NEW YORK UNIVERSITY

User Needs Report: eLearning Modules

Spring 2019

Context

eLearning modules are developed to add a point of interactivity to a website. Modules are typically embedded into a webpage to facilitate assessment, provide a single, self-contained way to present content, and allow variability in said presentation. Faculty, administrators, and instructional technologists typically leverage this format in conjunction with an LMS, such as Sakai / NYU Classes, for example, to gather metrics and analytics data as the user progresses through a given module.

A module can vary quite a bit in terms of complexity. Some modules are akin to a digital slideshow presentation, sometimes with a quiz for the user at the end. These can be used to present lecture content similarly to how a professor may run through his or her slide deck in a classroom setting. Others are miniature games, providing a user with several points of interaction and branching scenarios. These might be more akin to a simulation.

On a separate spectrum, module complexity is increased by accessibility requirements, analytics functionality (including the need to interact with metrics-gathering platforms, such as SCORM), and integration with a host of different environments.

Definitions

- **eLearning Modules**: Modules are most often web/browser-based applications that allow for interactivity between the user and the content. This can be in the form of assessment questions, responsive diagrams and imagery, game mechanics, or a variety of other scripted mechanisms.

Considerations

- eLearning Modules can be authored in a variety of platforms with varying output.
- Authoring platforms include Adobe Captivate, Articulate Storyline, Rise, Lectora, and many others. Open-source tools exist, such as Xerte, may offer similar functionality.
- Design of a module can vary wildly depending on its eventual use; authoring tools can allow for a huge variety in presentation and format of module.

  - Modules must be hosted somewhere, and security restrictions can be a concern for how they’re implemented.

  - Modules need to be sustainable, i.e., any future updates to content, browser compatibility, functionality, etc., should be straightforward and manageable for faculty members and staff. Long-term support needs should also be considered as related to administration of hosting platform, choice of authoring tool, and desired analytics platform.

  - Needs listed below do not differentiate between LMS requirements for supporting e-learning modules, requirements for the module-authoring tool itself, and requirements for hosting, as the lines between these requirements may differ depending on the tool and the LMS.

Baseline Needs

- SSO
- Accessible
- LTI-compliant
- NYU Analytics standards compliant
- Responsive across devices

- Authoring Needs
  - Ability to upload module from a link in the LMS (or as an embedded tool within the LMS)
  - Ability to pull analytics easily from the emodule
  - Preference is for adopted platform to offer the option for web-based tool use (to facilitate collaboration)
  - Ability to embed questions (surveys, quizzes, tests, etc.), text, video, audio, images, & web content/object into the module
  - Support for captioning directly in tool.
  - Support for composition in multiple languages; handle non-Western character sets
    - Ability to have multiple language versions in closed captions
  - Ability for voice-over narration - options:
    - Option for robovoice / computer-generated narration
    - Audio recorded within tool
    - Audio imported from another source
  - Shareable, templated style for module creation
  - Ability to share draft versions with collaborators and reviewers
  - Navigation options beyond uni-directional, one-chance-only, live progress though module:
    - Ability to save progress and:

      - Allow to skip previous content
      - Allow to pick up where left off
    - Ability to branch
    - Ability to allow user to skip to appropriate place in lesson if they answer in a certain way (e.g., correctly/incorrectly)
    - Ability to have rich options for tab ordering
    - Ability to search within the module for terms and phrases
  - Adaptive release needs:
    - Options to:
      - allow/not allow student to see correct answer to question before proceeding in module
      - Release next page in the lesson based on completion of prior page (options for release on passing, merely completing, or no restrictions)
      - Release lesson content based on completion of prior module (an entire module, as opposed to a page in a module)
      - Options for instructor to set module to:
        - Have students take assessment multiple times and record results multiple times
        - Have students take assessment once and only once and record result only once
        - Have students take assessment multiple times and record only final result
  - Robust store of stock imagery for use in creating lessons/modules
  - Ability for schools/departments to restrict creation of e-learning modules
  - Option to enable/disable print from module
    - Including certificate of completion or save/send to appropriate location
    - Printable version of entire module or parts of module
  - Ability to import slides from PPT, Google Slides, Prezi, etc.
  - Ability to customize module behaviors through scripting (JS)
  - Ability to screen capture/record

- Sharing Needs
  - Ability to embed in LMS through Lessons, Syllabus, Resources, etc.
  - LMS's ability to set deadline for completion - options not to accept after certain date, to accept but note late completion date, accept but take instructor-set automatic deduction for late submission
    - Ability to extend deadline for individual student or all students

- Reporting Needs
  - Consistent grading and compatibility within Gradebook.
  - Fix for SCORM-related reset of a completed module if the user reopens the module (gradebook only records the latest interaction with the module even if it

    isn't completed, but does not store the most recently-completed instance -- see Adaptive Release needs section above for related functionality requirements)
    - Options for notification of student completion/non-completion/progress of lesson module
    - Ability to print, export, and archive report/analytics from module
    - Ability to generate report within a specified time range
    - Ability to generate site-wide or department-wide report

Eventual Needs

- Ability to include 360º images., video, objects, and related content, as well as add interactivity to this content (questions, annotations, links, audio, etc.)

Nice to Have

- Ability to pass information between two or more modules (i.e., assessment data, user info, etc.)
- Adaptive learning capabilities (the ability of the module to customize content or the sequence in which it is delivered, the frequency of reminders to user to refresh knowledge, etc., based on the individual learner's learning patterns)