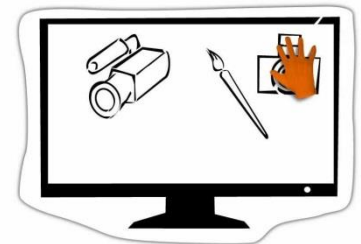


# *Educational Technology*



## Authoring tools

K. Vassilakis / M. Kalogiannakis



[www.youtube.com](http://www.youtube.com)

# Authoring tools



*commons.wikimedia.org*

- Definitions
- Categories
- Issues
- Recommendations
- Current trends

# Learning content in eLearning era

- Learning content should have educational soundness; thus, it should be informative, interesting and relevant.
- Given the fact that in eLearning era instructors have been moved from the foreground to background, educational material is taking the place of instructor.
- Ideally, competent educational material should work with students without instructor's participation.
- Interactive and adaptable with assessing, sequencing and tracking capabilities and in line with modern learning theories.
- Developing sufficient content, it is not straightforward process.
- There are needs for tools to develop content for delivering eLearning activities (tools for instructors).

# What is an authoring tool?

- Authoring tools are software that instructors use to create content, artifacts and premises for activities to support education & training.
- They offer capabilities and functionality to create, edit, review, test and configure teaching/learning activities.
- Range from simple tools (convert instructional slides to web pages) to advanced software (those for creating a wide array of sophisticated applications).
- May not be designed specifically for eLearning.
- They can be part of systems/platforms that perform broader eLearning functions (LMS, LCMS).
- Most developers of eLearning products use more than one tool during the production process, and a substantial number use four or more (one primary tool and auxiliary tools).



# Flash activity

**Point out some reasons why authoring tools are useful...**



[commons.wikimedia.org](https://commons.wikimedia.org)

# Why use authoring tools?

- Reduce technical overhead (unlike programming).
  - *With WYSIWYG interfaces allow users to easily manipulate and configure eLearning content and activities, using familiar visual metaphors.*
- Reduce the skill set requirements.
  - *Untrained users can start using a tool and producing training content very fast (sometime within minutes).*
- Automate time-consuming tasks.
  - *Usually offer streamlined and efficient approaches to the authoring process.*
- Educational organizations want tools that are easy to learn and can be used by a large team of people with different skill sets.
- Enforce standards and workflows.
- Generally, facilitate eLearning delivery.



# Categories of authoring tools

- Major categorisation (according to ADL):
  - *Self-contained (sufficient) authoring environments*
  - *Learning content management systems (LCMS)*
  - *Virtual classroom systems*
  - *Mobile learning development tools*
  - *Performance support developments tools*
  - *Social Learning development tools*
  - *External document converter/optimizer tool*
  - *Intelligent Tutoring Systems*
  - *Auxiliary tools*
- Categories not mutually exclusive.
  - *Many tools have attributes that qualify them for two or more categories.*



[en.wikipedia.org](http://en.wikipedia.org)

# Self-contained authoring environments

- Build *entire eLearning courses* using capabilities within the authoring tool.
- Incorporate WYSIWYG features.
- Object-oriented approach for structuring eLearning activities.
- Web site development tools.
  - *These are open-ended tools for web site design; they can be used for any type of web site or web pages, including e-learning (HTML, CSS & Javascript).*  
Dreamweaver®, Visual Studio®
- Rapid Application Development (RAD) tools.
  - *These are open-ended tools for designing robust interactive applications (usually for web delivery). Produce binary runtime files.*  
Flash®, Flex®





# Self-contained eLearning development tools

- Specifically designed to produce eLearning (referring directly to the term authoring tools)

*Template-based architecture with timeline & animation capabilities and predefined objects with configurable properties*

- Cloud-based eLearning development tools.

*Installed on a cloud server (web-based interface). Enable collaborative authoring, permission/role-based production workflows, centralized control and enforcement of standards.*

Claro<sup>®</sup>, Lectora Online<sup>®</sup>, Litmos Author, Udutu (free), SmartBuilder<sup>®</sup>, Oppia (open source), Xerte tools (open source).

- Desktop-based eLearning development tools.

*Perform better than web-based tools, and have more features, but do not support collaboration.*

e-Learning Suite<sup>®</sup>, eXe (open source), Xerte (open source), Adapt (open source) Captivate<sup>®</sup>, iSpring Suite<sup>®</sup>, SmartBuilder<sup>®</sup>



# Simulation development tools (self-contained)

- They are specifically designed for developing simulations and their component animations.
- System simulation (recording of what is happening in a computer screen - screencasts).
  - *Camtasia Studio®*, *Captivate®*, *Assima®*, *Articulate Storyline 360®*, *RADs* also
- 2D simulation development tools (simpler to learn)
  - *SimWriter®*, *GoAnimate®*
- 3D simulations tools. Model the physical worlds by rendering artifacts and environments and by using libraries of textures and skins.
  - *CodeBaby Studio®*, *Flex Builder®*, *Kuda®* (open source), *SimWriter®*.
- Video role play tools (video scenarios where users participate with cameras)
- Transmedia story-based tools (using multimedia create immersive learning scenarios )

# Some more self-contained

- Game development environments. Tools in this category are specific to a particular game engine or game standard.
  - *Unity®*, *GameStudio®*, *GameSalad®*
  - *RAD and simulation tools are used also*
- Virtual world development environments. They are specific to a particular virtual world or virtual world type.
  - *Second Life®*, *OpenSim (open source)*, *Open Wonderland (open source)*, *OpenQwaq (open source)*, *Virtual World Sandbox (open source)*
  - *RAD, simulation and game tools are used also.*
- Database-delivered web application systems. Separation of content and appearance. Developers store the content in databases and apply formats to them on a presentation layer at runtime. Flexible updating, but server software is required.
  - *ColdFusion®*, *ASP.Net®*



# Learning content management systems (LCMSs)

- Integration of authoring functions with content management (store, delivery).
- Users do not develop actual files during the authoring process.
- They assemble and deliver virtual learning objects dynamically at runtime from database and file elements from a central content repository.
- Provides great flexibility for reuse of content and media.
- Similar to the database-delivered web application.
  - *Atutor*<sup>®</sup> (open source),
  - *Learning Essentials*<sup>®</sup>
  - *Mindflash*<sup>®</sup>
  - *SAP Enterprise Learning*<sup>®</sup>,
  - *Xyleme LCMS*<sup>®</sup>



# Virtual classroom systems

- Platforms specifically for creating eLearning that is delivered via an online collaboration tool.
- Instructional designers use these systems to author synchronous or asynchronous virtual classroom training.
- The collaboration functionality is usually combined with the authoring functionality in one system.
  - *Most are capable of creating asynchronous eLearning (synchronous sessions can be recorded).*
- These are not standalone systems, because they require files to be generated externally and imported.
- Examples:
  - *Blackboard Collaborate<sup>®</sup>, Connect<sup>®</sup>, Classroom<sup>®</sup>,*
  - *Big Blue Button (open source), Adobe Presenter<sup>®</sup>*



<https://scherlund.blogspot.com>

# Mobile learning development tools

- Use mobile device screen templates to deliver content to mobile devices (mobile learning).
- Producing output files that work with the OS of mobile device.
- Most authoring tools can now deliver content to mobile devices, but:
  - *there are some specifically designed for mobile learning,*
  - *some are optimized to provide eLearning content through the device's web browsing capability (responsive design),*
  - *others target only specific devices with specific screen capabilities (for example, the Apple iPad®).*
- A key feature for these tools is HTML5 output, since that is supported “out of the box” by mobile devices. Responsive design is another key feature.
  - *LearnCast®, CellCast®, MASLO (open source – under development)*



# Performance support development tools

- Performance support are learning aids for providing employees/workers with the information and resources they require to perform a task.
- They are designed to support *informal training* by guiding employees when at work.
- Usually, they are on-the-job support tools, at the precise moment of they are needed (“just in time”, “just in place” support - the right information at the right moment).
- They should be designed to be easy to assimilate and easy to apply.
- Various tools for creating:
  - *Quick reference guides, Process Maps, Infographics, How-To-Videos, Checklists, FAQs, Learning Portals, mobile applications*



# Social learning development tools

- Create learning that is based on learner-generated content.
- Peer-to-peer communication, and collaboration provided by social media tools.
- Use of these features in eLearning is increasing rapidly.
- Support publishing learning modules that include formats as:
  - *Wikis (for example, Wikipedia®)*
  - *Social networking (for example, Facebook®)*
  - *Blogs (for example, Blogger®)*
  - *Micro-blogs (for example, Twitter®)*
  - *Picture sharing (for example, Flickr®)*
  - *Video sharing (for example, YouTube®)*
  - *Communities of practice (CoPs)*



<http://mediacause.com>



# External document converter/optimizer tools

- This category includes what is known as “rapid eLearning development tools”.
- Tools that enable developers to convert row content to eLearning material.
- Provide limited ability to develop eLearning from scratch.
- Designed to import external instructor-led training documents and convert them to web-based eLearning formats (HTML5) by adding some interactivity.
- Generally, they are simple and easy to use, thus enabling non-technical staff to do the authoring without requiring teams of developers with specialized skills.
- Web-based external document converter/optimizer tools
  - *AuthorPoint®*, *Brainshark Rapid Learning®*
- Desktop-based external document converter/optimizer tools
  - *Articulate Presenter®*, *iSpring Suite®*, *Wimba Create®*, *SmartBuilder®*



# Intelligent Tutoring System (ITS) tools - category

- An intelligent tutoring system (ITS) is a computer system that aims to provide immediate and customized instruction or feedback to learners, usually without intervention from a human teacher (*wikipedia*).
- Use artificial intelligence to mimic the behavior of an expert human tutor, sometimes including dialogues with learners.
- A key difference between ITS and other forms of technology-based learning is that many ITSs dynamically generate instruction (usually based on rule sets or on results of machine learning applications, that adapt to the student's ongoing correct and incorrect expression of concepts).
- Authoring is done mostly as custom programming by system engineers.

*xPST (open source), ASPIRE, SimCore®, Autotutor Lite*



# ITS -issues

- Effectiveness of ITS is questionable. They are problematic in detecting shallow learning (*wikipedia*).
- Fail to ask learners to explain their actions (*wikipedia*).
- They are expensive both to develop and implement (*wikipedia*).
- Some level of anticipation of student responses to questions (and consequent rule sets for dealing with them) currently is required to be programmed into ITS systems.
- Inherently limit the interaction and response choices and paths the student can take to learn (*wikipedia*).
- The AI “understanding” module of the ITS system (inference engine) sometimes needs to be “trained” in the content.
- ITS courses usually contain one or more pre-developed content modules, which could be standalone.
- No universal standards that would allow interoperating between an authoring tool and an ITS.



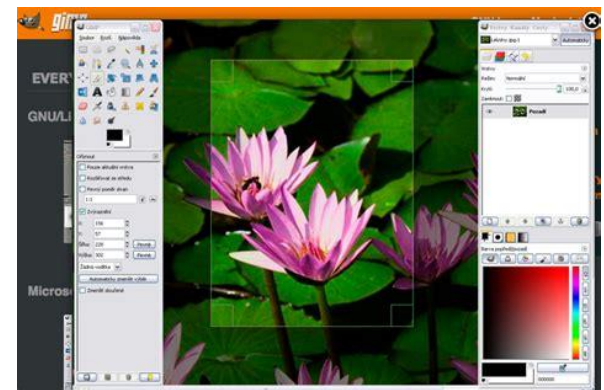
# Auxiliary authoring tools (1)

- E-learning assemblers/packagegers. Assemble objects authored in other tools into an organization/sequence of learning objects. Usually produce IMS-CP/CC, SCORM formats.
  - *RELOAD Editor (open source), SCORM Driver<sup>®</sup>, Trident<sup>®</sup>*
- Specific interaction object creation tools. Standalone applications that create modules to meet specific interactions needs that are impossible or difficult to create in your primary tool. Most of these interactions are geared towards assessment.
  - *Hot Potatoes<sup>®</sup> (free), iSpring Quizmaker<sup>®</sup>, Quizmaker<sup>®</sup>, Exam Engine<sup>®</sup>*
- Basic media asset production and management tools to create graphics, audio, video, and animation files.
  - *3DS Max<sup>®</sup>, Photoshop<sup>®</sup>, Illustrator<sup>®</sup>, Adobe Presenter<sup>®</sup> (video capture), Edge Animate<sup>®</sup>, Camtasia Studio<sup>®</sup>*



## Auxiliary Authoring tools (2)

- Word processors, page layout, and document format tools to create e-learning reference documents and store them in a convenient format (for example, PDF) that preserves their appearance.
  - *MS-Office®*, *Libre/open office*, *Acrobat®*, *Google Docs®*
- Database applications (accessed by an eLearning applications).
- Web-based collaboration tools to create collaboration mechanisms and peer-to-peer communication functions, normally for meetings
  - *elgg®*, *hum hub®*, *buddypress®*
- Web page editors to create web pages
  - *kompoZer*



<https://visualisingadvocacy.org>

## Flash activity

**Which category is more  
time consuming?  
costly?  
powerful?  
effective?**

# Comparison of categories

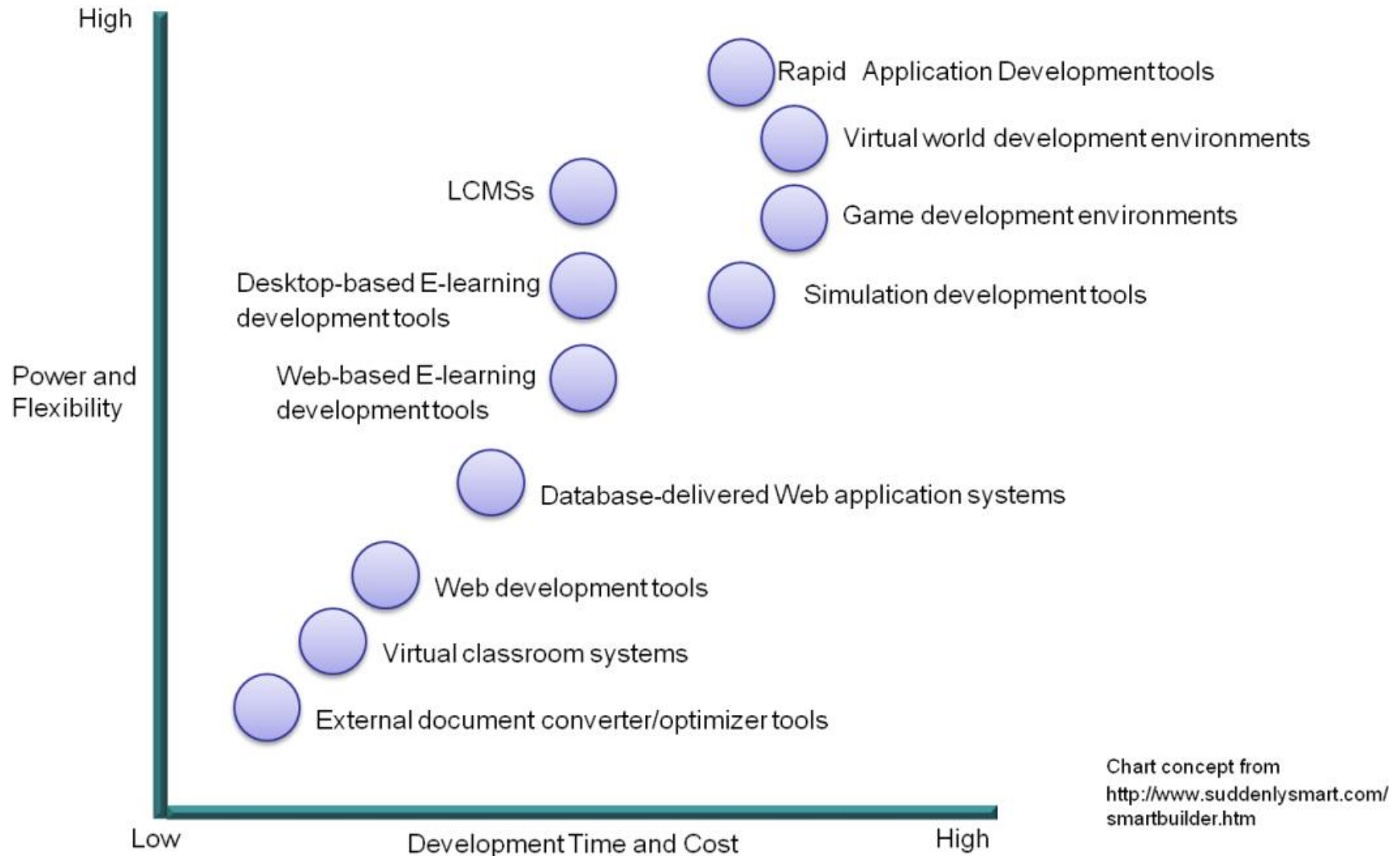


Chart concept from  
<http://www.suddenlysmart.com/smartbuilder.htm>



# Features



# Issues





# Rapid e-learning authoring tools (features/issues)

- These tools generally make the most sense where content already exists in a usable form.
- Address to developers with no experience.
- Shortened development time.
- Reduced cost - reduced effort.
- Ability to quickly and easily make changes and redeploy content (especially critical where content is volatile).
- Useful where only lower-level learning objectives need to be met.
- Template-driven architecture similar look and feel (this is not always good).
- Interactivity limitations due to the initial raw material.



# Mobile Learning authoring tools (issues -1)

- Operating systems and hardware specs are very different for mobile devices.
- Connection speed to data networks is highly variable (depends on location).
- Performance is generally considerably less than desktop computers (less memory, power processors etc.)
- Cell phones are highly personalized.
- There are different paradigms for interaction with mobile device (use fingers than mouse). Problems for rollover interactions.
- Many phones can dynamically shift portrait vs landscape orientation. Content may need to adjust accordingly.



# mLearning authoring tools (issues -2)



- There is a need to test developed content on many different platforms and emulators (which are not always 100% consistent with the actual device).
- Not all eLearning content is appropriate for mobile delivery (examples, decision support tools, job aids, help etc.).
- Output files can be standalone apps or browser-delivered. In the case of standalone apps, Flash<sup>®</sup> is currently not compatible with many devices.
- Authoring tools currently do not usually take advantage of all built-in functions (cameras, compasses GPS etc.).
- Standard interactive controls on the mobile platform work differently than on the desktop (radio buttons, rollovers etc.)

# Flash activity

**Stand alone  
or  
web-based  
mlearning applications?**



# mLearning application: standalone?

## ■ Native apps Pros

- Offer best-in-class user experience, with rich design option (games, 3D, augmented reality, accessing the file systems, etc.)
- Allows use of device features (GPS, compass) and offline use.
- Faster, with predictable performance and more secure.
- Relatively simple to develop for a single platform.
- Can charge for applications.

## ■ Native apps Cons

- Requires use of unique programming language.
- Cannot be ported easily to other mobile platforms.
- Updates generally require downloading a new file.
- Developing, testing, and supporting multiple device platforms is very costly.
- May require certification and distribution from a third party that you have no control over.



# or web-based mLearning application?

## ■ Web apps Pros

- Easy to create, using HTML, CSS, and JavaScript.
- Simple to deploy across multiple handsets (no separate versions).
- Content is accessible on any mobile web browser.
- Can be packaged & ported as a hybrid app (cross-platform tools).
- Features are consistent with features of a PC browser (thus user experience is similar).
- Easier to control access (since not available through app stores)
- Faster to deploy and update.

## ■ Web apps Cons

- Requires live Internet access.
- Optimal experience might not be available on all handsets.
- Browser support for interactivity and rich media varies.
- Don't always support native application features, like offline mode, location lookup, file system access, camera, etc.



# mLearning: When standalone? When web-based?

- Develop mLearning as a web application when:
  - *you seek cross-platform compatibility.*
  - *you can't support the development of native apps using proprietary Software Development Kits (SDKs).*
  - *Accessibility is a requirement.*
  - *Using more advanced capabilities of the device isn't required (e.g., offline, camera, accelerometer, gyroscope, etc.).*
  
- Develop mLearning as a native app when:
  - *You are charging for it (for profit).*
  - *You are creating a game.*
  - *You are using specific location information.*
  - *You are using system's devices (cameras, accelerometers, etc).*
  - *You are accessing the file systems.*
  - *There will be offline users.*



# mLearning: Good practices

- When a course is being developed for both mLearning and desktop, delivery should be developed for the mobile platform first, then modified for the desktop version, since this tends to drive simplifying the content.
- When both mLearning and desktop development, designate only certain screens for mLearning delivery.
- New authoring tools offer alternative formats that are dynamically determined by the content when it encounters the mobile device (detect the formats the device could support).
- Don't use radio buttons and rollovers.
- Use built-in cell phone themes, functions, and navigation if possible.
- Create content that is short (independent content).
- Use bullets to make contextual information more concise.
- Develop for users (user experience) instead of for devices.





# Open-source solutions

- Open-source authoring tools are attractive options, but they are not without costs (installation, customization, and support).
- Ideal situation: when an open-source tool is completely tailored to the requirements of the organization.
- It may be risky to assume that an open-source product will be usable straight out of the box. Most of the times customization is needed.
- Open-source software should have equal weight as proprietary software during acquisition evaluations.
- Staying in touch with the open-source community is very important as it can decrease customization costs enormously.
- It is also important to evaluate the strength and size of the open-source community, as well as the longevity of the product.
- Lacking adequate documentation can make a product difficult to install, use, maintain, and troubleshoot.
- It is particularly suitable for rapid prototyping and experimentation.



# Freeware and government off-the-shelf (GOTS)

- Freeware may or may not also be open-source. Freeware that is not open-source is a risky investment.
- Freeware may have restrictions on copying, distributing, and making derivative works of it.
- Most freeware also restrict to personal use, non-profit use, non-commercial use, etc.
- GOTS only applies to government entities.
  - *The government has direct control over most aspects of the product, including the source code.*
  - *The vendor or creator has given a license to the government entity who paid for it to freely use and share it within the government. The license does not permit the government to give or sell it to outside entities.*



# Hosted solutions

- Hosted tools are installed and managed on vendor's server.
- Advantages:
  - *Eliminates the cost of hardware and network infrastructure needed to support a local installation of the system.*
  - *Lowers your staff costs for administration and maintenance.*
  - *Puts less bandwidth load on the corporate network.*
  - *Content and feature updates can be accomplished without intervention by your staff.*
  - *Enables faster implementation.*
  - *Requires little or no internal technical support or development.*
- Disadvantages:
  - *Restricts opportunities and scope for local customization.*
  - *May not provide the level of security required.*



# Templates and skins

- Templates and skins can reduce dramatically the level of effort (LOE) when creating content.
- During the design phase of an educational material, authors may either develop their own skins and templates or choose them from a library.
- Templates and skins although used interchangeably are not the same. Several skins can use the same template.
- Authors choose the template or skin that applies to a screen they wish to build and populate the content.
- In this way, they can build screens without the need for skilled developer intervention.
- This saves huge amounts of time, reduces the requirement for technical expertise, and simplifies the authoring process.
- On the other hand, the use of templates can restrict creativity.



# Assessments

- Assessments within content strengthen portability.
  - *Internal content assessments can not always guarantee interoperability with eLearning platforms neither assessment interactions between different tools.*
  
- Also, most eLearning platforms offer a variety of tools for creating assessments.
  - *eLearning platform assessment is easier and ensures compliance with platform's tracking system.*
  - *Assessments delivered through eLearning platforms surely can store and verify student's level of knowledge.*
  - *On the other hand, most likely assessments cannot be exported and transferred in other platforms.*



# Compliance with standards

- Compliance with standards is very important for supporting:
  - *interoperability,*
  - *accessibility,*
  - *reusability and*
  - *durability.*
  
- Some well-known and universally accepted standards are:
  - *ADL SCORM (1.2, 2004)*
  - *AICC*
  - *IMS (CP, SS, CC)*
  - *Metadata (LOM, DC)*
  - *xAPI*
  - *Section 508 (accessibility for disabled people -in USA)*

# Responsive design

- It is not just a “fit to all layout”.
- Just shrinking or expanding elements on the screen is not enough.
- In addition to screen size, resolution, platform, and orientation of the device, the way portable devices are used should be also considered when responsive layouts are designed.
- Developers should design the layout of the content smartly adapted to the user, unlike designing separate layouts for each different device.
- Advanced authoring tools, offer various capabilities on creating responsive designs.

# Other issues

- Security considerations. Authoring tools must functionally meet the security demands of an organization. Applies to web-based tools only (content – organization level, roles):
  - *unauthorized login protection,*
  - *lock user out of capabilities (user profiles),*
  - *restrictions on content, creation permission levels, assigning roles.*
  
- Input-output file format. Support of many media file formats, import/export and conversion facilities.
  - *It important to know what file format an authoring tools can edit.*
  - *It is important to know the output format of an authoring tool.*
  - *Output file format affects the editability of the developed content within other authoring tools.*
  - *Adequate file format facilitates reusability.*
  - *HTML5 format is likely to become a universal format for eLearning.*





# Flash activity

**Can think of some possible selection criteria on quality and suitability of authoring tools?**



# Possible criteria for assessing authoring tools

- Support of a wide variety of instructional strategies and learning technologies (social media, simulations, serious games etc.)
- Sequencing and navigation (depending on learning responses)
- Assessment features (multiple choice, matching, drag & drop, etc., randomization, remediation, etc.)
- Technical characteristics of output (file format, compliance to standards etc.)
- Capabilities for authoring peripheral documents around content (glossaries, FAQ, help pages, etc.)
- Ease of learning and use (WYSIWYG, emulation/previewing, usability, etc.)
- Automation and process optimization (course navigation, metadata, spelling, etc.)



# Possible criteria for assessing authoring tools (2)

- Procurement, user training, support-maintenance, and documentation (license, pricing policy, community, etc.)
- Architecture & technical limitations (system requirements, open architecture, limits on objects, sizes etc.)
- Media handling (audio, graphics, animation, video)
- Programming features (scripting language, XML/JSON support, etc.)
- Collaborative authoring (web-based tools only)
- Access (authorization/ authentication, security)
- System performance
- Permissions and roles



# Recommendations

- As with most software, systems that are easier to learn and use have fewer capabilities, and vice versa.
- The learning goals and performance problems you are addressing should be clearly defined before choosing any particular tool.
- Determine exactly what capabilities you really need.
- Be sure to choose tools that are optimized to your instructional methods/practices.
- If technical support is missing it is better choosing a non-technical tool.
- Choose mature tools (avoid first releases).
- Check for the reputation of the vendor/developer/product.



# Current trends in authoring tools

- Team-based life cycle production and maintenance (ADDIE).
- Use of XML or JSON.
- Separation of content and appearance.
- Support for instructional design process.
- Learning object-centric architecture.
- Embedded best practice design principles.
- Automated metadata generation/extraction.
- Open architectures.
- Support for team-based learning.
- Support for learning through games and virtual worlds.
- Support for social media and semantic web/Web 2.0-3.0 technologies.



<https://en.wikipedia.org>