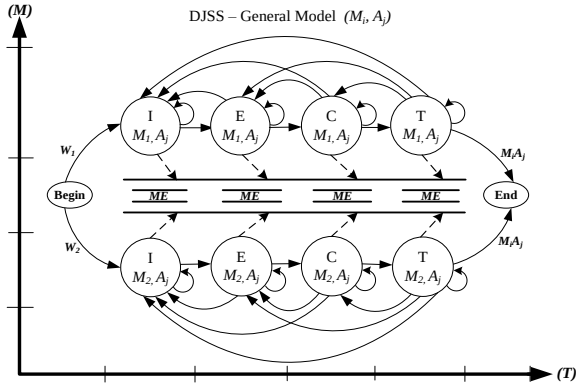
SCHEDULING OF TASKS IN A SOFTWARE DEVELOPMENT ENVIRONMENT USING A DYNAMIC JOB SHOP SCHEDULING (DJSS)

Sildenir A. Ribeiro[1, 2]
[1]Coordination of Industrial Automation
Federal Center of Technological Education of Rio de Janeiro (CEFET/RJ)
sildenir.ribeiro@cefet-rj.br

Eber A. Schmitz[1], **Antônio J. S. M. de Alencar**[1], **Mônica P. da Silva**[1]
[2]Tércio Pacitti Institute of Research and Computer Applications
Federal University of Rio de Janeiro (UFRJ)
{eber, ajuarez, monica}@nce.ufrj.br

RESUMO

This work proposes a Dynamic Job Shop Scheduling (DJSS) model to programing the task scheduling in a software production environment. The aim is to organize the insertion of tasks in the shop in a precedence order previously defined that obeys the mapping of the critical path and the critical chain to identify constraints in the productive process. The complexity and dynamism of the software development environment require a mature and adjusted process that enables management throughout its extension. Therefore, the Unified Process (UP) was used as the Software Development Process (SDP). The UP requires an efficient model for the processing of activities, given by the set of independent variables involved in the process. For this reason, a programming model based on dynamic task scheduling was developed. The problem around the DJSS is to identify one or more Production Lines (PL) in the shop with constraints of capacity at runtime. A capacity constraint is any element that disturbs the productive process, causing $L=\{Mi\ Aj\}$ to have delivery (D_A) of less than 100% of the artifacts, given by the set of jobs (j_n) scheduled. The resource with capacity constraint limits the production of the PLs, leading to bottlenecks in the production process. A bottleneck, in turn, is denoted by the production limitation of a machine (people) caused by one or more capacity constraints that imply maximum production, which in this case is defined by $D=\{M_i, A_j=100\%\}$. In this work, the maximum production is given by the set of tasks processed in each of the phases of the UP and according to the productive calendar, which delimits the deadline of each delivery and establishes the budget of hours for the shop. The set of tasks scheduled and "rescheduling" and properly processed in all phases define the total production of each machine. Figure 1 shows a general case of the DJSS model developed with two machines in an LP processing tasks in the four phases of the UP, knowing that one shop (S) is given by set of production lines (L). The general case can be denoted by: $L_M=\{M_i A_j\}$; $W=\{j_1, j_2\}$; $O = \{L_M + T\}$; and, $T = T_{M_1A_1} + T_{M_2A_1}(fase\ 1) + T_{M_1A_2} + T_{M_2A_2}(fase2) + (\ldots) + T_{M_1A_n} + T_{M_2A_n}(fase\ 4)$.



Where: I=Inception, E=Elaboration, C=Construction and T=Transition are the UP phases; T=Time; A=Artifacts; PL=Production Line; M=Machines; W= Predecessor Tasks; and J=Jobs.

Figure 1. Representation of the general case of DJSS

KEY WORDS. Dynamic Job Shop Scheduling**,** Software Development Process, Software Development Environment.