

## Cahier des charges

### **Implémentation et évaluation d'une interface pour l'enseignement de la preuve mathématique**

# Sommaire

<b>Sommaire.....</b>	<b>2</b>
<b>1.Introduction.....</b>	<b>3</b>
1.1.Concepts de base.....	3
1.2.Contexte.....	4
1.3.Historique.....	4
<b>2. Description de la demande.....</b>	<b>5</b>
2.1.Produit du projet.....	5
2.2.Les objectifs.....	5
2.3.Les fonctions du produit.....	5
<b>3. Contraintes.....</b>	<b>6</b>
3.1.Contraintes de délais.....	6
3.2.Contraintes matérielles.....	6
3.3.Autres contraintes.....	6
<b>4. Déroulement du projet.....</b>	<b>6</b>
4.1.Planification.....	6
4.2.Ressources.....	7
4.3.Organisation.....	7
<b>5. Annexes.....</b>	<b>7</b>
<b>6.Références.....</b>	<b>8</b>

# 1.Introduction

Lors de notre master, nous suivons l'UE Travaux Encadrés de Recherche (TER) qui nous permet d'avoir une expérience dans la recherche. Ainsi, il nous a été proposé quelques sujets et nous avons décidé de choisir "Implémentation et évaluation d'une interface pour l'enseignement de la preuve mathématique". Nous avons fait ce choix car, en dépit du fait d'être en master à fortes composantes en informatique, nous avons toujours de la curiosité envers les mathématiques. Ce projet combinant les mathématiques et l'informatique nous satisfait ainsi grandement.

Ce document présentera une vue d'ensemble de notre projet, son contexte et son historique, la description de la demande, les contraintes liées, et pour terminer nous vous présenterons le déroulement du projet qui a été réalisé lors de ce semestre et ce qui sera encore à effectuer avec les ressources nécessaires.

## 1.1.Concepts de base

**Assistant preuve mathématique:** Logiciel ou outil informatique conçu pour aider les mathématiciens à formuler, vérifier et souvent automatiser des preuves mathématiques. Ces assistants de preuve peuvent être utilisés dans le cadre de la vérification formelle, où l'objectif est d'assurer la validité des résultats mathématiques en utilisant des méthodes rigoureuses.

**API (Interface de Programmation Applicative):** Ensemble de règles et de protocoles qui permettent à différents logiciels de communiquer entre eux. Elle définit les méthodes et les données qui peuvent être utilisées pour faciliter l'intégration de différentes applications ou systèmes informatiques.

**Coq:** Assistant de preuve interactif qui utilise la théorie des types dépendants. Il permet de spécifier des théorèmes, construire des preuves étape par étape, et de vérifier formellement que ces preuves sont correctes.

**JScoq:** Interface web pour l'assistant de preuve Coq. Il permet aux utilisateurs d'utiliser Coq directement dans un navigateur web sans nécessiter d'installation locale.

**Preuve mathématique/Démonstration mathématique:** Argument logique et rigoureux qui établit la vérité d'une proposition mathématique. L'objectif d'une preuve est de convaincre de manière indiscutable que la proposition en question est vraie. Les mathématiques reposent sur un ensemble de règles logiques strictes, et une preuve doit suivre ces règles pour être considérée comme valide.

**Tableau de variation :** Graphique permettant de définir la variation d'une fonction sur des intervalles.

**Tactique :** Étapes pour résoudre un problème mathématique. Elles décomposent le problème en petites parties afin de le rendre plus gérable et à construire progressivement une preuve cohérente. Issu de la méthode "Diviser pour mieux régner".

## 1.2.Contexte

Pour mettre en contexte, les assistants à la preuve sont des logiciels chargés de vérifier la cohérence logique des démonstrations mathématiques. Ils sont capables de formaliser la plupart des concepts mathématiques, énoncer des théorèmes, et les démontrer. Ces outils sont généralement conçus pour des utilisateurs experts et nécessitent une formation au préalable et appropriée pour être utilisés efficacement.

L'enjeu global est d'intégrer ces assistants de preuve dans le domaine de l'enseignement des mathématiques de sorte à ce que les étudiants puissent passer le moins de temps possible sur l'apprentissage du logiciel et d'expérimenter comment celui-ci peut être d'un réel atout dans leur pratique.

La stratégie est d'intégrer des éléments graphiques, spécifiquement les tableaux de variation, largement utilisés dans l'enseignement mathématique, dans l'assistant de preuve Coq.

En résumé, on veut combler le fossé entre les approches traditionnelles de la preuve et les exigences formelles des assistants à la preuve, tout en enrichissant l'apprentissage par des représentations graphiques.

## 1.3.Historique

Les assistants de preuve existent depuis plus de 30 ans. Cependant, étant des logiciels peu connus jusqu'alors, ils ont été spécialement utilisés par les « matheux » pour vérifier des preuves mathématiques. De ce fait, de grands mathématiciens auraient voulu pouvoir utiliser ces assistants mais n'avaient pas la connaissance de leur existence. Plus tard, un engouement autour de ces assistants de preuve naît et le domaine d'utilisation s'élargit ( informatique, mathématique, didactique,...).

Une question s'est alors posée : Et si on mettait des assistants de preuve dans les cours de mathématiques ? C'est dans le cadre de ce questionnement que s'inscrit notre projet issu de LiberAbaci ([Enseigner les mathématiques à l'aide de Coq](#)). Les objectifs principaux sont en résumé : l'amélioration de la logique, la création de bibliothèques, et l'intérêt pour la partie graphique. Notre TER s'inscrit dans l'intérêt pour la partie graphique incluant trois tâches parmi les sept du projet LiberAbaci :

- **Tâche 1 : collaborations avec des enseignants**  
Tâche 2 : Fondement de la théorie des types
- **Tâche 3 : structures, inférence et hiérarchie**  
Tâche 4 : notations extensibles et langage de surface  
Tâche 5 : traitements automatiques
- **Tâche 6 : environnement interactifs**  
Tâche 7 : création de bibliothèque sur des domaines précis

Le but du projet est d'ajouter un registre sémantique au logiciel via les tableaux de variations. Les tableaux de variation étant une notion dans le programme de mathématiques

au lycée mais rappelée dans l'enseignement mathématique à l'université, vont permettre la facilité aux étudiants de refaire ce qu'ils connaissent déjà en utilisant l'assistant coq.

## 2. Description de la demande

### 2.1. Produit du projet

Nous devons créer une interface de programmation applicative (API) pour Coq en se basant sur JScoq. Cette API devrait permettre la possibilité de créer des tableaux de variations de manière interactive. C'est-à-dire que l'utilisateur pourra créer un tableau de variation d'un côté et Coq devrait être en mesure de vérifier si ce tableau avec ses contenus sont cohérents et corrects.

### 2.2. Les objectifs

- \_ Être capable de résoudre un exercice de mathématique via le logiciel (notamment un exercice sur les tableaux de variations).
- \_ Permettre une interaction conviviale entre l'utilisateur et le logiciel.
- \_ Pouvoir gérer les erreurs des étudiants (avec des retours graphiques).
- \_ Pouvoir manipuler un tableau de variation.
- \_ Maintenir le lien avec la preuve textuelle.

### 2.3. Les fonctions du produit

1. Mettre en place des tactiques textuellement pour réaliser un tableau de variation:

Les tactiques envisagées sont : "on veut démontrer que", "on pose", "on a", "on sait que", "on veut que". Les tactiques de coq sont codées en OCAML, dans ce langage nous pouvons donner à notre tactique le nom que l'on souhaite. Ces tactiques sont temporaires mais peuvent être définitives. Elles ont été choisies car elles répondent au mieux au problème de résolution d'un tableau de variation.

2. Dresser un tableau de variation graphiquement

Principalement, à l'aide de boutons et de zones d'édition, l'utilisateur devrait pouvoir effectuer les opérations suivantes pour dresser et compléter son tableau de variation.

- \_ ajout de lignes et colonnes (calculs intermédiaires, points d'intérêt)
- \_ ajout d'information dans les cases (valeurs, signes, variations)
- \_ validation des informations entrées, indication des éléments à prouver

3. Permettre à l'utilisateur d'être libre dans sa démarche

Pour créer son tableau de variation, l'utilisateur devrait pouvoir utiliser la partie textuelle et/ou la partie graphique du support selon ce qui lui semble adapté dans sa façon de faire.

Ceci a pour vocation de ne pas forcer l'utilisateur (un étudiant par exemple) d'adopter un raisonnement en particulier mais d'être dans une démarche d'apprentissage.

#### 4. Retour graphique sur les erreurs de l'utilisateur

Lorsque l'utilisateur fait des erreurs, le support devrait retourner une indication sur l'endroit où l'erreur se trouve. Toujours dans un cadre pédagogique, cette indication devrait aider l'utilisateur vers l'apprentissage et non le bloquer de sorte à ce que celui-ci puisse poursuivre son raisonnement même s'il y a des erreurs, les erreurs pouvant être un manque d'information par exemple.

#### 5. Implémenter notre interface sur JScoq

Implémenter notre interface sur JScoq afin de la standardiser, car ce dernier se trouve sur le web et donc tout le monde a accès au même service et dans les mêmes conditions contrairement au logiciel coq, et ainsi plus simple d'utilisation pour nos clients. Un travail en amont a été réalisé pour le choix de JScoq. Sur ce choix nous pourrions ainsi garder la partie qui permet la communication de JScoq avec coq et de changer l'interface graphique.

## 3. Contraintes

### 3.1. Contraintes de délais

Nous devons rendre le produit final à la fin de notre TER. Entre temps nous devons savoir si notre support satisfait les clients donc il devra être testé ou présenté dans une version alpha deux à trois semaines avant les partiels du second semestre qui seront en mi-mai.

### 3.2. Contraintes matérielles

Pour le bon fonctionnement du produit les utilisateurs devront avoir besoin d'un ordinateur en état de marche et du support que nous allons leur fournir.

### 3.3. Autres contraintes

***\*\*\*Spécifier les éventuelles contraintes à prendre en compte dans le cadre du projet (normes techniques, clauses juridiques, etc.)\*\*\****

## 4. Déroulement du projet

### 4.1. Planification

Dans un premier temps, nous avons dressé un diagramme d'échéance (Figure 1) que nous avons essayé de suivre dans cette première phase du projet. Cette première planification se divise en quatre parties :

1. Identifier l'objectif du projet.
2. Comprendre le concept et l'utilisation de coq.
3. Décortiquer coq.
4. Faire un état de l'art sur les technologies et les méthodes pour en prendre partie.

Une fois que ces quatre premières phases seront effectuées, nous devons pouvoir établir les étapes du développement.

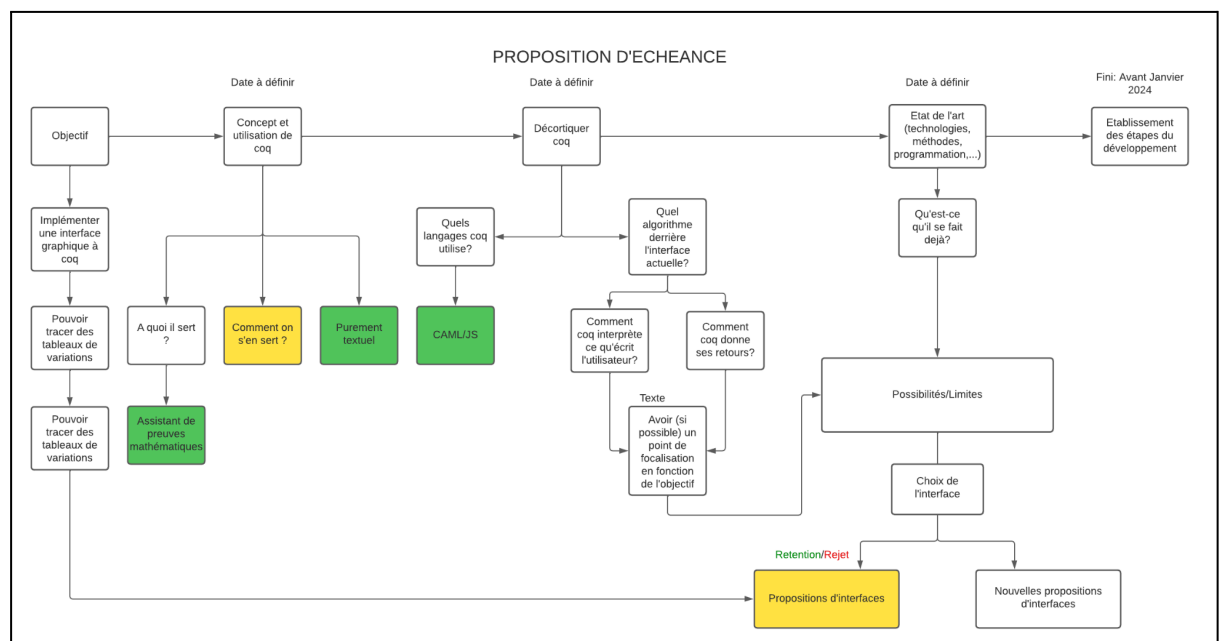


Figure 1 : Premier diagramme d'échéance et de tâches

## 4.2.Ressources

\*\*\*Lister les ressources humaines et matérielles que le client peut mettre à la disposition du prestataire\*\*\*

## 4.3.Organisation

## 5. Annexes

\*\*\*Lister et joindre au cahier des charges les éventuels documents que le client peut mettre à disposition\*\*\*

## 6. Références

\*\*\*Indique les références bibliographiques vers d'autres documents apportant des informations complémentaires\*\*\*