Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 19 (2014) 160 – 167

www.elsevier.com/locate/procedia

Robust Manufacturing Conference (RoMaC 2014)

Knowledge enriched short-term scheduling for engineer-to-order products

D. Mourtzis[a]*, M. Doukas[a], K. Vlachou[a], K. Fragou[a], C. Vandera[a]

[a]*Lab for Manufacturing Systems and Automation, Department of Mechanical Engineering and Aeronautics, University of Patras, Rio, Greece, 26500*

* Corresponding author. Tel.: +30 2610 997262; fax: +30 2610 997744. *E-mail address:* mourtzis@lms.mech.upatras.gr

Abstract

Contemporary shop-floors are highly affected by the ever-increasing complexity that is caused by the fluctuating customer demands. Therefore, a high degree of flexibility is needed and the scheduling of manufacturing tasks must be agile to changes. For addressing this challenge, this research work proposes a knowledge enriched short-term job-shop scheduling engine. More precisely, it focuses on the short-term scheduling of the resources of the machine shop, through an artificial intelligence algorithm that generates and evaluates alternative assignments of resources to tasks. Based on the requirements of a new order, a similarity mechanism retrieves successfully executed past orders together with a dataset that includes the processing times, the job and task sequence and the suitable resources. Afterwards it adapts these parameters to the requirements of the new order so as to evaluate the alternative schedules and identify a good alternative in a timely manner. The deriving schedule can be presented on mobile devices and it can be manipulated by the planner on-the-fly respecting tasks precedence constraints and machine availability. A case study from the mold making industry is used for validating the proposed framework.

Selection and peer-review under responsibility of the International Scientific Committee of “RoMaC 2014” in the person of the Conference Chair Prof. Dr.-Ing. Katja Windt.

Keywords: Manufacturing Systems; Scheduling; Mobile Applications;

1. Introduction

Modern manufacturing relies on the reuse of past knowledge that is contained in data repositories and IT systems, as well as in the form of tacit human knowledge. Knowledge constitutes a key factor for improving manufacturing performance, during design, planning and operational phases [1, 2]. The importance of knowledge reuse for a system’s planning phase is evident, as rough estimates indicate that more than 20% of an engineer’s time is spent on searching and absorbing information for a new project [3].

The production of engineer-to-order products, is a particular type of manufacturing system, which essentially relies on the expertise of human resources. Usually in this type of business, the scheduling of new orders in an already occupied manufacturing system, is performed empirically and using rules of thumb. However, with the rising complexity of production requirements and the increased penetration of IT systems in manufacturing, knowledge reuse is necessary for reducing the product development cycle. On the contrary, currently valuable knowledge generated and associated to products and processes in a daily basis, remains tacit and its reusability is confined to a specific operator or planner [4].

In this research work, a scheduling mechanism is proposed that is enhanced with an integrated knowledge reuse mechanism. The knowledge reuse mechanism retrieves executed scheduling cases, and through a Case-Based Reasoning methodology extracts information related to the modelling of the scheduling workload. The deriving workload model includes necessary input for a scheduler, such as the job structure and the task breakdown, the precedence constraints, and the processing and setup times. Alternative schedules are generated and are optimized using multiple conflicting criteria, such as flowtime and tardiness. The scheduling is performed using an Intelligent Search Algorithm with three tunable parameters, which are adjusted through a parametric investigation, using a Statistical Design of Experiments. All functionalities are exposed through the designed mobile app. This work extends the research presented in [5, 6] by enhancing the scheduling algorithm with knowledge reuse capabilities and by verifying the method in a case from the domain of engineer-to-order

Selection and peer-review under responsibility of the International Scientific Committee of “RoMaC 2014” in the person of the Conference Chair Prof. Dr.-Ing. Katja Windt.

doi:10.1016/j.procir.2014.05.012

products. Finally, it should be mentioned that the two engines are designed in a modular way. The scheduling engine is capable to function without input provided by the knowledge mechanism, if the latter is not available. Similarly, the knowledge mechanism is decoupled from the scheduling engine and can be used for extracting manufacturing information for different purposes, such as for the estimation of the delivery time of an injection mold [7].

2. State of the Art

Throughout the years, several methods have been proposed for knowledge reuse in the manufacturing domain. They aim to support designers and engineers in decisions related to modelling, design, prediction, monitoring, simulation, and optimization and in general knowledge intensive domains of manufacturing. There are two main ways to reuse past knowledge: reuse the past case solution and reuse the past method that constructed the solution [8]. Case-based Reasoning (CBR) process is an Artificial Intelligent technique that retrieves past experience to reuse for target problem; of course, the solutions of past cases may need to be revised for applying. The successful problem-solving experiences are then retained for further reusing [9]. The CBR method is utilized in this research work due to its suitability for complex ill-defined concepts, with unstructured knowledge and because case generalization is required [9].

The second area of interest in this research work is production scheduling. Scheduling of operations is one of the most critical issues in planning and managing of manufacturing processes. In most SMEs that cannot afford strong investments in software solution, scheduling is carried out empirically. The definition of an optimum solution is quite difficult, depending on the job shop environment, process constrains and performance indicators [10]. Numerous approaches have been reported for the modelling and solutions of the job shop scheduling problem. Wang et al., [11] proposed the development of an application using a genetic algorithm including a chromosome representation in seven different machines of a job floor that enables a dynamic job shop scheduling within complex production systems. Moreover, a task model allowing the representation of activities with optional parts and several scheduling algorithms to incorporate them into real time systems is described in [12]. Chryssolouris et al. [5] considering the issues rising from static scheduling proposed a dynamic scheduling problem to accurately reflect a real job shop scheduling environment. However, literature findings that focus on knowledge reuse as an enabler for improving scheduling performance are insufficient. Motivated by empirical knowledge, [13] proposes an efficient search method for the multi-objective flexible job-shop scheduling in order to reach high automation levels towards generating optimal or near-optimal production schedules. Another study proposed a data mining technique for discovering dispatching rules that improve scheduling performance [14]. The job-shop scheduling problem has been addressed using a knowledge enriched genetic algorithm in [15]. The idea was to imbue production system knowledge during the formulation of the initial population of the algorithm with the potential of faster and better convergence. The authors in [16] utilise data mining for optimizing the assignment of due dates to orders dispatched in a dynamic job-shop.

The third area of interest is mobile technology. Mobile technology evolves rapidly; in the last decade the use of mobile apps has outpaced traditional PC-based web-browsing [17]. The usage of apps doubled on average over the last year, with utility and productivity apps second in growth [18]. The adoption of apps focused on core manufacturing processes, which was up to now limited [19], is finding its way into activities such as manufacturing network design [20] as well as in other scientific domains [21, 22]. The necessary components of apps in order for them to be fully leveraged in manufacturing are presented in [23], where architecture, development, infrastructure, security, portfolio and privacy issues are investigated. Short-term estimations speak of apps boosting productivity by 5%-10% [24].

3. Knowledge Enriched Scheduling

3.1. Overview of the Method

The Knowledge Enriched Scheduling engine, hereby referred to as KES, consists of two mechanisms, namely: i) the knowledge extraction and reuse mechanism and ii) the short-term scheduling mechanism (Fig. 1).

Once a new order enters the system, a break-down of the product components into a Bill of Materials (BoM) structure is carried out. The product is characterized by a number of attributes (product features) that are used by the similarity mechanism for a pairwise attribute comparison. The outcome of the similarity comparison is an ordered list that contains the past cases ranked from the most to the least similar. By reusing the knowledge stored in these past cases, the expert planner is allowed to extract valuable information that helps him introduce the new order into the production system with the needed adaptations. The reusable information includes the required number and type of jobs, the number of tasks for each job and their precedence constraints, and finally the processing times for each task in specific machines. The output of this process is the necessary input for a scheduler. It noted here that specific process planning information, such as cutter selection, process parameters and fixture specification are beyond the direct scope of the proposed work.

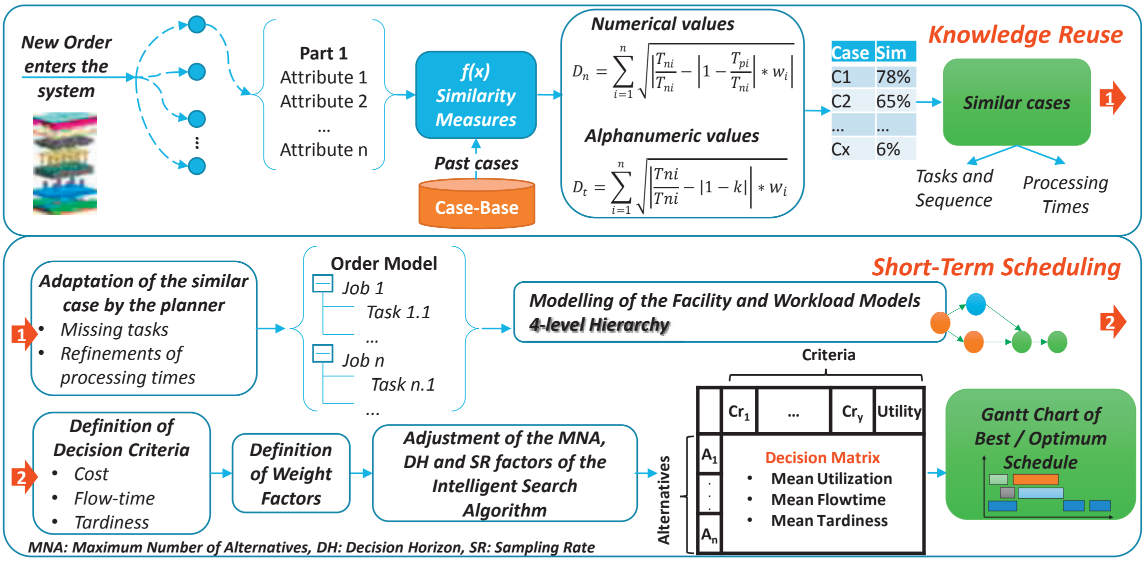

Fig. 1. Overview of the Knowledge Enriched Short Term Scheduling engine

The latter component of the KES is the short-term scheduling mechanism. After the identification of the most similar cases, the expert planner aggregates information that can be reused in the new case and adapts it. The adaptation is required in order to compensate for missing tasks that were not identified during the similarity measurement, or in order to imbue to the dataset the actual situation of the shop-floor (machine break-downs, availability). The result of the adaptation is the model of the workload and the facility. These models are inserted in the intelligent scheduling engine. The planner defines the decision-making criteria and their weight factors, which reflect the design and planning objectives of the system. Following on that, the definition of the tunable parameters of the scheduling algorithm are defined. The tunable parameters are the Maximum Number of Alternatives (MNA), the Decision Horizon (DH) and the Sampling Rate (SR). The description of the function of these parameters is described in section 3.2 below. The scheduling algorithm generates scheduling alternatives and through a decision matrix selects and displays the best in a Gantt chart form.

3.2. Modelling of the Facility and the Workload

The production facility is hierarchically divided into job-shops that contain work-centers, which in turn contain a number of resources. The latter are individual processors with diversified processing capabilities (machining technology, cycle times). Similarly, the workload model includes orders broken down into jobs, each containing a number of tasks.

Tasks are assigned to resources based on an intelligent multi-criteria search algorithm that generates, evaluates and selects the scheduling alternatives, as described in [25]. The Intelligent Search Algorithm (ISA) evaluates the alternatives in a decision matrix based on setup cost and processing time criteria. A utility function is used for ranking the alternatives and for selecting the highest performing one.

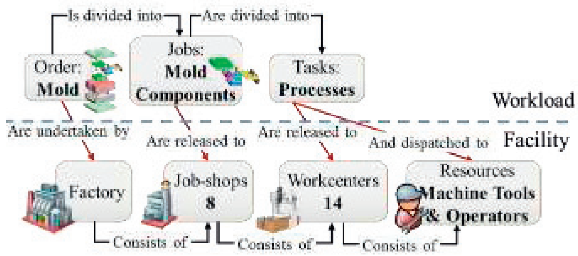

Fig. 2. The four-level hierarchical workload and facility model

As depicted in Fig. 2, the orders are dispatched to the facility, the jobs to the job-shops and the tasks to work-centers' resources. The resources are not parallel processors and their availability is subject to the system workload. The release of jobs and tasks takes into consideration constraints such as finite capacity, precedence relations and availability. The job and task modelling is presented in Fig. 3.

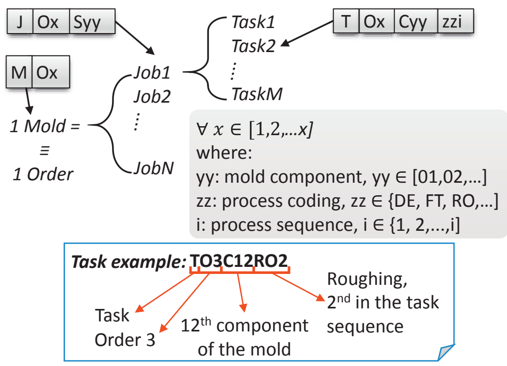

Fig. 3. Modelling of the mold order, jobs and tasks

3.3. Description of the Knowledge Reuse Mechanism

In today's evolving environment, planners require efficient methods for generating schedules for new orders that their industries take on in a fast and accurate manner. This task can be supported through the utilization of existing knowledge retrieved from past schedules. As a result, the first step in the workflow of the presented research work is the comparison of the new order requirements against past cases. This similarity measurement emphasizes on the differences exhibited between the basic attributes that characterize old and new orders. The past cases are retrieved using the Case-Based Reasoning methodology and are compared with similarity mechanisms. Similarity mechanisms recognize the type of attributes; numeric or alphanumeric values are considered. The alphanumeric attributes take discrete values and are matched with numbers between zero and one for normalization reasons. Moreover, both attribute types are multiplied with weight factors, considering their influence on the actual similarity between cases. Eqs. (1) and (2) are used for measuring the Euclidean distance through a pairwise comparison between the attributes of past and new cases. Eq. (3) aggregates the results of the two distance metrics.

$$D_n = \sum_{i=1}^{n} \sqrt{\left| \frac{T_{ni}}{T_{ni}} - \left| 1 - \frac{T_{pi}}{T_{ni}} \right| * w_i \right|}, \text{ numerical values} \tag{1}$$

$$D_t = \sum_{i=1}^{n} \sqrt{\left| \frac{T_{ni}}{T_{ni}} - |1 - k| \right| * w_i}, \text{ alphanumeric values} \tag{2}$$

$$S = (D_n + D_t)^2 \tag{3}$$

where: D_n=numerical distance, D_t=text distance, n=number of attributes, T_{ni}=ith attribute of the new case n, T_{pi}=ith attribute of the past case p, k=mapping for alphanumeric attributes and w_i=the weight of attributes.

The past case with the highest similarity index is analyzed first. The planner may retrieve the process sequence, precedence constraints, the components and the resources used in the past case, as well as processing and setup times. Moreover, based on their experience, expert planners have the capability to recognize if the retrieved data are adequate to describe the new order. In case they are insufficient, the planner adapts the dataset to the requirements of the new case. In order to further enhance the knowledge retrieval, in cases when the new product requires a different amount of components or processes than the retrieved most similar case then, the second similar case can be consulted and afterwards the third and so on. Either way, the similarity index must remain above the threshold of 60%, which is calculated based on historical observations, otherwise the retrieved information would be misleading. Indicatively, if the best match in terms of similarity index is fairly old in comparison with the new case, it is most probable that adaptations would be required to compensate for changes in the shop-floor, such as addition of new manufacturing resources and technologies. In this case, engineers are aware of the current state of the shop-floor and can replace the old resources with similar ones in the new process plan. Having decided the matching past similar cases, the task sequences are retrieved, the availability of the machines is confirmed and then the final combination of the new sequence of processes and components is settled.

3.4. Description of the Short-term Scheduling Mechanism

The operational policy behind the assignment of a task to a specific resource can be either a simple dispatching rule, or a multiple-criteria decision making technique described below. The advantages of dispatching rules derive from their simplicity, since they do not attempt to predict the future, but rather make decisions based on the present. Thus, these rules are very useful in factories that are extremely unpredictable, such as job shops. Also, they are spatially local, requiring only the information available at the location where the decision will be implemented. Finally, they are easily understood by human operators and are easy to implement [26]. On the other hand, the multiple-criteria decision making technique, involves the formation of several alternatives and their evaluation before assigning the available resources to pending production tasks.

Schedules are constructed on the basis of events occurring sequentially through time. Thus, the next scheduling decision is identified by moving along the time horizon until an event (release of a new order in the system or the completion of a task) is scheduled to occur that will initiate a change in the status of the system [27]. The set of pending tasks become eligible for release at the time a resource becomes available. Since the method considers a finite capacity problem, in case multiple jobs are competing for a resource, the Intelligent Search Algorithm (ISA) and the decision matrix with the criteria and their weighting factors are used to determine which task will be dispatched to which resource, optimizing the planning objectives.

In the ISA algorithm is the search of the solution space is guided by three adjustable control parameters, namely the Maximum Number of Alternatives (MNA), the Decision Horizon (DH) and the Sampling Rate (SR). MNA controls the breadth of the search, DH controls the depth and SR directs the search towards branches of high quality solutions [28]. The proper selection of MNA, DH and SR allows the identification of a good solution by examining a limited portion of the search space, thus effectively reducing computational time. A Statistical Design of Experiments [29] has been carried out to reduce the number of experiments and to identify an optimum set of these factors in order to obtain the results of the highest possible quality [6, 28]. The workflow of the algorithm follows:

Step 1: *Start at the root and generate alternatives by random assignments for DH layers until MNA*
Step 2: *For each branch (Step 1), create SR random samples until all the branch nodes are searched*
Step 3: *Calculate the criteria scores for all the samples belonging to the same alternative of Step 1*
Step 4: *Calculate the score of the branch as the average of the scores achieved by its samples*
Step 5: *Calculate the utility values of each alternative/branch*
Step 6: *Select the alternative with the highest utility value*
Step 7: *Repeat Steps 1-6 until an assignment has been done for all the nodes of the selected branch*

3.5. Description of the Knowledge Enriched Scheduling App

The scheduling and the similarity mechanism components have been implemented in C++ for validation purposes. The integrated knowledge enriched scheduling engine has been designed for implementation into a mobile app for the Android OS. The designed app allows data entry, selection of decision-making criteria, definition of weight factors and tunable parameters of the ISA, and results' visualization. The alternative with the highest utility value is displayed together with the scheduling Gantt chart and the mean values of the performance indicators (utilization, flowtime, and tardiness). The app will also allow operator interaction in case the derived schedule is not acceptable or needs refinements due to order prioritization and machine breakdowns among other reasons (Fig. 4). The precedence constrains, machine availability and capacity, and due dates are respected during rescheduling. Moreover, performance indicators are re-calculated each time a rescheduling occurs.

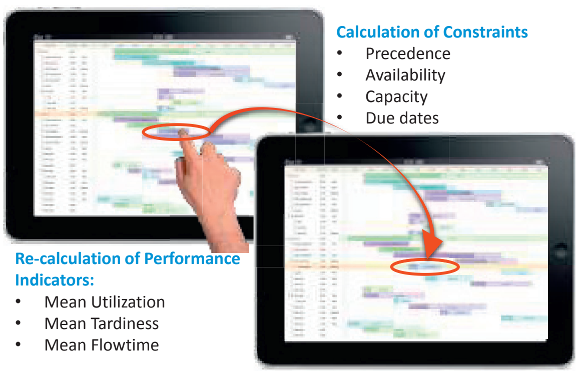

Fig. 4. Rescheduling performed by the operator through a tablet

4. Design of the Knowledge Enriched Scheduling App

Mobile apps deployed on the Android OS are based on a 3-tier architecture that consists of 3 layers (data, business, and presentation) following the rules of the Model-View-Controller architectural pattern. The Presentation layer includes the Graphical User Interfaces of the app and the Data Layer retrieves data from the back-end. Finally, the Business layer handles the data exchange between these two layers.

For the programming of the platform, the Android SDK (Software Development Kit) is required, which provides the developers the API libraries and tools necessary to build, test, and debug apps for Android. The back-end will be implemented with the Apache Tomcat v7.0.19, since it is fully compliant with the latest advances in web programming and servlet specifications. The supporting data model of the app is based on requirements' collection from a mold manufacturer. The application runs of devices with low specs for today's standards (ARM-based processor, 512 MB minimum memory and 300MB free minimum storage space, OS Android 4.0™ or later).

5. Industrial Case Study – Experiments and Results

The case study uses real data from a high-precision mold-making machine shop. The mold-shop best fits to the make-to-order business model, where custom molds and dies are designed and manufactured based on customer orders. Injection molds are one of a kind, first-time-right products that vary greatly in terms of quality, tolerances, and mainly functionality. Evidently, mold-making is highly specialized and knowledge depended industry. Once a new order is received, its planning follows. Work is delegated among engineers, based on their expertise, who are usually in charge of a project from start to end. The resources needed are determined by the project's particularities. In the current business model, short-term scheduling is performed empirically on a daily basis, in close collaboration between engineers, management, as well as work station supervisors. Unofficial oral meetings take place in order to schedule resources, and, if the situation demands it, the management department is involved in the decision making and work prioritization. However, no software tools are used to support short-term scheduling or to document the decisions made.

The shop-floor of this case study is comprised of 8 job-shops, which include 14 work-centers and 40 machines in total. The machines include high precision CNC machines capable of the following processes: milling, drilling, turning, electro-discharge wire cutting, sinking, grinding, tapping, roughing, polishing and hardening. Some operations are performed manually such as design, fitting, assembly, measuring and polishing. The hierarchical model of the production facility is included in Table 1. The dataset of the case study includes the documented processing times, tasks, sequences and resources used for the manufacturing of thirty (30) finished products that were carried in a timespan of approximately three years.

As described in section 3.3 above, the new order that triggers the scheduling mechanism, is first compared against documented past cases for the reuse of knowledge related to processes and product structure. In the case study, the new order is compared against all 30 documented past cases. Moreover, in actual production terms, five (5) orders (molds) are simultaneously executed in the shop-floor on average. Therefore, in the experiments below, four of the orders are already under processing and the new order enters the system eight calendar days later. The schedules for these orders have been inserted in the scheduling engine by the planner. The system is then rescheduled in order to accommodate the new order. The new mold order carries the identification number "13.23" and its basic attributes are shown in Table 2. A fundamental parameter that is taken into consideration is the stacks' shape. There are two options, namely molds with cylindrical and rectangular stacks. Since the mold 13.23 has cylindrical stacks, the attribute "length" is not considered during similarity. After a similarity calculation, the results indicate a similarity index of 83% between molds 13.23 and 12.20. The planner should then adapt the process plan of the latter in order to prepare the dataset for scheduling the first.

Table 1. The hierarchical model of the mold-making production facility

Job-shop	Work-center	No. of Resources
Design	Design	2
Milling	Roughing	14
	Grinding	5
	Air & Water Circuit Cutting	3
	Tapping and Threading	4
	Finishing	6
EDM	Sinking	3
	Wire EDM	5
	Drilling	1
Measuring	Measuring	1
Polishing	Polishing	1
Fitting	Fitting	1
Hardening	Hardening	2
Assembly	Assembly	1

Table 2. Attributes of the compared molds

Attributes	Mold 13.23	Mold 12.20	Mold 11.38
Number of cavities	6	6	4
Type of Hardening	Good	Very good	Good
Core Cap	No	No	No
Tamper Evident	No	No	No
Surface's Quality	Matte	Matte	Mirror
Number of components	10	13	12
Way of Injection	Cavity Side	Cavity Side	Cavity Side
Slides	No	No	No
Wall Thickness	0.6mm	0.6mm	0.7mm
Height	50mm	50mm	15mm
Width	60mm	70mm	200mm
Length	0	100mm	0

Table 3. Similarity measurements between the molds 13.23, 12.20 and 11.38

Attributes	Mold 13.23-12.20	Mold 13.23-11.38
Number of cavities	0.387298333	0.31622777
Type of Hardening	0	0.2236068
Core Cap	0.31622777	0.31622777
Tamper Evident	0.31622777	0.31622777
Surface's Quality	0.2236068	0
Number of basic components	0.18708287	0.2
Way of Injection	0.31622777	0.31622777
Slides	0.31622777	0.31622777
Wall Thickness	0.38729833	0.35355339
Height	0.2236068	0.12247449
Width	0.20412415	0.25819889
Length	–	–
Similarity Measure	8.28247155	7.50196978

The next most similar mold is 11.38, which is 75% similar with the 13.23. As indicated in Table 2, mold 13.23 differs from 12.20 in the attributes: "Type of hardening", "Number of basic components", and "Width". The components that are needed for manufacturing the 13.23 mold are less than the components of 12.20, so the planner should reuse the sequence of processes of 11.38 mold and observe that components, such as the bottom plates, are missing. Based on experience, the extra components are removed and the process sequence is customized for the new mold. The calculated similarity index between the three molds is shown in Table 3. Once the adaptation of the new case is complete, the scheduling algorithm generates and evaluates scheduling alternatives and their respective performance indicators. Fig. 5 depicts the schedules of the new order in the system (right) and an order that has a start date eight days earlier (left).

Fig. 5. Schedule visualisation for two orders

The tunable parameters of the ISA are defined using a Statistical Design of Experiments [29], which reduced the required number of experiments for determining the impact of tunable parameter on the cardinal preference of the decision-making process. The number of experiments was 25 and each tunable factor had five levels. By calculating the degrees of freedom of the three factors, a minimum number of 13 experiments was indicated to determine the value of each parameter. The Analysis of Means (ANOM) diagrams were created including the factors of each parameter and the utility value, according to which, the optimum value for each parameter were: MNA=100, DH=15 and SR=20. In order to benchmark the performance of the ISA, a comparison against a number of widely used dispatch rules is performed. The rules are: First In First Out (FIFO), Shortest Processing Time (SPT), Earliest Due Date (EDD), and Least Process Time (LPT) [30]. Each schedule is assessed with the mean values of the performance indicators: utilization, flowtime, and tardiness, which are given by the following formulas:

Tardiness $$MT(t_n) = \frac{1}{N^{comp}} \cdot \sum_{i=1}^{N^{comp}} \max(0, t_i^{comp} - t_i^{dd})$$

Flowtime $$MF(t_n) = \frac{1}{N^{comp}} \cdot \sum_{i=1}^{N^{comp}} (t_i^{comp} - t_i^{arr})$$

Utilization $$MU(t_n) = \frac{1}{N^{comp}} \cdot \sum_{i=1}^{N^{comp}} \left(\frac{t_i^{comp} - t_i^{start}}{t^{tot}}\right)$$

where: N^{comp} is the number of completed jobs up to time t_n, t_i^{comp} is the completion time of job i, t_i^{dd} is the due date of job i, t_i^{arr} is the arrival time of job i, t_i^{start} is the start time of job i, t^{tot} is the total operating time of the facility, and t_n is the time point at which all performance measures are calculated.

The diagrams of Figs. 6-8 reveal the superiority of the ISA. Still, in cases with specific optimization target, dispatch rules yielded high quality results. For instance, the EDD rule provided the best results in terms of flowtime and near zero tardiness.

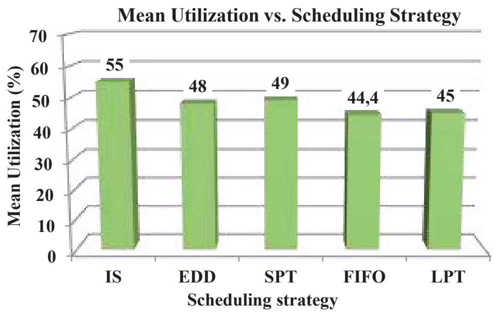

Fig. 6. Mean Utilization vs. Scheduling Strategy

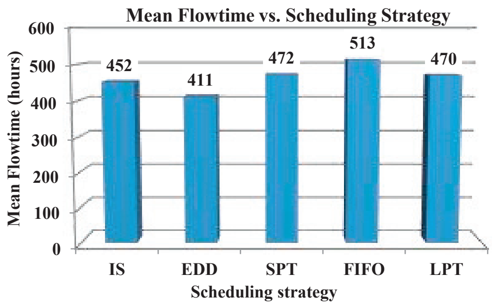

Fig. 7. Mean Flowtime vs. Scheduling Strategy

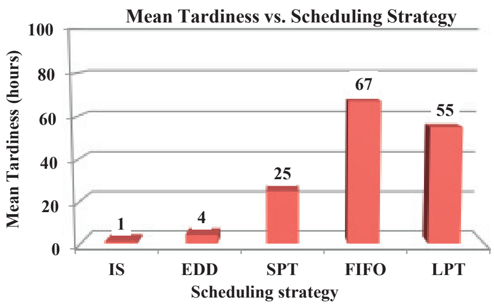

Fig. 8. Mean Tardiness vs. Scheduling Strategy

7. Conclusions and Future Work

The presented work focused on the short-term scheduling of manufacturing resources through the utilization of existing design and planning knowledge. The scheduling of tasks for the realization of engineer-to-order products is supported by a knowledge retrieval mechanism that is based on a Case-Based Reasoning procedure and similarity measurement. Both numerical and alphanumeric attributes are considered and the similarity between past and new cases is measured using the Euclidean distance. The scheduling is performed using an intelligent search algorithm that uses adjustable parameters that guide the search towards areas of high performance and configure its depth and breadth.

The results of the application of the methodology into a real-life pilot case with data obtained from the mold making industry verified that the short-term scheduling algorithm provides solutions of high quality in comparison to the historical values. The quality of the results indicate the usefulness of the engine for supporting the short-term scheduling of the manufacturing shop-floor. Moreover, the offered mobility, which is valuable for the dynamic nature of today's turbulent manufacturing environment, is achieved by the deployment of the scheduling engine on mobile devices.

A limitation of the proposed knowledge reuse approach is the necessity for pre-existing, sufficiently documented cases. The repository of past cases in the examined case study included 30 cases with 13 attributes each, and provided good results. The performance of the method in case of fewer cases with partial documentation, is expected to be lower. Yet, the gathering of this amount of information about previous cases is relatively easy, since these 13 attributes comprise basic characteristics of a mold, well-known to the planner, and a repository with 30 products can be built in a fairly short amount of time.

Future work will focus on the quantitative evaluation of the knowledge reuse and scheduling mechanisms. The company of the case study is currently testing the engine in real-life situations. The relative improvement of important KPIs (machine utilization, tardiness, etc.) for the SME will be reported in future work. Moreover, a series of interviews with the engineers will be organized to assess the quality of the produced schedules and the accuracy of the similarity measurement results. A long-term vision is the total integration of these mechanisms in the everyday practice of the company and their utilization through the developed app.

Acknowledgements

The work reported in this paper has been partially supported by the EC FP7 project "Applications for advanced Manufacturing Engineering – Apps4aME" (GA No: 314156).

References

[1] Chryssolouris G, Mavrikios D, Papakostas N, Mourtzis D, Michalos G, Georgoulias K. Digital manufacturing: history, perspectives, and outlook. Proc of the Inst of Mech Eng, Part B: J of Eng Manuf 2009:;223:451-462.
[2] Karmarkar US. Manufacturing lead times, order release and capacity loading. Handbook in Oper Resand Mgmt Sci 1993;4:287-329.
[3] Lowe A, McMahon C, Culley S. Information access, storage and use by engineering designers. J Eng Des 2004;30(2):30-32.
[4] Efthymiou K, Sipsas K, Mourtzis D, Chryssolouris G. On an integrated knowledge based framework for manufacturing systems early design phase. Procedia CIRPe2013 2013;9:121-126.
[5] Chryssolouris G, Subramaniam V. Dynamic scheduling of manufacturing job shops using genetic algorithms, J of Int Manuf 2001;12:281-293.
[6] Mourtzis D., Doukas M., Psarommatis F., Design of manufacturing networks for mass customisation using an intelligent search method", Int J

of Com Int Manuf, Published online: 24-03-2014, DOI: http://dx.doi.org/10.1080/0951192X.2014.900867
[7] Mourtzis D, Doukas M, Efthymiou K, Fragou K. Knowledge-based estimation of manufacturing lead time for complex engineer-to-order products, Procedia CIRP 2014;8; In Press - Corrected Proof.
[8] Aamodt A, Plaza E. Case-Based Reasoning: Foundational Issues, Methodlogical Variations, & System Approaches. AI Com 1994;7:39-59.
[9] Kolodner JL. An Introduction to CBR, AI Rev 1992;6:3-34.
[10] Michalos G, Makris S, Mourtzis D. A web based tool for dynamic job rotation scheduling using multiple criteria. CIRP Annals - Manuf Tech 2011;60(1):453-456.
[11] Wang JF, Du BQ, Ding HM. A Genetic Algorithm for the Flexible Job-Shop Scheduling Problem, Com in Comp and Inf Sci 2011;152:332-339.
[12] Brucker P. Scheduling Algorithms. 5th ed New York, Springer; 2007.
[13] Xing LN, Chen YW, Yang KW. An efficient search method for multi-objective flexible job shop scheduling problems, J of Int Manuf 2009;20:283-293.
[14] Li X, Olafsson S. Discovering dispatching rules using data mining, J of Sched 2005;8:515-527.
[15] Varela R, Vela CR, Puente J, Gomez A. A knowledge-based evolutionary strategy for scheduling problems with bottlenecks, Eur J Oper Res 2003;145:57-71.
[16] Sha D.H., Liu C.H., Using Data Mining for Due Date Assignment in a Dynamic Job Shop Environment, The International Journal of Advanced Manufacturing Technology 25, (2005) 1164-1174.
[17] Accenture. Mobile Web Watch Survey; 2012.
[18] Coldewey D. Mobile app use doubles, social apps triple in 2013: Report, NBC News, URL http://www.nbcnews.com/technology/mobile-app-use-doubles-social-apps-triple-2013-report-2D11917954; 2013.
[19] Fernandez-Pacheco DG, Molina-Martvnez JM, Ruiz-Canales A, Jimenez M. A new mobile application for maintenance tasks in photovoltaic installations by using GPS data, Ener Conver and Mgmt 2012;57:79-85.
[20] Mourtzis D, Doukas M, Vandera C, Mobile applications for product customisation and design of manufacturing networks. Manuf Lets, Elsevier, DOI: http://dx.doi.org/10.1016/j.mfglet.2014.01.002; 2014.
[21] Young HA. Scientific apps are here (and more will be coming). Cytokine 2012;59:1-2.
[22] Mobile ''apps'' provide parts data and calculate O-ring dimensions. Sealing Technology 2012;2:1-3.
[23] Gröger C, Silcher S, Westkämper E, Mitschang B. Leveraging apps in manufacturing. A Framework for App Technology in the Enterprise, Procedia CIRP 2013;7:664-669.
[24] Lunani M. Role-based mobile applications can help manufacturing companies dramatically boost productivity and growth., URL: http://www.cognizant.com/insights/perspectives/role-based-mobile-apps-transform-manufacturing; Accessed online on: 08-08-2013.
[25] Chryssolouris G, Dicke K, Lee M. On the Resources Allocation Problem. Int J of Prod Res 1992;30(12):2773-2795.
[26] Baker A., A survey of factory control algorithms which can be implemented in a multi-agent hierarchy: dispatching, scheduling, and pull, Journal of Manufacturing Systems, Vol. 17, No. 4, 1998, pp. 297-320.
[27] Lalas C, Mourtzis D, Papakostas N, Chryssolouris G, A Simulation-Based Hybrid Backwards Scheduling Framework for Manufacturing Systems, Int J of Comp Int Manuf 2006;19(8):762-774.
[28] Mourtzis D, Doukas M, Psarommatis F. A Multi-criteria Evaluation of Centralized and Decentralized Production Networks in a Highly Customer- Driven Environment. CIRP Annals – Manuf Tech 2012;61(1):427-430.
[29] Phadke MS. Quality Engineering using Robust Design, 1st ed, New Jersey, Englewood Cliffs: Prentice Hall; 1989
[30] Chryssolouris G. Manufacturing Systems: Theory and Practice, 2nd ed, Springer Verlag, New York, 2006.