

SYNTHESIS REPORT

FOR PUBLICATION

CONTRACT N°: BREU-CT92-0567

PROJECT N°: **BE** 4140

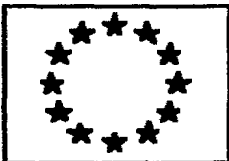
TITLE: **Dynamic Scheduling Toolbox**
ScheduleZ

PROJECT
COORDINATOR: **MJC²**

PARTNERS: **MJC²**
KP Foods
Politecnico di Torino (PoliTo)
Syco
Uninova
COMPAL

STARTING DATE: **1/09/92**

DURATION: **36 MONTHS**



**PROJECT FUNDED BY THE EUROPEAN
COMMUNITY UNDER THE BRITE/EURAM
PROGRAMME**

DATE 31/10/95

ABSTRACT

The goal of **ScheduleZ** was to create an object oriented toolset for the configuration of highly efficient dynamic scheduling systems for manufacturers and distributors of short shelf-life and fast turnover products. This goal was fully achieved and the results highly innovative. The approach and results are discussed in this document.

INTRODUCTION

ScheduleZ's overall objective was the research and development of a tool box for the creation of dynamic planners/schedulers targeted at manufacturers and distributors of short shelf-life and fast turnover products.

These manufacturers are faced with problems of decreasing lead times, smaller batch sizes, increasing product variety, and little opportunity to manufacture to stock. Consequently, efficient utilisation of resources (both capital and labour) is becoming evermore difficult.

Current product offerings in the market are too generic. They require the user to map his production onto a basic and restrictive set of functions which take no account of the particular manufacturing problems encountered in his market segment. The alternative, which is generally too costly to contemplate, is for the manufacturer to commission a totally bespoke system.

ScheduleZ created a tool box for these industries, that can be used to model manufacturing strategies and configure production planning/scheduling systems which explicitly map onto the real world production and supply chain problems encountered.

Specific objectives achieved were to research and develop

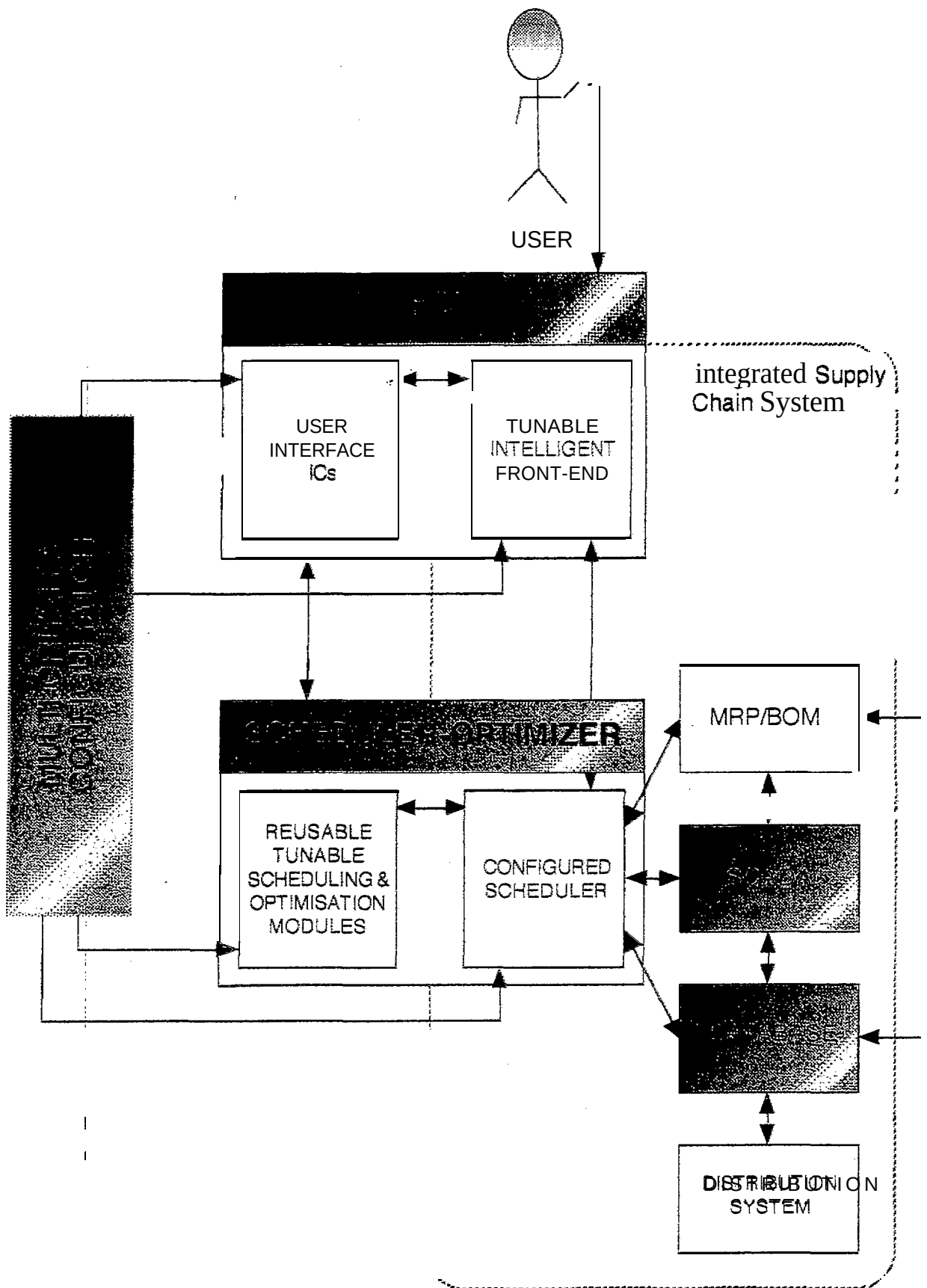
1. A multicriteria selection system which provides a high level graphical user-driven configuration system;
2. Graphical OOD libraries;
3. Finite capacity scheduling blueprints;
4. Tools for cooperative interaction with SCADA systems;
5. Ways of integrating with other supply chain processes.

The system makes use of innovative OOD reusable code libraries and frameworks. **ScheduleZ** was researched and tested in a live manufacturing environment and in a simulation background. A powerful software simulation environment was built to test

the integration of the scheduler with data from the shopfloor via SCADA. The approach was very successful.

The architecture of **ScheduleZ** can be seen in the following diagram.

The Consortium partners represented a vertically integrated team, including two major end-users (COMPAL and' KP Foods), system houses (MJC² and SYCO) and research establishments (UNINOVA and PoliTo). The team worked as an integrated unit and the results exceeded all expectations.



ScheduleZ

TECHNICAL DESCRIPTION AND RESULTS

Scheduling Tools

The team researched several approaches to scheduling and found out that a balance of standard mathematical algorithms and AI produce the best results. Two demonstrators were implemented corresponding to the **COMPAL** factory (of fruit juices and tomato pulps) and a **KP-Foods** factory (of potato and cereal snacks). For the **COMPAL** demonstrator **G++** (an extension of **C++**) was used while the **KP** demonstrator integrates **Prolog**, **C++** and finally at the **SCADA** level **G++**.

The **COMPAL** demonstrator was used to test the power of a **two-step** approach. Step one looks at the **annual** planning of the factory and was mainly solved through mathematical programming (OR techniques). Concerning step two (the short term planning for the next month) a greedy procedure followed by a local search has been applied improving the results both in term of CPU time and solution quality.

In the first prototype of the **COMPAL** demonstrator the scheduler for the short term planning problem was solved through a Branch and Bound package. Due to the combinatorial structure of the problem and consequently the large amount of time necessary to find an optimal solution the package used to be stopped **after** having visited a given number of nodes of the search-tree even if **optimality** had not been proved yet. Given that the obtained solution was sometimes not so close to the optimum and willing mostly to reduce the computation time, a new heuristic approach was developed. The experimental results of the research show that the new heuristics is extremely appropriated to this case and is able to find a good solution in a much more reduced **time**. The proposed heuristics decomposes the problem into three correlated phases to be solved consequently:

- search of a feasible solution by means of two greedy procedures
- improvement of the first solution applying a local search to each line independently

- iterative search of the final solution applying a local search line but taking also' into account the other lines.

The search of a first feasible solution is carried out by two greedy procedures. If the first greedy procedure is not able to find an optimal solution, it transfers the control **to** the second greedy procedure. If the second greedy procedure can not find a feasible solution either, the program finishes with an output message "non feasible" and the input data must be changed. To improve the first solution a TABU search is performed on each production line disregarding the other lines. This can be done because the neighbourhood considered in this TABU search is such that the numbers of shifts in each day will always be lower or equal to the number of shifts obtained in the first feasible solution. Thus, the constraint of the daily persomel available will be fulfilled for the whole search. The iterative search of the final solution makes on each line a TABU search equal to the previous one but without the limitation of not exceeding the **number** of shifts obtained in the first feasible solution. As a consequence, we must take into account the other lines in order to **fulfil** the constraint of the daily **personnel** available.

In the **KP** demonstrator factory staffing was considered to be a key **functionality** that the **ScheduleZ** system must implement. The first step was to develop a model for the factory labour requirements which strikes a balance between generality and practicability. This model was thought by **KP** staff **to** be expressive enough to provide a **labour** model of the **KP** factory and yet is general enough to be of use in many factories in the fast moving consumer goods manufacturing area. The philosophy has been to identify relationships between the staffing requirements of a factory and the production and machine usage within that factory. Four relationships have been chosen which span the **entire** staffing requirements of the factory, yet which are largely independent of one another. In this way, these four components can be used to build up an accurate model of the factory staffing by simply **summing** the contribution of each component.

Work has been undertaken to. allow for a completely configurable shift structure. In the first prototype of the **ScheduleZ** system, the shift structure was "hard coded" into several software modules, including:

- the main **Gantt** chart graphical display module;

- the report generating module;
- the scheduling and optimisation modules;

The new configurable shift structure allows the number of shifts per day, and the start time of each shift to be defined through the **Multicriteria Configurator** interface. The new structure was necessary as the shift pattern of a factory maybe subject to change as new **working practices** are introduced. In addition the work done in **providing** a configurable shift structure has brought major advantages to the **ScheduleZ** system.

Several of the data structures internal to the Core Scheduler are defined in relation to the shift configuration. An example of such a data structure is the “Stock Profile”. This records shift by shift how the **stock** levels for each of the factory’s products varies throughout the scheduling **period**, taking into account starting stock levels, sales and production. Its main purpose is to provide the Core **Scheduler** with a means of ascertaining the impact of a schedule on stock level as it searches for the most effective production plan. An auxiliary use of this data structure is to provide a stock profile corresponding to the chosen, optimised schedule. This is exported to the Front End and Report **Generating** modules of the **ScheduleZ** system to provide stock **information** to the user of the system. The algorithms and modules of the Core **Scheduler** employing such data structures have been revamped to take into account the variability in the number of shifts per day, the shift start times and shift durations. These modifications represent a significant increase in the generality and reusability of the modules of the Core Scheduler.

The method by which orders are requested from the system has been fundamentally changed. Previously a “GO-RATE” for each product for each week was given to the core scheduler, comprising the required total production of the given product for the given week. This corresponded to the working practice current at the Ashby factory at the time of the 12 month report. A new system has later introduced in which an initial stock (in cases) for each product is given together with expected daily **loadouts** from the factory and required minimum and maximum stock coverage. The new system has the advantages of generality, this system of specifying load-out is common in the area of fast moving consumer goods and of conforming with KP’s new method of specifying **orders**.

The new method of specifying orders has led to major changes to the core scheduling system, instead of planning to meet a specific order for each product for each week with a due date we plan to produce so as to keep stocks at all times within certain boundaries.

There is a need to allow the user of the ScheduleZ system greater control of the system, allowing existing scheduling patterns/practices to be retained in the schedules generated by ScheduleZ. In this way the transition between manual and automated scheduling systems will be smooth. In addition, it must be possible to gradually reduce the influence that these practices have on the system as time goes by, thus allowing a gradual evolution of working practices as the more efficient schedules of the ScheduleZ system are phased in. A system of "Schedule Frameworks" has been developed to cope with the situation described above. This was developed in the context of the KP Ashby demonstrator system and the COMPAL demonstrator, but the techniques and ideas presented apply generally to savoury snack production, and to many factories consisting of processing and packing stages. Clearly, the introduction of a Scheduling Framework System imposes new requirements on the Core Scheduler: in effect, a Scheduling Framework is an extra scheduling constraint. The Core Scheduler must use the sequencing provided by the Scheduling Framework wherever possible, and yet avoid producing bad schedules due to an inappropriate Scheduling Framework. Modifications to the Core Scheduler modules have been made to allow for the integration of the Scheduling Frameworks System.

Development of a generic optimizing algorithm based on a modified constraint propagation and branch and bound techniques brought good results to the KP scheduler. This algorithm is intended to take the feasible schedules produced by the Core Scheduler and improve changeovers and CSR (Customer Service Ratio a measure of adherence to delivery times). It has been decided to follow an approach based on constraint propagation and branch and bound techniques. Within the constraints set up, the optimizing criteria of this algorithm must be those defined by the Multicriteria Configurator. Constraints based on timings (that is speed of bagging, earliest possible start of production, quantity to be produced to fulfil stock holding requirements) and targets based on changeovers as prioritised in the schedule parameters editor were the first implemented.

Constraints based on unavailability (excluding a time period for production on a resource) and preassignment (fixing production for a time period on a resource) were developed later, as is the ability to resequence products of different pull and labour usage (when throughput and labour constraints become important).

Multicriteria Configuration

Multicriteria was implemented maximising the use of HCI. The user of a scheduler will no longer have to learn a command language or read a system manual.

Graphical Configuration and Preferences Objects were created to make the scheduler easily tunable. A new layer of data abstraction has been introduced to the Prolog database interface (on the COMPAL scheduler) to improve speed of implementation and the reusability and reliability of the system code. Two new graphical interface objects which use this data abstraction layer to extend the functionality of the editor matrices have been designed and implemented. In addition a tool to speed up the user's entry of information relating to colour has been implemented.

To briefly summarise, a user of the ScheduleZ system can enter or modify data pertaining to:

- scheduling parameters;
- factory machinery configuration;
- manufacturing constraints; and
- product information.

The data displayed in an editor matrix is stored as a sequence of Prolog facts in the on-line Prolog database. This information is available to the Core Scheduler via the Prolog database interface.

Two interface objects, namely, the "CONFIGURATION INTERFACE OBJECT and the "PREFERENCES INTERFACE OBJECT" are graphical elements which bring benefits to both the users and developers of the ScheduleZ system

- (1) Benefits to the system developers the Configuration Object allows new editor matrices to be implemented and modified more rapidly than was previously possible. The design of the Configuration Object was accompanied by an enhancement to

the interface of the Prolog database. These two developments have provided a high level of abstraction to the system developer which not only increases development speed **but** also greatly reduces the the likelihood of introducing coding bugs when an editor matrix is modified. This is particularly important where rapid **prototyping** techniques are used: evolutionary designs require that the system specification undergoes frequent appraisals and updates.

- (2) Benefits to the application users: the Preferences Object provides a means **of** modifying key visual attributes of an editor matrix, allowing the user to tailor each matrix to his or her own preferences. Moreover, the Preferences Object has been designed with a multi-user environment in mind, allowing for independent control over these attributes for each user of the system.
- (3) The third new interface object is the "COLOUR CHOOSER INTERFACE OBJECT" which is intended to speed the entry by users of data of type **colour**.

A Table Interface Component (**IC**) has been developed written using the Open Software Foundation object oriented Motif widget, set which forms the core of the common desktop environment (**CDE**) shortly to be adopted by all major suppliers of UNIX workstations. The development of the Table IC allowed the implementation of the entire graphical interface module library in Motif , thus bringing **all** the **ScheduleZ** partners to a common graphics standard(for Unix systems). The Table **IC** represents a major step in the changeover from the Open Look widget set.

Note however that because of the OOD approach the bulk of the work performed in implementing the Preferences Object, such as the design and implementation of program structures and files to store the visual state of the editor matrices, were reused by the Table **IC** with little or no modification.

The Scheduling Criteria Editor is a new component of the **Multicriteria Configurator** which has been successfully integrated into the 'demonstrator system at **KP's** Ashby site. A new button has been added to the main graphical **interface** of the demonstrator from which the Scheduling Criteria Editor can be

called up . All the usual functionality of an editor has been provided by way of file saves and loads, allowing the user to easily perform "What If?" experiments with different Scheduling Criteria settings. The existing scheduling and optimisation modules have also been interfaced successfully with the output of the Scheduling Criteria Editor, and extensions in the flexibility and generality of the capabilities of these modules have been made to accommodate the new scheduling criteria information available to them.

The MultiCriteria Configurator Interface consists of many Editors through which the user adjusts the parameters of the ScheduleZ system. In order to speed the access to these editors the Data Tree Object has been introduced. The Data Tree is a new graphical element of the Multicriteria Configurator Interface which provides a new style of access to the editors. It increases the speed at which an experienced user can locate a particular editor, but the main benefit has been in helping new users unfamiliar with the layout of the editors, who may have some difficulty in accessing the editors via the older menu-driven style of interface. The Data Tree has made access both easier and faster.

Finally, it should be noted that the original means for accessing the editors of the Multicriteria Configurator Interface (i.e. through a sequence of menus) is still available. This provides backward compatibility for existing users of the system.

The Object Oriented Approach and Tools

ScheduleZ not only developed powerful scheduling tools but also made use of innovative OOD reusable code libraries. In this field the achievements were well in excess of any expectations.

It is well known that the advantages, especially on reusability, offered by the OO technology can be achieved if a unitary computational model (based on a language of design patterns) and frameworks of reusable classes is adopted.

Framework based architectures, and framework construction have been, in fact, at the basis of all past ScheduleZ development by all the partners. In the first years of activity such frameworks were conceived and their prototypes used and developed by all partners with reference to distribution, persistency, graphical

user interface generation and SCADA. More recently, the need to generalise the scheduling architecture to quickly accommodate changes in the specifications has induced to modify the scheduling architecture introducing constraint programming.

In the last year of **ScheduleZ** some of the OO frameworks developed in the past were extensively reviewed, based on the experiences gained in the field testing and evaluation in both the COMPAL and KP-Food scheduling systems.

In the case of constraint programming the activity has resulted in the construction of a completely new framework. A lot of work has been in refining the OO classes which compose the architecture of the frameworks, in porting them to the different platforms and in preparing installation and user manuals.

A new distribution framework was adopted - **COLOMBO** - which allows the development of client-server OO applications. It uses in a transparent way the **TCP/IP** communication protocol to access locally or remotely located objects, these can be first class objects such the entire scheduler or they can simply be data accessed from a data-base.

The general characteristics of **Colombo** is to allow a seamless evolution from a local prototype of an application to a distributed implementation. In the last year of **ScheduleZ** a revision of the **Colombo** framework has been required to comply with the emerging standard **OMG-CORBA**.

In the **ScheduleZ** project the framework **Colombo** has been used to transform the COMPAL scheduling prototype, which originally was a stand-alone program, into a client/server application where the scheduler and its data-base reside on a remote server, accessed by multiple client user interfaces, through which multiple planners in the factory can interact on the same planning and scheduling problem.

The last version of **Colombo**, along with the architecture description, contains also the user manual, with the instructions for transforming applications from stand-alone to distribution.

Graphical Tools

According to basic rules on respecting human factors, two major user interface tools were designed with generality in mind and developed to help the operators plan and analyse the production for the Compal factory:

- The Strategic Planning Editor
- The Charts Module.

The Strategic Planning Editor is currently being used in the Compal demonstrator and consists of two parts:

- The Scheduler Forms
- The Plan Visualizer.

The Scheduler Forms give the user the ability to provide input to the scheduler. The layout of those tools tries to mimic the schedulers input file format. The output produced by the scheduler may then be visualised as the production plan in the Plan Visualizer.

Due to the generic OO approach other outputs such as stocks and accumulated productions may also be visualized using the Charts Module.

The Plan Visualizer is essential for the visualization of the production plan up to a full year. The user can inspect the plan up to a resolution of one hour. Two levels of detail were implemented to switch between time resolutions. There are also inspecting capabilities associated with production lots where one can see what is being produced in each production line or shift.

The Charts Module has been integrated with the work developed by MJC² for the ScheduleZ project. The best chart type for the data can be chosen by the user from the three types available

- Areas
- Bars
- Lines

Highly innovative shadowing techniques were applied to the graphics to enhance visual appearance and legibility. The area charts are the most sophisticated ones in order to allow the possibility of displaying

more, than one data set at a time. A special transparent-polygon rendering algorithm was designed and implemented for that. Thus one can see the entire stack of data series, even the parts that would eventually be obscured by the topmost series. The results are spectacular.

SCADA

ScheduleZ for its SCADA system has adopted the WorldFip standard.

In the previous year the architecture of an interface between the low level monitoring and control and the higher level planning and scheduling world, based on **WorldFip**, was developed and demonstrated in an early prototype (EXACTA exhibition Turin 1994). In the current year the architecture has been revised, completed and made compliant to the emerging standards. This was possible by the direct interaction and participation of ScheduleZ partners in the **WorldFIP** working groups for the definition of the system companion standards.

The **KP-Food** -demonstration is in fact heavily based on this **FIP** architecture and resulting framework of classes.

In this new release of the report the architecture is extensively described, a framework of classes has been developed for building MS-Windows based **SCADA** applications, and an user manual has been written for the construction of SCADA systems.

The field “testing and evaluation of the original **COMPAL** scheduler indicated that, in spite of its object oriented architecture, it still showed reusability problems. To solve these problems, especially in decision making systems, the modern trend is to integrate the object oriented technology with constraint programming.

The **SCADA** simulation framework extensively developed in the last six months of ScheduleZ is highly innovative. The **KP-Food** scheduler operates reactively from the data received from the shop-floor. In order to test the integration of the scheduler

with the field in controlled conditions it has been decided to prepare a simulation of the production environment of KP-Food to substitute to the real plant.

The simulation environment for preparing this simulation has been provided by the proposed SYCO framework RODIN. This framework allows to represent concurrent activities, manage Processes and events in simulated (discrete event simulation) and emulated (the time flow is controlled by the processor clock) environments.

With this framework the upper layer of the SCADA system has been modelled and implemented. It allows to control production lines based on the scheduler plans, to routinely monitor the line production, and to receive from the machines events of machine failures, start, stop and end of production lot. Moreover this layer interconnects the production with the host computer homing the scheduler facilities using the FIP communication network.

The model of the process used for simulation is, however, much more than a simple simulation. In fact the model of the application is a prototype of the real implementation. If desired, after completing the simulated integration test it could easily be transformed into the physical application monitoring the real production process. Extremely impressive results were achieved in such a short time.

CONCLUSIONS

The objectives of **ScheduleZ** followed closely the original Workprogramme. The SCADA work however took another implementation perspective. Instead of the original simple factory model to be installed in one of the lines at **KP**, it was developed in a simulation environment. This allowed the extension of the analysis to a much more complex network. Three workstations representing each layer of communication interact to drive the scheduler using shopfloor information.

ScheduleZ has achieved all the objectives it set out to reach and produced powerful tools and methods in :

1. Multicriteria graphical configuration systems;
2. Graphical OOD libraries;
3. Finite capacity scheduling blueprints;
- 4.00 tools for cooperative interaction with SCADA systems;

Powerful OO frameworks were designed and implemented and an extensive library of OO classes built.

Early exploitation of results is taking places with several of the partners migrating **ScheduleZ** results to their commercial and research environments. Commercial applications are being used or developed faster than estimated.

ScheduleZ was a very successful project which exceeded the expected results.

ACKNOWLEDGMENTS

All the partners in the **ScheduleZ** Consortium would like to acknowledge the support of the European Commission in the form of a Brite-Euram project (**BREU-CT92-0567 project 4140**) and also the advice and constructive critical analysis of our Project Officer Mr. Hans Pedersen and our PTA Mr John Crompton. Mr Hans Pedersen dedicated many hours to help **ScheduleZ** adapt to the original plan without major deviations. A special thank you from the **ScheduleZ** team !