The PDES Workbench

Roman VAN DER KROGT[1] and James LITTLE
Cork Constraint Computation Centre,
University College Cork, Cork, Ireland

Abstract. Central to the operation of a manufacturing plant is the planning and scheduling of the activities that take place. The quality of the plans and schedules produced has significant impact on the effectiveness and efficiency of a company. These schedules are restricted by the constraints imposed upon them by the design of the plant. However, while expert designers are able to roughly predict the outcome of a design in terms of its overall throughput, quantitative figures on future schedules are not usually produced. By implication, design constraints may be found to be expensive in scheduling terms during the operation of the plant and at a stage when improvements are hard or expensive to make.

This paper presents the PDES Workbench, a graphical system that is able to generate plans and schedules based on real-life manufacturing plant designs. This allows a manager or planner to assess the schedulability of a particular design at an early stage. When a schedule is found to be flawed in any aspect, it may not be apparent what to change in the design without degrading other aspects of the schedule. Therefore, the system is equipped with a case-based reasoning system that is able to proactively suggest ways of improving the design for improved scheduling results.

Keywords
Advanced planning and scheduling, process design optimisation

1. Introduction

Manufacturing processes are complex; often, involving a large number of resources and activities within a diverse set of restrictions. To achieve an efficient use of these resources, Advanced Planning and Scheduling (APS) techniques can be employed, such as discussed in e.g. [1]. The effectiveness of these techniques depends to a large extent on the process design. Indeed, we argue that the design of the process has a profound impact on the schedulability of the activities involved and the quality of the resulting schedule. Since the quality of the schedules has a direct influence of the production cost, it is important to take this into account during the design of processes. However, as suggested by Prasad [2], “process improvement is often perceived as an after-thought – a functional service to be called upon periodically for productivity improvement”. One of the reasons for this is due to the complicated interactions between the different activities in the manufacturing process. The designer of such a process cannot exactly predict the scheduling behaviour of a design, as it may introduce several constraints that only become apparent during the actual scheduling process.

The case studies for our research are based on a large manufacturer of contact lenses. These lenses are produced in a two-step process, which consists initially of a

---

[1] Corresponding Author; E-Mail: roman@4c.ucc.ie

moulding process that produces moulds. The moulds are then used (and later destroyed) in the *casting* step, which creates the actual lenses. Due to stability considerations, moulds can only be used within a specific time window after their production. There are also different ways in which the mould stock can be managed, further adding complications to the schedule. Aspects such as this make it hard, even for an expert, to predict the schedule from a design.

This paper introduces the PDES workbench, a prototype tool that can assess the quality of schedules that result from a certain design. Constraint-based techniques are used to model and efficiently produce the schedules we are interested in, whereas techniques from A.I. and more specifically case-based reasoning are used to learn from previous designs in suggesting improvements. The prototype was built for the particular manufacturing company that we described, but with adaptability and extendibility in mind. As such, we believe that it is also applicable to other manufacturing domains.

This paper is organised as follows. First, we briefly describe the techniques that we have employed, constraint-based scheduling and case based reasoning, and the reasons for choosing these approaches. We proceed in Section 3 by describing the specific case to which we have applied to tool, and show how the design of the tool follows from that. Then, we evaluate the tool by showing the kind of scenarios it can deal with and finally, we draw conclusions in Section 5.

2. Relation to Existing Techniques: APS & CBR

The two techniques upon which the PDES workbench is built are constraint-based planning / scheduling to determine the schedulability of a design, and case-based reasoning to improve upon flawed aspects of a design. This section briefly introduces the two techniques.

2.1. Constraint-based Planning and Scheduling

Constraint programming is a problem solving methodology built around the identification of variables within a problem, a domain of values for each variable, and a set of constraints that specify which combinations of values are allowed. A solution is an assignment of values to variables such that all constraints are respected. With the addition of optimisation criteria, constraint programming is a rich mathematical infrastructure that can be used to model and solve a variety of economically interesting problems such as scheduling [3;4].

Constraint programming is a proven technology in scheduling optimisation for manufacturing enterprises. For example, the scheduling of production and delivery as well as the efficient use of raw materials are all problems for which CP-based solutions exist from such vendors as SAP, Oracle, and i2. Other reasons for choosing a constraint-based solution over other scheduling techniques are the facts that it is both rich and efficient. Its richness enables us to create models that capture all the details of a particular design, while its efficiency ensures that we can quickly produce acceptable solutions.

2.2. Case-based Reasoning

Case-based reasoning (CBR) is another problem solving strategy. It is based on reusing experience gained in previous problem solving episodes [5]. CBR starts from a previously generated solution and adapts this solution to match the current problem. This method of problem solving is akin to the way humans appear to solve certain problems [6].

In a CBR system, expertise is embodied in a library of previous *cases*. Each such case consists of a description of the problem along with its solution. If a new problem is to be solved, the following steps are taken:

1. *Case retrieval*. The new problem is compared with the cases in the library, and similar cases are retrieved.
2. *Case reuse*. If the problem does exactly match one of the cases just retrieved, we can reuse the solution that was recorded. If an identical problem was not found, the retrieved cases are used to suggest a solution to the new problem
3. *Solution revision*. The solution that was constructed in step (2) is tested to see if it is indeed a valid solution to the problem at hand. If it is not, the solution has to be revised.
4. *Case Retention*. The last phase involves deciding whether the current problem and its solution should be committed to memory as a case, or whether it is too similar to the existing cases.

One of the advantages of case-based reasoning is that it can be used in situations where knowledge acquisition is hard or impossible due to the fact that there are no known rules governing such a complex situation. This is the primary reason for us to employ this technique.

3. Design of the Tool

We have developed a research prototype tool in collaboration with a large manufacturer of contact lenses. As we indicated in the introduction, the complicated process used to manufacture the lenses is both an example and the motivation of our work: the subtle constraints imposed by the design on the possible schedules are hard to predict. However, even though we built the prototype with a particular application domain in mind, it was designed to be both adaptable and expandable as we detail below.

The architecture for our system can be seen in Figure 1. Our implementation allows different manufacturing process designs to be evaluated through a flexible user interface which interacts with a constraint-based optimiser and a case-based reasoning module for design improvements. The prototype is based on common desktop tools, which are readily accepted as standard in many industrial organisations. The graphical frontend to the workbench uses MS Visio to represent the product routes through the factory and the actual layout of machines on the factory floor, see Figure 2. The advantage of this setup is that the user can easily relate the information on screen to the actual circumstances. The tool allows exploring what-if scenarios by easily being able to add machines to the current design, remove them, or alter their characteristics (such as production rates). It also permits the alteration of a number of parameters of the process itself (such as the level of buffer stock between the two steps of the process or the stabilisation times), as well as changes to demand patterns.

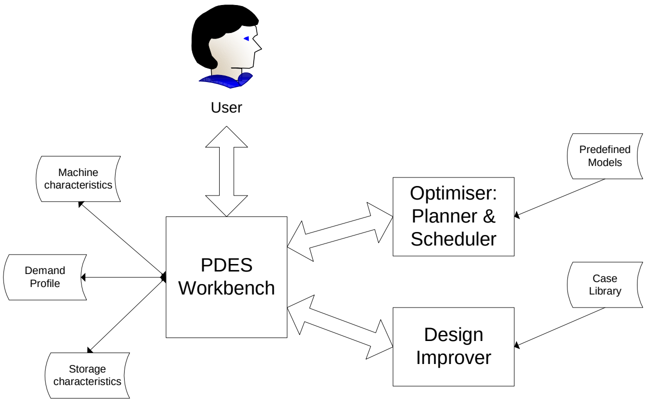

Figure 1. Overview of the architecture of the system.

Within the system, the process design is converted into a plan of necessary activities and then an optimisation data file. This is passed to the optimiser, where an appropriate pre-built model is selected (see [7]). The optimiser in this case is ILOG OPL Studio [8], a constraint-based problem solving technology that offers a variety of built-in solving techniques.

The models and their solution algorithms are selected such that they quickly produce an initial solution, and produce more refined solutions given more time. This way, the user may explore a number of scenarios quickly, while studying promising scenarios more thoroughly by allocating more time to the optimiser. Initial solutions are typically found within one to two minutes. The results that we present in the next section were achieved with 10 CPU minutes.

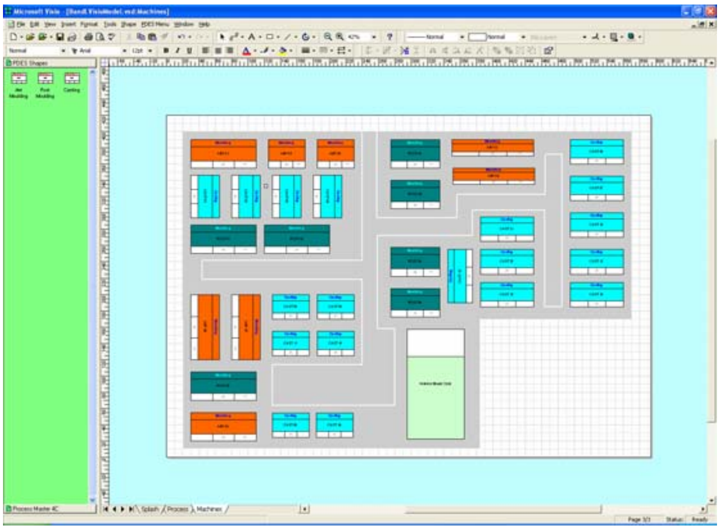

Figure 2. Graphical User Interface of the Workbench.

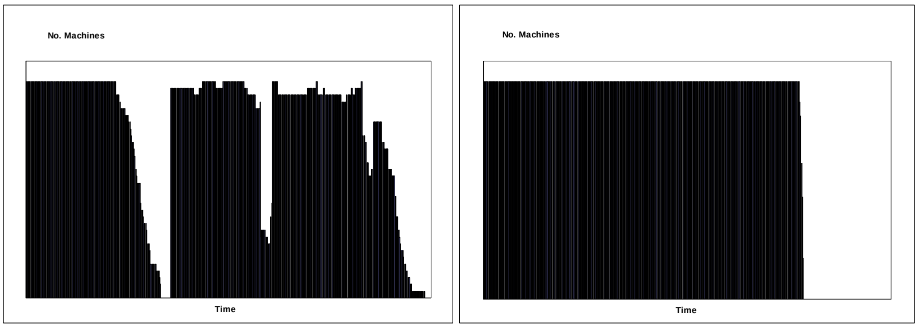

Figure 3. Examples of resource profiles.

Once the schedule is obtained, then the key performance indicators of manufacturing time, utilisation and lens production are calculated. These results are then presented to the user within MS Excel. This desktop tool has a sufficient level of functionality in statistical analysis and also in the representation of data as tables, charts and graphs. The next section contains a number of examples of the latter.

Should one or more performance indicators be unsatisfactory or even if the design is so flawed that no valid schedules can be produced for it, the user can ask the Design Improver subsystem to recommend changes to the current design. This part is implemented in Visual Basic using Excel files to store the case library. To obtain a recommendation, the user has to indicate which performance indicators are to be improved, and the relative importance of each of those indicators, in case a trade-off has to be made. The CBR system then compares the design and the demand pattern with previous designs and patterns, and selects those cases that resemble the current problem using a Euclidean distance measure. These cases are subsequently presented to the user, ranked according to the improvements that can be made to the chosen performance indicators. The user may then apply the recommended changes and evaluate the results again.

4. Evaluation

The PDES workbench has been validated by verifying a number of historical situations that the company experienced. In all cases, the predictions made by the planner / scheduler matched the experiences of the company and indeed suggested others which were not. The Improver subsystem was evaluated for user experience, as no historical data was available within the company for this type of system. The scenarios evaluated are presented below.

4.1. Historical Evaluation

The first scenario involved the level of buffer stock kept between the moulding and casting processes. The company currently holds more stock than necessary to ensure smooth casting operations. If casting were to fail, e.g. due to faulty moulds, then with insufficient stock the company would have to produce moulds again from the start. This would impact their schedule, by having to stop casting. In this scenario, we explored different levels of stock and showed the effect on the resulting schedules. One of the aspects here is the operations of the casting machines. Figure 3 shows the resource profiles for the casting machines for two cases.[2] On the left, we have a minimal amount of stock. As one can see, the casting machines are not fully used throughout, as there regularly are intervals in which not enough moulds are available to keep all casting machines running. On the right-hand side, we have a situation in which enough stock is kept to keep the machines running. Not only does this lead to a much smoother operation of the casting machines, also the time to produce the required products decreases. By exploring different levels of stock, and weighing the savings gained by a smoother and quicker casting against the cost of higher stock levels for different scenarios the company was able to confirm their current stock levels were satisfactory. However they could design lower stock levels and still achieve an acceptable schedule in which all lenses were manufactured in time.

The second scenario revolved around the material of the moulds. The supplier announced a new type of plastic, with different stabilisation parameters (defining the time window within which a mould is to be used). To investigate the effect of this new material on the manufacturing process, we simply had to change the parameters defining the moulding material and examine the quality of the produced schedules. As we can see from Table 1, it turned out that, perhaps surprisingly, the new material with the "improved" characteristics had a dramatic effect on the schedulability: unless full stocks were kept, there were instances where the new material prevented feasible schedules to be produced.

Table 1. Schedulability for different materials

material	stock level	makespan	moulding util%	casting util%
original	minimal	7.11	96.3%	88.2%
	50%	5.51	90.5%	85.6%
	full	5.18	96.3%	99.3%
new	minimal	-	-	-
	50%	-	-	-
	full	5.68 days	96.3%	99.3%

The third scenario explored changes to the demand pattern. In this case, one can leave the design of the factory as it is and input different sets of orders to see how well the design can deal with these sets. Alternatively, one can use the tool to see what the best response is to a foreseen change in orders. As a particular example, we refer to Table 2. Here, we investigate (for a representative, but smaller set of weekly orders and a scaled-down factory) the effect of a 12.5% increase in volume. As we can see from the second row, the current design cannot cope with this increase, as it would take 7.1 days to produce the weekly demand. The third and fourth rows explore the result of buying additional machines. The company saw this part of the system as accurately reflecting the outcomes of historical decisions. From this grew the confidence that this type of architecture could help support future decision-making.

---

[2] In the interest of the company, the figures are provided without scales.

Table 2. Effects from an increase in volume of orders

order size	# mould. mach.	# cast. mach.	mould util%	cast util%	makespan
100%	2x4	12	94.2	95.4	6.5 days
112.5%	2x4	12	95.4	99.6	7.1 days
112.5%	2x5	13	95.7	97.2	6.0 days
112.5%	2x6	14	95.8	93.6	5.3 days

4.2. Improving Design Using the Case-based Reasoning Module

We created a case library consisting of 15 designs and their schedules from three different order sets of different sizes (corresponding to 50%, 100% and 200% of an order set of representative size). We then presented the system with a flawed design, and asked the system to recommend design changes that would correct the flaw.

As an example, consider a design in which the total capacity of the moulding machines is not enough to produce enough moulds. This design and some of its KPI are listed in the first row of Table 3 below. When asked to improve upon the makespan of this design, the system is able to retrieve a similar design with two additional moulding machines, one of each type. This design resembles the current design to a high level, and improves the makespan by 10%. Notice that an improvement of 11% can be made by a more radical change in design. This design is presented only as the 2nd suggestion however, since it deviates from the current design too much.

Table 3. Designs suggested by case-base (simplified representation).

	# mould. mach.	# cast. mach.	mould util%	cast util%	makespan
Input	2x7	32	90.7	89.5	7.4 days
1st suggestion	2x8	32	92.5	90.8	6.6 days
2nd suggestion	2x4 (double speed)	32	96.6	92.8	6.5 days

5. Conclusions and Future Work

In this paper, we presented a tool to assess designs of a manufacturing process for their scheduling consequences. This is a valuable aid particularly for designers of complex processes. Such processes may introduce hidden constraints that only become apparent during the actual planning and scheduling of the resources. These hidden constraints make it hard to predict the behaviour of a system beforehand. The PDES workbench allows a designer to explore what-if scenarios early on in the design process to take the schedulability aspect of designs into account.

The prototype system was evaluated within a large manufacturing company. The company have viewed the system positively in addressing the types of problems they are faced with on a regular basis. Presently they are able only to evaluate a few scenarios when operation conditions change. The planning department now have visibility of the scheduling process and are able to plan the factory operation with some knowledge of the scheduling consequences. The tool has been used to validate historical decisions. In all explored cases, the results of the tool corresponded with the experiences in the factory. The case-based reasoning module, although not used in a real-world setting, was perceived as a useful addition.

The layout of their packaging plant is a current design problem for which the system is to be used to evaluate options. Here there are different performance of machines and dedicated machines to products. These types of constraints are already built into the prototype system.

One specific area that we are looking into is the automatic generation of optimisation models given a process diagram. This would greatly reduce the effort required to adapt the tool to a specific domain.

The design components of the manufacturing process such as route plans, resource layouts and demand profiles are common to other types of manufacturing. For this reason we intend to apply the same methodology to different manufacturing processes.

Acknowledgements

This work has received support from Science Foundation Ireland (Grant 00/PI.1/CO75), Enterprise Ireland Innovation Partnership (IP/2003/160) and Bausch and Lomb.

References

[1] Steger-Jensen, K and Hvolby, H-H. Constraint Based Planning in Advanced Planning and Scheduling Systems, in *Proc. 8th International Conference of Concurrent Enterprising*. Rome, Italy, 2002.

[2] Prasad, B. *Concurrent Engineering Fundamentals, Integrated Product and Process Design*, Vol. 1, Prentice Hall, 1996.

[3] Fox, M.S. *Constraint-Directed Search: A Case Study of Job-Shop Scheduling*. Ph.D. Thesis, Carnegie Mellon University, USA, 1983.

[4] Wallace, M. Practical Applications of Constraint Programming. In *Constraints*, 1(1-2):139-168, 1996

[5] Leake, D.*Case-Based Reasoning: Experiences, Lessons and Future Directions*. AAAI Press, 1996.

[6] Ross, B.H. Some Psychological Results on Case-Based Reasoning. In *Proc. DARPA Case-Based Reasoning Workshop*, pages 144-147. Morgan Kaufman Publishers, 1989.

[7] Selensky, E., Little, J. and Beck, C. Process Design for Efficient Scheduling. In *Proc. 23rd Annual Workshop of the UK Planning and Scheduling Special Interest Group*. Cork, 2004

[8] Lustig, I.J. and Puget, J.F. Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming. *Interfaces Journal*, 31(6):29-53, 2001.