TPO: A System for Scheduling and Managing Train Crew in Norway

João P. Martins
SISCOG
Campo Grande 378 – 3º
1700-097 Lisboa
PORTUGAL
jpmartins@siscog.pt

Ernesto Morgado
SISCOG
Campo Grande 378 – 3º
1700-097 Lisboa
PORTUGAL
emorgado@siscog.pt

Rolf Haugen
NSB BA
Oslo S
0048 Oslo
NORWAY
RolfH@nsb.no

Abstract

This paper reports some results of a long-term work in application of AI techniques whose ultimate goal is the development of tools for resource scheduling.

Most of the efforts so far have been devoted to crew scheduling in railways and the resulting tool is called CREWS. CREWS contains the basic knowledge for crew scheduling, remains constant across companies, and only needs to be extended with the particularities of each one (domain, labor rules, scheduling strategies).

The first successful deployment of CREWS was with Dutch Railways, as reported in (Morgado and Martins 1998). Since then, the initial modules of CREWS were deployed in several major European railways.

In parallel, new modules of CREWS were developed and deployed (the Roster Scheduler, the Allocator, and the Short-term Scheduler). This paper addresses the development of an application for the Norwegian State Railways (NSB) that includes these new modules. This application was deployed in 2000. The system is being applied to schedule and manage the work of 1,800 persons: 1,000 engine drivers and 800 guards (ticket inspectors) allocated to 39 bases across Norway.

Problem Description

European railways are going through a period of deep change. This is due to the European Directive that separates operation from infrastructure and to the principles of open market and free movement of persons and goods. As a result, railways companies are introducing new management styles to improve the results of the business. In particular, new tools are being searched to improve the use of human and material resources. This trend is similar to what was felt in other areas of transportation where the benefits of IT solutions have been extensively explored.

Copyright © 2003, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

In order to provide services, a railway company needs to manage three main resources, *track*, *rolling stock* (engines and cars), and *crew* (that operate the rolling stock). The management of crew is constrained by the timetable in effect, the rolling stock roster, the labor rules, the number of available workers and their skills, and a large number of other operational constraints. From a global point of view, this problem is addressed in a sequence of phases.

Phase 1: Long-term scheduling. Produces a plan of the work without taking into account the names of workers (but the available skills) nor the dates where the work is being planned (but its frequency).

Long-term scheduling deals with the regularity involved in the operation (there are services that are repeated at regular frequencies, e.g., daily, Monday to Friday, just on Fridays), leaving the exceptions and the irregularities to be handled at the short-term scheduling phase. Long-term scheduling is done in two sub-phases:

(a) *Scheduling duties* (or shifts) arranges tasks into a set of *duties* (sequences of tasks to be done by one person in one day). Constraints in this phase include, among many others, maximum duration of a duty, space continuity between adjacent tasks, transfer times between tasks with different equipment, places and times for meal breaks, compatibility of route and equipment skills. Constraints may be *hard* (cannot be violated) or *soft* (may be violated, but violations should be avoided and recorded).

(b) *Scheduling base rosters* arranges the duties produced in (a) into sequences of duties, rest time, and days off. The persons that will be associated with a base roster (in Phase 2) will perform the work in rotation. Constraints in this phase include, among many others, maximum week working time, rest time between duties, weekly days off, and available skills.

Phase 2: Allocation. Associates crew members to the weeks of the base rosters from Phase 1 (b), producing an *allocated roster*. All persons in a roster must have the skills required by the roster. The result of this phase can be seen as a *daily plan* obtained by instantiating the allocated roster for a certain timetable period.

Phase 3: Short-term scheduling. Deals with the irregularities that were not considered during long-term scheduling, such as services that operate in particular calendar days, rather than being repeated at a certain frequency. It also amends the daily plan for a certain number of days either because some of the tasks have changed (e.g., a change in the timetable due to track work) or because a person is not available to work in a certain calendar period.

Short-term scheduling works with particular days and with particular persons. The constraints handled in this phase include all the constraints in the duty and roster scheduling phases plus the constraints associated with particular persons, such as maximum working hours per year and holidays. Due to operational reasons, the violation of soft constraints becomes more permissive.

Phase 4: Dispatching. Is performed on the day where the operation takes place. It is similar to short-term scheduling but works in real-time. It takes as input the results of Phase 3 and real-time events. The changes introduced at this level may be due to delays, breakdowns or unexpected absence of crew.

Phase 5: Recording and reporting. Checks the work performed, taking the results of Phase 4, comparing it with the work that was planned in phases 2 and 3, updates personnel records, and feeds the payroll system.

Discussion

Phases 1, 2, and 4 address the problem of crew scheduling, which is known for being NP-hard (Garey and Johnson 1979). The problem is even more complicated when the quality of a solution depends on subjective constraints that are hard to describe in quantitative terms. Human planners, that acquire most of their knowledge through experience, usually carry out each phase manually.

Besides the human skills, the problem has to deal with the ever-changing data. Due to time constraints, Phase 1 is started well before its inputs – the final timetable and rolling stock plan – have been completed. Thus, planners not only have to deal, within a short period of time, with huge amounts of data to produce schedules (*scheduling*), but also have to handle changes that constantly pop up (*re-scheduling*), most produced by different departments and often incomplete and inconsistent as a whole. Phases 3 (programmed changes) and 4 (non programmed changes) correspond to re-scheduling problems. Changes must be incorporated in the schedules without much disturbance.

Another aspect that puts a high demand on planners is the increasing complexity of labor rules, required to comply with increasing social benefits given to workers. Because of this, scheduling crew is considered to be much more difficult than scheduling rolling stock (equipment) or producing the train timetable (scheduling the track).

In (Morgado and Martins 1998) crew scheduling is compared with job shop problems (Fox 1987) (Smith 1889). Here, we just outline the main differences between these types of problems: (1) Crew scheduling *has to deal with space constraints* to prevent space discontinuities, having to position crew, as passengers, where they are needed; (2) Crew scheduling *handles complex frequencies*, week frequencies (a train runs only on weekends), year periods (only during summer), and special days (only on holidays); (3) Duties *do not have fixed times*, as shifts in industry; (4) Labor *rules are very complex* and change often due to unions pressure. Worse, planners must account for *exceptions to the rules*. This requires flexibility in accommodating exceptions and in changing rules, without compromising the efficiency of the process.

Crew scheduling has been approached by traditional programming (supported by operational research), but the results obtained with automatic "black box" optimization algorithms had only limited success and have proven to be unsatisfactory in the following aspects: (1) when faced with a full size problem, these solutions tend to need computational resources that by far exceed what is available; (2) they cannot provide explanations about the decisions placed in the solution; and (3) solutions cannot be manipulated by human planners to adapt them to changing circumstances or to ill-represented constraints.

Application Description

The application was built with a standard scheduling tool (CREWS) with the following characteristics: (1) Provides full integration among the several phases of the problem, in terms of data and of functionality; (2) Uses AI techniques as an alternative to traditional computer technology; (3) Provides different modes of operation, manual, semi-automatic, automatic, and any arbitrary combination of them; (4) Detects the consequences of changes on the schedules produced; (5) Enables the planner to interact with the system, to propose alternatives, or to query decisions (a "white box" approach); (6) Is adaptable to changing circumstances; (7) Handles different types of personnel with different sets of rules and constraints.

The system reported here, called TPO, consists of the generic CREWS and CREWS_NSB (the part adapted to the rules and conditions of NSB). The goals set up for TPO were: (1) to provide decision support in crew scheduling and management; (2) to speed up the scheduling process; (3) to make scheduling more reliable; (4) to take a considerable workload from the planners; and (5) to keep the role of planners in taking the decisions.

TPO is composed of the modules shown in Figure 1: the Data Manager represents the interface between TPO and outside systems; the information is stored in a central database that is accessed and updated by components for producing duties (Duty Scheduler), rosters (Roster

Scheduler), for allocating staff to rosters (Staff Allocator), for handling short-term changes (Short-term Scheduler), for recording and for reporting the work done (Work Reporter and Work Reporter). The database, Data Manager, and Duty Scheduler are located in Oslo. The other components are located at the personnel bases across Norway.

Both TPO and CREWS are implemented in Allegro Common LISP and CLOS. The central server runs in a Unix machine. Clients run under the Windows NT / XP. The database is Oracle. Communication with outside systems is done via file transfer. Communication among modules is done using TCP/IP.

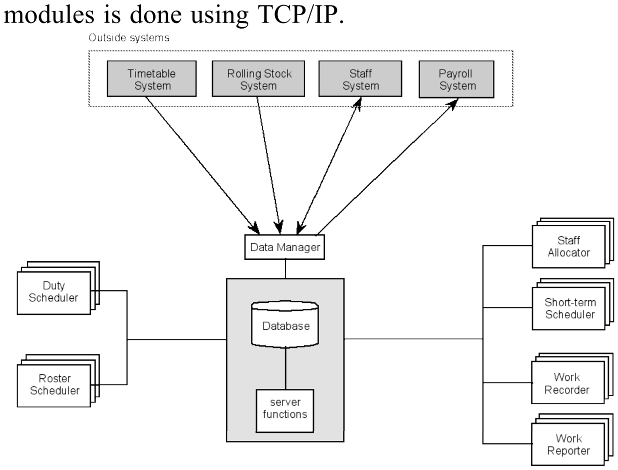

Figure 1: Architecture of the TPO system.

Data Manager. Handles communication between TPO and outside systems, supports the preparation of the input data, handles change in data, and maintains the consistency and completeness of data.

The Data Manager is also responsible for deriving the relevant personnel tasks out of the timetable and rolling stock roster. The generation of the personnel tasks is far from being trivial. Besides having to use different rules for different classes of personnel (drivers and guards), it has to derive many tasks that are not explicit in the input data such as attachments and detachments of rolling stock, attendance tasks and so on. Further details of the Data Manager can be found in (Morgado and Martins 98).

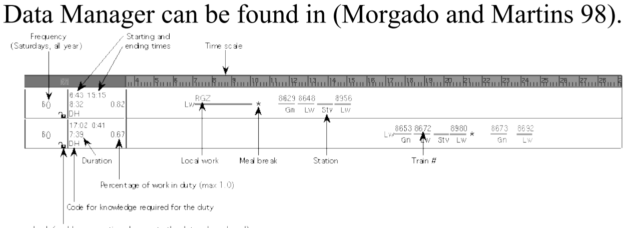

Figure 2: Example of two single duties.

Duty Scheduler. Assembles tasks into duties. A duty is a sequence of tasks that can be done by a crew member at a certain frequency (e.g., Monday to Friday). Duties may either be *single* (start and end at the home base without a sleep outside the base) or *composed* (start at the home base and return to it after one or more rests outside the base). Figure 2 shows examples of single duties. Duties have to satisfy several constraints: maximum number of working hours without a meal or a rest, maximum number of rests outside the home base, space continuity, maximum number of driving time, compatibility of line, rolling stock, and train knowledge, and so on. The Duty Scheduler provides four modes of operation as described below. Further details of the Duty Scheduler can be found in (Morgado and Martins 98).

Roster Scheduler. Assembles duties into base rosters, following labor rules that impose constraints upon the roster construction. It works from the perspective of a personnel base and a certain set of personnel skills.

Upon loading data, the Roster Scheduler identifies the days of the week in which the loaded duties are performed and places them in the appropriate weekday column in the graphical interface (Figure 3). Duties are shown in the week days that they occur. Each duty is shown as a solid horizontal line, on top of which is a duty identifier. The Roster Scheduler represents the roster in terms of an abstraction of duties, using just a timeline and a number to represent a duty, avoiding the detail of the tasks that compose the duties, but it is still possible to see a duty day in terms of tasks. The weekdays of the roster form a week (seven consecutive days, starting in any weekday) and will be linked to calendar days in the Short-term Scheduler.

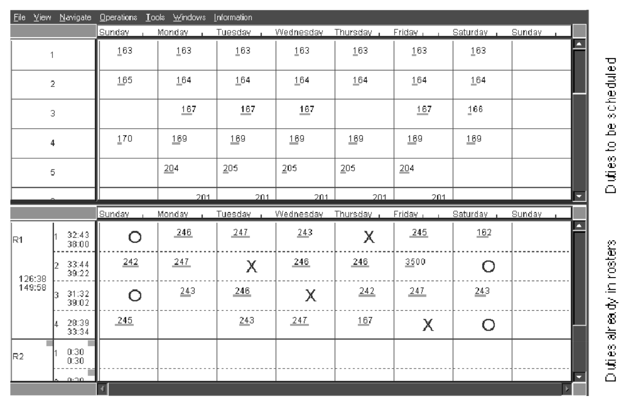

Figure 3: State in the Roster Scheduler.

The Roster Scheduler resorts to state-space search, using a modified version of beam search with heuristics. A state is a pair containing the duties that have to be scheduled (the candidates) and the rosters that have been constructed so far (Figure 3). States are generated by taking one candidate duty and placing it in a roster. The search is guided by a *strategy* consisting of: (1) A process for selecting the initial state; (2) A set of operators to generate the successors of a state that resort to constraints and heuristic knowledge to limit the number of successors; (3) An evaluation function, composed of cost and heuristic functions; (4) A test for deciding whether a final state was reached. The Roster Scheduler provides strategies, each one appropriate for a certain type of roster or for a certain type of operation.

The construction of the rosters can be done in for modes of operation:

Manual mode. The user specifies the number of weeks in the base roster. Afterwards, using drag-and-drop, the user tells the system either the duties to be moved from the candidates to the roster or the type of day off to insert in the roster. Whenever a duty is placed in the roster, the system automatically computes the number of hours worked and the amount of overtime (for extra payment), both at the week level and at the roster level (Figure 4). The system verifies all constraints imposed upon the roster and tells the user the constraints that are violated by the operation. If the user chooses to violate a constraint, the roster is shown with a violation indication. Pointing at the violation indication icon generates an explanation of the violation. The user may also move duties from the roster to the candidates (removing the effect of any previous decision – forward backtracking), from one roster week to another roster week or to another roster, or can do traditional backtracking by moving into a previous state;

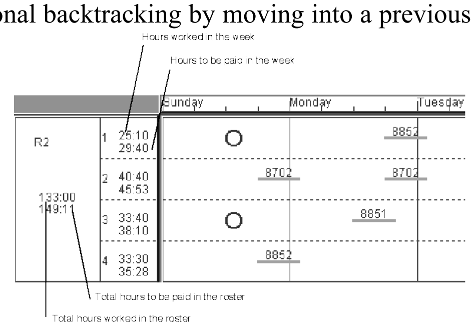

Figure 4: Work time information on a roster.

Semi-automatic mode. The system gives hints about how to construct the roster. The system computes the most constrained day of the roster and provides suggestions of how to fill this day with duties from the candidates, following the selected strategy. The role of the user is to select the proposal that he thinks is best;

Automatic mode. The system decides the number of weeks in the roster and schedules duties and days off, following the strategy selected by the user;

Mixed mode. Combines the previous approaches. The user constructs the roster by resorting to an arbitrary combination of the other modes of operation. The mixed mode shows the decision-support philosophy that was incorporated in CREWS since its inception. It provides a full cooperation between the user and the system, showing what is going on, providing explanations about the decisions taken by the system (with an explanation facility provided by the Roster Scheduler), enabling the interaction of the user on the work being done by the system, and taking the bulk of work from the user, when he selects to do so.

Staff Allocator. Takes the base rosters produced by the Roster Scheduler and the information about personnel and allocates a person to each week of the base roster. For example, in Figure 3, the system has to associate one person to week 1 of roster R1, one person to week 2 of Roster R1, and so on. The assignment must take into account the appropriate number of hours according to the terms of appointment. Currently, the allocation is only performed manually, based on seniority and preferences from the personnel. This module of CREWS is being extended to allow semi-automatic and automatic allocation based on preferences.

After the allocation has been completed, the system propagates the allocation to any period with the validity of the timetable. In the propagation phase any breaches to rules over individuals are detected.

Short-term Scheduler. Deals with day instances (calendar days) and with individual staff. It handles special days (e.g., Christmas) and information that is not placed on the regular operation schedule. The allocated rosters constitute the plan for the regular work and, if the world was static, nothing would have to change. However, changes pop up constantly.

The Short-term Scheduler receives the allocated rosters and the changes that have occurred in the meantime (special days, addition and cancellation of trains, modifications in trains, staff absences, holidays, etc.).

The Short-term Scheduler enables to change the schedule after the allocation has been performed on base rosters. These changes may be done from months to a few hours in advance. The Short-term Scheduler includes the functionality that is available in the Duty Scheduler, Roster Scheduler and Staff Allocator, however, these are applicable over calendar days. The Short-term Scheduler checks the staff records of any individual staff members in the schedule, records the changes, finds their implications (propagating the implications to any day or week of the scheduling period), and enables the production of a revised schedule to account for those changes.

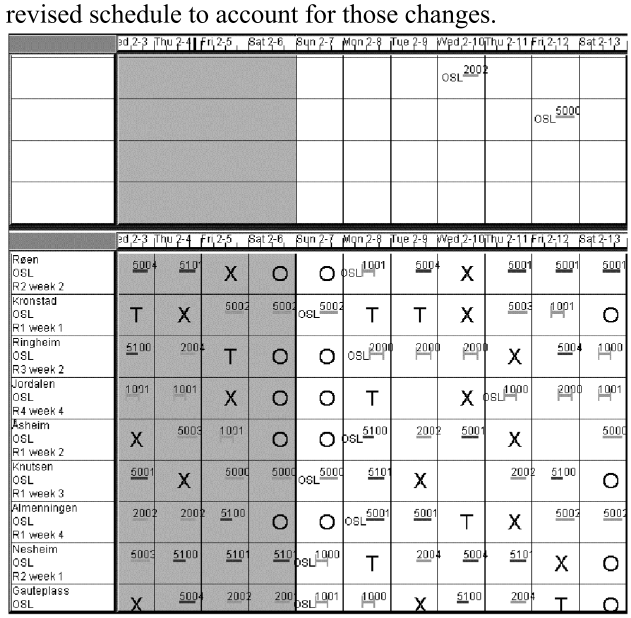

Figure 5: Short-term planning (duty level).

The Short-term Scheduler considers a period of the year and works for those days (shown in a white background in Figure 5). The users are presented with screens that are similar to those of the previous components, but adapted to the task at hand: (1) The screen is split into two parts, the top part shows the work that has to be placed in the schedule and the bottom part shows the schedule in effect; (2) The left column of the bottom part shows the names of the persons that are allocated to the lines of work – thus, each line following the name of a person shows the work scheduled for that person in different days; (3) The time scale may be zoomed to a few hours to several days; (4) The planner may either work at the duty level (allocating duties to crew members, using the concepts of the Roster Scheduler) or at the task level (allocating tasks to duties, using the concepts of the Duty Scheduler).

In this phase, constraints are handled at a more permissive level. Constraints that would never be violated at the long-term phase, are allowed to be violated at this phase, in order to guarantee that the services are fulfilled. The flexibility of constraint violation is associated with additional pay for staff. Thus, although more violations are allowed, these must be done parsimoniously because their violation influences the cost of the solution.

The Short-term Scheduler provides the four modes of operation available in the Duty and Roster Schedulers.

In manual mode, absences, sickness, or other reasons for not performing the work, may be introduced for any individual staff member. Upon entering the information, the system flags up all the duties that were planned to be done by the staff member during the absence period. Using drag-and-drop the users may amend the schedule at their will. The system's role in this mode of operation is the verification of the constraints placed upon the schedule, either related with the duties, the sequence of the duties or the individual crew members.

The manual mode also enables the selection of any candidate (non-scheduled duty or task) and filters the lines in the schedule where the candidate may be accommodated. This is an essential functionality for a re-scheduling tool.

The semi-automatic and automatic modes, although working according to a strategy as described for the Roster Scheduler, enable the selection of a subset of tasks (and / or duties) from the candidates in order to create the initial state of the search. The strategies follow reparative methods in order to accommodate the unscheduled tasks in the already scheduled duties.

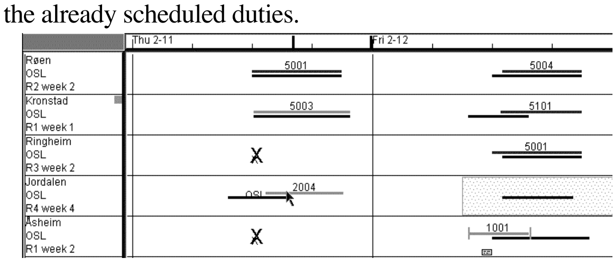

Figure 6: Deviations face to what was planned.

Work Recorder. Receives and records the actual working hours and overtime on an ongoing basis as the result of delays, absences, etc. In the Work Recorder, the work of any person for a day may be visualized and compared with the original schedule (Figure 6). For each duty, the user can modify values, such as the start and end time or stations. Tasks may also be added or deleted.

Work Reporter. Receives the results of the Work Recorder, performs all calculations for allowances and overtime according to rules. Provides information both to the payroll system and to the staff system (to record the equipment and routes followed by the staff, which is important in order to guarantee that knowledge of staff is kept up to date). The Data Manager performs this task.

Uses of AI Technology

AI technology is the backbone of the operation of the system. The scheduling algorithm is implemented as an hybrid constraint satisfaction and state-space search. The most visible part is state-space search, using a modified version of beam search (Bisiani 1987). The search tree generated serves as the unifying media of all modes of operation. Whenever the planner uses manual mode, the system generates the selected successors in the search tree. If the planner decides to remove any tasks from the duties, the system generates successors of the current state that correspond to the removal actions (forward backtracking). If the planner decides to undo some action, he just has to move up in the tree to backtrack to a previous position. The search tree and its states can be inspected at any moment during the search process.

Abstraction is used in most phases of the scheduling process to reduce the amount of detail present in this domain. Without it, none of the other techniques used in the system would be capable of coping with the huge number of alternatives present, many of them having only minor differences.

The concept of strategy defined above is also central to the success of both the automatic and semi-automatic modes. By combining clever heuristics (both in the generation and in the expansion of nodes) with adequate cost functions, the system can be fine tuned to optimize the relevant criteria chosen by the customer.

Knowledge is represented in the system using a frame-based formalism. Labor rules are represented in a mixed declarative and procedural language with a specific interpreted developed by SISCOG. This enables the separation of the rules from the code and the modification of the labor rules by the user company.

Another aspect of AI that is omnipresent is the use of constraints. These are used by the automatic and semi-automatic modes to select the most constrained tasks as preferential tasks to be used in node expansion.

Data dependencies are used in the Data Manager to find out what concepts depend on a given concept. These dependencies are set up in a way that was influenced by TMS systems (Doyle 1979, Martins and Shapiro 88).

Application Use and Payoff

A cost benefit analysis was made by NSB before the contract was signed. No net benefits were expected for the first year in production. It was assumed that the users needed time to learn how to use the application and that the efforts needed to take a new tool into use would outweigh the potential benefits during the first year. For the second year in full production differentiated benefits were expected for different parts of the system. On average, 2-3% cost reduction was expected as a consequence of reduced time to produce the schedules and improved efficiency of the schedule itself. On two points expectations were higher. (1) NSB expected to reduce the amount of overtime needed by 6% as a consequence of improved plans for long term and the use of the Short-term application. (2) Manual labor needed for payment calculation, was expected to be reduced to half, due to full automation of this function.

Between 2000 and 2002, NSB had two timetable periods in each calendar year, one from early January to the middle of June and one from June to early January. Due to various circumstances long term scheduling for guards went directly into production during the second timetable period of 2000 and has been in full production since then. For drivers long term scheduling has been in production in two regions since 2001. The largest region (Oslo area) is not yet 100% into production. Short-term scheduling is still in the pilot-implementation phase.

Quantitative measurements. The ideal effect study would be a measurement of how much time a planer uses to make a complete schedule before and after using the new tool. For a valid comparison all other factors should be kept constant. In NSB, these were far from constant.

Many changes took place in the period the system was implemented. Production was subdivided into business areas for guards. Freight was split from production and is now handled by a separate company. A new rolling stock system was implemented and numerous organizational changes affected the way tasks are subdivided among planners. A before-and-after study in this situation requires careful analysis of data.

Improvement in the quality of the plans themselves is expected to yield the largest benefit. This cannot be analyzed by comparing with the plan form a previous timetable period. The ideal experimental condition would be to have two sets of planners working in parallel on the same planning task, one using only manual methods, the other using the system. Then both time used and the final quality of the result could be compared. This kind of experimental setting cannot be established without considerable costs. Since the system was taken into production for guards first, it has been possible to make some comparisons between planning for guards and the corresponding manual planning for drivers for the same timetable period, but the planning task is not identical. Planning for drivers is more complicated because they have additional technical preparation- and disposal tasks. The real-time dispatchers found, however, that schedules made with TPO for guards contained practically no errors, while a considerable number of errors were found in the schedule that was produced manually for drivers. The number of errors is one indicator of quality. The other key indicator is the number of duties. For this, it has not yet been possible to create basis for comparative analysis but work on quantification of benefits has just started.

Qualitative evaluation. Anticipated effects from the use of long term scheduling are already apparent. NSB has more effective production of plans. Planners use less time to produce a complete schedule with ready printouts and all calculations of statistics. Management has been able to check consequences of new rules while bargaining with the unions was taking place. This had never been done before. Gained time is primarily used to make better plans and to create more alternatives for the next and later timetable periods. This also had never been done before.

Since all labor rules are built into the system, detailed knowledge of all rules is not required of all planners. Differences in rules and regulations for drivers and guards are now seen as insignificant. As a consequence, planners can now perform scheduling for both personnel groups and the planning units can be integrated. It is also expected that the knowledge built into the system will make it easier to recruit and train new planners. But this remains to be confirmed in practice.

More reliable and complete statistics have already contributed to improvement of cost control and pricing of services. Hidden costs have become visible and the use of the system has contributed to a reorganization of the scheduling process and a clarification of many issues that needed to be sorted out. Among these are numerous rule clarifications. This started in the system development process and continued in production.

Changes in working methods. In the organization that existed before the system, the same data was processed in sequential order by a string of organizational units from long term scheduling through payment calculation. Planning for guards and drivers was done in parallel within different departments. The same data was registered and processed many times with different tools. The process as a whole was clearly inefficient.

Requirements for a system that integrated the process were published in the call for tenders in 1996. NSB had a vision of a highly integrated planning process.

It turned out that no customized system that integrated all the steps in the required way existed. No suppliers (including SISCOG) had ever made any software for this. In this perspective it is not right to say that the system crated changes in working methods. At the initial stage, a perceived need for organizational change created the system. The vision of an integrated organization came first. Then the appropriate system was designed. Currently when he system is in production, we find that implementation of organizational changes lag behind. Change in working methods does not follow automatically when a new system is taken into production. Needs for changes must be identified, alternatives considered and management must make decision about regrouping of units, redistribution of tasks, training, transfer of personnel, etc. This process has started and will continue. At present, however, the initial vision of the wholly integrated organization remains a vision while the prerequisite system support is in place.

Application Development and Deployment

The project started in January 1998 following a call for tenders in EU, launched in December 1996. In the call for tenders, NSB produced a detailed set of requirements based on the analysis of current working methods. The job was well done, but as the project progressed, NSB realized that it is impossible to completely specify the application in advance. The initial requirement established the basis for common understanding of basic principles as to how the application should work. The development was done in several phases, each one addressing one of the modules of the system. The delivery was incremental, starting with non-official versions and progressing towards the final official version.

From NSB, it was necessary to understand the logic and constraints of the way CREWS is constructed in order to anticipate the kind of questions that SISCOG would ask. This understanding increased dramatically when the first versions were tested. Testing revealed ambiguities and situations that the application could not handle, or handled in a wrong way. A number of problems anticipated by NSB proved to be non-existent or already solved in CREWS. Fine-tuning continued after part of TPO was into production.

For SISCOG, there were two main challenges. The first was to customize the existing modules of CREWS (Data Manager and Duty Scheduler) to the reality of NSB. This reality was known to have highly complex rules (and no common interpretation of the rules among planners). The second was to develop new modules of CREWS (keeping them generic) alongside the customization of these modules to NSB.

On NSB side, the project was managed by two levels of authority: (1) the *steering committee*, composed by key persons in the organization such as the director of operations, the IT director, the IT manager of operations, the scheduling manager operations, and representatives of the unions; (2) the *project team*, composed of a project manager, a sub-project leader from the users, several planners serving as testers (partially in the project and partially devoted to their scheduling job), an IT-technical sub-project leader and test administrator.

Stability has been high in the project team, but not in the NSB organization. Top and mid management changed many times during the project. This has been a challenge for the project team. When the initial owners of the project disappeared, the idea of the system and the project had to be re-marketed and re-sold to new management.

Primary users. These are planners or dispatchers that use the system daily to perform their work. On the individual level, all variations in attitudes towards the system have surfaced. Some were very positive and oriented towards advantages and possibilities. Others were reluctant and eager to prove that the system could not be used. During the first training sessions, users and project team reached a common understanding that the system would become a useful tool after correction of some errors and some enhancements.

The use of the system brought up a number of issues that really had to do with conditions around the scheduling task. It turned out that 80% of complaints from long-term planers had to do with the input of timetable and rolling stock data or how the planning process was organized in one way or another. The system for rolling stock scheduling was not ready in the first phase of the TPO project. Data was converted from a very primitive spreadsheet and often came too late or was incomplete. The new rolling stock system was a significant improvement, but it turned out that key concepts had different meaning within personnel- and rolling stock scheduling. An enhancement to the interface was constructed to work around this problem. But a closer integration between rolling stock and personnel planning systems will be needed in the future.

Use of the system also brought up the need for consistent interpretation of labor rules. While discussions tended to be around boundary conditions, the users accepted the system as a standard tool.

Secondary users. These include mid managers with responsibility for production of plans and administrative staff. Secondary users do not work with the system but are users of statistics and reports. Some participated in user training to understand the possibilities and limitations of the system. But generally secondary users do not have full understanding neither of the work that planners do nor of the system. Unrealistic demands often came from these users. For example, it is crucial to understand that reports can only be extracted from schedules that have been created in the system. If the schedules made by the planners are not 100% complete, this will be reflected in the statistics. The efforts needed to create all the plans first were often underestimated and it was necessary to communicate actively with secondary users in order to prevent misunderstandings on this point.

Recipients of output. These are the drivers and guards or their representatives (the unions) that receive the schedules, normally in the form of printouts. Drivers and guards are very concerned about the printouts of duties and rosters. The duty diagrams used previously were drawn by hand. The quality depended among other things on the handwriting of the planner. Calculations were done manually (with handheld calculator) and there were many errors. Thus, the potential for improvement was obvious.

Complaints about the new printouts had to do with understanding the format and conventions. Discussions have been around minor graphical details that were corrected quickly. The new printouts have a much higher level of detail and are now fully accepted.

Summary of experiences from implementation

Taking a new system into use is a long-term investment and the key to successful implementation is commitment from management. The users will not experience advantages from day one. There is a threshold to overcome before the advantages become apparent. Endurance is required in the start up phase. The implementation activities take time and must be done at the expense of other activities. The work loads for planners (doing scheduling and project work) can be high at times and the work situation can be very stressed. The project team does not have the authority to allocate tasks to the planners. Line management must take the decisions. Therefore the project team cannot take full responsibility for implementation activities.

The advantage of a system that integrates many steps in the work process is not always obvious within one unit when seen separately. Use of the system can create benefits in the next link of the chain. Commitment, support and perseverance is needed on the level with responsibility for all links in the chain. The Short-term system is currently competing with other applications that are tailor made for handling parts of the process within the existing organizational units.

But no alternative exists that can integrate the whole work process the way TPO does. In NSB further adaptation of working methods is needed in order to harvest potential benefits.

Maintenance

The development contract was based on a Norwegian standard government contract made for procurement of one application. It was assumed that maintenance would be initiated after approval of the whole system. The TPO project turned out to be different. Several applications were delivered, tested and implemented successively. The maintenance contract was initiated for some applications before everything had been delivered.

The relationship between acceptance testing and approval of parts and whole was not explicit in the contract. Agreement about how to handle this was found during the project. Running implementation in production and continued systems development and testing in parallel caused considerable strain on resources both on SISCOG and on NSB in periods.

TPO is maintained both by NSB and SISCOG. NSB maintains the data and make changes to the underlying information. SISCOG provides changes to the system resulting from users requests for additional functionality.

Future Directions

NSB uses the Short-term Scheduler both for the short-term scheduling and dispatching phases (see section on Problem Description). While performing dispatching the daily events come from different sources, mostly by telephone, and are incorporated in the current schedule using the manual mode. As part of other contracts, both with S-tog and DSB, SISCOG is currently in the process of developing an independent dispatching component that receives real-time events, updates the schedule, computes their consequences, and provides decision-support in order to handle unexpected situations. Communication functions with crew members (via internet and SMS) are under development.

Acknowledgements

SISCOG would like to acknowledge the important role that our railway customers have played in the development of CREWS: NS (The Netherlands), WAGN (UK), CP (Portugal), S-tog (Denmark), DSB (Denmark), and VR (Finland). SISCOG thanks all its staff for their work and dedication. Special thanks go to António Frazão and João Paulo Varandas (TPO project leader) for all the effort, dedication, and night and weekends hours spent in order to make TPO a success. NSB would like to acknowledge the efforts, energy and relentless dedication to success contributed by Jan Oskar Kalland, the user representative and functional leader within the TPO project team in NSB.

References

Bisiani, R. 1987. Beam Search: *Encyclopedia of Artificial Intelligence*, S.C. Shapiro (ed.): 56-58, New York: John Wiley and Sons.

Doyle J. 1979. A Truth Maintenance System. *Artificial Intelligence* 12(3): 231-272.

Fox M.S. 1987. *Constraint-Directed Search: A Case Study of Job-Shop Scheduling*, San Francisco, CA: Morgan Kaufmann.

Garey M.R.; and Johnson D.S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, CA: W.H. Freeman.

Martins J.P.; and Shapiro S.C. 1988. A Model for Belief Revision, *Artificial Intelligence* 35(1): 25-79.

Morgado E.M. and Martins J.P., 1998. CREWS_NS: Scheduling Train Crew in the Netherlands. *AI Magazine 19* (1): 25-38.

Smith S.F. 1989. The OPIS Framework for Modeling Manufacturing Systems, Technical Report TR-CMU-RI-89-10, The Robotics Institute, Carnegie Mellon University.