation of a consistent layout, but they are not limiting the author’s creativity in any way.

As hinted above, from an implementation point of view, layout templates are created as local CSSs. Such style sheets are displayed and used as layout templates by BEAT if they are stored in a specific directory created by our extension of AContent. Adding or deleting a layout template is prerogative of the AContent administrator. After the content is exported from AContent as a packetized standard content, the CSS can be associated to the single page (in case the layout template is applied to the page) or to the whole packet (in case the layout template is applied to the whole lesson). Let us note that in both cases this strategy ensures that lecture layout does not strictly depend upon the platform which is finally devoted to provide it to users. This augments the content portability, which could be partially compromised by a wrong integration with the LCMS layout.

Page Template

Page templates are used to define the page structure. They have been designed on the basis of what the main slide creation office automation tools defines as “slide template”. A similar mechanism is also used by MS LCDS, in order to provide predefined page structures. The author selects the page structure from a set of available ones and, starting from it, can edit the structure by adding and removing subparts. Page structure is created by choosing the appropriate template from the page template selection panel (shown in Fig. 6).

Also in this case the content editor provided by AContent permits to the author to locally modify the page structure by removing parts of it or by adding new content. The author can, for example, create a page structure with an image and a text and remove the text to add an audio file. In this case the template mechanism is designed to support to the easy creation of content and, again, it does not limit the author creativity in any way.

Each page template is substantially an HTML fragment (to be included in a <body> element), normally structured in conformance with HTML standards. Similarly to layout templates, these HTML fragments are displayed and used as page templates by BEAT if they are stored in a specific directory created by our extension of AContent. Again, adding or deleting a page template is prerogative of the AContent administrator.
Finally, the last level of templates introduced by BEAT is structure templates. This mechanism has been inspired by some of the analyzed authoring tools, including MyUDUTU and MS LCDS. By using different mechanisms these applications provide authors with a way to create an empty standardized lesson starting from a set of predefined models.

In BEAT we offer the author to start the creation of a new lesson by using a structure template. Templates are organized as hierarchies of pages, each one associated to a page template. Each page of a given structure can be mandatory or not. Mandatory pages cannot be removed from the lesson.

Figure 7 presents an example of structure template. The lesson is organized in the following sub parts:

1. a mandatory page titled “Goals”, associated to the “list” page template;
2. an optional page titled “Overview”, associated to the “text” page template;
3. a substructure titled “Content” composed by:
   a. a mandatory page titled “Content part 1”, associated to a “text” template;
   b. a facultative page titled “Content part 2”, associated to a “MM” template;
4. a mandatory page titled “Assessment”, associated to the “multiple choices” page template;
5. an optional page titled “References”, associated to the “list” page template.

Each template can be shown as empty or filled with instruction to the author to support the editing of any lesson. Mandatory elements (in Fig. 7 Goals, Content part 1, Assessment) cannot be removed by the author so they will be part of the lesson. Facultative elements can be removed. New elements can be added to the template by authors. Figure 8 presents the content substructure after having added page titled “Content part 3” and associated to the “text” page template.

The structure templates are based on the manifest used to define the lesson organization, the same way the layout templates are based on CSS and page templates on HTML.

A prototype of BEAT is on line here: http://137.204.74.112/AContent_BEAT/.
On Designing the Lesson Structure

In addition to the three layers of template, BEAT provides a supported design procedure to drive the author during the creation process.

To guide the author in the choice of the structure template, we have associated some verbs for each learning model. These verbs refers to Bloom’s Taxonomy (Bloom et al. 1956, Churches 2009) and here are used to define a lesson learning goals and are displayed in the lesson property page when a new lesson is created (as shown in Fig. 9). According to the author's selection, the system composes a lesson with one or more structure templates.

![Figure 9: List of verbs associated at the structures template](image)

These structures are based on the following learning models (Guerra 2006): knowledge, meta-competency and creative based.

As we can see in the Table 1, the knowledge based model aims to promote self-learning in order to acquire knowledge base about topics that have been treated.

<table>
<thead>
<tr>
<th>Learning objectives</th>
<th>The learning objectives in this model are mainly specific, precise, explicit and so predefined, by referring to the reproductive type. The involved cognitive processes are: remembering, classifying, listing, executing and retrieving.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources and content</td>
<td>They can refer to a specific part of a discipline or a process/competence facilitated by example or hands-on exercises.</td>
</tr>
<tr>
<td>Activities</td>
<td>Didactical activities are linear and sequential i.e. <em>listen and repeat, apply the procedure, exercise the competence</em>.</td>
</tr>
<tr>
<td>CMC tools</td>
<td>Examples: e-mail, to support each user during the understanding process; <em>video conferencing</em> and <em>virtual classroom</em> to involve as many students; <em>repository</em> to facilitate users in sharing, searching and updating learning resources and content.</td>
</tr>
<tr>
<td>Teacher’s role</td>
<td>Mainly instructor oriented. He/she organizes <em>knowledge to be transmitted</em>, related to specific competences, in a sensible and linear way.</td>
</tr>
<tr>
<td>Learning retrieval</td>
<td>The teacher could propose learning retrieval together with other learning content and resources, also by means of different media. Learning retrieval activities are addressed to single users, couple of users or small group.</td>
</tr>
<tr>
<td>Learning time</td>
<td>It is predefined for all phases of learning activities.</td>
</tr>
<tr>
<td>Monitoring and evaluation</td>
<td>Monitoring and evaluation are supported by close-ended questions and structured questions; they are mainly individual. Most commonly tests used are: <em>true or false, multiple choice, cloze, matching question, etc.</em></td>
</tr>
</tbody>
</table>

Table 1: Knowledge based model

The meta-competency based model emphasizes the logic of a constructivist approach to knowledge construction. As synthesized in Table 2, it moves from the perspective of metacognition and development of knowledge and skills which encourages the systematic use of investigation tools (attitudes, methods, techniques) both at individual and at group level. This supports the conceptualization, the generalization, the transferability of knowledge artefacts and methods used to produce them.
<table>
<thead>
<tr>
<th>Learning objectives</th>
<th>General and specific, but not completely defined. The main involved cognitive processes are related to analysis, synthesis and evaluation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources and content</td>
<td>The resources and content are designed to promote learners' autonomy in broadening their knowledge and their troubleshooting abilities. The resources could be: hypermedia, text, exercises and open problems and tools, i.e. research algorithms, best practices, repertoires, etc.</td>
</tr>
<tr>
<td>Activities</td>
<td>Case study, problem solving, simulation, etc.</td>
</tr>
<tr>
<td>CMC tools</td>
<td>Examples: <em>forum</em>, to enhance the dialogue among users, starting discussion, carrying on specific or general topics; <em>wiki</em>, to promote cooperative (asynchronous) or collaborative (synchronous) writing sessions among participants; <em>blog</em>, to support the user during the reflection process; <em>chat</em>, to support or promote the simultaneous interaction among users; <em>repository</em>, to enhance sharing processes among users, while collecting learning artefacts.</td>
</tr>
<tr>
<td>Teacher’s role</td>
<td><em>Facilitator</em>, providing training and motivational support to learners. <em>Testimonial</em>, offering a model to follow, staking him/herself and collaborating with the group.</td>
</tr>
<tr>
<td>Learning retrieval</td>
<td>The learner (which is assumed to have experienced difficulties in socio-emotional dimension) could repurposed the same learning materials, but trying to boost his/herself esteem and/or changing the social context (i.e. study in small group).</td>
</tr>
<tr>
<td>Learning time</td>
<td>It is defined just in terms of guidance; it depends on the result carried out by each single user during the formative evaluation.</td>
</tr>
<tr>
<td>Monitoring and evaluation</td>
<td>Formative and summative evaluations with the support of semi-structured survey addressed to individuals and groups. Close-ended questions to reach reproductive learning objectives; open-ended questions to evaluate the learners’ commitment, creativity, etc.</td>
</tr>
</tbody>
</table>

**Table 2: Meta-competences based model**

The *creative based model* is focused on learner’s motivations and “emotions”. As reported in Table 3, this type of model pursues the activation of skills not easily measurable with docimological procedures hardly definable and strongly associated with the “sphere of individuality”. An example of these competences is the willingness to take different points of view, activating forms of so-called creative thinking.

<table>
<thead>
<tr>
<th>Learning objectives</th>
<th>Examples are: analysis (analyze, compare, lead), synthesis (synthesize, schematize, infer), intuition (groped solutions, formulate hypotheses), invention.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources and content</td>
<td>The learning materials are both theoretical and applicative, also linked to professional practice materials.</td>
</tr>
<tr>
<td>Activities</td>
<td>Strategies, even provocative, stimulating the students' reflections that go beyond the objective level of competence and skill. Examples are: case study and problem solving, role-play and simulation, cooperative and collaborative learning, etc.</td>
</tr>
<tr>
<td>CMC tools</td>
<td>Examples: <em>social networking</em>, to promote formal and informal learning processes; <em>blog</em>, to enhance the personal reflection and the discussion among learners; <em>chat</em>, to promote the simultaneous interaction about discussion topics, without teacher’s mediation.</td>
</tr>
<tr>
<td>Teacher’s role</td>
<td><em>Facilitator</em>, coordinator, consultant, witness.</td>
</tr>
<tr>
<td>Learning retrieval</td>
<td>Peer to peer, in case learners experience difficulties.</td>
</tr>
<tr>
<td>Learning time</td>
<td>It is not defined: it is individualized for each student.</td>
</tr>
<tr>
<td>Monitoring and evaluation</td>
<td>Both individual and group self-assessment by means of open-ended question.</td>
</tr>
</tbody>
</table>

**Table 3: Creative based model**
Conclusions and Future Work

The aim of this work is to present BEAT, an AContent extension designed and implemented to provide an effective support to didactic design of e-learning content.

The paper underlines the strong relationship between technical and pedagogical dimensions which will methodologically support the author during the development of learning resources.

Authors using BEAT can easily select a learning model, from a set of predefined ones that is currently based on three well-known models but can be extended or reduced by configuring the system. Each model is associated to a lesson structure, composed by pages based on predefined page templates. In addition, a layout template is provided to authors with the aim to support an easy and effective graphic design of the content. We are now going to test the tool in a large scale e-learning use case, to verify its effectiveness. Finally, we are currently working together with the IDI (Inclusive Design Institute, http://inclusivedesign.ca/) staff to integrate a part of BEAT functionalities into the AContent distribution.

References


Acknowledgements

The authors wish to thank Mauro Donadio, Francesca Trombetti, Matteo Boni, Sara Cenni, Michela Moretti, Andrea Reggiani, Ivan Traina, Elena Pacetti, Manuela Fabbri and Klara Ulikova for their precious support.