



## RESUME DU TRAVAIL PRATIQUE DE SEMESTRE

Candidat : Michele

Solari

Date de rendu : 17 juin 1999

Assistant : Christoph Hachen

# Simulateur graphique en Java pour lignes de production DSSPL

Pour effectuer des simulations de **lignes de production DSSPL** un simulateur en QNAP2 a été réalisé à l'EPFL. Pour simplifier la communication avec ce simulateur une **interface graphique** a maintenant été développé en **Java**. La raison pour le choix de Java était l'indépendance du système d'exploitation. Dans le futur l'interface graphique devra cacher complètement le simulateur et permettre l'écriture des fichiers nécessaires pour la simulation, la lancer et effectuer l'animation pour faciliter l'interprétation des résultats. A l'instant seulement le **module d'animation** et le module qui calcule la **performance** ont été codées. Avec les modules réalisés on peut d'une part vérifier si le simulateur sort des résultats correctes et d'autre part mieux comprendre l'évolution de la production avec une ligne de production DSSPL. Pour l'évaluation de la performance, trois paramètres ont été choisis. Ce sont les produits stockés, le rapport de livraisons effectués ponctuellement et le taux d'utilisation des machines. Ces paramètres sont calculés et écrits dans un fichier par l'interface graphique.

Pour le futur, il existe deux voies qui peuvent être choisies pour la continuation de ce projet. La première voie est de continuer avec le simulateur en QNAP2 et avec l'interface graphique codée en Java. Si on continue sur cette voie, il faudrait faire attention avec quels outils on écrit le code, parce qu'on risque sinon de perdre la portabilité.

La deuxième voie serait de commencer un simulateur codé en C++. Ceci permettrait de simuler des modèles proches à la réalité. Le prix à payer serait de devoir refaire tout le simulateur.

For the simulation of **Double Speed Single Production Lines** a simulator using QNAP2 has been developed at the EPFL. Now a **graphical user interface** has been written in **Java**. The reason for the choice of Java was to guarantee an independence of the OS. The goal of an GUI is to hide the complexity of the real simulator. The final GUI should write the necessary files for the simulation, launch the simulation and build an animation for an easy understanding of the results. Actually, only the modules making the **animation** and calculating the **performance** have been written. The developed modules give us the possibility to verify, if the simulator has no error and to understand better the evolution of the production with DSSPL. For the performance analysis three parameters have been chosen. These parameters are the components stocked during simulation, the ratio of products delivered at the right time and the utilization ratio of the machines. These parameters are calculated by the GUI and written in a file.

For future developments two ways can be chosen. The first would be to retain the chosen system, finish the GUI and continue evaluations using the QNAP2 simulator and the Java GUI. By choosing this way, the future development should be done with accuracy in order to assure that only classes of Java are used. Otherwise the independence will be lost.

Another way would be the development of a complete new simulator in C++. This allows us to simulate much more complex models that represent better the reality. The disadvantage would be to write a complete new simulator.