Building open educational resources: striving for flexibility and reuse

Imagine a utopian system which is made up of truly stand-alone open educational resources. The resources are distributed over the web; accessible using RESTful Web Services[1]. In this system we browse for resources using faceted navigation, bringing together the beginnings of a packaged course for the web. As we click through the freshly packaged course, we notice the license on each page; derived autonomously from the machine readable licensing layer for each individual resource. Branding our course is a cinch; we see the course instantly transformed as we choose from a selection of responsive web design themes, providing optimal viewing for our diverse audience.

The principles required to achieve something like this are platform independent; they are methodologies, supporting openness by providing flexibility. Technology will evolve, providing more and more opportunities but the content is the foundation and this is where we will now focus.

Content is open to the extent that its license allows users to retain, reuse, revise, remix and redistribute (Opencontent.org, 2014). The creative commons public copyright licenses provide flexibility and choice when it comes to licensing content. Version 4.0 of the creative commons core license suite, published on 25 November 2013, (Wiki.creativecommons.org, 2014) is more user-friendly and more internationally robust than ever before (Creativecommons.org, 2014). Creative commons now incorporates a unique and innovative three-layer design: the legal code layer, the human readable layer and the machine readable layer (Creativecommons.org, 2014).

In principle we have an entirely flexible framework for creating open educational resources, but how do we leverage this? Let's draw on 50 years of computer programming principles for inspiration.

Modularization, Loose Coupling, High Cohesion and the "don't repeat yourself" (DRY) principle, promote reuse of information. Ultimately "every piece of knowledge must have a single, unambiguous, authoritative representation within a system." (Pragprog.com, 2014). This makes sense and reflects how search engines see the web; promoting content which is believed to be authoritative and relevant. Let's discuss a few of these principles in turn.

Modularization allows resources to be split into logical entities. Modules exist independently from each other but can be combined to achieve a common purpose.

Loose Coupling; each educational resource is independent. There is no interdependency, ergo adding, removing or replacing a resource from a course would have no effect on the remaining resources per se. loosely coupled components are able to be mixed and matched in a variety of ways.

High Cohesion is where each entity has a very specific purpose, in a word simplicity. An example of this would be a resource which teaches one concept brilliantly and nothing else, making it the authoritative source of information on the topic, and therefore a valuable and reusable resource. The resource would be simple, almost, to the point of self-documenting.

Once the foundational principles are implemented in our design we can turn to nomenclature and metadata to begin our concrete implementation. The Learning Object Metadata (LOM) standard provides a robust framework which facilitates searching, evaluating, acquiring, and utilizing Learning Objects as well as sharing and exchanging Learning Objects across any technology (Ltsc.ieee.org, 2014) falling in nicely with our principles and making our utopian system feasible.

The next step is to combine everything we have discussed here with event-driven, non-blocking I/O platforms to create real-time applications that run across distributed devices. Emerging technologies like wearable devices are going to demand better design than conventional technologies. Our utopian system will continue to facilitate learning through ubiquitous computing as compact, inexpensive, networked devices progressively materialize as part of our everyday lives.

[1] You've built web sites that can be used by humans. But can you also build web sites that are usable by machines? That's where the future lies, and that's what RESTful Web Services shows you how to do. The World Wide Web is the most popular distributed application in history … they're missing out on its advantages (Richardson and Ruby, 2007).

References

Opencontent.org, (2014). [online] Available at: http://opencontent.org/definition/ [Accessed 1 May. 2014].

Wiki.creativecommons.org, (2014). *4.0 - CC Wiki*. [online] Available at: http://wiki.creativecommons.org/4.0 [Accessed 1 May. 2014].

Creativecommons.org, (2014). What's New in 4.0 - Creative Commons. [online] Available at: http://creativecommons.org/version4 [Accessed 1 May. 2014].

Creativecommons.org, (2014). About The Licenses - Creative Commons. [online] Available at: https://creativecommons.org/licenses/ [Accessed 1 May. 2014].

Pragprog.com, (2014). The Pragmatic Bookshelf | Welcome, Pragmatic Programmer!. [online] Available at: http://pragprog.com/the-pragmatic-programmer [Accessed 1 May. 2014].

Ltsc.ieee.org, (2014). IEEE LTSC | WG12. [online] Available at: http://ltsc.ieee.org/wg12/ [Accessed 2 May. 2014].

Richardson, L. and Ruby, S. (2007). *RESTful web services*. 1st ed. Farnham: O'Reilly.

Nodejs.org, (2014). node.js. [online] Available at: http://nodejs.org/ [Accessed 2 May. 2014].