

Intégration logicielle grâce à un
protocole normalisé (WebDAV) :
Coopération et documents

Résumé

protocole http : WebDAV pour les besoins de coopération et de gestion documentaire du laboratoire ISTIT (Equipe Tech-cico).

Abstract

of protocol http : WebDAV for the needs for co-operation and document management of laboratory ISTIT (Tech-cico Team).

Mots clés : Web, protocole, travail coopératif, documents.

Key Word : Web, protocol, co-operation, documents.

RESPONSABLES PEDAGOGIQUES

- *M. Xavier Pétard*
- *M. Aurélien Bénel*

Table des matières

1. Introduction.....	3
2. Spécification des besoins.....	4
2.1. Description de l'existant	4
2.2. Analyse des besoins	5
2.3. Démarche d'étude	6
3. Présentation de WebDAV.....	6
3.1. WebDAV en bref.....	6
3.2. Extensions WebDAV	7
3.3. Architecture WebDAV	7
4. Analyse des fonctionnalités.....	8
4.1. Définitions.....	8
4.2. Diagramme de classe.....	9
4.3. Etudes des mecanismes fonctionnels specifiques aux besoins	10
4.3.1. Gestion des collections.....	10
4.3.2. Gestion des propriétés	10
4.3.3. Gestion des listes de contrôle d'accès.....	11
4.3.4. Gestion des versions de documents	11
4.3.4.1. Définitions	11
4.3.4.2. Scénario de versioning.....	12
4.3.4.3. Espace de travail collaboratif	13
4.3.5. Mécanismes de recherche de ressources.....	14
4.4. Evaluation de WebDAV par rapport aux besoins	14
5. Présentation des outils et démarche de test.....	16
5.1. Comment fonctionne WebDAV ?	16
5.2. Présentation des outils	17
5.3. Démarche de test.....	19
5.4. Résultats des tests et analyse.....	19
6. Difficultés rencontrées	23
7. Conclusion.....	24
8. Bibliographie	25
9. Annexes.....	26

1. INTRODUCTION

Le principe de système de gestion de contenu s'est élargi. En effet, les recherches récentes se sont notamment portées vers la coopération entre collaborateurs travaillant sur les mêmes plateformes web. Les évolutions récentes se sont tournées vers le développement d'une extension HTTP permettant de couvrir les besoins émergents dans un contexte d'édition collaborative à distance.

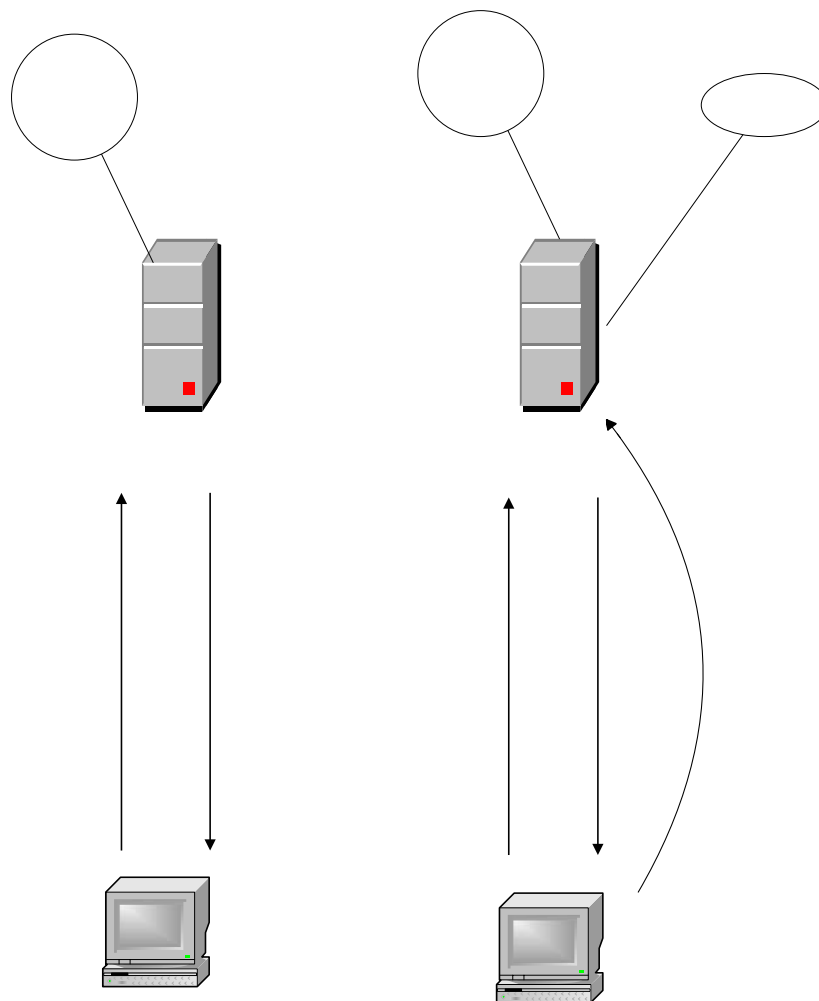
Le but de cette étude est de définir les principes de concordance de cette extension de protocole aux besoins exprimés par les laboratoires de l'UTT pour la gestion coopérative de leurs documents. Cette extension de protocole du nom de WebDAV sera donc notre point central d'étude et après avoir brièvement rappelé le contexte de cette étude et présenter le protocole WebDAV, nous conduirons une analyse des différentes fonctionnalités de WebDAV afin de déterminer leur pertinence par rapport aux besoins exprimés. Ensuite, nous déterminerons des choix applicatifs. Nous finirons par une brève conclusion reprenant ces différents points et apportant une réponse globale à notre recherche.

2. SPÉCIFICATION DES BESOINS

2.1. DESCRIPTION DE L'EXISTANT

Il existe deux interfaces de gestion de contenu accessibles via le web. Ces interfaces permettent à tous utilisateurs authentifiés de créer du contenu à travers la mise en ligne de documents, la rédaction de textes, d'articles accessibles par tous publics ayant droits. Ces plateformes collaboratives s'avèrent être un moyen efficace d'échange et de partage d'informations puisqu'elles s'appuient sur la possibilité de lire le contenu mais également de le modifier ou de le créer par quiconque. Les interfaces mises en place mettent en avant deux visions complémentaires : les structures et le contenu. En effet, ces applications sont faites de telle sorte qu'elles permettent de faciliter le choix dans la recherche de l'utilisateur en proposant un listage structuré des thèmes auxquelles se rapportent les documents. Ce listage permet de cibler un thème avec une précision la plus fine possible et permet également de faire des recherches croisées assurant le fait d'avoir le maximum de points de vues désirés dans la recherche d'un document précis.

Le contenu, une fois trouvé est rattaché à un thème central mais également à des thèmes connexes grâce à des liens directs rendant capables la navigation à travers l'application et de voir dans quel cadre le contenu désiré ou du moins le thème est rattaché à un autre thème.



Architecture systèmes

La première est une application en PHP appelé Agora accessible par le web via un navigateur. Cette application est une carte de thèmes multi points de vue traitant des logiciels open source. La deuxième application est un client JAVA appelé Porphyry qui permet de communiquer avec un serveur. Le dépôt de document se fait par un transfert FTP. Ce système s'adresse à des communautés d'experts qui souhaitent gérer des corpus documentaires en ayant la possibilité de définir leurs structures en fonction de leurs problématiques.

Nous ne rentrerons pas dans les détails fonctionnels de ces applications.

2.2. ANALYSE DES BESOINS

Compte tenu des systèmes de travail coopératif mis en place, il était nécessaire de s'appuyer sur des mécanismes fonctionnels de gestion des contenus. Ainsi de nouveaux besoins sont apparus pour permettre une certaine flexibilité dans l'édition collaborative de documents.

En général, les documents sont déposés et stockés sur des serveurs de contenu et sont décrits par des métadonnées qui sont en général des informations sur l'auteur du document, le titre du document, la date de création...Un utilisateur lambda peut consulter ce document et afficher ses informations.

La gestion de ces documents est aujourd'hui assez rudimentaire et il serait en effet intéressant d'apporter de nouvelles fonctionnalités afin d'enrichir les usages rendus possibles par les interfaces existantes. Ces fonctionnements se superposeraient à celles-ci.

Les mécanismes recherchés sont les suivants :

Besoins	Objectifs
<i>La gestion des versions d'un document</i>	Appliquer des mécanismes de versionning sur des documents accessibles et modifiables par une équipe collaborative.
<i>Créer des structures logiques de documents grâce à la gestion des métadonnées</i>	Structurer logiquement (Thèmes) la présentation des documents de manière à restituer l'information selon le souhait du ou des propriétaires des documents.
<i>Offrir des fonctionnalités de recherche de contenu</i>	Permettre l'interrogation des métadonnées d'un document pour la recherche d'un contenu
<i>Le dépôt de documents de manière sécurisé</i>	Adopter un moyen efficace de déposer des documents de manière totalement sécurisé

2.3. DÉMARCHE D'ETUDE

Les étapes de mises en œuvre de ce projet ont été déterminées au tout départ :

- Appropriation de l'existant :

La première étape a été de comprendre les architectures existantes et de prendre en compte le contexte général de l'étude. Il était également nécessaire de définir les objectifs.

- Recherche documentaire :

Cette étape consistait à rechercher des informations sur le protocole WebDAV afin de comprendre de manière générale quels étaient les fondements de ce protocole.

- Analyse fonctionnelle :

Cette étape a consisté à étudier en profondeur les différentes fonctionnalités de WebDAV afin de déterminer leur pertinence pour l'utilisation que l'on veut en faire.

- Comparaison et choix d'outils :

Mon but était d'analyser les différentes solutions logicielles côté serveur et côté client pour ainsi déterminer un choix d'outils compatibles.

3. PRÉSENTATION DE WEBDAV

3.1. WEBDAV EN BREF

WebDAV signifie Web-based Distributed Authoring and Versioning

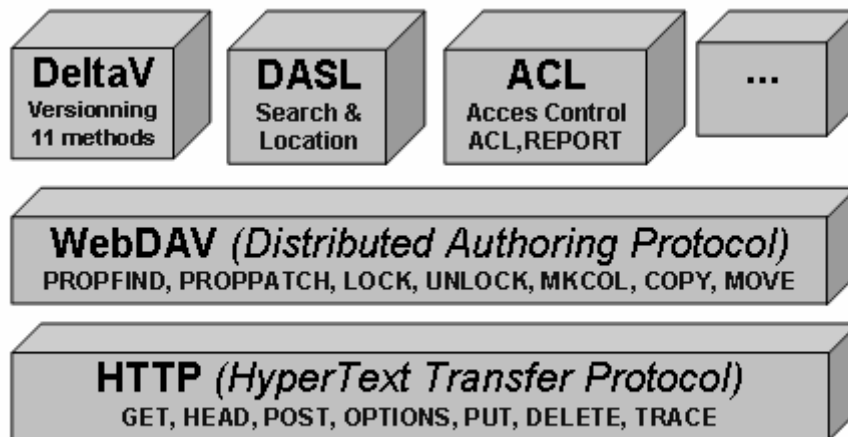
C'est une note IETF (Internet Engineering Task Force), proposée dès 1996, décrivant des extensions au protocole HTTP/1.1 qui permettent à des utilisateurs sur le WEB de participer à des processus éditoriaux (édition/gestion de documents) à distance.

Cette extension fournit des mécanismes permettant notamment de :

- Gérer des copies multiples de documents c'est-à-dire fournir à des collaborateurs une copie du document qu'il souhaite modifier et à partir de cela gérer les différentes versions d'un document ou bien rassembler dans un même document les différentes modifications opérées.
- Gérer des propriétés (métadonnées) de documents en apportant de nouvelles fonctionnalités au protocole HTTP
- Gérer le contrôle d'accès à un document
- Administrer des structures arborescentes de fichiers
- Verrouiller l'accès à un document
- Assurer le transfert de documents grâce à la sécurisation des données

3.2. EXTENSIONS WEBDAV

Le schéma suivant indique les superpositions de couches de protocole dans lequel s'intègre WebDAV et ses extensions. Ce qu'il faut savoir c'est que WebDAV étend les possibilités de http et possèdent ses extensions propres décrites ci-dessous.



Extensions de protocole http

Les spécifications de WebDAV sont décrites dans un document officiel référencé RFC 2518 fourni par l'IETF. Les protocoles définis ci-dessous ont également un document décrivant leur utilisation pour WebDAV.

DeltaV (RFC 3253)

Protocole permettant la gestion de versions multiples et de configuration de ressources Web de manière interopérable.

ACL (RFC 3744)

Protocole permettant de gérer le contrôle d'accès sur une ressource

DASL

Protocole de manipulation et interrogation des métadonnées WebDAV d'un document.

3.3. ARCHITECTURE WEBDAV

WebDAV fonctionne comme HTTP par un système de communication Requête/Réponse et possède des méthodes HTTP propres. L'entête HTTP est toujours présente mais les paramètres liés aux méthodes WebDAV sont passés soit dans un corps de requête XML, soit directement dans une entête HTTP (déconseillé quand le nombre de paramètres est important). Le corps XML doit être bien formé c'est-à-dire conforme aux spécifications pour que la requête soit acceptée par un serveur. Les réponses aux méthodes ont les mêmes exigences.

Structure d'une requête/réponse HTTP/WebDAV

>>Requête

Nom de méthode URL Version HTTP

Entête : Valeur

Entête : Valeur

...

< ?xml ... ?>

Paramètres de méthode

...

>>Réponse

Version HTTP code_statut signification

Entête : Valeur

...

< ?xml ... ?>

Réponses de méthode

...

4. ANALYSE DES FONCTIONNALITÉS

4.1. DÉFINITIONS

Avant d'entrer dans le détail des fonctionnalités afin de juger de leur pertinence, il était nécessaire de donner quelques définitions de termes propres à WebDAV

Ressource :

En général, on appelle ressource tout élément qui peut être caractérisé par une URI et possédant des propriétés. Ex : Fichiers, dossiers.

URI/URL :

Une URI (Uniform Resource Identifier) est une chaîne de caractères formatée qui identifie une ressource par un nom, une localisation ou une autre caractéristique.

Ex : name-uri: Mon livre : Chapitre 1

Une URL (Uniform Resource Locator) est une URI qui identifie une ressource par sa localisation

Ex : http://serveur/monlivre/chap2

Collection :

Une collection est une ressource qui contient des ressources membres identifiées par des URI. C'est en quelque sorte un ensemble de documents structurés de manière hiérarchique. En effet une collection peut être incluse dans une autre collection. La hiérarchie est syntaxiquement représentée par le caractère "/" géré par le mécanisme « d'espace de noms ».

Ex : La ressource identifiée par `http://serveur/monlivre/` est une collection.

Ressource membre :

C'est une ressource identifiée par un URI membre.

URI membre :

Un URI membre est un URI identifiant une ressource qui est incluse (par transitivité ou non) dans la collection primaire.

Ex : `http://serveur/monlivre/monchapitre/index.html` `index.html` est identifié par un URI membre de la ressource identifiée par `http://serveur/monlivre/`

URI membre interne :

Un URI membre interne est un URI identifiant une ressource qui est incluse directement (en relation directe) et non par transitivité dans la collection primaire.

Ex : `http://serveur/monlivre/index.html` est identifié par un URI membre interne de la ressource identifiée par `http://serveur/monlivre/`

Propriété (ou métadonnée) WebDAV :

Les informations sur les ressources sont décrites au moyen de propriétés composées d'un nom (défini par un URI) et d'une valeur. Ces éléments sont définis grâce à une syntaxe XML.

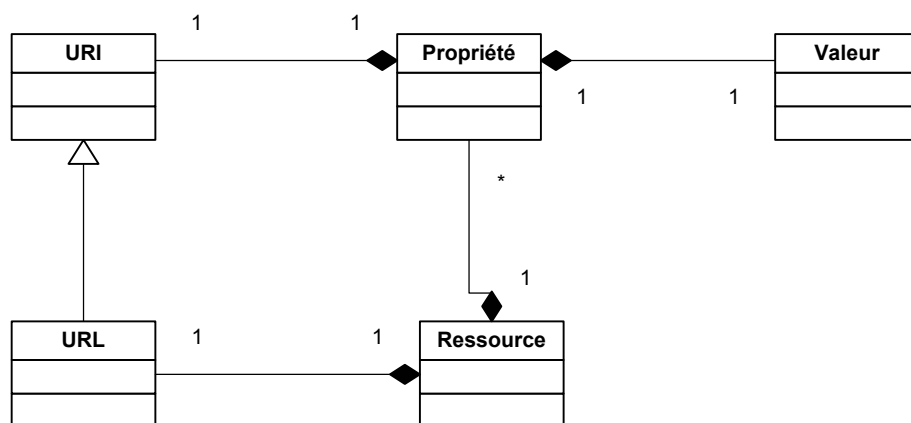
4.2. DIAGRAMME DE CLASSE

Diagramme de classe

Ce diagramme de classe montre les relations existantes entre une ressource et ses propriétés.

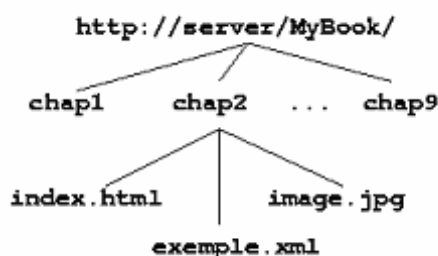
4.3. ETUDES DES MECANISMES FONCTIONNELS SPECIFIQUES AUX BESOINS

La spécification principale de WebDAV apporte 7 nouvelles méthodes au protocole HTTP : PROPFIND et PROPPATH pour la gestion des propriétés, LOCK et UNLOCK pour le verrouillage des ressources, COPY et MOVE pour la gestion des espaces de noms (Gestion des listes hiérarchiques et des URL), et MKCOL pour la création des collections. Il apporte aussi des modifications aux méthodes DELETE et PUT pour qu'elles soient applicables aux propriétés et aux collections. Nous ne rentrerons pas dans le détail de chacune des méthodes mais nous ne intéresserons qu'à celles qui correspondent à notre propos. Dans la suite, nous présentons les fonctionnalités WebDAV (+ fonctionnalités des extensions WebDAV) pouvant répondre à nos besoins

4.3.1. GESTION DES COLLECTIONS

Grâce à WebDAV, il est possible de constituer et de rechercher une liste hiérarchique de ressources (à la manière d'un répertoire de systèmes de fichiers) si elle répond aux principes d'espaces de noms URL HTTP. Les collections possèdent des propriétés comme toute les ressources WebDAV. Ce qui peut faciliter les mécanismes de recherche de ces collections grâce à l'indexation rendue possible par les propriétés WebDAV. C'est la méthode MKCOL qui permet de créer une collection au niveau de l'emplacement défini par la requête. Les collections réagissent aux méthodes COPY (copie d'une collection au niveau de l'emplacement défini par la requête) et MOVE (copie d'une collection au niveau de l'emplacement défini par la requête et destruction de la collection source).

Ci-contre une vue de la collection définie par l'URL `http://server/MyBook/`



Structure d'une collection WebDAV

4.3.2. GESTION DES PROPRIÉTÉS

WebDAV va permettre la création et l'accès à des propriétés de ressources ou de collections de ressources afin de permettre de décrire ces ressources et offrir des mécanismes d'indexation. Ces propriétés vont également permettre de décrire l'état d'une ressource. (Ex : ressource à l'état verrouillé). Pour résumer, WebDAV va permettre de définir des propriétés, d'affecter des valeurs à celles-ci pour une ressource et consulter des valeurs de propriétés.

Dans la typologie WebDAV, on distingue deux types de propriétés : les propriétés vivantes qui sont maintenues par le serveur (Ex : date de création de la ressource) et les propriétés mortes qui sont maintenues par l'utilisateur (Ex : sujet de la ressource). Les propriétés réagissent aux méthodes PROPPATCH (création, destruction, instanciation de propriétés) et PROPFIND (Récupérer les valeurs d'une propriété d'une ressource ou de tous ces membres si c'est une collection).

4.3.3. GESTION DES LISTES DE CONTRÔLE D'ACCÈS

Nous nous intéressons maintenant plus particulièrement aux fonctionnalités apportées par les extensions de WebDAV.

Nous étudions tout d'abord le protocole ACL (Access Control List) appliqué à WebDAV. WebDAV offre un mécanisme simple permettant de contrôler l'accès à des ressources. Cette possibilité est avantageuse dans le cas où des collaborateurs souhaitent déterminer quelles personnes ont le droit d'appliquer des méthodes HTTP sur des ressources quelconques. Les opérations possibles sont déterminées grâce à un ACL lié à une ressource. Un ACL contient un ensemble d'éléments de contrôle d'accès (ACE), qui identifie un utilisateur et un ensemble de privilèges qui lui sont accordés ou niés. Quand un utilisateur soumet une opération à une ressource pour l'exécution, le serveur évalue l'ACE dans l'ACL pour déterminer si l'utilisateur a la permission pour cette opération. Le protocole HTTP ne fournissait pas de méthode permettant de gérer les contrôles d'accès. Dès lors avec WebDAV, on retrouve les notions de privilèges d'individu ou de groupe, d'autorisation, de refus. L'ACL peut être consulté au moyen d'une propriété de ressource et peut être facilement modifié.

4.3.4. GESTION DES VERSIONS DE DOCUMENTS

4.3.4.1. DÉFINITIONS

Afin de décrire dans les grandes lignes l'apport de la couche DeltaV intégrée à WebDAV et le principe de collaboration, nous commençons par donner quelques définitions de termes.

Mécanisme de versioning :

Mécanisme qui s'applique à des ressources pour la gestion des versions de celles-ci.

Ressource versionnable :

Une ressource versionnable est une ressource qui peut être placée sous mécanisme de versioning. Cela peut être toute ressource non versionnée ou versionnée.

Ressource versionnée

Une ressource est par définition non-versionnée si elle est créée par la méthode PUT ou MKCOL. En fait, une ressource versionnable a été placée sous le mécanisme de versioning et ensuite elle a été archivée (Checked-in). Elle devient alors une ressource versionnée. Dans le même temps, il y a 2 ressources qui sont créées, une sorte de « dossier historique » qui contient une nouvelle ressource semblable à la ressource versionnable de départ.

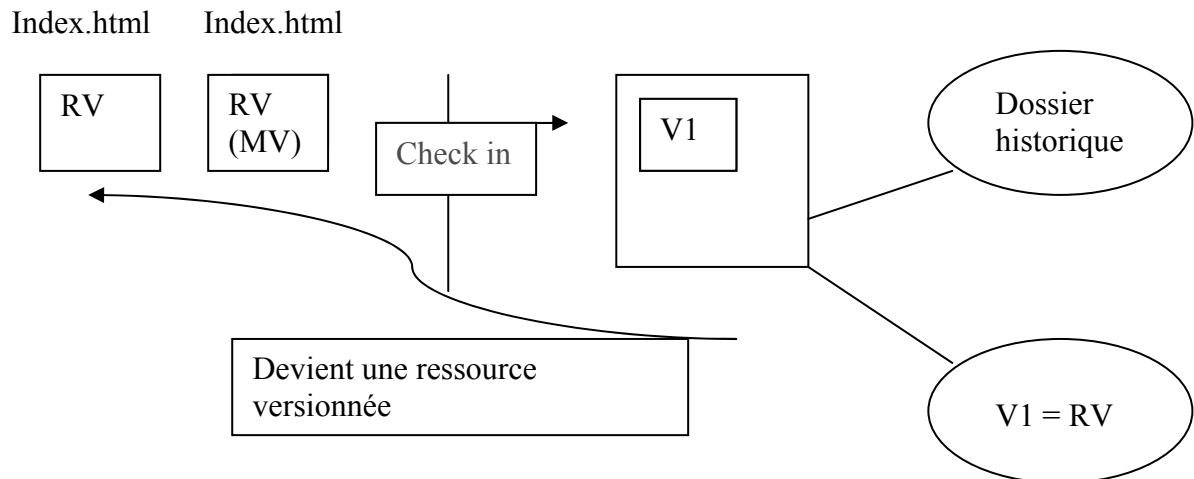
Extraction (check out)

L'opération d'extraction s'applique à une ressource versionnée qu'on a mis sous mécanisme de versioning. Cette opération va faire une copie de cette ressource pour créer une ressource en usage

(working-ressource) sur laquelle l'utilisateur va faire ses modifications avant l'archivage (check-in). La ressource de départ est verrouillée pendant ce temps là.

Archivage (check in)

Une fois la ressource en usage créé, l'utilisateur peut faire ses modifications. Une fois celles-ci terminées, l'utilisateur va archiver la ressource en usage. Dans le même temps, il y a 1 ressource semblable à la ressource versionnée de départ qui est ajoutée à l'historique. La ressource versionnée de départ est d'ailleurs déverrouillée.



RV : Ressource versionnable

RV (MV) : Ressource versionnable placée sous mécanisme de versioning

Exemple pour une ressource versionnable

Notions de prédécesseur/successeur

Une ressource versionnée va prendre plusieurs formes au cours des diverses opérations. Ces formes nous allons les appeler versions. Ce sont elles qui sont enregistrés dans le « dossier historique » et c'est l'une d'elle « celle du dossier de départ » qui possède l'URL principal.

Le prédécesseur d'une version est la ressource versionnée qui a été placée sous mécanisme de versioning qui a été extraite (checked out) avant d'être archivée (checked in).

Par analogie, nous avons la notion de successeur.

4.3.4.2. SCÉNARIO DE VERSIONING

Une fois défini les termes appliqués au versioning DeltaV, nous pouvons constituer un scénario de versioning.

Soit une ressource versionnable d'URL 'http://serveur/index.html'.

Cette ressource versionnable va devenir versionnée comme défini plus haut (Mécanisme de versioning et check in) avec la création d'une nouvelle ressource du nom de V1 semblable à index.html de départ et contenue dans l'historique http://his/V1.

Créons une autre version d'index.html.

C'est une ressource versionnée donc il faut la placée sous mécanisme de versioning. Ensuite l'utilisateur doit faire un check out pour créer une ressource en usage et travailler sur celle-ci. La

ressource versionnée est verrouillée. Une fois ces modifications finies, il les enregistre en faisant un PUT, ensuite il va faire un check-in avec donc la création d'une nouvelle ressource du nom de V2 semblable à index.html (versionnée) de départ et contenue dans l'historique http://his/V2. Dès lors, index.html va être semblable à V2 et va être déverrouillée pour que d'autres personnes puissent le modifier.

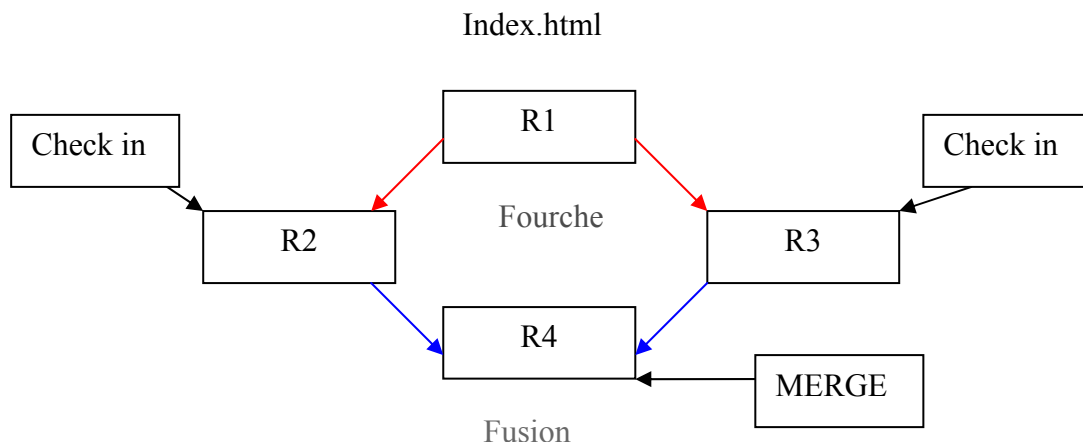
Ce qu'il faut savoir

- Les versions contenues dans l'historique ne peuvent pas être modifiées.
- Lors du check in, l'utilisateur peut choisir d'éviter les emprunts sur place (Checkout-in-place) qui consistent à faire des modifications sur une ressource sans créer de nouvelles versions. Dans ces conditions, il est utile d'appliquer les méthodes de verrouillage/déverrouillage LOCK/UNLOCK de la ressource afin d'éviter les pertes de modifications.

4.3.4.3. ESPACE DE TRAVAIL COLLABORATIF

Pour permettre à plusieurs collaborateurs de travailler sur une même ressource sans être contraints par les mécanismes de verrouillage expliqués précédemment, un mécanisme d'espace de travail est introduit pour permettre la réalisation des travaux menés en parallèle, puis ultérieurement fusionnés. C'est un espace de travail virtuel et personnel où un utilisateur choisit les ressources versionnées qu'ils souhaitent modifier. C'est ici aussi que sont créées les ressources en usage. L'activité permet de regrouper les ressources versionnées issues des modifications opérés par un utilisateur dans son espace de travail. On peut dès lors définir le mécanisme de fusions relatives à deux activités :

- Chaque utilisateur va copier une même ressource versionnée dans son espace de travail (Check out)
- Ils réalisent les changements qu'ils souhaitent sur les ressources en usage personnels
- Une personne va faire un check in
- L'autre personne va faire un autre check in mais étant donné qu'ils travaillent sur la même ressource versionnée, le serveur WebDAV détecte un conflit
- Le client de la deuxième personne récupère l'information
- La deuxième personne doit retravailler sur une ressource en usage pour reprendre ses modifications et les fusionner avec celles de l'autre personne à l'aide de la méthode MERGE
- Une nouvelle version de ressource est créée.



Mécanisme de versioning collaboratif

4.3.5. MÉCANISMES DE RECHERCHE DE RESSOURCES

Le protocole DASL intégré à WebDAV offre la possibilité de rechercher des ressources (recherche sur leurs propriétés ou leur contenu textuelle) répondant à des critères précis illustrés dans des requêtes. Grâce à un client supportant DASL, l'utilisateur va construire une requête au moyen de la grammaire DASL. Ensuite, il invoque la méthode SEARCH pour que la requête soit traitée côté serveur.

Les paramètres de la requête de recherche sont indiqués dans le corps XML de la requête. Les réponses (URL(s) de la (ou des) ressource(s) correspondantes ainsi que les propriétés et leurs valeurs ayant déterminés le choix) sont également renvoyées dans le corps XML de la réponse. Le champ de la recherche est défini par l'URL déterminé dans la requête.

4.4. EVALUATION DE WEBDAV PAR RAPPORT AUX BESOINS

Par rapport aux besoins exprimés, nous pouvons juger de la pertinence des fonctionnalités WebDAV pour notre problématique. Nous ne nous intéresserons principalement qu'aux fonctions dont on a besoin. Nous apportons une analyse critique pour tous les points abordés :

- Gestion des collections

WebDAV offre des moyens flexibles pour organiser ses documents au niveau du serveur WebDAV. Il est en effet possible de structurer une arborescence de ressources comme si l'on structurait un livre. Cependant on peut critiquer le fait que les ressources ont une unique visibilité pour tous les utilisateurs. En effet, la gestion des collections ne permet pas de structurer logiquement les documents en terme de contexte (Environnement de travail) en fonction des besoins des utilisateurs.

- Gestion des propriétés et mécanisme de recherche de ressources

WebDAV laisse une certaine liberté de modification des propriétés de documents. Ceci est intéressant dans le fait que cela permet de définir les champs d'indexation des documents et donc apporter plus de facilité dans la recherche de documents. WebDAV apporte un mécanisme de

recherche de ressources très intéressant dans le but de trouver sur le serveur WebDAV les contenus les plus pertinents.

- Gestion de version

Le mécanisme de versioning collaboratif est une des avancées les plus performantes qu'apportent WebDAV. En effet, il offre un cadre de développement coopératif et de travail collaboratif sur des documents partagés sur un serveur.

Besoins	Objectifs	WebDAV
<i>La gestion des versions d'un document</i>	Appliquer des mécanismes de versioning sur des documents accessibles et modifiables par une équipe collaborative.	WebDAV apporte un dispositif évolué permettant le travail collaboratif.
<i>Créer des structures logiques de documents grâce à la gestion des métadonnées</i>	Structurer logiquement (Thèmes) la présentation des documents de manière à restituer l'information selon le souhait du ou des propriétaires des documents.	WebDAV n'apporte aucune fonctionnalité pour gérer cela.
<i>Offrir des fonctionnalités de recherche de contenu</i>	Permettre l'interrogation des métadonnées d'un document pour la recherche d'un contenu	WebDAV dispose d'un mécanisme assez récent de recherche. Reste à savoir si les clients et serveurs WebDAV le prennent en compte
<i>Le dépôt de documents de manière sécurisé</i>	Adopter un moyen efficace de déposer des documents de manière totalement sécurisé	Cryptage des données + possibilité de passer par un client pour déposer un document.

Evaluation de WebDAV par rapport aux besoins

Fonctionnalités	Appréciation
Gestion des propriétés	5
Gestion des collections	3
Gestion des ACL	5
Gestion des versions	5
Recherche	5

Notes sur 5

Appréciation des fonctionnalités WebDAV par rapport aux besoins

Ces notes retiennent les appréciations résultant de l'analyse des fonctionnalités. Les notes maximales reflètent la concordance de ces fonctionnalités avec les besoins.

Pour la gestion des collections, on aurait pu souhaiter la possibilité de disposer des ressources selon les vues désirées par les utilisateurs. En effet, l'apport n'est que minimal puisque cela correspond à une gestion de l'arborescence de dossiers ou de fichiers par exemple. Cependant il ne faut pas non plus ignorer ce qu'apportent les collections dans un contexte de travail collaboratif. (Versioning de collection, Informations sur les ressources la composant ...).

5. PRÉSENTATION DES OUTILS ET DÉMARCHE DE TEST

5.1. COMMENT FONCTIONNE WEBDAV ?

WebDAV fonctionne en une architecture client/serveur. Nous donnons ci-dessous les différentes fonctions des serveurs et des clients WebDAV.

- **Serveur**

Le serveur est responsable des opérations principales liées au protocole WebDAV.

- C'est le serveur qui s'occupe des réponses aux requêtes.
- Il gère les différentes versions de ressources et détectent les conflits pour assurer la collaboration

- **Client**

Le client va faire l'interface avec le client. Il est important dans la mesure où c'est lui qui va apporter les fonctionnalités WebDAV à l'utilisateur indépendamment du serveur.

- En fonction des possibilités, il va permettre d'effectuer les fonctions décrites sur les ressources

Notions de classe de serveur WebDAV

Il existe deux classes de conformité à WebDAV. On parle de conformité d'un serveur ou d'une ressource. Une ressource ou un serveur qui est conforme à la classe 2 doit aussi être conforme à la classe 1. La classe 2 introduit la conformité à des nouvelles propriétés portant sur le verrouillage. En général, tous les serveurs actuels et ceux je vais présenter sont de classe 2.

5.2. PRÉSENTATION DES OUTILS

Ils existent plusieurs solutions logicielles sur le marché mais il faut savoir que l'on traverse un période de développement de logiciels supportant les fonctionnalités apportées par WebDAV. C'est pourquoi il n'y a pas vraiment pléthore de choix dans ce domaine. On compte une dizaine de produits (Clients et Serveur) en majorité disponible en Open Source.

Notre étude se limitera à l'analyse des outils les plus connus.

Serveurs : Apache Tomcat_Slide (Jakarta Slide Project), Apache (mod_dav), WebDAV Project

Clients : Dav explorer, MS Windows, Net_Http_client

Serveurs

✓ *Tomcat d'Apache*

Le Jakarta Slide Project est une compilation de modules offrant un système de gestion de contenu assurant les fonctionnalités principales de WebDAV comme par exemple le versioning ou le contrôle d'accès. Dans cette compilation, il est disponible notamment un serveur implémentant le protocole WebDAV, c'est le serveur **Apache Tomcat/Slide**.

✓ *Apache (mod_dav)*

Apache est un logiciel d'implémentation d'un serveur WEB. C'est une solution considérablement répandue puisque le logiciel serveur Apache est devenu le leader des fournisseurs de serveur http. Si le logiciel serveur Apache connaît un tel succès aujourd'hui, c'est parce qu'il permet de mettre en oeuvre de nombreuses applications et qu'il possède beaucoup d'avantages, outre celui d'être gratuit. Apache supporte évidemment WebDAV et permet l'installation aisée d'un serveur WebDAV grâce à son module mod_dav pré installé dans Apache 2.0.

✓ *Catacomb*

Catacomb est un module pour Apache (mod_dav) qui supporte WebDAV et les extensions DeltaV et DASL. Son originalité réside dans le fait qu'il stocke les ressources dans une base de données MySQL. Catacomb n'est disponible que dans une architecture Unix ou Linux.

✓ *WebDAV Project (Java Web Server)*

C'est une architecture système qui éprouve la sémantique et la syntaxe WebDAV. C'est une extension d'un serveur Web Java au quelle s'ajoute l'implémentation d'une servlet WebDAV qui permet la conformité de ce serveur aux règles devant être supportées par un serveur WebDAV de classe 2.

Clients

✓ *Slide*

Slide inclut un client WebDAV qui peut accéder à quelconque serveur WebDAV implémentant le protocole. Il est intéropérable avec n'importe quel serveur d'autres fournisseurs.

Sa force réside dans le fait qu'il est en parfait accord avec son serveur WebDAV dédié étant donné que ses fonctions sont directement liées aux possibilités du serveur WebDAV. De plus il est compatible au protocole WebDAV de niveau 2.

Le client intègre les différentes fonctionnalités de base de WebDAV à savoir :

- La sécurité (gestion des ACL)
- Le verrouillage
- Le versioning

De plus, cette application est totalement évolutive et mise à jour régulièrement puisque Slide propose une API WebDAV permettant d'agrémenter les fonctionnalités de cet outil disponible en Open Source.

Cet outil est exécutable via une boite de dialogue ms_dos.

✓ *Dav Explorer*

DAV Explorer est un client WebDAV en Java. Il peut être employé pour ajouter, supprimer, déplacer, verrouiller et ouvrir des dossiers d'un dépôt. Il peut être exécuté sur n'importe quelle plateforme de support de Java 2.

D'un point de vue fonctionnel, il exécute les fonctionnalités principales disponibles grâce à WebDAV :

- Création de nouvelles collections
- Affichage des propriétés de ressources
- Listage des versions de ressources
- Liste des contrôles d'accès
- Création de nouvelles versions
- Créations de nouvelles propriétés
- ...

Il est intéropérable avec de multiples serveurs WebDAV. Il supporte de nombreux systèmes d'exploitation comme MS Windows, Linux.

✓ *MS Windows*

Windows propose la gestion de ressources grâce au protocole WebDAV mais d'une manière différente des clients présentés précédemment. En effet, Windows est un client qui permet de gérer des dossiers WebDAV à la manière dont on gère des dossiers grâce à l'explorateur. Ceci étant, il faut au préalable l'installation d'un dossier Web –Microsoft Web Folder- qui va permettre la connexion au serveur WebDAV.

L'utilisation se fait avec le même principe qu'un serveur ftp et son drag and drop. Cette solution est très légère en terme de fonctionnalités puisqu'elle ne gère que l'aspect sécurité de WebDAV

✓ *Net_HTTP_Client*

Net_HTTP_Client est un composant Client http qui implémente toutes les méthodes **http** et la plupart des méthodes de **WebDAV**. Il possède les mêmes méthodes que le logiciel précédemment présenté. C'est une API PHP n'intégrant que les fonctionnalités WebDAV référencé dans le document RFC 2058.

✓ *Cadaver*

Cadaver est un client en ligne de commande qui fonctionne sous plateforme UNIX. C'est un client qui supporte toutes les fonctionnalités de WebDAV et les extensions DASL et DeltaV.

5.3. DÉMARCHE DE TEST

Les tests effectués sur les clients et serveurs se sont basés d'une part sur les informations données par les fournisseurs des serveurs et clients et d'autre part sur les tests des différentes fonctionnalités serveurs et clients par la manipulation de documents. C'est le client DAVExplorer qui va nous permettre de tester les fonctionnalités de ces serveurs étant donné sa complétude sur un système d'exploitation Windows XP. J'ai essayé d'adopter la plus grande rigueur lors de ces tests en vérifiant par exemple où était stocké les versions d'un document, si je pouvais ajouter/modifier des propriétés.

La démarche de test prend également en compte les difficultés qui pourraient se poser en terme d'installations des composants (Connaissances Java...), en terme d'utilisation (Difficulté d'appropriation des interfaces de ligne de commande...).

5.4. RÉSULTATS DES TESTS ET ANALYSE

Choix du serveur WebDAV:

Le serveur Tomcat_Slide d'Apache du Jakarta Slide Project offre les plus grandes garanties par rapport aux besoins exprimés. Sécurisé et multi plateforme, il reste une solution fiable et très flexible.

✓ *La gestion du protocole WebDAV*

Serveur WebDAV	ACL	Verrouillage	Versioning	DASL	Gestion des propriétés
Tomcat_slide	Gestion des versions basée sur la couche ACL conforme aux spécifications RFC 3744	Gestion du verrouillage de ressources Conforme RFC 2518	Gestion des versions basée sur le protocole deltaV	Intégration de DASL	Oui Conforme RFC 2518
Mod_dav	Le serveur mod_dav ne gère pas les fonctionnalités de listage des contrôles d'accès	Gestion du verrouillage des ressources Conforme RFC 2518	Le serveur mod_dav ne gère pas les fonctionnalités de versioning Oui (Avec l'ajout de modules comme Catacomb)	Non Oui (Avec l'ajout de modules comme Catacomb)	Oui Conforme RFC 2518
WebDAV Project	Le serveur java web server ne gère pas les fonctionnalités de listage des contrôles d'accès	Gestion du verrouillage des ressources Conforme RFC 2518	Le serveur java web server ne gère pas les fonctionnalités de versioning	Le serveur java web server ne gère pas les fonctionnalités DASL	Oui Conforme RFC 2518

✓ *Avantages/Inconvénients*

Serveurs WebDAV	Points Forts	Points Faibles
Tomcat d'apache	-Reconnu à travers le monde -Fiable -WebDAV intégré totalement -Très sécurisé	- Installation peu aisée pour un novice
Apache Mod_dav	-Serveur éprouvé -Grande performance -Evolutif	- Ne subit pas trop d'évolutions pour WebDAV -Installation de base manque de fonctionnalités cruciales (versioning)
WebDAV Project(+Servlet WebDAV)	-Multiplateforme -Trop peu fourni en fonctionnalités WebDAV	-Peu connu -pas mis à jour avec les dernières spécifications

Choix du client WebDAV:

Nous pouvons faire le choix de DAV Explorer qui s'avère la solution la plus fonctionnelle.

Limites de Slide :

Gestion de WebDAV : La version proposée sur le site du Jakarta Slide Project n'offre que les fonctionnalités de bases de WebDAV et intègre peu les avancées apportées par ACL, DASL. Cependant cette solution est évolutive puisqu'il est proposé une API permettant d'agrémenter les solutions de cette application

Ergonomie : L'interface d'utilisation : Fenêtre DOS n'est pas du tout fonctionnelle et offre très peu de libertés à l'utilisateur

Limites de DAVExplorer :

Gestion de WebDAV : Cette solution ne fournit pas les fonctions de recherche DASL.

Limites de MS Windows :

Gestion de WebDAV : Cette solution n'apporte que la partie sécurité apportée par WebDAV. Trop peu fonctionnelle pour en faire une solution d'appoint.

Limites de Net_HTTP_Client :

Gestion de WebDAV : Cette solution est vieille et ne correspond qu'aux spécifications de rfc 2058. Elle ne prend en compte que les fonctionnalités de base.

Ergonomie : il n'y a aucune interface puisque c'est une API qui propose des classes implémentant les fonctions WebDAV de base. Il est à noter qu'il est nécessaire de parser le XML pour construire les réponses renvoyées par le serveur.

Limites de Cadaver :

Gestion de WebDAV : Cette solution ne marche que sur plateforme Unix mais est assez complète.

Ergonomie : Fonctionne en mode ligne de commande (sur telnet par exemple).

✓ *La gestion du protocole WebDAV*

Client WebDAV	ACL	Verrouillage	Versioning	DASL	Gestion des propriétés
Slide	Grant Deny Revoke + API	Méthode Lock Conforme RFC 2058	Oui	Oui (avec méthodes disponibles dans l'API)	Uniquement méthode PropPut propget
Dav Explorer	Gestion du listage des contrôles d'accès	Gestion du verrouillage des ressources Conforme RFC 2518	Oui Check-out Check-in ...	Non	Oui Conforme RFC 2518
Windows	Non	Non	Non	Non	Non
Net_http_client	Non	Méthode Lock Conforme RFC 2058	Non	Le serveur java web server ne gère pas les fonctionnalités DASL	Oui PropFind PropPatch
Cadaver	Non	Gestion du verrouillage des ressources Conforme RFC 2518	Oui Check-out Check-in ...	Oui	Oui Conforme RFC 2518

6. DIFFICULTÉS RENCONTRÉES

Les principales difficultés que j'ai rencontrées résident dans le fait qu'ils n'existent pas beaucoup de documents sur le net traitant des problématiques WebDAV et des outils applicatifs existants. Mis à part le site Web www.webdav.org, je n'avais pas vraiment d'appui pour réaliser mes recherches et mes analyses. Je peux notamment ajouter que la majeure partie des documents était en anglais, ce qui me bloquait parfois dans la compréhension de termes techniques et altérait les analyses que j'en faisais.

Cependant traiter d'une problématique comme celle-ci m'a vraiment satisfait puisque j'ai pu aborder des notions qui étaient pour moi inconnues. J'ai pu comprendre l'importance que revête aujourd'hui le travail collaboratif et découvrir les recherches qui vont faire avancer le Web et son utilisation.

7. CONCLUSION

WebDAV apporte certaines solutions aux besoins exprimés. Sa gestion du travail collaboratif ne peut pas laisser indifférent. Du fait qu'il soit simple et qu'il apporte des solutions non négligeables pour la gestion du versioning, de la collaboration entre utilisateurs Web. Ces services collaboratifs peuvent être en complément des infrastructures actuelles du laboratoire et sont très faciles à intégrer. Il serait intéressant d'ailleurs d'appliquer en contexte réel ces systèmes pour ainsi juger par l'expérimentation sur des documents de l'utilité de ce protocole et déterminer les difficultés d'utilisation qui pourraient se poser dans un cadre multi-utilisateur. Bien que certaines fonctionnalités ne se retrouvent pas dans les définitions actuelles, on peut réfléchir aux développements spécifiques qui viendraient combler les manques actuels ou bien combiner les fonctions de plusieurs outils séparément. Il faut noter cependant que la plupart des outils applicatifs ne sont pas encore très évolués mais l'on peut s'attendre dans quelques années à voir émerger des outils fonctionnels qui apporteraient beaucoup plus au travail collaboratif.

8. BIBLIOGRAPHIE

Sites Internet

- Stein G., Whitehead J., *Welcome to WebDAV Resources* [en ligne].
Disponible sur : < <http://www.webdav.org> >
- West L., *HTTP Client Class* [en ligne]. Disponible sur :
< http://lwest.free.fr/doc/php/lib/index.php3?page=net_http_client&lang=fr >
- Apache Software Foundation, *Slide* [en ligne]. Disponible sur :
< <http://jakarta.apache.org/slide/> >
- Jourdan N., *RFC-editeur.org* [en ligne]. Disponible sur : < <http://abcdrfc.free.fr/> >
- IETF, *Request for Comments* [en ligne]. Disponible sur : < <http://www.ietf.org/rfc.html> >
- Bénéol A., *Porphyry : un outil de lecture-écriture pour experts* [en ligne].
Disponible sur : < <http://www.porphyry.org> >
- Laboratoire Tech-CICO de l'ISTIT (Troyes), *Yellow Pages for Open Source Software* [en ligne]. Disponible sur : < <http://www.yeposs.org> >
- Communauté des développeurs du site XMLfr, *<XML>fr* [en ligne].
Disponible sur : < <http://xmlfr.org> >

9. ANNEXES

Annexe 1 : Exemples d'interfaces d'implémentation

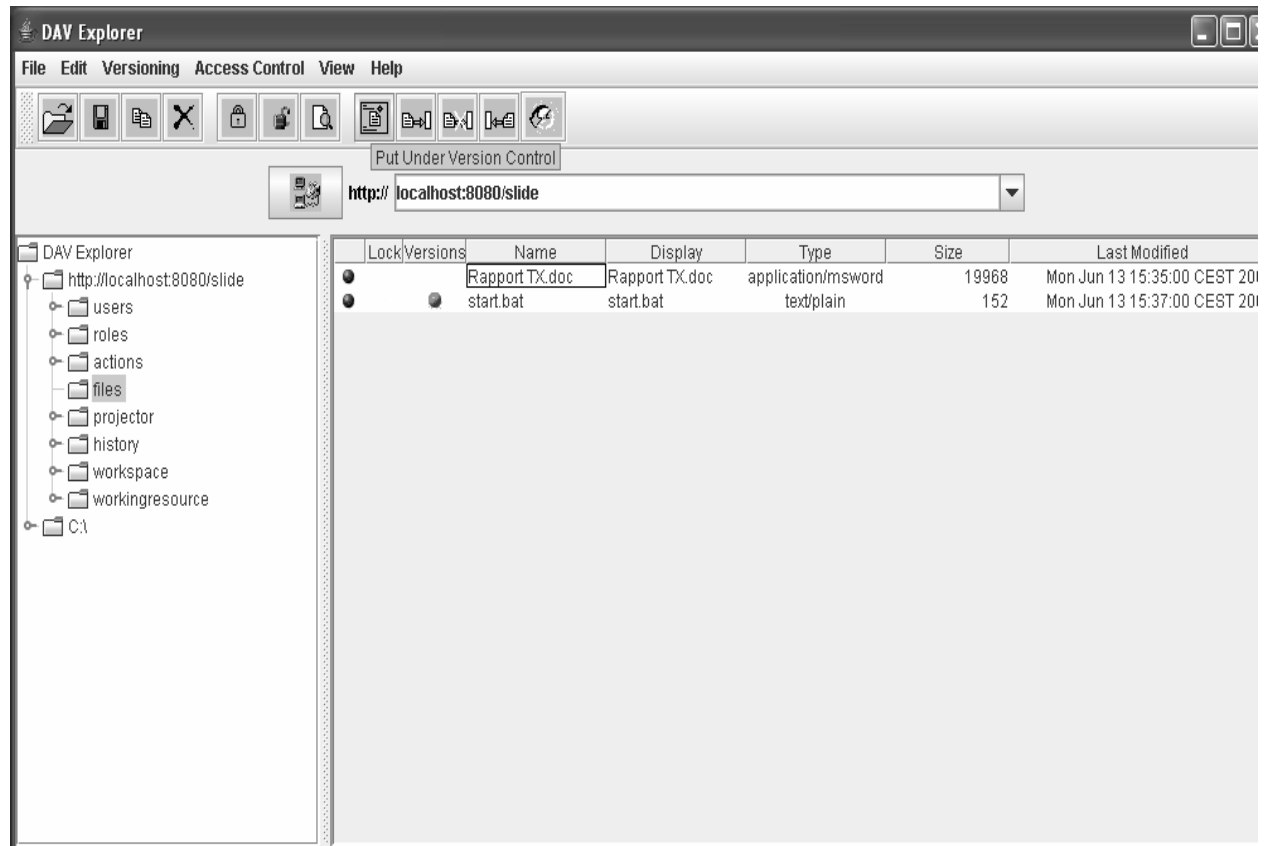
✓ *Slide*

```

syntax error. "help" for more info
[ Slide ] $ help
Commands:
options <http_URL!path>      Print available http methods
open [http_URL]             Connect to specified URL
close                       Close current connection
exit                        Exit Slide
help                        Print this help message
debug <ON!OFF>              set debugmode
lpwd                        Print local working directory
lcd [path]                  Change local working directory
lls [-lF] [path]           List contents of local directory
pwc                         Print working collection
cc [path]                   Change working collection
ls [-lF] [path]            List contents of collection
url                          Print working URL
status                      Print latest http status message
get path [file]             Get a resource to a file
put <URL!file> [path]      Put a given file or URL to path
mkcol path ...              Make new collections
delete path ...             Delete resources
copy source destination     Copy resource from source to destination path
move source destination     Move resource from source to destination path
lock path [-t<xxx!inf>] [-s<SHARED!EXCLUSIVE>] [-oOWNER]
                             Lock specified resource.
                             default:
                             lock file -t120 -sexclusive with current owner
unlock path [-oOWNER]       Unlock specified resource.
                             default:
                             unlock file with current owner
locks [<path>]              Displays locks on specified resource
propget path property ...   Print value of specified property
propgetall [<path>]         Print value of all properties
propput path property value Set property with given value
set URLencode <on!off>     Set URL encoding flag, default: on
set debug <on!off>         Set debug level, default: off
acl [<path>]                Displays the ACL of path
principalcol [<path>]       Displays the principal collection set of path
grant [<namespace>] <permission> lon <path>] to <principal>
deny [<namespace>] <permission> lon <path>] to <principal>
revoke [<namespace>] <permission> lon <path>] from <principal>
versioncontrol path         set a versionable resource under versioncontrol
versioncontrol URL path    create a new versioncontrolled resource at path
based on history URL
checkout path                checkout of a checkedin resource
checkin path                 checkin of a checkedout resource
uncheckout path              undoing changes on Resource since checkedout <
cluding checkout>
report path [<property>]    getting report <version-tree> for any resource
eReport path                 getting report <expand-property> for any resource
e LReport path [<property>] ON [<historyuri>] getting report <locate-by-history>
mkws path ...                Make new workspace
update path target           Update a resource identified by path to v
sion identified by target
begin                         starts a new transaction <only if server supp
ts this>
commit                       commits the transaction started by begin <only
if server supports this>
abort                         aborts and rolls back the transaction started
y begin <only if server supports this>
Aliases: help=?, open=connect, ls=dir, pwc=pwd, cc=cd, lls=ldir, copy=cp,
move=mv, delete=del=rm, mkcol=mkdir, propget=propfind, propput=proppatch,
exit=quit=bye
Comment : Once executed, the debug mode will be active.
          Because it's not triggered by methods yet.
[ Slide ] $ _

```

Interface d'implémentation (Ligne de commande) → Liste des fonctions WebDAV

✓ *DAV Explorer**Interface de DAV Explorer*