

Ontwikkeling van Leer Objecten met TYPO3 CMS en Adobe Flex

Introductie

Dit is een korte introductie van een concept voor de verdere ontwikkeling van een Application Management Systeem waarin de voordelen van TYPO3 CMS (T3) aanvullend zijn op de ontwikkeling van Learning Objects (LO) met Adobe Flex SDK.

Gebieden waarbij de module inzetbaar is:

- productie van dynamische applicaties t.b.v. Elektronische Leeromgevingen (ELO's),
- data lay-out en visualisatie,
- advanced interface development,
- template based Application development in Flex.

De module maakt het mogelijk om gestructureerd te werken met al bestaande CMS functies in T3. Hierdoor kunnen complexe web based applicaties in Adobe Flex door meerdere teamspelers tegelijk worden ontwikkeld, bijvoorbeeld door een team van content ontwikkelaars, applicatie ontwikkelaars (developers), vormgevers en interface designers.

In de eerste plaats kwam de module tot stand om op een flexibele manier LO's te produceren waarbij het kunnen hergebruiken en aanpassen van de applicaties een belangrijke voorwaarde was.

A learning object is a resource, usually digital and web-based, that can be used and re-used to support learning.

Learning objects offer a new conceptualization of the learning process: rather than the traditional "several hour chunk", they provide smaller, self-contained, re-usable units of learning.[1]

They will typically have a number of different components, that range from descriptive data to information about rights and educational level. At their core, however, will be instructional content, and probably assessment tools. A key issue is the use of metadata. (http://en.wikipedia.org/wiki/Learning_object)

Eén van de mogelijkheden is dan ook een LO-repository te starten waar onderwijsinstellingen en uitgevers het door hen ontwikkelde materiaal kunnen delen. Daarnaast kunnen de hierin opgeslagen applicaties ook dienen als sjabloon voor nieuwe projecten.

De CMS features als role en content management kunnen samen met sourcecode van de applicatie geïntegreerd tot een herbruikbaar package.

In ontwikkeling:

In de verdere ontwikkeling van de module staan de volgende punten steeds centraal:

- hoogwaardige applicaties
- digitale duurzaamheid
- productiviteit
- kosten reducerend door hergebruik en bundeling van krachten
- 100% open source GPL
- web-based / non-lineair productie platform

De basis van dit concept is de inzet van de bestaande API functies ten behoeve van class / framework templating (LO Packaging, MXML en Actionscript class), SCORM Packaging en het genereren van swf files via de TYPO3 IDE. De integratie van dit laatst genoemde proces is inmiddels gerealiseerd in een beta extensie.

De criteria voor verder ontwikkeling van features die een meerwaarde kunnen

zijn in het ontwikkelproces van een LO liggen vooral in het benutten van bestaande T3 functies in combinatie met de mogelijkheden van Adobe Flex.

De onderstaande features worden momenteel (door)ontwikkeld:

- prototype repository via extension-manager TYPO3 CMS
- SCORM import en export
- metadata beheer SCORM
- onderzoek ELO integratie

Begin 2009 kunnen deze functionaliteiten door middel van een pilot worden getest. De verschillende rollen in het web gebaseerde productie platform kunnen zo aan de praktijk getoetst worden. Als de pilot positief is bevonden, kan het systeem in de loop van 2009 verder worden uitgewerkt naar aanleiding van de vraag uit de verschillende disciplines (kennis, redactioneel, development en didactiek).

Robbert Streng

September 2008

mailto:development@robbertstreng.nl

Tijdpad:

2008		2009					2010
alpha	beta	1.0	1.1	1.5	1.9	2.0	
basis integratie TYPO3 CMS & Adobe Flex3 SDK		eerste release forge.typo3.org	open standaarden, import/export features SCORM export / import	pilot	evaluatie / start 2.0	release 2.0 beta start repository	
MOD 90% mod1: Compiler module 90% mod2: DYNAMIC CODE EDITOR 80% mod3: STATIC CODEVIEW 80% mod4: CLASS/CODE GENERATOR 90% PI1: AMF SERVICE INTEGRATIE 99% PI2: FLASH SWF EMBEDDING 100%		Documentatie	Wishlist 1.1: - scorm export/import - packaging - remote compiler - prototype kickstarter				
Totaal uren							
430 uren *	200 uren *	220 uur	200 uur	100 uren			

* al gerealiseerd

recente ontwikkelingen RIA development

Al voor de eerste Ajax toepassingen, was het in elke browser die een flashplugin had mogelijk om asynchronous data uit een bestand of een database te halen tijdens de interactie met de gebruiker. Het telkens moeten herladen van content die al geladen was gaf onnodig veel bandbreedte en snelheidsverlies.

In recentere versies van Flash werden XML library's sneller en beter integreerbaar met ander toplevel functies. Toch was het parsen van content nog wel lastig omdat assets alleen via de library ingebakken konden worden in de flash movie.

Met de komst van Flex ontstonden er nieuwe mogelijkheden om primaire en secundaire content te scheiden en toch de applicatie zelf dynamisch te houden. De MXML (Adobe's Flex markup in een XML achtige structuur) zelf kan uit een Applicatie management systeem komen en alle eigenschappen en classen van die applicatie kunnen beschreven worden middels het MXML.

Pagina of applicatie?

Delen staat in dit concept op een hoog plan. In de eerste plaats ontwikkelde ik de module om learning objects te kunnen produceren op een flexibele manier waarin hergebruik bovenaan stond.

Het handig even lost komen van de gedachte webpagina. Een pagina kan in TYPO3 ook een systeem map, trashcan of een DAM (digital assets) punt zijn.

Een page id (PID) uit de TYPO3 pagetree is een verzamel punt van alle informatie die bij RIA applicatie nodig is. In het geval van de t3flex module kun je aan een FLEX applicatie de benodigde informatie onderdelen aan een unieke id koppelen.

Dit maakt het mogelijk alle informatie bij elkaar te houden en relatief te houden bij de uitwisseling van dynamische leerobjecten. Alle content, code, binary's kunnen op deze manier in hun nog altijd bewerkbare structuur worden garchiveerd en worden uitgewisseld en worden gemuteerd.

Doordat de compiler door de t3flex module wordt aangestuurd ontstaat de mogelijkheid tijdens het compilen alle informatie uit het verzamelpunt (PID) op te halen en als geïntegreerd geheel te publishen.

Het halfproduct is nu een prototype van een project in al zijn aspecten gekoppeld aan een unieke PID. Zo kan een prototype als template dienen voor specifieke projecten.

Het is mogelijk via de bestaande extensie manager in TYPO3, een applicatie

template repository op te bouwen waar onderwijs instellingen gezamenlijk generieke prototypes kunnen uitwisselen met de grootst mogelijke flexibiliteit en vrijheid ten aanzien van het eindproduct, het statische leerobject.

Het streven is deze module ELO compateble te maken door SCORM export te integreren en eindproduct met de met TYPO3 toegevoede Metadata (zie beschrijving wikipedia) de uiteindelijke implementatie op elke ELO is af te spelen.

Learning Object Metadata is a data model, usually encoded in XML, used to describe a learning object and similar digital resources used to support learning. The purpose of learning object metadata is to support the reusability of learning objects, to aid discoverability, and to facilitate their interoperability, usually in the context of online learning management systems (LMS).
(http://en.wikipedia.org/wiki/Learning_object_metadata)

Overzicht funtionaliteiten:

MOD1: COMPILER MODULE

- COMPILER [compile/swf/ant]
- REMOTE COMPILER [compile/swf/ant]
- static file overview
- comiler log file
- compiler option edit
- swf version backup
- swf result to content element records
- service-config.xml properties editor

MOD2: DYNAMIC CODE EDITOR

- code editor with highlighting MXML / XML / ACTIONSCRIPT3 / PHP (for amf3 services)

MOD3: STATIC CODE VIEW

- this module shows the output of PI3 -> markerparser

MOD4: CODE GENERATOR / WIZZARD

- wizzard for:
 - AMF services creation via SQL table forms (like ext: ???)
 - CAIRNGORM generator for domain framework and classes

MOD5: REMOTE COMPILER

- user management / custom compiler rights/options
- set your host to compile for only some IP from special IP - list

- 1) GET CODE FROM YOUR ONLINE CODE PROJECT
- 2) BUILD LOCAL with REMOTE assets
- 3) SEND THE COMPILED SWF back to the REMOTE PROJECT for EMBEDDING

PI1: AMF SERVICE REMOTE GATWAY PROXY / &mode=brower shows your AMFHP browser [for debugging]

- GATEWAY proxy for AMFPHP (amf3)
- SERVICE BROWSER on same ID

PI2: FLASH SWF EMBEDDING VIA TYPOSCRIPT

- EMBEDDING VIA:

```
lib.myFlash < FLEXOBJECT
lib.myflash.contanerStyle = width:10;height:10;border:none;
```
- TYPOSCRIPT VARS -> FLASHVARS:

```
lib.myFlash.flashvars {
    myFlashvar1 = VAR1
    myOtherFlashVar = 10
}
```
- MOVIEPARAMS to SWFOBJECT

PI3: MARKER PARSER / CODE PUBLISHER / PACKAGER (zip / flex archive)

- GET PARSED TYPOSCRIPT IN YOUR MXML / ACTIONSCRIPT CODE markers

like:

```
###TYPYSCRIPT_CONSTANT:lib.mySpecialConstant### -> just for
value
###TYPYSCRIPT_SETUP:lib.mySpecialTypoScriptPath### -> parsed
typoscript in your code!
```

LIB: some special classes:

When installing the module generates some folders:

**uploads/tx_t3flextoolkit/
files e.d.**

this path you can find attached xml config

**fileadmin/flexFiles
fileadmin/flexFiles/assets
fileadmin/flexFiles/swf
fileadmin/flexFiles/styles
fileadmin/flexFiles/src
fileadmin/flexFiles/fileBackup
fileadmin/flexFiles/lib
fileadmin/flexFiles/templates
snippets -> template
fileadmin/flexFiles/documentation
fileadmin/flexFiles/services
fileadmin/flexFiles/services/amfphp**

**general folder for al Flex related files
general shared assets
compiled flashfiles (swf)
special folder component skinning
workspace folder for al Flex source

folder for global shared SWC library's
code template .tmpl for ActionScript / MXML

Folder for project related documents
main folder for AMF3 services
service folder, put amfphp service classes here...**

Verwijzingen:

Typo3 CMS

<http://www.typo3.org/>

Typo 3 is a free, feature rich, Content Management Framework built with PHP and running under many Unixes and Windows. The complete feature list is available at typo3.com. [bron: typo3.org]

Adobe Flex

<http://flex.org/>

Flex is a framework that helps you build dynamic, interactive rich Internet applications. Flex applications are delivered on the web via the Flash Player or to the desktop via Adobe Apollo. You can build applications using the free Flex SDK or Flex Builder, an Eclipse based IDE. You can also try Flex online without downloading anything. [bron: flex.org]

LO's

Learning objects

http://en.wikipedia.org/wiki/Learning_Object