



Rapport de projet de Génie Logiciel 1^{re} phase
Master 2 Pro. Ingénierie de la Communication Personne-Systèmes

ANIMATHS

UN ENVIRONNEMENT WEB INTERACTIF POUR LA MANIPULATION DIRECTE ET L'ANIMATION DE TRANSFORMATIONS D'OBJETS MATHÉMATIQUES

Maxime Lefrançois & Édouard Lopez

UPMF IC2A – Master 2 Pro. Ingénierie de la Communication Personne-Système

Date de soutenance : 11 janvier 2010

Projet encadré par Mohamed El Methni

Enseignant Chercheur – Chargé de mission TICE de l'Université Pierre-Mendès-France.

TABLE DES MATIÈRES

Contexte du projet.....	4
Introduction	4
Cadre du projet.....	4
Les objectifs du produit AniMaths	5
Expression du besoin	5
Contexte économique et d'exploitation du produit	6
Exigences sur le produit.....	7
Capacités fonctionnelles.....	7
Description des fonctionnalités.....	7
Interopérabilité	9
Sécurité.....	9
Exigences non fonctionnelles.....	10
Fiabilité	10
Facilité d'utilisation	10
Rendement.....	10
Maintenabilité	10
Portabilité.....	11
Recherche de solutions techniques	12
Représentation des équations	12
LaTeX	12
Flash.....	13
SVG.....	13
MathML	14
Plate-formes d'utilisation.....	15
Navigateur(s).....	16
Mozilla Firefox.....	17
Technologies de manipulation	17
JavaScript	17
Présentation de Google Web Toolkit.....	18

Environnement de travail	19
IDE.....	19
Collaboration	19
Guide des bonnes pratiques d'après Google I/O '09	21
Paradigme Modèle-Vue-Animateur (MVP)	21
GIN (GWT INjection).....	22
Conception et début du développement	23
Architecture du projet	23
Implémentation des éléments MathML	24
Représentation objet cohérente des objets mathématiques.....	24
Fonctionnalités implémentées	25
Conclusions – ouvertures	26
Ouvertures.....	26
Bilan personnel.....	26
Maxime :.....	27
Édouard :.....	27
Références.....	28

CONTEXTE DU PROJET

INTRODUCTION

Nous nous sommes souvent imaginés un environnement virtuel pour manipuler des équations mathématiques... passer un terme à gauche et voir le signe changer ; voir un développement ou une factorisation s'opérer sous forme d'animation ; etc...

Nous pensons que les technologies modernes du web auxquelles nous sommes formés en M2 ICPS nous permettent largement de réaliser un tel environnement d'interaction : MathML/OpenMath; Technologies XML (SVG pour les graphiques animés etc...); Java Server Faces...

De plus, nous avons trouvé sur le web un certain nombre de composants logiciels libres écrits en JAVA qu'il serait intéressant d'étudier et de reprendre : éditeurs d'équations en MathML ; moteurs de résolution algébrique. D'un autre côté, à notre connaissance, les tentatives existantes d'applications mathématiques réellement interactives se limitent soit à des applets Java interactives mais trop spécialisées, soit à un formulaire simple où l'apprenant doit entrer sa réponse à une question, soit dans le meilleur des cas, à des environnements de conception d'équations couplés à un moteur algébrique qui permet de vérifier égalités et équivalences...

L'idée du projet AniMaths serait de proposer une interface web où un utilisateur peut manipuler directement les objets mathématiques. L'apprenant se retrouverait alors face à un genre de casse-tête (puzzle mathématique) qu'il devrait manipuler pour trouver la solution au problème. Les élèves pourraient alors comprendre par l'exemple et acquérir par imitation quelles manipulations les mathématiques l'autorisent à effectuer sur une expression ou une équation. Nous aimerions aussi intégrer la possibilité d'animer les manipulations d'objets mathématiques.

Dans ce projet, on propose une représentation arborescente des manipulations sur un objet mathématique, où l'ensemble des nœuds représentent l'ensemble des expressions égales (ou des équations équivalentes), et où chaque arc représente l'utilisation d'un axiome, ou d'une opération élémentaire.

CADRE DU PROJET

Ce projet de Génie Logiciel prend place dans le cadre du Master 2 Ingénierie de la Cognition, de la Création, et des Apprentissages (IC2A) spécialité Ingénierie de la Communication Personne-Système (ICPS) de l'Université Pierre-Mendès France (UPMF) de Grenoble.

Il est encadré par Mohamed El Methni et prend place dans le service Technologies de l'Information et de la Communication adaptées à l'Education (TICE) et de l'Enseignement à Distance (EAD) de l'UPMF¹.

¹ Organisation TICE-EAD:
http://tice.upmf-grenoble.fr/75800044/0/fiche___pagelibre/&RH=TICEFR_1&RF=TICEFR1_1.

Ce service est chargé de diffuser la politique TICE de l'établissement au sein des composantes et de coordonner l'enseignement à distance à l'UPMF. Il pilote les projets inter-universitaires TICE du site (GU), de la Région Rhône-Alpes et nationaux (FIED, SDTICE MESR).

Les missions du Service TICE-EAD sont les suivantes :

- Favoriser l'innovation pédagogique en proposant une infrastructure matérielle et logicielle favorisant l'émergence de nouveaux projets et la création de contenus pédagogiques en ligne ou en accompagnement des enseignements présentiels (présentiel enrichi) ;
- Aider les enseignants à se familiariser avec l'usage des Technologies de l'Information et de la Communication (TIC) et les appuyer dans leur mise en pratique ;
- En réponse aux besoins identifiés, trouver des solutions, soit au sein de l'UPMF, soit avec les autres universités du site (GU), les universités françaises ou étrangères avec lesquelles nous partageons des projets communs ou en faisant appel à des ressources externes ;
- Maintenir, moderniser et développer l'Enseignement À Distance (EAD) ;
- Développer les actions nationales de la Sous Direction des Technologies (SD TICE)

Le Service TICE-EAD est responsable du développement à l'échelle nationale de :

- C2i Niveau 1 et 2 ;
- Universités numériques thématiques (Université Ouverte des Humanités, Université Numérique Juridique Francophone) ;
- Canal U, la vidéothèque numérique de l'Enseignement Supérieur.

LES OBJECTIFS DU PRODUIT ANIMATHS

EXPRESSION DU BESOIN

Certains apprenants du secondaire présentent de graves lacunes en mathématique. Celles-ci sont d'origine multiples et peuvent les handicaper dans la poursuite d'études dans les filières scientifiques. Parmi ces lacunes, on retrouve souvent une incompréhension des transformations sur des expressions ou des équations. Ces notions sont abordées progressivement au collège, et revues et complétées au lycée.

On pense que les professeurs leur montrent plusieurs fois pendant le cours, mais que dans certains cas l'assimilation n'est que partielle, voire inexistante. Une fois seul devant un exercice, une méconnaissance des transformations mathématiquement admissibles est source de blocage.

D'un autre côté, nous souhaiterions apporter aux enseignants du secondaire qui ont du matériel informatique de présentation dans leur salle, un outil pour représenter des transformations par des animations. Cela permettrait de limiter l'impact des explications verbales trop abstraites pour l'apprenant.

On pense que les apprenants ont besoin d'un guide pour voir quelles manipulations les mathématiques les autorisent à opérer sur des expressions ou sur des équations, et qu'ils ont besoin d'outils d'animation pour visualiser ces transformations et pour acquérir des automatismes.

D'autre part, les programmes poussent de plus en plus les enseignants à animer des activités informatiques avec les élèves, ce logiciel serait une aubaine pour eux. On pense que les enseignants pourraient également apprécier ces mêmes outils de visualisation pour faciliter les explications des manipulations mathématiques, et pour mener des démonstrations sans risquer de noyer les apprenants dans des explications verbeuses qui peuvent leur paraître abstraites.

A terme, le projet AniMaths se veut de répondre à ces deux besoins. Pour cela, nous proposons de concevoir et de développer un environnement web interactif pour la manipulation directe d'objets mathématiques, et l'animation rigoureuse de transformations mathématiques.

CONTEXTE ÉCONOMIQUE ET D'EXPLOITATION DU PRODUIT

Nous ciblons deux publics très différents:

- Les apprenants : nous nous limiterons dans le cadre de ce Master à cibler les apprenants du secondaire, niveau collège (ce qui sera peut-être précisé au fil de la réalisation du projet). A terme, nous aimerions que le produit cible l'ensemble des apprenants du secondaire, qu'il prenne en compte leur niveau de connaissances théorique et évalué, qu'il s'adapte à ce niveau et qu'il accompagne les apprenants dans leur progression
- Les enseignants : nous souhaitons permettre aux enseignants du secondaire de faire des manipulations directes devant des élèves, et peut-être d'enregistrer une séquence de manipulations pour présenter une démonstration devant la classe.

L'internationalisation du produit ne fait pas partie de nos objectifs dans le cadre de cette année, mais nous essaierons de rendre son implémentation future plus facile par l'utilisation de bonnes pratiques de conception.

Le produit se positionne sur le marché du logiciel libre, et gratuit ; Nous rejoignons la philosophie de l'Université Pierre Mendès France, qui sera aussi propriétaire de notre travail, et qui impose que le produit fini soit libre et gratuit.

EXIGENCES SUR LE PRODUIT

Comme ce projet vient de naître, nous n'avons pas encore de cahier des charges très contraignant. Nous proposons néanmoins dans cette partie une expression qualitative des exigences, qui seront flexibles et que l'on pourra classer dans deux catégories : celles qui sont souhaitées à la fin de la seconde phase, et celles que l'on envisage en cas de continuité du projet.

Globalement, nous nous limitons dans ce projet à la manipulation d'objets mathématiques appartenant à un *corps commutatif*.

Certaines des exigences décrites ci-dessous ont d'abord fait l'objet de recherche des solutions technologiques adaptées.

CAPACITES FONCTIONNELLES

DESCRIPTION DES FONCTIONNALITES

Pour le moment, on souhaite intégrer les fonctionnalités :

- permettre la manipulation directe d'objets mathématiques
 - Ne jamais autoriser l'affichage d'erreurs mathématiques

Nous nous limiterons aux manipulations suivantes :

- transformations possibles sur les équations
 - addition ; soustraction ; multiplication ; division ;
 - passage d'un terme additif d'un membre à l'autre ;
 - passage d'un terme multiplicatif d'un membre à l'autre...
- transformations possibles sur un bloc additif
 - regroupements (commutation ; association) ;
 - calculs d'addition pour des nombres...
- transformations possibles sur un bloc multiplicatif :
 - regroupements (commutation ; association) ;
 - factorisation ; développement ;
 - calculs de multiplications pour des nombre...
- transformations possibles sur les fractions :
 - réduction ; mise au même dénominateur...

Ces manipulations devront être rendues possibles par l'utilisation de :

- fonctionnalités de sélection :
 - sélectionner un objet ou un groupe d'objets, joints ou disjoints de manière cohérente (c'est à dire seulement si il existe des manipulations possibles sur ces objets) ;
 - changement de sélection à l'aide de touches du clavier ou de boutons ;
- fonctionnalités d'interaction par le biais de barres d'outils et de menus contextuels, de boîtes de dialogue :
 - exemple de fonctionnalités accessibles par menu contextuel : pour interagir avec la sélection (ex : "développer" ; "sortir de la parenthèse" ; "factoriser par ...?")
 - exemple de fonctionnalités accessibles par des boîtes de dialogue (ex : "qu'est-ce qui est en commun entre les différents termes ?")
- fonctionnalités de glisser-déposer: on clique, on choisit un niveau de sélection, puis on glisse-dépose le terme ailleurs, et le système propose une nouvelle expression du type (resp. équation) égale (resp. équivalente))
 - on ne peut glisser qu'un objet préalablement sélectionné
 - lorsqu'on glisse un objet, on doit avoir un retour informatif sur là où il est possible de le déposer.
- permettre la visualisation de transformations mathématiques par le biais d'animations
 - suivant si la fonctionnalité est activée, animation permet de revisualiser la manipulation qui vient d'être effectuée ;
 - représentation unique des transformations par des flèches ; utilisation de couleurs...
- une liste des transformations opérées accompagnées de descriptions, et avec la possibilité de rejouer l'animation d'une ou plusieurs des étapes.
 - génération automatique de texte (explication rigoureuse des transformations, par exemple :

Je multiplie les deux membre par $(-1/7)$
Je divise à droite et à gauche par (-7)
La parenthèse a $(+2)$ pour facteur, donc si je sort $(+x)$ de la parenthèse, j'obtiens $(+2*x)$
- l'édition d'objets mathématiques
 - assurer la consistance de l'objet mathématique
 - limiter les possibilités d'édition en fonction des notions abordées dans le logiciel
- nous nous limiterons à la résolution de polynômes du premier degré pour le moment.

Si nous arrivons à implémenter l'ensemble de ces fonctionnalités avant la fin de la seconde phase, ou si le projet venait à être poursuivi, nous proposerions les fonctionnalités :

- tutoriel pas-à-pas pour découvrir l'environnement de manipulation des objets mathématiques.
- autres manipulations mathématiques, incorporation des puissances, des fonctions, de la théorie des ensembles, traitement de la logique...

- adapter les possibilités d'interaction et la granularité des actions élémentaires au niveau de l'apprenant
- suivre l'évolution du niveau et des compétences de l'apprenant
- proposer aux enseignants de participer à la conception de modules de formation (cours et évaluation)
- intégrer une base de connaissances comportant
 - une modélisation des apprenants
 - une modélisation des domaines d'enseignement ciblés
 - des modules de formation
- intégrer un moteur algébrique de raisonnement
 - vérifier
 - représenter des manipulations solutions pas-à-pas
- élargir les applications à différentes matières scientifiques (physique, mécanique...)

INTEROPERABILITE

Il n'est pas question pour le moment d'entrées-sorties des données vers d'autres programmes. Nous ne formulerons donc aucune exigence vis-à-vis de l'interopérabilité du produit, mais nous partons sur l'idée que le produit doit être le plus portable possible, et disponible sur un maximum de plate-forme. Ce point sera abordé dans la partie sur la plate-forme d'utilisation et les technologies utilisées.

SECURITE

Pour le moment, aucune donnée concernant les utilisateurs ne sera stockée, donc nous ne formulerons aucune exigence d'un point de vue de la sécurité des données.

Cela peut être amené à évoluer suivant l'avancement de notre travail. En effet, si l'on souhaite créer une communauté pour soutenir notre projet, il sera important de mettre en place des outils de type forum, wiki, base de données d'exercices, etc. Pour assurer un minimum de contrôle sur les contenus soumis, il serait nécessaire que les utilisateurs s'identifient.

EXIGENCES NON FONCTIONNELLES

FIABILITE

Si ce n'est que des questions de rigueur mathématiques (les équations doivent rester équivalentes, les expressions doivent rester égales), il nous faudra également s'assurer que le service est fonctionnel et disponible aux utilisateurs. L'aspect fonctionnel devra être assuré par un ensemble de tests, quand à la disponibilité elle devra être assurée par le service responsable du serveur, *i.e.* probablement un service rattaché au département TICE-EAD.

FACILITE D'UTILISATION

Le programme s'adresse à des utilisateurs novices, nous nous attacherons donc si nous avons le temps, à la définition d'aides contextuelles, de tutoriels pas-à-pas, et nous proposeront des menus et des raccourcis pour permettre une interaction plus rapide avec l'environnement de manipulation.

RENDEMENT

Pour le moment, nous pouvons simplement exiger que le système ne devienne pas trop lent lorsque l'équivalent de quelques classes d'apprenants se connectent ensemble (disons 60 connexions à la louche).

Cette exigence n'influe pas véritablement sur la conception du programme, mais plutôt sur le choix du serveur qui l'hébergera.

Nous verrons que la plupart des interactions possibles seront codées en javascript dans un script que chaque utilisateur téléchargera automatiquement lors de la première connexion, les communications avec le serveur seront donc limitées.

MAINTENABILITE

Nous voulons développer une application web, aucune exigence n'était imposée au départ, mais nos recherches nous ont poussées à nous orienter vers la boîte à outils GWT (Google Web Toolkit), et le respect des guides des bonnes pratiques (*Best Practises*) décrites lors de la conférence des développeurs Google de 2009 (Google I/O '09, voir plus bas dans ce document).

En effet, il nous semble important que la structure du programme puisse être comprise rapidement pour qu'il soit repris par d'autres équipes de développeurs, qui pourraient vouloir ajouter de nouvelles fonctionnalités. Le respect des guides bonnes pratiques devraient nous permettre de prendre en compte ce besoin de modularité.

Les raisonnements et les orientations que nous avons prises seront détaillés dans le chapitre suivant.

PORTABILITE

Nous avons rapidement choisi de se baser sur Java pour notre projet, et de placer le code sous licence Apache 2.0 qui est une licence libre et permissive.

Le choix d'utiliser à la fois équations mathématiques, graphiques et animations est forcément un facteur limitant à la portabilité de notre programme. Nous devons utiliser des recommandations du W3C qui semble être trop récentes et mal implémentées dans nombre de navigateurs web actuels.

Nous verrons donc dans le chapitre suivant que seul un navigateur existant permet l'utilisation de MathML (équations) et de SVG (images vectorielles) pour notre utilisation.

Pour ce qui est des animations et des interactions, elles devront être codées en JavaScript.

Néanmoins, nous veillerons à ce que les modifications à apporter au programme pour qu'il fonctionne sur les autres navigateurs (lorsqu'ils proposeront la gestion native du MathML) soient minimales.

RECHERCHE DE SOLUTIONS TECHNIQUES

Nous éliminons dès à présent les supports statiques (livres, télé, simple contenu HTML). Nous avons choisi une orientation web :

- Nous pensons que les technologies modernes du web auxquelles nous sommes formés en M2 ICPS nous permettent de réaliser un bon environnement d'interaction. Nous avons cherché du côté des technologies XML (MathML pour les équations, SVG pour les graphiques animés, etc...) ; et des plate-formes Java de développement web (Java Server Faces...)
- Pour une meilleure visibilité (atteindre davantage de monde par le télé-travail et les multiplate-forme)
- Pour la robustesse autorisée par l'informatique dans les nuages (*cloud-computing*),

REPRESENTATION DES EQUATIONS

Nous avons différentes possibilités pour l'affichage d'équations et de graphiques. Voici un descriptif de certaines d'entre elles et les raisons de notre choix de les conserver ou non :

LATEX

Bien connu dans le milieu de la recherche pour l'élaboration de rapport ainsi que pour sa qualité de rendu, LaTeX s'avère inadapté à notre projet car les documents qu'il génère sont :

- *sémantiquement pauvre*, bien que ce soit un langage à balise la sémantique n'est pas suffisamment fine au niveau des formules mathématiques et elle est inaccessible à l'utilisateur dans le document final (PDF ou images).
- *difficilement manipulable*, LaTeX ne dispose pas d'API pour manipuler des formules mathématiques et les objets qui la compose. Il existe bien des applications permettant de générer des images à partir de formule LaTeX, notamment [L2P](#) ou [latex2png](#), mais l'interaction que nous souhaitons fournir aurait nécessiter la génération d'un nombre d'images bien trop grand et trop complexes à déterminer.

FLASH₂

Adobe Flash est une plate-forme multimédia permettant la manipulation d'objets vectoriels ainsi que leur animation via une IDE très évoluées. Si le *rendu d'objets vectoriels est excellent*, le problème de cette plate-forme réside dans l'absence de langage de représentation des formules mathématiques ; on retombe sur les mêmes difficultés que LaTeX : *pas de sémantique native*, de ce fait il nous aurait fallu élaborer une librairie de rendu visuel des équations, ainsi qu'une librairie d'animation des divers objets.

De plus, les contenus générés par Adobe Flash sont *difficilement accessibles et manipulables* hors de l'environnement du lecteur flash. Une autre critique se porte sur le fait que Adobe flash est une *technologie propriétaire* qui va à l'encontre de l'ouverture du web et pour laquelle les rares alternatives libres sont encore peu fiables.

Le dernier point et certainement le plus important, nombre d'universités et d'écoles *bloquent les contenus flash*. De ce fait notre application ne pourrait pas être utilisée en cours, ce qui va totalement à l'encontre de nos objectifs.

SVG₃

SVG (Scalable Vector Graphic), est une recommandation du W3C. Elle permet un *rendu vectoriel* de qualité, mais comme le Flash, le rendu des formules mathématiques nécessiterait l'élaboration d'une librairie de rendu.

Contrairement à Flash, SVG est un *standard ouvert*, facilement intégrable dans un document web et totalement accessible ; c'est un langage à balise de type XML, qui possède donc une sémantique intrinsèque, mais c'est une *sémantique orientée vers le graphisme* (SVG signifie Scalable Vector Graphic).

SVG est donc parfaitement adapté à l'affichage de graphiques. Pour ce qui est de l'animation de graphiques SVG, c'est possible soit par le biais de fonctions JavaScript, soit directement en se basant sur la recommandation SMIL 2.0 du W3C. Cette recommandation est implémentée dans Internet Explorer mais pas encore dans Mozilla Firefox⁴. SVG est un standard donc l'implémentation est un sujet actif dans les différentes communauté de navigateurs web. Le support des animations est prévu pour le future⁵ et peut aisément être contourné avec des librairies JavaScript^{6 7}.

SVG sera utiliser pour le rendu et l'animation éventuel des graphiques, mais pas pour le rendu et la manipulation des équations elles-mêmes.

² Adobe Flash @ Wikipedia: http://en.wikipedia.org/wiki/Adobe_Flash

³ SVG, Recommandation W3C: <http://www.w3.org/TR/SVG11/>

⁴ Mozilla SVG Statut: <http://www.mozilla.org/projects/svg/status.html>

⁵ procheSupport de SVG:

<http://a.deveria.com/caniuse/#agents=All&eras=All&cats=SVG&statuses=rec,pr,cr,wd,ietf>

⁶ Raphaël: a JavaScript API for SVG: <http://dev.opera.com/articles/view/raphael-a-javascript-api-for-svg/>

⁷ YUI 2: Animation: <http://developer.yahoo.com/yui/animation/>

MATHML

MathML (Mathematics Markup Language) est une recommandation du W3C pour la formulation de document contenant des formules mathématiques il est donc ouvert et utilisable par tous. Étant basé sur XML, le MathML en hérite les propriétés suivantes :

- intégration web simple,
- flexibilité des contenus,
- validations des documents,
- API de manipulation DOM,
- sémantique intrinsèque.

On constate que par son appartenance à la famille XML, il est possible d'utiliser conjointement MathML, SVG et XHTML. Chacun de ces langages pouvant jouer le rôle d'hôte pour les deux autres via l'utilisation d'espaces de noms. Les contenus décrits par MathML peuvent également aller de la simple addition à de complexes résolutions de systèmes, cette possibilité de changement de complexité (*i.e.* différents niveaux scolaires) est un aspect essentiel du projet. Il serait donc possible aux enseignants d'utiliser notre framework comme outil d'accompagnement tout au long du parcours scolaire de l'apprenant (cycle collège, lycée, etc...).⁸

Parler de validation est quelque peu abusif, car il ne s'agit que de la bonne forme du document (respect de la syntaxe), et de la validité face à une DTD ou un schéma. Et ceci par l'intermédiaire d'un processeur XML, *en aucun cas cela nous permet de valider une formule !* L'API DOM du W3C pour la représentation et la manipulation des documents a été étendue à MathML, il est donc possible d'avoir des contenus manipulables. Le framework de manipulation des équations sera décrit plus tard dans le projet.

Ici la *sémantique est orientée vers les mathématiques* et donc totalement adaptée à notre besoin, ce qui nous facilitera la manipulation des formules et des documents lors de l'exécution. Le statut de recommandation du W3C, confère à cette technologie une visibilité non négligeable, on constate qu'elle est très largement supportée par les différents acteurs du domaine.⁹

Pour ce qui est du rendu visuel, on verra qu'il est variable en fonction de la plate-forme de travail (moteur de rendu, CSS). Ceci nous oblige, à l'heure actuelle, à poser des contraintes d'utilisation au client (navigateur web), ce qui va à l'encontre de la philosophie du projet. Mais nous parions sur une amélioration de la situation, grâce notamment, à l'élaboration de la fonte STIX soutenue par de nombreuses associations scientifiques (American Mathematical Society, etc.)^{10 11}.

⁸ XHTML+MathML+SVG: <http://www.w3.org/TR/XHTMLplusMathMLplusSVG/>

⁹ MathML software list: <http://www.w3.org/Math/Software/>

¹⁰ Profile MathML pour CSS : <http://www.w3.org/TR/mathml-for-css/>

¹¹ STIX project page: <http://www.stixfonts.org/>

Critères Technologies	intégration web	Sémantique	Animation	Rendu visuel	autre
LaTeX	aucune	aucune	difficile	excellente	-
Flash	Faible	aucune	excellente	complexe	propriétaire
SVG	Forte	inadéquate	librairie	complexe	standard W3C
MathML	Forte	adaptée	librairie	excellente	standard W3C

Tableau: récapitulatif des différentes technologies envisagées pour l'affichage d'objets mathématiques, et des critères de décisions associés.

PLATE-FORMES D'UTILISATION

Les objectifs du projet nous ont guidés dans le choix de la plate-forme de travail ainsi que dans le choix de l'environnement de travail des utilisateurs. L'objectif du projet étant d'offrir des possibilités de manipulation nouvelles aux usagers, nous ne pouvions bien évidemment pas envisager les médias classiques livres ou support de cours statique. Le fait que nous souhaitons développer un framework réutilisable nous a encouragé à aller voir du côté des technologies libre. C'est pour cela que nous sommes orienté vers les langages web et Java.

La situation un peu plus claire, nous en sommes venu à discuter de quelle manière était il le plus judicieux de servir le contenu. Nous avons proposé en premier de faire une applications locale, mais cette configuration nous faisait perdre l'avantage de Java et des langages web, à savoir d'être multiplate-forme. De plus, une application coté client, aurait demandé aux utilisateurs de l'installer et de la configurer, aux développeur (nous) de fournir des mécanismes de mise à jour et de synchronisation avec les dernières versions du framework et des exercices.

Nous nous sommes donc orienté vers un système centralisé. La encore nous avons hésité entre plusieurs options : servir des applets Java, utiliser XUL¹² ou rester dans un environnement totalement web. L'utilisation d'un applet Java suppose d'avoir installé le plugin correspondant et isole le contenu de cette dernière du reste de la page. XUL Faces quand à lui est limité à Firefox, donc trop limitant. L'utilisation de contenu web natif nous ramène a un environnement intégré et cohérent, qu'il serait possible de réutiliser de façon ultérieur sans avoir à développer toutes les méthodes pour cela (exportation, inclusion, etc.).

¹² FacesXUL Faces: <http://xulfaces.sourceforge.net/>

Nous avons donc décidé de travailler à l'intérieur d'un navigateur web et donc d'en réutiliser toutes les possibilités, à savoir : support MathML, JavaScript, DOM, application serveur, mise à jour factorisées possibilité de base de données d'exercices, etc.

NAVIGATEUR(S)¹³

À ce moment, nous avons rencontré quelques difficultés inattendues. Bien que MathML soit une recommandation du W3C, les différents navigateurs du marché le supportent de façon plus ou moins poussée : Internet Explorer ne le supporte que par l'intermédiaire du plugin MathPlayer, Firefox et Opera semblent avoir un support natif, quant à Google Chrome et Safari ils semblent avoir un support moins avancé que Firefox ou Opera. Nous détaillons ci-dessous notre recherche d'information sur le sujet.

La consultation des différents centres de documentation, de divers blogs, wiki nous a rapidement montré qu'Internet Explorer n'est pas capable de rendre un document MathML de façon native. Pour visualiser des documents de ce type, il est nécessaire d'ajouter le plugin MathPlayer. Ceci est regrettable car Internet Explorer est le navigateur web dominant, et le support de MathML reste encore incertain dans les prochaines versions. On espère que comme pour CSS, la pression de la concurrence poussera Internet Explorer à implémenter ce standard.

L'utilisation de la suite de test du W3C¹⁴ nous a permis par la suite de mieux identifier les lacunes des différents acteurs. Il s'est avéré rapidement que Chrome avait un support limité aux expressions MathML simples ; les expressions plus complexes révélèrent rapidement les lacunes au niveau des parenthèses et autres éléments. Or les parenthèses sont essentielles pour les manipulations les plus simples (factorisation, groupage, etc.). La situation reste là aussi confuse pour l'avenir. Il nous faut donc considérer que Chrome ne sera pas compatible avec notre framework, ce qui est dommage quand on sait que c'est maintenant le 3^e navigateur le plus utilisé.

Opera semble avoir un support plus avancé depuis la version 9.5. Le rendu est correct, de nombreuses expressions complexes sont rendues correctement. Mais comme certaines informations l'indiquent, ce n'est pas un réel support de MathML mais un contournement par l'intermédiaire d'un profil CSS¹⁵. Cette approche risque d'entrer en conflit avec les manipulations que nous souhaitons faire et s'avère incomplète¹⁶. Il nous faut donc écarter Opera des clients utilisables. Heureusement le support de MathML est prévu pour la prochaine mouture du navigateur.

¹³ Support MathML, aperçu rapide :

<http://a.deveria.com/caniuse/#agents=All&eras=All&cats=DOM,Other&statutes=rec,pr,cr,wd,ietf>

¹⁴ MathML test suite @ W3C: <http://www.w3.org/Math/testsuite/>

¹⁵ Opera MathML via CSS: <http://dev.opera.com/articles/view/can-kestrels-do-math-mathml-support-in/>

¹⁶ Opera CSS insuffisant: <http://golem.ph.utexas.edu/~distler/blog/archives/001588.html>

MOZILLA FIREFOX

Les principaux acteurs du marché des navigateurs web on était éliminés progressivement, nous commençons à questionner la viabilité de notre approche par le web. Heureusement il s'est avéré que Mozilla Firefox, le second navigateur web en part de marché¹⁷, supporte MathML dans sa version 3.5, et de façon poussée. En effet, que ce soit pour les expressions simple ou complexes le support est tout à fait correct. Et bon nombre d'indicateurs cité précédent recoupe nos observations et place Firefox en première place pour le support de MathML.

Outre son support de MathML, Firefox 3.5 dispose d'une extension permettant la création et l'édition de document mathématique, appelé FireMath¹⁸. Ceci pourrait nous permettre de créer rapidement des formules pour faire des tests, permettra aux enseignants utilisateurs de notre projet de créer facilement leur propre document MathML en restant dans un environnement unifié. Et donc minimiser les mauvaises surprises. Bien évidemment la situation n'est pas parfaite et les tests que nous avons menaient nous montre que la manipulation de MathML à l'aide de CSS est problématique. En effet, le profile pour formater les documents MathML avec CSS vient tout juste d'atteindre le stade de *Candidate Recommendation*. Il est donc normal que nous ayons constatés des lacunes de ce coté là.

Ce problème nous permet finalement de nous orienter vers la solution finale. Les tests précédent nous on montrer que si le CSS était problématique, il était tout à fait possible de faire les manipulations avec du JavaScript. Rappelons que nous souhaitons utiliser Java et les technologies web. Des recherches rapides nous amenèrent à considérer et adopter la boite à outils Google Web Toolkit (GWT), qui nous permet de générer des contenus web (MathML, JavaScript) avec une programmation en Java. Nous détaillons plus tard dans ce document ce qu'est GWT.

TECHNOLOGIES DE MANIPULATION

Travaillant avec des documents MathML, qui possède l'API DOM il nous est donc possible de gérer l'interactivité avec des technologies web comme JavaScript. De façon à réduire la charge de travail et à avoir un framework robuste, nous avons envisager d'utiliser des librairies tierces plutôt que d'écrire notre propre librairie de gestion. Nous avons donc considéré des librairies JavaScript comme JQuery, Prototype, JSmath, etc.

JAVASCRIPT

Pendant cette première phase de projet, nous avons procédé à des essais de quelques scripts pour tester l'implémentation de MathML, SVG et des animations en SVG dans différents navigateurs, e qui a nous a permis de se rendre compte que l'implémentation de l'arbre DOM et des propriétés des éléments spécifiés dans la recommandation du W3C sont loin d'être parfaitement implémentés dans Mozilla Firefox 3.5. Nous avons de ce fait longtemps cherché à trouver des moyens pour contourner ces limitations, par exemple, il

¹⁷ Part de marché des navigateurs: http://en.wikipedia.org/wiki/Usage_share_of_web_browser

¹⁸ Site officiel de Firemath: <http://www.firemath.info/>

est impossible d'ajouter un écouteur sur un élément MathML ; il est difficile de récupérer la position et la taille des éléments...

Lors de cette période de tâtonnement nous avons compris que l'utilisation de JavaScript rend très compliquée la manipulation correcte des notions de programmation orientée objets avec classes, héritage etc... C'est ce qui nous a poussé à abandonner l'idée d'utiliser des frameworks Java de développement Web tels que Struts, ou Java Server Faces, et à envisager la solution GWT (Google Web Toolkit).

PRESENTATION DE GOOGLE WEB TOOLKIT¹⁹

Cette section est basée sur l'article de Wikipedia sur GWT.²⁰

Google Web Toolkit (GWT) est un ensemble d'outils logiciels développé par Google, permettant de créer et maintenir des applications web dynamiques mettant en oeuvre JavaScript, en utilisant le langage et les outils Java. C'est un logiciel libre distribué selon les termes de la licence Apache 2.0.

GWT met l'accent sur des solutions efficaces et réutilisables aux problèmes rencontrés habituellement par le développement AJAX : difficulté du débogage JavaScript, gestion des appels asynchrones, problèmes de compatibilité entre navigateurs, gestion de l'historique et des favoris, etc.

GWT est articulé autour d'un concept original : lors de la phase de développement, l'application est écrite en Java de façon classique, dans un environnement de développement intégré Java, et peut être déboguée avec les outils Java habituels. Une fois l'application prête à être déployée, le compilateur GWT la traduit en pur Javascript, avec support automatique et transparent pour les principaux navigateurs, dont Mozilla Firefox. Le code JavaScript généré utilise des techniques d'HTML dynamique et de manipulation du DOM (Document Object Model) pour les aspects dynamiques de l'interface.

Ce principe est rendu possible par les différents composants de GWT:

- le compilateur Java vers JavaScript, qui a pour sortie un code JavaScript optimisé, et fortement compacté de manière à accélérer le chargement et l'exécution ;
- un navigateur spécialement modifié : la *Google App Engine*, pour permettre l'exécution (et le débogage) de code Java natif sans nécessiter la compilation JavaScript ;
- une bibliothèque d'émulation JRE : il s'agit d'une implémentation en JavaScript d'un sous-ensemble de la bibliothèque de classes Java standard (en particulier quasiment tout le paquet `java.lang` et une partie de `java.util`) ;
- une bibliothèque de composants graphiques contenant des *widgets* de base permettant la construction d'une interface graphique.

GWT est souvent appelé abusivement un *framework*, mais n'en est pas véritablement un car il impose peu de choses au développeur; comme son nom l'indique, il s'agit d'une boîte à outils qui offre des solutions permettant de développer plus facilement des solutions web/AJAX de dernière génération, en profitant des outils et compétences Java existants, et en faisant abstraction de la complexité habituellement liée à ce genre de technologies.

¹⁹ Site officiel de GWT: <http://code.google.com/webtoolkit/>

²⁰ GWT @ Wikipedia: http://en.wikipedia.org/wiki/Google_Web_Toolkit

La version 2.0 de GWT est disponible depuis le mois de décembre 2009, c'est cette version que nous utiliserons pour notre projet. Cette nouvelle version simplifie la conception puisque différents plugins sont proposés pour émuler la Google App Engine sur les différents navigateurs. Il n'est donc plus nécessaire de procéder à une longue compilation à chaque modification pour observer le résultat dans le navigateur Mozilla Firefox.

De plus, Google propose également un plugin de développement pour la plate-forme de développement Eclipse²¹.

ENVIRONNEMENT DE TRAVAIL

IDE

Nous avons dû faire un choix entre différents environnements de travail existants pour Java. Après quelques essais nous avons établi une liste succincte des avantages et inconvénients de l'IDE NetBean et de celle d'Eclipse²² :

	NetBean	Eclipse
Avantages	<ul style="list-style-type: none"> - simple - léger 	<ul style="list-style-type: none"> - formés dessus - GWT bien intégré
Inconvénients	<ul style="list-style-type: none"> - pas de formation - GWT faiblement intégré 	<ul style="list-style-type: none"> - complexe - lourd

COLLABORATION

Ce projet en équipe est l'occasion de se pencher sur la recherche d'un moyen pour échanger de façon efficace les fichiers du projets. Les différentes phases du projet étant de différentes natures, nous avons profité de la première phase pour effectuer des essais. Les solutions utilisées sont : SSH et SVN. Ci-dessous nous détaillons notre approche.

²¹ Plugin GWT pour Eclipse: <http://code.google.com/webtoolkit/gettingstarted.html>

²² Java IDEs – NetBeans vs Eclipse vs JDeveloper :
<http://www.indicthreads.com/1513/java-ides-netbeans-vs-eclipse-vs-jdeveloper/>

SSH (SECURE SHELL)²³

Dans cette première phase le développement était marginal et nous avons peu de fichiers à échanger. Pour ce faire nous avons mis en place une solution temporaire qui nous permettait de déposer les fichiers sur mon serveur SSH (Édouard). Le serveur déjà existant il n'avait plus qu'à créer le compte utilisateur adaptés, attribuer les droits et les accès suffisants.

De cette manière nous avons pu échanger divers scripts et tests que nous faisons sans cesse passer par e-mail. Cependant cette approche possède quelques lacunes :

- **disponibilité**, le serveur étant hébergé sur mon ordinateur portable l'*uptime* (temps où le serveur est allumé) et la disponibilité du service ne pouvait être assurée de façon suffisante pour la deuxième phase du projet.
- **absence d'alertes**, les nouveaux fichiers pouvaient passer inaperçus si nous n'en discutons pas.
- **gestion des versions**, l'absence d'alertes n'est pas nécessaire en soi si l'on peut connaître les différences apportées au projet entre deux consultations. C'est le rôle d'un gestionnaire de version que d'identifier les modifications et d'en faire part aux développeurs.

Le problème de version va s'accroître lors de la phase 2, c'est pourquoi nous avons recherché une solution dès le début.

GOOGLE CODE

Dans l'optique de la phase 2, il nous fallait donc un gestionnaire de fichiers. Pour éviter les problèmes de disponibilité nous avons décidé d'héberger notre projet chez un service professionnel. Ayant tout deux des comptes Google et vu de leurs services (SVN, interfaces web, wiki, etc.) nous avons décidé de créer un compte Google Code. Celui-ci est disponible à l'adresse suivante :

<http://code.google.com/p/animaths/>.

Le projet est placé sous licence Apache 2.0 qui est une licence libre. Nous avons choisi cette licence car elle est permissive, puisqu'elle permet la réutilisation du code, y compris dans des applications commerciales²⁴. De ce fait, nous permettons à qui le souhaite de réutiliser notre framework.

Google Code permet également de choisir entre divers systèmes de version, nous avons choisi d'utiliser SVN car c'est un gestionnaire très répandu. Un autre facteur de choix est l'existence de plugins SVN pour Eclipse. Parmi les différents plugins SVN existants, nous avons choisi d'utiliser le plugin subclipse qui semble être le plus répandu et le plus abouti. À noter que certains utilisateurs indiquent que le plugin ralentit Eclipse de façon non-négligeable. Si besoin nous utiliserons des clients tiers : TortoiseSVN pour Windows® et la console pour Linux.²⁵

²³ SSH @ Ubuntu-fr: <http://doc.ubuntu-fr.org/ssh>

²⁴ Pourquoi choisir la licence Apache 2.0: <http://fossfaq.com/questions/15/why-would-you-choose-the-apache-license>

²⁵ Which Eclipse Subversion plugin should I use?: <http://stackoverflow.com/questions/185486/which-eclipse-subversion-plugin-should-i-use>

L'hébergement final du projet nécessite un serveur Java tel que Tomcat, nous pourrions pour la phase de déploiement faire héberger le projet par les services informatiques de l'UPMF. ²⁶

GUIDE DES BONNES PRATIQUES D'APRES GOOGLE I/O '09 27

Chaque année a lieu la conférence annuelle des développeurs Google (Google I/O). En Mai 2009, Ray Ryan, membre de l'équipe de développeurs de Google Web Toolkit, propose une présentation sur les bonnes pratiques à observer lors de la conception d'une application GWT. Le simple programme *HelloWorld* qui découle de ces recommandations est extrêmement lourd, mais promet néanmoins d'augmenter drastiquement la rapidité de développement.

Nous avons retenu l'utilisation :

- d'un bus d'évènements (EventBus), qui centralise les inscriptions des écouteurs et tous les évènements. L'EventBus est censé aider à rendre le code "propre", en proposant un couplage très faible entre écouteurs/écoutés ; observeurs/observables.
- du paradigme de conception d'interfaces Modèle-Vue-Animateur (Model-View-Presenter ou MVP) ;
- de l'utilisation de la librairie GIN (GWT INjection) : l'équivalent GWT de la librairie Guice.

Nous donnons un bref aperçu de ces deux premiers points dans les sections suivantes.

Bien que nous ne comptons pas développer une application GWT de la taille de celles de Google (Google Docs, GMail, Google Maps), nous pensons que c'est un bon exercice de s'attacher à respecter ces recommandations, d'autant plus que cela devrait permettre d'accroître la maintenabilité du code, et sa reprise potentielle par d'autres développeurs.

PARADIGME MODELE-VUE-ANIMATEUR (MVP)²⁸

Le paradigme Modèle-Vue-Animateur (Model-View- Presenter ou MVP) propose une meilleure séparation de l'architecture d'un logiciel. Il se distingue du paradigme Modèle-Vue-Contrôleur (MVC) en cela qu'il propose de supprimant définitivement le lien entre la vue et le modèle (en MVC, la vue peut être mise à jour directement sans passer par un contrôleur, comme c'est le cas pour certains objets graphiques de la bibliothèque Java swing)²⁹.

²⁶ Déploiement de GWT: <http://code.google.com/webtoolkit/doc/latest/DevGuideDeploying.html>

²⁷ Google I/O 2009 - Best Practices for Architecting GWT App: <http://code.google.com/events/io/2009/sessions/GoogleWebToolkitBestPractices.html>

²⁸ Model-view-presenter (MVP) : <http://en.wikipedia.org/wiki/Model-view-presenter>

²⁹ TotT: Be an MVP of GUI Testing: <http://googletesting.blogspot.com/2009/02/with-all-sport-drug-scandals-of-late.html>

- Le modèle est défini par une interface qui déclare les méthodes dont a besoin l'animateur. C'est la couche métier, de façon similaire au paradigme MVC.
- La vue est définie par une interface qui déclare les méthodes dont a besoin l'animateur. Elle affiche des données mais est complètement passive. C'est une coquille vide qui ne gère absolument plus rien, sauf de signaler des événements utilisateurs au bus d'événements de l'animateur.
- L'animateur agit sur la vue et sur le modèle. Il tient un rôle d'animateur, de médiateur.

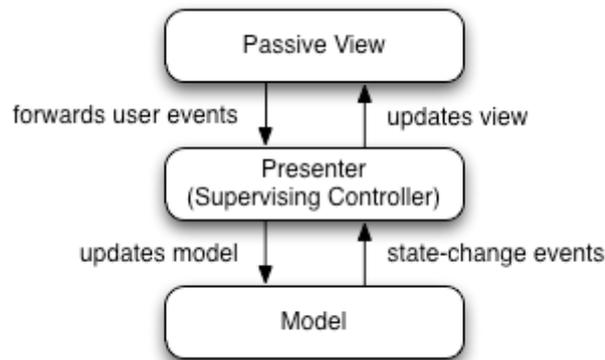


Figure: dans MVP, la vue et le modèle communiquent par l'intermédiaire de l'animateur.

Une petite anecdote : dans sa présentation, Ray Ryan affiche une tortue sur la diapositive présentant le paradigme Modèle-Vue-Contrôleur, et un lièvre sur celle qui présente le paradigme Modèle-Vue-Animateur. D'après lui, lorsque les applications grossissent, l'interaction entre la vue et le modèle peut ralentir le développement, ce qui n'est pas le cas pour le paradigme MVP.

GIN (GWT INJECTION)³⁰

GIN est un framework d'injection automatique de dépendance. Il est basé sur Guice, et utilise donc des annotations Java pour faciliter l'injection. L'initialisation du programme est alors automatiquement réalisée à partir d'une arborescence de classes qui décrit les dépendances entre elles.

Aucune version de la bibliothèque GIN n'est encore officiellement sortie, mais Google recommande son utilisation et affirme qu'elle est bien éprouvée au sein de sa communauté de développeurs. Ainsi, et puisque la bibliothèque est distribuée sous licence Apache 2.0, il nous a été possible de récupérer une archive jar pour notre projet.

³⁰ Site officiel de GIN: <http://code.google.com/p/google-gin/>

CONCEPTION ET DEBUT DU DEVELOPPEMENT

Après avoir testés l'implémentation des recommandations du W3C dans Mozilla Firefox et après avoir choisi d'utiliser GWT et de suivre les bonnes pratiques, nous avons commencé l'implémentation de l'application.

ARCHITECTURE DU PROJET

Ci-dessous l'architecture de notre projet à l'heure actuelle :

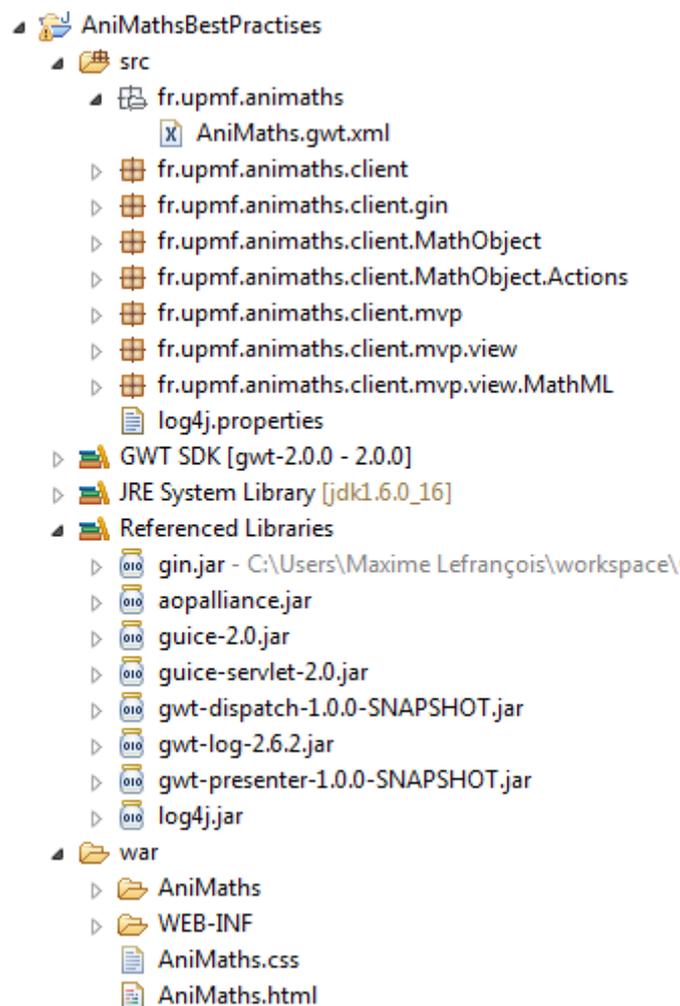


Figure: arborescence du projet dans Eclipse.

Outre les librairies GWT et JRE de base, on observe des dépendances directes librairies GIN et GWT-Presenter. Les ressources statiques sont directement posées dans le dossier war/, son sous-dossier AniMaths/ accueille les différents fichiers JavaScripts générés lors de la compilation.

Pour ce qui est des classes sources, sans rentrer dans les détails, toute notre application est concentrée dans le paquet fr.upmf.animaths.client, c'est à dire qu'on a un application quasi100% compilée en JavaScript. On envisage pas encore d'interaction avec le serveur, sauf pour l'envoi des équations à résoudre.

L'idée du tout-sur-le-client permet de limiter les échanges avec le serveur, d'augmenter la rapidité des interactions, au prix du temps de téléchargement de la librairie JavaScript lors de la connexion. (cette librairie ne dépassera vraisemblablement pas quelques centaines de Kilo-octets)

- Le fichier AniMaths.gwt.xml contient la déclaration des composants de l'application, et les librairies exportées en JavaScript ;
- Le paquet client.gin est réservé à la déclaration des modules de la librairie GIN ;
- Le paquet client.mvp contient la définition des Animateurs, et des Widgets statics de la vue (les différentes parties de l'écran).
- Le paquet client.mvp.view.MathML contient une implémentation des widgets spécifiques aux éléments MathML que l'on peut afficher dans la page
- Le paquet client.MathObject contient des classes modèles de représentation objet des objets mathématiques. Nous allons détailler cela dans la section suivante.

IMPLÉMENTATION DES ÉLÉMENTS MATHML

Dans ce rapport, nous n'exigeons pas de connaissances du MathML de la part des lecteurs, et nous n'entrerons donc pas dans les détails des différents éléments de la recommandation.

Nous pouvons néanmoins dire que nous nous sommes avus dû trouver le moyen d'accéder aux différentes fonctionnalités natives de gestion DOM qui permettent de créer des éléments MathML qui doivent déclarer un espace de nom pour être correctement affiché. L'implémentation a été faite correctement pour Mozilla Firefox, et sera différente pour Internet Explorer. Nous avons néanmoins proposé une implémentation pour ce navigateur. Nous n'avons pas pu la tester, mais – lorsqu'il sera capable d'afficher nativement du MathML – la modification du code que nous avons proposé sera réduite ; voire inexistante.

REPRESENTATION OBJET COHERENTE DES OBJETS MATHEMATIQUES

Nous avons besoin d'une méthode pour représenter une équation ou une expression mathématique de manière arborescente. En effet, le MathML ne permet pas à lui seul d'assurer la consistance mathématique des objets décrits.

Nous avons mis au point une hiérarchie d'objets mathématiques qui permettent – une fois compilés en MathML – d'assurer une cohérence mathématique quelque soit l'arborescence utilisée. Nous avons implémenté ce qui nous semblait être minimal pour le projet de génie logiciel. Cette architecture est facilement extensible et on peut imaginer à l'avenir des implémentations diverses (différents ensembles, différentes structures algébriques, Anneaux, Espaces vectoriels, ...)

Nous avons choisi le terme MathObject pour désigner cet ensemble d'objets mathématiques. Voici une liste des classes implémentées et de leurs spécificités :

- **MathObjectElement**
classe abstraite dont dérivent tous les objets MathObject. Elle définit :
 - des méthodes que les éléments MathObject doivent implémenter ;
 - certaines constantes pour qualifier le type d'objets (en rapport avec la priorité des opérateurs)
 - une méthode pour déterminer – en fonction de ces types – si un objet a besoin d'être mis entre parenthèses, ou pas.
- **MathObjectEquation**
objet qui possède deux membres MathObjects ;
- **MathObjectNumber**
nombre (objet MathObject Atomique)
- **MathObjectIdentifieur**
variable ou constante représenté par un caractère (objet MathObject atomique)
- **MathObjectSignedElement**
Élément signé : possède un fils MathObject, et peut être mis entre parenthèses. Le signe peut être implicite.
- **MathObjectAddContainer**
Conteneur d'éléments signés : possède une liste d'éléments signés, et peut être mis entre parenthèses.
- **MathObjectMultiplyElement**
Élément multiplié (ou divisé) : possède un fils MathObject et un opérateur * ou /. L'opérateur peut être implicite.
- **MathObjectMultiplyContainer**
Conteneur d'éléments multipliés : possède une liste d'éléments multipliés, et peut être mis entre parenthèses.

Nous avons éprouvé ce choix de conception en implémentant un générateur aléatoire d'expressions, dont on fera la démonstration lors de la soutenance.

FONCTIONNALITES IMPLEMENTEES

Nous avons déjà implémenté une partie des fonctionnalités de sélection :

- Un élément survolé est mis en sur-brillance ;
- On peut le sélectionner en cliquant dessus ;
- On peut alors le dé-sélectionner en re-cliquant dessus, ou en appuyant sur la touche 'Echap' ;
- On peut changer de sélection à l'aide des touches directionnelles, c'est à dire qu'on se déplace dans l'arborescence MathObject.

OUVERTURES

Nous avons beaucoup à faire pour cette seconde phase du projet, mais si l'avancement du développement le permet, nous serions heureux de pouvoir tester une première version de ce programme en situation réelle de TD, dans une classe de collégiens. Ceci nous permettra également d'avoir des retours sur ce que les utilisateurs finaux attendent de ce genre de projet.

BILAN PERSONNEL

Ce sujet nous passionne l'un comme l'autre, pour des raisons peut-être légèrement différentes. Nous avons l'opportunité de travailler sur un sujet innovant, qui nous place à la limite de ce qui est possible avec les navigateurs web actuels. C'est grisant, mais ça peut aussi s'avérer être un terrain glissant.

D'un point de vue de la formation, ce projet nous offre l'opportunité de travailler concrètement sur des technologies web et utiliser de nombreux enseignements de la formation :

- IHM, pour l'aspect interactif innovant,
- Langages et Technologie Web, pour le JavaScript, MathML et toute la plate-forme d'utilisation,
- Documents multimédia, pour XML, SVG, MathML, etc.,
- Plates-formes pour le développement Web, pour Java et l'IDE Eclipse,
- Recherche d'informations, pour toutes la première phase de recherche.

D'un point de vue plus personnel, le projet nous permet d'acquérir de nouvelles connaissances et d'approfondir celle que nous avons déjà. Cela nous permet de nous ouvrir et former à :

- d'autres plates-formes de développement Web pour Java (GWT),
- des techniques de travail collaboratif avec SVN
- aux problèmes de licence logiciel,
- au contexte des navigateurs et des recommandations du W3C
- de nouveaux paradigmes de développement : modèle MVP, méthode Agile³¹ et Extrême Programming³².

³¹ Méthode Agile @ Wikipedia: http://fr.wikipedia.org/wiki/M%C3%A9thode_agile

³² Extreme programming @ Wikipedia: http://fr.wikipedia.org/wiki/Extreme_programming

MAXIME :

D'un point de vue personnel, je suis heureux de la manière dont M. El Methni a accueilli cette idée de projet, j'apprécie l'opportunité de travailler sur un sujet qui traite de l'enseignement, en particulier celui des matières scientifiques.

Je donne personnellement des cours particuliers à quelques élèves, et j'aurais aimé qu'il existe ce genre de logiciel pour leur montrer ce qu'on peut faire et ce qu'on ne peut pas faire avec une équation.

Je place beaucoup d'espoirs dans la réalisation d'un tel logiciel : il devrait permettre aux apprenants de se lancer dans la résolution d'équations de plusieurs manières, de manière à ce qu'ils comprennent qu'il n'existe pas qu'une seule bonne solution (celle du prof), mais que parfois même en tâtonnant on arrive à trouver la solution au problème, pourvu qu'on respecte la rigueur des mathématiques.

ÉDOUARD :

Dès la première année du Master Maxime est moi avons discuté de ce genre de problématique, il semblait très intéressé par l'enseignement et souhaitait utiliser les TIC pour améliorer l'apprentissage de concepts par les étudiants. L'enthousiasme avec lequel nous avons discuté était partagé tout comme l'idée que l'apprentissage soit facilité si l'étudiant manipule et construit lui-même son savoir. Cette idée en tête, la proposition de Maxime ne pouvait que m'intéresser. Et savoir que notre encadrant serai M. El Methni, notre enseignant de la théorie des graphes en 1re année, fut un argument de plus pour rejoindre le projet.

L'aspect qui m'a le plus attiré dans la proposition de Maxime est certainement de travail sur un projet innovant et philanthrope. Innovant car, à notre connaissance, ce genre d'application n'existe pas sur internet. Les projets les plus similaires au notre³³ utilisent MathML et un peu de SVG, mais sans exploiter le potentiel des technologies disponibles. Tous les projets rencontrés se contentent de créer des contenus de la même manière qu'ils l'auraient fait sur des supports classique (livres, tableau noir). Dans un monde où l'informatique est ubiquitaire, où la manipulation directe est de plus en plus présente en IHM et où la réalité augmenté est chaque jour un peu plus présente³⁴, notre projet prend logiquement sa place.

Le projet est ambitieux, et nous permet de découvrir de nouvelles compétences : GWT, MVP, test et qualité, suivi de développement, etc. et d'en renforcer de nombreuses : technologies web, travail collaboratif, expertise web, licence logiciel, etc.

³³ Math Animated™: <http://mathanimated.com/>

³⁴ Pranav Mistry: The thrilling potential of SixthSense technology @ TED: http://www.ted.com/talks/pranav_mistry_the_thrilling_potential_of_sixthsense_technology.html

RÉFÉRENCES

1. **Service TICE-EAD de l'UPMF:**
http://tice.upmf-grenoble.fr/87264623/0/fiche___pagelibre/&RH=TICEFR&RF=TICEFR_1
2. **Adobe Flash @ Wikipedia:**
http://en.wikipedia.org/wiki/Adobe_Flash
3. **SVG, Recommandation W3C:**
<http://www.w3.org/TR/SVG11/>
4. **Mozilla SVG Statut:**
<http://www.mozilla.org/projects/svg/status.html>
5. **Support de SVG:**
<http://a.deveria.com/caniuse/#agents=All&eras=All&cats=SVG&statuses=rec,pr,cr,wd,ietf>
6. **YUI 2: Animation**
<http://developer.yahoo.com/yui/animation/>
7. **Raphaël: a JavaScript API for SVG**
<http://dev.opera.com/articles/view/raphael-a-javascript-api-for-svg/>
8. **XHTML+MathML+SVG:**
<http://www.w3.org/TR/XHTMLplusMathMLplusSVG/>
9. **MathML software list:**
<http://www.w3.org/Math/Software/>
10. **Profile MathML pour CSS :**
<http://www.w3.org/TR/mathml-for-css/>
11. **STIX project page:**
<http://www.stixfonts.org/>
12. **XUL Faces:**
<http://xulfaces.sourceforge.net/>
13. **Support MathML, aperçu rapide :**
<http://a.deveria.com/caniuse/#agents=All&eras=All&cats=DOM,Other&statuses=rec,pr,cr,wd,ietf>
14. **MathML test suite @ W3C:**
<http://www.w3.org/Math/testsuite/>
15. **Opera MathML via CSS:**
<http://dev.opera.com/articles/view/can-kestrels-do-math-mathml-support-in/>
16. **Opera CSS insuffisant:**
<http://golem.ph.utexas.edu/~distler/blog/archives/001588.html>
17. **Part de marché des navigateurs:**
http://en.wikipedia.org/wiki/Usage_share_of_web_browser
18. **Site officiel de Firemath:**
<http://www.firemath.info/>
19. **Site officiel de GWT:**
<http://code.google.com/webtoolkit/>
20. **GWT @ Wikipedia:**
http://en.wikipedia.org/wiki/Google_Web_Toolkit
21. **Plugin GWT pour Eclipse:**
<http://code.google.com/webtoolkit/gettingstarted.html>
22. **Java IDEs – NetBeans vs Eclipse vs JDeveloper :**
<http://www.indicthreads.com/1513/java-ides-netbeans-vs-eclipse-vs-jdeveloper/>
23. **Pourquoi choisir la licence Apache 2.0:**
<http://fossfaq.com/questions/15/why-would-you-choose-the-apache-license>

24. **Which Eclipse Subversion plugin should I use?:**
<http://stackoverflow.com/questions/185486/which-eclipse-subversion-plugin-should-i-use>
25. **Déploiement de GWT:**
<http://code.google.com/webtoolkit/doc/latest/DevGuideDeploying.html>
26. **Google I/O 2009 - Best Practices for Architecting GWT App:**
<http://code.google.com/events/io/2009/sessions/GoogleWebToolkitBestPractices.html>
27. **Model-view-presenter (MVP):**
<http://en.wikipedia.org/wiki/Model-view-presenter>
28. **TotT: Be an MVP of GUI Testing:**
<http://googletesting.blogspot.com/2009/02/with-all-sport-drug-scandals-of-late.html>
29. **Site officiel de GIN:**
<http://code.google.com/p/google-gin/>
30. **SSH @ Ubuntu-fr:**
<http://doc.ubuntu-fr.org/ssh>
31. **Méthode Agile @ Wikipedia:**
http://fr.wikipedia.org/wiki/M%C3%A9thode_agile
32. **Extreme programming @ Wikipedia:**
http://fr.wikipedia.org/wiki/Extreme_programming
33. **Math Animated™:**
<http://mathanimated.com/>
34. **Pranav Mistry: The thrilling potential of SixthSense technology @ TED:**
http://www.ted.com/talks/pranav_mistry_the_thrilling_potential_of_sixthsense_technology.html