

# A formal framework of human–machine interaction in proactive maintenance – MANTIS experience

Špela Poklukar, Gregor Papa & Franc Novak

To cite this article: Špela Poklukar, Gregor Papa & Franc Novak (2017) A formal framework of human–machine interaction in proactive maintenance – MANTIS experience, *Automatika*, 58:4, 450-459, DOI: [10.1080/00051144.2018.1465226](https://doi.org/10.1080/00051144.2018.1465226)

To link to this article: <https://doi.org/10.1080/00051144.2018.1465226>



© 2018 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 31 May 2018.



Submit your article to this journal [↗](#)



Article views: 703




View related articles [↗](#)



View Crossmark data [↗](#)



# A formal framework of human–machine interaction in proactive maintenance – MANTIS experience

Špela Poklukar, Gregor Papa  and Franc Novak

Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia

## ABSTRACT

The general concept of MANTIS project is to provide a proactive maintenance service platform architecture that allows to monitor essential system parameters and schedule maintenance in order to predict and prevent imminent failures. Human–machine interaction (HMI) is an important integral part of the platform by providing the right information in the right modality to the users when needed. As MANTIS comprises 11 distinct industrial use cases, the design of such HMI presents a great challenge. The framework presented in this paper originates from the scenario-based design and can be treated as a part of the overall scenario-based usability engineering approach. The framework has been conceived from an extensive list of HMI features extracted from the descriptions of use-case scenarios provided by each industrial partner. Due to the broad range of representative industry environments including production asset maintenance, vehicle maintenance, energy production management and health equipment maintenance we believe that the resulting HMI framework can be applied in different cases in practice and the paper would also be of general interest to the readers.

## ARTICLE HISTORY

Received 4 December 2017  
Accepted 12 April 2018

## KEYWORDS

Proactive maintenance;  
scenario-based design;  
usability engineering

## SUBJECT CLASSIFICATION CODES

include these here if the journal requires them

## 1. Introduction

Proactive maintenance is the popular approach to system maintenance that became feasible with the advent of pervasive sensing technology providing efficient context-aware solutions. The overall goal of the Electronic Component Systems for European Leadership Joint Undertaking project MANTIS – Cyber Physical System based Proactive Collaborative Maintenance [1], is building a proactive maintenance service platform that will enable novel maintenance strategies in different environments (e.g. industrial machines, vehicles, renewable energy assets). It upgrades preventive and predictive maintenance with the proactive maintenance. Preventive maintenance relies on periodic maintenance execution (i.e. periodic tests, calibrations, replacement of components), while predictive or condition-based maintenance relies on physical measurements and reacts when a certain threshold is reached. Proactive maintenance benefits from the above two strategies and focuses on the problems before they occur.

Human–machine interaction (HMI) is a generally accepted term for real-time interaction and communication between human users and a machine via a human–machine interface [2,3]. Hereby, the term “machine” indicates any kind of dynamic technical system and it relates to different technical and production processes in diverse application domains. Beside traditional functionalities of HMI, such as presentation and

processing of information, advanced features include explanation and adaptability based on user and application models and knowledge-based systems for decision support. While MANTIS strongly emphasizes autonomy, self-testing and self-adaptation, human role remains one of the important factors in system operation. The increased degree of automation in the control of dynamic technical systems does not replace human users but rather modifies the interface between both. Appropriate matching of both leads to user-centred design.

The human role in HMI is twofold: controlling, which comprises continuous and discrete tasks of open- and closed-loop activities, and problem-solving which includes the higher cognitive tasks of fault management and planning. Since MANTIS comprises 11 distinct industrial use cases, the design of a common maintenance service platform and consequently a common HMI framework presents a great challenge. In order to avoid possible inconsistencies and pitfalls we followed established principles and guidelines of HMI with emphasis on scenario-based design [4], which are well documented in books and research papers. However, in spite of a thorough theoretical background the reports of their application in practice are relatively few. From this perspective we believe that it makes sense to share our expertise with others, addressing similar problems in the future.

The rest of the paper is organized as follows: in Section 2 an overview of theoretical works related to scenario-based design are presented. Section 3 gives the scenario-based design procedure for the MANTIS HMI and the resulting functional specifications are outlined in Section 4. Section 5 describes the MANTIS HMI model, and Section 6 shows some comparison with similar systems. Further steps and conclusions given in Sections 7 and 8 provide lessons learned that might be useful to others developing or implementing similar complex HMI solutions.

## 2. Scenario-based design

As stated in [1], scenarios highlight goals suggested by the appearance and behaviour of a system. They exemplify what users try to do with the system, what procedures they adopt to carry out a task successfully or what actions may possibly lead to errors or even hazardous states. They also allow to collect users' feedback and interpretation of their experience when interacting with the system. A secondary advantage is that scenario descriptions can be created before a system is built and its impacts felt [5].

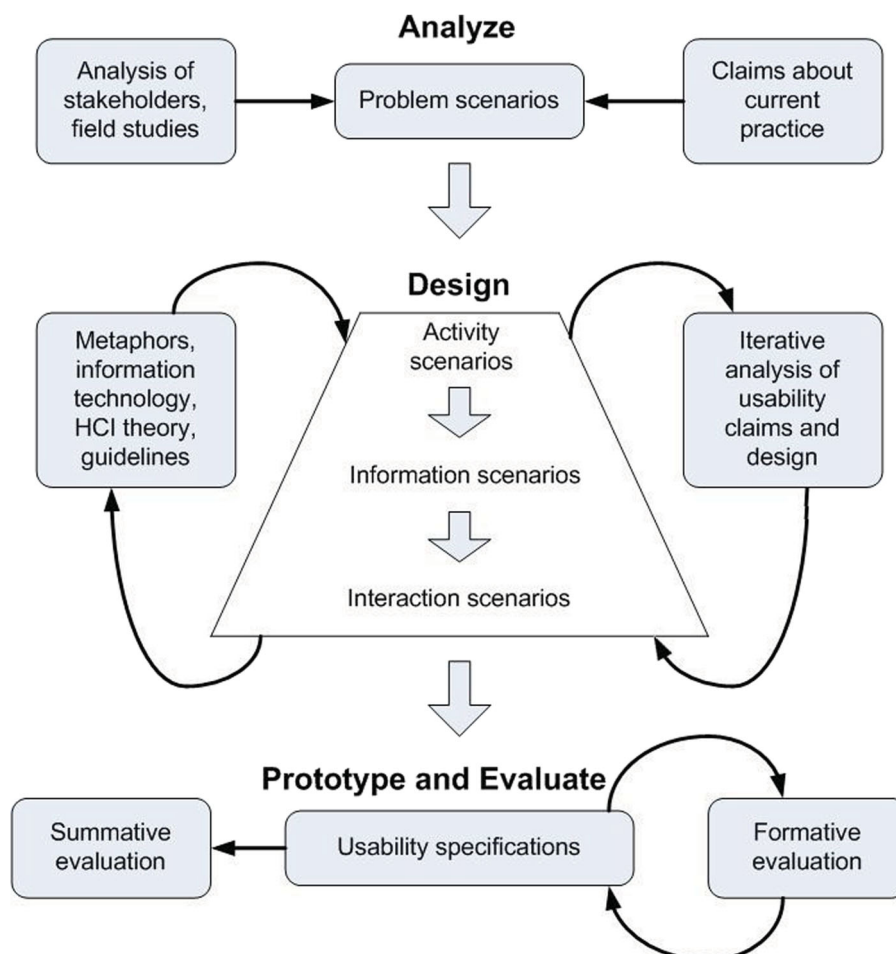
Scenarios have a plot; they include actions and events that may occur when users perform given tasks.

They describe things that users do in different possible circumstances of system operation. Scenarios also provide means for evaluating individual design decisions and estimating their impact on system usage. In this regard, they can also be used to identify and plan evaluation tasks for usability tests [6].

Figure 1 shows an overview of the scenario-based usability engineering framework proposed in [5]. The framework encompasses a task flow from problem analysis to design and then to evaluation. Scenarios are used to identify and analyse requirements, envision new design options, support prototyping and implementation, and organize evaluation.

The initial phase focuses on requirement analysis and comprises interviews with users, surveys, field studies, and brainstorming among users and developers. The objective is to formulate problem scenarios, which reflect typical tasks that users are supposed to perform on the target system. The scenarios are furthermore refined by claims that expose important features and their impact on users' experiences.

The design phase consists of three substages. The first substage focuses on pure functionality and identifies typical or critical services that people will seek from the system. The second substage deals with information issues (i.e. details about information related to the



**Figure 1.** Overview of scenario-based usability engineering framework.

user tasks). The third substage involves the design of interaction scenarios, which describe the details of user actions when performing a given task. Such a scenario describes how the user interacts with the system and what responses the system provides. The information needed to carry out the task is also implicitly specified. Usability aspects of each set of scenarios are analysed by claims that expose the possible positive and negative consequences of key design features. The design solutions are evaluated by two established approaches: formative and summative.

Formative evaluation is carried out to guide redesign, when the system is still in the preliminary stages of design when its basic concept and functionalities are being defined. Some typical user-oriented questions that a formative study would attempt to answer might include: what do users think about using the product, are the system's functionalities useful to the user, how easily can the user learn to use the system, etc. On the other hand, summative evaluation serves for system verification. Scenarios exemplify particular themes and guide evaluation through usability specifications. User tasks with specified usability outcomes are performed and evaluated repeatedly to guide redesign work. Comprehensive discussion on the use of scenarios in requirements analysis, information design, interaction design, prototyping and usability evaluation is given in [5].

While in the above framework emphasis is given to usability engineering issues, model-based design is successfully applied in other problem domains. Scenarios are, for example, incorporated in a strategy to provide flexible solutions that allow future system changes to be accommodated with minimal alterations to the existing system [7].

Scenario planning is also a widely accepted management process for decision support activities [8]. Scenarios are defined as a management tool for identifying a plausible future and a process for forward-looking analysis. The above work addresses the problem of process and support of knowledge-based scenario management in decision-making. A generic, knowledge-based, life cycle based approach for scenario management is proposed that supports a range of activities from idea generation to final use of the scenarios. Key phases of the life cycle are idea generation, scenario planning, organization, development, execution, analysis, evaluation, and decision support.

Scenario-based specifications have a wide acceptance in industry. Scenarios describe how system components, the environment and users' work interact and are well suited for initial study of intended system behaviour. In [9], the authors propose a process for elaborating system behaviour based on implied scenarios, which are the result of a mismatch between the behavioural and architectural aspects of scenario-based specifications. By iteratively detecting and validating

implied scenarios, it is possible to incrementally elaborate the behaviour described both in the scenario-based specification and models.

Scenarios can be abstracted and categorized and thus support model-based generation of user interfaces. Task models such as models using the Concur Task Tree (CTT) notation can be applied. A review of task models can be found in [10] and a taxonomy for the comparison of task models in [11], respectively. Alternatively, Object-Oriented Methodology for Software Production and associated Conceptual Model [12] has been tailored for the creation of information system applications.

Scenario-based approaches are applied in different areas in practice. In emergency management, evidence-based user scenarios are used for the production of bow-ties (i.e. risk evaluation methods that can be used to analyse and demonstrate causal relationships in high-risk scenarios) in the analysis of ground collisions [13]. Another recent work [14] deals with emergency response in disaster management. The paper addresses current scenario design processes and proposes an alternative approach for simulation exercises, which are a widely adopted training tool for enhancing preparedness for emergency response. It introduces a conceptual design of an adaptive scenario generator.

In software design, scenario-based approach is used for specifying software requirements in [15]. Basic flow regarding the successful use of the system, as well as alternative flows describing abnormal or less frequent interactions of the system are employed. While the latter are frequently inadequately described the paper proposes an approach for automatically recommending alternative flows from a basic flow by extracting the essential use-case patterns based on the occurrence patterns of the agents and measuring the verb similarity between the main verbs of each scenario. The proposed solution enhances software requirements specification which is one of the key factors for successful software development. In [16], scenario retrieval method using differential scenario is described. In a scenario-based software development, a lot of scenarios are typically described in order to clarify the whole behaviours of the target software. Reusing scenarios of similar software systems is established practice. A differential scenario is used to assess to what extent the two software systems match. The paper proposes a method for scenario retrieval using differential information between the two scenarios. A prototype system for creating and visualizing differential scenario is also described.

In the field of medicine, scenario-based design is elaborated in relation to usability standards in [17]. The focus of this paper is on developing scenarios of use for interactive medical devices. Scenarios are integral to the international standard for usability engineering of medical devices IEC 62366:2007. In [18], scenarios



were used to test and gain feedback on a pilot information system for the analysis of resource management. Both routine and disaster settings were considered in created scenarios. Scenario-based design proved to be a useful tool to validate the information system design.

And, to cite an example in the area of web design, scenario-based design and validation of REST (Representational State Transfer) web service compositions are described in [19]. In the paper, an approach to design and validate RESTful composite web services based on user scenarios is presented. Unified Modelling Language (UML) is used to specify the requirements, behaviour and published resources of each web service. REST web services are built on the principles of the REST architectural style that produce scalable and extensible web services.

### 3. Scenario-based approach in MANTIS HMI design

The purpose of MANTIS HMI is to provide an intelligent, context-aware HMI by providing the right information, in the right modality and in the best way for users when needed. To achieve this goal, the user interface should be highly personalized and adapted to each specific user or user role. Since MANTIS comprises 11 distinct use cases from quite diverse areas including production asset maintenance, vehicle maintenance management, energy production asset management and health equipment maintenance, the design of HMI that supports common proactive maintenance service platform presents a great challenge.

The applied approach focused on the requirements, common to most of the use cases and yet specific for proactive and collaborative maintenance. For this purpose, scenario-based design as described in [5] was adopted. In the analysis phase, a set of problem scenarios that reflected technological, functional, and business requirements of the use-case owners were collected. It turned out that in this early phase many different aspects and topics still needed to be discussed. Namely, the HMI in some use cases has already been implemented and put in operation before the MANTIS project actually started, while others were still in an early development phase. For efficient use of proactive maintenance service platform some modifications of the existing HMIs were required. On the other hand, these HMIs could also provide some reference for the others still under development.

In the requirement analysis phase, problem scenarios provided by the use-case owners varied according to the focus and task description, some insufficiently and others excessively detailed. A number of iterations, each directed by claims checking the adherence to the MANTIS mission and suggested revisions have been performed. A list of user scenario requirements comprising more than 400 items was generated. Each

requirement has been identified through its name, description, nature (business, functional, technological), type, implementation date, priority and the related use-case scenario(s). Sixteen type categories were defined including adaptability, availability, interface, reliability, safety, security, usability, and others. Notice that the above scenario requirements have been derived considering functionalities of the whole maintenance platform. For the HMI design, only a subset of the collected requirements came into account. Furthermore, the problem scenarios had to be refined with more details on user activities, information and interaction. This was done in a number of interactions with individual use-case owners.

In the first step, activity scenarios representing typical services that users would seek from the system were defined. These early scenarios provided a general view on the system and primarily focused on pure functionalities (without any implementation details).

Next, the scenarios were refined by providing more details about the information that the system will provide to users, such as displayed data during normal operation, abnormal operation or an incoming fault, the status of system components, generated alerts and alarms, etc.

Finally, the interaction details were included. The resulting interaction scenarios present typical maintenance situations, identify target users, define the devices which will be used in the concrete HMI, give details on the interfaces and processed information and describe how the required user tasks are performed. The developed scenarios represent the basis for the development of the formal HMI framework as well as for usability testing.

In the following, a generic MANTIS HMI is specified to the extent that does not introduce any constraints for the use cases, but at the same time describes the most important features of the MANTIS HMI that should be considered when designing the HMI in individual use cases.

### 4. Functional specifications

The functional specifications provided in this section are the result of refinement of the problem scenarios and the derived user scenario requirements. Functional specifications describe the HMI functionalities that are present in most use cases and abstracted from the specific situation of every single use case.

They are not meant as a replacement of MANTIS HMI requirements specifications for a separate use case, but rather serve as a reference point when writing ones.

Derived functional specifications are grouped into five categories: monitoring production assets, maintenance tasks scheduling, data analysis, reporting and communication. Each of them is shown in more details below.

### 4.1. Monitoring production assets

MANTIS HMI displays

- current, historical and predicted parameter values of monitored production assets and expected range of these parameters;
- comparison between actual and estimated wear-out or predicted remaining useful life;
- various statistics of historical parameter values of monitored production assets;
- possible failures of production assets together with some additional description, such as current and historical parameter values related to the faulty asset and possible feedback from other;
- current data in real time;
- alert if the monitored asset parameter is out of predefined range. The alert includes additional information on monitored asset parameter such as historical values of the parameter.

MANTIS HMI provides functionalities that

- enable the user to sort and filter monitored production assets, select different data sources, and select time range of monitored parameters;
- enable the user to select and flag the data;
- allow automatic generation of reports on monitored parameters and transfer the monitored data to other users.

### 4.2. Maintenance tasks scheduling

MANTIS HMI allows

- the user to see all relevant maintenance tasks together with some additional information such as description of the task (including suggested time schedule), relevant production asset related information (e.g. sensor logs, maintenance history and statistics), guides, manuals or instructions for maintenance task, task progress information and client information.

It provides

- the input of task-related information, such as task acceptance or rejection, task progress (e.g. start and stop indication), assigning resources (e.g. necessary time and equipment);
- the input of asset-related information such as asset status, image of the failure (in case of failure), and feedback to the system (e.g. identification of the failure root cause, estimation of the actual wear-out);
- spare parts managing. This may include the inquiry of spare part availability, ordering spare parts and vendors contact information;

- maintenance tasks rescheduling (automatically, based on MANTIS maintenance optimization and manually by the user).

Maintenance display tasks

- enable filtering and sorting;
- are updated immediately after a new maintenance task is scheduled.

MANTIS HMI provides functionalities that

- generate automatic reports on maintenance activities and transfer the maintenance-related data to other users;
- display alert in case of production asset failure, or if the spare parts required for scheduled maintenance task are not available. The alert contains additional information, such as description of the failure, production asset status, or additional information on spare parts.

### 4.3. Data analysis

MANTIS HMI provides functionalities that

- give remaining useful life of the assets, predicted future values of monitored parameters, comparison between predicted and actual parameter values and feedback from other users in textual as well as graphical form;
- allow the user to manage prediction models. This includes model inspection, activation or deactivation of the model, updating, generating and evaluating predictions;
- display alert if the prediction performance of MANTIS system is below the predefined threshold. The alert includes additional information on prediction performance.

### 4.4. Reporting

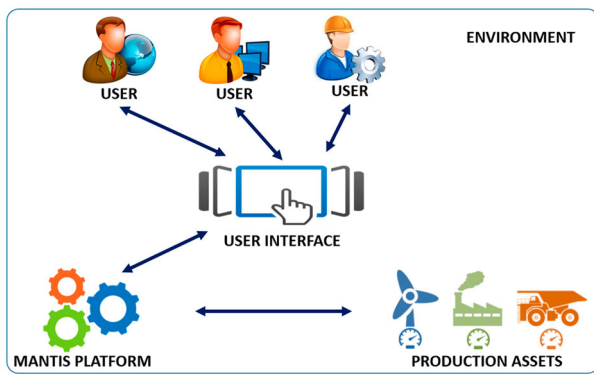
MANTIS HMI provides functionalities that

- generate automatic reports in pdf or html form;
- enable to process spoken reports;
- enable the user to manually generate reports. This includes information input (textual and graphical) and data export.

### 4.5. Communication

MANTIS HMI provides functionalities that

- support textual, visual and audial communication among the users;



**Figure 2.** Main parts of the MANTIS HMI.

- enable the transfer of different data sources, images, videos, and documents among the users as well as to and from the MANTIS platform.

## 5. HMI model

In the case of MANTIS, most of the user interface modalities identified when collecting the problem scenarios and deriving the above functional specifications follow conventional ways of graphical user interface interaction. On the other hand, MANTIS deals with a number of quite divergent physical systems, which may lead to diverse HMI solutions. It is therefore imperative to preserve a unique concept in their design in order to exploit the advanced features of the target proactive maintenance service platform in an optimal way. For this purpose, a generic HMI model (Figure 3) has been elaborated and is described in more details below.

### 5.1. General view

MANTIS HMI comprises five main elements (see Figure 2): user interfaces, users, MANTIS platform, production assets, and environment.

Through their user interfaces, several different users within the use case communicate with MANTIS platform, which in turn communicates with production assets. Interaction can take place in both directions. Users can not only access the information, retrieved from production assets and stored in the platform, but provide an input to the MANTIS system as well. They can initiate an operation which is then carried out by the platform, such as rescheduling maintenance task, or respond to a system-triggered operation, for example alarms. On the other hand, through the MANTIS platform, users can also communicate among themselves. In addition to the straightforward communication in terms of the textual or video chat functions, the users can also communicate via other established workflows.

The last but not least part of the interaction is the environment. Although it cannot be treated as a direct

link between the user and the system nor as a part of communication among the users, the environment can influence the HMI through the context-aware functionalities.

From the users' point of view, MANTIS HMI supports five main high-level user tasks associated to proactive and collaborative maintenance: monitoring production assets, maintenance tasks scheduling, data analysis, reporting and communication.

While monitoring production assets, maintenance tasks scheduling and data analysis are vital for proactive maintenance, reporting and communication enable collaboration among different user roles. Each of these tasks is carried out by a number of MANTIS-specific functionalities that can be classified as user input, system output, user- or system-triggered operation. These functionalities, together with the relations among them are described in detail in the following sub-sections. The described functionalities refer to the functional specifications listed above, and cover all the main aspects of MANTIS HMI. They are general enough to be applicable to any current MANTIS use case as well as to potential future use cases.

### 5.2. High-level tasks functionalities

#### 5.2.1. Monitoring

The most important functionality that supports the production assets monitoring is the real-time display of parameter values, measured by multiple sensors in the MANTIS system. User interface most commonly displays the actual current and historical parameter values as well as the predicted future parameter values. In most cases, the display of the current parameter values is changing with respect to the ab/normality of the parameter value. It is often required to display the expected (normal) range of the parameter values, the comparison between the predicted and actual parameter values or the remaining useful life of the asset, and various statistics of historical parameter values of the monitored production asset.

In addition to monitoring the parameter values, MANTIS HMI also displays possible failures of production assets together with some additional description, such as current and historical parameter values related to the faulty asset and possibly the feedback from other users.

Although user input is normally not required for monitoring of the production assets, MANTIS HMI enables the user to flag and comment the data and thus to provide additional information that might not be measured with the sensors.

In monitoring the production assets, it is of vital importance that the data are displayed in real time. The MANTIS HMI therefore frequently updates the parameter values. Another important system-triggered

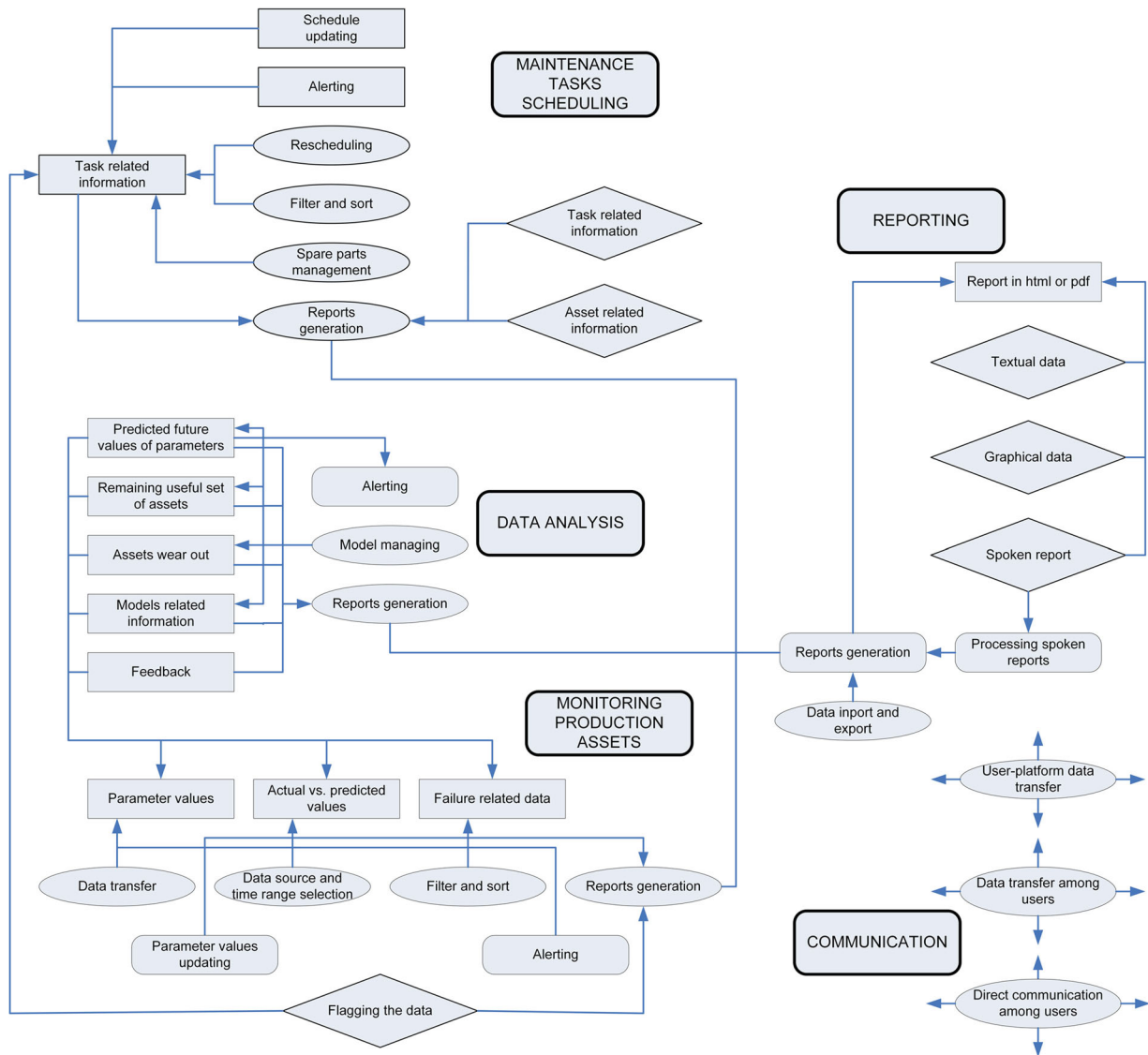


Figure 3. MANTIS HMI model.

operation is also the display of alerts if the monitored asset parameters are out of predefined range. The alert includes additional information on the monitored asset parameter such as historical values of the parameter. Both operations have influence on the display of parameter values and possible failures. To be precise, the update of the monitored parameter influences the values of the parameter, while the alarms usually influence the display of the parameter values as well.

To advance the navigation among different production assets and monitored parameters, MANTIS HMI enables the user to sort and filter monitored production assets, which allows the user to quickly find the information needed. It is also possible to select different data sources and the time range of monitored parameters, which makes the monitoring more flexible and tailored to the users' current needs.

Last but not least, the interface automatically generates reports on monitored parameters and transfers the monitored data to other users.

### 5.2.2. Maintenance tasks scheduling

The main functionality concerning scheduling maintenance tasks is the display of the maintenance tasks schedule, produced as a result of MANTIS system intelligent functions supporting proactive maintenance. The display of the maintenance tasks schedule allows the user to see all relevant maintenance tasks together with some additional information such as description of the task (including suggested time schedule), relevant production asset related information (e.g. sensor logs, maintenance history and statistics), guides, manuals or instructions for maintenance task, task progress information and client information. In addition to user input data, information displayed in the schedule is the main source of automatic report generation.

To follow the progress of maintenance activities, MANTIS HMI allows the user to input some of the task and critical asset-related information, such as task acceptance or rejection, task progress (e.g. start and stop indication), assigning resources (e.g. necessary



time and equipment), critical asset status, or image of the failure (in case of a failure). An important aspect of the user input is providing a feedback to the system (e.g. identification of the failure root cause, estimation of the actual wear-out, and similar), which may have a considerable impact on the improvement of predictive algorithms. The user feedback can be taken into account directly as an input to predictive models, or indirectly as a domain expert knowledge that can provide an important insight into the quality of predictive models.

When a new maintenance task is scheduled according to the maintenance tasks scheduling algorithms, the system triggers the schedule-updating operation. If the newly scheduled maintenance task is considered critical for the operation of production assets, the system may trigger an alert as well. These two system-triggered operations can influence the display of the maintenance task schedule by changing the schedule or/and by modifying the graphical display of the schedule, which happens mostly in the case of a critical maintenance task.

Operations such as manual rescheduling, filtering and sorting maintenance tasks, and spare parts managing can also be initiated by the user and have an effect on the maintenance task schedule display. Aside from these operations, user can trigger the automatic report generation as well. In this case, the system gathers task-related information and the user input to generate a report in pdf, html or any other desirable format.

### 5.2.3. Data analysis

As the data analysis is one of the crucial tasks in proactive maintenance, it is important that it is supported by the MANTIS HMI. Although in most cases data analysts use software, specialized in data analysis, it is useful to have additional user interface to reduce the time of frequent tasks or to allow users who might not be specialized in data analysis to perform basic operations.

MANTIS HMI therefore displays production assets wear-out, remaining useful life of the assets, and predicted future values of monitored parameters. To evaluate the performance of the predictive algorithms the interface also displays the comparison between predicted and actual parameter values and feedback from other users in textual and/or graphical form.

The user is also able to some extent to manage the prediction models. This includes model inspection, activation or deactivation of the model, as well as updating, generating and evaluating predictions. In addition to the influence that model management has on the display of different parameter values in scope of data analysis, applying a model has a significant impact on every aspect of the proactive maintenance. In case of applying a new model, the predicted parameter values and estimated remaining life of the production assets are automatically updated and the maintenance tasks

are rescheduled. If the new estimation of the remaining useful life of the asset is lower than the previous one, this might also trigger some possibly indispensable alarms.

Beside the model management, the user is also able to trigger automatic report generation. The report contains the information displayed on the data analyst's user interface, and optionally the description of the used models.

The prediction performance of MANTIS system can be estimated from the comparison of the predicted and actual parameter values or the feedback from users working on the field. If the performance is below the predefined threshold, MANTIS HMI displays an alert with additional information on prediction performance.

### 5.2.4. Reporting

When the users with different roles in maintenance process trigger report generation, the system provides a means to produce the report in pdf, html or other required formats. The report contains all the relevant information related to the tasks the user is performing and the input that the user has provided.

In addition to automatically generating the content of the report, the MANTIS HMI allows the users to input any additional information, either by means of importing the data from different data sources or manually input textual or graphical information.

Since reporting is more of a by-product than a vital part of the maintenance process, it is important to reduce the workers' time and effort dedicated to this task. The system is therefore planned to process spoken reports, which is especially beneficial for maintenance technicians in the field.

### 5.2.5. Communication

An important aspect of proactive and collaborative maintenance is the communication not only between the user and the system but also among different users. Enhanced communication not only boosts working productivity but also helps to avoid human mistakes caused by misunderstanding.

## 6. Comparison with similar systems

The HMI model, as a common point, is elaborated in different papers from different perspectives, with different levels of details and different goals. However, due to the extensive and divergent MANTIS use-cases, reports on similar systems are relatively few.

Scenario-based modelling is addressed in [20] on simple examples, which can serve primarily for illustrating purposes rather than dealing with the concrete problems exemplified in MANTIS. We also do not share the experience of the authors who state that

scenario-based modelling is no more than a requirements generator. In our case, it proved to be an effective means for HMI design from the early activity scenarios through information and interactive scenarios up to prototype design and usability testing.

The integrated scenario-based design methodology described in [21] provides an integrated approach combining a vision of potential users, business aspects and technological challenges throughout the design process. The proposed method has been developed and implemented in research and development projects ADAMOS, e-SENSE and SENSEI and has as a final goal development of a new set of scenarios that are reflecting the technology innovation, the business opportunities and user benefits. While focusing on a completely different subjects, it has common points with MANTIS in the way the scenario-based design process that combines three high-level phases (scenario elaboration, application requirements and acceptance studies) is elaborated. The advantage of [21] is that the approach is evaluated through *Conception Assistée par l'Usage pour les Technologies, l'Innovation et le Changement* method (User Oriented Design for Technology, Innovation and Change) which is a sociological qualitative method for investigating the user experience that is shaped to the study of user and social acceptance of innovative services and applications. Due to the tough time schedule the application of such evaluation methods was not possible in our case. On the other hand, the scenario-based design process in MANTIS includes more detailed functional analysis resulting in extensive high-level task functionalities which are the basis of the developed HMI model.

In [22], the implementation of a method capable of automatizing the process of test cases construction under a perspective of usability tests and human factors is proposed. It is performed through business requirements documentation based on scenarios. The method execution cycle has common points with the approach employed in MANTIS, however, the main difference is in the scenario requirements elicitation. Furthermore, the proposed method does not incorporate extraction of tasks functionalities, which is one of the main features of the proposed MANTIS formal framework.

## 7. Further steps

The described HMI model is general enough to cover all MANTIS use cases. Since they are quite divergent only a part of the above functionalities actually take place in an individual use case. The next steps for each use case are therefore as follows:

- Elaboration of requirement specifications. Numerous project reports including requirement specifications available on the web can serve as example. However, a recommended practice is to stick

to a standard. In our case, a template from ISO/IEC/IEEE 29148:2011 was adapted.

- Identification of content elements. Elaborated requirement specifications contain all content elements of a given use case, together with specified inputs/outputs and associated functions.
- Application of HMI model. Identified content elements together with their input/output connections are activated in the HMI model. All remaining parts of the model are omitted.
- Implementation of HMI prototype that with identified content elements and their functionalities, allows the users to perform their tasks to achieve their goals. This is done via an iterative process of design and usability testing.
- Usability testing. Exploratory test is made in an early design phase to find possible shortcomings early enough before the first version of HMI is actually built. Assessment test is made when the basic HMI functionalities are implemented but not yet optimized. The purpose is to help use-case owners and HMI-developing partners to improve their implemented HMIs towards the final version. Validation is the final testing phase. It is used to measure the efficiency of the implemented HMI. In our work, we followed the guidelines provided in [23–25]. Both assessment and validation tests are based on scenarios refined from those provided for identification of MANTIS functionalities. Detailed description of this step is beyond the scope of the paper.

## 8. Conclusions

Scenario-based approach proved to be an efficient way to collect essential information regarding HMI requirements of individual use cases, to identify the related content elements, as well as to design and implement usability tests of the HMI prototype. The developed HMI model allows us to conceive the HMI framework of a given use case and to define its particular content elements, their functionalities and input/output data at the desired level of details both at early design phase and at implementation. The framework can serve also other purposes, like for example, to identify the content elements involved in context awareness issues and provide basis for analysing the developed solutions. While the whole approach originated from the set of MANTIS use cases, the resulting framework including functional specifications and HMI model is general and can be used in other use cases in system maintenance.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This work was partially funded from the Electronic Component Systems for European Leadership Joint Undertaking

(ECSEL JU) under grant agreement No 662189 (MANTIS). This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Finland, Denmark, Belgium, Netherlands, Portugal, Italy, Austria, United Kingdom, Hungary, Slovenia, Germany. The authors also acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0098).

## ORCID

Gregor Papa  <http://orcid.org/0000-0002-0623-0865>

## References

- [1] MANTIS Project [Internet]. Mondragon, Spain: Mondragon Goi Eskola Politeknikoa; [cited 2017 Oct 3]. Available from: <http://www.mantis-project.eu/>
- [2] Boy GA. The handbook of human-machine interaction: a human-centered approach. Aldershot: Ashgate; 2011.
- [3] Sears A, Jacko JA. The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. 2nd ed. New York (NY): Taylor & Francis; 2006. (Human factors and ergonomics).
- [4] Rosson MB, Carroll JM. Usability engineering. Scenario-based development of human-computer interaction. San Francisco (CA): Morgan Kaufmann; 2002.
- [5] Bate I, Emberson P. Incorporating scenarios and heuristics to improve flexibility in real-time embedded systems. In: Goddard S, Liu JC (Steve), editors. Proceedings of the Twelfth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06); 2006 Apr 4–7; San Jose (CA), USA. Washington, DC: IEEE Computer Society; 2006. p. 221–230.
- [6] Carroll JM. Making use – scenario-based design of human-computer interactions. Cambridge (MA): Massachusetts Institute of Technology; 2000.
- [7] Ahmed DM, Sundaram D, Piramuthu S. Knowledge-based scenario management—process and support. *Decis Support Syst.* 2010;49(4):507–520.
- [8] Uchitel S, Kramer J, Magee J. Incremental elaboration of scenario-based specifications and behavior models using implied scenarios. *ACM Trans Softw Eng Methodol.* 2004;13(1):37–85.
- [9] Limbourg Q, Vanderdonck J. Comparing task models for user interface design. In: D Diaper, N Stanton, editors. The handbook of task analysis for human-computer interaction. Mahwah, NJ: Lawrence Erlbaum; 2003. Chapter 6; p. 135–154.
- [10] Meixner G, Seissler M. Selecting the right task model for model-based user interface development. Proceedings of ACHI 2011, the Fourth International Conference on Advances in Computer-Human Interactions; 2011. p. 5–11.
- [11] Pastor O, España S, Panach JI, et al. Model-driven development. *Inform Spektrum.* 2008;3(5):394–407.
- [12] Kraiss KF, editor. Advanced man-machine interaction. Berlin: Springer; 2006.
- [13] Cahill J, Geary U, Douglas E, et al. User and design requirements and production of evidence: using incident analysis data to (1) inform user scenarios and bow ties, and (2) generate user and design requirements. *Cogn Technol Work.* 2018;20(1):23–47.
- [14] Noori NS, Wang Y, Comes T, et al. Behind the scenes of scenario-based training: understanding scenario design and requirements in high-risk and uncertain environments. In: Comes T, Bénaben F, Hanachi C, Lauras M, Montarnal A, editors. Proceedings of the 14th ISCRAM Conference; 2017 May 21–24; Albi (France). Brussels: International Association for Information Systems for Crisis Response and Management; 2017. p. 948–959.
- [15] Ko D, Park S, Kim Y, et al. Suggesting alternative scenarios using use case specification patterns for requirement completeness. *Int J Softw Eng Knowl Eng.* 2016;26(6):927–951.
- [16] Shiota E, Ohnishi A. Scenario retrieval method using differential scenario. *IEICE Trans Inf Syst.* 2016;99(9): 2202–2209.
- [17] Vincent CJ, Blandford A. Usability standards meet scenario-based design: challenges and opportunities. *J Biomed Inform.* 2015;53(Feb):243–250.
- [18] Reeder B, Turner AM. Scenario-based design: a method for connecting information system design with public health operations and emergency management. *J Biomed Inform.* 2011;44(6):978–988.
- [19] Rauf I, Siavashi F, Truscan D, et al. Scenario-based design and validation of REST web service compositions. In: Monfort V, Krempels K, editors. Proceedings of the International Conference on Web Information Systems and Technologies; 2014 Apr 3–5; Barcelona (Spain). Cham: Springer; 2015. p. 145–160.
- [20] Petkovic D, Raikundalia GK. An experience with three scenario-based methods: evaluation and comparison. *Int J Comput Sci Netw Secur.* 2009;9(1):180–185.
- [21] Forest F, Lavoisy O, Chanal V. Integrated scenario-based design methodology for collaborative technology innovation. Proceedings of the Future of Innovation Conference; 2009 June 21–24; Vienna (Austria). p. 99.
- [22] Panicucci B, de Castro P, Aquino PT. Human factors aspects in test cases formalization. *Procedia Manuf.* 2015;3:1938–1945.
- [23] Endsley J, Rubin J, Chisnell D. Handbook of usability testing. Indianapolis (IN): Wiley; 2008.
- [24] Usability.gov [Internet]. Washington (DC): Department of Health & Human Services; 2017 [cited 2017 Oct 3]. Available from: <https://www.usability.gov/>
- [25] Usability Net. [Internet]. London: Serco Usability Services; 2006 [cited 2017 Oct 3]. Available from: <http://www.usabilitynet.org/tools.htm>