

## Un service "maillage" sur le WEB ?

### description d'une expérience

#### 1. Introduction, pourquoi cette expérience ? :

Source d'informations inépuisable, télétravail, offre d'achats et de services , le Web est devenu incontournable dans la vie moderne. Deux des auteurs (M.B. et F.E), retraités mais toujours passionnés, se sont posé la question suivante : est-il possible, par le seul emploi de leur navigateur de permettre à des professionnels d'accéder à des services hautement interactifs sur le WEB? La disponibilité du logiciel graphique "WebGL" intégré dans les navigateurs récents les plus répandus ainsi que l'énorme saut technologique procuré par le déploiement de la fibre optique semblait rendre le problème envisageable. Pour en être sur, ils ont entrepris , à partir d'une maquette d'étudiants que l'un d'eux encadrait, de créer un prototype totalement opérationnel en y incorporant des applications qu'ils connaissaient bien : visualisation de modèles CAO et de maillages , tessellation et maillage de modèle CAO, opérations et changement de formats sur les maillages. Pour être complet, le prototype comporte comme tout service, des fonctions de gestion telles quel' enregistrement des utilisateurs, pseudo, mot de passe, consommation des ressources et ... moyen(s) de paiement du service. Le troisième auteur (N.G) s'est chargé d'adopter le point de vue de l'utilisateur pour évaluer l'efficacité de la solution, domaine d'intérêt et limitations.

#### 2. Objectifs et contraintes

Les applications doivent être simples à utiliser, accessibles même à des utilisateurs occasionnels. Aucun logiciel à télécharger, ou à mettre à jour.

Formalités administratives simplifiées , prépaiement type "carte de téléphone" .

Sécurité et confidentialité : données personnelles minimales ( pseudo , mot de passe). Les données d'entrées et résultats des applications ne résident sur le web que lors de la cession : elles sont détruites dès la déconnexion de l'utilisateur (ou fermeture de la page web).

Utilisation du protocole https.

#### 3. Les moyens mis en œuvre:

Le projet a été développé à coût externe minimal , utilisant au maximum savoir faire et créations antérieures des deux protagonistes , ce qui a forcément influencé sur le choix des solutions techniques retenues. A noter qu'aucun des deux ne connaissait plus le "web" et son fonctionnement que n'en connait un utilisateur "standard".

- la "Plateforme web"

La solution la plus simple semblait être un hébergement mutualisé, avec serveur web standard préinstallé. Mais bien trop rigide pour y intégrer les applications développées

en C++ se présentant sous forme d'exécutables et bibliothèques partageables. Pas question non plus d'utiliser un "générateur de site". Choix retenu: location chez un hébergeur d'un "VPS", machine virtuelle avec configuration juste nécessaire pour effectuer des tests réalistes des applications : 2 cœurs, mémoire RAM 4Go, SSD 60 Go. Système : Windows 64 bits (médiocres performances pour le partage des ressources, mais très commode pour les développements avec l'accès "bureau distant").

- **Le logiciel serveur**

Beaucoup de sites Web utilisent le serveur "APACHE". Plus récent, et mieux adapté semble-t-il aux applications très interactives le serveur "Node.js" a été retenu. Mais son plus gros avantage pour des développeurs c'est sa très grande compatibilité avec le langage "JavaScript", lequel est utilisable pour les développements tant du côté "serveur" que du côté "client" (ce qui n'est pas le cas pour la solution APACHE). La base de données "MySQL" est utilisée pour enregistrer les données personnelles utilisateur.

- **Les développements réalisés pour le prototype:**

Outre les fonctions de service (début, fin de session etc..) effectuées en JavaScript, on note:

- i. Côté client :

La présentation des pages web (structure générale, onglets, boîte de dialogue...) a été définie directement à l'aide d'un éditeur selon standard HTML, leur aspect dynamique par des fonctions développées en JavaScript, et surtout, la visualisation 3D (avec zoom, rotation, éclairage...) est développée en JavaScript en faisant appel aux fonctions WebGL incluses dans le navigateur.

- ii. Côté serveur :

Le lien avec les applications : récupération et interprétation des requêtes, lancement des exécutables spécifiques des applications, récupération des résultats et transfert au navigateur de l'ordinateur de l'utilisateur.

#### 4. Les applications expérimentées:

- **Visualisation 3D de maillages**

Cette application fondamentale sert d'organe d'importation, visualisation et export des données d'entrée et résultats des autres applications. L'import d'un fichier maillage situé sur l'ordinateur de l'utilisateur s'effectue selon la séquence suivante : mise en forme d'une requête d'import (objet HTML du type "form"), transfert du fichier (protocole HTTPS entre le navigateur et node.js), lancement de l'exécutable "appli visu" chargé de convertir les données maillages en données graphiques et finalement transfert des données graphiques à l'ordinateur client pour affichage.

Initialement conçue pour une utilisation sur ordinateur "local", programmée en C++ et librairie graphique OpenGL, cette application a dû être profondément modifiée pour les raisons suivantes:

- i. objectif: effectuer les opérations courantes localement sans faire appel au serveur: zoom, translation, rotation, visibilité etc... d'où réécriture en JavaScript
  - ii. La librairie WebGL supportée par les navigateurs est différente de celle d'OpenGL et de surcroît n'en offre pas les fonctions de niveau élevé
  - iii. création d'une arborescence permettant à l'utilisateur de parcourir la structure du maillage
  - iv. création d'une interface utilisateur pour environnement web (HTML et JavaScript)
- **Tessellation de modèle CAO (format Step et Iges)**

Cette application nommée "QuikMesh" est ancienne. Longuement éprouvée, elle a fait l'objet d'une présentation au MICAD 2002 et est toujours diffusée, en particulier comme module intégré dans un logiciel professionnel. Son intérêt est de créer une triangulation "parfaite" c'est à dire de topologie mathématiquement conforme, étanche et de précision garantie, même lorsque le modèle CAO n'est pas parfait lui-même : raccordement imprécis des faces, et toute une série de petits défauts énumérés par exemple dans la norme de qualité SASIG. En plus des deux paramètres fondamentaux, erreur de corde maximale et angle maximal entre 2 triangles voisins d'une même face, l'utilisateur peut définir 2 paramètres supplémentaires longueur maximales des arêtes, et rapport de forme maximal. Initialement conçu pour alimenter les premières machines de fabrication additive, QuickMesh s'est avéré très utile pour toute application située en aval de la conception (infographie de qualité, simulation, prototypage, fabrication) en lui fournissant des données géométriques de précision garantie sous une forme facile à exploiter.

QuickMesh a pu être intégré dans sa forme d'origine: exécutable + librairie dll pour environnement "Windows", sans aucune modification.
  - **Contrôle, réparation, opérations sur un maillage**

Cette application permet de détecter et si possible corriger automatiquement les éventuels défauts d'un maillage, qu'il ait été importé ou créé sur le serveur: défaut de connectivité, maillage non conforme, éléments dégénérés, en collision, quasi recouvrement, dupliqués, fentes et "petits trous". Les défauts sont présentés graphiquement (de façon rudimentaire, amélioration de l'ergonomie prévue).

Lorsque le maillage d'entrées comporte plusieurs sous ensembles, chacun représentant un volume, mais manifestement en collision, toutes les intersections ou recouvrements

d'éléments sont résolus pour former un objet unique formé de plusieurs compartiments. Chaque compartiment ou groupe de compartiments peut être sélectionné individuellement, rendu visible ou non et exporté, (opération booléenne sur un ensemble d'objets).

Une fonction complémentaire permet d'évaluer la qualité globale d'un maillage selon plusieurs critères de forme portant sur chaque élément. Présentation sous forme de coloriage et tableau statistique.

- **Mailleur surfacique dans l'espace 3D pour applications de simulations**

Le mailleur était initialement fourni par une entreprise partenaire spécialiste de ce type d'application et intéressée par l'expérience. Le partenaire ayant été absorbé, le partenariat s'est terminé et le vide a été comblé par la création "maison" d'un mailleur frontal. On l'appellera "Remesh". Son développement et son fonctionnement s'appuie sur QuickMesh : Remesh se sert de la triangulation créée par QuickMesh comme données d'entrée et en réutilise maintes fonctions de base. Mais il peut aussi être utilisé en tant que Remaillleur d'un maillage importé et même dans toutes ses options si le maillage importé est organisé en sous ensembles, chacun correspondant à une face du modèle CAO d'origine. (ce qui est le cas de la triangulation générée par QuickMesh). Remesh est limité aux éléments "triangle". Il crée 2 types de maillage et pour chacune 2 options:

- i. taille d'arête fixe définie par l'utilisateur
- ii. taille variable, adaptation automatique selon la courbure de la surface; l'utilisateur définit une taille minimale et maximale d'arête et un coefficient de progression (rapport max de longueur d'arête partageant un même point extrémité)
- iii. option maillage "face par face" : chaque frontière de la face CAO d'origine est considérée comme une ligne de contrainte à respecter qu'aucune arête ne doit franchir.
- iv. option "transcareau" dans laquelle les lignes de contrainte sont uniquement les frontières de face qui constitue une arête vive une discontinuité de courbure. Détection automatique, selon un paramètre angulaire défini par l'utilisateur.

## 5. Charge du serveur, précautions

Par principe un serveur web doit répondre à de multiples requêtes de multiples utilisateurs.

Afin d'éviter les brusques surcharges du serveur, les requêtes sont triées selon l'estimation des ressources nécessaires pour la traiter. Les plus légères sont lancées immédiatement, les autres en particulier celles qui requièrent "executable" et/ou transfert de gros fichiers, sont mises en file d'attente et traitées et traitées séquentiellement.

## 6. Sécurité et confidentialité:

- **informations personnelles:** à part pseudo et mot de passe et adresse mail nécessaires pour son identification, aucune information personnelle n'est demandée à l'utilisateur . Le mot de passe n'est pas archivé sur le serveur, seule une signature obtenue par un mécanisme de cryptage irréversible est archivée ( base de données MySql) avec le pseudo et les informations concernant la consommation des ressources .
- **Les données :** Les données (entrée ou résultats) ne sont pas conservées sur le serveur. Elles sont automatiquement détruites lors de la déconnexion de l'utilisateur ou la fermeture de la page web. Au delà d'une certaine période d'inactivité de l'utilisateur, la déconnexion est automatique et ses détruites.

## 7. Mécanisme de paiement:

Prévu principalement pour des utilisateurs occasionnels, un mécanisme aussi simple que possible. Basé sur un système d'unité de service, Chaque utilisation d'une des applications consomme des unités de service, d'autant plus que le volume d'informations traitées est important. Lors de ses premiers tests ,un utilisateur dispose librement d'une centaine d'unités de service. Ensuite, l'utilisateur peut acheter des unités et comme sur les sites de ventes, il est redirigé sur un système de paiement bancaire. Deux systèmes ont été testés : "paypal" bien connu et "Monetico" utilisé par plusieurs banques françaises. Le système "paypal" a été retenu ... parce qu'il ne coûte rien quand on ne l'utilise pas, mais son fonctionnement est assez obscure contrairement à Monetico très bien documenté.

## 8. Tests et résultats :

- **Point de vue des développeurs:**  
L' "expérience" s'est déroulé conformément aux objectifs prévus. Globalement, les performances obtenues correspondent à celles d'un ordinateur local de mêmes caractéristiques que l'ordinateur virtuel distant du Web ( 2 cœurs de rapidité standard). Le temps de réponse pour le transfert et l'affichage d'un maillage de 10 Méga octets est inférieur à 2 secondes. Pour un tel maillage, rotation, déplacement et désignation sont très fluides.
- **Point de vue des utilisateurs:**  
Un grand merci par avance aux utilisateurs qui nous feront part de leurs remarques et critiques constructives. Et si vous souhaitez participer plus activement à cette expérience, c'est avec plaisir que nous accepterons votre aide.  
contact@meshprocess.com  
et n'hésitez pas à faire quelques tests  
[meshprocess.com](http://meshprocess.com)

