

Automated Processing of Planning Modules in Factory Planning by Means of Constraint Solving Using the Example of Production Segmentation



Julian Graefenstein, David Scholz, Oliver Seifert, Jan Winkels, Michael Henke, and Jakob Rehof

Abstract For the adaption of factories, essential data are required as a basis in factory planning. Often these data are either stored in some form, at some location, or on some data medium, respectively, or are not available at all. Preparing these data for the planning process in a planning-appropriate manner can result in high effort. In order to counteract this situation, a data warehouse system can be used in the context of Business Intelligence for initially providing the data in a centralized and consistent form. The advantages of an up-to-date and consistent data base are shown by an example of the production segmentation. With the planning of the factory adaption by means of planning modules, which can be orchestrated individually, it is possible to process planning tasks automatically or partly automated. A given example of a vice production, which can be produced in four variants, was used to show the benefits and explain the approach in detail. Constraint solving, the modular planning process and the data available in the data warehouse enable the segmentation to be processed automatically and thus reduce planning time.

Keywords Factory planning · Planning modules · Constraint solving · Production segmentation · Data warehouse

1 Introduction

Due to emerging and ever-growing circumstances of rapidly changing environment of producing industry, companies have to face the important challenges of dynamic and complexity in order to remain competitive. In particular, shortened and more

J. Graefenstein (✉) · D. Scholz · M. Henke
TU Dortmund, Chair of Enterprise Logistics, Dortmund, Germany
e-mail: graefenstein@lfo.tu-dortmund.de

O. Seifert · J. Winkels (✉) · J. Rehof
TU Dortmund, Chair of Software Engineering, Dortmund, Germany
e-mail: jan.winkels@tu-dortmund.de

dynamic product life cycles force companies to fundamentally rethink and optimize the reactivity and adaptability of companies and their production systems [1, 2]. Current trends, such as Industry 4.0 and cyberphysical systems, Internet of Things, or digital shadows of factories itself, suggest solutions to problems and challenges caused by massive changes in companies' environment. The interconnection and broad communication not only between people within a production system but also between people and machines enable manufacturers to compete with the aforementioned challenges and allow a variable production focused on variable products up to lot size one which is mandatory for mass customization [3]. Technologies which can share and process information and interact with the physical reality are called cyberphysical systems. An example is an automated guided vehicle (AGV) which can transport material from an individual place to where it is needed. A worker signals the system that he needs a specific item and an AGV will get the item to the worker in the shortest way and time [4]. As every thing is talking to each other in the meaning of Internet of Things, the AGV knows where it can find the item which is the nearest and then navigates by itself to the item and the worker. The problem with these solutions of Industry 4.0 is that they are not directly usable in existing production systems and can only be used out-of-the-box in special cases and therefore require a considerable effort in implementation [5–8, 9]. Especially regarding the preparation for the implementation of Industry 4.0 technologies, many companies are still at the beginning [8, 10]. If such technologies are used, the preliminary step that needs to be completed first in order to cope with the increasingly relevant challenges is to plan the necessary changes or to adapt the own production.

The usage and therefore the necessary implementation of such technologies is one aspect to make mass customization a reality. Before taking the second step, using technologies like cps and others, one has to make the first step – planning the system. If there is no production system, which is capable of competing with the aforementioned challenges in general or no production system at all, the usage of new technology is irrelevant. In order to prepare existing production systems to compete with the new challenges and satisfy the customer requirements, it has to be optimized and adapted for mass customization production. Due to rapidly changing customer needs and therefore shortened product life cycles, changes of the production system have to be made in short amount of time.

In the context of the adaptation of existing factories, factory planning plays a decisive role. To cope with the complexity, the planning process itself has to be shortened and optimized, respectively, by the use of new planning methods. This has to be the goal for the near future in order to be as adaptable as possible on a cost-effective basis. Already through the reorganization or adaptation of existing production systems, inherent potentials in the system can be used to meet cost-effectively the challenges. Therefore, new investments for Industry 4.0 technologies may be avoided in the first place, and the existing resources can be used more efficiently by reorganizing them.

This paper addresses the challenges in adapting the existing factory and production system to new requirements and a new approach to optimize the planning

process for this adaptation itself. To handle complexity, a specific topic was chosen to demonstrate the possibilities of the new approach. With the production segmentation as one aspect of factory planning and in more detail the parts family formation, the new approach of a modularized planning process is shown. This paper first shows the state of the art of factory planning processes and approaches. It is shown what problems classical approaches have to compete with and why information is one of the most important resources in planning. Second the paper presents the new approach of modularized planning with the support of constraint solving. In the third part, the specific example of parts family formation is presented to introduce the fourth part in which the technical computation will be explained. As a fifth and last part, the following research question will be answered, and an outlook will close the paper.

1. How can structured information improve planning?
2. Can parts of a factory planning process be automated, executed, and calculated?

2 State of the Art

As already mentioned, shorter and more dynamic product life cycles require an increased reactivity for an adaptation of existing systems. Traditional and established factory planning models and concepts are not geared up for the new challenges and are not dynamic in their structure but are more static through the sequential processing of specific planning phases [11–15]. In order to ensure the responsiveness for efficient adaptation planning, more dynamically shaped models are required, which can be orchestrated according to the individual planning case and reduce the planning time to the best possible rate. Some more recent approaches focus on a modularized and standardized process design, in contrast with the traditional, sequential models, and allow for individually orchestrated planning process depending on the planning situation [16–18]. The modular design of planning tasks enables a defined and standardized processing of individual planning processes, which have a certain degree of individuality because of their defined interfaces and input and output information in their arrangement in the planning process [18]. The challenge in planning also includes the optimal combination problem of individual planning modules to optimize the planning time most efficiently and to guarantee the highest degree of responsiveness.

The basis for planning are the data of the company which are necessary for processing the respective planning task, not taking into account if a modular and dynamic or a sequential and phase-oriented structure is used [13]. The necessity of a consistent database in the context of factory planning was already stated in the last decade and created the vision of a concept of the digital factory [19–22]. The efficient use of methods and tools of the digital factory is only possible if a consistent

database is also available. Theoretical models, such as VDI guideline 4499 “digital factory,” presuppose a consistent database in which all necessary or existing company data are prepared in a data structure – first, to make existing knowledge available to each participant at any time and second, to make the planning itself more efficient. This includes the “standardization of planning processes and their results,” the “reduction of planning costs and times through the reuse of models and sub-results,” the “increase of planning security through rapid analysis and evaluation of possible planning alternatives,” and the “elaboration and reuse of best-practice solutions with regard to models and results” [23]. In contrast to this ideal status of a consistent database, reality looks different. Often the company data about products, processes, and general corporate knowledge are found in numerous file versions of various dates in different formats on a non-specified server and are seldom used. Only when a project starts, the required data for the actual planning are defined and have to be processed first, in order to use them properly. In the worst case, this effort has to be made again for the following project because the data was not kept updated during the time between the first processing initiated by the project and the start of the second project. With regard to Industry 4.0, this lack of data quality and data availability is an obstacle to the use of the associated technologies. The use of cyberphysical systems requires an information and communication system, which is able to process information in real time [24–26]. This necessary prerequisite also offers the possibility to use not only cyberphysical systems but also to profit from prepared and directly available data during adaptation planning. In contrast to that, a completely implemented and, in the context of Industry 4.0, functioning data infrastructure can open up new possibilities. New kinds of data, generated in real time by several cyberphysical objects and other sensor-equipped and communicating technologies, can be used to support the planning process.

An updated and consistent database is therefore necessary for efficient processing of a planning project or adaptation planning. A possible infrastructure for fulfilling these requirements is provided by so-called data warehouse systems. These systems store large amounts of data from different sources and formats and can provide them in a consistent form [27–30]. This offers the advantage of already existing data, for example, from different departments, without the need to be formatted to the same formats. In addition, large amounts of data collected over several years can be processed and made available by using a data warehouse system. A data warehouse is a tool of business intelligence, which is the umbrella term for a collection of other IT-based management tools, and has its origins in the 1990s of the last century [31–33]. The advantage of today’s data warehouses is that they can process a much larger amount of data compared to the time they were initially introduced, therefore also gaining more value in the context of factory planning. If the necessary basis for the planning is centrally provided and prepared at any time, new concepts of automated planning can be developed and used [32].

3 Constraint Solving in a Modularized Planning Approach

If the factory planning process is modular, individual planning tasks can be isolated and processed in parallel, considering the dependencies between the individual planning modules. Through the additional use of an algorithm in combination with the underlying data availability, realized by a data infrastructure, such as the already mentioned data warehouse, these planning tasks can be processed automatically or at least partly automatized. Therefore, different planning steps of the factory planning process can be defined as planning modules. These are characterized by a uniform structure which includes input and output information, global restrictions, and a module-specific process structure, which is characterized by processing of different planning tasks. As an example, the production segmentation as one possible task of factory planning provides a separate planning module within a pool of numerous additional modules (see Fig. 1) which can be orchestrated individually to a planning process according to the circumstances of the given planning project. This planning module for the production segmentation is standardized in general, which means that defined input and output information are given, as well as global restrictions. These input and output information and the global restrictions are

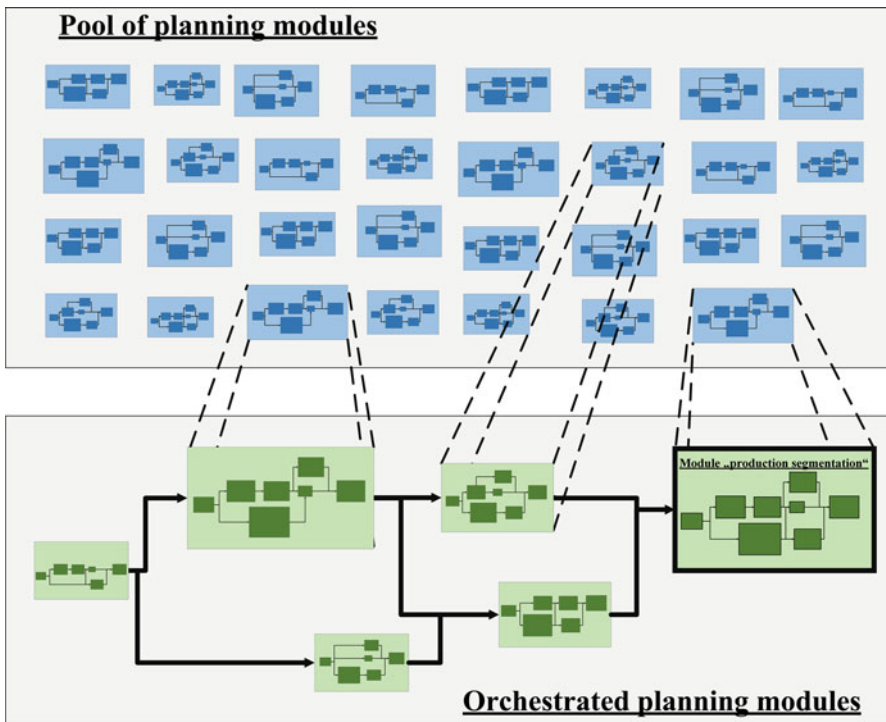


Fig. 1 Pool of planning modules

referred to as local or global constraints. Depending on the individual module, the defined sequence for processing the respective planning modules can be split into further modules whereby each processing step of the defined sequence can be a separate submodule.

Therefore, the orchestration of the individual planning modules can be carried out very finely granular, which allows the highest possible degree of parallelization resulting in a reduction of planning time. Since the necessary input information for processing a planning module is defined and known, this information is retrieved from a specific database (e.g., from a data warehouse) and can be used for processing the planning module. If this input information, which under ideal circumstances already exists, leads to the desired goal of the processing, the planning system can perform this processing automatically. If, however, information is missing or found insufficient, it must be generated by the planner or entered manually into the system. The planning system supports this interference by the planner and the generation of the missing information. Since it is known, which aim is targeted, in this case generating missing or insufficient information, the system can show the planner an ideal way to generate this information by orchestrating the necessary modules. This means that the entire planning can be processed in a partially automated manner. All planning modules, in which the information is sufficiently available, can be edited by the planning system itself to some extent. Only missing information is generated by the designer himself. That way, the planner assists the planning system during the process and even receives support to generate the missing information by the system itself. These interventions by the planner are considered as creative planning tasks, which cannot be processed by the planning system. Therefore the role of the user, in this specific case the planner, is still very important and cannot be substituted by a planning system. As a conclusion, the mentioned planning system functions as an assisting tool for the planner and vice versa.

4 Scenario: Parts Family Formation and Cluster Analysis as Part of Production Segmentation

Being an important component of the structural design, production segmentation is intended as a specific example of the modular planning system presented. Based on established methods of vertical production segmentation according to Wildemann, parts family formation and cluster analysis are done for segmentation using a fictitious scenario. This scenario addresses the optimization of the production of a vise using segmentation. By segmenting the production into smaller, more manageable units, it is intended to reduce throughput times and inventory and to increase the flexibility [15, 34, 35]. The focused product of the vise includes four different variants, which shall be produced using a fixed production program. For enabling various calculations, the applied information and data should be efficiently integrated as input information for the module “production segmentation.” For general determination of respective segments, certain information including the plan

of action, the production program, and parts lists is required. This data can be used for an automated calculation of parts family formation and cluster analysis with the help of algorithms described by Debnar, Bruestle, and Kosturiak [36]. The selection of suitable algorithms can be characterized as a creative, manual task and cannot entirely be automated [37].

This modular planning system addresses the task of selecting suitable algorithms and involves the user as support for the decision. Subsequently, the modular planning system and the underlying framework can perform the calculation of the respective output information, which finally comprises the respective parts family formation automatically. The aim of this approach in this scenario is to obtain an ideal segmentation of the production by means of certain restrictions or global constraints. In this case, the fixed shift schedules represent a global constraint. This means that the planning system has to make a segmentation, which is aligned according to the existing shift schedules. Depending on the shift schedule – either a one-, a two-, or a three-shift plan – the parts family formation and in conclusion the segmentation are performed accordingly. Taking into account the available personnel and the additional consideration of the production program, appropriate segments can be created. Furthermore, additional costs for night shift allowances and other aspects can be taken into account. Also conceivable are other global constraints, such as the available area or the available budget for possible purchases of machines etc.

5 Constraint Solving Basics

In order to be able to orchestrate individual modules to a processing sequence, the technique of constraint solving as well as combinatorial logic is used. In this section the concept of the constraint solving problem (or constraint satisfaction problem, CSP) will be explained. The aim of this part is to find a solution for a specific problem, which meets all the requirements of the given question.

As a rule, the solution represents a concrete assignment of target variables. The term “constraint” refers to the conditions and restrictions that apply to these variables. Those can be predefined value ranges but also relationships to other variables. For example, the constraint set $\{x + y = 5, X > 1, y > 1, x > y\}$ is applied to the two target variables x and y . Using constraint solving methods, the target $x = 3$ and $y = 2$ can be calculated, which fulfills all given conditions. In this example this is the only solution.

In many cases however it is also conceivable that several solutions exist for a problem. In contrast to other heuristics or optimization methods, apart from the question whether the solution fulfills all constraints, no further statements about solution quality are made. It is not intended to find the best solution possible, but rather to determine whether a solution exists under the existing restrictions. If this is not the case, the constraints are contradictory. For the computer-assisted solution of this kind of tasks, software is already available. The constraint solver “Z3” of the Microsoft Research Group is used for this approach [38].

To use CSP techniques for the orchestration of individual planning modules, Z3 is embedded in a software architecture which implements recursive process. The structure of the architecture is described in the next chapter. The overall process works as follows: As already mentioned, each module has specific in- and outputs. That means that each module has specific preconditions that must be fulfilled in order to execute the module. By executing the module, several post-conditions (that might be preconditions for other modules) are created.

In order for a module to be processed, all necessary inputs of this module must be present. The presence of the data inputs can be modeled as a constraint and can be done either via a Boolean variable or via concrete metric values. If a module is to be executed, it can be checked via constraint solving whether this module is ready.

For the overall process which is to be orchestrated, a target variable is defined. After the specification of this target value, a module is determined which can realize this target value as output. The next step checks whether the inputs of this module modeled as constraint are present in the database and whether the module can generate a solution. If a solution cannot be found, it is determined by stepwise “removal” of individual constraints from the constraint quantity, which leads to the non-satisfiability of the problem. This means that an attempt is made to compute a solution while ignoring a constraint.

This makes it possible to derive which constraints constitute the non-satisfiability of the original problem with complete constraint set. This can be used to determine which input data of a module is missing in the database of the system. These missing data are then interpreted as the new target variables of the module, for which the procedure is repeated. Therefore, again suitable modules are determined, and the nondetachable input constraint of the original module is replaced by the input constraints of the newly determined module. Again, an attempt is made to find a solution under the now-expanded constraint set. The process runs until a solution has been found, i.e., all inputs are completely covered by the data base. This would provide an appropriate starting point for a processing cycle.

The module sequence which has been passed up to then corresponds to the reverse processing sequence. If the constraints of a module cannot be solved in the course of the procedure, this shows that no processing sequence can be generated from the existing information in the data base. In this case, no constraint quantity can be derived from the available modules, for which a solution can be generated, which then requires intervention by the user. This can either automatically remove constraints from the constraint set of the problem in order to generate a solution, or manually enter missing or unpredictable data.

6 Software Architecture

In order to implement the automated approach described in the previous chapter, a software architecture that supports the entire workflow has been developed. The architecture is based on the MVC concept (model / view / controller) established in software development. In this concept, the individual components are implemented

in separate layers strictly separated from each other. The lowest layer (model layer) contains the data with which the application works. The controller layer contains the actual program logic, and the view layer is used for the user interaction surfaces. The MVC paradigm allows a flexible program design, which ensures the reusability of the individual MVC modules and complex MVC components as well as a reduced overall complexity, especially in large applications. The following advantages are particularly apparent: The application logic is clearly separated from the data model and the user interactions (Separation of Concern), which allows to use and modify each software component independently. Existing systems can easily be expanded by adding new modules and MVC components.

We used that advantage by integrating several external and independent software components into the architecture. The complete architecture is shown schematically in Fig. 2. The information required for this application is provided in the already mentioned database and forms the data layer of the MVC concept. At this point, all available domain information is stored in the form of structured XML files. Within the XML files, the information is processed in machine-readable form and is provided with all information relevant to the orchestration process. Figure 3 illustrates an example of such a data block.

The XML files can be modified at any time and adapted to changing conditions. In order to differentiate between the various data, the database was divided into at least three different files. Each file contains different categories of information and constraints. These are parts lists, process sequences, and the production program.

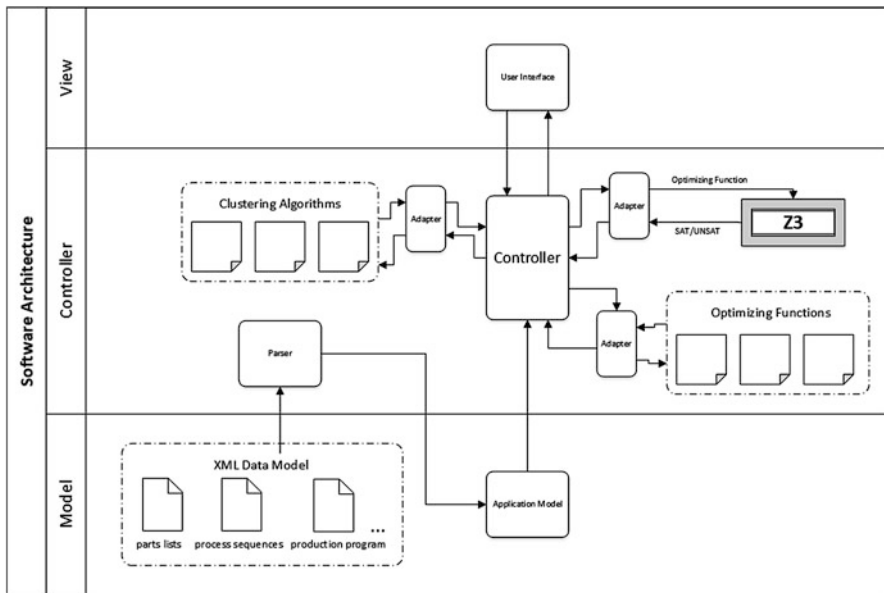
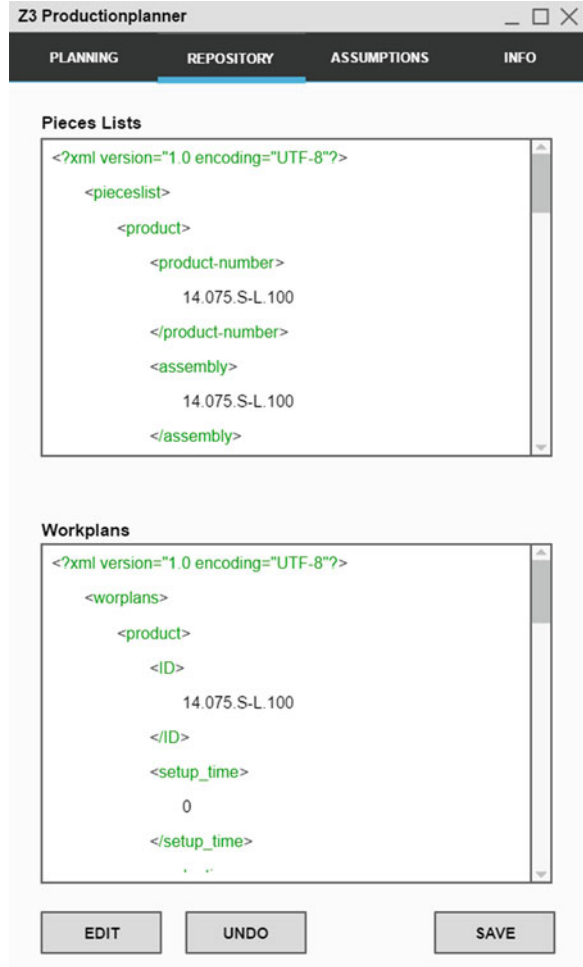


Fig. 2 Software Architecture Scheme

Fig. 3 Example of the XML-Data Model



The XML format offers the advantage that it can be read, written, and understood by both humans and machines. This allows planners to maintain data without the need for extensive IT knowledge and without the use of complex modeling software. At the same time, the format also can easily be generated automatically from other systems (data warehousing) [39].

The data provided in this way is processed by a parser and loaded into the application’s working memory. They are available in the form of a dynamic data structure. By means of a suitable clustering algorithm (see chapter “[Datamodels for PSS Development and Configuration: Existing Approaches and Future Research](#)”), a parts family formation is already carried out in the existing data. The individual clustering algorithms are modularly available in the controller layer and can be freely selected and exchanged. We implemented a universal adapter that allows us

to add and remove clustering algorithms dynamically as long they are implemented in JAVA. The user can later choose which algorithm is to be used. There is also an adapter available that provides the opportunity of adding several individual optimization functions to the software. These functions are used to calculate the desired target variable and are to be computed and validated by the Z3 constraint solver.

In the view layer, the user enters a desired calculation target and some of the selected conditions by means of a graphical user interface (GUI). The inputs are then recorded and processed by the controller. The controller then collects the needed data and the desired clustering algorithm and optimization function to create the quotations and functions that represents the desired calculation target and the constraints that exists in the given domain.

If the user wants to calculate, for example, the number of shifts required under a given number of machines, he specifies this in the interface. The number of layers is defined as the target variable to be calculated and the number of machines as a constant constraint. It is also possible to specify the number of machines as a limit value, which must not be exceeded. In this case, the actual number of machines is variable, and a concrete solution would then also be calculated. The software now builds a matching optimization function from the individual modular components. Let's assume the user wants to use a very simple optimization function like $\{x = (x_{M1} * a_{M1}) + (x_{M2} * a_{M2}) + \dots + (x_{Mn} * a_{Mn}); \min(x)\}$, where x_{Mi} = number of shifts required for a machine Mi and a_{Mi} is the number of machines of a certain type Mi , which are used in the given production program. While each a_{Mi} is known by the specified restrictions of the user, each variable x_{Mi} must be calculated using information from the database. As a result, the software substitutes, by means of knowledge from the data base, each variable to be calculated by a function, which provides the desired variable as a result. If this function also contains unknown (and still to be calculated) variables, the procedure for this variable is repeated. This results in a systematic complex optimization function and a set of several constraints that covers the entire problem area.

If a variable is found, for which there is no calculation basis for the database, the planner is informed that he must supply the relevant information. The already mentioned problem of the nonexistent data basis is partially solved in the current scenario (XML data available). Although there is information on parts lists, plans of actions, and shifts schedules, it does not contain the existing machines. Despite the fact that this information is available somewhere in the company, it is unfortunately not stored in a central database and processed for use. This missing information can imply additional, unnecessary effort to implement the data, which could have been avoided in advance by a consistent data management.

The step by step substitution of variables usually results in a complex function with several limitations and constraints regarding this function. The function and the constraint set are then sent to the constraint solver Z3. Z3 now calculates whether this formula can be satisfied and returns SAT for a satisfiable or UNSAT for an

Fig. 4 User Interface with validated result

The screenshot displays the 'Z3 Productionplanner' application window with a dark header bar containing the tabs: PLANNING, REPOSITORY, ASSUMPTIONS, and INFO. The 'PLANNING' tab is active. The interface is organized into nine numbered steps:

- Select Workplans**: 'Assembly Plans of the Products' with a 'Repository' button.
- Select Parts List**: 'Required Time an Costs of the Products' with a 'Repository' button.
- Define Production Period**: 'FROM' 01.01.2018 and 'TO' 31.12.2018.
- Define Type of Shift Model**: Radio buttons for '3-Shift-Model (24 h/day)' (selected) and '2-Shift-Model (16 h/day)'.
- Define Personnel Costs**: 'Hourly Rates' for 01.01.2018 EUR / h.
- Define Batch Sizes**: Four rows for different products, each with a '100' value in a text box and 'Pieces' label.

Required Amount of Product 14.075.S-L.100	100	Pieces
Required Amount of Product 14.075.D-L.100	100	Pieces
Required Amount of Product 14.100.S-L.100	100	Pieces
Required Amount of Product 14.100.D-L.100	100	Pieces
- Limiting Production Cost**: 'Material and Manufacturing Costs' 10000 EUR.
- Start Calculation**: An 'OK' button.
- See Results**: A checked checkbox and the text 'Production is possible under the given Constraints'.

unsatisfiable formula. If the formula can be fulfilled, a valid variable assignment of all variables including the target size is output (see Fig. 4). With this solution, the planner can then continue to work. If the formula is unfulfillable, this means that the desired calculation target cannot be realized under the given conditions.

7 Summary and Outlook

The described work illustrates that with the help of a consistent database and a specific software architecture, the aforementioned planning modules in the specific scenario can be processed automatically. In the described example the planning module “production segmentation” was used to clarify the functionality of several in- and output information of the planning module itself in combination with the available data and the computing software which answers the second research question. The lack of information is one of the main problems for the planning system. Only with the assistance of the user the planning system can work properly. That clarifies the circumstance that the user and the planner, respectively, play a decisive role in the presented system. As mentioned before, the data provided for the whole planning process is crucial and has to be provided in short time. Therefore, the usage of a data infrastructure as provided by a data warehouse would be suitable. The ideal case would be a data system that provides needed data in a consistent and prepared form in real time. This would mean that planning data is always available at any point, which could drastically reduce time of planning. It also answers the first research question which was formulated in the introduction. Structured information can optimize planning time and reduce cost in a very efficient way. Without the effort to gather data for every new project over and over again, it is possible to start with a project more quickly. Especially for the usage of the presented modular approach, a structured database like the aforementioned data warehouse is mandatory.

The presented planning system with the underlying software architecture is not only suitable for assisting and automatically processing the planning task “production segmentation.” It works for a wide range of different planning modules, which can be added to the system. Therefore, the software architecture is also built in a modular manner. Different modules for optimizing different aspects, like in these example algorithms for parts family formation and cluster analysis, can be added with less effort. The decision of whether to choose one or another optimization tool belongs to the planner. In the future development, the software is also to provide detailed feedback on the reasons for the nonfulfillment of a function in order to give the user recommendations for adjustments. Other data categories like 3D data of the machine pool or the factory building could also be considered to be added to and evaluated by the system. With this evaluation and the preparation of such data, respectively, in a form usable for the planning process, it could support reducing the planning time. With that knowledge, the user and the whole system can be developed to a further stage of precision.

Acknowledgment This article was written within the Research Training Group GRK 2193 “Adaption Intelligence of Factories in a Dynamic and Complex Environment” funded by the Deutsche Forschungsgemeinschaft (DFG).

References

1. Reinhart, G.: Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. München: Carl Hanser Verlag GmbH & Co. KG (2017). Online verfügbar unter <https://doi.org/10.3139/9783446449893>
2. Weyer, S., Schmitt, M., Ohmer, M., Gorecky, D.: Towards industry 4.0 – standardization as the crucial challenge for highly modular, multi-vendor production systems. IFAC-PapersOnLine. **48**(3), 579–584 (2015). <https://doi.org/10.1016/j.ifacol.2015.06.143>
3. Vogel-Heuser, B., Diedrich, C., Pantforder, D., Gohner, P.: Coupling heterogeneous production systems by a multi-agent based cyber-physical production system. In: 2014 12th IEEE International Conference on Industrial Informatics (INDIN). 2014 12th IEEE International Conference on Industrial Informatics (INDIN). Porto Alegre RS, Brazil, 27.07.2014–30.07.2014: IEEE, pp. 713–719 (2014)
4. Böckenkamp, A., Weichert, F., Stenzel, J., Lünsch, D.: Towards autonomously navigating and cooperating vehicles in cyber-physical production systems. In: Niggemann, O., Beyerer, J. (Hg.): Machine learning for cyber physical systems. Selected papers from the international Conference ML4CPS 2015. Berlin, Heidelberg: Springer Vieweg (Technologien für die intelligente Automation, Band 1), pp. 111–121 (2016)
5. Trenkle, A., Furmans, K.: Der Mensch als Teil von Industrie 4.0: Interaktionsmechanismen bei autonomen Materialflusssystemen. In: Birgit Vogel-Heuser, Thomas Bauernhansl und Michael ten Hompel (Hg.): Handbuch Industrie 4.0 Bd.3. Logistik. 2. Aufl. 2017. Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg (Springer Reference Technik), pp. 45–60 (2017)
6. Kalhoff, J.: Praxisbeispiel: Automatisierter Schaltschrankbau. In: Manzei, C., Schlepner, L., Heinze, R. (Hg.) Industrie 4.0 im internationalen Kontext. Kernkonzepte, Ergebnisse, Trends. 2., neu bearbeitete und erweiterte Auflage, pp. 211–214 (2017)
7. Hermann, M., Pentek, T., Otto, B.: Design Principles for Industries 4.0 Scenarios. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). 2016 49th Hawaii International Conference on System Sciences (HICSS). Koloa, HI, USA, 05.01.2016 – 08.01.2016: IEEE, pp. 3928–3937 (2016)
8. Abersfelder, S., Bogner, E., Heyder, A., Franke, J.: Application and validation of an existing industry 4.0 guideline for the development of specific recommendations for implementation. AMR. **1140**, 465–472 (2016). <https://doi.org/10.4028/www.scientific.net/AMR.1140.465>
9. Kagermann, H., Wahlster, W., Helbig, J.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group. acatech - National Academy of Science and Engineering. Frankfurt/Main. Online verfügbar unter https://doi.org/http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report_Industrie_4.0_accessible.pdf (2013)
10. Breyer-Mayländer, T.: Management 4.0 – den digitalen Wandel erfolgreich meistern. Das Kursbuch für Führungskräfte (2017)
11. Aggteleky, B.: Fabrikplanung. Werkentwicklung und Betriebsrationalisierung. 2., durchges. Aufl. der Neuausg. München: Hanser (1987)
12. Kettner, H., Schmidt, J., Greim, H.: Leitfaden der systematischen Fabrikplanung. Mit zahlreichen Checklisten. Unveränd. Nachdr. der Ausg. 1984. München: Hanser (2010)
13. Grundig, C.: Fabrikplanung. Planungssystematik – Methoden – Anwendungen. 5., aktualisierte Aufl. München: Hanser (2015)
14. VDI Richtlinie 5200, 2011: Fabrikplanung – Planungsvorgehen
15. Wiendahl, H.-P., Reichardt, J., Nyhuis, P.: Handbook Factory Planning and Design. Springer, Berlin/Heidelberg (2015)
16. Schuh, G., Franzkoch, B., Burggräf, P., Nöcker, J.C., Wesch-Potente, C.: Frei konfigurierbare Planungsprozesse in der Fabrikplanung. wt Werkstatttechnik online. **99**(4), 193–198 (2009)
17. Nöcker, J.: Zustandsbasierte Fabrikplanung. Zugl.: Aachen, Techn. Hochsch., Diss., 2012. 1. Aufl. Aachen: Apprimus-Verl. (Edition Wissenschaft Apprimus, 2012, 6) (2012)

18. Graefenstein, J., Scholz, D., Henke, M., Winkels, J., Rehof, J.: Intelligente Orchestrierung von Planungsprozessen. Anwendung von logikbasiertem Constraintsolving in der Fabrikplanung. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*. **112**(4), 209–213 (2017)
19. Dombrowski, U., Tiedemann, H., Bothe, T.: Visionen für die digitale Fabrik. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*. **96**(3), 96–100 (2001)
20. Bracht, U.: Ansätze und Methoden der Digitalen Fabrik. Simulation und Visualisierung. Magdeburg (2002)
21. Zäh, M.F., Patron, C., Fusch, T.: Die Digitale Fabrik. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*. **98**(3), 75–77 (2003). <https://doi.org/10.3139/104.100614>
22. Bracht, U., Masurat, T.: The digital factory between vision and reality. *Comput. Ind.* **56**(4), S.325–S.333 (2005). <https://doi.org/10.1016/j.compind.2005.01.008>
23. VDI Richtlinie 4499, 2008: Digitale Fabrik – Grundlagen
24. Siepmann, D.: Industrie 4.0 – Grundlagen und Gesamtzusammenhang. *Industrie 4.0 – Struktur und Historie*. In: Roth, A. (ed.) Einführung und Umsetzung von Industrie 4.0. Grundlagen, Vorgehensmodell und Use Cases aus der Praxis, pp. 17–34. Springer Gabler, Berlin/Heidelberg (2016)
25. Shrouf, F., Ordieres, J., Miragliotta, G.: Smart factories in Industry 4.0. A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In: 2014 IEEE International Conference on Industrial Engineering and Engineering Management. 2014 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). Selangor Darul Ehsan, Malaysia, 09.12.2014 – 12.12.2014: IEEE, pp. 697–701 (2014)
26. Zhong, R.Y., Dai, Q.Y., Qu, T., Hu, G.J., Huang, G.Q.: RFID-enabled real-time manufacturing execution system for mass-customization production. *Robot. Comput. Integr. Manuf.* **29**(2), 283–292 (2013). <https://doi.org/10.1016/j.rcim.2012.08.001>
27. Humm, B., Wietek, F.: Architektur von Data Warehouses und Business Intelligence Systemen. *Informatik Spektrum*. **28**(1), 3–14 (2005). <https://doi.org/10.1007/s00287-004-0450-5>
28. Navrade, F.: Strategische Planung mit Data-Warehouse-Systemen. Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler/GWV Fachverlage GmbH Wiesbaden (2008). Online verfügbar unter <https://doi.org/10.1007/978-3-8349-9762-3>
29. Schoenherr, T., Speier-Pero, C.: Data science, predictive analytics, and big data in supply chain management. Current state and future potential. *J. Bus. Logist.* **36**(1), 120–132 (2015). <https://doi.org/10.1111/jbl.12082>
30. Hu, H., Wen, Y., Chua, T.-S., Li, X.: Toward scalable systems for big data analytics. A technology tutorial. *IEEE Access*. **2**, 652–687 (2014). <https://doi.org/10.1109/ACCESS.2014.2332453>
31. Kemper, H., Baars, H., Mehanna, W.: Business Intelligence – Grundlagen und praktische Anwendungen. Eine Einführung in die IT-basierte Managementunterstützung. 3., überarbeitete und erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag/GWV Fachverlage GmbH Wiesbaden (Studium) (2010). Online verfügbar unter <https://doi.org/10.1007/978-3-8348-9727-5>
32. Hoffmann, M., Kreisköther, K., Büscher, C., Meisen, T., Kampker, A., Schilberg, D., Jeschke, S.: Optimized factory planning and process chain formation using virtual production intelligence. In: Jeschke, S., Isenhardt, I., Hees, F., Henning, K. (eds.) *Automation, Communication and Cybernetics in Science and Engineering 2013/2014*, pp. 881–895. Springer International Publishing, Cham (2014)
33. Larson, D., Chang, V.: A review and future direction of agile, business intelligence, analytics and data science. *Int. J. Inf. Manag.* **36**(5), 700–710 (2016). <https://doi.org/10.1016/j.ijinfomgt.2016.04.013>
34. Wildemann, H.: Die modulare Fabrik. Kundennahe Produktion durch Fertigungssegmentierung. 5., überarb. und erg. Aufl. München: TCW-Transfer-Centrum (TCW, 2) (1998)
35. von Garrel, J., Schenk, M., Seidel, H.: Flexibilisierung der Produktion. Maßnahmen und Status-Quo. In: Schlick, C.M., Moser, K., Schenk, M. (eds.) *Flexible Produktionskapazität innovativ managen. Handlungsempfehlungen für die flexible Gestaltung von Produktionssystemen in kleinen und mittleren Unternehmen*, pp. 81–126. Springer Berlin Heidelberg, Berlin/Heidelberg (2014)

36. Debnar, R., Bruestle, T., Kosturiak, J.: Fertigungssegmentierung durch Anwendung von Gruppentechnologie und Simulation. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb. **94**(10), (1999)
37. Beller, M.: Entwicklung eines prozessorientierten Vorgehensmodells zur Fabrikplanung. Zugl.: Dortmund, Techn. Univ., Diss., 2010. Dortmund: Verl. Praxiswissen (Fabrikorganisation) (2009)
38. de Moura, L., Bjørner, N.: Z3: An Efficient SMT Solver, International Conference on Tools and Algorithms for the Construction and Analysis of Systems – TACAS 2008: Tools and Algorithms for the Construction and Analysis of Systems, pp. 337–340
39. VDI/VDE Richtlinie 3690 Part 2: XML in der Automation – Überprüfung fachlicher Modelle nach XML