

CATUV : un catalogue des UVs dans un serveur d'applications Internet

Utilisation de serveur d'applications Internet "Zope" pour implémenter un Catalogue des Unités de Valeur

Michel.Vayssade@utc.fr

Université de Technologie de Compiègne

AMUE – Paris – 28 Mars 2002



Spécifications : les demandes du client

- application consultable sur le web (interne et externe)
- bilingue français / anglais

Le catalogue des uvs de l'UTC / UTC's courses catalogue



[La liste des UVs](#)

[Les filières](#)



[Course list](#)

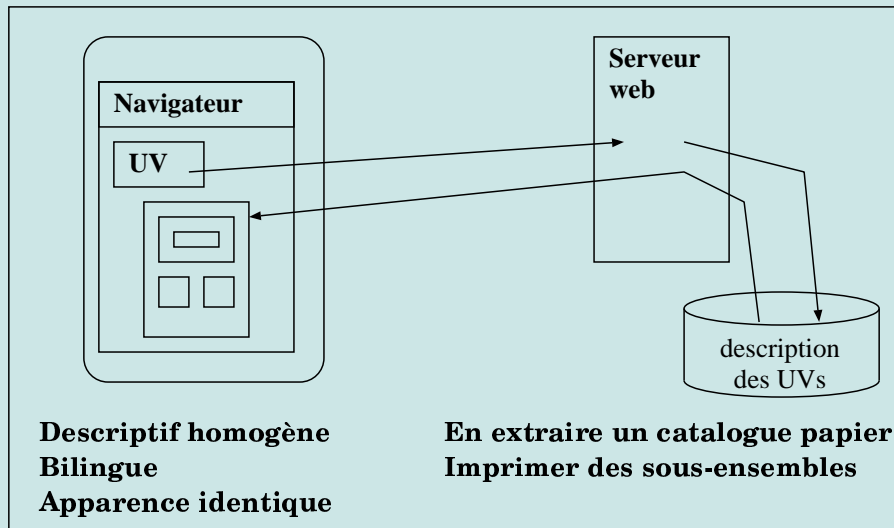
[Paths](#)

©UTC - DAE 24 sep 2001



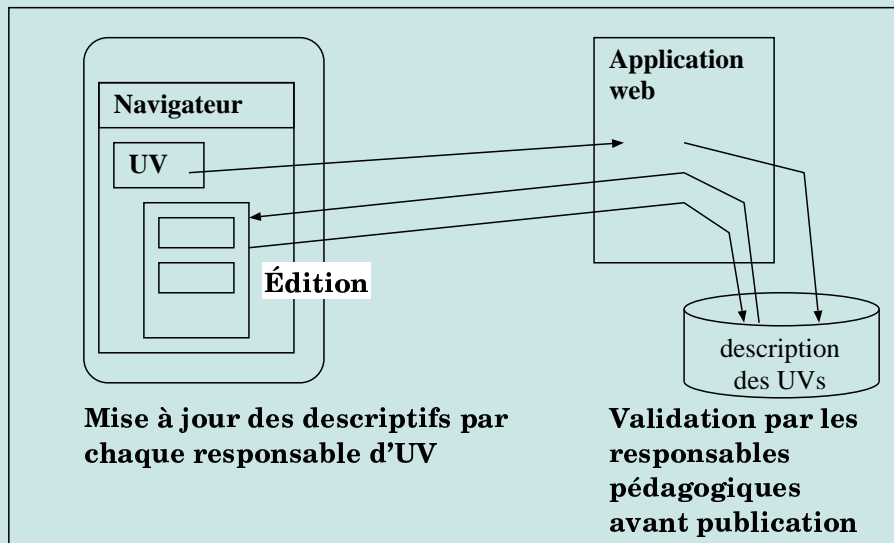
Spécifications : les demandes du client

- avec un descriptif homogène de toutes les UVs
- avec une apparence identique de toutes les pages
- sortie automatisée d'un catalogue papier à partir du contenu du site



Spécifications : les demandes du client

- modification des descriptifs des UVs par les enseignants eux-mêmes
- contrôle éditorial et pédagogique avant publication
- travail de modification sur la nouvelle version des textes sans modifier le site existant



Solutions possibles

Serveur web + PHP + stockage persistant

- stockage sous forme de base de donnée
- stockage sous forme de fichiers html ou XML

Serveur web + servlets et JSP + stockage

- stockage sous forme de base de donnée
- stockage sous forme de fichiers XML
- stockage sous forme d'objets java sérialisés

Serveur d'application

- type J2EE (weblogic, websphere, iplanet, jakarta/tomcat)
- Zope



Solutions possibles

soit, compte tenu du contexte et de la taille de l'application, trois voies "raisonnables" :

- Serveur web + PHP + fichiers XML
- "tout java", J2EE, servlets, JSP, objets java
- Zope



CATUV : catalogue des UVs - Choix d'une solution

solution (1) (PHP+XML) : testée dans un projet étudiant ; fonctionnelle, mais avait un certain nombre de défauts la rendant difficile à utiliser telle quelle : la non prise en compte à la conception du caractère bilingue, obligeait à ré-écrire le code PHP et le code XSLT (c'est-à-dire presque tout !)

solution (2) (Java+J2EE) : apparait comme largement "overkill" sur cette application, même avec un serveur "léger" tel que Jboss

solution (3) (Zope) : apparaissait intéressante sous plusieurs aspects :

- solution légère facile à mettre en œuvre
- occasion de tester quelque chose de nouveau et intéressant avec de faibles risques, l'application étant petite
- open-source
- intégrable, si besoin, à l'existant connu (base de données SQL, serveur web, PHP)

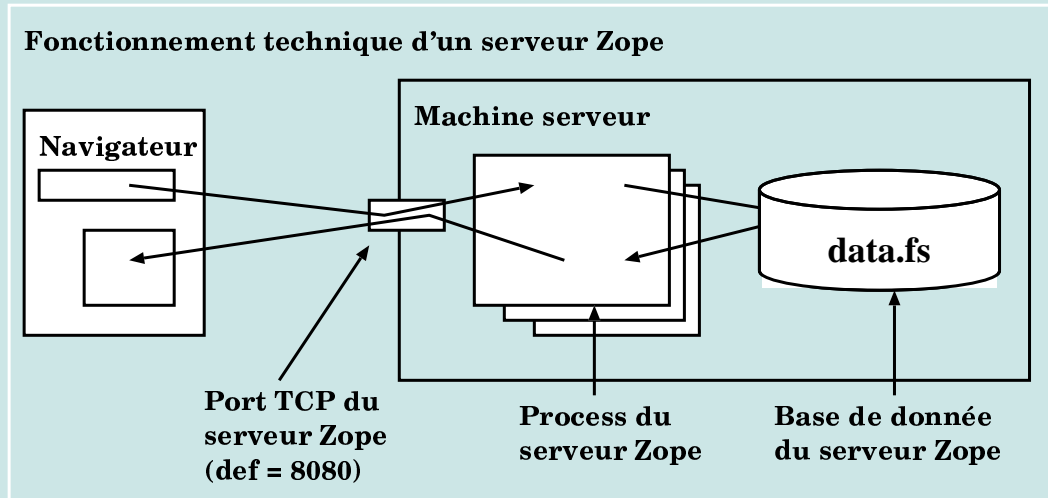


Zope : qu'est ce que c'est ?

- Zope est un serveur d'applications Internet
- open-source
- fonctionnel et fiable
- utilisable seul comme serveur web ou en collaboration avec Apache
- utilisé suffisamment : il y a une communauté active
- développé par Zope Inc. qui l'a mis en open-source et vend du service (création de sites web dynamiques et e-commerce).



Fonctionnement d'un serveur Zope



Toutes les requêtes se traduisent par l'exécution de code dans le serveur.

Ce code fabrique "à la volée" la page html renvoyée au client.



Fonctionnement d'un serveur Zope

Zope utilise deux langages :

- un langage de script ("**dtml**") : similaire à PHP dans son principe, un script dtml peut être "mêlé" à une page html et être exécuté à la suite d'une requête ; il va de même construire dynamiquement le code html renvoyé au client. Le client n'a aucun moyen de voir le source dtml.
- le langage natif de Zope : **python**. Il est possible d'ajouter à un site Zope des scripts python, mais aussi de **créer de nouveaux objets Zope** qui seront des instances d'une classe python que l'on va écrire et intégrer dans le serveur Zope.



Fonctionnement d'un serveur Zope

En quoi Zope est-il un **serveur d'applications** ?

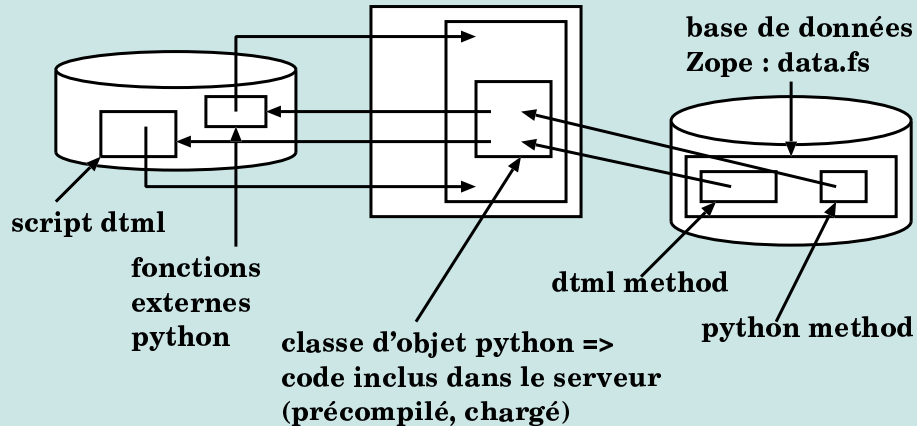
- les process Zope (multithread) conservent le code applicatif en mémoire : il n'y a pas de surcoût lié à la création d'un process à chaque requête ;
- bon, mais le mod-CGI et le mod-PHP de Apache fait la même chose ...
- le code en mémoire peut garder un **état** => plus facile d'implémenter des sessions
- ce qui est gardé en mémoire, c'est le code des **objets** au sens de la POO ; par exemple ce code peut conserver une session SQL ouverte.
- le concepteur de l'application peut choisir de construire son site avec des pages entièrement statiques ou bien des pages entièrement dynamiques, avec tous les intermédiaires possibles.



Fonctionnement d'un serveur Zope

Fonctionnement d'un serveur Zope

Différents endroits où on peut mettre du code



Toutes les requêtes se traduisent par l'exécution de code dans le serveur.

Ce code fabrique "à la volée" la page html renvoyée au client.



Fonctionnement d'un serveur Zope

Le serveur Zope gère une collection d'objets organisés selon une hiérarchie arborescente comme un ensemble de répertoires et de fichiers.

Parmi ces objets, on trouve :

- des répertoires
- des scripts dtml
- des scripts python
- des objets applicatifs



Fonctionnement d'un serveur Zope : La hiérarchie d'objets

The screenshot displays the Zope management interface in a web browser. The address bar shows the URL `http://virgo.utc.fr:8080/manage`. The browser's navigation buttons (Back, Forward, Reload, Stop) and search/print options are visible. The Zope logo and the text "Logged in as admin" are present in the header. Below the header, there are tabs for "Contents", "View", "Properties", "Security", "Undo", "Ownership", and "Find". The "Contents" tab is active, showing a list of objects under the "Folder at /" path. A search bar with the text "Select type to add..." and an "Add" button is located above the table. The table lists the following objects:

Type	Name	Size	Last Modified
<input type="checkbox"/>	Control_Panel (Control Panel)		2001-01-20 22:39
<input type="checkbox"/>	QuickStart (Zope QuickStart Outline)		2001-01-20 22:39
<input type="checkbox"/>	acl_users (User Folder)		1998-12-30 03:23
<input type="checkbox"/>	catuv		2001-09-26 16:28
<input type="checkbox"/>	hello (Hello from user mv)	1 Kb	2001-08-31 11:33
<input type="checkbox"/>	index_html	1 Kb	2001-09-25 17:21

The left sidebar shows the "Root Folder" hierarchy:

- Control_Panel
- QuickStart
- acl_users
- catuv
- FIL FIL
 - les_uvs
- users
 - gb
- mv

© Digital Creations
Refresh



Fonctionnement d'un serveur Zope : La hiérarchie d'objets

Faire une requête sur un serveur Zope revient le plus souvent à "appeler" un des objets de la hiérarchie. L'objet va alors exécuter sa méthode "view", ce qui a pour effet de se faire voir (montrer le rendu de l'objet) dans le navigateur du client.

Tout objet a une méthode "view" : s'il n'en définit pas, il hérite de celle de son objet parent.

Plus exactement, plutôt que d'héritage (réservé aux instances d'objets python) ; on parle, sous Zope, **d'acquisition**. Si un objet ne définit pas la méthode "view", le serveur va remonter la hiérarchie des objets à la recherche d'une telle méthode, et exécuter la première qu'il trouve sur l'objet.



Fonctionnement d'un serveur Zope : L'acquisition

L'acquisition est différente de l'héritage car elle est **dynamique**. Si on place une nouvelle méthode dans la hiérarchie, "plus près" de l'objet invoqué, c'est cette nouvelle méthode qui sera invoquée par le mécanisme d'acquisition, et non pas la méthode de la classe dont hérite l'objet par construction.

==> l'acquisition fonctionne lors de la recherche d'une méthode **dans la hiérarchie des objets et répertoires** et non pas seulement dans la hiérarchie d'héritage.

Quand un objet est situé dans un répertoire, les "objets-méthodes" de ce répertoire sont vus comme des méthodes de l'objet. On peut les invoquer, les appeler entre elles et passer des paramètres.



Stockage des données par Zope

Zope peut accéder aux bases de données classiques (MySQL, PostgreSQL, Oracle, Sybase, ..).

Mais il possède aussi sa propre base de données objets : il y stocke des objets python sérialisés.

Zope peut aussi accéder au système de fichiers de la machine hôte et manipuler des fichiers.



Zope est un vrai serveur d'application, comportant des caractéristiques intéressantes :

- puissance fonctionnelle
- développement rapide (une fois passée la courbe d'apprentissage)
- maintenance facile (le code python est facile à lire)
- s'intègre facilement à un existant
- gère les protections et la sécurité
- multi-plateforme (grâce à python)



CATUV est une petite application : 2000 lignes de script dtml en 25 fichiers et 2200 lignes de python en 9 fichiers (dont 1 de 950 lignes et 1 de 580).

- chaque UV est représentée par un objet python, instance de la classe "UV"
- une liste des UVs est fournie à l'application pour l'initialisation ; elle peut être modifiée par la suite ;

l'application a deux sous-ensembles :

- la partie consultation, accessible sans mot de passe
- la partie modification, accessible avec mot de passe, elle permet à chaque enseignant d'entrer et modifier les caractéristiques et le descriptif d'une UV.

Les deux sous-ensembles fonctionnent avec du html "de base".



La partie "modification" a deux sous parties :

- saisie / modif des caractéristiques (responsable, nombre d'heures, type d'examen, etc ...); elles sont stockées dans les variables d'instance des objets "UV".
- saisie / modif du descriptif de l'UV : cette description est constituée de 5 documents html. Ces documents ont les mêmes titres pour toutes les UVs et sont formatés de la même façon, afin de faciliter la lecture. Ces documents sont stockés dans une arborescence (un répertoire par UV) sous forme de fichiers XML.

L'interface de saisie / modif des documents descriptifs, accessible par le web permet aussi de faire des copies de sauvegarde des fichiers afin de faciliter les modifications lors du passage d'une version à l'autre du catalogue.



Cette édition va se faire en deux étapes :

- 1 - chaque objet "UV" est sauvé sous la forme d'un fichier XML
- 2 - UV par UV, tous les fichiers XML sont traités par une feuille de style XSL pour produire un source LaTeX transformé en postscript et/ou pdf.

On peut ainsi sortir au choix un catalogue complet ou bien un sous-ensemble. Un étudiant pourra ainsi se faire envoyer le fichier pdf du catalogue des UVs auxquelles il souhaite s'inscrire et le consulter hors ligne.



L'application CATUV : **conclusion**

Une petite application ayant permis de tester réellement un outil de développement et de déploiement très intéressant pour implémenter une interface web en frontal d'un stockage de données.

On peut ainsi loger dans Zope la partie interactive d'une fonction de traitement de données et ne laisser dans les bases de données "traditionnelles" que la partie "conservation" et "cohérence" des données. L'intérêt d'une telle séparation est multiple :

- séparation des rôles des développeurs
- simplification des programmes
- facilité d'adaptation et de développement de nouvelles fonctions

L'utilisation d'un serveur Zope permet d'accéder facilement aux techniques modernes de programmation d'applications d'entreprises connectées et interfacées au web, et ceci **sans payer le prix élevé du passage aux technologies Java et J2EE.**

