

Bookmarks and Links: why the LMS should understand what's inside the SCO and visa versa

Version 1.0, 10th June 2008

Ian Leader, Director of Consulting (ian.leader@line.co.uk)

Copyright © 2008 LINE Communications Group Ltd.

Distributed under Creative Commons Attribution 2.0 UK: England & Wales Licence (<http://creativecommons.org/licenses/by/2.0/uk/>)

1. Abstract

This white paper discusses the SCORM requirement that a Learning Management System (LMS) should have no knowledge or understanding of the content of a SCO. We believe this is a restriction that is hampering the usefulness and interoperability of e-learning in the real world, and should be lifted. We describe a mechanism by which this might work.

2. Problem definition

One of the great strengths of the web browser and web site is the ability to link to specific pages, and share these links via email, placing them in other web pages or documents, or creating bookmarks. Most modern web sites support this, either by having a unique URL for each page, that appears in the address bar, or by providing a 'link' feature in the user interface of a page which will generate and display a link that can be used in this way.

Unfortunately this is not the case in SCORM-conformant LMSs. SCORM requires that the LMS should treat a whole SCO as the smallest unit of content, and provides no mechanism for accessing a page or section within that SCO.

This means that pages within e-learning content are much harder to share than other web pages. The only way to achieve this in today's SCORM is to create content where each SCO contains one page. In our experience, this is highly impractical, particularly if you are trying to create strongly branded content, as is the case with the vast majority of our clients. Furthermore, we are not aware of anyone in the UK who has taken this approach on a significant and successful e-learning content development project.

We believe that there is significant benefit in being able to launch e-learning content at a particular page or section, so that learners can be directed to specific areas of content without additional navigation or searching on their part. We also believe that this should not be separated from the progress tracking functionality of LMSs, as learning should be recognised whether it is accessed through formal channels (e.g. searching a catalogue) or informal channels (e.g. clicking on a link that a colleague has emailed to you). Furthermore, tracking gives owners of LMSs and content valuable intelligence as to the use of content and provides a starting point for ROI calculations and future investment decisions.

3. Use cases

There are three use cases we are considering:

LINE

1. A learner wants to share a page or section within an e-learning course with colleagues
2. A learner wants to bookmark a page or section within an e-learning course – either on a browser or a ‘social bookmarking’ site
3. A learner wants to access a page or section within an e-learning course either with a bookmark they stored earlier or a link that was sent to them.

For a real world example, consider the following: In 2005, we developed a substantial e-learning course on lubricants for a major player in the energy sector. We provided a bookmarking facility within the content, which stored bookmarks in the LMS (using the suspend data field) so they would be remembered between sessions. This was very well received, and almost immediately learners and managers started asking how they could email these bookmarks to colleagues and direct reports. Because of the limitations we discuss here there was no way for them to do this, other than writing a list of section names and page numbers into an email.

4. Proposed solution

We propose that this problem be solved by modifying or extending SCORM in three areas:

1. relaxing the requirement that the LMS should consider the SCO as the smallest unit of content
2. providing a standard way for the LMS to launch a SCO, specifying that it should navigate directly to a particular page or section, and
3. providing a standard way for the SCO to find out what URL should be used to launch the SCO, at a particular page or section

4.1. SCO as smallest unit of content

We propose that the LMS should be somewhat aware of the internal structure of a SCO, as a prerequisite to the 2nd and 3rd points in our proposed solution. However, this is a limited awareness: we are not proposing that the LMS should know the exact internal structure of the SCO, only that it should be aware that there is an internal structure, and pass a reference to that internal structure, when directed to do so. This is not entirely different from the suspend_data and launch_data and lesson_location fields in the current versions of SCORM, which ultimately hold data related to the internal structure of the SCO.

4.2. Launching the SCO at a particular point

We propose that there should be a standard mechanism for the LMS to launch the SCO at a particular page or section of its structure. This could be achieved by creating a new run-time data field, which is set by the LMS when it launches the content. For example, cmi.launch_location. This would contain a page number, section reference or some other text ‘key’ that would direct the SCO where to open upon launch.

It is important to note that the LMS does not know the full details of this structure. Instead it is blindly passing this through, having been directed to do so. This therefore requires that SCORM dictate that an LMS must have a mechanism by which it can be directed in this way, and we propose that this be through an HTTP GET request, ie. a URL. The structure of this URL would be up to the LMS implementation, but at a minimum it would need to include:

- A unique reference to the content object in questions, and
- A reference to the page or section at which it be launched

In addition, this URL would be user-independent, so that any user could use it (subject to verifying their user credentials to the LMS).

4.3. Creating a launch URL within a SCO

For this proposal to work in practice, there needs to be a simple, learner-friendly way to create the URLs referred to in the previous section. We propose that responsibility for doing this lie with the SCO, but that the LMS must pass the necessary information to the SCO through the run-time data model. For example, a pair of data fields such as:

- cmi.launch_url.prefix
- cmi.launch_url.suffix

would allow the SCO to create a page or section specific URL by sandwiching the page reference between the prefix and suffix. For example, with a prefix of:

```
http://lms.mycompany.com/launch?course_id=1234&bookmark=
```

and an empty suffix, the SCO could easily create a link like:

```
http://lms.mycompany.com/launch?course_id=1234&bookmark=section_2_3
```

Which could then be exposed to the learner through a 'link to this page' option in the SCO user interface (as is found in the majority of modern content-generating web sites), like the following example from Google maps:



Our proposed solution makes no assumption about what constitutes 'section_2_3'. It could be a web page, a particular frame in a flash movie, or even a PDF document.

5. Integration and other technical issues

We have identified three issues that might arise in the implementation of this proposal:

1. Some LMSs do not support user-independent hyperlinks to launch content (we are aware of at least two major players that have no support or limited support for such functionality). A user-independent link to launch content is clearly a pre-requisite for a link that takes the learner to a sub-section of that content. However, we believe that in the majority of cases this will either not be a problem, or will be relatively simple to resolve.
2. The format of the URL for the link will vary significantly between different LMS implementations. In particular, not all implementations may work with a section reference at the end of the LMS. This is the reason we have proposed that both prefix and suffix strings be available if needed when creating the ULR
3. Designing and implementing a link / send / bookmark panel puts an additional burden on content authors, and as with all functionality implemented within content, could lead to inconsistencies between different content intended for the same audience. An alternative approach is for the LMS to handle the act of generating the link. This approach would work as follows:
 - a. The LMS would provide a URL prefix and suffix that is combined with the section reference from the content

- b. This URL opens a web page in the LMS (either as a browser popup or iFrame) that contains the user interface for viewing a link or sending it.
- c. The user interacts with this UI, which can provide the same linking and sending options as described above, as it is aware of which section reference has been provided – although the LMS still has no other knowledge of the internal structure of the SCO
- d. One benefit of this approach is that the linking user interface could be customised to support functionality specific to an organisation, such as an organisation's internal 'social bookmarking' site.

6. Summary and recommendations

In summary, we propose a mechanism by which hyperlinks can be created that link directly to particular pages or sections of a SCO, without requiring the LMS to be aware of the full structure of the SCO. These links could be used for sharing content via email, referencing content in web pages, or bookmarking using browser or 'social bookmarking' functionality to name a few. We recommend that this be implemented in SCORM 2.0.

7. LINE Communications

LINE Communications is a learning and communications specialist. We advise our clients on every aspect of learning, communications, technology and related business issues. For almost 20 years, we have been working with clients in government, multinationals and the MoD, helping them develop and train their employees and partners and communicate with internal and external audiences. LINE has offices in London, Sheffield and Zurich. For more information see <http://www.line.co.uk>.