



olivier.burdet@epfl.ch



jean-luc.zanella@epfl.ch

i-STRUCTURES, LE PROJET

OLIVIER BURDET ET JEAN-LUC ZANELLA, LABORATOIRE DE CONSTRUCTION EN BÉTON, ENAC - EPFL

BÂTISSEURS, ARCHITECTES ET INGÉNIEURS

La construction est une activité qui a de tout temps fait partie des activités humaines importantes. C'est d'ailleurs, avec l'art, une des traces les plus anciennes qu'une civilisation laisse, et qui permet de se rendre compte de l'étendue qu'elle avait atteinte. Les plus anciennes constructions connues datent de plusieurs millénaires avant J.-C.

Au cours des âges, l'art de bâtir s'est transmis de génération en génération, presque sans qu'en subsistent des traces écrites. Ce n'est qu'à partir de la Renaissance que les connaissances mathématiques et scientifiques du comportement des structures ont permis une compréhension formelle de leur fonctionnement. Cependant, la responsabilité de bâtir est restée entre les mains d'une seule et même personne, le *bâtitseur*, souvent identifié comme l'architecte, mais dont les compétences recouvraient l'ensemble de la construction.

C'est au cours de la Révolution Industrielle que s'est produite la séparation entre les deux figures principales de la construction de l'ère moderne: l'architecte et l'ingénieur. Depuis, cette séparation n'a fait que croître, tant la somme de connaissances et techniques à maîtriser a augmenté tant pour les architectes que pour les ingénieurs. A l'heure actuelle, il serait illusoire de vouloir réunir ces deux professions, car rares seraient ceux qui pourraient dominer un tel ensemble de connaissances.

UN NOUVELLE APPROCHE PÉDAGOGIQUE

C'est dans ce contexte que, lors de la création de la faculté d'architecture (Accademia di Architettura) de la nouvelle Université de la Suisse Italienne (USI), l'approche choisie par Aurelio Muttoni, qui en a été le premier professeur de structures, a été de favoriser pour les étudiants architectes une compréhension globale du comportement des structures, accompagnée d'une culture étendue des types de structures classiques et contemporains. En allégeant le bagage scientifique et mathématique au profit de méthodes de calcul graphique développées au

XI^e siècle, le cours se concentre sur les aspects principaux du fonctionnement des structures: flux des forces, transmission des charges aux appuis, dimensions principales. Loin d'être une simplification abusive de la théorie, cette approche permet au moyen d'outils simples de comprendre le mode de fonctionnement de tous les types de structures, d'en estimer les dimensions principales et favorise ainsi le contact ultérieur entre l'architecte et l'ingénieur.

Depuis sa création en 1996, ce cours a connu un grand succès auprès des étudiants, tant grâce à l'enthousiasme de l'enseignant et de ses assistants que par la richesse de son approche, dans laquelle des essais sur maquettes, une résolution graphique et une intégration dans le monde de structures réelles se mêlent efficacement. Les structures conçues par les

étudiants architectes lors des semestres ultérieurs de leurs études reflètent à la fois toute la liberté du créateur et la compréhension du mode de fonctionnement des structures qu'ils imaginent [2].

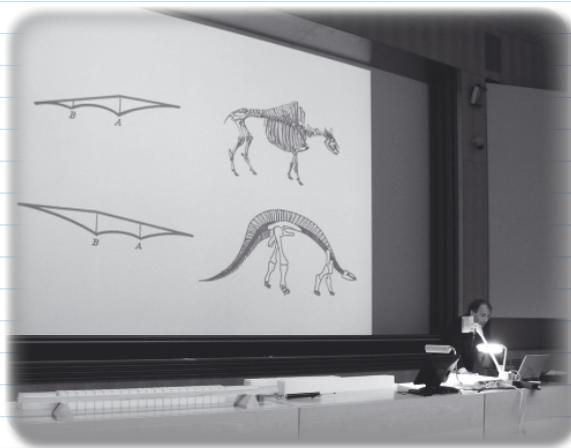


fig. 1
Cours en classe

LE PROJET i-STRUCTURES

Le projet **i-structures** a été proposé dans le cadre du second appel du Campus Virtuel, en décembre 2000. Il a été accepté au cours du printemps 2001 et a officiellement débuté le 1^{er} juillet 2001 pour se conclure le 31 décembre 2003. Il fait partie des cinquante projets de toute la Suisse et de tous les domaines de l'enseignement supérieur qui ont été sélectionnés lors des deux premières phases du Campus Virtuel.

Parmi les conditions pour être accepté dans ce programme, il était nécessaire que le projet rassemble des enseignants de plusieurs hautes écoles en Suisse. Dans le cas de ce projet, c'est tout le territoire national qui est couvert, puisque, outre l'EPFL (ENAC - IS-BETON, Prof. Aurelio Muttoni) comme leader, les partenaires sont la faculté d'architecture de l'Université de la Suisse Italienne (USI) à Mendrisio (Accademia, Prof. Massimo Laffranchi), l'Institut de statique et de construction (IBK, Prof. Edoardo Anderheggen) de l'EPFZ, la Faculté de communica-

tion de l'USI (USI-COM, Profs. Edo Poggia et Jean-François Perret), et la HES du Tessin, section informatique (SUPSI, Prof. Federico Flückiger). Ces partenaires ont tous apporté des contributions importantes au projet, mais la diversité des cultures académiques et des approches a rendu la gestion de ce projet plus délicate.

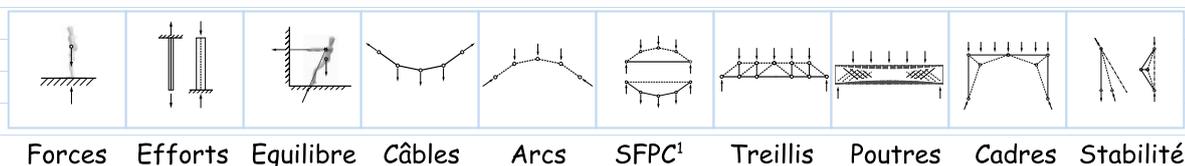
L'équipe EPFL était composée des personnes suivantes:

- Prof. Dr Aurelio Muttoni, chef de projet
- Dr Olivier Burdet, coordinateur
- Eric Tonicello, responsable de l'environnement PHP et des exercices
- Jean-Luc Zanella, responsable des bases de données et de l'authentification.

LES IDÉES-FORCES

L'idée fondamentale lors du démarrage du projet a été de transcrire dans un environnement informatisé les propriétés novatrices du cours de structures du Prof. Muttoni, en renforçant les aspects les plus novateurs et en allégeant certains aspects trop lourds ou répétitifs par l'utilisation de programmes informatiques pour la résolution graphique des structures. La résolution numérique des problèmes par la méthode des éléments finis devait permettre d'offrir une solution simple au problème non résolu dans le cadre du cours du calcul des déformations et des structures complexes. L'environnement envisagé dès le départ devait permettre une navigation simple, intuitive et refléter la nature fortement consécutive du cours, représentée par le parcours des structures, qui donne, de gauche à droite, tout le menu du cours.

fig. 2
Le parcours des structures



¹ Structures funiculaires à poussée compensée

En ce qui concerne le développement, la maintenance et la pérennité du cours, il était dès le début clair que les moyens considérables à disposition dans la phase supportée par le Campus Virtuel ne dureraient pas éternellement, et qu'en conséquence il était nécessaire que l'encadrement normalement disponible pour le cours soit à même de le modifier, de le corriger et d'en gérer tous les aspects sans faire appel à des moyens extérieurs coûteux.

Le nom i-structures rappelle bien entendu les structures, qui en sont le sujet principal. Le préfixe **i** a plusieurs significations, qui ont renforcé ce choix: *intuition, innovation, Internet, interactivité*.

COMPOSANTES DU PROJET

La réalisation des idées-forces a conduit à entreprendre presque simultanément plusieurs chantiers: définition de l'interface graphique et de navigation, environnement de rédaction du contenu, *applets* de calcul graphique et de calcul par éléments finis, environnement de sauvegarde des fichiers sur le serveur. D'autres composantes sont venues s'ajouter par la suite: contenu du cours proprement dit, gestion des renvois et du glossaire, sauvegarde des exercices et tests ainsi que les statistiques de suivi de l'utilisation.

Une analyse préliminaire des environnements standards d'enseignement en ligne nous a rapidement convaincus que l'état de ces outils (en fin 2001) ne permettrait pas un développement simple, efficace et flexible du contenu, tout en permettant de le réorganiser ultérieurement. C'est pourquoi une solution *maison*, comprenant un nombre minimal de fonctions, a été développée. Il ne s'agit pas à proprement parler d'une plate-forme, mais plutôt d'un environnement de développement évolutif, supporté par une série de scripts exécutés sur le serveur. Le but de ces scripts était de faciliter la mise en œuvre des fonctions de base du cours, tout en permettant l'insertion simple de fonctions particulières: questionnaires au milieu du texte, renvoi automatique à une partie quelconque du cours, ouverture d'*applets* avec les paramètres de l'utilisateur, etc.

ENVIRONNEMENT DE RÉDACTION

Dès le début du projet, il était clair que les auteurs du contenu (les professeurs, les assistants) devaient être au centre de l'activité de création du contenu, et ce non seulement lors de la création initiale, mais également lors des modifications ultérieures. Ces considérations étaient particulièrement importantes dans le cas du projet i-structures, puisque, dès le départ, il était nécessaire de créer un cours en deux langues en parallèle, avec une partie de la rédaction en français à traduire en italien et le restant à traduire d'italien en français. Ce processus a conduit et conduit toujours à des modifications successives du contenu. C'est pourquoi il aurait été très compliqué et coûteux de faire en plus appel à une tierce personne pour effectuer le codage des pages du cours.

La solution qui a été choisie était donc de rédiger les pages du cours en utilisant un logiciel que tout le monde sait utiliser, et qui se prête particulièrement bien à la rédaction de textes: Microsoft Word. S'ils sont bien construits, les documents Word possèdent une structure forte, qui reflète précisément l'intention de l'auteur: les styles de paragraphe définissent la fonction d'un morceau de texte, alors que l'auteur a la possibilité d'utiliser des mises en formes particulières pour le texte: gras, italique, indice ou exposant, caractères grecs, etc. L'insertion de caractères spéciaux, de symboles, d'équations, de tableaux et de figures est naturelle et simple. Malheureusement, chacun sait que la fonction **Enregistrer en tant que page Web** de Word donne des résultats inutilisables, puisque les auteurs de cette fonction ont semble-t-il voulu recréer sur le Web une image exacte de l'image imprimée depuis Word. Or ce n'est évidemment pas ce qui est recherché pour un cours en ligne: le format pdf convient bien mieux!

Par contre, sachant que l'information nécessaire au codage HTML est effectivement déjà présente dans la structure du fichier Word, il suffit de la récupérer et de créer automatiquement les fichiers HTML correspondants. Cette fonctionnalité a été créée dans Word lui-même: au lieu d'appeler la fonction standard d'enregistrement HTML, une fonction spéciale a été développée en Visual Basic pour Applications (VBA). Son utilisation est simple, le code HTML résultant est sobre et compact et la création de fichiers capables de générer l'ensemble du cours i-structures en Microsoft Word facilite la mise à jour du cours, ainsi que sa traduction en d'autres langues.

L'exemple de code en VBA donné en fin de cet article est une version simplifiée de celui qui nous utilisons. Il est capable de **générer une page HTML à partir d'un fichier Word quelconque**. Pour garder ce code le plus simple possible, il ne reconnaît que quelques styles: Titre 1, Titre 2, Titre 3, Equation (centré) et HTML (récrit le contenu du paragraphe sans modification). Tous les autres styles sont considérés comme contenant du texte normal. Ce code permet la transformation des images et autres objets insérés dans le document Word (comme objets in-line, entourés de carrés noirs dans Word et non comme objets flottants, entourés de carrés blancs). Par contre, l'identification et la traduction en HTML des mises en forme (italique, gras, souligné, couleur) ne sont pas incluses, de même que la gestion des caractères spéciaux (grec, indices, exposants) ou des tableaux. L'extension de ce code pour le transformer en un environnement complètement fonctionnel, incluant ces fonctionnalités ainsi que des commandes pour la gestion de la navigation et le découpage en plusieurs pages est laissée au soin du lecteur. L'effort de programmation pour cela n'est pas énorme.

Il s'est avéré à l'usage qu'un nombre restreint de styles permet la création des tous les contenus nécessaires au cours, y compris les questionnaires et les exercices en ligne. Lorsque le besoin se fait sentir d'ajouter une extension supplémentaire, généralement par l'inclusion d'un nouveau style, son implémentation ne pose pas de problème particulier, vu la modularité du code et la proximité entre le texte de base d'une part et le résultat au format HTML d'autre part.

Publication des pages Web

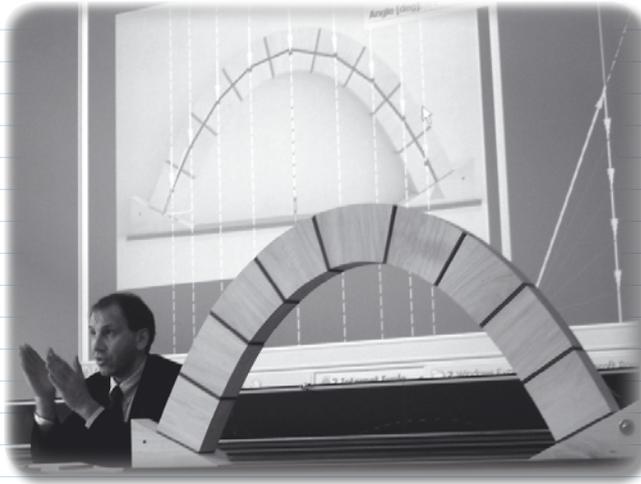
Une fois le contenu du cours traduit en HTML, il faut encore le publier sur le Web. Dans la mesure où les auteurs du cours ont pour but principal la rédaction du contenu, et non pas la mise en page ou la navigation dans le site, ces tâches sont réservées à l'environnement de publication Web. Essentiellement, il s'agit de quelques balises insérées automatiquement lors de la création du code HTML, qui appellent de courts scripts en PHP qui ont pour fonction de créer l'interface graphique, le système de navigation et les utilitaires de renvois, glossaires, commentaires, etc.

Cet environnement a été conçu dans la même optique que celui de rédaction: une fois le cours en fonction, il est possible d'ajouter ou de modifier des pages sans faire appel à un programmeur: l'environnement sait reconnaître ces nouvelles pages et créer les liens dont elles ont besoin. Les renvois à d'autres pages du cours, proches ou lointaines, se fait avec une syntaxe unique, et n'exige pas, tout comme l'appel au glossaire, de traduire ces liens: ils peuvent être repris tels quels d'une autre langue, et ils fonctionnent correctement dans la nouvelle langue de traduction. Cette fonctionnalité est particulièrement intéressante en cas de traduction du cours par une personne qui n'est pas familière avec le contenu: il lui suffit de laisser intacts dans Word les liens et autres particularités du codage, et celui-ci fonctionnera dès la traduction. En pratique, il s'est avéré que de petites corrections sont parfois nécessaires, mais d'une ampleur faible par rapport au résultat obtenu.

Applets

Le cours de structures de Mendrisio avait montré deux limites particulièrement contraignantes pour l'enseignant et les étudiants: l'utilisation de la statique graphique, qui est plus simple et plus intuitive que la statique analytique des ingénieurs, permet certes de résoudre facilement les problèmes posés, mais cette résolution prend passablement de temps. Au début du cours, alors que les étudiants apprennent eux-mêmes à maîtriser cette méthode, ce n'est pas un vrai désavantage. Plus tard, quand cet apprentissage est terminé, il reste cependant toujours nécessaire de passer par la résolution graphique pour obtenir les résultats. Le temps pris par ces résolutions est alors plutôt stérile. De plus, la résolution de variantes d'une configuration déjà étudiée n'est possible que moyennant une nouvelle résolution complète, ce qui accentue encore le côté répétitif de cette approche et limite la liberté d'exploration de l'enseignant et des étudiants.

fig. 3
Utilisation simultanée de l'applet i-Cremona et d'une maquette lors du cours en classe



Un autre désavantage de cette approche est que les résultats obtenus par la statique graphique concernent les *efforts* dans les structures et ne donnent pas d'indications sur les *déformations* qu'elles induisent. Ceci est un désavantage, car la conception des structures nécessite que leur comportement en service soit également satisfaisant, notamment avec des déformations faibles.

C'est pourquoi il a été décidé dès la conception du projet i-structures que des *applets* spécialisés de haut niveau seraient nécessaires pour d'une part faciliter la résolution des problèmes par la méthode de la statique graphique, et d'autre part pour permettre un calcul complet et exact de toutes sortes de structures

par la méthode des éléments finis. Pour être efficaces, ces *applets* devaient à la fois être *légères* pour permettre leur téléchargement depuis le serveur et inclure des fonctionnalités sophistiquées, comme l'identification des utilisateurs ainsi que le stockage et la récupération des données sur un serveur central. Evidemment, étant donné la nature même de la statique graphique, ces programmes devaient offrir une interface graphique complète pour être fidèles à la méthode elle-même.

L'utilisation d'*applets* offre les avantages que tous les utilisateurs utilisent toujours la dernière version du programme, que leur identification peut être contrôlée lors de chaque accès et que les données stockées sur le serveur sont disponibles depuis toute machine connectée à Internet, sans nécessiter un moyen manuel de transfert des données (disquette, CD, clé de données). L'aspect de moindre performance des *applets* par rapport aux programmes a été considéré, mais le taux de croissance des performances des machines rend cette question secondaire, dans la mesure où la performance actuelle des *applets* est déjà suffisante.

Finalement, les *applets* offrent en outre l'avantage de pouvoir être paramétrées pour être appelées directement depuis une des pages du cours ou des exercices avec un jeu de paramètres prédéfini par l'auteur d'une leçon. A l'usage, cette manière de faire s'est avérée très efficace pour les exercices, en limitant au maximum les problèmes d'accès aux données pour les élèves.

En l'état actuel, le cours i-structures dispose de deux *applets* Java, dont la versatilité permet de couvrir l'essentiel du contenu du cours. Ce faible nombre limite l'effort à fournir par les étudiants pour connaître le fonctionnement de ces programmes et leur permet de se concentrer sur leur application aux cas rencontrés lors de l'étude.

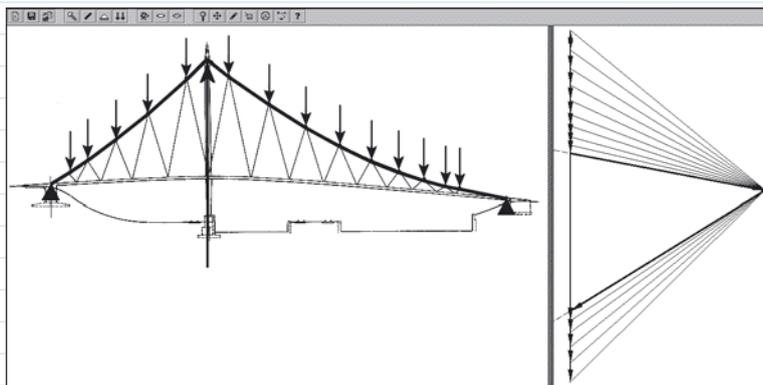
NAVIGATEUR

L'intérêt d'utiliser des *applets* Java plutôt que des composants ActiveX du monde Windows s'est vite révélé essentiel lorsque le cours de structures a reçu comme salle d'exercices les salles informatiques de la Coupole, équipées de stations de travail Sun. L'utilisation de ces machines, une fois les étudiants familiarisés avec leur mode de fonctionnement, n'a posé aucun problème de fonctionnalité ou de performance. Le navigateur utilisé sur ces machines est Mozilla. Sur PC ou Mac, les navigateurs classiques permettent de naviguer sans problème sur le site. Pour permettre l'authentification, l'activation des cookies est nécessaire; pour l'utilisation des *applets*, une machine virtuelle Java récente (> 1.4.0 est requise); pour la navigation, l'activation de JavaScript est conseillée, mais pas indispensable.

i-CREMONA

L'*applet* i-Cremona, nommée d'après le mathématicien Cremona qui en fin du XIX^e siècle a développé cette

fig. 4
Exemple d'application de l'applet i-Cremona



construction graphique pour la résolution des systèmes de treillis, permet la résolution graphique de systèmes comprenant plusieurs forces. La possibilité de résoudre simplement une série de problèmes avec des paramètres presque identiques permet à l'étudiant d'explorer de multiples solutions et d'en dériver une compréhension profonde du mode de fonctionnement des structures.

La possibilité d'utiliser un dessin ou une photographie de la structure à investiguer comme arrière-plan pour la résolution permet d'inclure des exemples réels dans l'enseignement, sans avoir à faire recours à des données trop simplifiées qui cesseraient d'être stimulantes.

i-STATIC

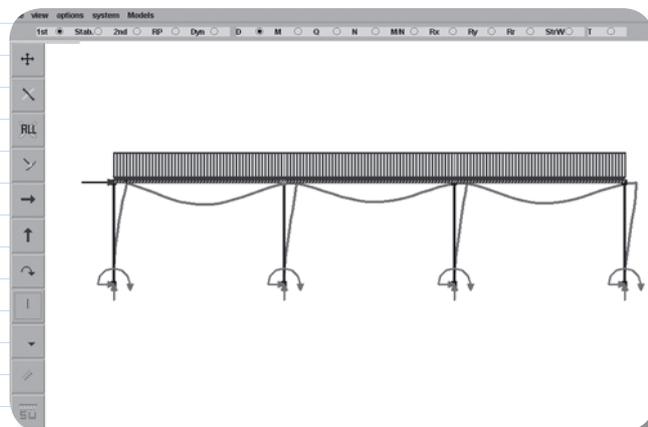


fig. 5
Exemple d'utili-
sation de l'applet
i-Static

L'applet i-Static représente un encore plus grand défi: mettre à disposition à l'intérieur d'une *applet* un programme de calcul par éléments finis de niveau presque professionnel, apte à calculer des systèmes complexes, en fait bien plus complexes que ceux que rencontrent normalement des étudiants de première année. L'idée est qu'une fois familiarisés avec cet outil, les étudiants continueront à l'utiliser pour comprendre le fonctionnement de structures simples ou compliquées qu'ils rencontreront au cours de leurs études.

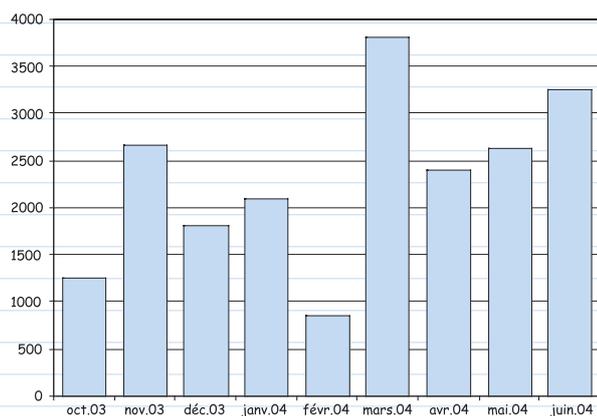
Ce programme n'a nullement le but de se substituer aux programmes professionnels de calcul des structures, tels que ceux utilisés

par les ingénieurs, mais plutôt de servir de complément à une première approche conceptuelle, aussi bien pour l'architecte que pour l'ingénieur, avec le seul but de valider ou d'infirmer la possibilité pratique de réaliser ce qui vient d'être conçu.

INFRASTRUCTURE INFORMATIQUE

L'infrastructure informatique est souvent un des composants essentiels d'un projet d'enseignement par Internet. En fait, à entendre les remarques des utilisateurs, c'est souvent également une des sources principales de problèmes. C'est pourquoi le projet i-structures a d'emblée été conçu pour éviter autant que possible ces problèmes, en adoptant une structure et des moyens de publication simples et universellement disponibles. Pour permettre une opération ne nécessitant pas l'intervention de spécialistes, l'essentiel de l'intelligence de l'environnement réside dans les quelques scripts qui réalisent la mise en page et les fonctionnalités spéciales du cours. Les auteurs ou le responsable du site ne doivent pas connaître dans le détail le fonctionnement de ces scripts, mais simplement respecter un certain nombre de règles simples dans les noms des fichiers et leur disposition.

fig. 6
Authentifications
par le système
GASPAR par mois



AUTHENTIFICATION

L'authentification est un point important de tout système d'enseignement en ligne, particulièrement lorsqu'il s'agit de suivre l'avancement des étudiants dans leurs exercices et lors des examens. Pour les étudiants de l'EPFL, c'est le système d'authentification GASPAR qui a été utilisé. L'avantage de ce système est d'être utilisé de manière générale pour l'identification des étudiants, ce qui leur évite d'avoir à mémoriser un mot de passe supplémentaire. En pratique, cet avantage s'est avéré moins déterminant que prévu dans la mesure où, en début de premier semestre surtout, les étudiants ne sont pas encore familiarisés avec

GASPAR. A l'usage, ce choix s'est cependant avéré positif, avec très peu de problèmes d'authentification. Plus de 20'000 connexions utilisant GASPAR ont été effectuées par les 300 étudiants au long des neuf mois du cours.

NAVIGATION

La navigation dans le site a été conçue de façon à donner le plus de liberté possible aux étudiants: les pages n'utilisent pas de cadres (sauf en de rares exceptions) et il est possible de placer une ou l'autre page dans ses *Favoris* pour y accéder rapidement. Il est en tout temps possible d'accéder directement à n'importe quelle page du cours. Cela nécessite au plus 4 ou 5 clics, souvent seulement un ou deux. La navigation peut se faire de manière consécutive au moyen de flèches de navigation en haut et en bas de chaque page, au moyen de la table des matières générale ou de l'index de chaque leçon, ou encore au moyen d'un lien placé dans l'une ou l'autre page par son auteur.

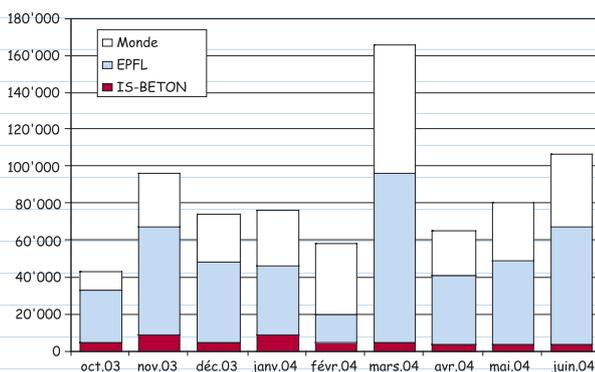
Dans tous les cas, il est possible de naviguer en arrière pour retrouver une des pages précédentes. Dans la mesure où la navigation s'appuie sur les mécanismes standard du Web, ces opérations sont aussi rapides que le navigateur utilisé le permet.

GLOSSAIRE

Le glossaire est un outil important du cours i-structures. Parce que les étudiants viennent de milieux divers, ils n'ont pas nécessairement le même niveau de connaissance du vocabulaire technique. Le glossaire permet aux étudiants de trouver rapidement le sens de mots qu'ils ignorent ou ne comprennent pas précisément. Les auteurs du cours peuvent, de manière similaire à un appel de lien distant, signaler une définition du glossaire pour un mot ou un groupe de mots donnés, avec l'effet que ce mot apparaîtra en évidence dans le texte, avec une petite fenêtre donnant sa définition lorsqu'on y passe la souris. Un clic sur le mot permet d'en voir la définition dans le glossaire et de chercher d'autres mots.

MATÉRIEL ET ENVIRONNEMENT

fig. 7
Pages accédées
par mois



Le nombre total de pages délivrées par le serveur i-structures au cours des neuf mois sur lesquels s'est déroulé le cours est d'environ 800'000 (sans compter les images), avec un maximum d'environ 40'000 pages par jour en période d'examen. Le serveur utilisé est actuellement un serveur Dell monoprocesseur Xeon 3.06 GHz, avec 1 GB de mémoire. Le serveur Web est IIS 6.0, les scripts sont en PHP 4.26 et les données sont stockées dans une base MS SQL Server. Ce système a une capacité amplement suffisante pour les cours de Lausanne et de Mendrisio. Il faut noter que le premier tiers du cours environ s'est déroulé en

utilisant un serveur PIII 1.0 GHz avec 512 MB de mémoire, qui était suffisant, même s'il était extrêmement sollicité. Dans la figure 7, la part attribuée au monde (adresses IP hors des adresses des partenaires du projet) est à comptabiliser pour l'essentiel comme des accès au site par des étudiants de l'EPFL depuis leur domicile.

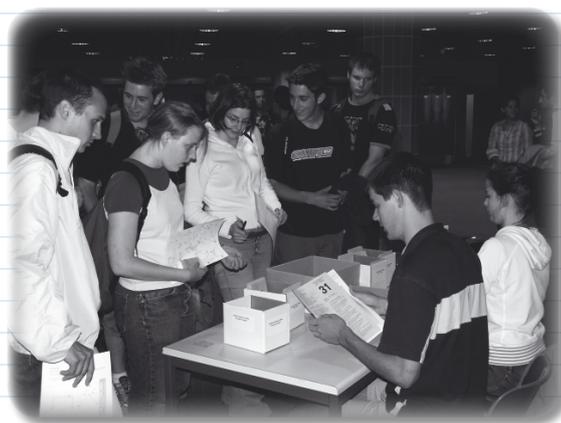
SCÉNARIO PÉDAGOGIQUE ET ORGANISATION

Dans la mesure où le cours de Lausanne était entièrement nouveau, et que l'environnement de cours n'avait pu être testé que partiellement, notamment en ce qui concernait la pleine charge induite par plus de 300 étudiants, l'approche suivie a été de proposer un enseignement ex cathedra similaire à celui qui est proposé à Mendrisio: un cours classique faisant appel à de nombreuses maquettes et démonstrations en classe. L'enseignant faisait de plus largement appel aux ressources de i-structures: *applets*, contenu en ligne, etc. Les exercices avaient lieu en ligne dans des salles informatiques qui leur sont réservées lors de la semaine. Il était bien entendu possible dès le début de faire ces exercices à un autre moment ou depuis un ordinateur quelconque, par exemple à la maison.

Dans une phase ultérieure, qui a commencé d'être appliquée au second semestre, il est prévu de remplacer certaines leçons ex cathedra par des leçons purement en ligne, où les étudiants étudient la matière de manière autonome.

EXERCICES ET EXAMENS

fig. 8
Rendu des feuilles
à la fin du test



Le fonctionnement des exercices, qui sont obligatoires mais ne comptent pas pour l'établissement de la note finale, a donné satisfaction, même si la période initiale a donné lieu à des réglages tant dans les aspects informatiques que dans les procédures de communication avec les étudiants. Les deux tests intermédiaires qui font office d'examen ont eu lieu selon le même modèle, mais avec une structure plus contraignante pour l'identification des étudiants. Malgré le grand nombre d'opérations nécessaires pour cela, le système utilisé a donné toute satisfaction et n'a pas pris un temps exagéré.

L'avantage des questionnaires en ligne est que toutes les questions de format choix multiples ou nu-

mériques peuvent être corrigées automatiquement, ce qui allège considérablement la tâche des correcteurs qui ne doivent plus corriger que les quelques questions dans lesquelles il est demandé à l'étudiant de produire un dessin ou de développer un thème. L'efficacité de cette méthode a permis de sensiblement réduire le temps consacré à la correction des exercices et examens, au profit d'une augmentation du temps d'assistantat proprement dit.

DURABILITÉ, PÉRENNITÉ, MODIFICATIONS

Comme il a été dit, la durabilité, la pérennité et la possibilité de procéder en tout temps à des modifications de contenu et même de structure du cours étaient à la base de l'approche qui a été suivie. Ces buts ont effectivement été atteints, de telle sorte que la maintenance du projet est simple et ne doit pas nécessairement faire appel à des spécialistes du Web ou de la programmation, mais peut être pour l'essentiel effectuée par les auteurs eux-mêmes, c'est-à-dire par les enseignants et leurs assistants. Les quelques moyens supplémentaires nécessaires sont aisément disponibles au sein du laboratoire IS-BETON. Il s'agira essentiellement d'assurer le passage d'une année académique à l'autre, avec l'archivage des comptes et des données des étudiants de l'année précédente, ainsi que des exercices.

Le projet **i-structures** a obtenu un soutien de la part du Campus Virtuel Suisse pour la phase de consolidation 2004-2006. Ce soutien permettra de consolider les aspects du cours qui nécessitent encore un soutien trop intense, comme la gestion des exercices et l'utilisation des *applets*, de façon que le cours puisse fonctionner, par la suite, avec les moyens normaux d'un cours.

Conclusions

Le projet **i-structures** a été l'occasion pour un grand nombre de personnes dont la plupart n'avaient aucune expérience de l'enseignement en ligne d'affronter et de résoudre les problèmes liés à la création d'un contenu destiné à être utilisé de manière autonome. La taille du projet, tant par le nombre de personnes impliquées que par le nombre d'étudiants concernés a nécessité des réflexions très approfondies tant sur le comment que sur le pourquoi d'un tel cours. Dans l'ensemble, les résultats sont satisfaisants et correspondent aux attentes initiales [1].

La réponse des étudiants à ce mode d'enseignement a en général été positive, même si certains ont exprimé des craintes quant à l'utilisation de l'informatique.

L'équipe de l'IS-BETON se réjouit d'avoir pu mettre en place le nouveau cours de structures de l'ENAC et vous invite à visiter le site à la page <http://i-structures.epfl.ch>. Les utilisateurs EPFL peuvent utiliser directement leur compte GASPAREL, les autres peuvent se créer un compte automatiquement en suivant les indications données en cliquant sur **login** depuis l'entrée du site.

Bibliographie et liens Web

- [1] **Burdet O., Laffranchi M.**, *Transposing multimedia into a virtual course*, 4th International Conference on New Educational Environments, 1.3/11-14, Lugano, Switzerland, English, **2002**.
- [2] **Muttoni A.**, *L'art des structures; Une introduction au fonctionnement des structures en architecture*, Presse Polytechnique et Universitaires Romandes, 271 p., Lausanne, Switzerland, **2004**.
- [3] Site i-structures: <http://i-structures.epfl.ch>, et http://i-structures.epfl.ch/Articles/articles_f.php pour les articles concernant le projet.
- [4] Site Campus Virtuel Suisse: <http://www.virtualcampus.ch/>

Code Visual Basic d'UNE MACRO TRANSFORMANT UN fichier Word en HTML

```

Option Explicit

'(c) 2004 Olivier Burdet: autorisation d'utiliser et de modifier ce code pourvu
' que cette notice reste présente avec le nouveau code
' Merci de vos remarques à olivier.burdet@epfl.ch

Sub toHTML() 'Routine principale effectuant la traduction de Word en HTML

Dim i As Integer, sHTML As String, texteBrut As String, leStyle As String
Dim numImages As Integer, tmp As String, nomFichOut As String
Dim racineNomsImages As String, Titre As String

Titre = ""
With ActiveDocument 'traite le document actif dans Word
    racineNomsImages = Left(.Name, InStrRev(.Name, ".") - 1)
    sHTML = "" ' contient tout le fichier HTML
    For i = 1 To .Paragraphs.Count 'boucle sur tous les paragraphes du document
        leStyle = LCase(.Paragraphs(i).Style) ' le style de paragraphe est central
        tmp = traiterPara(.Paragraphs(i), numImages, racineNomsImages, texteBrut)
        ' traiterPara fait en fait toute la traduction en HTML, sauf les styles
        If Len(tmp) > 0 Then
            If leStyle = "titre 0" Then 'style spécial pour le titre de la page
                If Titre = "" Then Titre = tmp
                tmp = "<h1>" & tmp & "</h1>" & vbCrLf
            ElseIf leStyle = "titre 1" Then
                tmp = "<h1>" & tmp & "</h1>" & vbCrLf
            ElseIf leStyle = "titre 2" Then
                tmp = "<h2>" & tmp & "</h2>" & vbCrLf
            ElseIf leStyle = "titre 3" Then
                tmp = "<h3>" & tmp & "</h3>" & vbCrLf
            ElseIf leStyle = "equation" Then 'les équations sont centrées
                tmp = "<p align=""center"">" & tmp & "</p>" & vbCrLf
            ElseIf leStyle = "figure" Then 'les figures aussi
                tmp = "<p align=""center"">" & tmp & "</p>" & vbCrLf
            ElseIf leStyle = "légende figure" Then ' les légendes de figures en italique
                tmp = "<p align=""center""><i>" & tmp & "</i></p>" & vbCrLf
            ElseIf leStyle = "html" Then ' on ne fait rien, on respecte !
                tmp = Replace(Replace(texteBrut, ""<", ""&lt;""), ">", ">"")
                'un peu brutal pour "escaper" les balises
            Else ' on pourrait ajouter ici d'autres styles à formater différemment
                tmp = "<p>" & tmp & "</p>" & vbCrLf ' les autres styles sont du texte
            End If
            sHTML = sHTML & tmp & vbCrLf
        End If
    Next
End With

sHTML = "<html>" & vbCrLf & "<head>" & vbCrLf & "<title>" & Titre & "</title>" & vbCrLf &
    & "</head>" & vbCrLf & "<body>" & vbCrLf & sHTML
sHTML = sHTML & "</body>" & vbCrLf & "</html>"

nomFichOut = Left(ActiveDocument.FullName, InStrRev(ActiveDocument.FullName, ".")) & "htm"
ecrireFichier sHTML, nomFichOut 'on écrit tout le fichier d'un coup

Shell "CMD /C "" & nomFichOut & """, vbHide ' on appelle la page web pour la visualiser

End Sub

Function traiterPara(lePar As Paragraph, numImages As Integer, racImages As String, _
    texteBrut As String) 'c'est là que la traduction se fait

Dim i, n As Integer, parRange() As Long, numRange As Long
Dim txtRange As Range, imgRange As Range
Dim nomImage As String
Dim s As String

numRange = 0
With lePar ' on se prépare à séparer le paragraphe
    ReDim parRange(1, numRange) ' entre texte et images
    parRange(0, numRange) = .Range.Start
    parRange(1, numRange) = .Range.End
End With

n = lePar.Range.InlineShapes.Count
For i = 1 To n ' boucle sur les images

```

```

numRange = numRange + 1
ReDim Preserve parRange(1, numRange) ' on stocke la position du texte entre les
With lePar.Range.InlineShapes(i) ' images
    parRange(1, numRange) = parRange(1, numRange - 1)
    parRange(1, numRange - 1) = .Range.Start
    parRange(0, numRange) = .Range.End
End With
Next
s = ""
For i = 0 To n ' boucle sur toutes les images:
    ' on commence par un bout de texte
    Set txtRange = ActiveDocument.Range(Start:=parRange(0, i), End:=parRange(1, i))
    texteBrut = txtRange.Text
    If Len(texteBrut) > 0 Then
        If Right(texteBrut, 1) = Chr(13) Then
            texteBrut = Left(txtRange.Text, Len(txtRange.Text) - 1)
        End If
    End If
    s = s & texteBrut ' c'est ici qu'il faudrait traiter gras, italique, grec, etc.
    If i < n Then
        numImages = numImages + 1 ' maintenant, on traite une image
        ActiveDocument.Range(Start:=parRange(1, i), End:=parRange(0, i + 1)).Select
        nomImage = sauverImage(racImages & Right("0" & numImages, 2))
        s = s & vbCrLf & "<img src="" & nomImage & "" border=""0"" align=""absmiddle"">"
        & vbCrLf
    End If
Next ' et on continue...
If Len(s) > 1 Then ' finalement, on nettoie un peu le résultat
    If Right(s, 2) = vbCrLf Then
        traiterPara = Left(s, Len(s) - 2)
    Else
        traiterPara = s
    End If
Else
    traiterPara = ""
End If
End Function

Function sauverImage(nomDesire As String) ' sauve une image "inline" au format GIF ou JPG
Dim nomFichHTML
Dim nom As String, nomRep As String, nomRepSup As String, nomImage As String, ext As String
Dim fs As Object, fol As Object, f As Object
nomFichHTML = ActiveDocument.Path & "\temp.htm"
Selection.Copy
Documents.Add DocumentType:=wdNewBlankDocument
Selection.Paste ' on copie l'image dans un nouveau fichier Word
' et on le sauve en utilisant... Save as HTML !
ActiveDocument.SaveAs FileName:=nomFichHTML, FileFormat:=wdFormatHTML, _
    Password:="", AddToRecentFiles:=False, _
    SaveNativePictureFormat:=False, SaveFormsData:=False, SaveAsAOCELetter:=False
nom = ActiveDocument.FullName
nomRepSup = ActiveDocument.Path
nomRep = Left(nomFichHTML, InStrRev(nomFichHTML, ".") - 1) _
    & Application.DefaultWebOptions.FolderSuffix
ActiveWindow.Close
Set fs = CreateObject("Scripting.FileSystemObject")
fs.DeleteFile nom, True
Set fol = fs.getFolder(nomRep)
For Each f In fol.Files
    ext = LCase(Mid(f.Name, InStrRev(f.Name, ".") + 1))
    If ext = "gif" Or ext = "jpg" Then
        nomImage = f.Name
    End If
Next
nomDesire = nomDesire & LCase(Mid(nomImage, InStrRev(nomImage, "."))) 'on renomme l'image
fs.CopyFile nomRep & "\" & nomImage, nomRepSup & "\" & nomDesire
fs.DeleteFolder nomRep ' et on fait le ménage
sauverImage = nomDesire
End Function

Sub ecrireFichier(s As String, nomFich As String) 'écrit un fichier texte contenant "s"
Const nFich = 1
Open nomFich For Output As #nFich
Print #nFich, s
Close nFich
End Sub ■

```