

USC Announcements channel version 3

Marius Constantin, USC
marius@usc.edu

1. Announcements channel version 2 (CU/Unicon) roll out at USC
2. Behavior. Features required by USC
3. Announcements channel version 3.
Differences from previous versions
4. Performance considerations
5. Validation and Command framework
6. Text XML (“xml snippet”) framework
7. Future of the project
8. Q&A

1. Announcements channel version 2 roll out at USC

- in February 2005 USC rolled out Unicon's Academus 1.3 w/ the Announcements channel (version 2)
 - good documentation, easy to deploy and configure
 - very well received by schools and Student Affairs
 - currently has about 550 announcements distributed among 30 topics

2. Behavior. Features required by USC

- it was observed that the current channel's performance started to deteriorate as the list of announcements grew larger
- decision to try and find a workaround to improve the performance and also add a few new features

The revised channel was supposed to:

- be “fast” in displaying the announcements list
- scale for a pool of about 30K announcements
- offer a more flexible sorting mechanism
- add sorting for the topic list
- add topic categories for grouping similar topics
- ensure database locking for all object editing operations
- update topics and announcements not at request-time but in a separate process

3. USC Announcements version 3. Differences from previous version. Performance

- flattened the DOM (Document Object Model) structure of topics and announcements (search, sorting)

- every object has its own XML representation that it caches in text format. This cached value only gets refreshed as part of the edit or update processes

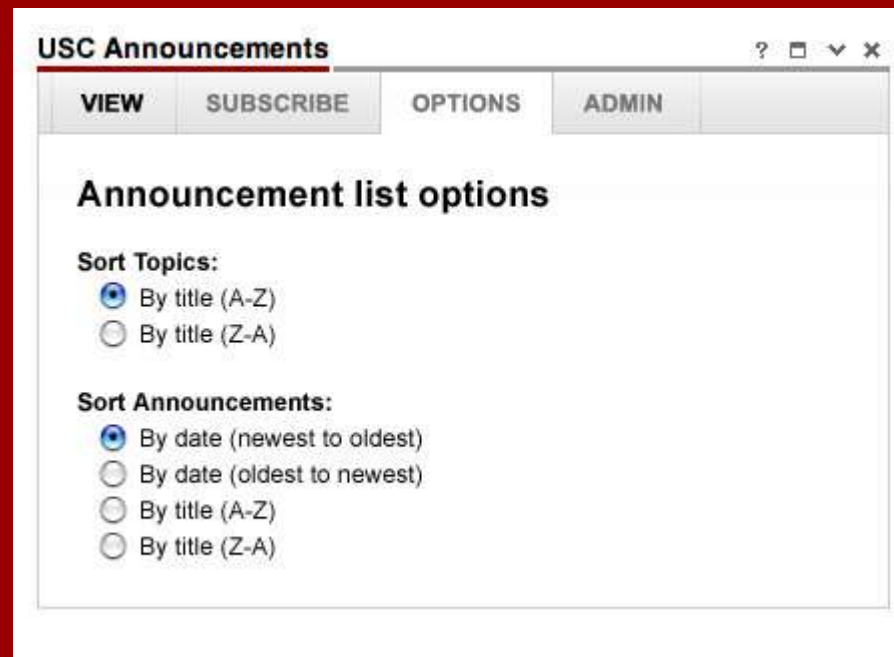
e.g. The XML representation of a Message instance would be

```
<message title="Title" id="1" date_created="04-04-2005">  
  <body>Hello <strong>world</strong></body>  
</message>
```

- implemented a framework to manipulate XML in text form
- reorganized the command and validation modules
- integrated a better error/info display module
- more flexible sorting. Entities can be sorted by any of their fields

For example, to make the “Appearance date” and “Title” fields available for sorting the announcements list, a channel administrator needs to modify three lines in the properties file:

- `comparator.message.column_list = date_appears, title`
- `comparator.message.sort_order = DESCENDING, ASCENDING`
- `comparator.message.column_list.labels = date, title`



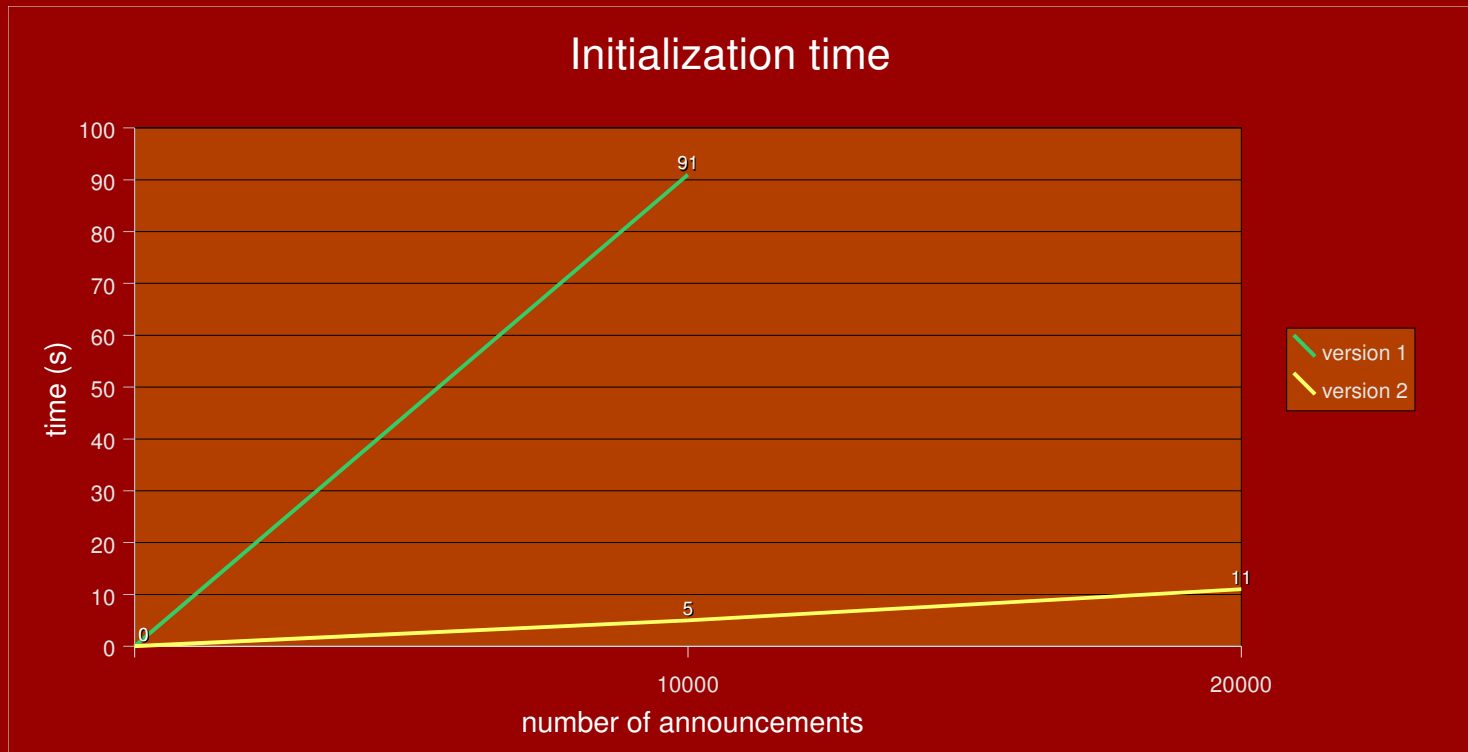
Users can then choose among these values as part of their preferences edit.

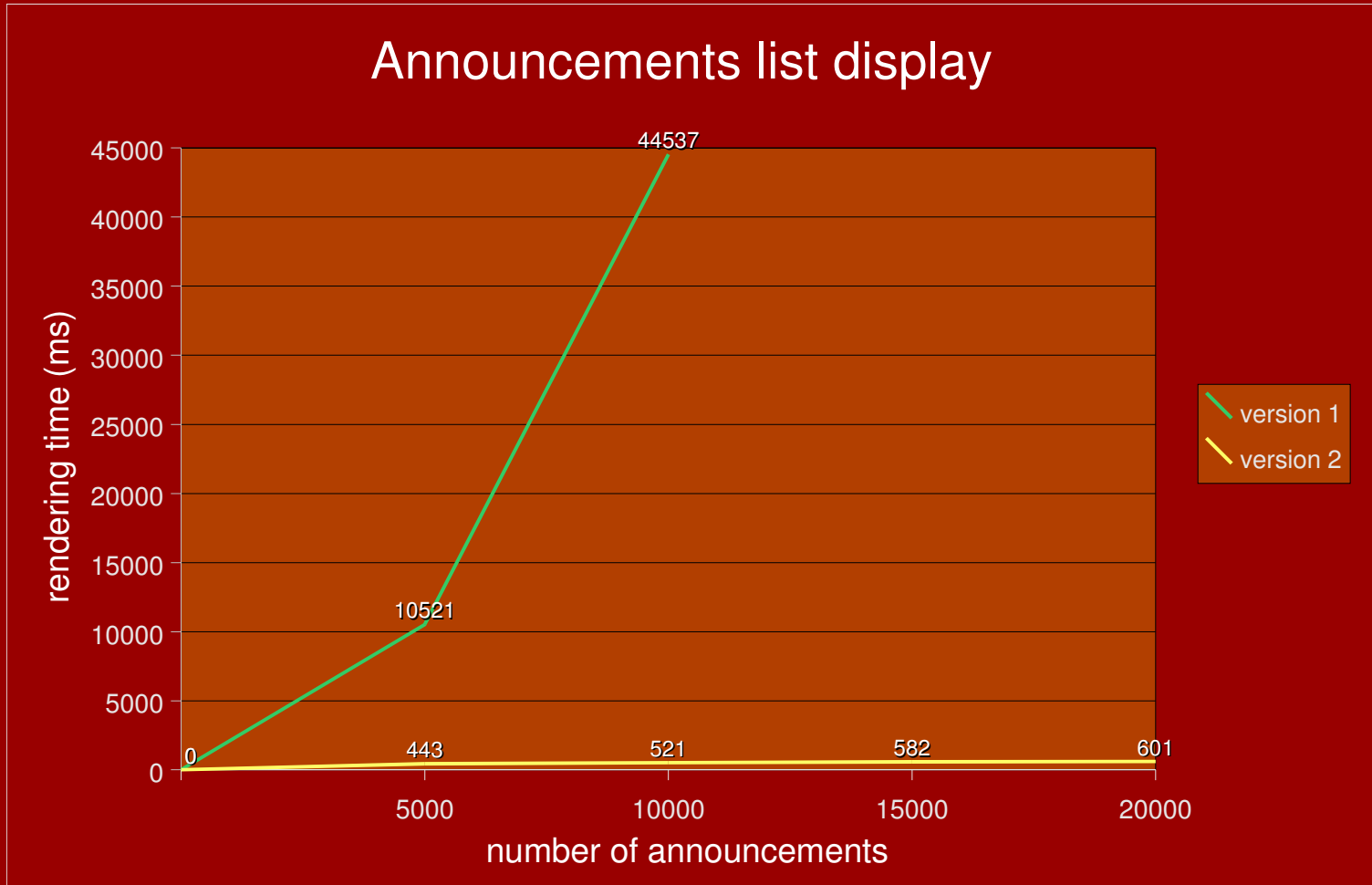
(list of changes, continued)

- “per-subscription” caching, as opposed to “per-user session” caching (saves memory and processor cycles)
- database locks for topic category, topic and announcement editing, with proper display of lock expiration
- support of remote Topic Stores (as opposed to remote Topics)
- reorganized channel distribution: separated the Java classes, the properties files, the XSL stylesheets and the media files (images)
- GUI improvements:
 - javascript calendar
 - extended the use of the htmlArea to Topic and Topic Category
 - tabular organization for faster access to all the main functions, throughout the channel
- date/time values are “xsd:dateTime” compliant
- topic and message lists are updated from the registered topic stores and the local store by a separate thread

4. Performance considerations

- very fast in displaying announcements. Since searching is done against a list and not a DOM instance, and since elements are presorted (indexed,) the displaying of announcements is done in linear time
- times of about 1 second for save and update, in a list of 20K announcements, single user





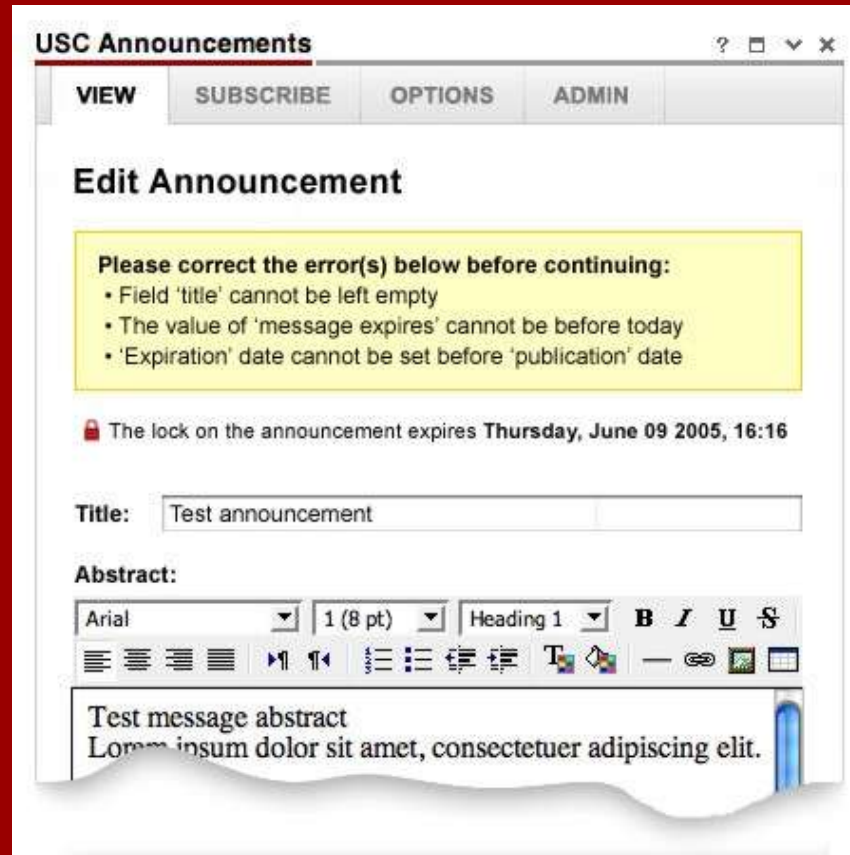
Note: the machine used for performance testing was a P4/2.6 GHz with 1 GB RAM, running RHE3; MySQL 4.2

5. Validation and Command framework

- Struts-like, but much lighter, with the main channel class being the controller
 - a “command” (equivalent to a Struts “action”) may or may not have a validator. If it has, the validator is being invoked before the control is deferred to the command
 - support for command forwarding (chaining of commands.) A command can forward to another at the end of execution. Each chained command's validator, if exists, gets invoked
 - forwarding can be done by either passing a “forward_to” parameter in the HTTP request, or in the code (by calling “setForward” method of the command class)
 - will be loaded off an xml file

5.1 Error/Info display module

- makes it easy to display errors and other information to the user



- configured in a separate properties file

5.2 Configuration and usage example for the “delete topic” operation

Error/Info message config:

```
error.delete_topic = Topic {0} could not be deleted
```

```
info.delete_topic = Topic {0} was successfully deleted
```

```
image.delete_topic = delete.gif
```


Code sample:

```
InfoMessageBag infoBag = new InfoMessageBag();
```

```
infoBag.add(“delete_topic”, “Test topic”);
```

```
setInfoBag(infoBag);
```

Result



X Topic ‘Test Topic’ was successfully deleted.

6. Text XML (“xml snippet”) framework

- in case of large lists, much faster than working with a DOM structure
- eliminates the need of recompilation in case changes in the XML schema occur
- snippet files contain the prototype of the XML representation of objects (“snippet_topic,” “snippet_message”)
 - `<topic title="[TITLE]" id="[ID]" date_created="[DATE_CREATED]"/>`
- XML gets regenerated when an instance becomes “dirty”
- interface `ITextXmlRepresentable` (“`getTextXml`”)
- types `Snippet` and `SnippetBag`

6.1 Xml snippet interpreter

- “one-pass” replace
- supports loops
- will support embedding by reference

6.2 Snippet example

Following snippet

```
%START LOOP topic_list group_id="[GROUP_ID]"%  
  <topic title="[TITLE]" id="[ID]" date_created="[DATE_CREATED]"/>  
%END LOOP%
```

produces

```
<topic_list group_id="10">  
  <topic title="Topic1" id="1" date_created="2005-05-13T10:21:59-08:00"/>  
  ...  
  <topic title="TopicN" id="N" date_created="2005-03-15T12:25:01-08:00"/>  
</topic_list>
```

6.3 Planned enhancements

The “xml snippet language” will be extended to support embedding of other snippets (“referenced”). Previous “snippet_topic_list” would look like

```
%START LOOP topic_list group_id="[GROUP_ID]"%  
  %REF snippet_topic%  
%END LOOP%
```

where “topic_snippet” file contains the Topic definition

```
<topic title="[TITLE]" id="[ID]" date_created="[DATE_CREATED]"/>
```

This way, a single topic snippet definition can be used by different other snippets (topic list, topic edit, announcements edit, etc.) without having to redefine the topic in each one of them.

7. List of features to be implemented and the future of the project

- topic and message search
 - code enhancements to further speed up the announcements list display and save memory
 - implement remote Topic Stores
 - enhancements to the validation module. Generalization so it could be used by other channels
 - Ant scripts for deployment and migration from previous versions
 - eventual migration to portlet
-
- code will be made available to the open source community
 - documentation on deployment and customizations

8. Questions and Answers

