WHITE PAPER

MANAGING RULE CHANGES WITH WORKFORCE SCHEDULING SOFTWARE

By Terry Schilling
Marketing Manager
Tugboat Software Inc.
Newport Beach CA 92660
www.tugboatsoftware.com

August 4, 2004

TABLE OF CONTENTS

THE PROBLEM

When large manufacturing, retail and services operations consider implementing an automated solution for workforce scheduling, they face a conundrum. Namely, how can a software application accommodate the range of business rules and policies that govern workforce scheduling when rules and policies must change to satisfy new objectives? Business rules and policies that govern workforce scheduling are not like accounting rules that may remain unchanged for very long periods of time. This white paper describes how Tugboat Software's solution, Schedule Optimizing Software (SOS), addresses the question of rules changes and discusses related concepts used by SOS. This paper is not intended to be a comparative analysis with other offerings or methodologies.

INTRODUCTION

The SOS application automatically generates workforce schedules in large service and production operations; typically with 200 to 1000-plus employees. The software includes standard tools along with a unique application of technology that together, combine business rules with service or production plans to automatically create job assignments for employees. Some of the tools will be familiar to anyone reviewing HRIS solutions that are available today; a Graphic User Interface (GUI) with pull-down lists, along with parameters and data stored in a fourth generation embedded database server. Many business rules can be stored as system parameters and modified by the user at will. Other rules, and their interdependence, are controlled through a library of procedures. These parameters and procedures are then processed with a proprietary mathematical algorithm. The result is a deterministic, yet flexible workforce scheduling system. It is unencumbered by the necessity to reengineer a significant body of code in order to accommodate changing business rules and objectives.

SOLUTION OVERVIEW

Any rule that can be defined either by a quantity, Boolean logic such as true or false, or can be selected from a list, uses GUI to store and then modify the rule when necessary. Rules, or policies that govern the priority and interdependence of rules, business objectives, and policies are managed with a procedure library. Using the procedure library and a proprietary math engine, job assignments are then optimized in a completed schedule that conforms to the working culture and business requirements of the organization.

PARAMETRIC RULES

Using parameters to modify rules is ideal for those rules that have an identifiable pattern, and that change within a predictable range of variability. They are easily modified using GUI. For example; when deciding if an employee can take a discretionary day off, a key parameter is the number of employees, among those having the same skill, that remain available for assignment or on-call. This quantity can be modified up and down to reflect seasonal demand. Parametric rules may span multiple levels, such as shifts, departments, crews and jobs.

Parametric rules can be enabled with GUI if they meet any of the following tests;

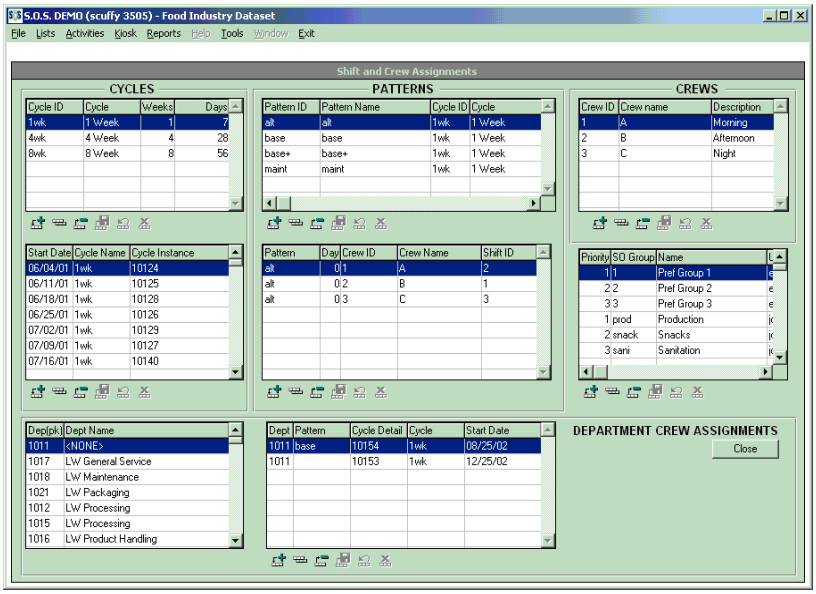
- Is it quantifiable?
- Does it follow Boolean logic, e.g., true or false?
- Can all the alternative choices be listed?

Examples include:

- Shift, crew or team rotation parameters.
- Number of days off, as a test for various conditions.
- Skill and/or grade designation and progression.
- Employee preference for shift or job assignment.
- Absentee condition and history and many, many more.

Wherever possible, SOS brings parametric rules to the user with GUI.

SHIFT & CREW ASSIGNMENT SCREEN



This sample screen, using a typical GUI user interface, controls the pattern of days when a crew is assigned to work. It also controls a corresponding cycle, which defines how often and when the pattern is repeated.

- A “pattern” is defined by a crew, shifts, and a sequence of days, i.e., Monday thru Friday or Tuesday thru Saturday.
- A crew is a group of employees that is required to work for a specific pattern of days.
- Some departments and their crews, such as maintenance, may have a fixed pattern.
- The pattern for other departments, such as services or manufacturing, may vary depending on the needs of the business, for example when supporting a 24 by 7 operation.

In this example, each of the parameters has an allowable range. The parametric data from this screen are in turn called by a procedure that reflects the rules within a particular facility. Once the parametric rules are defined, this screen enables one to change the pattern and cycle.

POLICY RULES

Business objectives, as they pertain to workforce scheduling, are often expressed as a policy or rule. These “policy rules” do not meet the tests for parametric rules as they usually cannot be quantified, matched to a Boolean operation or listed. Policy rules are often interdependent where one rule will take precedence over other rules. Often, they are hierarchical, triggering a yes or no answer to a test. Usually, a single test is only part of a proscribed sequence whereby a condition is tested against one rule, and then the next rule, and so on until all the rules related to the policy have been tested. While policy rules may use data from parametric rules and the results are quantifiable with an outcome that can in turn be measured, overall, policy rules are rarely quantifiable.

Examples of policy rules include:

- Conditions under which a junior employee can be bumped from an assignment in order to accommodate a more senior employee.
- When employees are restricted to one shift or crew, or when they can float between shifts and crews.
- When an 8-hour job assignment can be split into multiple segments. For example, splitting a job into two 4-hour segments that can then be assigned as overtime for employees on the two neighboring shifts. Or when three 8-hour assignments can be converted into two 12-hour assignments.

Within the same facility, policy rules are often applied differently to different groups of employees. Policy rules may change from one department to the next, and will have different priorities when compared with other policy rules. Policy rules, by their nature do not pass the test for parametric rules. SOS uses a procedure library for managing policy rules.

PROCEDURE LIBRARY

A “procedure library” is a collection of software routines that, taken together, encode the policy rules that govern the workforce scheduling needs of a business. Policy rules are modified or added as business requirements change. Though separated from the user interface, in SOS the procedure library is relatively easy to change. First of all, the procedures are modular. In a one-to-one relationship, a specific procedure executes a specific policy. So it is easy to find the procedure involved when change is necessary. When a rule does change, or a new rule must be added, programming skill is required. However using a procedure library, such changes are made without having to search through large routines and overhaul other procedures. Programming costs are thus kept to a minimum. A programmer, familiar with the development tools used for SOS can be trained to conveniently combine procedures to represent new business rules.

PROCEDURE LIBRARY - AN ILLUSTRATION

ASSIGN OPEN CLASS
[In Seniority Order]

A. Find Open Job: Earliest Start / Qual'd - on Emp's Bid Shift.

ASSIGN
OR

B. [If Advance] Find Open Job: Earliest Start / Qual'd - on Preceding Shift.
(2nd looks to 1st, 3rd looks to 2nd) ….. assuming 1st does not go to 3rd

ASSIGN
OR

C. Bump Most Jr. already assigned:
as long as 'Bumpee' is also Jr. to our current Emp / Qual'd
– on Emp's Bid Shift.

ASSIGN
OR

D. Find Open Job: Earliest Start / Qual'd - on Emp's Following Shift.
(see shift prefs: 123,231,321)

1 2 3
2 3 1
3 2 1 …..not strictly following

ASSIGN
OR

E. Bump Most Jr.: 'Bumpee' Jr. to our Emp / Qual'd – on Emp's Following Shift.

ASSIGN
OR

F. LOOP back to (D) for 2nd following shift. (according to personal shift prefs)

Shown above is the English language specification for a procedure from the SOS procedure library. It is the main driver, a fairly high level procedure. It calls other programs from the procedure library used in the process of assigning workers to jobs for which they are qualified. Notice how employees are assigned according to their seniority and if necessary, junior employees can be bumped from an assignment to make room for the senior employee. It illustrates how one rule and its priority interact with other rules, and it suggests how policy rules are managed at a higher level and with greater flexibly by using a procedure library rather than with parameters alone.

Having a business rule defined as a policy in a procedure library also enables SOS to address business objectives that go beyond the policy rules referred to so far. If procedures are called from the library in a carefully proscribed sequence, and are then processed with the SOS math engine, the user is able to optimize the assignment of the entire group of employees according to a high-level business objective.

OPTIMIZATION, A MATHEMATICAL ALGORITHM

The SOS application provides more than what can be achieved with a purely rules-based system, one that assigns employees strictly according to a set of rules. The advanced algorithms in SOS's proprietary math engine optimizes the job assignments for any schedule. Job assignments are "optimized" when the greatest number is filled - the highest number mathematically possible - while conforming to both policy constraints and a hierarchy of preferences. The goal of optimization is to make the best possible use of the skills of each and every employee. Put another way, optimization makes the best possible use of the all the skills and preferences throughout the entire pool of employees.

In order to optimize assignments, an understanding of the interrelatedness of policy rules is critical. In fact, this interrelatedness expresses part of an organization's business logic, and in real business situations, all policy rules are not created equal. One rule will take precedence over others. One rule will affect how, or even if, other rules are applied. This is what distinguishes SOS. This software application captures and implements business logic at more than one level; a procedure library where policy rules are stored, modified, and combined with parametric data, and; a math engine where an overarching policy is achieved by optimizing job assignments with an advanced mathematical algorithm.

Optimization can be thought of as the policy that expresses what is most desirable for a large subset of employees. It can be used to achieve varying objectives. For example employee assignments can be optimized so that;

- The preferences of each employee, for shift and crew or job are considered while still filling all the assignments according to seniority and skill; or
- The maximum number of employees wishing to take time off is satisfied while still meeting the need for overtime; or
- Requests for overtime are evenly balanced among those qualified; or
- The maximum number of employees wishing to start at the earliest job-start time is achieved according to seniority; or
- When faced with changes in demand for service or production, the highest and best use of the skills in the workforce is achieved.

Depending on the desired outcome, there are any number of ways that job assignments can be optimized. However, optimization makes sense only in the context of the culture in a specific facility. While scheduling policies reflect the culture of an organization, they also reflect the unique history of each specific facility. So, the criteria for optimization will differ not only from one organization to the next, but will also differ from one facility to another within the same organization.

Business managers face a continual challenge in using labor resources efficiently when confronted with changing demands for service and products. Modern organizations, whether in service and manufacturing, are now able to change the configuration of their operation at will. Facilities are designed so that service areas can be expanded or contracted, and production lines reconfigured between shifts. The accompanying scheduling requirements are daunting and can put tremendous pressure on both the workforce and management personnel.

To respond to this situation, service and manufacturing employees are required to have multiple skills. Their level of proficiency in these skills makes them more or less valuable to the organization. So now, an employee is identified not by their nominal job, but by the variety of skills they command and can perform. With a large workforce, management faces a subtle problem here. How to assign employees so that the greatest possible number work at their highest and best skill level without having to resort to the use of excess overtime along with its higher cost? Indeed, as labor is usually the major cost of service or production, using employees at their highest and best skill level is essential to the efficient operation of the organization. The ability to achieve this goal for efficiency in a predictable manner argues persuasively for the utilization of the unique optimization technology in SOS

IN SUMMARY

The SOS application from Tugboat Software combines parametric rules, those that are quantifiable, with policy rules, those that are driven by current and changing business needs, and provides a scheduling solution that is optimized for top-level business objectives. By combining parametric tools, a procedure library, and a math engine, SOS translates business rules into an accurate and efficient workforce schedule in a deterministic manner.