Université de Genève

Application informatique

Fonctionnalités supplémentaires à un outil de calcul symbolique

Genève, le 22 mai 2019

Luka Lukic

Table des matières

1. Introduction

1.1 Projet initial

Le projet initialement proposé par le client Mr. Berthod consistait à développer une application moderne smartphone d'un outil de calcul symbolique d'unités physiques inspirée du site web et de l'application widget qu'il a déjà créé.
L'application aurait aussi des fonctionnalités supplémentaires comme le parsing des unités physiques aux exposants fractionnaires, la mémorisation de combinaisons, la possibilité de définir de nouveaux symboles… Et finalement une interface intuitive aurait conclu le travail.
À la suite d'une discussion avec le superviseur du travail, Prof. Buchs, il s'est avéré que la charge de travail était bien trop importante et qu'il était nécessaire de revoir le contenu du projet.

1.2 But du projet

Il a finalement été décidé que le projet allait consister à apporter des modifications à l'outil déjà existant sur le site web : http://mafalda.unige.ch/vuc/converter.html .
Les modifications consistent à développer un système qui permettent de mémoriser les unités déjà utilisées dans l'application, même en quittant celle-ci et de modifier la structure de la base de données des unités et du parser pour permettre le parsing des unités aux exposants fractionnaires.
L'outil étant implémenté en HTML/CSS et JavaScript, c'est donc naturellement ces langages sur lesquels il a fallu travailler.

1.3 Fonctionnement de l'outil et du parser

L'outil de calcul symbolique permet de convertir des expressions contenant une valeur numérique et des unités physiques en de nouvelles expressions avec d'autres unités choisies par l'utilisateur.
Un exemple simple : On entre $50\ kg\ m/s\text{^}2$, le parser reconnaît alors la dimension de l'entrée et nous dit que c'est une force. Et on donne l'unité de conversion voulue, qui est par exemple le dyne. Puis en cliquant sur le bouton 'convert' on obtient le résultat numérique de la conversion. (cf : *Figure 1*)

Le fonctionnement du parser repose sur une base de données très détaillée, dans laquelle on retrouve les unités physiques qui sont caractérisées notamment par un tableau contenant les exposants des unités du système international, qui, composées, valent les unités physiques.
Un exemple : l'unité de la force $[N]$ vaut dans le système international $[kg\ m\ s^{-2}]$.
Rappelons les unités du systèmes internationales : $[rad, m, kg, s, A, K, cd, mol]$. Le tableau des exposants des unités du système international pour le Newton $[N]$ est donc :

$$[0, 1, 1, -2, 0, 0, 0, 0]$$

Le but du parser est donc d'analyser l'entrée pour se ramener au tableau des exposants simplifié et donc, il peut ensuite déterminer la dimension de l'entrée.

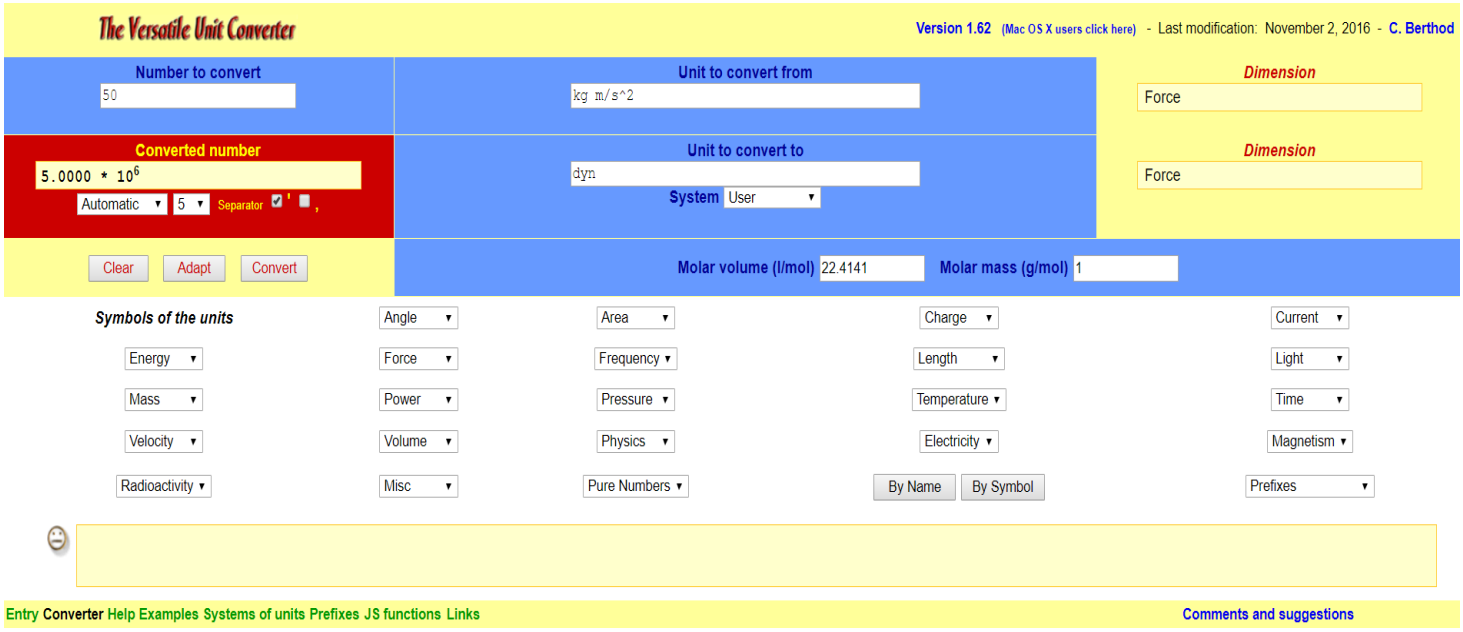

Figure 1 : Interface du site web avec les entrées de l'exemple donné en 1.3

Enfin l'utilisateur, connaissant la dimension de son entrée, entre l'unité d'arrivée de la conversion.

Les fonctionnalités supplémentaires permettraient donc de mémoriser localement pour chaque utilisateur les unités qu'il a déjà utilisées dans le menu 'Unit to convert from' de l'interface de la *Figure 1*, pour faciliter l'utilisation de l'outil, et permettraient aussi d'entrer des unités contenant des exposants fractionnaires (il est fréquent de rencontrer des formules physiques avec des racines carrées par exemple).

2. Solution proposée

La solution proposée pour mémoriser les unités déjà utilisées est l'implémentation d'un menu qui puisse les mémoriser en utilisant les Cookies.
Les Cookies sont des fichiers textes stockés dans l'ordinateur qui permettent de sauvegarder des informations qu'on a déjà données dans un site web et qui peut donc éviter de devoir les entrer lors de chaque accès à celui-ci.
Le menu permet donc de mémoriser les 10 dernières différentes unités valides que l'utilisateur a utilisé pour la première fois.
Ci-dessous, on peut voir dans la *Figure 2* le menu rempli avec 10 unités utilisées, la première à avoir été ajoutée est celle du haut : « m », et la dernière est celle du bas : « month ».
Lorsque le menu est rempli et qu'on utilise une nouvelle unité qui n'est pas déjà dans le menu, alors celle-ci va écraser la plus vieille unité qui a été ajoutée dans le menu, c'est-à-dire, dans le cas présent : « m ». C'est ce qu'on observe dans la *Figure 3* où « Hz » remplace « m ».

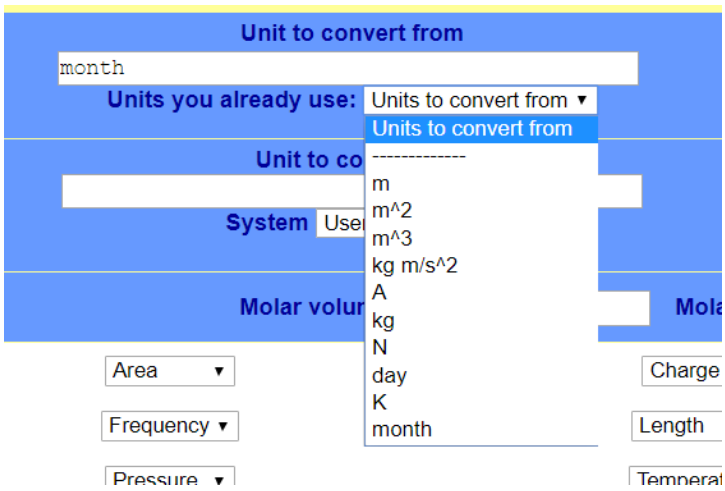

Figure 2 : Menu mémorisant 10 unités

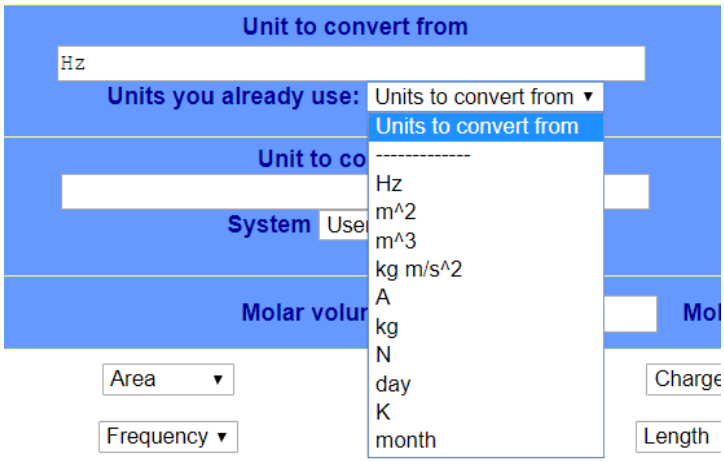

Figure 3 : Menu mémorisant une nouvelle unité à la suite du menu précédent

De plus, si l'on clique sur une unité du menu, alors celle-ci est recopiée dans le champ 'Unit to convert from', et la dimension est calculée immédiatement.
Evidemment, lorsqu'on quitte le site web, ou même le navigateur, les unités sont toujours présentes dans le menu lorsqu'on y revient.

La solution apportée à l'extension du parser pour y intégrer les exposants fractionnaires est partielle, étant donnée la tâche complexe. Elle consiste en une modification de la base de données du code en remplaçant les tableaux des exposants, par des tableaux contenant des sous tableaux pouvant représenter les numérateurs et dénominateurs des fractions et à la conception de quelques fonctions utiles à la manipulation des fractions.

3. Travail accompli

La première partie du travail a consisté à se documenter sur le développement web et sur les langages HTML et JavaScript. En effet, c'est la première fois que j'en fais. Il était donc nécessaire de passer du temps sur cela afin de comprendre de quelle manière interagissent

ces deux langages et comment programmer en les utilisant. J'ai pu, par la suite, commencer à étudier le code déjà fourni de l'application et comprendre sa structure et son fonctionnement principale, sans aller dans les détails.
Dès le début du projet, l'idée de l'utilisation des Cookies me semblait être une bonne idée, je me suis donc aussi documenté sur la structure et les spécificités de l'implémentation des Cookies dans le langage JavaScript.
Les Cookies sont stockés dans une unique variable nommée 'document.cookie', et son contenu est de la forme :

$$name1 = value1; name2 = value2; \dots; namen = valuen;$$

Où $namei$ est le nom d'un cookie et $valuei$ est sa valeur, $\forall\, i \in [1, n]$. Evidemment, cette variable est spéciale et on ne peut pas la manipuler comme une variable de type string normale. Par exemple pour supprimer un cookie de la variable il faut ajouter une commande dans la variable :

$$name1 = value1; expires = \ date\ of\ expiration;$$

En précisant à quelle date d'expiration le cookie va-t-il disparaître, (en fait ça ne suffit pas vraiment il faut préciser d'autres paramètres).

J'ai finalement décidé de faire un menu statique qui permette de conserver en mémoire 10 unités tel que décrit dans la section 2. Et j'ai donc commencé son implémentation HTML et JavaScript.
La première structure des cookies consistait à stocker les unités dans des différents 'names'. Il a été décidé plus tard de les stocker sous cette forme :

$$unitfrom = unit1 - unit2 - unit3 - \cdots - unitn;$$

Il a fallu aussi réaliser les tâches suivantes :

- Définir une fonction qui permette de modifier 'document.cookie'.
- Définir une fonction qui permette d'obtenir les cookies et de les afficher dans le menu à la bonne place.
- Définir une fonction qui permette de recopier l'unité sur laquelle l'utilisateur a cliqué (dans le menu) dans le champ adéquat et calculer et afficher la dimension de celle-ci immédiatement.

Bien sûr il fallait prendre en compte le fait qu'on ne veut pas ajouter deux fois la même unité dans le menu et créer un système récursif qui permette d'écraser les plus vieilles unités déjà présentes et qui permette aussi de rajouter une unité qui vient d'y disparaître. Il fallait aussi prendre en compte le fait qu'on veuille que le menu soit actualisé dès que l'on clique sur le bouton 'convert' ou 'adapt' et pas seulement en revenant sur le site web. Une autre tâche était de vérifier que les unités ajoutées au menu soient valides en les faisant passer à travers le parser (sur lequel il a fallu faire des modifications).

Un nombre incalculable de tests manuels a été réalisé pour vérifier le fonctionnement récursif du menu et le fait qu'il n'accepte pas deux fois les mêmes unités.
Les tests ont été fait sur un serveur web local car certains navigateurs ne permettent pas d'offrir l'utilisation des Cookies à partir d'un fichier HTML (Chrome ne le permet par exemple pas, alors que Firefox si).
J'ai aussi été en contact avec mon client tout au long de ce projet, et il m'a fait part des améliorations voulues, et j'ai pu lui demander si le fonctionnement de mon menu fonctionnait chez lui.

Il s'est avéré que des problèmes apparaissaient sur certains navigateurs et que des Cookies étrangers s'immisçaient dans mon programme, alors même qu'il a fait les tests sur l'url suivante : http://mafalda.unige.ch/vuc-new/converter.html
Les Cookies étrangers venaient notamment du domaine 'unige.ch', auquel je n'ai pas eu affaire lors de mes tests initiaux. C'est ce qu'on peut voir dans la *Figure 4* où on observe des cookies étrangers du nom de '_ga', '_gat', '_gid' apparu après avoir accédé au site web : https://moodle.unige.ch/ et actualisé la page web de l'outil.
Après avoir eu les accès au serveur web de mon client j'ai pu revoir la structure de mes cookies et trouver un moyen d'éliminer les cookies indésirables qui peuvent s'immiscer dans la variable 'document.cookie' à n'importe quel endroit. En parallèle des tests ont été faits sur différents navigateurs et différentes OS pour vérifier si tout fonctionne correctement.

Finalement, la base de données des unités du code a été modifiée permettant de préparer la modification du parser pouvant reconnaître des expressions aux exposants fractionnaires. Le code a été modifié en conséquence pour assurer son bon fonctionnement et certaines fonctions utiles ont été ajoutées comme par exemple une fonction qui permette de déterminer une fraction irréductible à partir d'une fraction donnée (en utilisant le pgcd). Le parser en tant que tel n'a pas pu être modifié pour les raisons qui seront émises dans la section suivante.

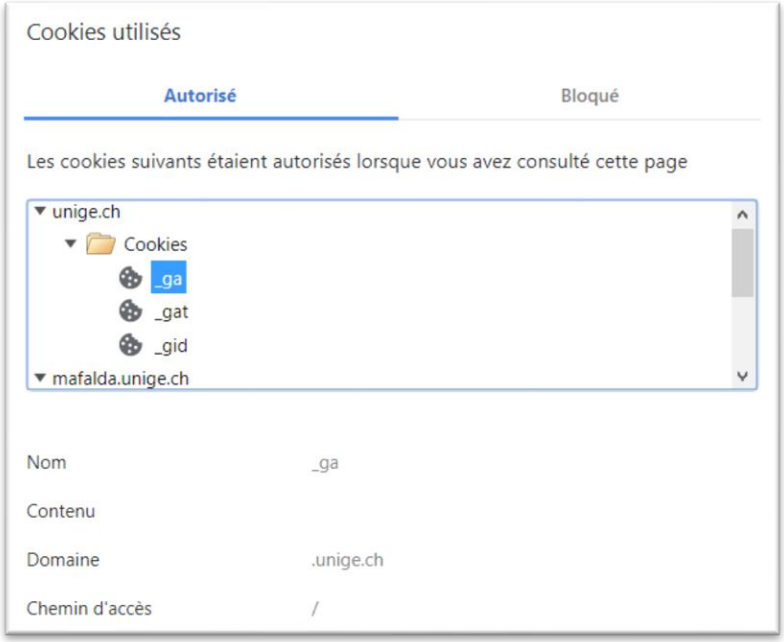

Figure 4 : Cookies étrangers apparus après avoir accéder à moodle.unige.ch

4. Discussion

4.1 Difficultés rencontrées

Il y a eu durant ce travail plusieurs difficultés auxquelles j'ai été confronté. Premièrement, comme cela a été dit précédemment, je n'avais jamais fait de développement web et donc

jamais travaillé avec HTML et JavaScript avant ce projet. Il a donc fallu un certain temps nécessaire pour se documenter à ce sujet.
La difficulté majeure du projet a été d'essayer de comprendre le code déjà fait. Il m'a fallu aussi un certain temps pour comprendre son fonctionnement principal. Il est, à mon sens, toujours plus difficile de reprendre le travail de quelqu'un plutôt que de commencer un projet soi-même. Le parser qui a été fait par mon client est extrêmement complexe, et peu commenté, et sachant qu'il est largement relié au reste du code, avec la gestion des évènements (des messages d'erreurs par exemple), il m'a semblé impossible d'arriver à le modifier en garantissant de le garder fonctionnel tout simplement car je n'ai pas pu comprendre son fonctionnement intégral.

Une autre des difficultés majeures que j'ai eu durant le projet a été le problème des Cookies étrangers évoqué précédemment. Arrivé tardivement, il a nécessité de revoir mon implémentation des Cookies. Il m'a fallu un bon nombre de tests et de méthodes différentes pour arriver à résoudre le problème.

Enfin une autre difficulté a été les tests manuels, et sur différents navigateurs, de manière générale. J'ai passé beaucoup de temps à faire des tests manuels pour vérifier si la récursivité, et les propriétés décrites à la section 3, de mon menu fonctionnent correctement. Le Professeur Buchs, nous avais parlé d'outils permettant d'automatiser ces tests. Il aurait été judicieux de les essayer.

4.2 Améliorations proposées

Une amélioration graphique sympathique aurait été un menu déroulant dynamique (qui grandit au rythme des unités qui s'y ajoutent) au lieu d'un menu statique.

Sinon, un problème de l'implémentation présente peut amener une piste d'amélioration. Le menu ne garde pas à tous les coups en mémoire les 10 dernières unités utilisés. Par exemple, si depuis la *Figure 3*, là où l'utilisateur vient d'utiliser le « Hz », il utilise ensuite le « m^3 ». Le « m^3 » étant déjà dans le menu, rien ne va se passer. Il utilise ensuite le « V » et le « Joule », alors le « V » va écraser le « m^2 » et le « Joule » va écraser le « m^3 ». Or le « m^3 » a été utilisé 3 unités auparavant.
On pourrait par exemple créer un nouveau Cookie qui permette de sauvegarder l'ordre de la dernière utilisation des unités présentes dans le menu et celui-ci vérifierait que le « m^3 » est à la troisième place et non la dernière, il ne faudrait donc pas l'écraser, mais écraser l'unité qui a été utilisée le plus anciennement, à la dixième place. Il faudrait aussi dans ce cas revoir l'implémentation de la fonction qui écrit les unités sur le menu.

5. Conclusion

Ce travail a été, à mon sens, très enrichissant. Il reflète très certainement un bon nombre de situations et de problèmes que l'on peut avoir dans le milieu professionnel en tant qu'informaticien. Il m'a aussi permis d'apprendre des choses dans des domaines que je ne connaissais pas auparavant, et je suis content d'avoir pu apporter une solution fonctionnelle à un problème donné. On entend souvent parler des Cookies, encore plus aujourd'hui car la

sécurité de l’information et la confidentialité font partie des problèmes majeurs du Web. Il me permet de mieux me représenter ce qu’ils sont réellement et comment ils fonctionnent.