2013 IEEE International Conference on Systems, Man, and Cybernetics

Knowledge-driven adaptive production management based on real-time user feedback and ontology updates

Daria Kazanskaia, Sergey Kozhevnikov, Vladimir Larukhin, Petr Skobelev, Alexander Tsarev, Yaroslav Shepilov
Smart Solutions Ltd.
Samara, Russia
skobelev@smartsolutions-123.ru

***Abstract*—This paper presents the principles of the knowledge-driven adaptive management in manufacturing. The problems of real-time resource allocation, reaction to the unexpected events, on-the-fly update of the knowledge stored in ontology are considered. The possibilities of simultaneous change of the existing factory or workshop processes and schedules according to the information provided by the users are described. Finally, the possible ways of development of the presented approach and its application in production resource management are considered.**

Keywords-production management, ontology, multi-agent technology, real-time scheduling

I. Introduction

The modern production often faces a problem of creating up-to-date schedules and long-term plans and of keeping them relevant in changing environment. These issues become even more crucial when talking about ramp-up phase or small lot production. Both cases deal with numerous change request by customers, additional orders and discovered production issues that require fast changes in production processes and new plans and schedules that correspond these processes, new order settings and new capacity distribution.

These issues are covered within ARUM (Adaptive Production Management), a collaborative project within the EC "Factory of the Future" initiative, intends to address these topics by introducing an innovative architecture based on novel ICT solutions to handle new challenges in production and ramp-up of complex and highly customized products, namely those in small-lot production. In particular, the focus is in the development of mitigation strategies to respond faster to unexpected events and the implementation of systems and tools for decision support in planning and operation.

ARUM solution combines the most up-to-date approaches to solving the named problems: using multi-agent technology for adaptive scheduling and strategic planner, intelligent enterprise service bus (ESB) and p2p networks for modules interaction, combination of ontology and user interface enabling users to update information on production and orders execution "on the fly".

The latter is a focus of the present paper.

This paper is organized in the following way: the first section gives an overview of the solved problem, the second one describes the architecture of ARUM solution and the role of ontology in it. In the third section, an example of adaptive resource management and scheduling using on-the-fly ontology updates is given. The last two sections describe the future development of the approach and give a brief conclusion.

II. Problem Statement

The acquiring, formalizing, collecting, storing and re-use of the knowledge obtained by experienced worker is one of the key problems of modern companies, since it is not only the base for relevant decision making, but is of great help in improving schedules of resources and efficiency of business but also teaching new employees and improving the competencies of mature ones.

The gathered information can be used for creating the long-term plans and short-term schedules.

However, the today production requires not only collecting the existing knowledge on factory environment and processes, but also its updating according to the current situation and occurred events. In this case the main role is played by the end-user, who can provide this information such as production issues, the methods of dealing with them, information on tasks and orders execution (required time, instruments and materials) and thus it is necessary to provide the clear and friendly user interface that will be understood even by the people that are now very experienced in using PC or smartphones encouraging them to interact with the system and share their knowledge.

Another issue is the proper storing of the required information. One of the most flexible and intuitively clear solutions is using ontology. The problem in this case is describe the core of ontology, the main concepts of the domain and their relations and attributes, which later will be used to describe specific factory or workshop environment and production processes.

However, this information will just be idle without the modules that can use it. Within ARUM project, the modules using multi-agent approach will be developed providing strategic planning and operational scheduling in real time. The latter is mostly influenced by the data stored in ontology and

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.809

even more by the information provided by user, because the schedule created by the system should be reflect to the current situation in the workshop/factory and be relevant with respect to all events and changes that can happen during order execution process.

This strong bound between schedulers, ontology and user interfaces needs fast and reliable interaction mechanisms that can help to retrieve data within the seconds.

III. STATE OF THE ART

During the recent decades there appeared a number of tools for creating and supporting of ontologies that provide not only standard editing and browsing functions, but ontology documentation support, import and export abilities for different formats and languages, graphical editing, managing ontology libraries and other facilities.

The most known ontology management tools include Ontolingua [1] providing web-based ontology editor, tools for accessing, analyzing and aggregating ontologies; Protégé [2], one of the most popular tools for domain representation and OntoEdit [3]. Both Protégé and OntoEdit are Java-based and support plug-ins.

Another tools include OilEd [4], WebOnto [5] and KADS22 [6].

Nowadays there exist several successful applications of the ontologies in various Internet projects.

Ontolingua Server (http://ontolingua.stanford.edu/) [1] is a tool for creation, support and utilization of ontologies stored at the server. The terms, types and relations (axioms) describing the knowledge fragment are introduced in ontology. The system uses its own language called Ontolingua that implements the principles of object-oriented approach and is an extension of KIF language for knowledge exchange and interaction between programs. The tool is designed for web-applications, it has HTML interface for visual creating and supporting of ontologies.

In paper [7] the example of creating a large-scale knowledge base in the form of ontology is considered. The base of ontology is The CIA's World Fact Book the collection of geographical, economical, social and other facts described using SGML. Specially designed parser retrives the knowledge and writes them in the form of KIF-axioms.

SMART software [8] developed as a part of Protégé, knowledge modeling support system, is using an algorithm of semi-automatic merging and adjusting of ontologies. SMART helps to discover inconsistencies in ontological statements that were probably created as a result of user actions and suggests the possible solutions.

Ontological representation of the content combined with linguistic ontology can increase both reaction speed and accuracy of the provided information for "yellow pages" and product catalogues. OntoSeek, information and search system [9] is developed for semantically oriented search of the information and uses the mechanism of meaning matching mechanism managed by ontology and modeling system.

The ontologies are also applied in multi-agent system, however this approach is not common. Moreover, as we can see from the presented examples, now the use of ontologies is focused mostly on semantic data analysis and domain description [10].

The present paper considers the advantages of using the ontology combined with multi-agent system and tools providing easy description of production environment and processes and on-the-fly updates.

IV. ARCHITECTURE OF THE SOLUTION

ARUM solution is based on the concept of iESB (intelligent enterprise service bus) that connects all the modules forming the system, data storages and legacy systems.

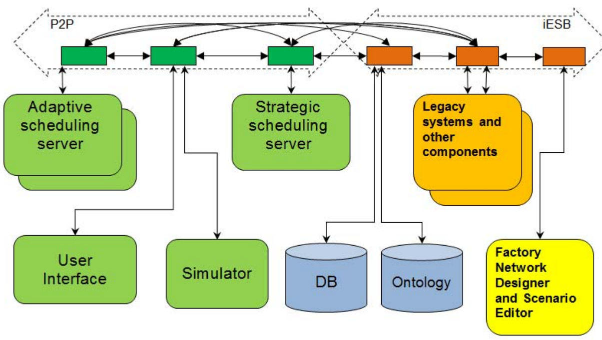

Figure 1. Architecture of ARUM solution

Although iESB is a crucial part of the system, the main work on planning and scheduling is done by Strategic Planner and Operational Scheduler, which operation depends not only on the used algorithms, but also on the data on factory or workshop environment and current situation. This information can be provided and adjusted by Network Designer (information on factory environment) and Scenario Designer (snapshot of the current situation).

All information used by these modules is stored in knowledge base, which consists of three layers:

- Ontology is the formal description of the domain specifying all classes, relations between them and their properties. This layer is edited in Ontology Management Tool.
- Model (ontological model) is the representation of the factory/workshop environment based on the ontology and including the instances of classes that exist in this environment (for example, processes (not running, but potentially possible), parts, equipment models, skills, workers). This layer is edited in Factory Network Designer.
- Scene is the representation of the current and scheduled situation in factory/workshop based on the information provided in the model. Scene includes the exact states of the physical objects, current and scheduled values of attributes and relations (for example, what exactly workers do or will do, what processes are running,

what parts and how many are now in stock or will be produced, what orders and when will be finished). This layer is edited in Scenario Designer.

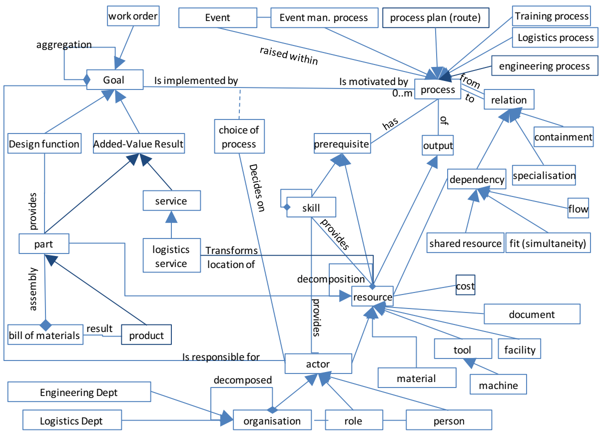

Figure 2. ARUM solution ontology

The ontology within ARUM solution complies with ISA 95 standard and plays two significant roles:

- As a knowledge base that stores information on the existing factory or workshop environment and technological process (fixed part of the ontology);
- As a mechanism for collecting up-to-date information on task execution details, non-conformities, on-the-fly changes in ontological processes (flexible part of the ontology).

OMT plays the main role in specifying the factory ontology, since it allows the users to set key concepts and relations between them and rules. OMT operates in connection with other ontology-related modules and tools as displayed in the picture below.

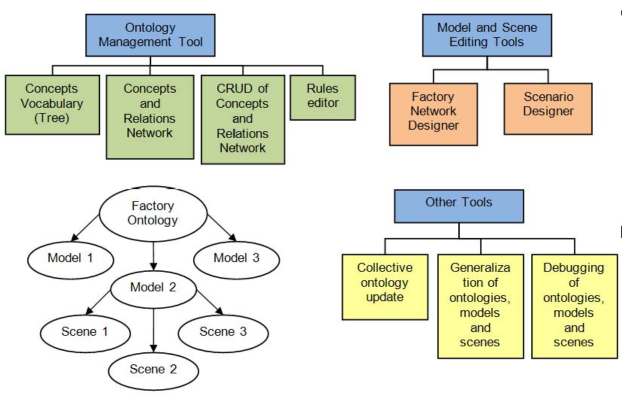

Figure 3. Ontology Management Tool components

This mechanism of gathering the most relevant data is a key factor for operational scheduling, since it requires the information on the unexpected events or non-conformities that can influence the execution of the order and overall schedule of workshop or factory.

During the scheduling process, the system uses the information on technological processes, worker skills and competencies, data on equipment condition and availability to specify the order execution: divide an order into operations according to the required process and assign workers with the required skills and available equipment to execute them.

After an order execution is started, the worker receives the tasks with the set execution time (based on the previous experience). When worker completes the tasks, the information on execution time is entered into the system and thus the stored time is updated.

While executing the task, worker can face different non-conformities. She/he can let the manager and engineers know about it by entering the information via specific interface (user terminals or tablet PC). This information is stored in ontology. The engineers can specify the ways of dealing with this kind of non-conformities that will be stored in ontology too (instrument is not applicable, part is missed, etc.). Thus, when user will face this issue next time, the system will immediately suggest to select one of the stored solution options.

V. Adaptive production Management

Let us consider an example of adaptive production management and scheduling using on-the-fly ontology updates.

There is a production process described in ontology. The process consists of three sequential operations. There is a non-conformity processing process described in ontology. The process includes decision-making points and therefore has optional paths. There are N orders (demand for installed seat) waiting for processing. There is everything to start production for a few orders (materials, pre-products, details, documentation).

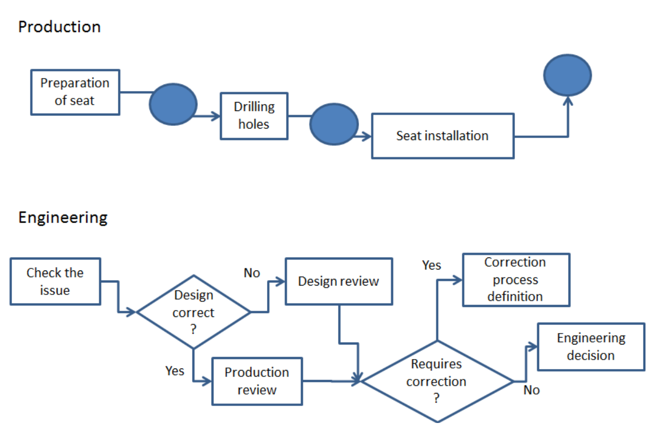

Figure 4. Technological process description

Manager goes to the orders list and starts scheduling of the first order (O#1), i.e. activates the demand for installed seat). O#1 is scheduled for the available resources. Each operation is allocated to its own resource. One station is used for all operations in the production process. Manager gets the scheduled time of installed seat delivery and other KPIs.

Manager activates (allows automatic scheduling for) next order. He sees the scheduling progress on the screen (how many demands are satisfied, how the KPIs change). After scheduling is complete, (all demands are satisfied) the full schedule and KPI reports are available.

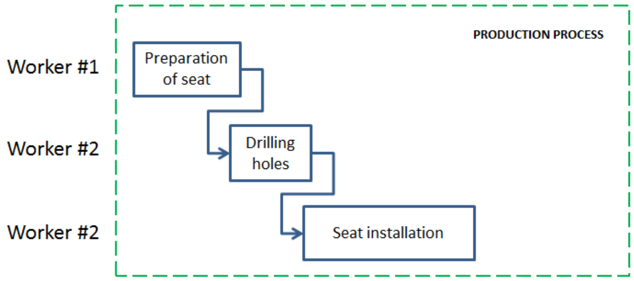

Figure 5. Schedule for one order

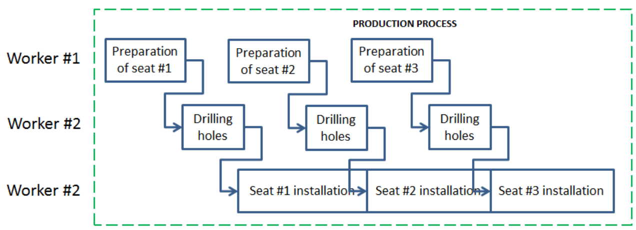

Figure 6. Schedule for three orders

Worker #1 can see his schedule (seat #1, #2, #3 preparation) on the terminal.

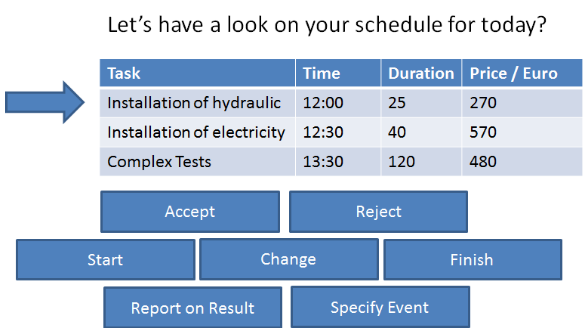

Figure 7. Example of the user interface

Worker #1 starts "preparation of seat #1" operation using terminal. Worker #1 finishes "preparation of seat #1" operation and specifies that the result fit expectations as "prepared seat #1" state.

Worker #2 starts "making holes #1" operation according to the specification attached to the operation description. Worker #1 starts "preparation of seat #2".

Worker #2 finishes "making holes for seat #1" operation with the result "seat #1 with installation holes" state.

Worker #3 starts "installation of seat #1" operation and discovers that a hole is made in a wrong place and cannot continue this work and specifies new event using OMT.

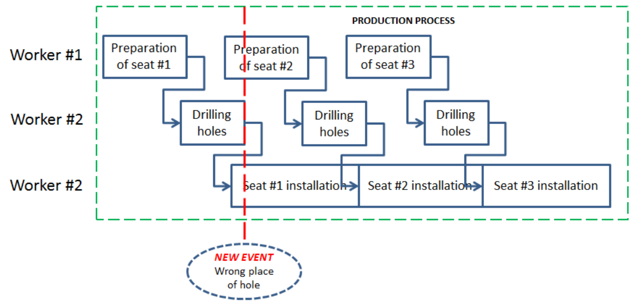

Figure 8. Non-conformity discovered during process execution

Worker #3 describes event "seat #1 with installation holes" as not compatible with the "aircraft A". He presses the button "Production issue". The system shows the list of known issues (delay, no materials, problem with instrument, not enough skills, etc.).

If there are no such issue in a list he can use the "Another reason" button that leads to the scene editor. Worker #3 presses it. In the Scene Designer he creates a new position in the hole specification and declares its incompatibility with the aircraft specification.

The system sees that the order cannot be scheduled without updating of the technology and finds an engineer responsible for seat installation.

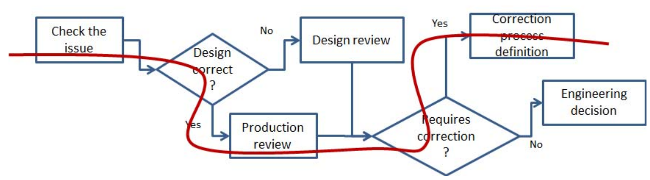

Figure 9. Process for non-confrmity elimination

Lead engineer sees the request from the installation demand agent. He selects the assumed incompatibility processing path for the current situation that requires additional operation on seat correction. Lead engineer defines (or accepts a proposed) preliminary time for additional production works (correction) and lets the scheduler to schedule the plan.

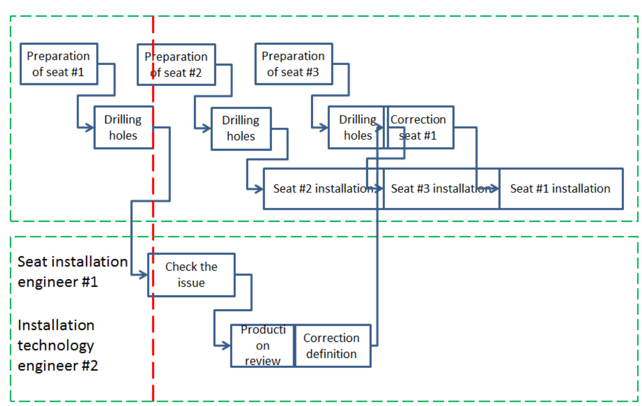

Figure 10. Schedules including engineer works

Manager is notified on new schedule and KPIs.

Workers #2 and #3 receive new schedule.

Worker #2 receivers a warning that “drilling holes for seat #1” operation was done not correctly and a request to confirm the correction time provided in the process selected by engineer.

Engineers get their schedules. Engineer #1 starts “check the issue” operation.

Workers continue production according to the new schedules. Worker #1 finishes “preparation of seat #2” operation. Worker #2 starts “drilling holes for seat #2” operation.

Engineer #2 finishes “check the issue” operation and specifies that there are no problems with the design.

Worker #2 is notified that the design is correct and he should not allow the fault with the “drilling holes” operation anymore. At this stage his experience or bonus points can be reduced as a result of previous incorrect actions.

Worker #1 starts “preparation of seat #3” operation. Worker #2 finishes “drilling holes for seat #2” operation. Worker #3 starts seat #1 installation. Worker #2 starts “drilling holes for seat #3” operation.

Engineer #2 finishes “production review” and “correction definition” operations and specifies using OMT how exactly to fix the seat. It can now be installed at the right place without additional operations for correction. This solution, i.e. alternative branch of technological process consisting of several operation is saved in ontology. If the worker will face this issue again, she/he can just select it from the list and the system will automatically suggest the solution.

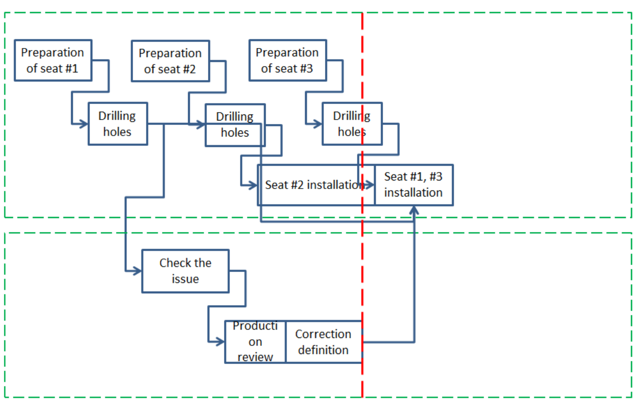

Figure 11. Final schedule

Worker #2 is notified that the drilling of the holes for the seat #1 were done wrong. Worker #2 finishes “drilling holes for seat #3”.

Worker #3 finishes seat #2 installation. Worker #3 starts seat #1, #3 installation according the new process defined by the engineer as a solution of the issue. Worker #3 finishes seat #1, #3 installation. Process successfully completed.

VI. Future Development

The suggested approach will be developed within ARUM solution in the following directions:

- Development of the ontology and description of the processes and the flexible part of ontology;
- Development of the user interface for updating information on technological processes and non-conformities;
- Development of the interaction mechanisms between ontology, interfaces and other modules of the system such as schedulers.

VII. Conclusion

The presented approach to production management is potentially of great use, since it provides facility both for data collection, its on-the-fly updates and immediate utilization in the management process. As we can see from state-of-the-art investigation, the existing ontologies and editing tools focus mostly on processing Internet text, but not for modeling the production processes and immediate updates by users. Thus, developing this approach requires creating new tools to support all the suggested facilities and probably new OMT that will easily support the proposed three-layer architecture and provide convenient API that can be applied in user interfaces.

Acknowledgment

This paper was prepared within Adaption project Management (ARUM) project of FP7 Factory of the Future program (Grant agreement no: 314056).

References

[1] A. Farquhar, R. Fikes, J. Rice, “The ontolingua server: A tool for collaborative ontology construction”, International journal of human-computer studies, pp. 707-727, 1997.

[2] M. Musen, “Domain ontologies in software engineering: use of Protege with the EON architecture”, Methods of Information in Medicine-Methodik der Information in der Medizin, pp. 540-550, 1998.

[3] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke, “OntoEdit: Collaborative ontology development for the semantic web”, pp. 221-235, 2002

[4] S. Bechhofer, et al, “OilEd: a reason-able ontology editor for the semantic web”, KI 2001: Advances in Artificial Intelligence, pp. 396-408, 2001.

[5] J. Domingue, “Tadzebao and WebOnto: Discussing, browsing, and editing ontologies on the web”, pp. 93-112, 1998.

[6] G. Schreiber, “Knowledge engineering and management: the CommonKADS methodology”, pp. 28-37, 2000.

[7] A. Farquhar, R. Fikes, J. Rice, “Building a large Knowledge Base from a Structured Source: The CIA World Fact Book”, Intelligent Systems and their Applications, IEEE, vol.1, pp. 47-54, 1999.

[8] N. Noy, M. Musen, “SMART: Automated Support for Ontology Meriging and Alignment”, pp. 1-24, 1999.

[9] N. Guarino, C. Masolo, G. Vetere, “Ontoseek: Content-based access to the we“, Intelligent Systems and Their Applications, pp. 70-80, 1999.

[10] P. Skobelev, “Ontologies of actions for situation-driven real time management of enterprises”, Design Ontology, pp. 6-38, 2012.