

Conception et développement d'un outil d'emploi du temps via des services web

30 janvier 2013

1 Encadrement

- Encadrants : Arnaud Malapert et Mohamed Rezgui.
- Laboratoire : I3S Équipe : CeP “Contraintes et Preuves”.
- Emails: `firstname.lastname@unice.fr`

2 Contexte

Nous recherchons des étudiants pour l'élaboration d'un logiciel d'emploi du temps basé sur des services web et offrant une expérience utilisateur fluide et agréable. Votre mission, si toutefois vous l'acceptez, est de réaliser plusieurs interfaces utilisateurs permettant de saisir les données nécessaires à la création d'un emploi du temps, puis de visualiser cet emploi du temps. Les problèmes d'emploi du temps font partie de la famille des problèmes de planification d'horaires. Schématiquement, on veut allouer des quarts de travail aux employés respectant les contraintes légales ou autres de manière à satisfaire au mieux un certain nombre d'objectifs ou de préférences. Le contrat de travail de chaque employé spécifie les contraintes légales relatives à ces horaires de travail. Il existe souvent des contraintes supplémentaires relatives au secteur d'activité ou au type d'organisation. Ensuite chaque employé spécifie des contraintes de disponibilité (congrés, maladie ...) et émet des préférences quant à ces horaires de travail (période de repos, durée d'activité ...). Les contraintes et objectifs de l'employeur consistent principalement à répondre à la demande en fournissant une bonne qualité de service tout en veillant à contrôler les coûts (heures supplémentaires, travail de nuit, ...) et conserver des conditions de travail correctes pour ces employés. En effet, les

conditions de travail influent directement sur la productivité, la fidélité et la satisfaction des employés.

On retrouve ces problématiques dans toutes les organisations que ce soit dans le secteur public (administrations, hôpitaux, université ...) ou privé (compagnies aériennes, grande distribution, usines ...). Certaines entreprises commercialisent des solutions pour de grandes organisations où une organisation rationnelle du travail est une clé du succès. Ces entreprises ciblent donc un marché restreint, mais où quelques riches clients suffisent à la survie de l'entreprise. Dans notre cas, nous ciblerons un marché étendu mais diffus, la petite et moyenne distribution.

3 Cas d'utilisation

Prenons l'exemple d'un commerce avec une dizaine d'employés. Chaque employé indique ses disponibilités (congés, maladies, formations) et ses préférences (8h-17h et un repos le mercredi) en remplissant un agenda accessible par internet. L'employeur indique les contraintes de production et celles relatives à chaque employé ainsi que les objectifs à remplir. Toutes ces informations sont centralisées afin qu'un algorithme de recherche opérationnelle ou d'intelligence artificielle traite le problème. Notez qu'on ne sait pas trouvé une solution en temps polynomial aux problèmes d'emploi du temps qui sont NP-Complets. De plus, vous avez peut-être déjà remarquer dans la vie courante que ces problèmes sont insolubles ! Il faut souvent relâcher des contraintes puisque nous ne possédons pas le don d'ubiquité ! Si un emploi du temps est établi, on souhaite générer différents rapports, par exemple les emplois du temps individuels des employés, un planning général et des graphiques et tableaux et graphiques récapitulant les charges de travail.

4 Objectif

L'objectif de ce PFE est de proposer une application pour planifier emploi du temps d'une petite organisation. Concrètement, il faut centraliser les informations fournis par les utilisateurs (employés) et l'administrateur (employeur). L'application doit offrir des interfaces intuitives et efficaces pour la saisie de ses informations. Ces informations permettent de définir une instance d'un problème d'emploi du temps traité par l'application. Il faudra aussi déterminer et mettre en œuvre un algorithme de recherche opérationnelle ou d'intelligence artificielle pour traiter le problème. Finalement, l'application offrira des fonctionnalités avancées de reporting (plannings, rap-

ports, tableaux, graphiques) qui sont essentielles pour la compréhension et la mise en œuvre de la solution proposée.

Toutes les interfaces doivent être basées sur des services web afin d’offrir une expérience utilisateur agréable. Côté client, l’application doit utiliser des formats de données répandus et ouverts afin de garantir une interopérabilité avec d’autres systèmes et la pérennité des données, par exemple le format [iCal](#) pour les agendas. De plus, il serait souhaitable d’utiliser les API existantes comme [Google Agenda API](#). De la même manière, la chaîne de services pourrait être pilotée en utilisant le langage JavaScript avec la technologie AJAX puisque l’application est dynamique. Il est recommandé de prendre une librairie JavaScript tels que [jQuery](#) ou [Mootools](#) pour faciliter la maintenance de l’application et améliorer la compatibilité entre les navigateurs internet. À titre d’exemple, les graphiques des rapports pourraient être générés avec la bibliothèque [Highcharts](#). Côté serveur, la technologie JEE (Java Enterprise Edition) pourrait être un bon choix pour centraliser toutes les informations fournis par le côté client. En cas de succès, les étudiants pourront prospecter des entreprises évaluer l’application avec le soutien des encadrants. Par exemple, un bon argument de vente consisterait à montrer que l’application peut générer des solutions meilleures que celles qui ont été effectivement mise en œuvre dans le passé.

5 Travail à réaliser

Dans un premier temps, les étudiants devront définir un cahier des charges sur la conception et le design des interfaces de saisie. Les interfaces à destination des utilisateurs doivent être simples et intuitives. Les interfaces à destination de l’administrateur seront nécessairement plus complexes car celui-ci doit saisir de nombreuses informations provenant de sources diverses (contrats de travail, demande des clients, ...). L’accent sera alors mis sur la rapidité de saisie et l’intégrité des données.

Parallèlement, les étudiants devront définir de manière rigoureuse le problème d’emploi du temps traité par l’application. Ils passeront aussi en revue la littérature sur ce type de problème afin de choisir une méthode appropriée. On mettra l’accent sur des méthodes classiques mais efficaces pour résoudre ces problèmes qui peuvent être mises en œuvre rapidement. Idéalement, on souhaiterait modéliser le problème et le transmettre ensuite à un solveur (une application tierce) qui se charge de le résoudre.

Finalement, les étudiants devront compléter le cahier des charges pour la génération des rapports. Différentes technologies pourront être mises en

œuvre en fonction du public du rapport. Par exemple, un employé consulte son emploi du temps sur son ordinateur/tablette/smartphone alors qu'un autre l'affiche dans sa cuisine. Par contre, la comptabilité préférera sûrement recevoir des feuilles excel et des pdf.

Après cette phase de conception, les étudiants définiront un planning pour la réalisation de l'outil. Les étudiants devront suivre un cycle de développement bien établi afin de produire du code source et un logiciel de qualité.

Ce travail pourra être éventuellement prolongé en stage rémunéré.

6 Environnement

Le travail proposé nécessite des goûts (et des aptitudes) en matière de conception d'interface ainsi que pour l'algorithmique et la programmation. Les étudiants choisiront les langages et technologies utilisés au cours du projet (les encadrants gardent un droit de veto).

7 Références

- [aSc TimeTables](#)
- [Timetabler plus](#)
- [Timetabler 6](#) par Century Tech
- [Visual Timetabling](#)
- [Hyper Planning](#)
- [Mana Time](#)
- [Gestion et Réservation de Ressources](#)
- [PHPMyWeek](#)