Interactive Wiki for Special-Purpose Machines

Thomas Herpich

Institute of Information Systems
Hof University of Applied Sciences
95028 Hof, Germany
Email: thomas.herpich.2@iisys.de

Valentin Plenk

Institute of Information Systems
Hof University of Applied Sciences
95028 Hof, Germany
Email: valentin.plenk@iisys.de

***Abstract*—Machines in processing plants are frequently generating failures that must be manually fixed by the operators. Machines often lack advanced assistance systems for to address these failure cases. While some new developments try to use only machine data, in many applications, the human knowledge of the operators can be very useful. In this paper, we propose a new assistance system used to merge machine data with the operator's knowledge. This system is tested with an industry partner. The test results are used to create design considerations, compare different reasoning algorithms, and check the influence on the machine downtime.**

***Keywords*–Human Machine Interfaces; User Assistance Systems; Special-purpose Machines.**

I. INTRODUCTION

Digitally controlled machines and systems in production are becoming increasingly intelligent. As part of in-process control, they continuously check their own status to control the production process and monitor compliance with the specifications. Maintenance and repair requirements can also be predicted in reasonable time so that consistent quality in the production process is permanently ensured. Failure times are reduced accordingly, relieving the operating personnel of numerous routinal tasks during the operation of the machines.

Despite the increasing intelligence of these computer-controlled machines, problems still occur in production processes which cause the systems to come to a standstill. In these cases, the problems are often more complex, for example because a plant consists of several components from different manufacturers, whose interaction is not regulated in any manufacturer's user manual. Despite digital technologies in the control of each individual component of the plant, manual intervention by the operator is necessary in such cases. The operator must then be able to identify the reasons for the malfunction, initiate suitable measures and put the system back into operation. They are often assisted only by the user manual, or an assistance system that is different for each manufacturer and machine [1], [2].

In [3], Oehm *et al.* created and evaluated a design for an assistance system for processing plants. In this paper, we propose an implementation for that design which was slightly adjusted for special-purpose machines used by our project partner. In [4], we described an approach to automatically extract the knowledge only from available machine data without integration of the operators. In [5], we showed that this approach is not always feasible, because the data only represents a small part of the work that must be done. To accommodate this problem, the machine operators can document information about the current situation and the appropriate steps to fix the failure in the new software. This information is then mapped with the current state and should be automatically retrieved when this situation occurs the next time. Available knowledge can also be reused by the operators.

With this approach, the software should be able to provide useful information automatically after a training phase. This should reduce the downtime, especially with inexperienced workers or infrequent failure causes. The software learns automatically, which data can be used to distinguish different failure cases and is machine independent, if the generated data can be made available to the system.

The new system is called "IISYS Machine Wiki".

In Section II, we describe the structure of all involved systems, Section III gives the description of the new assistance system. This work ends with a conclusion and future work.

II. SYSTEM DESCRIPTION

This section gives an overview of the involved systems. Figure 1 shows the structure of the machine we are using and the new assistance system.

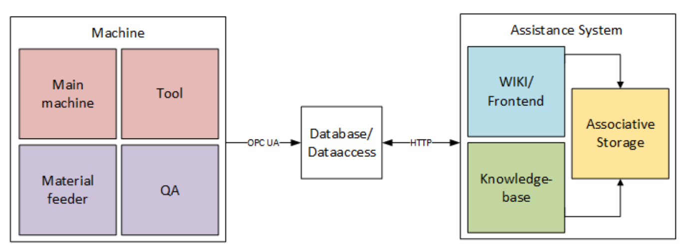

Figure 1. System structure.

a) Machine: The machine (left part of the figure) is a stamping press (Main Machine). It uses a changeable tool for the work. There are also some peripheral machines like a material feeder, an automated Quality Assurance (QA), and the commission. All machines are generating data consisting of setpoints, sensor values and machine states. This data is collected via Open Platform Communications Unified Architecture (OPC UA) and then saved in the time series database influxdb (center in figure). For one machine, there are around 7500 variables available, where some are not used. More details are provided in our previous work [5]. From there, all additional software can access all data generated since the data collection started (currently around two years).

 ISBN: 978-1-61208-829-7

b) Assistance System: The newly created assistance system (right part of the figure) consists of the following parts:

- Frontend: This is the user interface that is used by the machine operators and the foreman.
  It is created as a Web application with JavaEE, so it can be used with any computer and from any machine in the factory.
- Knowledgebase: The knowledgebase stores all information gained from the user. It also maps it with the corresponding failure. The knowledgebase is then used to train a model to find the correct information for any occurring situation.
- Associative Storage: This storage maps error states to machine data and is used to find the appropriate wiki page for any given situation.

III. Description of the Assistance System

This section describes the new system and its features.

A. Algorithm

This section describes the algorithm used in the new system. Figure 2 shows all steps done by the system. Symbols in blue use the user interface, green interact with the knowledgebase and yellow process the information from the knowledgebase.

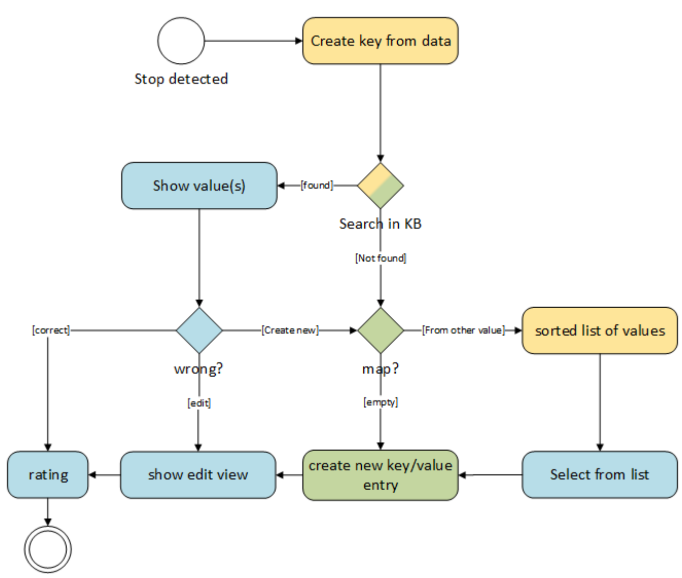

Figure 2. Activity diagram new system.

The software continuously checks incoming machine data for the stop condition, as described in Section III-A1. When a stop is found, the following sequence starts:

- Build the key representing the current machine state from the available data, as described in Section III-A1.
- Search for the key in the knowledgebase. This is described in Section III-A2.
- If the key is found, the corresponding value (wiki page) is shown to the operator.
- The operator can request to change the shown wiki page, see Section III-A4.
- If the operator proposes another page or no page could be found in the knowledgebase, a new mapping must be created, see Section III-A4. This new mapping can be filled manually with information (case "[empty]") or linked to a value from another key.
- To create this link, a list with possible values is shown. This list is sorted by the knowledgebase key, where the most equal key will be shown first. Section III-A2 describes how this works.
- When this mapping is created, the operator can edit the new page.
- At the end, the operator can rate the proposed sequence with 0 to 5 stars.

1) Stop Detection: The new system must detect if the machine is in an error state. This can be accomplished by analyzing new incoming data from the machine. This data contains one main error code (OPCGeneralInterface.ErrorMessage). This code is currently used by the machine to display an error text. While many error causes within the main machine can be distinguished by this code, every failure from the peripheral devices is only reported with one equal error code. The code contains the number 0 while the machine is running and a positive integer when in error state. The "ErrorMessage" can change during the stop while the operator fixes the error and can be used to display additional help.

An additional variable (OPCGeneralInterface.State) contains the operation mode, where 4 indicates normal operation, 0/1/2 a stopped machine, 3 manual set up and 5 an error.

The time span from error begin until the return to normal operation is called stop.

The machine sends the same main error code for every failure caused by peripheral devices, so it is not possible to distinguish all states. However, the machine data contains additional fields that can be used:

- Around 250 bit-fields (Table "Ems_db"). These bits correspond to different errors and states from the peripheral machines.
- 15 - 20 float values containing measurements like temperatures or the pressforces and some changing settings.

The distinct combinations of those fields would lead to around $2^{250} * errorcodes$ machine states. The algorithm used in this software must automatically decide which fields and corresponding values should be used to distinguish between different states.

2) Knowledgebase: The knowledgebase maps machine state information to the wiki pages that should be proposed. This information is used to find wiki pages that can be proposed to an operator in case of a machine failure.

Different methods can be used to compare the machine data of different situations and to distinguish between them. In [6], the authors use a distance-based approach where one number (called distance) is calculated from selected machine variables which is then used to compare the data. A drawback of this method is that the variables to be used must be known to generate a useful result.

An approach that decides automatically which variables to use and that generates decision rules are decision trees. A random forest uses multiple decision trees combined as a voting system. These trees are trained using different combinations of the available fields. It can also leave out variables that are not helpful in distinguishing the different situations.

Any of these approaches need a training set in form of past situations mapped to the correct wiki pages. This training set will be generated during the first months of use. Until this is available, the software only uses the main error codes and the operator needs to select which proposed page is the best for the current situation.

3) Show Wiki Pages: When a machine failure is detected, our system uses the current machine data to search for appropriate wiki pages. In the best case, that is exactly one page, but if the software cannot distinguish some cases, more than one page is proposed to the operator. These pages are displayed in tabs as described in Section III-B. These tabs are sorted with the first tab being the best matching page. The operator can select the best page for the current case or create a new one. A similarly sorted list is used if the operator searches for wiki pages that can be mapped to the current situation, as used in Section III-A4. When using a random forest, the best entry is the most proposed entry from the forest. If the order for some entries cannot be determined clearly, the system uses the ratings created for this page in previous situations. An additional factor is how often the page was used and displayed in previous cases.

4) Create and Use Wiki Pages: The operator has different possibilities to change wiki pages and map them to the current situation. This section describes these possibilities.

a) Create a New Page: When the algorithm could not find any wiki page to display, the operator must create a new page, which is mapped with the current situation with a new entry in the knowledgebase. This is also available if there are pages available, but no page suits the current needs.

b) Create Page from Template: The operator can create a new page based on another page. The content of the existing page is copied to the new page, which can be changed by the operator. These changes will not apply to the base page as opposed to the operation in the next paragraph.

c) Associate an Existing Page with an Error: With this action the operator can map the current situation to an existing wiki page which was not proposed. This creates a new mapping in the knowledgebase. When this page is edited, the changes will be shown in all situations that are mapped to this page.

d) Edit Current Page: This action does not change any mappings in the knowledgebase. If the displayed page is mostly correct, but there are some misleading or missing information, the operator can edit this wiki page. The change applies to all machine states that are mapped to the same page.

e) Replace a Mapping: This action removes the current mapping with the situation to the selected page and then starts the action “Associate an existing page to this error”.

B. User Interface Design

The design for the UI was worked out with the project partner in [7]. The software is designed according to user experience guidelines given by [1], [8]–[10].

The wiki system will be displayed on a screen next to the machine with a keyboard and mouse attached. This setup is already available and used for other tasks by the operators, so there is no new hardware needed.

Figure 3 shows the main page of the application.

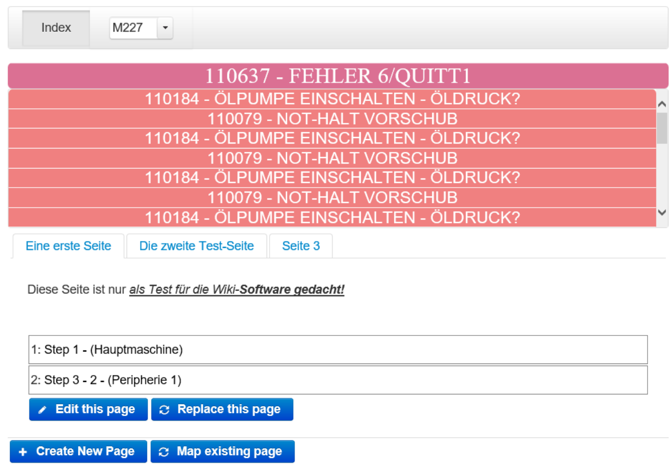

Figure 3. Screenshot main page.

The operator can see the current error that caused the failure in dark red. Below is a list of additional occurring error messages while the error is fixed, as suggested by the operators. If the machine is running with no error, the background changes to green.

All possible wiki pages that are found for the current situation are displayed as tabs below the error code. The user can select the most useful page. From within these tabs, the tasks described in III-A4 can be executed:

- “Edit this page”
  Edit the selected page. This change must be approved by a foreman before it will be shown in the main page.
- “Replace this page”
  This action removes the selected page as possibility for the current situation. After this action, the operator can select another page that suites the current failure.
- “Create New Page”
  The operator can create a new wiki page. This page is automatically mapped to the current situation.
- “Map existing page”
  The operator can select an existing wiki page that suites the current failure. This selected page will then be mapped to the current situation. As opposed to “Replace this page”, the mapping of other pages with this situation will not be removed.

The tasks “Edit this page” and “Replace this page” are only available if the algorithm has found at least one wiki page to show for the current situation. If no pages are available, the tab list remains empty and the operator can create a new page or map an existing one.

When the error is fixed, the operator has the possibility to rate the proposed page with 0 to 5 stars. This rating is then used to improve the algorithm as described in Sections III-A2 and III-A3.

Every wiki page consists of a title and an error description which can be entered through a rich text editor to enable some formatting as well as the insertion of images. The error description should contain a detailed description of the problem as well as some information how the operator identified the exact problem. The operator can enter the steps that are necessary to fix the failure. These steps are structured in different categories (e.g., Main Machine, Periphery QA) where both can only be created by a foreman. This is to ensure that every operator uses the same wording for the same steps. Figure 4 shows the UI to edit the step list with the page title and error description on the top. The step list can be reordered by the buttons on the right.

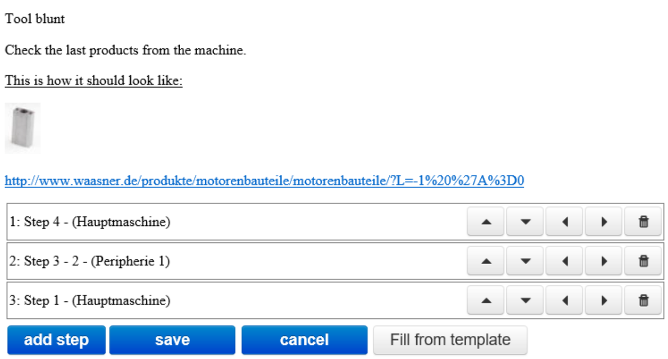

Figure 4. Screenshot create step list.

The list can be expanded by clicking the button “add step” which opens the UI shown in Figure 5.

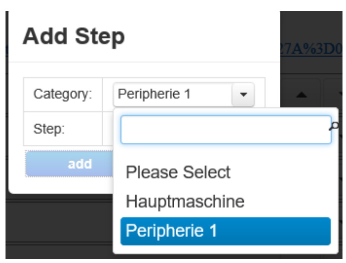

Figure 5. Screenshot add step to list.

The steps are grouped by category, so the user must select a category first and then a step from that category. Every important input field contains a search box and predefined values to simplify the usage.

Admin users have an overview of past stops that can be used to create page mappings for past situation to improve the algorithm.

IV. Conclusion and Future Work

A. Assistance System

With the created assistance system, the operators can save and share their knowledge with other users. This knowledge is stored in a wiki-based structure. Every wiki page can be freely mapped to any situation, and every mapping can be removed.

To improve the (perceived) usefulness of the software, the users from the project partner were integrated into the development, as suggested by [8] and [9]. They are integrated through multiple meetings and a test phase where all users will be able to give additional feedback.

The roll out began in June 2020 for the first machine and is scheduled to be expanded to other machines by August 2020.

B. Test machine

While we are waiting for data from our project partner, we created a simple test machine with similarities to the real machines. Our machine picks up either a magnetic or plastic chip, then moves the transport arm in different directions and releases the chip at a predefined location. The machine generates 64 bits representing different problems and sensors in the machine but has no common error code that identifies a problem cause. Then we will simulate different runs, some of them will fail with different causes. Many of these causes can be fixed with the same procedure. The new software is used to create wiki pages for these causes and map them to the runs. Different causes will then be simulated again, and the software should propose the correct wiki page as mapped before

C. Future Work

In the next months, the system will be running 24/7 in full production on two machines.

The data generated by the project partner and from our test machine can then be used to evaluate:

a) Reasoning Algorithm: The proposals of the new system can be compared to proposals only generated with the main error number. The software should at least be as useful as using only the main error code. This will be measured by comparing the needed time to fix the errors and the rating operators can give in the system. The amount of remappings or new page creation is also a good measure. If many situations have multiple wiki pages mapped to them this would indicate that cases cannot be distinguished past the main error code. The results can additionally be compared to other reasoning algorithms like distance-based algorithms which can be used with the same knowledgebase.

b) User Experience: The operators will be questioned with standardized user experience questionnaires, e.g., the User Experience Questionnaire (short version) [11], on how satisfied they are with the selected design. With this information, the software will be changed to improve the acceptance of the new system or to decrease the time needed to fix failures. This feedback could also be used to create some guidelines for assistance systems.

c) Downtime: An advanced assistance system should decrease the downtime caused by fixing machine failures. To verify this, a baseline must be created from the current production. This can be done with the machine data available for the last two years. That baseline can then be compared to new data generated after the test phase 1.

d) Match Steps with Data: Some of the steps entered by the operators can be mapped with changes in the machine data. This mapping could enable the system to ensure that a step is carried out in the correct way by checking the machine data. It could also allow the system to generate a basic wiki page by recording the changes in data while an operator fixes a failure and searching the corresponding steps mapped to these changes. The basic page can then be edited by the operator to insert the missing steps.

 ISBN: 978-1-61208-829-7

References

[1] S. Delisle and B. Moulin, "User interfaces and help systems: From helplessness to intelligent assistance," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 117–157, Jun. 2002, ISSN: 1573-7462. DOI: 10.1023/A:1015179704819. [Online]. Available: https://doi.org/10.1023/A:1015179704819.

[2] A. Maedche, S. Morana, S. Schacht, D. Werth, and J. Krumeich, "Advanced user assistance systems," *Business & Information Systems Engineering*, vol. 58, Aug. 2016. DOI: 10.1007/s12599-016-0444-2.

[3] L. Oehm *et al.*, "Cooperative fault diagnosis by operator and assistance system for processing plants," in *Technische Unterstützungssysteme, die die Menschen wirklich wollen (Band zur zweiten transdisziplinären Konferenz 2016)*, 2016, pp. 375–384, ISBN: 978-3-86818-089-3 (Print) bzw. 978-3-86818-090-9 (Online).

[4] V. Plenk, "Improving special purpose machine user-interfaces by machine-learning algorithms," *Proceedings of CENTRIC 2016*, pp. 24–28, 2016.

[5] T. Herpich, "Commissioning and evaluation of a software system consisting of c#- and java applications for real time retrieving and saving as well as analyzing machine events from a opc server," Bachelor-Thesis, Hof University - Institute of Information Systems (iisys), Mar. 7, 2019.

[6] V. Plenk, S. Lang, and F. Wogenstein, "An approach to provide user guidance in special purpose machines and its evaluation," *International Journal On Advances in Software*, vol. 10, no. 3, pp. 167–179, 2017.

[7] T. Herpich, "Designing an interactive wiki assistance system for special purpose machines," Hof University - Institute of Information Systems (iisys), Feb. 25, 2020.

[8] M. Schelkle and C. Grund, "Identifying design features to increase the acceptance of user assistance systems: Findings from a business information visualization context," in *13th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2018)*, Jun. 2018.

[9] C. Aringer-Walch, S. Besserer, and B. Pokorni, "User needs for a digital assistance system in the context of industry 4.0. an explorative study in the area of assembly," in *Technische Unterstützungssysteme, die die Menschen wirklich wollen (Band zur dritten transdisziplinären Konferenz 2018)*, Dec. 2018, pp. 139–150, ISBN: 978-3-86818-245-3 (Print) bzw. 978-3-86818-246-0 (Online).

[10] M. Funk, M. Hartwig, N. Backhaus, M. Knittel, and J. Deuse, "User evaluation of assistance systems for industrial assembly," in *Technische Unterstützungssysteme, die die Menschen wirklich wollen (Band zur dritten transdisziplinären Konferenz 2018)*, Dec. 2018, pp. 213–221, ISBN: 978-3-86818-245-3 (Print) bzw. 978-3-86818-246-0 (Online).

[11] D. M. Schrepp. (2017). "Short version of the user experience questionnaire," [Online]. Available: https://www.ueq-online.org/Material/UEQS_Items.pdf (visited on 01/21/2020).