



Universitat de les Illes Balears

Departament de Ciències Matemàtiques i Informàtica

Màster en tecnologies de la informació i les comunicacions

MEMÒRIA DE TREBALL DE FI DE MÀSTER

*Desenvolupament d'un connector de Drupal
per afegir capacitats d'agregació de continguts
i adaptabilitat del disseny de fragments de
contingut*

Data: 20 de juny de 2011

Autor,

Espina de la Portilla, Joan

Director/s,

Juiz García, Carlos

Desenvolupament d'un connector de Drupal per afegir capacitats d'agregació de continguts i adaptabilitat del disseny de fragments de contingut

Resum

Els agregadors de contingut han fet variar el mètode de funcionament dels servidors cau. Abans els servidors cau emmagatzemaven pàgines completes, en canvi, des de l'aparició dels agregadors de contingut i la major interacció per part dels usuaris en les aplicacions web, els servidors cau han passat a emmagatzemar fragments d'aquestes pàgines. Moltes pàgines han passat d'oferir un contingut estàtic i poc personalitzable a oferir un contingut molt dinàmic i altament personalitzable per l'usuari. Aquesta personalització pot venir de permetre a l'usuari seleccionar diferents fonts d'origen de la informació. De igual manera es permet a l'usuari l'agrupació de les diferents fonts d'informació (agregació de continguts). Aquest treball s'ha centrat en la selecció i adaptació d'un programari que permeti la gestió de fonts d'informació i així com la gestió de la seva agrupació. També s'ha treballat en la definició de les interfícies necessàries per interactuar amb una eina de millora de les agrupacions de contingut. A partir de la recol·lecció de mètriques dels continguts, l'eina de millora, calcula l'agrupació més eficient dels continguts (adaptabilitat del disseny dels fragments de contingut). D'aquesta manera el servidor cau gestiona més eficientment els fragments que emmagatzema dels diferents usuaris.

Paraules clau:

servidor cau, agregació de continguts, sistema de gestió de continguts, Drupal, PHP, connector

1. Introducció

Als darrers anys s'ha produït un canvi en els mètodes d'utilització de les aplicacions web. Aquest nou model d'utilització s'ha donat a conèixer amb el nom de Web 2.0. L'aparició de noves tècniques i noves eines ha comportat un canvi en la utilització de les aplicacions web. Els usuaris han passat de ser només consumidors d'informació (lectors) a ser membres actius en la creació i recopilació d'informació i continguts (productors).

Un tipus particular d'aplicacions Web 2.0 són les anomenades: agregadors de continguts. Aquestes aplicacions es basen en la creació de pàgines web mitjançant la recuperació i agregació d'informació. Aquesta informació es pot obtenir de diferents maneres i de diferents orígens. Per exemple: es pot obtenir a partir de fonts remotes d'informació a través de serveis específics (RSS, *Webs Services*,...), amb consultes a bases de dades o també a través de petites aplicacions o *widgets* que generen continguts. Les pàgines creades amb agregació de continguts poden ser també públiques o privades, d'ús individual o d'ús compartit. Segons el tipus de pàgines existeixen també diferents tipus d'aplicacions: blogs personals o institucionals (públics i d'ús general), pàgines d'inici personalitzades (privades i d'ús particular), plataformes de publicació de pàgines personalitzades (*personalized dashboard publishing platforms*) entre d'altres. A totes elles, les condicions, les formes d'utilització i les seves característiques canvien respecte a les aplicacions web tradicionals.

Els treballs enfocats a millorar el rendiment de les aplicacions web tradicionals són abundants. Però la majoria de tècniques utilitzades amb anterioritat no són aplicables, o no ofereixen gran millora dels resultats, quan s'apliquen a aquests nous tipus d'aplicacions. Entre les tècniques utilitzades es pot destacar la utilització de servidors cau de documents web.

Les aplicacions de servidors caus basen el seu funcionament en la reutilització de peticions web. Quan es sol·licita una pàgina per primera vegada, aquesta s'emmagatzema temporalment al servidor cau, i les següents peticions de la mateixa pàgina són contestades utilitzant les còpies temporals del servidor cau. D'aquesta manera es redueix la càrrega de peticions sobre el servidor web i disminueix el temps de resposta, ja que recuperar una pàgina d'un servidor cau és molt més ràpid que haver de generar els continguts una altra vegada al servidor web. Aquestes còpies temporals del servidor cau romanen vàlides fins que algun contingut es veu alterat al servidor web d'origen. Però mentre la pàgina no ha variat és molt probable que altres usuaris, o fins i tot el primer usuari, sol·licitin la mateixa pàgina. Llavors el servidor cau mostrarà la pàgina emmagatzemada localment sense necessitat de sol·licitar la pàgina, cada vegada, al servidor web.

Però en un entorn d'agregació de continguts existeix el problema de la freqüència d'actualització del contingut. Aquesta freqüència és molt més elevada que a una aplicació web tradicional. De la mateixa manera, el grau de personalització de pàgines és més elevat. Això provoca que la probabilitat sigui menor de que dos usuaris sol·licitin la mateixa pàgina (a causa d'un alt grau de personalització) i també que un mateix usuari sol·liciti dues vegades la mateixa pàgina abans que aquesta hagi canviat (major freqüència d'actualització).

Una forma d'arreglar aquest problema és la reducció de la unitat mínima que s'emmagatzema al servidor cau. En lloc de que el servidor cau gestioni pàgines completes es configura per que el servidor cau sigui capaç de gestionar fragments d'aquestes pàgines. D'aquesta manera s'aconsegueix un grau de personalització i unes freqüències d'actualització que afavoreixen el rendiment del servidor cau. El problema és que si creix molt el número de fragments es pot obtenir un temps de sobrecarrega (pel procés d'unió i el gran número de connexions per crear una sola pàgina) major a l'estalvi de temps proporcionat pel servidor cau. Per tant, s'ha de cercar l'equilibri entre el número de fragments del contingut original per així mantenir un bon rendiment del servidor cau i que no suposi massa sobrecarrega pel sistema.

A la Figura 1 es pot veure un exemple de 3 pàgines (P1, P2 i P3) que comparteixen part dels continguts. Això podria representar 3 configuracions diferents de continguts agrupats per 3 usuaris de manera diferent. Llavors, encara que hi ha informació repetida, el servidor cau tradicional ho tracta com 3 pàgines diferents. És aquí on es veu la necessitat de fragmentar els elements que gestionarà el servidor cau. Els continguts dels diferents usuaris es divideixen en fragments. Ara el servidor cau gestiona fragments de pàgines enlloc de pàgines completes. D'aquesta manera s'augmenta la possibilitat de que diferents usuaris sol·licitin el mateix fragment i així reduir les peticions al servidor web d'origen.



Figura 1 : Pàgines i fragments

A les aplicacions d'agregació de continguts es pot associar el fragment mínim de pàgina a cada un dels elements de contingut agregats. A partir d'aquesta màxima fragmentació es poden unir elements fins arribar al punt adequat d'unions. Però, com determinar el punt adequat d'unions? Aquest és el problema que es vol resoldre.

2. Treball previ

Trobar el millor número de fragments en que s'han d'agrupar els elements de contingut és el treball d'investigació del doctorant Carlos Guerrero. Per tal de calcular el número de fragments ha creat un sistema autoadaptable, que mitjançant la monitorització dels paràmetres de caracterització dels distints elements de continguts (mida, freqüència d'actualització, freqüència de peticions, número d'elements pare i número d'elements fill) determina quin és el millor disseny de fragments d'una pàgina. El disseny de fragments indica quins elements de contingut es tornaran a l'usuari units al servidor i quins no. D'aquesta manera es determinarà els conjunts d'elements podrà gestionar de manera independent el servidor cau.

Aquest Treball de Fi de Master parteix de la necessitat de tenir un sistema per gestionar les agregacions de continguts, guardar informació sobre la caracterització dels elements de contingut (freqüències, mida,...), de interactuar amb el sistema autoadaptable que determina els estats de les relacions entre els fragments i finalment ha de ser capaç de dialogar amb el servidor cau per l'intercanvi de fragments de contingut per mitjà del llenguatge de marcat utilitzat pels servidors cau: *Edge Side Include* (ESI).

Aquest projecte vol cercar, a partir d'una eina ja existent, la manera de resoldre les necessitats plantejades al paràgraf anterior. Una altra premissa que s'exigeix a l'eina és que sigui de programari lliure, no tant pel cost del programari com per la possibilitat d'estudiar, i adaptar en cas necessari, el seu codi font.

Segons l'article de Challenger J. et al. (2005) una solució per emmagatzemar la informació dels fragments és utilitzar una estructura de graf d'objectes amb dependència, que realment es tradueix en un graf acíclic dirigit, en definitiva una estructura d'arbre. On els fragments de contingut s'organitzen de forma jeràrquica, de manera que els nodes de l'arbre representen els elements de contingut i les arestes representen les relacions d'agregació entre els diferents nodes.

A la Figura 2 es pot veure un exemple de la representació de diferents elements de continguts utilitzats per diferents usuaris. Es poden veure 3 nodes arrel, que representen 3 pàgines diferents. Cada node representa algun element de contingut. I les agrupacions de diferents elements de contingut donen lloc als fragments, que és l'element que gestiona el servidor cau. El servidor cau modifica la unitat bàsica de gestió: passa d'emmagatzemar pàgines completes a emmagatzemar fragments de pàgines-

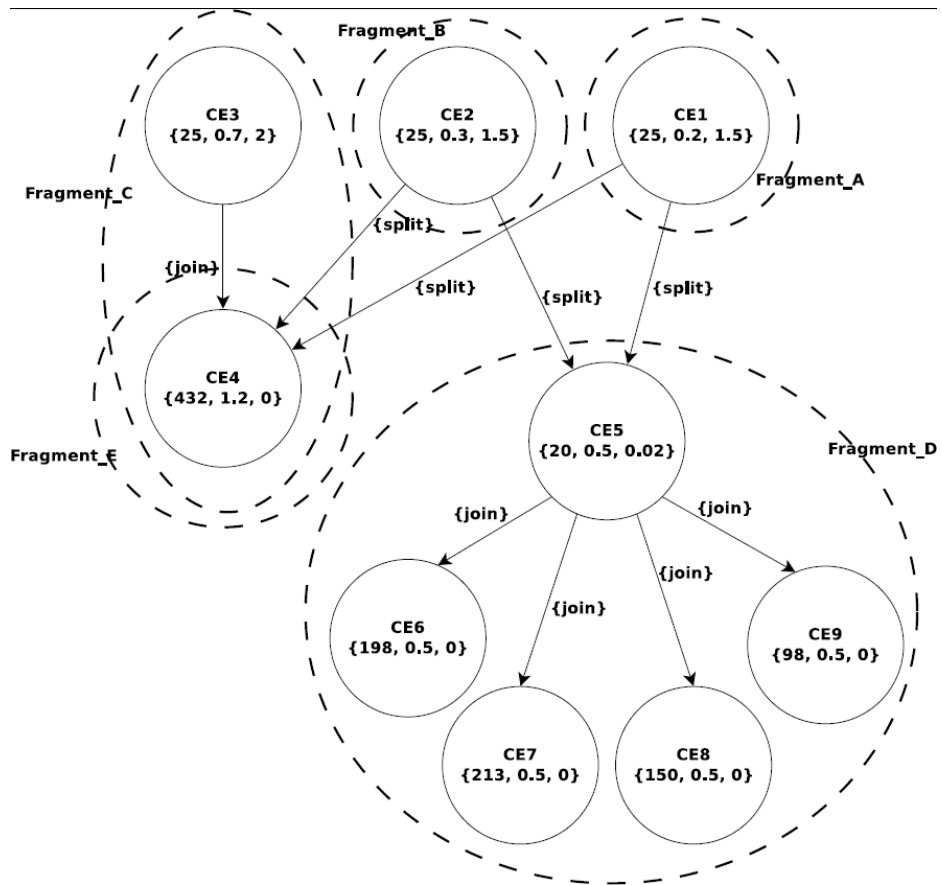


Figura 2 : Estructura d'arbre a partir de les pàgines, fragments i elements de contingut.

3. Descripció

En aquest punt es detallen les diferents tasques realitzades al projecte juntament amb les diferents funcionalitats de l'eina desenvolupada.

3.1. Selecció entorn desenvolupament

Per tal de seleccionar l'entorn que millor s'adapti a les necessitats primer es comparen diferents tipus d'aplicacions que facilitin les agrupacions de continguts.

3.1.1. Estudi entorns existents

A la actualitat existeixen molts gestors de continguts que permeten la lectura de fonts de notícies integrades dins el propi gestor. Per tal de seleccionar el programari més adient per l'aplicació a desenvolupar s'han comparat diversos programes o entorns que permeten l'agrupació de continguts.

Primerament s'han comparat alguns entorns en línia que permeten la personalització de les pàgines per a cada usuari i permeten recuperar continguts externs per mostrar-los tots al mateix entorn. S'han comparat les funcionalitats de llocs web com *iGoogle* o *inicio.es*. Aquests dos llocs web permeten la personalització de les pàgines d'inici dels usuaris, definint objectes que mostren informació obtinguda d'altres llocs web, com per exemple el temps, titulars de premsa, resultats esportius, etc. Aquesta part sí que interessa per l'aplicació, però s'ha trobat una dificultat, no és té accés al codi font, i la possibilitat de programar és molt reduïda. Disposen d'APIs de programació però amb poques funcionalitats i amb la limitació que no és té mai accés a la resta de codi del servidor.

És en aquest moment quan es decideix explorar la utilització de gestors de contingut que es puguin instal·lar i modificar segons les pròpies necessitats. Es parteix dels 3 gestors de continguts més utilitzats i més estesos en el moment de l'inici del treball: *Wordpress*, *Drupal* i *Joomla*.

Aquests 3 entorns seleccionats en primera instància, compleixen els 2 requisits definits a l'apartat 2 de la memòria: són eines ja existents i són de codi lliure.

3.1.2. Selecció entorn desenvolupament

Per tal de seleccionar l'entorn que millor s'adapta a les necessitats s'estudien les característiques dels 3 entorns.

Els 3 comparteixen una sèrie de propietats que seran molt útils a l'hora de fer feina:

- programari lliure: no tant pel fet que el seu cost sigui gratuït, sinó perquè així es té accés al codi font i es pot modificar en cas que sigui necessari.
- desenvolupats en PHP: tant pel fet de facilitat de programació com pels requeriments en cas d'instal·lar-ho a un servidor. En cas d'instal·lar-ho a un servidor contractat no és gaire complicat trobar servidors que suportin PHP.
- importància i presència al mercat: encara que no sempre el més estès és el millor, el fet que hi hagi tota una comunitat d'usuaris ens assegura que hi ha tota una comunitat de desenvolupadors i tota una col·lecció de productes ja desenvolupats, que ens poden servir o ajudar al nostre desenvolupament.

Disseny i complements

Per configurar el disseny de les pantalles els 3 entorns utilitzen els Temes, que permeten l'assignació en conjunt de tots els paràmetres necessaris de disseny (colors, tipus i mides de lletres, normalment a partir d'estils CSS).

Els 3 entorns també disposen d'una col·lecció extensa de complements que permeten afegir funcionalitats.

Existeixen molts complements (més de 6.000) i dissenys gratuïts encara que també hi ha disponibles complements de pagament més orientats a l'entorn professional que a l'entorn domèstic.

Usabilitat i documentació

Encara que un usuari amb pocs coneixements de programació és capaç de crear un lloc web amb qualsevol dels 3 entorns la realitat és que no són els 3 iguals.

WordPress permet la posada en marxa molt ràpida d'un lloc web, encara que la seva naturalesa com gestor de blogs pot limitar o dificultar la seva utilització com a gestor de contingut. Modificar els temes és una tasca senzilla, per això moltes vegades blogs de *WordPress* amb diferents temes o plantilles presenten un aspecte molt diferent.

Drupal presenta més dificultat a l'hora d'instal·lar i també és més complexa la presentació de la informació, però en canvi ofereix gran flexibilitat a l'hora de desenvolupar nous complements.

Joomla és més rígid tant amb la gestió del contingut com amb la presentació de la informació, per això els llocs gestionats amb Joomla presenten normalment aspectes molt semblants.

Pel que fa a la documentació els 3 entorns disposen de documentació en anglès al seu lloc web, encara que hi ha disponibles molts manuals i ajudes traduïts. També disposen de fòrums on les comunitats d'usuaris i desenvolupadors plantegen dubtes i la seva resolució.

Conclusions i selecció

Els 3 entorns ofereixen una sèrie d'avantatges que finalment es poden resumir de la següent manera:

- *Wordpress*: permet posar en marxa un lloc web d'una manera ràpida, fàcil d'actualitzar amb un bon disseny.
- *Drupal*: molt assenyalat si es vol fer algun desenvolupament addicional, ja que permet afegir algun mòdul ja existent o programar mòduls que s'integrin amb les funcionalitats del propi entorn.
- *Joomla*: molt eficient per gestionar gran quantitat d'informació sempre que aquesta es trobi organitzada de forma jeràrquica .

En el cas d'aquest treball s'ha optat per la utilització de l'entorn *Drupal*, que permet el desenvolupament de mòduls addicionals que s'integrin amb el contingut i funcionalitats del propi gestor.

3.2. Estàndards utilitzats

Una vegada seleccionat l'entorn de desenvolupament, aquest ha condicionat la utilització de les estructures i nomenclatures de variables, funcions, arxius i directoris marcats per l'entorn. Això es tradueix en que les funcions programades s'han d'adaptar a les pautes utilitzades en els mòduls de *Drupal*, principalment les funcions que s'integren amb nucli del programa i amb els mòduls ja existents. Els noms dels arxius i dels directoris també ha de seguir les normes definides. Com a la majoria de llenguatges de programació, a part de les normes de sintaxis definides, existeixen tota una sèrie de recomanacions que també s'han utilitzat. Per exemple en els noms dels arxius, per diferenciar quins són de funcions dels usuaris i quins contenen funcions per l'usuari administrador, l'arxiu que inclou totes les funcions que només pot realitzar l'administrador conté 'admin' en el seu nom. En el cas de les variables, les úniques que necessiten adaptar-se a la sintaxi de *Drupal* són les que reben els paràmetres dels menús de navegació, que normalment correspon a algun element de la URL.

Un altre element que s'ha seleccionat per tal que s'adapti a algun estàndard correspon al llenguatge utilitzat per definir l'estructura de les pàgines de cada usuari. Aquesta estructura en forma d'arbre permet l'intercanvi d'informació amb el sistema autoadaptable. Ja que un arbre es pot tractar com un graf dirigit s'ha cercat un llenguatge estàndard de definició de grafs. A partir de la descripció i comparativa realitzada per Rodriguez (2002), on es centra en la comparativa de 3 esquemes de definició de grafs (*Heidi*, *GraphML* i *XGMML*) i finalment selecciona (segons la justificació d'avantatges i una mica de preferència personal) l'esquema

GraphML com a millor llenguatge XML per definir grafs, s'ha decidit utilitzar aquest esquema per realitzar els intercanvis d'informació necessaris entre el mòdul de *Drupal* i el sistema autoadaptable.

A la Figura 3 es pot veure un exemple simple de la descripció d'un graf. Es pot observar la definició d'un graf amb 6 nodes i 5 arestes.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This file was written by the JAVA GraphML Library.-->
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="directed">
    <node id="n0"/>
    <node id="n1"/>
    <node id="n2"/>
    <node id="n3"/>
    <node id="n4"/>
    <node id="n5"/>
    <edge source="n0" target="n2"/>
    <edge source="n1" target="n2"/>
    <edge source="n2" target="n3"/>
    <edge source="n3" target="n5"/>
    <edge source="n3" target="n4"/>
  </graph>
</graphml>
```

Figura 3 : Exemple de graf utilitzant *GraphML*

L'esquema de definició *GraphML* permet afegir els elements addicionals que possibiliten intercanviar la informació necessària i no estàndard entre el mòdul de *Drupal* i el càlcul de la seva agrupació. Es poden definir atributs i claus addicionals segons les necessitats de cada graf. Al cas d'aquest treball s'han definit les següents claus:

- pels nodes: número de pares, número de fills, mida (especificada en bytes), freqüència de consulta i freqüència d'actualització.
- per les arestes: una única clau que defineix si el node fill es retorna juntament amb el pare o bé es retorna per separat.

3.3. Esquema de dades i comunicacions

A continuació es descriu el model de dades (Figura 4) definit al desenvolupament del treball. Per no interferir amb les taules dels propis continguts del *Drupal* s'ha decidit crear l'estructura necessària dins *Drupal* per a la gestió de la informació necessària.

La instal·lació de les darreres versions de *Drupal* permet la utilització de diferents gestors de bases de dades, per això l'entorn proporciona una API per programar els mòduls amb independència del gestor utilitzat. En el cas del treball s'ha utilitzat *MySQL*, però s'ha fet utilitzant l'API de bases de dades de *Drupal* que possibilita la portabilitat a un altre gestor.

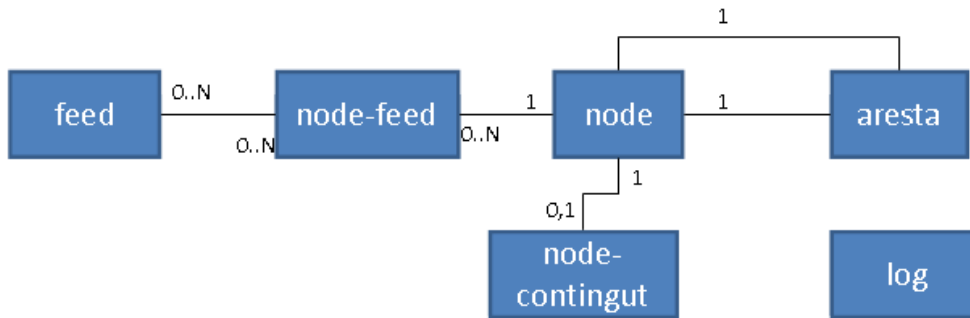


Figura 4 : Model de dades

A continuació es detalla la informació que conté cada taula del model de la Figura 4: la taula *feed* emmagatzema la definició de les fonts de notícies o elements de contingut, la taula *node* emmagatzema la informació del node, per permetre agrupar varies fonts de notícies a un mateix node. A la taula *node-feed* hi ha les relacions entre les diferents fonts i els diferents nodes, la taula *aresta* permet crear l'estructura de graf dirigit (o arbre) definint les relacions entre els diferents nodes, la taula *node-contingut* s'utilitza de memòria cau local, per tal de poder comprovar el contingut de la consulta anterior i saber si hi ha el mateix contingut o bé aquest s'ha modificat.

I finalment, sense relació directe amb les taules anteriors, s'ha afegit la taula *log* que actuarà de registre de tota l'activitat realitzada. També servirà per obtenir les dades estadístiques de freqüència consultes i freqüència d'actualitzacions.

Durant l'execució normal del mòdul desenvolupat hi ha altres comunicacions entre *Drupal* i el sistema autoadaptable. Cada vegada que hi ha una modificació dels grafs dels usuaris el mòdul ho ha de notificar al sistema autoadaptable per tal que aquest actualitzi els càlculs dels estats de les arestes. D'altra banda el sistema autoadaptable ha de poder modificar els estats de les arestes emmagatzemats a la taula *aresta* una vegada realitzats els càlculs pertinents.

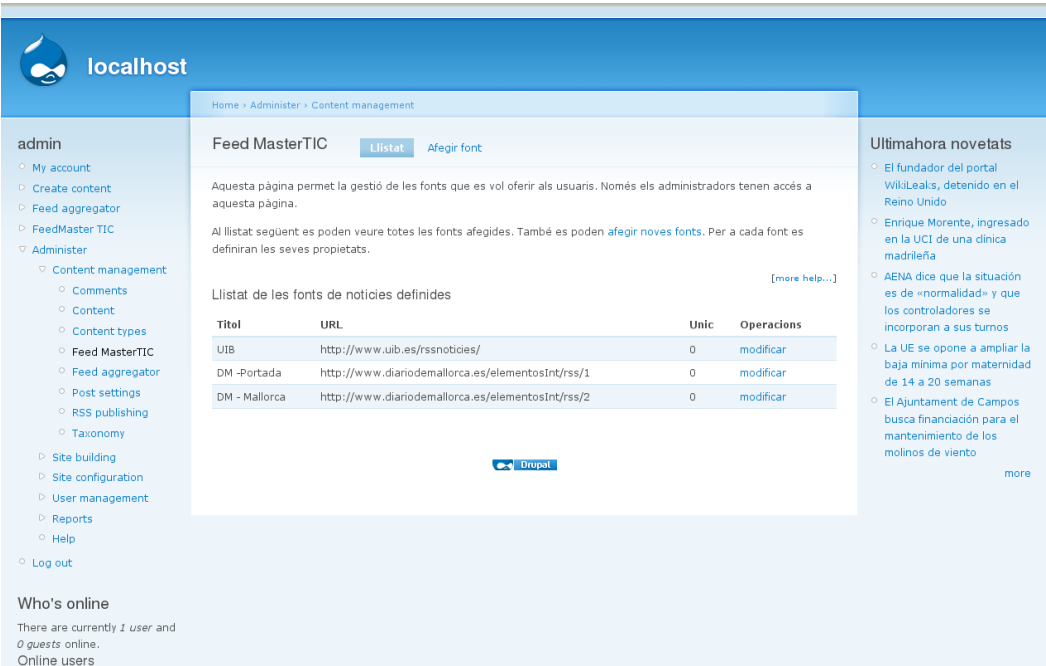
Als apartats següents es descriu la interfície d'usuari i la interfície definida que permet la comunicació entre el mòdul de *Drupal* i el sistema autoadaptable.

3.4. Interfície d'usuari

A continuació es presenten les diferents accions que pot realitzar un usuari de forma interactiva amb l'aplicació.

3.4.1. Gestió de les fonts

Aquesta és una tasca reservada per l'usuari administrador. És l'únic usuari que pot definir fonts d'informació. Com es pot veure a la Figura 5, aquesta pàgina es troba dins les opcions reservades a l'administrador i baix el menú Gestió de continguts.



The screenshot shows the Drupal administration interface for 'localhost'. The main content area is titled 'Feed MasterTIC' and includes a 'Llistat' button and an 'Afegir font' button. Below this, there is a text block explaining the page's purpose and a table of defined news feeds. The table has the following data:

Títol	URL	Unic	Operacions
UIB	http://www.uib.es/rssnoticies/	0	modificar
DM -Portada	http://www.diariodemallorca.es/elementosInt/rss/1	0	modificar
DM - Mallorca	http://www.diariodemallorca.es/elementosInt/rss/2	0	modificar

The sidebar on the left shows the 'admin' menu with 'Content management' expanded. The sidebar on the right shows 'Ultimahora novetats' with a list of news items.

Figura 5 : Administració de fonts

Les fonts d'informació poden ser de dos tipus: fonts de notícies (en format RSS), o bé continguts d'altres llocs web. Aquests darrers poden ser tant continguts del propi servidor de contingut o també continguts allotjats a un servidor extern. La casella de verificació "Font de notícies" permet diferenciar amb quin format es presenta la informació a l'usuari. En el cas que la font sigui una font de notícies RSS s'extreu el titular de cada node d'informació i en el cas que la font sigui un altre contingut no es dona cap tipus de format a la sortida. A la Figura 6 es pot veure la informació necessària que s'introdueix quan s'afegeix una font d'informació.

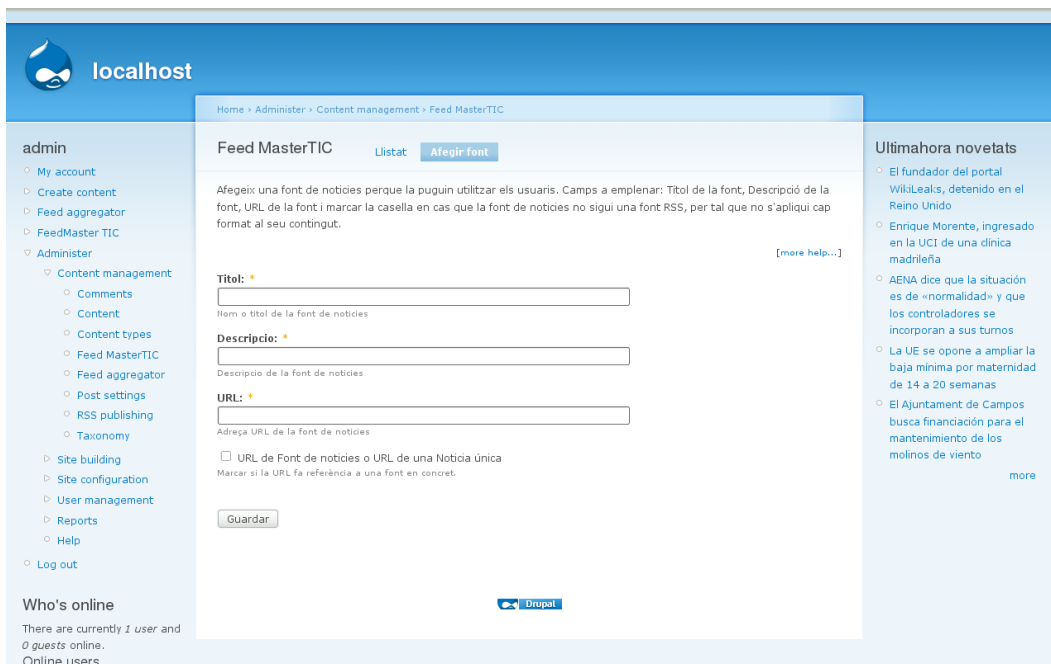


Figura 6 : Afegir nova font

3.4.2. Gestió dels nodes

Una vegada l'administrador ha definit les fonts de notícies llavors cada usuari les pot agrupar o agregar al seu gust. Al menú Feedmaster TIC els usuaris poden veure els nodes ja existents al sistema i crear nous nodes amb els elements de contingut desitjats.

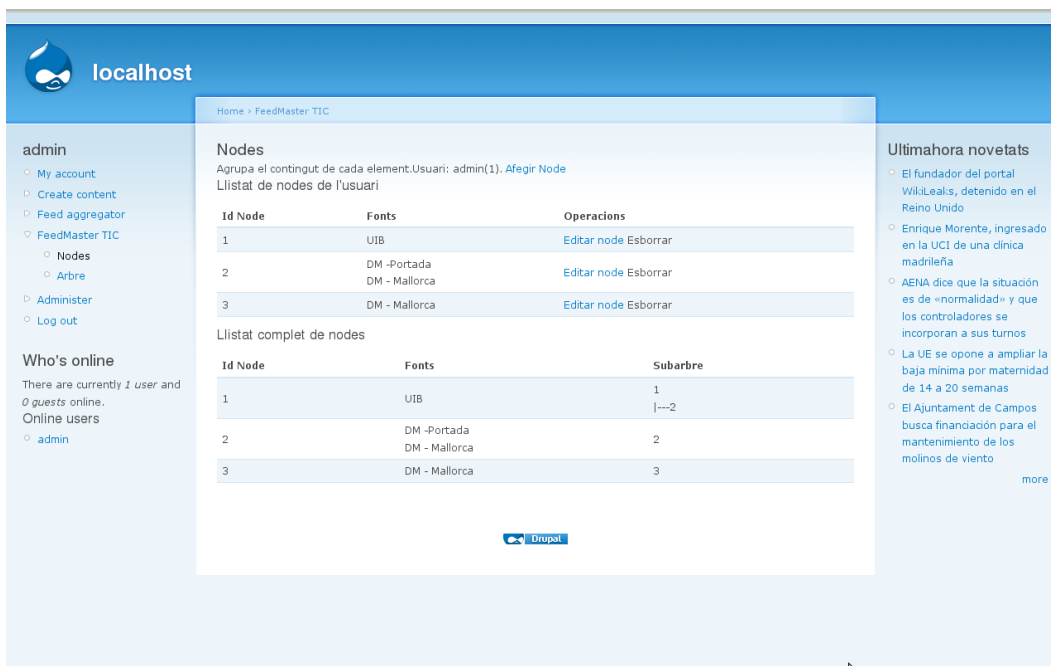


Figura 7 : Veure nodes

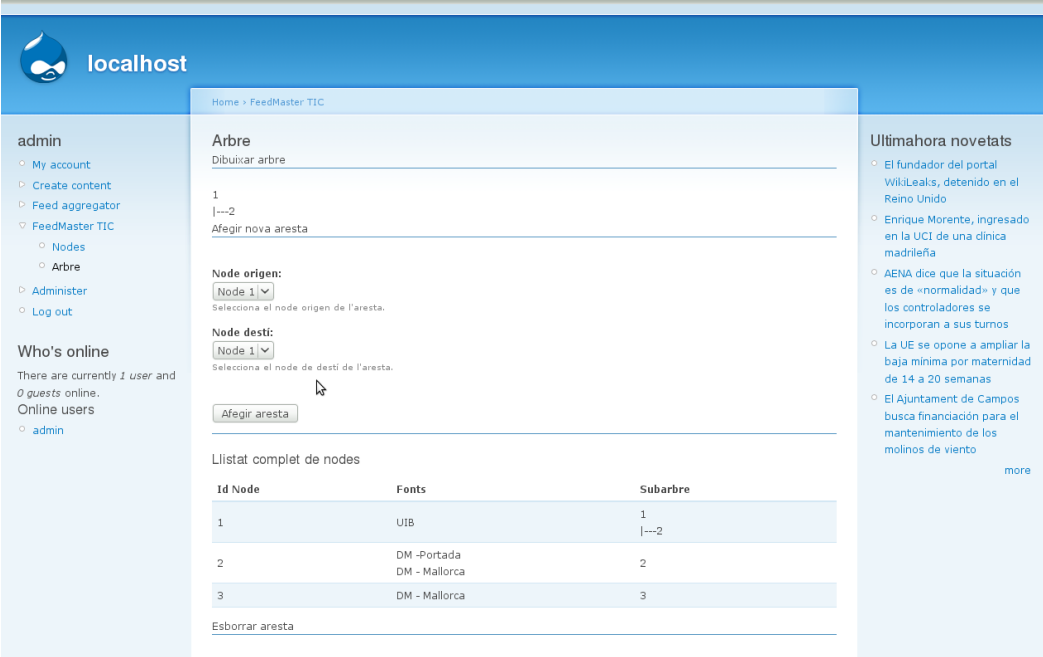
A la mateixa pàgina de gestió de nodes (veure Figura 7) l'usuari pot veure en primera instància els nodes del qual ell és propietari i la part inferior pot veure tots els nodes que hi ha definits al sistema, així en cas de trobar un node amb el contingut pertinent ja no és necessari definir un nou node i l'usuari pot decidir compartir el node d'un altre usuari.

3.4.3. Gestió dels arbres o pàgines

En aquesta pàgina del sistema es possibilita a l'usuari definir les agrupacions dels continguts de cada node. Cada bloc d'agrupacions es pot associar a una pàgina diferent del propi usuari o bé d'usuaris diferents. Aquí l'usuari pot seleccionar l'origen i destí per definir una relació entre dos nodes. Aquesta relació d'agregació correspon amb una aresta de l'arbre.

En aquesta opció de menú l'usuari pot realitzar les accions següents: afegir una aresta, esborrar una aresta, consultar l'arbre personal i veure un llistat complet dels nodes definits al sistema.

A la Figura 8 es mostra com veu l'usuari aquesta informació i com la modifica. A la part superior es mostra l'arbre o arbres de cada usuari. A continuació es mostren les opcions per afegir noves arestes, seleccionant el node d'origen i node destí. Per ajudar a aquesta tasca, a continuació, es mostren tots els nodes definits al sistema, juntament amb els seus subarbres. Això redueix la possibilitat de crear cicles al arbre. El mòdul no ho permet ja que controla els cicles dins el graf, però amb aquesta visualització del subarbre de cada node s'evita el missatge d'error. Finalment a la mateixa pàgina hi ha les opcions necessàries per esborrar arestes.



The screenshot shows a web interface for managing a tree structure. The main content area is titled 'Arbre' and includes a 'Dibuixar arbre' section with a diagram showing a node '1' connected to a sub-tree '|--2'. Below this is a form to 'Afegir nova aresta' with dropdown menus for 'Node origen' and 'Node destí', both currently set to 'Node 1'. A table below lists all nodes in the system:

Id Node	Fonts	Subarbre
1	UIB	1 --2
2	DM - Portada DM - Mallorca	2
3	DM - Mallorca	3

At the bottom of the table, there is an 'Esborrar aresta' button. The interface also features a sidebar with user navigation options and a 'Who's online' section.

Figura 8 : Manteniment de l'arbre

3.4.4. Visualització dels continguts

També s'ha definit una opció al menú on cada usuari pot veure els continguts de tots els nodes. L'aplicació consulta cada una de les fonts definides i agrupa els continguts segons l'estructura d'arbre dels nodes, per després donar format al contingut de cada element i mostrar-ho dins l'espai reservat als continguts de *Drupal*.

En cas de que l'usuari hagi definit diferents subarbres per l'agrupació dels continguts, aquests subarbres es mostren com pàgines diferents. A la Figura 9 es pot veure un cas en que l'usuari ha definit només un subarbre. Només apareix una pàgina (Pàgina 1) a la part superior.

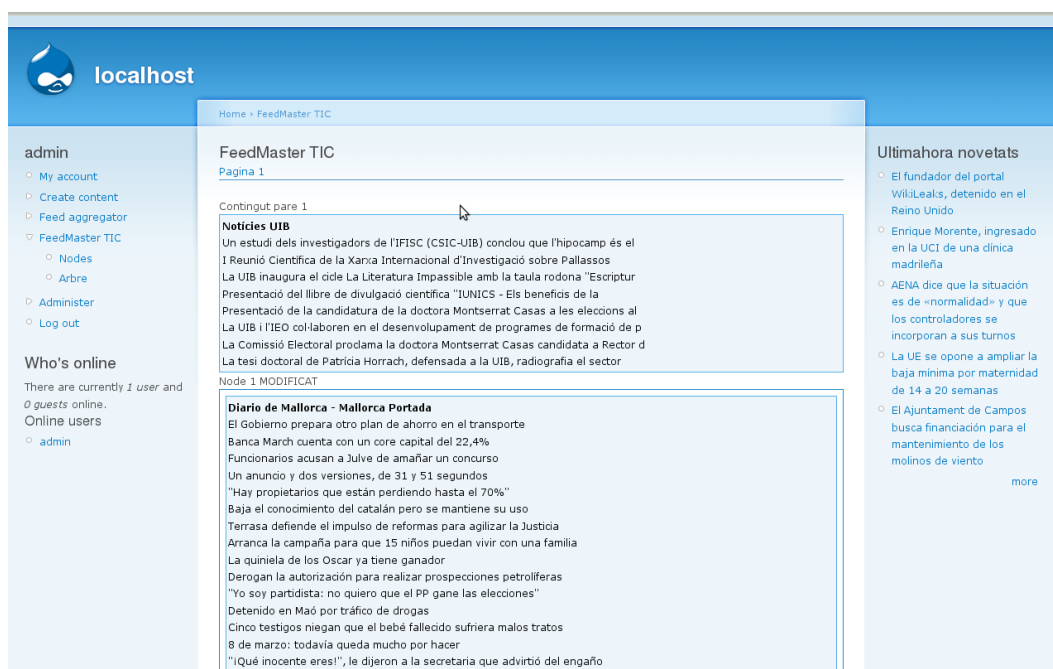


Figura 9 : Veure continguts

3.5. Interfície entre aplicacions

Una altra part del projecte és permetre l'intercanvi d'informació entre el mòdul de *Drupal* i el sistema autoadaptable. Per poder realitzar aquest intercanvi d'informació s'han definit 2 mòduls en PHP (veure apartats 3.5.1 i 3.5.2) que interactuen amb el mòdul de *Drupal* per tal de poder accedir a la informació emmagatzemada que necessita el sistema autoadaptable per calcular els diferents estats. D'altre banda s'ha definit la funció que ha d'interactuar amb el servidor cau per obtenir els continguts dels diferents fragments sol·licitats (veure apartat 3.5.3).

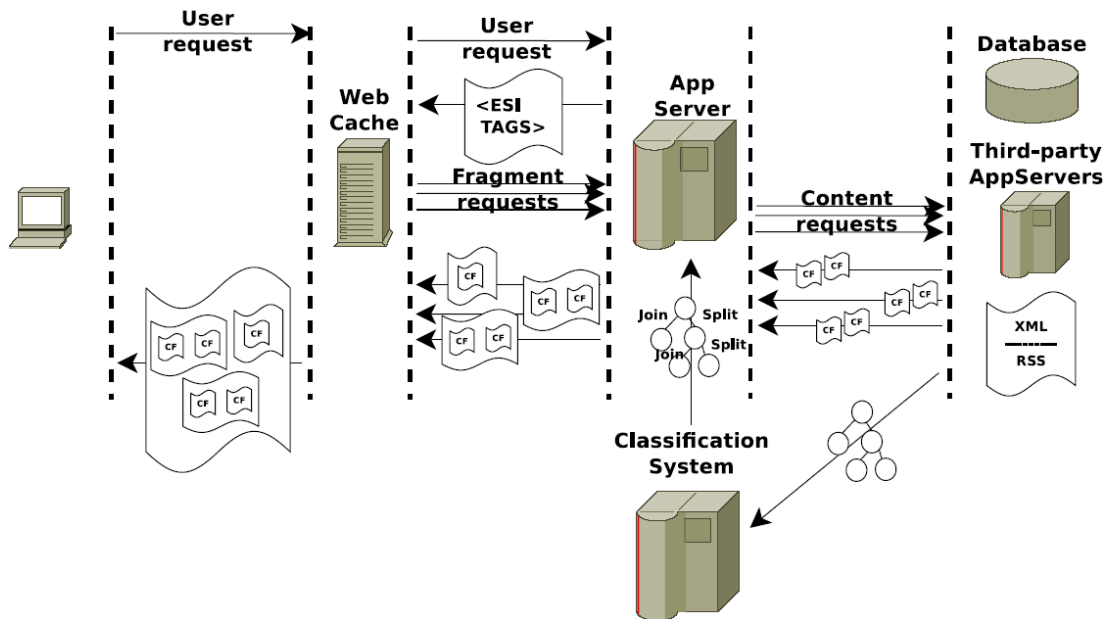


Figura 10 : Esquema servidor cau

A l'esquema anterior de la Figura 10, es pot veure on es situen tant el mòdul de *Drupal* (App server a la figura) i el sistema autoadaptable (Classification System a la figura). Aquests dos sistemes necessiten intercanviar informació per tal d'agrupar els fragments d'informació de manera més eficient. Una petició per part d'un client d'un document al servidor cau genera les peticions dels diferents fragments al mòdul de *Drupal*. Aquest darrer pot retornar o bé continguts, obtinguts directament dels proveïdors de contingut o bé etiquetes ESI que interpreta el servidor cau. Aquestes etiquetes poden donar objecte a noves peticions al mòdul de *Drupal*, en cas que el servidor cau no tingui el contingut actualitzat. Finalment el servidor cau unirà els fragments ja disponibles al propi servidor amb els obtinguts de nou des de l'origen del contingut, per tornar-ho tot al client.

A la Figura 11 es pot veure un cronograma d'un exemple de interacció entre el mòdul de *Drupal* i el sistema autoadaptable. En el moment que hi ha qualche canvi en el model de dades (s'afegeix una aresta, s'elimina una aresta) el mòdul de *Drupal* avisa al sistema autoadaptable per tal que aquest generi una petició de canvis realitzats al model. En aquest moment es genera un arxiu XML amb format *GraphML* per traslladar la informació que necessita el sistema autoadaptable per calcular els nous estats de les arestes. A continuació el mòdul de *Drupal* reb, també en format XML, els nous estats i els aplica al model.

S'ha definit aquesta estructura de comunicacions per permetre, en cas de necessitat, variacions de les comunicacions sense necessitat de definir noves funcions. Per exemple, es pot definir un càlcul periòdic dels estats, simplement generant les peticions de l'estructura de forma periòdica. En qualsevol cas així es possibilita que qualsevol de les dues aplicacions pugui iniciar la sol·licitud del càlcul dels estats.

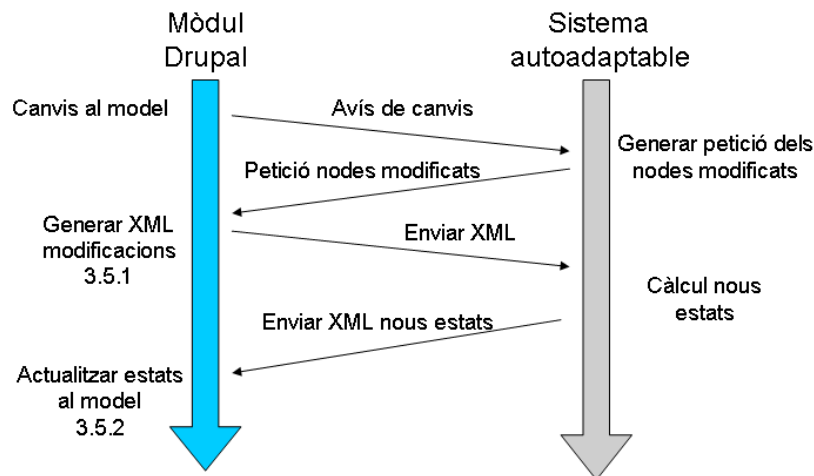


Figura 11 : Cronograma de les comunicacions

3.5.1. *Obtenir estat del graf*

Aquest mòdul respon a la petició de nodes modificats del sistema autoadaptable per sol·licitar al mòdul de *Drupal* l'estat de l'arbre o d'un subarbre.

La funció definida genera un arxiu XML amb la informació corresponent. Per sol·licitar la caracterització d'una sèrie de nodes s'utilitzen els següents paràmetres:

- Node: s'utilitza per sol·licitar la informació de caracterització d'un determinat node o bé l'arbre complet.
- Tipus: les freqüències de consulta i les freqüències d'actualització es poden calcular de 2 maneres: per temps o per número de peticions.
- Quantitat: aquest paràmetre defineix la quantitat de temps o de peticions que s'han de tenir en compte per calcular les freqüències anteriors.

3.5.2. Modificar estat del graf

Una vegada calculats els estats de cada aresta, a través d'aquesta funció, el mòdul de *Drupal* processa l'arxiu XML amb els nous estats i els trasllada al model. En aquest cas no hi ha paràmetres que permetin diferents possibilitats. Simplement a partir de l'arxiu XML calculat pel sistema autoadaptable s'actualitzen el estats de les arestes a la base de dades. L'arxiu XML ha de contenir la informació de caracterització de les arestes.

3.5.3. Obtenir el contingut d'una pàgina

Finalment s'ha definit una funció per mostrar el contingut de cada node a l'usuari. Aquesta funció és la que, a partir dels estats de cada aresta, obtén el contingut directament o bé genera l'etiqueta ESI per tal que la interpreti la memòria cau. De fet, l'usuari real d'aquest mòdul és la memòria cau.

4. Resultats

Una vegada desenvolupat el mòdul de *Drupal* s'ha procedit a la instal·lació en un entorn de proves. Per tal de poder confirmar el correcte funcionament del mòdul s'ha efectuat la seva avaluació per parts. S'han realitzat les següents comprovacions:

- Interfície d'usuari
- Recuperació de la informació de caracterització per part del sistema autoadaptable
- Actualització de la informació al mòdul de *Drupal*
- Recuperació de continguts – memòria cau local
- Recuperació de continguts per part del servidor cau

Primer s'ha avaluat tota l'interfície d'usuari, on s'ha pogut confirmar el correcte funcionament dels diferents formularis que permeten la interacció per part de l'usuari per tal de decidir les seves pròpies agrupacions de continguts. S'ha confirmat que les dades introduïdes als diferents formularis es guarden correctament dins la base de dades. De la mateixa manera s'ha comprovat que la informació d'agrupació emmagatzemada a la base de dades es recupera i es mostra correctament a les pàgines dels usuaris.

La següent part avaluada ha sigut l'intercanvi d'informació amb el sistema autoadaptable, tant l'exportació de la informació de caracterització com l'actualització de la informació a la base de dades de *Drupal* una vegada realitzats els càlculs al sistema autoadaptable. S'han realitzat els diferents tipus d'exportació en funció dels paràmetres de consulta (node, tipus, quantitat) i s'ha confirmat que es generen correctament els arxius en format *GraphML* amb la informació de caracterització corresponent. De la mateixa manera, per actualitzar la informació, s'ha comprovat que el mòdul de *Drupal* també processa correctament l'arxiu XML generat pel sistema adaptable amb els nous estats. S'ha llegit correctament la informació i s'ha actualitzat a la base de dades.

La següent part comprovada és la que fa referència a la memòria cau local que emmagatzema els continguts dins el mòdul de *Drupal* per després obtenir els paràmetres de caracterització. En aquest cas també s'ha comprovat que es guarda correctament la informació i s'actualitza correctament la taula de registre (taula Log del model de dades) amb la informació que desitjada. Finalment s'ha confirmat que les peticions rebudes per part del servidor cau es resolen correctament. Per fer això s'ha utilitzat un emulador d'usuari que sol·licita pàgines al servidor cau. El mòdul de *Drupal* retorna els continguts sol·licitats, tant els continguts que es retornen directament com les etiquetes ESI que generen noves peticions de continguts al mòdul de *Drupal*.

Per realitzar totes aquestes proves s'ha creat un model extret a partir de l'estructura de pàgines i continguts de *The New York Times*. És adir s'han definit els diferents elements de continguts i les agrupacions basant-se en l'estructura del lloc web. Aquest model s'ha introduït de forma manual al mòdul de *Drupal*. Realitzant aquesta tasca s'ha detectat que seria interessant disposar d'algun sistema que permeti la simulació de les operacions dels usuaris. S'ha pensat que es podrien definir algunes primitives que permetin la simulació de la interacció d'usuaris sense la necessitat d'usuaris reals que consultin continguts i modifiquin les seves agrupacions. Per això ja s'ha iniciat el desenvolupament d'una llibreria de primitives que han de permetre que un sistema extern simuli l'activitat dels usuaris.

5. Conclusions

A continuació es realitza l'anàlisi de resultats i es fan algunes observacions de qüestions detectades durant aquest desenvolupament. També es comenten possibles treballs futurs a partir de la situació actual del mòdul de *Drupal*.

5.1. Anàlisi de resultats

S'ha desenvolupat l'eina proposada complint els 2 objectius definits com a requisits:

- Utilitzar un programari ja existent
- Utilitzar programari de codi lliure

S'ha aconseguit desenvolupar una eina que permeti la gestió de les agregacions dels continguts dels usuaris complint els 2 objectius anteriors. Després de la comparació s'ha seleccionat el programari *Drupal* i, utilitzant el llenguatge PHP, s'ha desenvolupat el mòdul corresponent.

A causa del temps dedicat al TFM (quasi 1 any) hi ha hagut un fet que pot afectar a molts altres desenvolupaments llargs. Durant aquest any ha hagut diverses actualitzacions del programari *Drupal*. Algunes han sigut actualitzacions menors (per problemes de seguretat) i alguna ha afegit noves funcionalitats. Es va iniciar la programació del mòdul amb la darrera versió en aquell moment (v6.17). Després de diverses actualitzacions menors el mòdul s'ha finalitzat amb la versió actual (v6.20). Encara que a principi d'any ha aparegut la versió 7, que afegeix noves funcionalitats i millores, la versió 6.20 és encara vàlida i suportada a causa de la gran quantitat d'instal·lacions que la utilitzen.

Aquesta problemàtica de les versions existeix als 3 entorns de desenvolupament comparats al punt 3.1 Selecció entorn desenvolupament. Amb un esquema similar, cada versió de l'aplicació conté diferents subversions. El canvi de versió comporta millores i novetats i normalment les subversions (o versions menors) sorgeixen per corregir algun error detectat. Si l'error pot representar algun perill greu per la integritat de les dades o del maquinari, llavors es recomana immediatament l'actualització. En cas contrari es pot seguir utilitzant la versió ja instal·lada.

Una altra cosa que s'ha observat a l'hora de programar en una eina ja existent ha sigut el cost d'aprenentatge de l'entorn seleccionat. El fet de no conèixer el sistema *Drupal* ha implicat un alt cost, en quan a temps, a l'hora d'iniciar la programació. Per tal d'integrar el mòdul desenvolupat i poder utilitzar totes les llibreries de funcions de *Drupal*, ha implicat invertir moltes hores d'estudi per conèixer l'entorn. Encara que hi ha molta documentació i manuals

disponibles no ha llevat aquest alt cost. Això explica que moltes empreses s'especialitzin en desenvolupar només en alguns entorns, i fins i tot, en determinades versions.

Finalment comentar que no s'afegeixen resultats quantitatius, ja que el TFM consisteix en el desenvolupament d'una eina de gestió, que no millora el temps de rendiment o resposta de forma aïllada, sinó que és un element de un sistema més complex. Es poden veure resultats quantitatius de la millora del rendiment a l'article Guerrero et al.(2011).

5.2. Treballs futurs

Durant el desenvolupament s'han detectat algunes línies de treball que permeten ampliar i millorar les funcionalitats del mòdul de *Drupal*. A continuació es detallen aquestes propostes:

- Nova API per permetre modificacions dels grafs/arbres des d'aplicacions externes

Actualment ja es troba en desenvolupament una llibreria de funcions per permetre la gestió dels arbres, nodes i continguts des d'aplicacions externes. Això permet la simulació de l'activitat de diferents usuaris a partir d'una aplicació externa.

- Traducció multi-idioma

Drupal és un sistema que permet múltiples idiomes. Inicialment el mòdul s'ha desenvolupat en català. El propi entorn de *Drupal* inclou una extensió per facilitar les traduccions, però a més a més existeixen mòduls de tercers que faciliten les tasques de traducció. El mòdul de traducció de *Drupal* facilita la traducció interactiva de les cadenes preparades per ser traduïdes (marcades amb etiquetes especials al procés de desenvolupament). Amb altres mòduls es poden exportar les cadenes que s'han de traduir per enviar-les a serveis de traducció externs i després, una vegada traduïdes als idiomes desitjats, es poden incorporar al sistema.

- Portar a *Drupal 7*

Com ja s'ha comentat abans, el mòdul es va començar a programar amb la versió 6.17 de *Drupal* i s'ha finalitzat utilitzant la versió 6.20. Encara que aquesta darrera versió és totalment funcional i suportada pels desenvolupadors de *Drupal*, ja existeix la versió 7. A la documentació en línia de *Drupal* hi ha l'explicació detallada de les modificacions que s'han de fer als mòduls desenvolupats per a la versió 6 per tal que funcionin correctament amb la versió 7.

Referències i bibliografia

Butcher, M., 2008. Learning Drupal 6 – Module Development. Packt Publishing.

Challenger, J., Dantzig, P., Iyengar, A. And Witting, K., 2005. A fragment-based approach for efficiently creating dynamic web content. In: ACM Trans. Internet Technol., vol. 5, no. 2, pp. 359 – 389.

Drupal Documentacion Team, 2000-2011. Drupal. <http://drupal.org/documentation>. Darrera actualització 15/3/2011.

Guerrero, C., Juiz, C., Puigjaner, R., 2011, Evaluation of a Fragment optimized Content Aggregation Web System. Proceedings of the 2011 Fourth International Conference on Technologies & Applications. Glyndwr University Press. North Wales.

PHP Documentation Group, 1997-2011. Manual PHP. <http://php.net>. Darrera actualització 25/2/2011.

Rodriguez Roca, P., 2000. Lenguajes Canónicos para la Descripción de Grafos: Estudio y Transformación entre esquemas de Distintos Modelos de Datos.

Veintimilla, P., 2008. Leer y escribir XML con PHP 5.x. [<http://blog.undermedia.com.ec/index.php/leer-y-escribir-xml-con-php-5x>]. Darrera actualització 12/6/2008.