

Introduction

Une grande firme d'ingénierie fait appel à votre équipe pour développer une application (RecyclApp) qui permet de faire le design d'un centre de tri.

Un centre de tri est une usine qui reçoit en entrée le contenu (pêle-mêle) des bacs à recyclage. Différentes machines et/ou stations manuelles, permettent de séparer le contenu en différentes catégories/produits. Dépendamment des équipements (stations) choisis et de la façon dont elles seront inter-connectées, on obtiendra en sorties des ballots de matière plus ou moins purs.

La conception d'un centre de tri en fonction des exigences du client (i.e. du taux de pureté des extrants) est une tâche extrêmement complexe : une multitude d'équipement doivent être sélectionnés de manière à ce que le design final permette la gestion du flux de matière sans voir apparaître des goulots d'étranglement. Actuellement, la conception de ces centres est réalisée de manière traditionnelle (par essai et erreur) par des ingénieurs d'expérience. Les plans sont réalisés sous Visio et les calculs avec Excel. Ce travail est lourd et fastidieux.

L'application que vous développez permettra aux concepteurs de faire le design du centre de tri de manière interactive. Le concepteur (utilisateur) pourra définir le panier de produit consommé par l'usine (kg/h de chaque produit entrant), placer et déplacer les équipements sur le plan (à l'aide de la souris), spécifier le comportement de chaque équipement/machine/station, connecter les sorties et entrées des machines entre elles à l'aide de convoyeurs. Il pourra aussi voir les quantités en kg/h de chaque produit passant en tout point du réseau.

Description détaillée

L'application utilisera la souris pour placer les stations (équipements de tri) dans le plan et les connecter entre eux à l'aide d'arcs (représentation d'un convoyeur). Il est également possible de placer des points d'entrée de matière, des points de sortie de matière (sorties de l'usine) et des points de jonction (représentant la fusion de plusieurs convoyeurs).

Des fenêtres permettront de définir les paramètres des différents éléments. Par exemple, pour une entrée, on peut définir les quantités (Kg/h) de chaque produit qui entre par ce point.

Pour une station on peut notamment définir un nom, une description, le nombre de sorties de la station. On peut ensuite définir une matrice de récupération spécifiant, pour chaque produit en entrée, le « pourcentage » de ce produit qui se retrouve à chacune des sorties. Il y a une sortie par défaut pour chaque station. Les produits pour lesquels on n'a pas spécifié de pourcentage se retrouvent par défaut à 100% dans cette sortie.

Chaque arc (ou convoyeur), relie une sortie d'un nœud (station, entrée, sortie, jonction) à l'entrée d'un nœud. Une jonction est le seul type de nœuds qui peut avoir plusieurs arcs entrants.

Un arc est visuellement représenté par une ligne avec une flèche pleine à son extrémité. Chaque arc peut être

d'une couleur différente, choisie par l'utilisateur.

Une station est visuellement représentée par une boîte rectangulaire (d'une couleur au choix de l'utilisateur). On y affiche certaines informations, par exemple son nom. On peut également décider d'afficher une image représentant la station. L'utilisateur décide si l'ensemble du schéma doit afficher l'information textuelle ou les images. Dans tous les cas le cadre du rectangle est toujours présent. Chaque sortie de la station est identifiée par un petit no.

En cliquant sur (ou en passant la souris au-dessus?) un élément (nœud ou arc), il est possible de voir toute l'information se rapportant à cet élément (quantité des produits en transit, etc.)

Pour chaque arc on peut notamment connaître les quantités des différents produits qui transitent par celui-ci. On peut également obtenir cette information pour un nœud donné (il s'agit de la somme des arcs entrant).

Toute l'édition doit se faire en mètres (et non pas en pixels). Il est possible de zoomer/dézoomer lors de la visualisation. Une grille peut être activée par l'utilisateur pour faciliter l'édition (l'utilisateur spécifie la distance en mètres entre chaque point). On peut rendre la grille « magnétique » (dans ce cas on peut placer un élément seulement sur un point de la grille, pas entre deux points). Lors de l'insertion d'une image, on spécifie sa taille en mètres. Lors du déplacement de la souris dans le plan, on peut avoir en tout temps les coordonnées cartésiennes (en mètres) associée à la position de la souris ou de l'élément sous la souris.

Concepts avancés

Certaines stations particulières ne réalisent pas un tri à proprement parler; elles transforment plutôt certain des produits en entrée en un certain nombre d'autres produits (exemple naïf : transformer des « gros » morceaux de plastique en un certain nombre morceaux de taille moyenne et petite). Encore une fois, une matrice permettra de définir quels pourcentages d'un produit donné est transformé en d'autres produits.

Le schéma peut être exporté sous la forme d'une image.

Il est possible d'enregistrer / charger son travail dans un fichier. On demande également un mécanisme permettant d'annuler/répéter des opérations lors de l'édition du réseau (undo/redo).

Chaque station et arc a une capacité maximale en kg/h. Le système doit clairement détecter si le système modélisé est correct (tout est connecté, les taux d'utilisation des équipements est inférieur à 100%, etc.) et offrir une rétroaction à l'utilisateur.

Consignes à propos du projet

- Le travail doit être réalisé en équipe de 4.
- Le forum peut être utilisé pour trouver des coéquipiers. Chaque équipe doit se trouver un **nom d'équipe constitué d'un seul mot**. Une fois l'équipe constituée, vous devez en aviser Martin

Savoie (martin.savoie.2@ulaval.ca) comportant le nom de l'équipe, et la liste des Noms, Prénoms, IDUL.

- Le projet doit être réalisé en Java
- L'environnement de développement est NetBeans
- Il est possible d'utiliser l'environnement Eclipse si tous les membres de l'équipe sont d'accord et que vous obtenez l'autorisation écrite du professeur.
- L'utilisation de toute autre librairie autres que les librairies standard Java 6 / 7 / 8 est interdite à moins d'obtenir l'autorisation du professeur par courriel.
- Les diagrammes UML doivent être produits avec Visual Paradigm.

Remarques

Certains éléments du descriptif de projet **sont volontairement flous à ce stade** (si nous vous transmettions des spécifications parfaites accompagnées de diagrammes UML... vous n'auriez pas à faire l'analyse et ce ne serait plus un projet complet).

Il vous appartient de faire la lumière là-dessus et de développer une bonne compréhension du projet. Vous serez appelé à poser des questions, en classe de même qu'à vos « conseillers » (Martin Savoie et Guillaume Lorquet) qui vous accompagneront tout au long du projet. Si vous doutez de quelque chose n'hésitez pas à poser des questions.

Livrable no 1 - Analyse

On vous demande de produire les livrables (artefacts) associés aux deux activités (disciplines) suivantes du Processus Unifié : Modélisation métier (Business Modeling) et Exigences (Requirements). Ces livrables doivent se trouver dans l'état où ils seraient aux termes de la première itération de la phase d'élaboration.

Pour connaître de façon plus détaillée les livrables (artefacts) attendus vous pouvez consulter le tableau suivant inspiré du manuel.

Discipline	Artifact Iteration	Incep. II	Elab. EL.En	Const. CL.Cn	Trans. T1..T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model SW		ss	rr	
	Architecture Document Data Model		s		
Implementation	Implementation Model ^(code)		s	r	r

Vous devrez consulter le manuel (et les notes de cours pour « découvrir » en quoi consiste chacun des artefacts demandés¹. Indice : une section du livre se nomme justement « Itération 1 de la phase d'élaboration »; voyez la table des matières². Notez-bien que vu la petite taille du projet, on s'attend à ce que des cas d'utilisation (au sens défini par le Processus Unifié) couvrant la totalité des fonctionnalités soient définis.

Finalement, vous devez en plus produire un plan de travail (diagramme de Gantt, Trello³, ou autre) illustrant le travail prévu par itération (au sens défini par le Processus Unifié) jusqu'à la fin de la session. On suppose donc que vous définirez la durée de vos itérations subséquentes et choisirez les cas d'utilisation pour lesquels vous réaliserez la conception (Design), implémentation et les tests à chaque itération.

Votre rapport doit prendre la forme d'un seul document continu (format PDF). Vous devrez également fournir le fichier Visual Paradigm (un seul fichier) utilisé pour produire les schémas. Les schémas doivent se trouver à la fois dans le fichier Visual Paradigm et dans le rapport.

¹ L'étudiant avisé se demandera sûrement pourquoi nous ne produisons pas une liste exacte des sections et schémas que devra contenir le rapport. La raison est la suivante : l'objectif du travail est de vous donner l'occasion d'apprendre en quoi consiste le processus unifié, et de le mettre en pratique. Vous serez évalué là-dessus. Vous fournir une simple liste de schéma à produire vous ferait passer à côté de cet objectif. Soyez cependant assuré que les lectures proposées vous permettent de le découvrir... de même que le cours qui est littéralement un guide sur « comment bien réussir le projet ».

² Erratum version française du manuel (mauvaise traduction) : La partie III devrait se nommer « Itération 1 de la phase d'élaboration » et non pas « ~~Élaboration de l'itération 1~~ »

³ <http://trello.com>

Consignes supplémentaires :

- Les diagrammes UML doivent être produits avec Visual Paradigm.
- Vous devez soumettre un seul fichier .zip appelé **NomEquipe-tp1.zip** contenant au minimum:
 - o **NomEquipe-tp1.pdf**
 - o **NomEquipe-tp1.vpp** (fichier Visual Paradigm)
- **Vérifiez la date et l'heure de remise sur Pixel**
- **Une pénalité de 1% par erreur de retard est comptabilisée**

Amusez-vous bien!