Conversion of the YDP Learning Content to Common Cartridge Package

Jaroslaw Dziedzic

Young Digital Planet S.A. Educational ePublishing, Content and Technology,
Ul. Slowackiego 175, 80-298 Gdansk, Poland
jaroslaw.dziedzic@ydp.com.pl
http://www.ydp.eu

Abstract. This paper describes experiences earned during the development of a software application that could convert some YDP specific learning content into Common Cartridge.

1 Introduction

Young Digital Planet has almost twenty years of experience in digital learning content creation. We have created several generations of applications and content. The experience and ideas have accumulated over years, resulting in broad range of top-quality educational content, advanced applications and tools, as well as effective and flexible data formats for storing and processing the digital content.

We observe emerging standards in the field of digital learning content data formats with interest, and we keep evaluating them and assessing their usefulness for the packaging and distribution of our content. The ASPECT project gives us the opportunity to take a look at the new Common Cartridge emerging standard, analyse it thoroughly and decide, if we can implement it in our products, and how to do it.

It would not be possible to reach a deep understanding of any new concept just by reading the documentation. Therefore, in order to gain some knowledge and experience, we decided to produce a tool that will try to convert our existing content to the CC format. Our goal was to have our content with all, or most of its features, packed into the Common Cartridge and running in the CC player.

2 YDP Content Concept vs. Common Cartridge Concept

Right from the start of reading the CC specification it became obvious, that the CC concept differs in many ways from our existing content. These differences could break the project just at the beginning, but fortunately we found some methods of changing our content to suit the Common Cartridge way. Let's look first at the key concepts for YDP content and CC.

2.1 YDP Content

Our content is usually organised in lessons that consist of several pages. Each page contains some educational content (text, pictures, slideshows, video, sound clips, interactive animations etc.) and exercises related to this content. The student learns not only by reading/viewing the static or interactive content, but also by solving exercises and observing the feedback from the computer. The process is interactive, exercises are tightly integrated with the learning content and it is impossible to separate them from the material presented on the page. The scoring of all the exercises in the lesson is usually presented and summed up on the last page of the lesson, and the student has the possibility to return back to any exercise, read the lesson again, sort out his answers and see the result.

We use several different kinds of interactive exercises in addition to typical multiple choice and fill-in-blanks. Pages can have sub-pages (using tab-sheet controls), buttons that launch some actions or just open popup pages with additional content etc. Everything is laid out by our graphic designers to form harmonious and attractive environment for learning content and exercises.

Our entire content is presented to the student by means of a dedicated player written in Adobe Flash. We have no concept of static content displayed outside of the player – for instance as a PDF or HTML page. Everything is played by our flash application. The player must always accompany the XML and data files.

2.2 Common Cartridge

As we understand from the specification, the Common Cartridge is based on the concept of the complete separation of a learning content and exercises. The only reason the exercises are included – is to test the student's knowledge, acquired while reading/viewing the static resources. Exercises cannot be mixed with the learning content and do not participate in student's interactive learning experience. They simply serve as an assessment tool or they form a pool of exercise resources (Object Banks) aimed for a teacher, who wants to use them in his custom-made lessons and examinations.

While there is really great freedom about the way the static content can be built (HTML, PDF, flash objects etc.), this is not the case with interactive exercises. They can be described only by means of the old QTI 1.2.1 data format (constrained even further by the CC profile), and played back by the player integrated with the server software. There is no place for the dedicated player here, so the content producer has really limited possibilities with regard to choice of exercises and the way they are presented to the student. In fact – even though the CC specification claims that it can handle 6 exercise types – there are just two distinct exercise types to choose: multiple choice (in 3 simple variants) and fill-in-blanks. And even these two types have important limitations that allow for their use in really simple cases only.

On the other side – the lack of a dedicated player can be an advantage for some content producers that are interested in simplicity. Creating only the authoring software for QTI 1.2.1 could be – in theory – simpler than creating both authoring software and a dedicated player. It would be true – if the QTI 1.2.1 was straightforward and easy to implement. Unfortunately – it is not the case, and I will explain it in details later in this document.

One advantage of a CC approach without the dedicated player can be easily seen. The CC package doesn't have to care about communication with the LMS server, there is no need to think about the exercise status, scoring etc. – because all these issues are handled by the standard player integrated with the server. The content producer simply has to describe exercises using some standard data format. It is a really great idea. But the choice of QTI 1.2.1 as the data format for interactive exercise representation, and restricting this format in the CC profile, makes it difficult to create anything except some really simple content.

2.3 Key Differences

The main difference between the YDP and CC content is the existence of the dedicated player for the YDP content. It doesn't exist in Common Cartridge and all interactive content for CC must be described by means of QTI 1.2.1 files.

Another important difference is the complete separation of learning and examination in CC – while in YDP content most exercises are alternated with static content on lesson pages, and they together form an interactive learning experience. Obviously, we also make use of the "examination" concept (just like in CC) and the series of exercises that form an assignment also exists in our content – but they are not dominant. We believe, that solving exercises is a part of a learning process and exercises should not be limited to examine the students only.

The really limited number of exercise types available for CC is another problem for implementing a conversion of our content. It is not possible to use connection exercises, word jigsaw, sentence jigsaw, element grouping, text item identification, crosswords, marking elements, colouring pictures, solving mathematical equations and many more It makes the straight conversion impossible.

And last, but not least – the visual appearance. We put much effort in graphic design of our lessons. Pages are beautifully laid out by graphic designers to satisfy both the aesthetics and the ergonomics. As this would not be a problem to obtain the same level of aesthetics on the Common Cartridge static pages – it is impossible to achieve it with assignments. Using the old QTI 1.2.1 there is no way to express the layout and graphical appearance of the page. We must reconcile ourselves to simple text exercises laid out one per page. We are aware of the fact that some aspects of visual appearance can be defined using "text/html" instead of plain text. But this potentially powerful feature is not well explained in the specification – so we don't know, what is the intended scope of the HTML that can be used in QTI <mattext> elements. It would be rather unrealistic to believe, that one can put a complicated HTML there – with pictures, tables or JavaScript for example. So we believe, that "text/html" means just some text attributes – like bold, italic or font colour and size. While it is useful to have such possibilities, they would not allow us to design the layout and create a visually stunning page.

2.4 The Solution

As described before, it is impossible to make a straight conversion of our content into CC, because of the limited number of exercise types available, and because of the strict separation of lessons and exercises in CC. However, we've invented a workaround for this problem.

The CC specification allows us to put any "web content" into the package, so we can also include our player – which is, in fact, just a big SWF file that can be run in a flash plug-in in a HTML browser. The player will not be able to communicate with the LMS server (because it is not the CC way of doing things), but it will be able to play our content without any problems, with all kinds of exercises and multimedia files, with attractive appearance and with the pages layout that we designed. But from the CC server's point of view – it will be just the static content and the exercise results and state will not be reported to the server. From the student's perspective, however, lessons with educational content alternated with exercises still form a very valuable learning experience: interactive exercises can check student's answers, show hints, display feedbacks and show results. The results and exercise state are transient and will not survive when the student closes his session with the system, but they assist the student during the course of learning.

And then it comes to the examination. We have to stick with the CC convention and try to represent some exercises in QTI 1.2.1. We decided that for each lesson we will prepare some additional pages with simple exercises that can be converted to QTI, and they will form an examination after the lesson.

So each of our packages will consist of an interactive lesson with exercises that do not report results to the LMS, and the assignment with some exercises related to that lesson – prepared in QTI 1.2.1.

3 Implementation of the YDP → CC Converter

We have designed and implemented the application that takes a number of lessons from our existing content and includes them in the Common Cartridge package. The application is written in Java, and it has the user interface that makes it possible to select lessons for conversion and specify the conversion type. The lesson data can be interpreted in two different ways, depending on the type selection made. It can be either "lesson" or "test".

3.1 Lesson Conversion

If a "lesson" conversion type is selected, all the data files (various XML and media files) are copied into the package directory structure and appropriate entries for each file are added in imsmanifest.xml file. The directory containing all the files looks almost identical, like in our original content; the location of files in relation to each other is preserved. The only change to the original content was the necessity to change filenames of most xml files – I will explain it in details later.

For each directory that contains the data for a single lesson, a HTML file is generated during the conversion. This HTML contains a Flash Player applet, that loads our dedicated player (in SWF file), and this player loads end executes the lesson content described in XML files.

The dedicated player is also copied to the package into a separate directory, and is referenced from each generated HTML file. All the files described above are registered in imsmanifest.xml file as a "webcontent" data type. From the server's point of view, they just form a HTML page with the embedded SWF file.

Everything should run on the CC server as a static content – i.e. without the possibility to report the exercise results to the server, and without the persistent storage of exercise state. However, the content in the cartridge should look and behave identically to our existing applications.

We validated generated packages using the Cartridge Validator application (Common Cartridge Test Tool). This application proved really useful and helped to find some bugs in the package structure. Finally, we had the validated package that truly conformed to the CC specification.

The final step for static content conversion was to run it in some reference Common Cartridge player and verify that it works just like expected. The problem was the apparent lack of the reference CC player on the IMS Global site. We couldn't find this important resource, so the only option was to use the player from Icodeon for testing purposes. Thankfully, we had no difficulties getting the access to Icodeon site to try their player.

We had only one problem during this stage of implementation. The package generated just like described above didn't seem to work It seemed strange, because we expected that the package that validates correctly will, at least, show some signs of working But we had just a grey rectangle. After some debugging we found that most of our XML files looks like they did not exist on the server (we got error 404 for them), while some others XML files and all media files (JPG, SWF, MP3 etc) were okay. All the missing files had something in common: they were XML files, but their filename extension was different: .page or .flo for example. The possible explanation of the problem could be the MIME-type settings on the server (ignoring all unknown file types), or the package import algorithm, that rejects unknown file types. Anyway – we had to rename all the files to XML. It was not that easy, because there was the need to find all internal references to renamed files in XML and change them. Of course, we finally succeeded and we've got our content perfectly running on the Common Cartridge server!!!

3.2 Exercise Conversion

If a ”test” conversion type is selected, the YDP content is treated in a different way. It has to be converted to QTI file. Since this conversion is impossible for our stock content, we use the content created especially for this purpose. It contains only two exercise types (single/multiple choice and fill-in-blanks), and text modules for presentation of questions. We tried to get the most out of the QTI specification and used advanced features, like feedbacks. However, the results were mixed, some features that we wanted to implement were acceptable in the CC specification, some were not. The specification is not clear in many places, and it was an additional difficulty. We were also surprised with the fact, that even these simple exercise types allowed in the ”CC profile” do not always work as expected: we couldn’t convert our fill-in-blanks exercises to CC, because they usually (in fact – almost always) contain many blanks in a text. And it looks that ”CC profile” restriction to the QTI doesn’t allow us to have more than one gap in a single exercise, and this single gap must be at the beginning or at the end of a sentence...

But the main problem was the exercise scoring method in QTI 1.2.1, which is overly complicated and difficult to implement. Wouldn’t it be better to select some simple and standard scoring methods for the exercise, instead of this big and complicated <resprocessing> section for each item? Anyway – the problem with <resprocessing> is much more serious: despite its complexity, its abilities are really small. This is because of the CC Profile restrictions that force to use the SCORE variable only in ”all or nothing” manner. It means that if we have the multiple choice exercise with ten checkboxes, five of them being correct answers – than we have no way to express that we want 20 points for each correct answer. All we can do is to set 100 points if all five correct checkboxes were selected. In case of any error, we have to set 0 points for the entire exercise. This single deficiency makes the whole CC system useless for the representation of any high quality content. Really, only very simple exercises, like choice with a single correct answer (with radio buttons instead of checkboxes), or fill-in-blanks with a single gap, can be correctly scored.

The YDP converter creates QTI files that can be positively verified by the Common Cartridge Validator. However, when we run them in the CC player, which is available for us, only fragments of exercises appear and they do not act as expected. Perhaps it is because the Icodeon player is also under development and maybe we just run into some not implemented features, or just bugs. But at this stage of development we really need a rock-solid reference player to verify our implementation in practice.

3.3 Problems and Suggested Solutions

The most difficult part of this project was to study the specification. The Common Cartridge Specification document [1] is not precise, not coherent and even sometimes internally contradictory. Some simple, obvious topics are explained with such a great amount of detail that makes them almost unreadable (chapters 3.3 and 3.4 about file location in a package and references between them, for instance) and difficult to understand. On the other hand, some topics that should be explained with more details are too brief. This is the case with the QTI description (chapter 4.9). The diagrams for QTI are helpful, but they are contradictory to the description sometimes, especially when it comes to differences between QTI 1.2.1 and QTI Common Cartridge profile. We couldn't find definite answers for many questions – for instance whether one can use pictures or video in exercises. The diagram in paragraph 4.9.8 of the specification [1] suggests that it is possible (see MaterialSelection box), the description in paragraph 4.9.1.4 and 4.9.2 says – that only text is allowed. Such inconsistencies are frequent in this document. Writing a new version of the "CC profile" specification, similar in quality to the QTI 1.2 specification document [2] (which is, in contrast, very clear, precise and easy to read) would be very much appreciated by developers.

The Validator tool proved to be really very useful. However – it didn't find a problem with XML files that have the filename extension different than .xml. We suggest updating this tool, including also filename extension validation.

There is an urgent need for the official reference Common Cartridge player, which would be available to any developer that is interested in CC package creation. We just can't convert anything to CC if we can't see the proof that everything is working correctly. The specification, that doesn't give clear answers for many questions, makes the problem even bigger. Many issues just have to be tested in a reliable player. The reference CC player is on the top of my wish list today.

4 Conclusions

Despite the obvious limitations of the Common Cartridge, we want to continue our efforts to familiarise ourselves with this standard and prepare some software tools, that would allow the conversion. However, Common Cartridge has too many limitations at the moment to be considered our main format for packaging the content. But we acknowledge, that it has not been designed for such a task. It was designed to make possible the creation of small exchangeable items, learning objects that can form building blocks for a teacher, who wants to create her own course. We would be happy to be able to create such small building blocks in the future, based on our existing content. But for this application to be successful, the CC standard should evolve and allow for more interesting, more complicated content.

We would appreciate some improvements and changes – the replacement of the "QTI 1.2.1 CC Profile" with something better would be our first postulate. One possible solution could be a move to QTI 2.0 that solves many problems of the old version (it has quite broad choice of exercise types, more logical and concise syntax, support for HTML-like layout control and CSS styles). Or, as a minimum, the new version of the CC profile still based on QTI 1.2.1, but with less restrictions (especially in <material> sections and in the <resprocessing>

area). We are open to discuss these issues and exchange ideas to make this standard better and really useful for the software industry.

References

[1] IMS Common Cartridge Profile, Version 1.0 Final Specification
(`http://www.imsglobal.org/cc/ccv1p0/imscc_profilev1p0.html`)

[2] IMS Question & Test Interoperability, Final Specification Version 1.2
(`http://www.imsglobal.org/question/index.html`)