Smart Factory: System Logic of the Project EPIK

EPIK: Efficient Staff Assignment Through Intelligent and Adaptive Cooperation- and Information Management in the Area of Production

Cornelia Hipp, Thomas Sellner, Janina Bierkandt
Fraunhofer Institute for Industrial Engineering
Stuttgart, Germany
cornelia.hipp | thomas.sellner |
janina.bierkandt@iao.fraunhofer.de

Philipp Holtewert
Fraunhofer Institute Manufacturing Engineering and Automation
Stuttgart, Germany
philipp.holtewert@ipa.fraunhofer.de

Abstract **– The production process in manufacturing industries is getting more complex and demands dynamic solutions, which can react flexible on different events of the production environment (e.g., alerts). Mainly, the static production process could benefit in efficiency, if the product environment would be smart and would delegate recognized tasks intelligently to workers. This idea is the main topic of the research project EPIK. Within this paper, the project is introduced and the main system logic is discussed. A description is included how the context data of the production environment is analyzed, which conclusions are drawn and how a detected task is assigned to the most appropriate worker.**

Keywords: adaptive interfaces; smart production; intelligent environment; system logic; human machine interface; smart factory; information management

I. Introduction

The research project EPIK (effective staff assignment through intelligent and adaptive cooperation- and information management in the area of production – German original: „Effizienter Personaleinsatz durch intelligentes und adaptives Kooperations- und Informationsmanagement in der Produktion“) improves the production process with help of a smart environment and individual support of workers with help of mobile and adaptive devices.

EPIK collects context data of the production environment, e.g., which dysfunctions occur on which machine and which workers with necessary qualifications are close by. Subsequently, EPIK decides which worker should be ordered to solve the problem and contacts him on his mobile device. In the decision process, EPIK takes into account several aspects, like how much work does the individual worker already has or has the new task high-priority, that the worker should interrupt the current job. When selected, the worker gets individual information to solve the task. This takes place with an adaptive individual mobile device of the worker (compare Figure 1).

The aim of this system is to optimize the production process and to make the staff assignment more efficiently. This will be achieved through three aspects:

Optimal operating grade of resources regarding employees and reduction of non-productive working time: With usage of mobile devices for staff it is possible to make monitoring of the production process without being fixed to a place. A fixed assignment of staff to machine can be canceled (which is currently standard). Therefore, tasks of different machines can be delegated to different staff. Workers will be supported in work with guidance of a mobile device. As a consequence, non-productive time of workers will be reduced.

Accurately fitting of resource management trough intelligent delegation of work tasks to adequate workers: Before delegating the work task, EPIK calculates how much time different workers will approximately need to accomplish the task. Furthermore, the requested qualification will be matched to the deposited qualification of the worker. At last, it will be looked-up, when the worker conducted the same task the last time. EPIK takes these figures into account to calculate an accurate fitting of workers to open tasks.

Raising efficiency or individual job performance with help of individualized and personalized support for the worker:
The mobile device gives context relevant information to solve tasks. The device will display adaptively information to the individual needs.

Thiel et al. [1] point out that a more efficient working place is getting more and more important in western countries, since the demographic change is leading to a lack of qualified staff and that personnel costs have gotten a major amount of the total production cost. Therefore, EPIK addresses this problem thoroughly, since it makes an efficient assignment of tasks to workers and help workers to get continuously qualified through individual support of the mobile device.

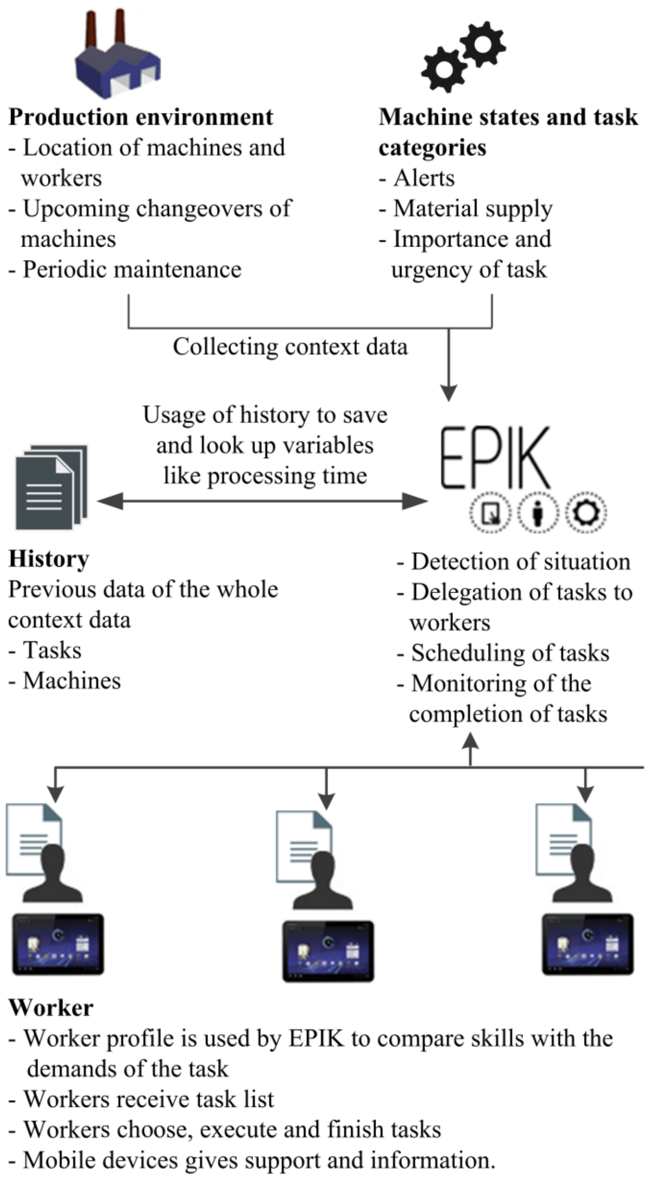

Figure 1. Overview how EPIK captures and analyses the production context with subsequent communication to the worker

II. Situation Detection and Overall EPIK-Algorithm

The aim of EPIK is enhancement of efficiency with help of an optimal operating grade and individual support of the single worker. This demands among other things a dynamic, adaptive user interface with high flexibility. According to Schlegel & Thiel et al. [2] central control systems and static user interfaces have been used for years and the demand for a more flexible possibility in the environment of production is very high. The Federal Government Department of Education and Research in Germany identified this problem as well and claims within a scientific development program [3] that the complexity within production has risen and that there is the necessity for a more flexible production process, which has the ability to observe information, deduce findings from them and to transfer this into change of performance.

EPIK takes this problem into account and collects therefore several context data of the environment, deduce situations from the context data, finds intelligent solutions and contacts the workers flexible on their mobile devices. To find these smart solutions, EPIK chooses the approach of situation detection.

The various context data are changing dynamically and permanently (e.g., the position of the worker) what allegorize high demands at the system. The approach of the situation detection and reaction of EPIK is based on the approach of Kluge [4]:

Scene analysis

Context data is captured continuously and can be from different kind. Context data can be positions (persons, objects, machines, etc.), states (machines, persons, etc.), and task categories. When EPIK detects a situation all context data will be refreshed.

Activity recognition

Subsequently, all captured attributes are put into relationship of each other. For example, moving workers are identified as “walking” due to changing context data. Partial solutions are calculated for the questions, how long does it approximately take each worker to solve the task, which worker is suitable regarding the needed qualification and which worker should practice and learn the task.

Planning

At this step a continuative interpretation of the situation proceeds. Regarding the calculated three lists in the previous step, EPIK makes a weighted comparison to find the best fitting worker to the task (see Section V Assignment of Tasks to Workers). The result of this is going into the process of the scheduling, where EPIK calculates at which position the task should be sorted in the worker-task-list (see chapter VI Scheduling). The scheduling regulates the order of the tasks for the worker and reacts, if the worker refuses the task and start again the selection process to find the next eligible worker. The user gets detailed information on the mobile device about the task.

The context data of EPIK is changing dynamically and is not static (and therefore, not foreseeable). As a result within *scene analysis* and *activity recognition* it is important to work with updated values, to ensure a best possible matching of workers to tasks. This dynamic environment of the system means high demands for the implementation, computing power and algorithms of EPIK. According with Schöning [5] the best solution for dynamic programming is to use a bottom-up-implementation. Schöning [5] writes that partial solutions should be found in advance and saved in a table. The total solution should refer to the partial solutions. This is picked up in EPIK with the previous described lists. The total overview of the overall EPIK algorithm is displayed in the following Figure 2.

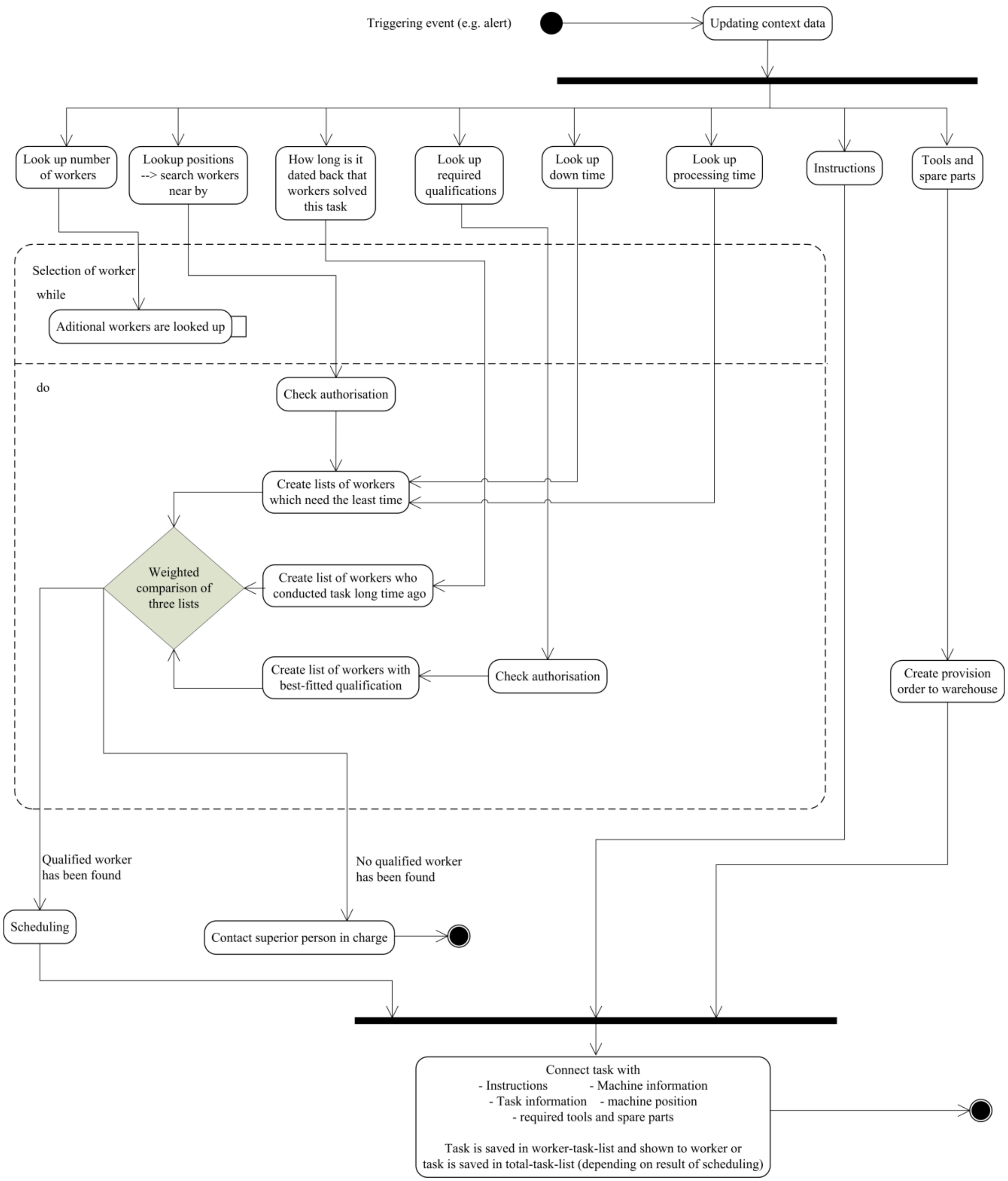

Figure 2. Overview of the overall EPIK algorithm

With the help of the algorithm the goal of reaching an optimal operating grade of resources will be achieved. Workers won't have non-productive working time any more, since the system monitors and detects non-productive working time and will give new tasks to the workers, which are not occupied.

Furthermore, the algorithm ensures an accurately fitting of resource management trough intelligent delegation of work tasks to adequate workers. This is the second approach of the EPIK-project to enhanced efficiency within the industry.

III. System Architecture

To get a better understanding about the system architecture, Figure 3 shows the different system components and their communication ways. The figure demonstrates how the EPIK System is embedded in an IT infrastructure of a production hall and how the System gets the context data, e.g., machine and employee. Especially the machine states are gathered through connecting to the Manufacturing Execution System (MES). Besides the machine conditions the following information are identified as context data. (1) Current values and availability of items in the storehouse are saved in a storehouse items profile. (2) After the situation detection (cf. Section II, Situation detection and overall EPIK-Algorithm) concrete task profiles are generated. (3) At least, every employee is represented by an employee profile that contains several information like the task history, his or her qualification profile, allocated machines, and the current assigned task. To cluster the different responsibilities and tasks of the individual components several managers are implemented. Each of them is modularity implemented so that they can be adapted and extended to other industrial user needs. In the following the certain managers and their tasks are described:

Context Manager: The whole context data that are described above are collected, interpreted and updated by the Context Manager. A further task of the Context Manager is to combine the context data to a concrete situation and chooses the best qualified employee by taking into consideration of the situation parameters. After choosing the best qualified employee the context Manager triggers the Message-Manager that forwards the generated Task to the assigned employee. For that reason this manager plays a key role in the EPIK System.

Recognition Manager: Before the Context Manager interprets the context data, the Recognition Manager monitors the complete machine states by connecting to the MES-Interface. Information are examined and rejected if they are irrelevant for the EPIK-System. In contrast relevant information are gathered and used to generate Task EPIK objects which are forwarded to the Context Manager that completes this object with situation parameters, e.g., from the task profile.

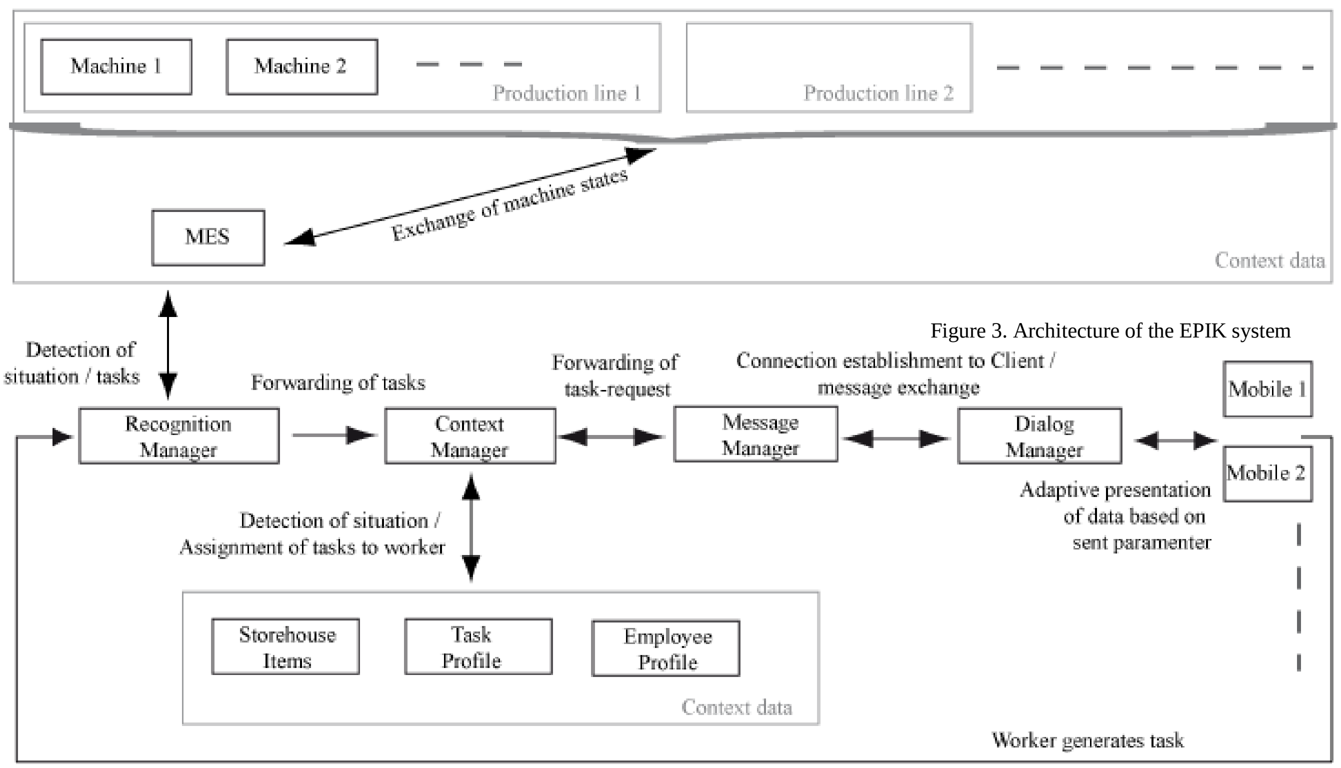

Figure 3. Architecture of the EPIK system

Message Manager: For forwarding the task request and to enable a back channel for the employee a Message Manager is implemented. This manager uses the Extensible Messaging and Presence Protocol (XMPP) for sending and receiving Message. We defined constants that are attached in the subject of the message in order to interpret the intent of the message and thereupon execute the certain EPIK function.

Dialog Manager: Besides improving the production process with individual support of workers with help of

 ISBN: 978-1-61208-225-7

mobile devices a further benefit of the EPIK-System are the adaptive user interfaces. Therefore, a Dialog Manager that is implemented on the mobile devices is responsible for best fitting information representation to the user.

IV. Complexitiy, Urgency and Importance of Tasks

During the preparation for the system logic, it became apparent that the following three attributes are necessary for an efficient assignment of tasks to staff:

The **complexity** of tasks describes the range of difficulty to solve the task. Simple actions with low complexity can be accomplished by unskilled workers. This parameter is important for EPIK to appraise, which worker is able to solve the task and how long will the process time approximately be.

Importance differentiates between important tasks and unimportant ones.

Urgency is characterized by time factor. Tasks which have to be done immediately are very urgent.

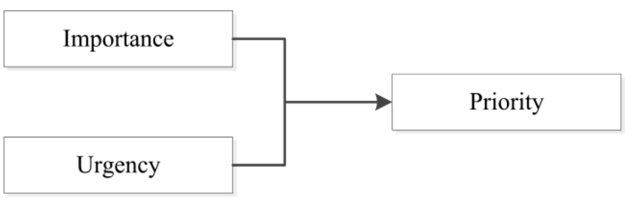

Figure 4. Correlation between importance, urgency and priority

Within this classification, importance and urgency determine when a task has to be done, and in which sequence the several tasks has to be worked on. These two attributes are combined to the **priority**:

- The priority is highest, when the task is urgent and important. These tasks have to be done immediately and other tasks can be interrupted, if they have a low priority.
- Tasks with high importance but low urgency has to be done ongoing and continuously. EPIK is scheduling these tasks ongoing, when there is not a very high priority task.
- High urgency but low importance means that staff with low qualification can do this work immediately.
- Tasks, which are neither important nor urgent have the lowest priority and are scheduled at the end of the tasks.

In this Project, a default set of situations with the associated attributes are specified and declared. For example, an alert is specified as highly important and urgent whereas an upcoming scheduled maintenance is determined as a lower prioritized task. In addition, EPIK provides the functionality for supervisors to parameterize the classification of situations.

The classified tasks are scheduled in the entire task-list of EPIK, which is described more precisely in Section *VII Scheduling*.

V. Assignment of Tasks to Workers

To enhance the efficiency in the production process, a sub-goal of EPIK is to find accurately fitting staff to solve the tasks. To solve this goal, EPIK tries to find the best person for the task and takes different aspects into account:

For the assignment, information of the worker is needed. EPIK collects implicit and explicit data of the worker. For example after the successful completion of an assigned task, EPIK registers the needed time of the worker and increases his qualification value.

Figure 5 displays the selection process for the staff assignment.

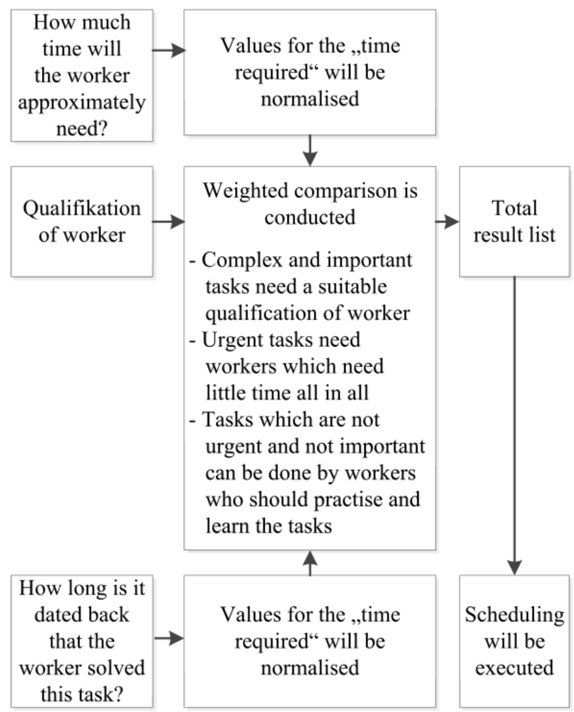

Figure 5. EPIK creates three partial solution lists with workers. Subsequently a weighted comparison is conducted.

The first step is to create three sorted lists which are ordered to the following criteria:

How much time will the worker approximately need? Data of previous tasks from the history are consulted as well as the position of the worker in relation to the location of the machine and task.

The **qualification of the workers** will be compared to the necessary qualification to solve the task. EPIK looks therefore in the profile of the worker and additional in the history of the tasks. The reason for the look in the history is that the workers are increasing their qualification ongoing while using EPIK. Therefore, it is possible that EPIK allocates a task to a worker without the correct qualification, so that the worker enhances his expertise level.

How long is it dated back that the worker solved this type of task? (If the worker never finished this task, the longest duration of the other workers will be used).

The reason to look up the last work on the task is the idea to raise all workers to a similar expertise level. This is to back up redundantly the know how within the company. Therefore, workers with an old date can be preferred within the assignment of tasks to workers. The positive effect is that the workers keep in practice and learned things will not be forgotten immediately.

The final choice of worker is done due to a weighted comparison of the described three lists. The exact weighting can be adjusted individually due to the needs of the company. The weighting factor is differing between the varying tasks.

For very urgent tasks, the factor time is weighted strongly. In contrast therefore, the qualification of the worker is highly weighted, when the task is very important. For little complex and tasks not-important tasks, the thought of training has the dominant weighting. An overview of the weightage is displayed in Figure 6.

Task	Qualification of worker	Approximately required time of worker (normalised)	How long is it dated back that worker solved task (normalised)
High importance	High weighted factor		
High complexity	High weighted factor		
High urgency		High weighted factor	
Low importance			High weighted factor
Low complexity			High weighted factor
	Value worker qualification	Value worker needed time	Value of last task solving
		Result: Qualification + Needet time + Last task solving	

Figure 6. Weighting factors used in EPIK

VI. Scheduling

New tasks will be delegated from EPIK to the best possible worker, even if the worker is still working on another task. Therefore, each worker gets a list of tasks by the time. To ensure that not one worker is always chosen from EPIK (and gets an enormous long list of tasks) and other workers are not contacted at all, EPIK controls by each task assignment the length of the task list of the chosen worker.

Additionally, it is been checked whether the new task has a high priority and where the task should be sorted in within the task-list of the worker. This sorting and allocation of the tasks after priority and length of task-lists is called *scheduling* in EPIK.

The scheduling is working with two different types of task lists:

The **total-task-list** of the whole EPIK system saves all tasks within the production process, which are known already on a stack.

- Tasks, which are already allocated to workers, are marked as allocated and the worker-ID is saved.
- The task list is sorted according to the priority (compare *chapter III complexity, urgency and importance of tasks*). After a defined interval, the total task list is calculated and sorted again. The reason therefore is to avoid that tasks with a low priority won't be done at all, because they would stay at the last position. With passing time, the tasks are getting higher in the priority.
- Is a task completed, it will be deleted from the total task list and will be saved within the history.

The second list is the **worker-task-list**, which are the individual tasks of the specific worker.

- New tasks are sorted in according to the priority.
- The worker-task-list has a maximum capacity of five tasks.
- Is the length of the list longer than five after sorting in the new task, the last task of the list will be removed from the individual worker-task-list. This task will be marked within the total-task list as not allocated.
  One special case is, when the worker has only high-priority tasks. In this case, the last task will be removed from his list and will be allocated directly to another worker.

The priority is very important for the scheduling, since it is the parameter for the order of the tasks. The priority is very important as well to identify high-priority tasks, which has to be done immediately. To identify these, a threshold is used within EPIK. If the priority of a task is above the threshold, EPIK grabs the attention of the worker to ensure, that he got to know about the task. Furthermore, EPIK controls that the task will be conducted promptly.

VII. Conclusion

The Production process is under continuous change [3] and therefore has to be flexible and dynamic. The competition pressure of internationalization demands efficient systems to afford staff costs in western countries any more. EPIK demonstrates solutions for these needs, but can show only the start of the improvement. Questions like “how can the user-interface support efficiently for often changing international workers” are left by and should be studied more thoroughly. Aspects of staff motivation can be addressed as outlook as well. This means, how can the user interface assist the worker to keep motivated on the job? This question refers a lot to ongoing research projects regarding the user experience of a user interfaces. Up to now, in the production environment, the focus is still based on the usability of the user interface, with the question, “how can the interface assist the user to work faster”. The idea of, that the user interface could have such a good user experience, that it makes the user more motivated or happier is in the context of production seldom referred. Palviainen & Väänänen-Vainio-Mattila [6] is one of these references and describes that there is great potential regarding user experience (UX) in the automation industry. Palviainen & Väänänen-Vainio-Mattila point out that “… there are several reasons to assume that the UX paradigm will influence the machinery industry. The level of automation is increasing and the nature of work is shifting from monotonous, low level process control towards expert and team work.” Therefore it would be very interesting to make thorough studies to find out how can the user experience affect the efficiency of production.

VIII. Acknowledgments

The project EPIK is financed by the Baden-Württemberg Stiftung gGmbH in the frame of the innovation and research program "Effiziente Produktion durch IKT".

References

[1] Thiel, S. Pfannstiel, S. and Bierkandt, J (2011), “Adaptive Cooperation and Information Management for Efficient Deployment in Production”, 21st international Conference on Production Research, Stuttgart, Germany, 5pp.,references without page number.

[2] Schlegel, T., and Thiel, S. (2010). „Interaction generation for decentralized process execution”. (D. Pham, E. Eldukhri, & A. Soroka, Hrsg.) Dunbeath, Scotland, UK: Whittles Publishing.

[3] Scientific development programm “Research for the Production of Tommorow” of the Federal Government Department of Education and Research in Germany 2012: http://www.bmbf.de/foerderungen/17740.php, retrieved: May, 2012.

[4] Kluge, A. (2004), „Wissenserwerb für das Steuern komplexer Systeme“,. Germany: Pabst Science Publishers.

[5] Schöning, U. (2001), „Algorithmik“, Berlin: Spektrum, Akad. Verl.

[6] Palviainen, J and Väänänen-Vainio-Mattila, K. (2009), “User Experience in Machinery Automation: From Concepts and Context to Design Implications”; Human Centered Design, Lecture Notes in Computer Science, 2009, Volume 5619/2009, 1042-1051, DOI: 10.1007/978-3-642-02806-9_119