

Projet OptiTool

CR Réunion 1 – 10/09/18

Contenu de la réunion :

- Présentation du projet et de l'organisation au cours du temps
- Présentation d'ADTool et des arbres d'attaques et de défense
- Définition des tâches à réaliser pour la semaine prochaine

A faire pour la semaine prochaine :

- Lire le polycopié
- Définir les rôles de chacun et décider comment les rôles tournent (oui ou non, si oui, à quelle fréquence)
- Lire l'article « attack trees » de Schneier
(https://www.schneier.com/academic/archives/1999/12/attack_trees.html)
- Lire le rapport de Sara résumant les arbres d'attaque et de défense
- Lire l'article de Kordy et Widel sur l'utilisation d'optimisation linéaire et des ADTrees pour résoudre le problème de sélection optimale des contre-mesures
(<http://people.irisa.fr/Barbara.Kordy/papers/iFM17.pdf>)
- Jouer avec ADTool (l'utiliser) (<http://satoss.uni.lu/members/piotr/adtool/>)
- Lire le manuel d'ADTool (en particulier la section concernant les formats XML)
(<http://satoss.uni.lu/members/piotr/adtool/manual.pdf>)
- Créer un git
- Choisir un outil pour manager le projet (tâches, temps ...)
- Créer un alias ou une adresse email/mailling list
- Choisir si on veut faire les réunions en anglais ou en français

Rôles

Il faut définir les différents rôles (chef de projet, responsable communication, rapporteur pour les réunions, expert Java, expert optimisation linéaire, rédaction des rapports) et les attribuer à chacun. Il faut également décider comment les rôles tournent (après chaque deadline ?). Attention, prendre en compte pour les rôles les personnes travaillant dans les assos et les périodes de rush (FOG, FGO).

Il faut se diviser les tâches et utiliser ensuite l'expertise de chacun.

Rapports (1^{er} semestre)

3 rapports :

- Rapport de pré-étude (existant + positionner le projet dans l'existant)
- Rapport de spécifications fonctionnelles (fonctions logicielles de l'outil qu'on va créer)
- Rapport de conception (comment va-t-on implémenter les fonctions du projet – toutes les classes)

Attention, le rapporteur reste le même tout le long du projet donc pour le 2^e et 3^e rapport, il ne faut pas répéter tout ce qui a été dit dans le 1^{er} rapport, etc.... Par ailleurs, le rapporteur n'est pas un expert en sécurité : adapter le discours !

+ rapport de planification (planification du projet pour le 2^e semestre): voir avec le prof de gestion de projet ce qu'il souhaite exactement. Utiliser MS Project pour le rapport et la présentation mais pas forcément dans la gestion de projet. **Penser à suivre les heures passées sur le projet (tâches réalisées par chaque personne, temps consacré/pers, nombre d'heures sur le projet/equipe ..)**

Les rapports sont à rédiger en Latex (à envoyer en PDF à Barbara) ! (Barbara : exemple de rapport ?)
Relecture des rapports : 48h pour que Barbara corrige !

Documents à envoyer à Barbara

- Compte-rendu de réunion par mail (avant la fin de semaine – mercredi si possible) (ce qu'on a fait + qu'est ce qu'on doit faire pour la prochaine fois)
- Au plus tard le dimanche soir, mail pour dire les sujets à discuter pour la prochaine réunion.
- Mail à envoyer à l'adresse IRISA de Barbara !

Notes

Les notes sont les mêmes pour tout le groupe, sauf cas exceptionnel.

Organisation

- Réunion toutes les semaines avec Barbara (environ 1h)
- Réunion du groupe (en physique – jeudi aprem ? ou sur facebook ? – à discuter)

Soutenances projet

Répétition avec l'encadrant interdit → répétition avec un autre prof.

Deadlines

Pour le 24 : faire le plan du rapport de pré-étude (description de l'existant + positionnement du projet dans l'existant) : décrire ADTool, prototype, méthodologie, solveur (mpsolve), arbres d'attaques

ADTool

Pourquoi avoir développer ADTool ? normalisation des arbres d'attaques (car avant tous les arbres différents, personne n'utilisait les mêmes logiciels de dessins ...) / arbre plus lisible et plus joli / permet à tout le monde de dessiner des arbres d'attaques et de défense sans aucune compétence particulière dans un logiciel.

Durant le projet : utiliser la « current version » (Beta Phase). Prendre le code tel qu'il est et le mettre sur le git.

Pont entre ADTool – prototype : XML (export) (regarder en détail la syntaxe dans le manuel)
ADTool → parse le fichier en XML → transformation en problème d'optimisation → envoi à mpsolve qui résout le problème → retour en ligne de commande (pour l'instant)

Projet

Dans Edit, rajouter un choix « optimize » (3 problème d'optimisation : temps, cout, proba).
Si possible, livrer mpsolve dans le jar d'ADTool (via jenkins ?)