TACT
An Integrated Transportation Problem Solved with CHIP

H. Simonis
P. Charlier
P. Kay

COSYTEC SA
4, rue Jean Rostand
F-91893 Orsay Cedex
France

1 Abstract

In this paper we describe a decision support system for an integrated transport problem in the food industry. The system was developed by COSYTEC using the constraint programming tool CHIP. The TACT program covers planning, scheduling and resource assignment aspects of operational fleet management. The different problem solvers are integrated in an interactive user interface, which allows the planner to guide and control the creation of a schedule. We present the use of different global constraints inside the problem solvers, which allow us to efficiently handle the many conflicting constraints. Finally we discuss the main benefits gained by the use of constraint technology inside this project.

2 Introduction

Constraint programming is a method of solving highly combinatorial problems given a declarative description of the problem and a general constraint propagation engine. This technology, developed over the last ten years, has been very successfully used to solve production scheduling and resource assignment problems [Wal95] [Sim95] [Sim96]. In this paper we describe the use of constraints to solve a quite complex transportation, planning, scheduling and assignment problem which integrates the use of multiple types of resources. This application, called TACT, was developed by COSYTEC for a customer in the food industry. It uses the CHIP constraint solving system [DVS88] [Sim95a].

The core part of the CHIP constraint solver, and essential for the application described here, is a set of global constraints, which allow to express complex conditions on sets of variables in a simple way. These constraints use rather strong, semantic methods derived from finite mathematics and Operations Research to check global conditions for solvability.

Three of these global constraints are especially important for the problem described the here:

The cumulative constraint [AB93] is used to express cumulative resource limits over time. These can take the form of disjunctive or cumulative machines or can be a form of consumer/producer constraint [SC95] which model the use of consumable resources.

The diffn constraint [BC94] expresses non-overlapping constraints of n-dimensional rectangles. It is used here mainly to express assignment conditions, stating that tasks assigned to resources do not overlap.
The cycle constraint [BC94] finds circuits in directed graphs. This constraint is central to the application, as it controls location continuity for resources like drivers and lorries. It also handles time constraints between consecutive tasks for the same resource and work rules and time limits on the overall amount of work done by a resource.
CHIP is available in different product forms [Sim95a], for example as C and C++ libraries. In this application, we use the Prolog based version of CHIP, extended with an object layer CHIP++ and modules for graphical user interfaces and data base connections.

3 Related Work

Compared to the number of constraint applications for scheduling, the number of systems handling transportation problems with constraints is quite limited.
The Pilot system [BKC94] [COS95] handles the day-to-day re-planning for pilots and cabin crew of short and medium haul flights of SAS. It tries to cover "open flights", flights for which resources are missing, by re-organising the existing schedule.
The DAYSY Esprit project, a joint development by Lufthansa, Sema Group, the University of Patras and COSYTEC, builds a day to day management system for Lufthansa.
A system developed by PrologIA for the crew scheduling of Air Littoral [GC95] re-implements a Simplex based set covering system with a special 0/1 Simplex constraint solver.
Another Esprit project, SuperBus, (PrologIA, Brunel Univ.) [Vet96] aims at crew scheduling for public transport systems.
The EVA application [COS94], developed by GIST for EDF, the French electricity utility, schedules the transport of nuclear waste between the power stations and the re-processing site in France. The program is used to optimize the use of the special transport containers and to minimize reactor downtimes due to transport delays.
EBI in Turkey has developed a program to schedule lorry transport [COS95] between warehouses and customers, which combines lorry capacity planning and route optimisation.
All these applications only look at some aspect of the transportation problem, i.e. pilots or aircraft, but not at all resources which are required. TACT must handle the complete problem in order to find a feasible solution.

4 Problem

The problem solved can be described in a simplified form as follows: In the food industry, birds (chicken and turkeys) are grown on a large number of farms. When the birds reach a certain age, they are transported to the factories, where they are killed and processed into different types of products. For each day, an order book describes the different farms to be visited, the number of birds of different types to be collected and the order in which the birds must be delivered at the factories. The arrival of the lorries at the factories must be scheduled carefully, in order to avoid either running the factory out of stock or to deliver the birds too early, which would be unacceptable for quality control reasons.

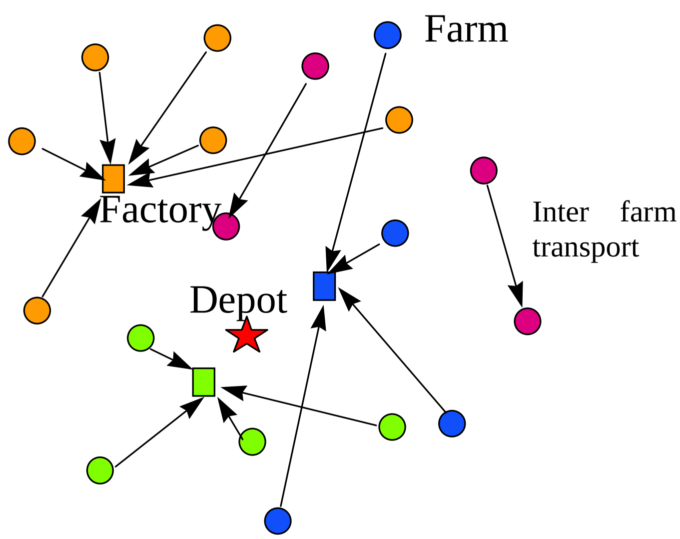

Figure 1: Order book – Required transport

In addition, other transport must be scheduled between breed and grow-out farms to distribute the next generation of birds at the correct time.
The order book is usually known well in advance, but often changes at the last minute in order to react quickly to some new event. Reaction time is crucial, since the affected personnel must be notified in time of any changes in the plan.
The company uses its own fleet of vehicles for the transport, but sometimes must hire in extra drivers or vehicles to handle the work load. The farms are at a distance of between 5 minutes and 3 hours driving time from the factories, so that geographical locations play an important role in the scheduling. Some farms can only be reached by small lorries, since they are at the end of difficult roads. For a number of farms, environmental constraints limit the time when lorries can drive to the farms.
The actual transportation process requires a number of interdependent resources:

- bird catching teams
- drivers
- lorries
- trailers
- fork lifts and fork lift trailers

We will now describe some of the constraints attached to the different types of resources.

4.1 Catching teams

The catching team is a group of persons which on the farms catches the birds and collects them into crates. The rate of work, their working time and their preferred work (catching turkeys or chicken, male or female) varies for each team. The teams start work at the depot, drive to a farm with a mini-bus, collect a number of birds there and either continue to another farm or return to the depot. For sanitary reasons, it may be necessary to return to the depot to change before going to another farm. The total working time and the number of crates collected by day are limited. Between two working days, the teams must have a certain rest time. This means that a team which works late on Monday can not be used very early on

Tuesday. If the work load can not be handled by the given number of teams, it is possible to hire in outside help at a significantly higher cost.

4.2 Drivers

The drivers will always clock on at the depot, collect a lorry and make some trips to farms and from there to the processing sites. At the end of the day, the driver must drop the lorry at the depot and clock off there. In a normal day, they will make between two and five trips (depending on distances, etc). After delivery at a factory, the lorry is cleaned and can be used for another trip. The driver may change lorries several times during a day, he is not assigned full-time to one particular vehicle. All drivers can drive all lorries in all possible configurations. The drivers do not have fixed clock-on and clock-off times. They follow a roster, which for each week gives the sequence in which drivers must clock-on. The usual working rules apply, i.e. maximal number of hours working, maximal number of hours consecutive driving, maximal mileage per day are all limited. While a lorry is loaded on a farm, the driver can take the legally required rest period. At the end of a trip, the driver may be required to change the lorry configuration, i.e. drop a trailer or change the large crates for turkey with smaller crates for chicken. This may require an empty trip to another factory in order to pick up these new modules.

4.3 Lorries

The factory fleet consists of different types of vehicles, some single lorries of 24 ton weight, to lorries with trailers with a total weight of 32, 35, 38 or 42 ton total weight plus some articulated trailers which can be used with outside haulage. Not all lorries are available at all times, there are times reserved for maintenance or inspections. The birds are transported on the lorries in crates which are placed in modules. These modules are loaded and unloaded with fork lifts.

The number of lorries available is given. If for some reason more lorries are required, they can be rented, but at a significant extra cost.

4.4 Trailers

The trailers are not always linked to a particular lorry, the can be coupled and uncoupled at the different factory sites and at the depot. It is sometimes necessary to deliver a trailer to another factory site, uncouple it there and then continue the trip with the lorry alone, while the trailer is picked up by another driver/lorry.

4.5 Fork lifts

The loading operation at the farm requires a forklift for moving the modules from and to the lorry. This forklift must be brought on a special trailer from the depot to the farm by the first lorry to arrive and should be returned to the depot at the end of the day. This means that the first and last lorry can not take a trailer to the farm. In certain situations, a fork lift can stay at a farm over night if it will be used in the next morning. The fork lifts are a scarce resource, i.e. their use must be carefully controlled.

5 Previous solution

Before the introduction of the TACT system, the scheduling of the transport problem was done by hand. The scheduler started in the morning using the current order book for the next day. Usually, it took about 6-8 hours to find a solution. If the order book was changed in a significant way, much of the work had to be redone, delaying the publishing of the schedule. The two experts in the company could produce a very good schedule in most situations, but it was nearly impossible for an outsider to understand the process. Often, some rule would be bent in order to find a solution more quickly or to patch a solution after some change in the input data. As a result, it would be often unclear which constraints were actually handled correctly in a given schedule.

6 Solution Approach

Given the complexity of the transportation problem, it is understandable that only a decomposition can lead to manageable sub-problems of feasible size. In the manual solution, the scheduler were cutting the problem into three sub parts. In TACT, we follow the same approach.

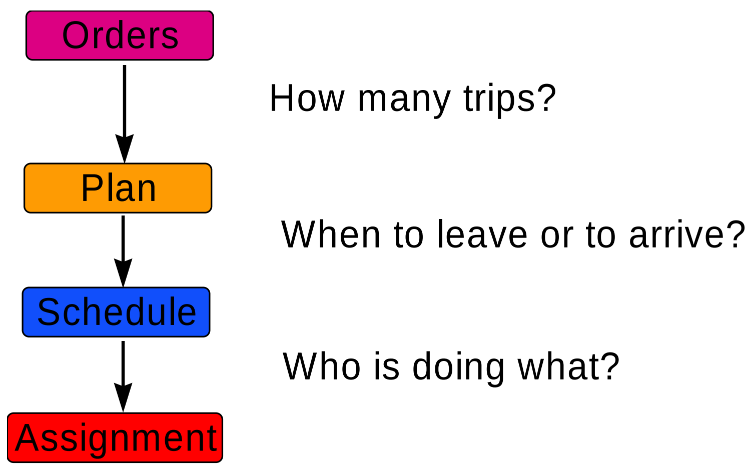

Figure 2: Solution Approach

In a first step, called trip cut, the number of trips and the different lorry types for each entry in the order book is determined. This creates a set of trips, each transporting a given number of birds on a particular type of vehicle.

The second step, called line balancing, is to schedule the trips generated in the first steps in such a way that the supply for the factories is guaranteed and that operational constraints of the loading and transport are taken into account.

The third step consists in assigning individual resources to all activities. Depending on the type of resource, the timing may have to be re-evaluated at the same time.

After each step, the results are presented to the user in the graphical user interface, where he can react to the system. Quite often, the user may decide to re-run the trip cut (first step) with some changes in the parameters before continuing with the second phase.

Each of these phases are run for the orders of a single day, although the generated schedules (30 hours) overlap in time. For each day, the schedule of the previous day is considered as fixed work in progress. Only the fork lift planning is made for one week at a time, in order to minimize the transport required.

We will now discuss some important aspects of the different problem solvers. For space reasons, we can not present the complete model in each case.

6.1 Trip Cut

The trip cut is done by a program which solves the problem for each farm independently. Input data required are the details of the farm, such as which lorry type can reach the location,

the number and weight of the birds, but also the expected temperature on the day of transport. Based on the temperature, and the weight, a table limits how many birds can be packed into each crate. Another constraint is the free capacity of the lorry, in order to respect the total weight limit of each vehicle. The need to transport a forklift at the beginning and at the end of the work on the farm imposes constraints on the number of single lorries to be used.
All these constraints are fed into a small integer programming model, which produces the best mix of vehicles to transport the birds. Some special rules are used to avoid cases where the last lorry is used to transport a very small number of remaining birds, while all other trips are running at capacity. It is better to require a certain minimum load for all lorries. From the number of birds we deduce the loading time for each lorry, i.e. the duration of the stay at the farm.

6.2 Line Balance

The line balance takes the trips generated in the first step and schedules them in time. Important constraints there arise on the farm and on the factory side of the transport. At the farm, all loading operations must be scheduled consecutively, i.e. once the loading has started, all further loading must be scheduled without break until all birds have been loaded. This constraint is required to minimize stress on the birds.

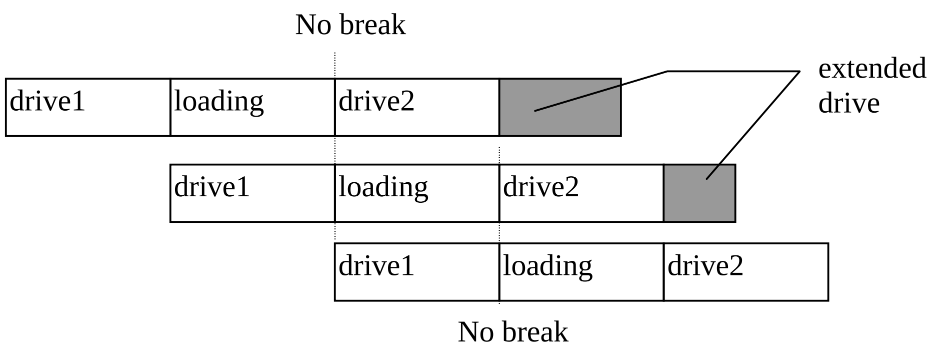

Figure 3: Constraints on farm

At the factory side, the birds are required at a given rate, which changes slightly with time of day and bird type. A certain amount of stock for the factory is required, this corresponds roughly to one lorry load. The birds should not be delivered too early, again to reduce the stress level and since storage capacity at the factory is limited. Unfortunately, the loading rate at the farm (depending on the team) is much smaller than the unloading rate at the factory. In order to balance the schedule, two solutions exist.

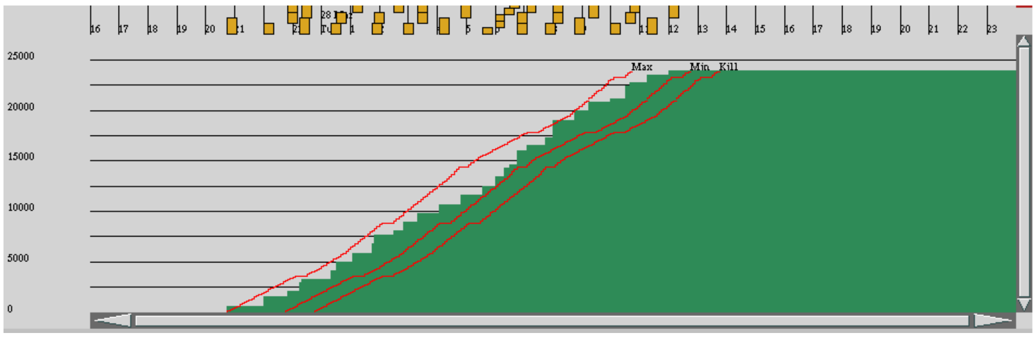

Figure 4: Producer /consumer constraint for factory intake

One can overlap transport of birds from different farms, so that deliveries are interleaved in time. Unfortunately, this is only possible in a limited way, as the factory requires the birds in the order given in the order book.

The other alternative is to change the transport time for loads from the same farm by adding extra driving time so that the load arrives later (extended drive). This can help to match arrival times to the demand, but creates extra cost for drivers and other resources. The objective is to find a schedule which delivers loads just in time, while minimizing extended driving time and perturbations of the order book.

The solver for the line balancing is implemented using producer/consumer constraints [SC95] for the different factories. The constraint assures a minimal stock level and at the same time a maximal stock level given by the storage space in the factory.

The solver also uses a diffn constraint to limit the number of activities that can be scheduled at a farm at the same time. In many places, only one lorry can be used for loading at any given time. At other places, multiple lorries can be loaded if they are at different ends of the farm.

Another constraint takes the number of fork lifts on a farm into account. A cumulative constraint limits the number of concurrent uses of fork lifts on a farm.

The initial sequence of trips in the order book strongly restricts the domain of the time variables in this solver. Constraint propagation of the global constraints further restrict domains, so that solutions are found quite quickly in one or two minutes.

If the problem is over-constrained, i.e. no solution can be found with the given constraints, the system automatically relaxes the limit on the maximal stock at the factory. It is then the responsibility of the user to change parameters or to re-arrange the trip-cut in order to find feasible solutions. This is quite a common situation, which is usually solved by discussing changes with the factory schedulers.

6.3 Resource Assignment

The resource assignment uses the schedule generated in the line balancing and assign individual resources to all tasks. At the same time, it takes location continuity [Sim94] into account, i.e. makes sure that a lorry starts and ends trips at the right place.

This constraint is handled with the cycle constraint. An activity is represented as a node in a directed graph, with a link to another node if the activity can be followed by that activity. This constraint not only expresses location continuity, but also constraints on the temporal sequence and special conditions like hygiene rules prohibiting certain activities before others.

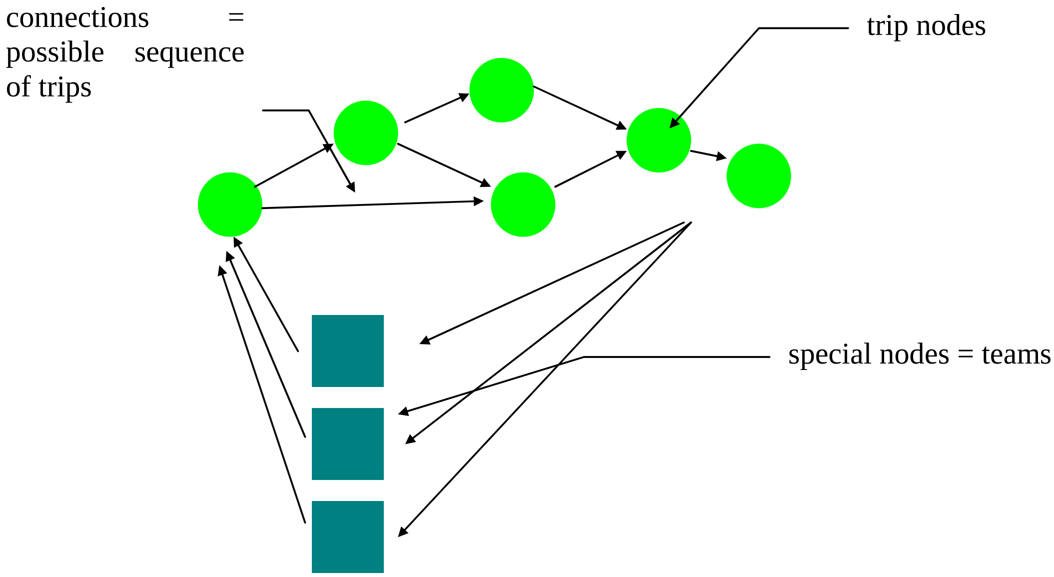

Figure 5: Graph of teams and activities

Resource assignment is also handled with a diffn constraint. This constraint assigns tasks to resources, making sure that no tasks overlap on the same resource.

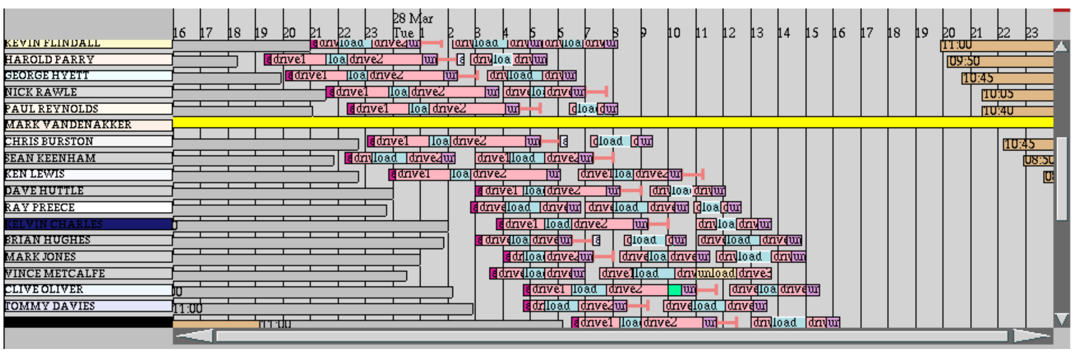

Figure 6: Assignment for lorry drivers

Two redundant cumulative constraints are also used for each solver. One controls the cumulative resource use over time, i.e. cumulates the resource consumption at each time point and restricts this by the overall number of available resources.
The other constraint implements a bin-packing redundant constraint as explained in [AB93]. This constraint projects resource use onto resources ignoring time.
For all resources, dummy activities are introduced to express work in progress (yesterday's unfinished schedule) or unavailability (rest periods, maintenance, training). These dummy tasks block an area in the diffn constraint or cumulative, so that no task can be assigned there.
The assignment strategy used is a multi-snake labeling. This means that for each node the successor/predecessors are assigned rather than a labeling on start or machine assignment. If some activity can not be covered by the resources already in use, a new cycle is started by

linking a special node to this activity. A graphical view of this assignment strategy is given in figure 7.

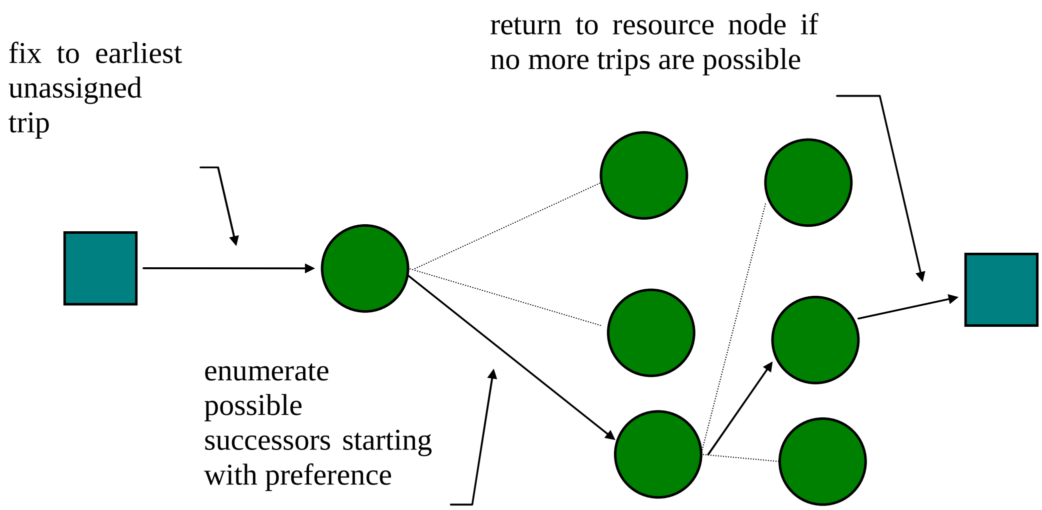

Figure 7: Assignment strategy

6.4 Handling over-constrained problems

An important aspect in the resource assignment is the handling of over-constrained problems. If the resource assignment doesn't find a solution in a reasonable time this can either mean that there is no solution or that the heuristic is not good enough to find it. If there is no solution, this can be because there are not enough drivers, or not enough lorries of some type or not enough catching teams. It is important for the user to find the real cause of the problem, in order to avoid costly mistakes like renting another lorry when more drivers for the existing lorries are required.

In order to provide some form of automatic explanation, the system automatically relaxes different types of constraints if no solution can be found.

Another specialized method is the calculation of lower bounds for some type of resource in independent solvers. This method covers the given amount of work with work profiles of a given duration, ignoring all other constraints. A detailed description together with the constraint model of this lower bound calculation is given in [Sim96a]. Note that the latest version of CHIP includes these lower bound calculation as options for the global constraints cumulative, diffn and cycle.

7 System Architecture

In this section we describe some architectural aspects of the system.

7.1 Overview

The overall system architecture is shown below. The system is running on a UNIX workstation which also runs the ORACLE database. This database holds all persistent information of the data. The order book is transferred from an AS/400 system, which holds the commercial system. The results of the schedule are uploaded to the data-base, reports and statistics are obtained using database tools.

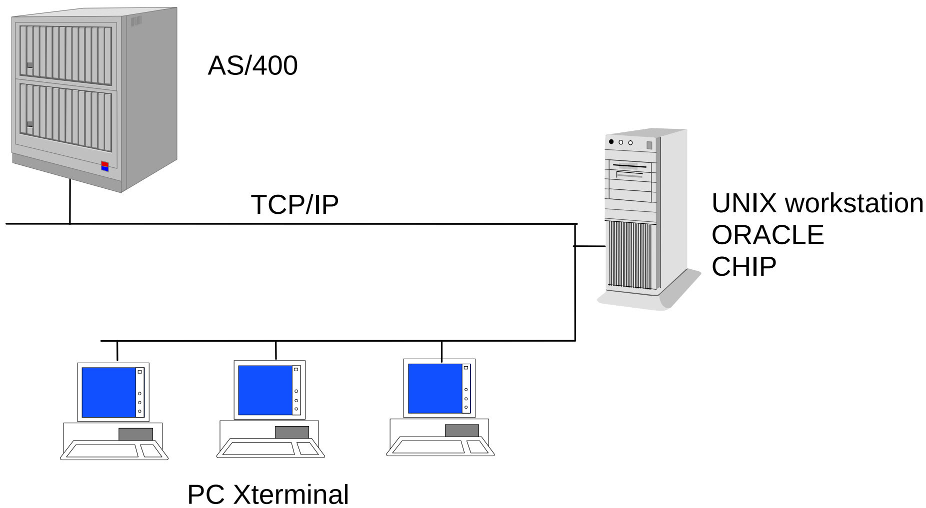

Figure 8: Overall system architecture

The scheduler is connected to the application via a PC, which runs the graphical interface with an XWindows server. The solver is run on the UNIX workstation. The different users can operate on different scenarios loaded from the same data base or can work independently on individual files. Normally, only one user is responsible for producing the operational schedule. Others may work on strategic issues or a planning scenario.

7.2 Data model

A common data model is used to describe the object model of the application, the data manipulation tools of the graphical user interface, and the database model for the SQL database. This method, described in more detail in [KS95], gives a single point of reference for changes of the data model. Files for the graphical user interface, the table creation, down- and upload are automatically generated from the description file.

7.3 System size

The complete TACT application comprises around 36000 lines of CHIP code, including roughly 7000 lines of problem solver code. This includes all data management, but excludes the AS/400 connection and database reports, which were developed by the customer.
The problem size handled by TACT is given by the following figures. The customer handles about 1 000 000 chicken and 100 000 turkeys per week, using around 30 lorries and 45 lorry drivers. A typical schedule will consist of roughly 500 tasks which are assigned to multiple resources.
The application was developed in a rather short time period of 10 month, largely due to the massive re-use of code possible within the CHIP framework.

7.4 User Interface

A graphical user interface is very important for a complex application like TACT. The user can inspect and change all underlying data of the application, from farm information to loading limits or the order book. Results of the different scheduling phases are shown in different graphical forms as lists, Gantt charts or stock level diagrams.

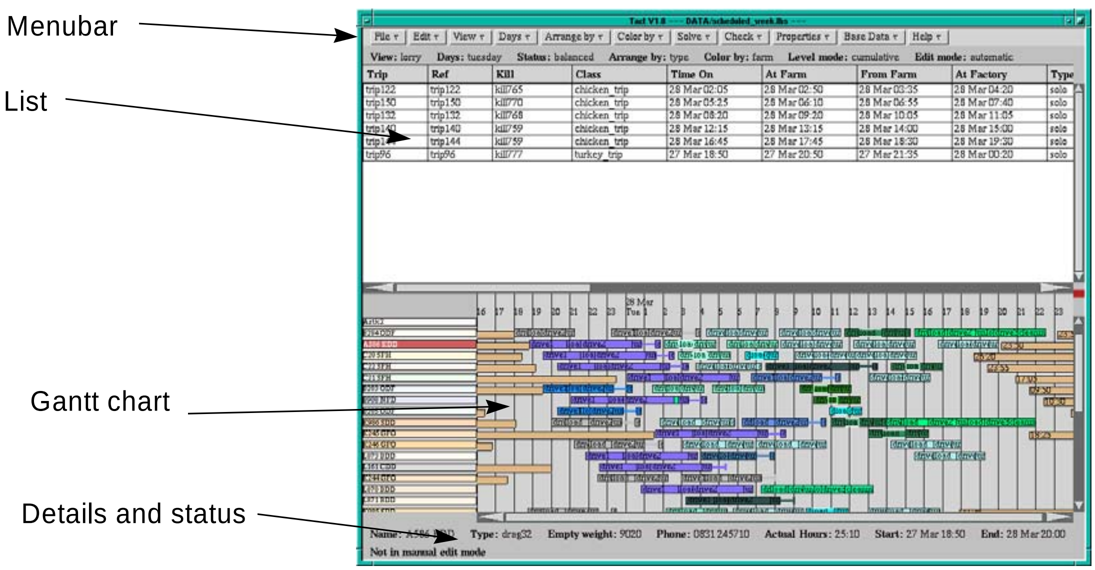

Figure 9: User Interface

The user can modify the schedule interactively and then check if the modifications satisfy the constraints. In some situations, the user may wish to violate a constraint in the system in order to solve a particular problem. The automated solver, for example, does not allocate over-time to catching teams. The scheduler, on the other hand, can decide to use over-time for some team and to discuss this with the affected persons., rather than calling in outside contractors for this work. This negociation process can not be automated, but is crucial to finding good compromises between different objectives.

8 Evaluation

The system has been operational since beginning of 1995 and has completely replaced the manual scheduling process. While a manual solution required up to 8 hours of work, it is now possible to find solutions in as little as 15 minutes (about 5 minutes of constraint solving). This gives more flexibility to the scheduler who can devote more time reacting to sudden changes in the order book or to compare different scenarios balancing contradictory objectives.

The TACT application is not used as a black-box problem solver, but rather as a decision support tool for the scheduler. It speeds up the generation of schedules and ensures that constraints and regulations are taken into account. The scheduler can control the different solvers by enabling/disabling constraints or selecting different heuristics.

It is difficult to compare resource utilization results before and after the introduction of the system as business in general has changed significantly. But it is clear that the tool allows to analyze a problem earlier and in much more detail so that expensive last minute changes are avoided.

A return on investment in 6 month has been quoted by the customer, in particular due to reduced capital expenditure on new lorries.

In three years of operation, a number of changes were introduced into the system to handle new constraints or restrictions. The company has updated its lorry fleet, so that constraints on capacity and farm access had to be changed. The contractual status of the catching teams changed, so that work-rules, time limits and overtime pay had to be revised. The loading constraints balancing full and partially empty vehicles was completely revised. All these changes could be implemented quite rapidly without affecting other parts of the constraint model.

9 Conclusion

In this paper we have presented the TACT system, an integrated transportation problem solved with CHIP. The system covers the complete solution process from the initial decisions on using the different lorry types over the scheduling of arrivals at the factories to the resource assignment for individual lorries, drivers and catching teams. The overall system mimics the previous manual approach to the problem, but speeds up the scheduling process significantly.

The system is operational now for three years. In this time it has completely replaced the previous scheduling method and has shown its flexibility handling a significant number of changes in the business process.

10 Bibliography

[AB93] A. Aggoun, N. Beldiceanu
Extending CHIP in Order to Solve Complex Scheduling Problems
Journal of Mathematical and Computer Modelling, Vol. 17, No. 7, pages 57-73
Pergamon Press, 1993

[BKC94] G. Baues, P. Kay, P. Charlier
Constraint Based Resource Allocation for Airline Crew Management
ATTIS 94, Paris, April 1994

[BC94] N. Beldiceanu, E. Contejean
Introducing Global Constraints in CHIP
Journal of Mathematical and Computer Modelling, Vol 20, No 12, pp 97-123, 1994

[COS94] COSYTEC
CHIP User's Group Meeting
November 1994

[COS95] COSYTEC
CHIP User's Group Meeting
November 1995

[DVS88] M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun, T. Graf and F. Berthier.
The Constraint Logic Programming Language CHIP.
In Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88), pages 693-702, Tokyo, 1988.

[GC95] N. Guerinik, N. Van Caneghem
Solving Crew Scheduling Problems by Constraint Programming
Proc CP 95, Cassis, France, Sept 1995

[KS95] P. Kay, H. Simonis
Building Industrial CHIP Applications from Reusable Software Components
PAP95, Paris, April 1995

[Sim94] H. Simonis
The Use of Exclusion Constraints to Handle Location Continuity Conditions
COSYTEC Technical Report, November 1994

[SC95] H. Simonis, T. Cornelissens
Modelling Producer/Consumer Constraints
Proc. Principles and Practice of Constraint Programming, Cassis, France, September 1995

[Sim95] H. Simonis
Application Development with the CHIP System
Proc Contessa Workshop, Friedrichshafen, Germany, September 1995, Springer LNCS

[Sim95a] H. Simonis
The CHIP System and its Applications
Proc. Principles and Practice of Constraint Programming, Cassis, France, September 1995

[Sim96] H. Simonis
A Problem Classification Scheme for Finite Domain Constraint Solving
Proc workshop on constraint applications, CP96,Boston, August 1996

[Sim96a] H. Simonis
Calculating Lower Bounds on a Resource Scheduling Problem
Proc workshop on constraint programming, ASIAN96, Singapore, December 1996

[Vet96] E. Vetillard
Applications de la programmation logique avec contraintes aux problemes de transport
JFPLC96, Clermont-Ferrand, 5-7 June, 1996, Hermes

[Wal95] M. Wallace
Survey: Practical Applications of Constraint Programming
Technical Report, IC Parc, September 1995