6

From KPI Dashboards to Advanced Visualization

Goreti Marreiros[1], Peter Craamer[2], Iñaki Garitano[3], Roberto González[4], Manja Gorenc Novak[5], Aleš Kancilija[5], Quinten Marcelis[6], Diogo Martinho[1], Antti Niemelä[7], Franc Novak[8], Gregor Papa[8], Špela Poklukar[8], Isabel Praça[1], Ville Rauhala[7], Daniel Reguera[3], Marjan Šterk[5], Gorka Urchegui[2], Roberto Uribeetxeberria[3], Juha Valtonen[7], and Anja Vidmar[5]

[1]ISEP, Polytechnic Institute of Porto, Porto, Portugal
[2]Mondragon Sistemas De Informacion, Spain
[3]Mondragon Unibertsitatea, Arrasate-Mondragón, Spain
[4]Tekniker, Spain
[5]XLAB, Ljubljana, Slovenia
[6]Ilias Solutions, Belgium
[7]Lapland University of Applied Sciences Ltd, Finland
[8]Jožef Stefan Institute, Slovenia

New technologies are being developed towards Industry 4.0 such as the establishment of smart factories, smart products and smart services embedded in an internet of things and of services. As a result, the development of prognostic and collaborative technologies have become a necessity. New technological solutions that can make the best use of existing physical assets while following company's important metrics to provide a quick overview of business performance are now starting to be developed. Examples of such solutions include the development of dashboards that can show the Key Performance Indicators (KPI) for processes involved in the production site and Human-Machine Interfaces (HMI) that allow the visualization of data collected directly from the machine in different formats such as graphic, table or even through augmented and virtual reality, while their adaptive interfaces can present relevant information according to the user and the context.

This chapter explores these new technological solutions in the context of the MANTIS project and consists of four main sections. The first section presents the HMI technology and the specifications, design principles and recommendations, requirements, and modelling of the HMI that guides and follows the principles of the MANTIS Reference Architecture (see Chapter 2). The second section discusses the concept of adaptive interfaces and the two main approaches that were considered (Context-aware and Interaction based/driven). The third section discusses advanced data visualization methods for HMIs and presents different scenarios according to the different use cases defined in MANTIS. In the last section, the usability testing methodology considered for industrial HMIs is discussed.

6.1 HMI Functional Specifications and Interaction Model

Human-machine interaction denotes real-time interaction and communication between human users and a machine via a human-machine interface [Techopedia, 2011]. 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. HMI for proactive maintenance should therefore contribute to:

- enhanced monitoring of shop-floor conditions (i.e., the machines health condition, the efficiency of the production lines, and the safety of workers);
- automatic self-adaptation of control strategies based on the context related to user status, machine status, general environment and time;
- user-friendly, ergonomic and intuitive interaction between workers and machines, and consequently positive user motivation leading to higher efficiency and safety.

While MANTIS strongly emphasises autonomy, self-testing and self-adaptation, human role remains one of the important factors in system operation. The human role 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. The increased degree of automation in control of dynamic technical systems does not replace the human users, but rather modifies the interaction between both. Appropriate matching of both leads to a user-centred design.

To propose a user-centred design in the project comprising eleven use cases from four different sectors transpired to be a challenging task. Diversity of MANTIS use cases resulted in a wide range of requirements that could hardly fit a common MANTIS HMI structure. Design and development of a common MANTIS HMI is not only difficult but also most likely to result in a poor usability of the products. One of the goals of the MANTIS project is therefore to offer a common ground for designing user-centred, usable and use case specific HMIs. The goal has been achieved by developing the human-machine proactive maintenance-based interaction model that on one hand covers all proactive maintenance related requirements of every use case and stays general enough to be applicable to every MANTIS and potential future use case. The MANTIS HMI model has been conceived to provide means that would help to identify the HMI content elements and their relationships of a given use case. Together with the functional specification, described later in this section, it may serve as a reference point for writing use case specific requirements specifications and for designing the user interaction.

6.1.1 HMI Design Principle Followed in the MANTIS Project

MANTIS followed Scenario-Based Design (SBD) [MANTIS Consortium, 2016] which is an established approach for describing the use of a system at an early point in the development process. Narrative descriptions of the envisioned scenarios help to guide the development of the system and serve, among others, as a basis for setting efficient human-machine interaction. Use case owners described some typical problem scenarios and refined them through the activity, information and interaction phase and as a result to provide the scenarios that would include sufficient details for HMI prototyping [MANTIS Consortium, 2016].

Scenarios gathered in this document are the result of iterative process of the SBD phase, which is reflected in their common structure:

- situation that describes the circumstances in which the scenario occurs, focused on the perspective as seen by the user;

- device which holds the interaction, most suitable for the nature of the user's activity;
- information, available for the user (*What options are available in the interface*);
- the way user interacts with the interface (*How is it done*);
- current implementation.

The activity scenarios describe pure functionality of HMI. They have been refined to information scenarios through the *What options are available in the interface* section of each scenario and further particularized to interaction scenarios through the *How is it done* section. This way the scenarios are elaborated to the point where they provide the details of user action and feedback.

A common structure of each scenario intended to unify the diversity of requirements, imposed by the wide range of distinct use cases, and to gather functional specifications, common to all use cases and specific for proactive maintenance.

6.1.2 MANTIS HMI Specifications

To provide the right information, in the right modality and in the best way for users when needed, the user interface should be highly personalized and adapted to each specific user or user role. Any unification of the HMI design might impose the constraints that could result in an HMI with a poor usability.

The approach, adopted in the MANTIS project, therefore focuses on the requirements, common to most of the use cases and specifics for proactive and collaborative maintenance. In the following section, 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.

6.1.2.1 Functional specifications

The specifications provided in this section, are the result of refinement of scenarios, provided by industrial partners. Functional specifications describe the HMI functionalities 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 may serve as a reference point when writing ones.

Scheduling of Maintenance Tasks

- 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 asset related information (e.g., sensor logs, maintenance history and statistics), guides, manuals or instructions for maintenance task, task progress information and client information;
- MANTIS HMI allows adding 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);
- MANTIS HMI allows adding 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, . . .);
- MANTIS HMI allows spare parts managing. This may include the inquiry of spare part availability, ordering spare parts and vendors contact information;
- MANTIS HMI allows maintenance tasks rescheduling (automatically, based on MANTIS maintenance optimization and manually by the user);
- Maintenance tasks display enables filtering and sorting;
- Maintenance tasks display is updated immediately after a new maintenance task is scheduled;
- MANTIS HMI is able to automatically generate reports on maintenance activities and to transfer the maintenance related data to other users;
- MANTIS HMI displays an alert in case of 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, asset status, or the additional information on spare parts.

Monitoring Assets

- MANTIS HMI displays current, historical and predicted parameter values of monitored assets and expected range of these parameters;
- MANTIS HMI displays comparison between actual and estimated asset wear and/or predicted remaining useful life;
- MANTIS HMI displays various statistics of historical parameter values of monitored assets;
- MANTIS HMI displays possible failures of assets together with some additional description, such as current and historical parameter values related to the faulty asset and possible feedback from other;

- MANTIS HMI allows the user to sort and filter monitored assets, select different data sources, and to select time range of monitored parameters;
- MANTIS HMI allows the user to select and flag the data;
- MANTIS HMI is able to automatically generate reports on monitored parameters and to transfer the monitored data to other users;
- MANTIS HMI displays data in real time;
- MANTIS HMI displays an alert if the monitored asset parameter is out of a predefined range. The alert carries additional information on the monitored asset parameter such as historical values of the parameter.

Data Analysis

- MANTIS HMI is able to display 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;
- MANTIS HMI allows the user to manage prediction models. This includes model inspection, activation or deactivation of the model, updating, generating and evaluating predictions;
- MANTIS HMI displays an alert if the prediction performance of MANTIS system is below the predefined threshold. The alert will carry additional information on prediction performance.

Reporting

- MANTIS HMI is able to generate automatic reports in pdf or html format;
- MANTIS HMI is able to process spoken reports;
- MANTIS HMI allows the user to manually generate reports. This includes information input (textual and graphical) and data export.

Communication

- MANTIS HMI supports the textual, visual and audial communication among the users;
- MANTIS HMI enables the transfer of different data sources, images, videos, and documents among the users as well as to and from the MANTIS platform.

6.1.2.2 General requirements

The following general requirements have also been identified by industrial partners and must require that:

- MANTIS HMI performs just-in-time;
- MANTIS HMI supports different types of devices, including mobile phones, tablet computers, laptop computers, network computers and machine displays;
- MANTIS HMI supports at least the most commonly used operating systems such as Windows, Android, and IOS;
- MANTIS HMI supports multiple levels of alerts;
- MANTIS HMI allows user's input such as confirmation of the alert and providing feedback to the intelligent MANTIS features;
- MANTIS HMI is able to automatically generate reports on alerting activities and to transfer the maintenance related data to other users.

6.1.3 MANTIS HMI Model

As stated in [Techopedia, 2011], human-machine interface (HMI) is a component of certain devices that are capable of handling human-machine interactions [Boy, 2011]. The interface consists of hardware and software that allow user inputs to be translated as signals for machines that, in turn, provide the required result to the user. Since HMI technology is ubiquitous, the interfaces involved can include motion sensors, keyboards and similar peripheral devices, speech-recognition interfaces and any other interaction in which information is exchanged using sight, sound, heat and other cognitive and physical modes are considered to be part of HMIs.

The initial phase of scenario-based design approach commonly applied for all MANTIS use cases resulted in an extensive set of divergent scenarios, which required considerable additional activities to get the descriptions suitable for HMI design.

In the following we describe a generic static model that can be used together with the requirement specifications of each individual use case to formalize the structure of the target HMI implementation. The model has been conceived, in particular, with two ideas in mind: to provide means that would help to identify the HMI content elements and their relationships of a given use case and to unify (as much as possible) the HMI design of different use cases, which would be useful for comparison of implementations and exchange of good practices. When setting up the model structure we follow the concepts of descriptive models applied in task analysis [Diaper, et al., 2004] and add specifics of MANTIS gathered maintenance scenarios, denoted as MANTIS high level tasks. For each of these high level tasks we provide a list of functionalities supporting the given high level task. In addition to the model, a Requirements Specification template was used to identify the HMI content elements and their functionalities supporting the high level tasks of a given use case.

MANTIS human-machine interaction comprises five main elements (Figure 6.1):

- user interfaces;
- users;
- MANTIS platform;
- assets;
- environment.

MANTIS platform allows communication with several different users through their user interface. Interaction between the users and the platform is bidirectional; users can not only access the information retrieved from assets and stored in the platform, but can also provide an input to the MANTIS system. In case of proactive maintenance, user feedback is especially valuable since it is providing additional input to the prediction algorithms. Users can also initiate an operation which is then carried out by the platform, such as rescheduling maintenance task, or respond to a system triggered operation such as handling an alarm. Another aspect of the user interaction is the communication between different users through the MANTIS platform. In addition to the straightforward communication in terms of the textual or video

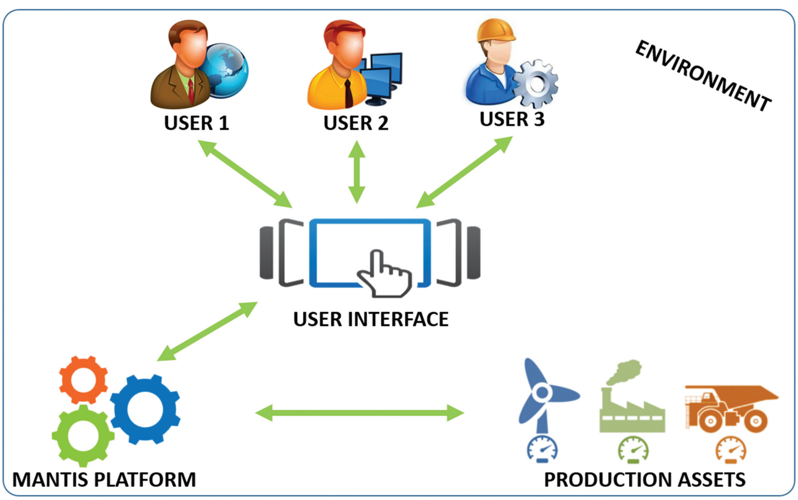


Figure 6.1 MANTIS human-machine interaction.

chat functions, the users can also communicate via established workflows or shared widgets.

Although environment cannot be treated neither as a direct link between the user and the system nor as a part of communication between the users, it can significantly improve the human-machine interaction in terms of context-aware functionalities.

Proactive and collaborative maintenance oriented human-machine interaction within the MANTIS system supports five main high-level user tasks:

- monitoring assets;
- data analysis;
- maintenance tasks scheduling;
- reporting;
- communication.

These tasks were identified as the key user tasks in the initial process of user-centred design. Monitoring the assets, data analysis and maintenance task scheduling proved to be vital for proactive maintenance, while reporting and communication allow collaboration between different user roles. Tasks can be carried out using several MANTIS specific functionalities that can be classified as user input, system output, user- or system- triggered operation. Functionalities, described later in this section, cover all the main aspects of MANTIS human-machine interaction and are general enough to be applicable to any MANTIS or potential future use case.

6.1.3.1 Functionalities supporting high level tasks

A more detailed description of each functionality previously identified is now provided including the expected Output, Input and/or related Operations.

Scheduling Maintenance Tasks

Output:

- Maintenance tasks schedule together with some additional information for each task, such as:
    - description of the task (including suggested time schedule);
    - relevant asset related information (e.g., sensor logs, maintenance history and statistics);
    - task progress information;
    - client information;
    - spare parts related information (availability, vendor information).

Input:

- Input of task related information, such as:
    - task acceptance/rejection;
    - task progress (e.g., start and stop indication);
    - assigned resources (e.g., necessary time and equipment).
- Input of asset related information:
    - asset state information (e.g., description, image, failure description);
    - feedback to the system (e.g., identification of the failure root cause, estimation of the actual wear).

Operations:

- User-triggered operations:
    - task rescheduling;
    - filtering and sorting maintenance tasks;
    - report generation (automatic or manual);
    - spare parts management.
- System-triggered operations:
    - update when new task is scheduled;
    - alert (when new task is scheduled, spare parts are not available, etc.).

In scheduling maintenance tasks, the display of maintenance tasks schedule, produced as a result of MANTIS system intelligent functions, is the most important functionality. It offers the overview of all relevant maintenance tasks and provides them with additional, task related information such as task description (including suggested time schedule), relevant asset related information (e.g., sensor logs, maintenance history and statistics), maintenance personnel support such as guides, manuals or instructions for a maintenance task, task progress information and client information. Information, displayed in the schedule is, together with the user input data, one of the main sources for automatic report generation.

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 failure). Such input is important for monitoring the maintenance activities' progress and especially for providing feedback to the system (e.g., identification of the failure root cause, estimation of the actual wear, etc.), which may have a considerable impact on improvement of predictive algorithms. The user feedback may be taken into account in two ways. It can serve as a direct input to predictive models, or it can be used indirectly as a domain expert knowledge that can provide an important insight in the quality of predictive models.

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

Manual rescheduling, filtering and sorting maintenance tasks, and spare parts managing are user-initiated operations and can affect the display of the maintenance task schedule. In addition to these operations, users can trigger the automatic report generation. In response to this user action the system gathers tasks related information and the user input to generate a report in any desired format.

Monitoring

Output:

- Current, historical and predicted parameter values of monitored assets, together with the expected range of monitored parameters;
- Comparison between actual and estimated wear or predicted remaining useful life;
- Possible failures of assets together with some additional description (e.g., current and historical parameter values related to the faulty asset, possible feedback from other users).

Input:

- Flagging the data.

Operations:

- User-triggered operations:
    - sort and filter monitored assets;
    - select different data sources;
    - select time range of monitored parameters;
    - generate reports on monitored parameters;
    - transfer the monitored data to other users.

- System-triggered operations:
    - continuous updating of the monitored parameters;
    - alert when monitored asset parameter out of predefined range.

The most common assets monitoring related functionality is definitively the real time display of parameter values, measured by multiple sensors in the MANTIS system. Amounts, displayed on the user interface, vary from the actual current and historical parameter values to the predicted future parameter values. In case of abnormal values of these parameters, the interface can adapt the display to alert the users. It is often required to display the expected (normal) range of the parameter values, the comparison between the predicted and actual parameter values or remaining useful life of the asset, and various statistics of historical parameter values of the monitored asset.

Other monitoring features include the display of the possible assets failures 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 the user input is typically not required for monitoring itself, MANTIS HMI should allow the user to flag, label and comment the data. In this way, the users can provide the additional data that might not be captured by the sensors.

Real time display of information is often vital for an efficient maintenance process which means that the MANTIS HMI should be able to frequently update the parameter values. Also, the display of alerts if the monitored asset parameters are out of predefined range is another important system-triggered operation. It is often helpful if the alert carries some additional information related to the monitored asset parameter such as historical values of the parameter. Both operations have influence on the display of parameter values and the display of possible failures. While an update of the monitored parameter changes the values of the parameter itself, alarms or alerts have influence only on the display of the parameter values.

MANTIS HMI should allow the users to sort and filter monitored assets to advance the navigation among different assets and monitored parameters. To make the monitoring more flexible and tailored to the users' current needs, the interface should allow the selection of different data sources and the time range of the monitored parameters.

Finally, the interface should be able to produce automatically generated reports that include various information about the monitored parameters and to transfer the monitored data to other users.

Data Analysis

Output:

- Assets wear;
- Remaining useful life of the assets;
- Predicted future values of monitored parameters;
- Comparison between predicted and actual parameter values;
- Feedback from other users.

Operations:

- User-triggered operations:
  - Prediction models management:
    - model inspection;
    - activation or deactivation of the model;
    - updating, generating and evaluating predictions.
  - Report generation.
- System-triggered operations:
  - Alert when prediction performance of MANTIS system is below the predefined threshold.

Since the data analysis is one of the key tasks in proactive maintenance, it is important that it is supported by the MANTIS HMI. In most of the MANTIS use cases, data analysts are already using various software. However, in order to reduce the time of frequent tasks it might still be useful to have an additional user interface. Such interface can also allow the users that are not specialised in data analysis to perform some basic data analysis operations such as displaying some basic statistics or choosing between predefined models.

Displaying the production assets wear, remaining useful life of the assets, or predicted future values of monitored parameters can therefore represent valuable features of the MANTIS HMI. Also, the of comparison between predicted and actual parameter values and feedback from other users in textual and/or graphical form can aid in evaluating the performance of the predictive algorithms.

To some extent, the users can also be able to manage the prediction models. These functionalities are usually limited to model inspection, activation or deactivation of the model, and updating, generating and evaluating predictions. Manipulation of the prediction models influences not only on the display of different parameter values in scope of data analysis, but does also have a significant impact on every aspect of the proactive maintenance. Designers of such interfaces should pay a special attention to the automatic update of the predicted parameter values and estimated remaining useful life of the assets in case of applying a new model. Usually, the maintenance tasks should be rescheduled as well. 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.

The results of the model management should be reported with the use of automatic report generation feature of the MANTIS HMI. The report should contain the information, displayed on the data analyst's user interface, and optionally the description and interpretation of the used models.

The prediction performance of the MANTIS system can be estimated from the comparison of the predicted and actual parameter values or the feedback from the users working on the field. If the performance is below the predefined threshold, MANTIS HMI should display an alert with additional information on prediction performance and send the relevant data to other users whose work is influenced by these models.

Reporting

Output:

- Report in pdf or html format.

Input:

- Textual information input;
- Graphical information input;
- Spoken report.

Operations:

- User-triggered operations: data importing and exporting;
- System-triggered operations: processing spoken reports.

Generating reports can be triggered manually by the user or automatically by the system on regular bases. If a user triggers the report generation, the system should be able to produce the report in any required format, most commonly: pdf or html. The report should contain all the relevant information related to the maintenance process, assets and the input that the user has provided.

In addition to the content that is automatically generated by the system, the MANTIS HMI should allow 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.

Although the reporting is more of a by-product than a vital part of the maintenance process, it can significantly reduce the workers' time and the effort dedicated to this task. Some advanced reporting features include process the spoken reports, which could especially benefit the maintenance technicians on the field.

Communication

Operations:

- User-triggered operations:
  - textual, visual and audial communication between the users;
  - transfer of different data sources, images, videos, and documents among the users;
  - transfer of different data sources, images, videos, and documents to and from the MANTIS platform.

Communication is an important aspect of proactive and collaborative maintenance. Communication is not present only between the user and the system but also between different users. Enhanced communication not only boosts the working productivity but also helps to avoid the human mistakes caused by misunderstanding. MANTIS HMI should therefore support both direct messaging in form of chat or video and indirect communications via shared widgets or established workflows.

For each of these high level tasks we provide a list of functionalities supporting given high level task (Figure 6.2). In addition to the model, we provide a Requirements Specification template, which can serve for identification of the HMI content elements and their functionalities supporting the high level tasks of a given use case. More details are also available in [Poklukar et al., 2017].

6.1.4 HMI Design Recommendations

Functional specification and thereby derived interaction model, described in the previous section, cover the functional aspect of the MANTIS HMI. In order for resulting interfaces to be intuitive and easy to use, the established design principles should be applied. Regardless of technical implementation choices, design recommendations and guidelines can be provided that facilitate HMI implementations to fulfil the requirements of maintenance-based user interaction and offer a good user experience.

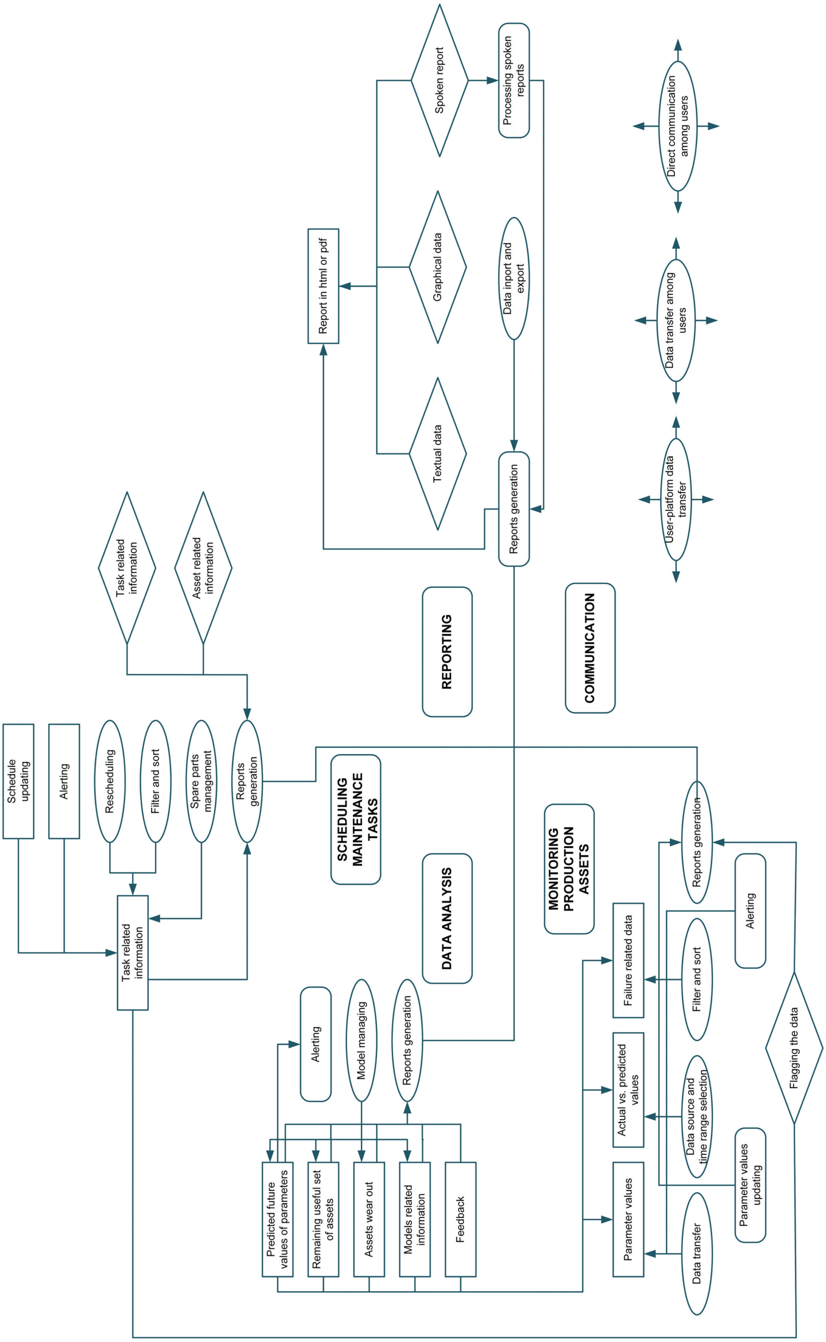


Figure 6.2 MANTIS HMI model.

The first step towards a personalised interface that enables the user's maximum efficiency is the definition and design of different HMI representation types, associated with different scenarios, users (workers, administrators, services, managers, etc.), and platforms. The selection of the HMI representation types was accomplished by extracting the requirements of interaction scenarios from the MANTIS use cases. All use cases require the use of PC interfaces (desktop/laptop) and most require mobile devices (smartphone/tablet). A few need to use industrial PC consoles or specialized external devices as well.

A central feature of most use cases is displaying and responding to alarms and monitoring processes by displaying data in the forms of tables, charts or graphs. Some scenarios also require the interface to display maintenance plans and guides on how to perform tasks, to exchange messages with other users, and/or to input reports on the performed maintenance tasks. Non-textual input is required by some use cases, such as sound and video recording and text-to-speech and speech-to-text.

Based on user interface design best practices, but focussed on collected MANTIS requirements, provided design guidelines strive to find a balance of being specific enough to add value over generic UI design literature, while still being applicable to most MANTIS use cases. They are also applicable to industrial maintenance use cases outside of MANTIS. The guidelines relate to the structure of the interface (a levels-based screen hierarchy and navigation between the screens), layout of the screen elements, visual design (general appearance of the interface, colour selection, etc.), and data representation. Furthermore, interactions and collaboration between users are addressed, for example managing alarms and events, creating reports, and communicating via messaging and chat.

The aspect of supporting different HMI platforms is taken into account by analysing their specifics and differences. We argue that for most interaction scenarios all platforms can be supported adequately by implementing the HMI as a web application. Responsive design should be used to adapt automatically to the screen size, input capabilities, and other specifics of the particular device a user chooses to employ. To this end, we recommend following the model-view-controller architecture for web application development. We also briefly list some of the popular web development back- and front-end frameworks that could serve as a basis for implementations. Finally, some suitable libraries for data visualisation are also listed, e.g., for graphs and charts.

6.1.5 MANTIS Platform Interface Requirements

The definition and design of different HMI representation types is associated with different scenarios and different users, e.g., workers, administrators, services, managers, etc. This includes defining the modalities of adaptive user interfaces in order to setup a context sensitive monitoring environment as well as taking the aspect of supporting different HMI platforms into account, e.g., web, mobile. The maintenance personnel should get mobile and easy-to-use extensions to the existing industrial dashboards (usually) on fixed-position screens. Maintenance procedures are presented via user-friendly, ergonomic and intuitive human-machine interactions, which might include the use of monitors, cameras and other HMI-specific sensors.

The devices that can be used for human-machine interaction range from general-purpose input- and output-capable devices, either static (desktop PCs), portable (laptops), or mobile (tablets, smart phones). Certain scenarios also include special-purpose PCs integrated into industrial equipment, as well as special purpose output (line monitors) and input devices (e.g., GoPro cameras).

Preferably each MANTIS HMI implementation should be usable on PCs and mobile devices, rather than requiring two separate implementations. The two most prominent interface elements are alarms and guides. Alarms play an important role in cyber-physical systems in general and in their maintenance in particular. Guides need to be taken into account when the operators need to follow the prescribed maintenance processes or machine failure interventions. The other elements are more generic; however, numerous design guidelines and examples can be provided to facilitate the implementation of HMI that can fulfil the objectives of MANTIS.

6.1.5.1 Analysis of different interface types

User interfaces can be divided into many types based on the input and output devices that a user can interact with (Wikipedia). The most traditional is the combination of keyboard and mouse input with screen output. Touchscreens serve as both input and output. Additional types of interaction can be facilitated with audio (microphones for voice recording or recognition, speakers or headphones for voice synthesis or playback of pre-made sounds) or video devices (cameras). Finally, pointing devices (mouse, touchscreen, trackballs, user's hand recognized by a camera, etc.) can be used to recognize user's gestures rather than as pointing to a particular object on the screen.

Types of user interfaces were strictly divided in the past, but nowadays trends in design are more oriented into combining different existing types, exploring new ones, to enable most natural interaction between user and machine (natural user interfaces) [Wigdor, et al., 2011].

Smartphones and tablets have become two most commonly used types of mobile smart devices in everyday life. Most of the users are therefore familiar with their UI and should not need any device-specific training.

Smartphones and tablets run various operating systems (most commonly, Android, iOS or Windows platforms are used) with a graphical user interface. Mobile devices offer a wide variety of additional hardware characteristics, which can be used as a part of application HMI, e.g., playing sound for alarms. The camera has become an almost essential part of mobile devices. Every device is also equipped with a microphone. Mobile devices connect to networks wirelessly, either to local networks via Wi-Fi or to the internet via mobile networks. Because of the size and hardware characteristic, mobile devices can be brought almost everywhere [Dunlop, et al., 2002].

The characteristics of the smartphones and tablets enable the use of different HMI types. The basic type of a mobile HMI is the graphical user interface, most commonly using a touchscreen as a combined input and output device. A first type of advanced HMI is the gesture interface, which enables inputs made in a form of hand or stylus gestures. To improve user experience, some additional features, such as photo capturing and video or voice recording can be added to the HMI, as an example, for error reporting.

The second type of advanced HMI that is discussed in this context, is voice user interface, where the input is made by voice commands. Voice recordings can be added to error reports for later playback to other users or provide real-time interaction between the machine and its user. An example is voice dialling on mobile phones. Similarly, generated voice output is meant to be interpreted and acted upon by the user, such as in the case of voice commands of car navigation devices.

Display Size

The main difference between a smartphone and a tablet is the display size. While smartphones are smaller and "pocket-sized", the tablet is able to leverage on a larger diaplay to show more information. From an analysis on smartphone commercialized by market leaders, it was found that display diagonal size ranges between 3.5″ and 6″, which means HMI should be adjusted to hardware restrictions. In this kind of scenario, the user interface should provide only basic information, and be menu based to enable easy access to application features.

Tablets display diagonal sizes ranges between 7″ and 12″. Larger display provides easier access and interaction. The larger display is also crucial when more urgent information needs to be available, so there are little actions needed to navigate to them.

Controls

The controls are one of the most notable aspects when user is selecting a device for specific MANTIS task. Mobile HMI is usually controlled by means of a touch screen display with hand or stylus. Graphical user interface is combined with menu user interface, which means there are visual buttons on the display to lead interaction with certain actions. Most of the mobile devices have some physical buttons (depends on the manufacturer), where basic functions of devices can be applied, for instance on/off button, camera shortcut etc. Finally, smartphones and certain tablets can provide simple haptic feedback to the user through vibration.

Other possibilities of controlling tablets or smartphones are with external controls, such as an external keyboard or access through an external device, for instance, PC-smartphone connection, Bluetooth connection, and remote control.

User Role

The mobile HMI is typically used by users who only need to see a limited amount of information and alerts.

Examples of roles in MANTIS use case scenarios interacting with mobile HMI are:

- An operator, who is a person usually dealing is usually dealing directly with the technical process. Due to the type of work, operators may have wet, dirty hands or wear gloves, which means they are incapable of using a keyboard or mouse. Devices are therefore touch-screen based and should work with gloves as well as hands. Some scenarios require them to also be mobile, while in other scenarios the cyber-physical system being worked on contains fixed operator consoles. In case of production line noise, voice user interfaces, excluding alarms, can only be used with caution or not at all;
- Other user roles, such as maintenance team member, 3rd level development support, service technician, who are working both on the

terrain and in the office but only need to see certain information, alerts and notifications, are using mobile HMI as well.

Additional Sensors and Context Sensitivity

Mobile devices can also include other sensors, such as a GPS receiver, air temperature and pressure sensor, accelerometer, compass, fingerprint reader, etc. Some of those may be useless in industrial scenarios. For example, in production plants where ambient temperature is important, highly accurate temperature sensors will be permanently installed, rather than relying on measurements from a worker's tablet. The compass may not work well in vicinity of large metal structures. Furthermore, the MANTIS use case scenarios do not explicitly mention human interaction with data from such sensors; therefore these aspects are not further considered in this chapter.

On the other hand, the MANTIS deliverable on context-awareness explored when, where and how the human-machine interaction can be supplemented by implicitly taking into account the context of the task being performed. For example, if the mobile user's location can be determined accurately enough, the HMI could automatically switch to the screen relevant to the machine closest to her.

The context is not limited to the sensors on the mobile device, but rather includes any data related to the task but not explicitly being handled (typed in, read) by the user. Some of the goals of the MANTIS project are measuring, recording, and statistically processing huge amounts of data, all of which provide additional context for HMI tasks. Moreover, the latest global trend of making industrial machines connected and intelligent shows that context-sensitive HMI is an important topic and not just limited to MANTIS.

6.1.5.2 PC HMI

PC platforms are more commonly used professionally, for instance in industry, production and design. In industry, it is usually used as a set of desktops in the control room, or as industrial PC located on the production site (extreme environment). The majority of average users are able to manage PC devices.

PCs, used in MANTIS scenarios, can be divided into three groups, desktop PC, laptop and industrial PC. All of them have some advantages, which can be used in different environments, with different user types. They are typically connected into local wired (e.g., Ethernet) networks, which are secured more easily than the wireless networks used by mobile devices.

PCs are highly customizable. Their operating systems are different than those on mobile devices (different versions of Windows, Linux, MacOS, etc.) and they usually, but not always, use a graphical user interface. Some older or specialized software still uses the command-line interface (CLI). Desktop PCs are dependent on peripherals (external components), such as a display, keyboard, mouse, to enable access to the HMI. Through the use of external components, PCs are much more customizable than smartphones and tablets. Except for laptops, PCs are not mobile, which means they are engaged in the certain environment (office, production site, etc.).

Interactions on PCs are usually restricted to keyboard and mouse, although hybrid laptops with touchscreens are also popular in some circles. External components, such as microphone and cameras, can be used where needed for additional interaction modes. However, there is less need for voice-based interactions (voice commands and voice generation) than with mobile devices, because with PCs there is rarely a need for hands-free interaction and because the keyboard is much more usable for entering text then the on-screen emulated keyboards of mobile devices.

Display Size

The currently common display diagonal size of external monitors can range from 19″ to 30″ and 13″–17″ for laptop screens. Wide-screen ratio 16:9 and 16:10 are common formats on the market, though more information can be shown on traditional 4:3 screens. Some external monitors can be switched from landscape to portrait orientation. In any case, MANTIS HMI design should be adjustable to one format or another, to avoid vital information not being available. Display size, compared to mobile HMIs, is larger and therefore can display greater amount of information on the screen. Users like data analysts, in need of having to look at plenty of data at once, need large displays; mobile display sizes do not meet the requirements.

Controls

Desktop PC can only be controlled by external input and output modalities.[1] Common modalities are based on vision (screen), sound (audio outputs) and sense of touch (vibrations, movements).

Considering users' needs, different components are used for input. The computer keyboard allows the user to enter typed text and the mouse allows the user to input spatial data to a computer. On a desktop computer,

[1] A modality is a path of communication employed by the user to carry input and output.

a virtual keyboard might provide an alternative input mechanism for users with disabilities who cannot use a conventional keyboard, or for bi- or multi-lingual users who switch frequently between different character sets or alphabets. There are many variations of pointing devices, such as 3D mice, joysticks, etc. Some other devices, such as a digital pen, digitizing tablet, high-degree of freedom input and composite devices are available on the market. Most common input devices for imaging, which are used to digitalize video or image to the computer are for instance digital camera, webcam, fingerprint scanner etc. Audio input allows the user to capture sound.

The output is another aspect of HMI. Displays visually represent text, graphic, and other video material. The visual material can be also printed on paper by other output devices, such as printers, 3D printers, etc. Audio can be heard through speakers or headphones. More uncommon is the haptic technology, which provides tactile feedback using the sense of touch, vibrations, the motion of the user.

User Role

PC HMI is usually used by users based in one place, such as an office, control room or production site. PC users need to manage a larger amount of data and therefore need a more powerful device. There are usually no restrictions for keyboard or mouse use.

Examples of users from MANTIS use case scenarios interacting with PC HMI:

- Maintenance managers are responsible for long-term analysis of tool usage. They communicate with the business manager. For a better overview, a maintenance manager uses PC HMI, where all available data can be shown. Important notifications can be also received via an e-mail;
- The data analyst is responsible for analysing the results from prediction/learning algorithms. His work is connected to a large amount of information, which can only be represented on PC display. He or she can access all live streaming data and historical data from several data sources. Best HMI for this user role is the graphical user interface as there is many data to be displayed;
- Production manager, qualifier, maintenance planner, maintenance planner unit, plant operator need to use a PC HMI due to a larger amount of data, better access to other applications and permanent workspace.

6.1.6 Recommendations for Platform Selection

The main factors defining which type of HMI to be used in certain situation are:

- users (needs, preferences, capabilities);
- user tasks, interactions and goals;
- platform (hardware, software constraints);
- environment (noise, lighting, dirt, vibrations, etc.).

MANTIS scenarios present common situations in terms of maintenance of the system and trouble-shooting:

- monitoring the system (e.g., monitor the information on screen, reading reports);
- simple interactions;
- analysis;
- usage of the machine and trouble-shooting;
- communication.

The selection of HMI to be used for certain situation can be summarized as follows:

- For monitoring the machines and generated data, both mobile and PC HMIs are required. The preferred selection of mobile device is tablet as it can present more data on screen due to larger display size. The users, performing monitoring tasks, are diverse – from machine operator, to maintenance engineer and data analyst. The selection of the HMI, based on the user is therefore more dependent on the location of the user rather than their role. The users with fixed workplace location will prefer the PC HMI with larger display size, while users working on the terrain will prefer mobile HMIs due to ease of accessibility. Mobile HMI can be used as a remote extension to PC HMI as well;
- For simple interactions with HMI (confirmations, calculations, ratings etc.), buttons and menus are mainly used as interface elements. As trivial interface elements to use and implement, there is no limitation for usage to either HMI. The main difference is the control function; normally touch for mobile HMI and keyboard with mouse for PC HMI;
- For analysis of system data, generated reports or financial aspects, responsible by data analyst, PC HMI is preferred in all MANTIS scenarios. For component analysis and usage of various analysis tools by some maintenance managers, tablet is required as a HMI;

- For usage of the machine, troubleshooting and following guidelines and instructions when repairing the machine, both mobile and PC HMIs are required. The preferred selection of mobile device is tablet, while the PC HMI is mainly presented as industrial PC. The users of PC HMI can be defined as machine operators and the users of tablet are defined as service technicians, support team or maintenance engineers, without fixed workplace, moving from site to site, or working remotely;
- For communication between users (e.g., text, audio or video chat), both mobile and PC HMIs are required. Communication is not limited to any user role and device, it could be used on various devices (smartphone, tablet, desktop PC and laptop PC) by various users. It can be assumed that the selection of HMI by specific user is affected by other factors or tasks rather than chat requirements.

All the user roles enumerated for mobile and PC HMIs can be supported well with a web-based HMI. To run well on the type of device each user role typically uses and to ensure good user experience, the web-based HMI has to be implemented carefully so that it is able to adapt to different devices and display sizes.

A web application intended for both mobile and PC users must be able to adapt to screen sizes from 4″ to 30″ in both portrait and landscape orientations. However, individual screens of the web HMI only have to adapt to the typical screen sizes of the devices that will be used for tasks that the screen is part of. The screens intended for mobile devices will still be usable on PCs. The opposite, using a mobile device for a task intended for PC, will be possible as a workaround (e.g., the PC broke down) but not recommended for extended periods.

The same can be said for the difference in controls. The tasks intended for mobile devices should require at most a minimal amount of typing, thus the inputs are limited to multiple-choice buttons and alternative modalities (sound, camera). The workflow should lead the user through well-defined procedures using linear navigation. No high-precision pointing should be required unless a stylus is used.

The tasks intended for PCs, however, can offer richer navigation, can use typing as the default input method and should offer keyboard shortcuts for buttons, menus, and navigation. Obviously, multi-touch gestures must not be required.

LED-based projection technology is currently getting build into smartphones and tablets, giving them the potential of projecting a big (touch) screen on any surface. Although a very promising solution to the current screen size limitations, this technology is still in its infancy and there are many obstacles to overcome before it will become mainstream.

6.1.6.1 Web-based HMI

Web applications are increasingly being used in all domains where the application is not an isolated island, but has to also interact with other users and/or data residing elsewhere. A well-known example are office suites (Google Drive, MS Office Online), where web applications enable collaboration in the sense of multiple users simultaneously editing the same document. On the other hand, they are less feature-rich than their desktop counterparts, the interaction is less responsive in certain cases and might require a continuous web connection to work.

In case of MANTIS maintenance HMI, most use cases require both PC and mobile HMI. The operating systems and consequently the native software development stacks (programming languages, integrated development environments, libraries and frameworks for user interface creation, debuggers, etc.) are completely different in these two cases, requiring significant duplicated effort. Windows platforms do strive to unify mobile and desktop development to a degree. However, only supporting Windows as a mobile platform is currently limited and has low availability of hardware. Web applications, on the other hand, run out of the box on all PC and mobile operating systems, making it much easier to support both types of devices.

Another advantage of web applications is that they can be deployed and managed centrally and are easier to ensure that all users use the same (latest) version [Miller, 2008]. They need continuous access through a network with a (local) server to work. Nowadays, all devices are already suitable for connections, either through Wi-Fi or mobile network. Since most of MANTIS scenarios involve cooperation of multiple users. This is a requirement regardless of whether web or native applications are used.

The upcoming evolution in web applications is the ability to run and store data when no connection to a server is present. In this case, data is managed on the local browser and once the connection restores, all data will be synchronized. This is already the case in some progressive applications and receives increasing supported from browser vendors.

6.1.6.2 Responsive design

The user interface of the MANTIS platform has to be flexible and modular, to easily adapt layout, content and appearance to different screen sizes. In this context responsive design approach proposes an efficient and suitable method to solve the challenges of modularization, flexibility and scalability and to optimize user experience across devices of varying sizes and capabilities.

Responsive design allows a page to adapt layout and content to viewing contexts across a spectrum of digital devices. Responsive design approach supports the adjustment of device's resolution, size, and layout, from smartphones to desktop PCs. Devices such as tablets and smartphones also support orientation changing, providing two possible screen widths. Developers now have new web standards like Hypertext Markup Language version 5 (HTML5), and Cascading Style Sheets version 3 (CSS3), enabling them to design and build user-sensitive sites that respond to a range of contexts and device capabilities [Gardner, 2011].

"Mobile first" is a more recent design paradigm whereby web-based user interfaces are optimized for mobile use first and have a graceful fall back when being used on devices with larger screens instead of the other way around. This is done by gradually offering more information and options once the real estate allows it. Starting from the essential must-haves on small screens to additional nice-to-haves on larger screens.

Marcotte [Marcotte, 2010] outlined a method for creating fluid layouts that are screen-resolution agnostic and capable of dynamically changing according to user context. He describes responsive design as having three elements:

- a fluid layout that uses a flexible grid, which in turn ensures that a website can scale to a browser's full width;
- images that work in a flexible context, whether fluid themselves or perhaps controlled through overflow mechanisms;
- media queries, which optimize the design for different viewing contexts and spot-fix bugs that occur at different resolution ranges.

6.1.7 Interface Design Recommendations for MANTIS Platform

Although MANTIS use cases come from different domains (production assets, vehicles, energy production and health equipment), common recommendations can be provided that apply to most or all of them. The objective was to provide a set of guidelines that, if followed by the various MANTIS HMI designs, will ensure a consistent interaction between the users and the system with the common pitfalls avoided. The user will be provided with simple, intuitive, legible displays that are suitable for the intended purpose. These recommendations can be applied to any potential interface, emphasizing maintenance tasks, alarm management, pro-activeness and collaboration.

Based on the understanding of the use case scenarios and a design philosophy, a set of recommendations for designing interfaces for MANTIS platform was established and refined:

- The **structure** of the interface has to be defined (a four-level screen hierarchy and organization of screen flows) and the implemented logic of navigation controls and menus that allow the user to navigate more easily through the HMI;
- Overall **screen layout** has to be defined including a consistent arrangement of interface elements and distribution of information on the screen to enable a fast orientation of the user;
- The **visual design** section has to describe the general appearance of the interface and specify the properties of the interface elements, such as colour selection and display text;
- Decision has to be made about the most effective way of **data presentation** using components such as tables, charts, bar charts and line graphs;
- **Interaction** and **collaboration** betweenusers have to be defined, for example managing alarms and events, creating reports, following step-by-step guides and communicating via messaging and chat.

6.2 Adaptive Interfaces

A great challenge in human-machine interaction is to ensure that the information presented on the interface is meaningful and relevant, properly represented, location-aware and targeting the appropriate person. It is also important to bring the attention of the user at the most suitable moment, instead of disrupting him/her with information overload.

In a proactive and collaborative maintenance platform, gathering and combining condition monitoring data with contextual information can provide numerous benefits such as a better productivity, improved decision-making, accuracy of predictions and process optimisation, together with an enhanced usability and personalisation on the HMI.

6.2.1 Context-awareness Approach

Context-aware computing is a paradigm where applications and services use environmental information acquired by sensors (such as user and device location, state, time, nearby places, people and devices, etc.) to provide relevant information and/or services to the user.

The concept of context has been researched over the last decades in software engineering, especially in areas such as Natural Language Processing, and more generally in Human-Computer Interaction.

Context makes the interaction with computers easier by adapting the information to the user, discriminating what is relevant and what is not, so that the user in human-machine interactions can focus on high-level tasks, which is very important in scenarios of information overload, especially as we move towards a world of ubiquitous and pervasive computing and the Internet of Things.

6.2.1.1 Context and context awareness fundamentals

The term "context" has been addressed by some authors such as Abowd and Dey [Abowd, et al., 1999], which identified the two main types of context: primary and secondary context. These contexts differ through four main categories which are location, identity, activity and time.

Following this study, [Perera, et al., 2014] referred to the main difference between each context as:

- **Primary context:** Any information retrieved without using existing context and without performing any kind of sensor data fusion operations;
- **Secondary context:** Any information that can be computed using primary context by using sensor data fusion operations or data retrieval operations.

Figure 6.3 shows some examples of context categorization from a conceptual and operational perspective:

The term "context awareness" refers to the ability of computing systems to acquire and reason about the context information and subsequently adapt the corresponding applications accordingly.

This term was first introduced in the research field of pervasive and ubiquitous computing by Schilit and Theimer in 1994 in a paper entitled "Context Aware Computing Applications" [Schilit, et al., 1994], describing software which adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time.

Abowd and Dey [Abowd, et al., 1999], provided the most widely accepted categorizations of context aware features which are the presentation of information and services, automatic execution of services an tagging of context.

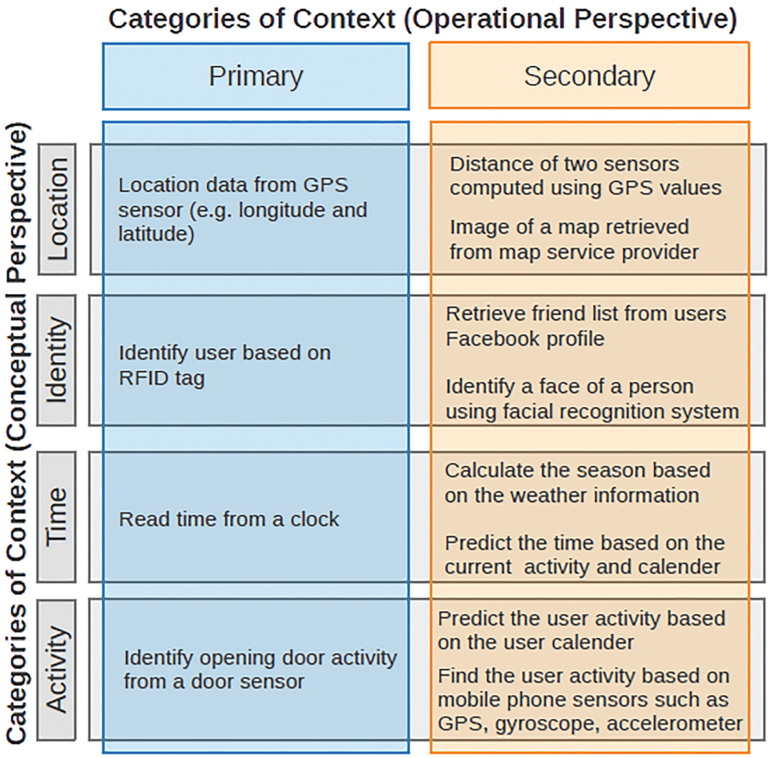


Figure 6.3 Categories of Context [Perera, et al., 2014].

Another interesting study is the one done by Barkhuus and Dey [Barkhuus, et al., 2003] where they identified three levels of interactivity in context awareness based on the user interaction which are personalization (based on the preferences and expectations of the user), passive context awareness (updated contextual information is presented to the user) and active context awareness (application autonomously changes its behaviour according to the sensed information).

6.2.1.2 Context lifecycle in context-aware applications

Perera et al. [2014] selected ten popular data lifecycles [Hynes, et al., 2009; Chantzara, et al., 2005; Ferscha, et al., 2001; Wrona, et al., 2006] to analyse them in their survey. After reviewing these works, the authors stated that applications use typically four phases when processing context, from the moment it is acquired from sensors in raw format, to the moment it is consumed by the end-user application:

- **Context Acquisition:** Contextual data is captured from the environment using sensors;
- **Context Modelling:** The collected data needs to be represented in a meaningful manner through a context model;
- **Context Reasoning:** Modelled data needs to be processed to derive high-level context information from low-level raw sensor data;
- **Context Dissemination:** High-level and low-level context need to be distributed to the consumers who are interested in context.

6.2.1.3 Adaptive and intelligent HMIs

Information overload, variety of heterogeneous users and cognitive overload for decision making are different problems to deal with the process of HMI design and development. Different research areas such as Intelligent User Interfaces (IUI) and Adaptive User Interfaces (AUI) face with those problems applying intelligence during the process, investigating new algorithms and promising techniques for user, context and content adaptation.

Intelligent User Interfaces (IUI)

Intelligent User Interfaces (IUI)[2] is a multidisciplinary area inside the Human-Computer Interaction (HCI) research field that aims to improve human-computer interaction by applying technology to those interfaces [Ehlert, 2003].

Over the years, many researchers from different fields (as shown in Figure 6.4) have influenced and made improvements on areas related to IUI, for example in psychology (advances in cognitive sciences or human perception), in artificial intelligence (improvements in user modelling, or in machine learning to predict user behaviours), and in the HCI field (new visualization and interface evaluation techniques to have a better user experience and usability).

Intelligent interfaces can **adapt to user, context and situation,** they have the **ability to communicate** and they have the **ability to solve different problems**, improving the usability, flexibility [Maybury, et al., 1998] and user experience adding Artificial Intelligence.

Adaptive User Interfaces (AUI)

Adaptive User Interfaces is a subtype of IUIs that improve the interaction with the user with knowledge taken from this user [Langley, 1997]. We can

[2]IUIs can be described as interfaces that, using intelligent technology, can improve the communication between the machine and the end user.

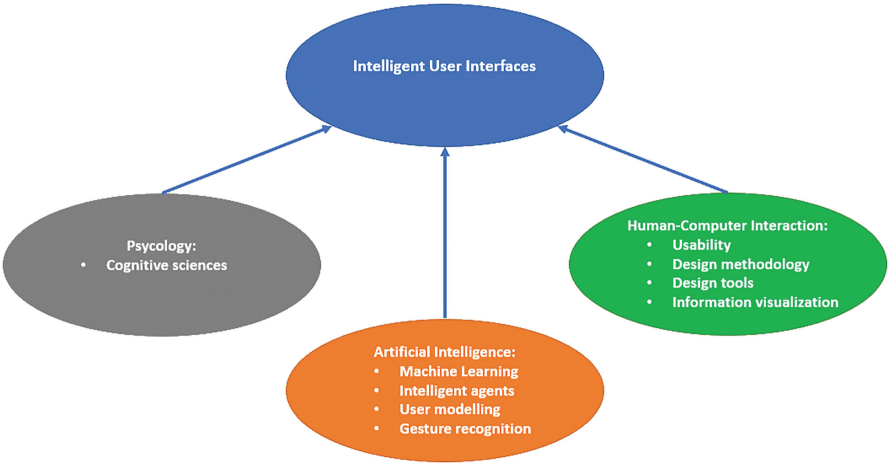


Figure 6.4 Research Fields Influencing the Development of Intelligent User Interfaces.

divide them in two main groups according to the type of feedback that the user must provide:

- **Informative Interfaces.** For example, recommender systems or information filtering, where the interfaces select or filter information for the end user to improve the user experience;
- **Generative Interfaces.** For example, systems for planning or document preparation, where the generation of a useful structure is needed.

AUI was recognized as very promising and challenging area [Norcio, et al., 1989]. The purpose of this field is to provide a user experience that is automatically created via machine-learned processes. We can define AUI as the intersection between HMI and Machine Learning.

A user model is the description and knowledge of the user maintained by the system. User modelling is concentrated on individual users' knowledge, goals, plans, emotions, personality, ability etc. [Kobsa, 2011]. Early researches of adaptive interface models are based on user data.

Context-aware user interfaces play an important role in many adaptive human-computer interaction tasks of location-based services. Examples of works in this area is the context-aware adaptive models for mobile location proposed by [Feng et al., 2015], the personalized traveler information system (ATIS) presented by LBS, [Lathia, et al., 2012] or the AUI based on various possible contexts such as handicap user profile proposed by [Zouhaier, et al., 2013].

6.2.1.4 Context awareness for fault prediction and maintenance optimisation

The use of context information within prediction and maintenance-related processes is one application of context awareness mechanisms. Here, the focus is to enhance the reasoning/modelling systems (e.g., diagnostics, prognostics, maintenance scheduling) to improve decision support.

Concerning predictive analytics, the predictive ability of a system is enhanced by contextual information present in the environment [Kiseleva, 2013]. Besides that, context can also be used to improve the Remaining Useful Life (RUL) prediction, as expressed in [Ahmadzadeh, et al., 2012; Thaduri, et al., 2014]. Different approaches can be used to obtain an accurate RUL using operational context, there could be better ways to make this prediction.

An example of an approach would be to predict the degradation of an equipment (e.g., machine tool) based on how it is being used using "Fingerprint" is the recorded data obtained periodically when monitoring a sensorised machine doing the same set of predefined operations (see Figure 6.5).

Another approach is to calculate a more accurate RUL, where context information concerning the future operational conditions have an impact in the final prognosis scenario. This approach was applied in a study done in [Ferreiro, et al., 2012] to predict the RUL of aircraft brake wear.

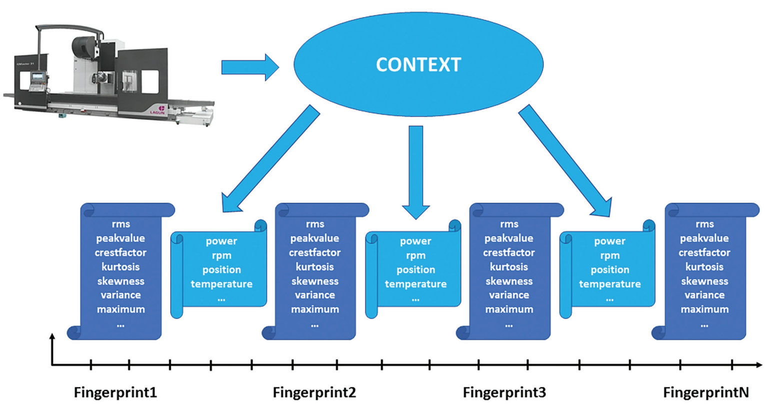


Figure 6.5 Context-Dependent Data Integration in Maintenance Scenarios (POWER-OM, Lulea Technology Univ., 2015).

6.2.1.5 Context awareness for maintenance personalisation and decision-making

Personalization has become an important aspect in many areas, e.g., personalization of car interfaces [Endres, et al., 2010; Garzon, et al., 2011], in smart home environments [Ma, et al., 2005], automatic profile selection [Coutand, et al., 2006], healthcare [Zhang, et al., 2005; Hashiguchi, et al., 2003; Koutkias, et al., 2001; Koutkias, in drugi, 2001].

Finding proper patterns and accurately predicting the results could provide better personalization and adaptation. Here, context has an important role, because the behaviour of persons/users may be related on the context (i.e., location, time, access device).

Context awareness allows the development of personalized services that automatically adapt to the user's situation, and in this sense, context management mechanisms can complement prediction models so that predictive analytics decisions can be more accurate.

One of the main challenges in this area is to construct the mechanisms which would detect what context is, how to integrate context into the prediction models, or monitoring the stream of contextual data over time to detect anomalies.

Results could be displayed in various human-machine interface applications. One of the main characteristics of modern intelligent user interfaces is the integration of multiple users with diverse needs and requirements. Benefits of such personalization include improved safety, added comfort, increased efficiency, or enabling access for users with special needs.

Personalization in maintenance could be reached by using different artificial intelligence concepts to predict next steps and help maintenance workers in decision-making tasks. These tasks could consist of fault detection and diagnosis, detecting anomalies, scheduling suggestions, choosing proper maintenance concepts, optimizing energy consumption during the operations, informing operators about actions, or planning a repair action, among other things, which could be displayed through different HMIs. HMIs in maintenance are customized to the worker's need and depends on the type and environment of industry.

To make accurate predictions in maintenance, qualitative data (context information) is needed which is usually historical data of maintenance activities; large amounts of sensor measurements, history of user interactions, anomalies, faults, etc.

Wearable devices are important in the plant maintenance, because they allow the user's hands to remain free to do the work and currently rely on voice recognition and voice response [Nagamatsu, et al., 2003; Nicolai, 2005; Nicolai, et al., 2006; Stiefmeier, et al., 2008].

As stated by Lee in the article "Cyber physical systems: Design challenges" [Lee, 2008] cyber-physical systems (CPS) are integrations of computation and physical processes, where physical processes affect the computation and vice versa. The addition of context information to the monitoring and prediction in maintenance activities can contribute to improve maintenance approaches, enhancing the cost, time and quality of the processes.

The large amount of data collected using sensors can be used for detecting and analysing anomalies and faults in large and complex systems. Data-driven approaches leverage on this large amount of data which is collected by CPS and is used to learn the necessary models automatically; recognize unusual situations, optimize energy consumption during the operations and inform operators who use this information to modify system processes, or plan for repair or maintenance. System's engineers and experts can use this information to take further actions (e.g., update operations procedures or redesign the system).

The data-driven prognostic approach [Swanson, et al., 2000; Niggemann, et al., 2015; Krueger, et al., 2014; Jämsä-Jounela, et al., 2013; Zhang, et al., 2015] could be used to determine the fault and predict the amount of time before it reaches a predetermined threshold level.

6.2.1.6 Context awareness approaches in a proactive collaborative maintenance platform

Here are three generic context awareness approaches that could be incorporated in a collaborative maintenance platform to provide some kind of benefit to its users:

- **Adaptation to scenarios**
  From alerting and warning situations, to special events in the state of the production process, or changes in the location of a user, these are examples of scenarios where the use of context awareness could be relevant to deliver the right information, at the right moment, in the right format, to the right person by means of an adaptive and intelligent user interface;

Table 6.1 Context Awareness Approaches in a Proactive Collaborative Maintenance Platform

Context Awareness Approach	Benefits
Adaptation to scenarios	• Personalisation • Usability • Maintenance optimisation • Better ergonomics
Enhanced reasoning algorithms	• Improved decision-making • Better maintenance planning • Accuracy of predictions • Cost savings • Maintenance optimisation
Personalized maintenance suggestions	• Productivity • User experience • Enhanced asset management

- **Enhanced reasoning algorithms**
  The goal here is to use operational information as a contextual extra input to reasoning algorithms (diagnostics, prognostics, scheduling, etc.) to optimize their results;
- **Personalised maintenance suggestions**
  Context could be used to make personalized maintenance suggestions to users when performing everyday processes in the system to improve their productivity and overall experience on the HMI.

Table 6.1 summarizes the main benefits of the approaches described above.

6.2.2 Interaction Based/Driven Approach

Nowadays interaction analysis (e.g., clickstream analysis) is one of the most frequently used techniques to understand user behaviour while he is using the interface. This understanding can help not only in the interface design and development process but also providing some inputs to carry on intelligent adaptation. This interface adaptation, commonly called adaptive user interface, can enhance usability and user experience.

User interaction data have detailed information of how users perform actions with the different elements of the interface such as visualization elements, buttons or icons, and also how the user navigate between them. Interaction history gathered in these datasets can be described as a collection of different timestamped actions performed by a single user whilst is using the interface [Nguyen, et al., 2017].

Analysing these datasets can give us information about user sequences or recurrent patterns [Soh, et al., 2017] Furthermore, this analysis can be used as relevant information to perform different automatic adaptation in the interface [Dev, et al., 2017].

Therefore, it is necessary to have different tools which allow to track and store all the interaction between user and the interface. In the context of MANTIS project this interaction driven approach will be focused on defining a methodology to track and store navigation and actions performed by the user with the interface.

6.2.2.1 Introduction

When the user accesses a web interface and interacts with it, a digital *fingerprint* is recorded. This digital fingerprint can be defined as the record of actions and steps performed by the user in a time slot. These actions have been captured and stored automatically with a timestamp key that allow us user tracking.

The issue has been approached from two different perspectives at it is shown in Figure 6.6:

- Navigation between the different interfaces. Extracted and parsed from the web server logs. Analysing this data can give us information about the common paths and navigation flows;
- The interaction carried out in the different interfaces. Extracted automatically from the interface via JavaScript. Analysing this data can give us information about the common actions or sequences.

For the user interaction capture and storage system Elastic[3] technological framework has been proposed. Elastic is a tool which is often used in Log Analytics use cases.

Elastic is an open source platform based on Lucene[4] that allows save, search and display information stored in different indexes. Elastic provides different tools such as Logstash[5] for data parsing, ElasticSearch[6] for data storage and Kibana[7] for data visualization. The main point to be considered is that the information should be indexed by time (*Timestamped data*) and

[3]http://elastic.co

[4]https://lucene.apache.org/core/

[5]http://elastic.co/products/logstash

[6]http://elastic.co/products/elasticsearch

[7]http://elastic.co/products/kibana

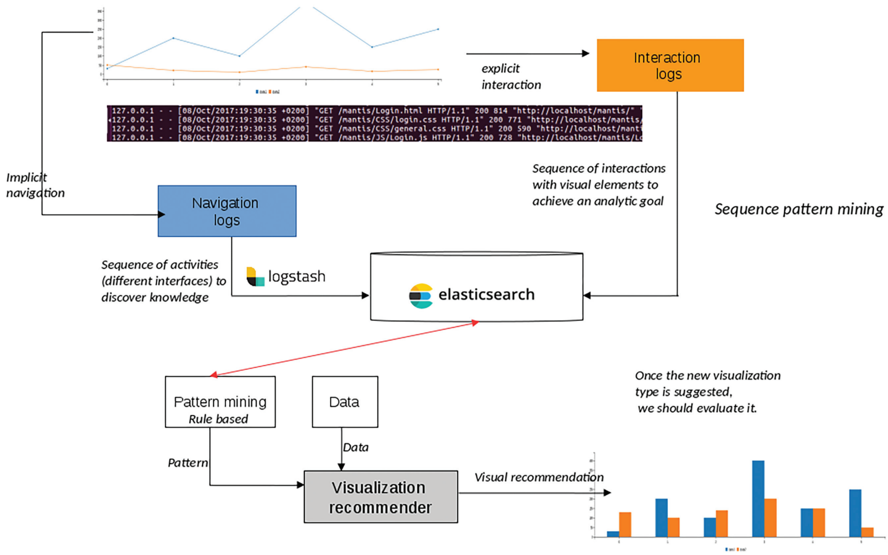


Figure 6.6 Interaction driven approach for visualization recommender.

ElasticSearch is a good option for that purpose. ElasticSearch is a powerful tool with many advantages such as:

- speed: ElasticSearch uses distributed inverted indexes;
- APIs: ElasticSearch offers a REST API and uses JSON schema;
- variety of plugins: Logstash for data parsing and processing or Kibana for developing different dashboards and interactive visualizations;
- real time index updates: Very important when you are monitoring a scenario;
- available for different languages: Python, Java, Ruby or Node.js.

6.2.2.2 Navigation tracking and storage

When a user is navigating among different interfaces, this navigation information can be extracted from the access log files. A log file is a collection of events and actions that is stored in the web server. In the context of the MANTIS project, common log file format will be considered. This format, NCSA Common format[8], is an standardized text file that is generated by different web servers.

[8]https://httpd.apache.org/docs/trunk/logs.html#common

Access log files can have the following structure:

192.168.2.22 - -[12/Feb/2018:15:20:02 +0200]"GET /mantis? P=1 HTTP/1.1 " 200 136 " http://mantis.com/true "" Mozilla/5.0 (Windows NT 6.1; rv: 24.0) Gecko/20100101 Firefox/24.0 ".

Parsing this file can give us the following information:

- user IP address;
- date time. Timestamp value that will be used for index;
- method. (GET or POST);
- HTTP protocol version;
- response status code: 10x Informative response, 20x successful response, 30x Redirection, 40x Client error or 50x Server errors;
- system information: Operative system and browser.

Accessing the server log files and parsing them with Logstash, raw information can be converted into a format that allows traceability. This information will be stored in an ElasticSearch index.

Indexing by 'datetime' let us to analyse the navigation flow and detect the most common paths while using the interfaces. This information can be used for instance to improve the process by reducing the number of clicks performing some action.

As it is shown in Figure 6.7, this navigation capture and storage can be done in an automatic and non-intrusive way. Other frameworks or solutions have been found in the literature [Hashemi, et al., 2016; Atterer, et al., 2006] but they are not as flexible as this approach. Each navigation will add a new register in the access log file and in turn will be recorded in the ElasticSearch index.

6.2.2.3 Action logs

Other important component for the interaction analysis is the register of different actions performed by the user with the interface. Those actions will be executed through different devices such us mouse, keyboards or tactile interfaces. One of the main advantages of using web based interfaces is that JavaScript can be used for adding functionalities to the interface.

JavaScript[9] is an event-driven, dynamic and multi-paradigm programming language that allows to add functionality not only to the client side but also to the server side. JavaScript is proposed as the programming language to track and send the interaction to the ElasticSearch index.

[9]https://developer.mozilla.org/en-US/docs/Web/JavaScript

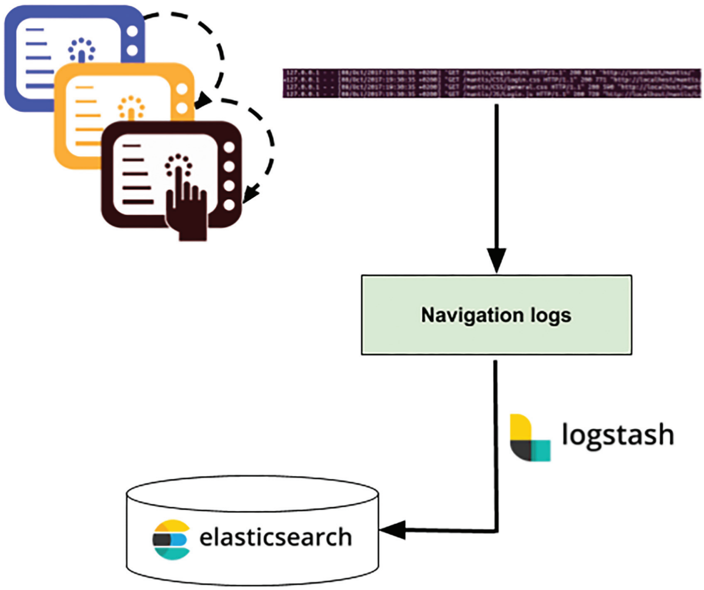


Figure 6.7 Navigation parse and index on ElasticSearch.

To track and store all the interactions with the interface performed by the user, different triggers must be defined and implemented. These triggers will record insert new record into the ElasticSearch index. The triggers will be associated to different interaction events.

In the context of MANTIS project only will be considered six kind of interaction events: mouse events, keyboard events, focus, drag and drop events, clipboard events and view events. Once the user performs an action on the different interface elements, the trigger will be fired and automatically upload new record to ElasticSearch index via JavaScript. Figure 6.8 shows the process of event triggering and storing into ElasticSearch index.

Apart from the system 'datetime' date and time other important information must be stored, for instance: which interface the user is using, with which element the user has interacted or what kind of action was triggered by the user. This way, we can trace interactions performed on the different interfaces and in turn we can perform an analysis for instance to detect patterns. Note that this should only be done on a temporarily basis for study purposes and with clear and written consent of the monitored user.

As we can see in Figure 6.9, for each of the interactions carried out in the interface the following information is stored in the ElasticSearch index:

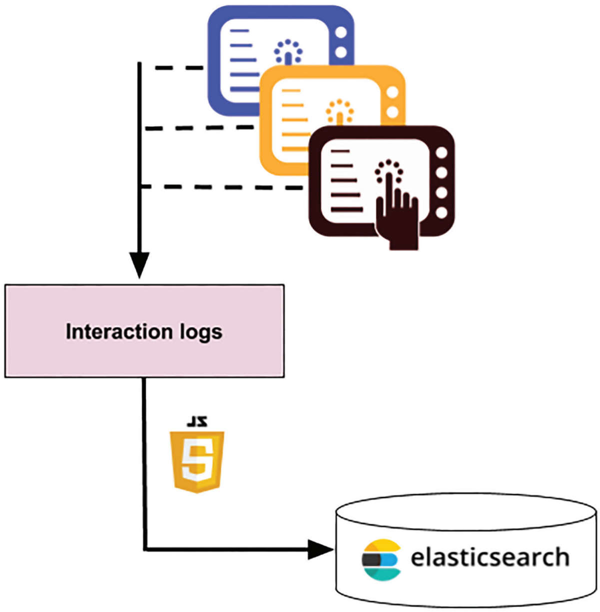


Figure 6.8 Interaction capture and store in ElasticSearch index.

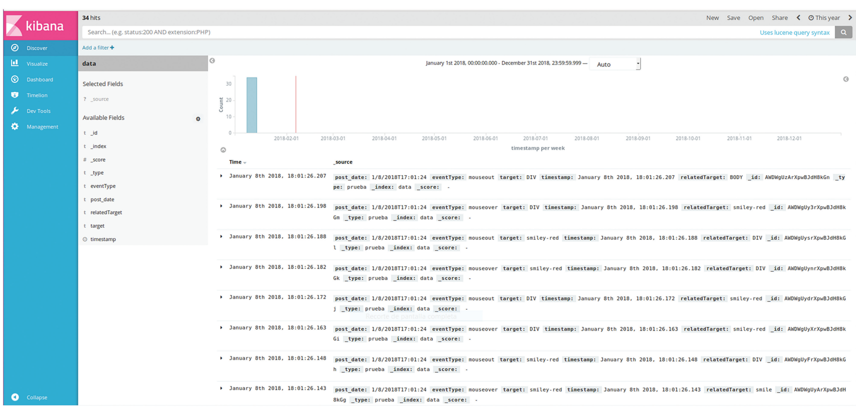


Figure 6.9 Index structure created in ElasticSearch to capture the interaction of the user with the interface.

- type of event that the user has made: Click, mouse movement... These events are predefined by JavaScript;
- timestamp of when the action was taken. With this information the traceability can be done;

- interaction element;
- which interface;
- ElasticSearch index name.

This methodology of capturing navigation and interaction allows storing the interaction with the interface in a non-intrusive and transparent way. On one hand, we have information about navigation flows indexed by time and on the other hand, we have the information about different performed actions between user and interface.

This dataset can be used as a powerful set of information to perform different analysis techniques in order to detect patters, provide recommendation or develop automatic adaptation engines.

6.3 Advanced Data Visualizations for HMIs

The visualization requirements of a data analyst or a highly skilled maintenance expert go beyond the basic graphs and other widgets typically present on industrial dashboards. Most importantly, such experts need to be able to choose what data to visualize and how in order to gain further insight.

6.3.1 Visualization of Raw Data

The specific requirements in the different maintenance use cases, in MANTIS and beyond, are so diverse that no single visualization tool can cover them all. On one hand, raw data can be visualized by various methods in order for a human expert to explore relations between data. On the other, visualizing the results of analysis and decision-making algorithms, such as those presented in Chapter 5, can provide insights into their operation and help improve the algorithms or validate their results. Thus, an overview of some data visualization tools is given here. Most tools require specific input formats and thus need to import the data rather than use it directly from the data store already set up for maintenance data. One of them, Kibana, was chosen as applicable to the widest array of maintenance use cases and therefore the process of its integration is also described.

6.3.1.1 Visualisation tools overview

There is an abundance of utilities for producing the basic kinds of graphs, from office software to web graphing libraries to the built-in capabilities of scientific computing tools. The latter (e.g., Matlab, Octave, R) are most suitable for data analysts and already routinely used for such purposes.

Certain tools stand out due to innovativeness of the visualizations they offer. Plot.ly[10] is a tool with libraries for JavaScript, Python, and R. It can, for example, intuitively illustrate how various machine learning algorithms work on given datasets[11]. Inner workings of neural networks can also be visualized in interesting, interactive ways[12]. This can be particularly engaging if the problem being solved by the neural network is also visual in its nature, such as optical character recognition[13] or, for a maintenance-related example, optical recognition of worn-out machine components.

Finally, specialized machine learning tools, such as Orange[14], typically also include visualization capabilities. Orange enables data analysis, machine learning, and visualization via interactive workflows, as shown in Figure 6.10. The upper right part shows a workflow where an example data set is clustered using the k-Means method and then shown as a scatter plot (upper branch of workflow). The raw data is shown as mosaic (lower branch). The actual mosaic visualization is shown in the lower part. A powerful feature is the “Find informative ...” button included in most Orange visualizations. As

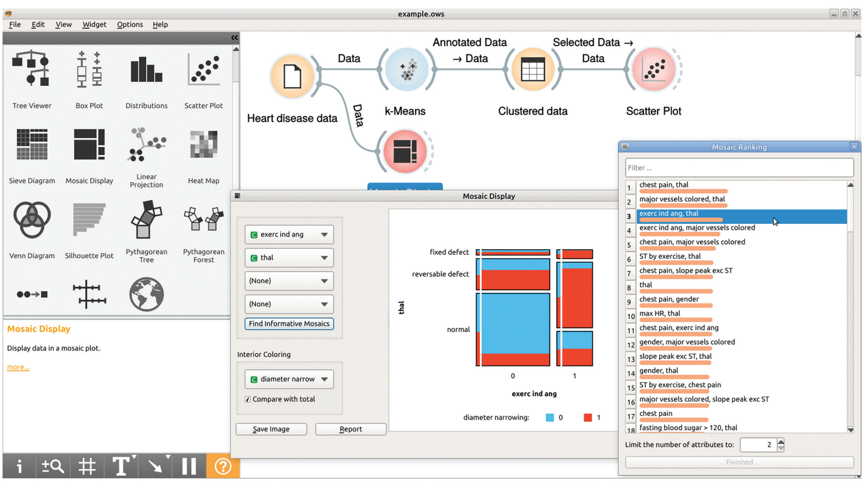


Figure 6.10 Data analysis and visualization workflow in Orange.

---

[10]https://plot.ly/

[11]See the on-line example at https://plot.ly/~jackp/16209/machine-learning-classifier-comparison.embed.

[12]A. W. Harley, ”An Interactive Node-Link Visualization of Convolutional Neural Networks,” in ISVC, pages 867-877, 2015.

[13]See interactive example at http://scs.ryerson.ca/~aharley/vis/conv/.

[14]https://orange.biolab.si/

shown in the lower right corner, this suggests which attributes of the data set to base the visualization on for most informative results. The left part of Figure 6.10 is the Orange visualization toolbox, listing all available types of visualizations.

Orange can be recommended as the most universally applicable tool with a shallow learning curve, while Matlab, R, and similar tools offer more flexibility and possibilities of automation of processes at the expense of having to learn the respective programming languages. However, neither of them scales well to truly large datasets of gigabytes or more.

6.3.1.2 Scenario 1: Kibana

Kibana[15] is a visualization plugin for ElasticSearch[16], and the latter is a distributed, highly scalable indexing and search engine. ElasticSearch can store a huge number of schema-free JSON documents. The fields of the documents are automatically indexed and can be searched for using a powerful query syntax. The documents of different types can either be stored separately (in different indices in ElasticSearch terminology) or in the same index, such that the search query determines whether all documents or just those with certain attributes present should be searched for. Architecturally, ElasticSearch is a service based on Apache Lucene indexing/search library, accessed via a REST/HTTP(S) API.

Kibana is implemented as an interactive web portal into ElasticSearch. It provides a user interface for performing queries on the stored data, but more interestingly, it can produce various visualizations. It is particularly powerful for viewing datasets that include a time dimension and allows interactive selection of time ranges, scales, filters, and aggregations. Together with Elastic search's support for large datasets, it is well suited to the maintenance use cases that continuously monitor processes and assets and thus, over longer time, invariable end up with large datasets.

The first step of integration of Kibana into a maintenance MANTIS prototype consists of development of a service that continuously requests new data from the data store, such as a MIMOSA database, and adds it to the ElasticSearch index. ElasticSearch also provides libraries for common programming languages, which are more straightforward to use than the HTTP-based REST API. Within the MANTIS project, a MIMOSA-to-ElasticSearch import service was developed in Python. The service simply

[15]https://www.elastic.co/products/kibana
[16]https://www.elastic.co/

adds the JSON objects returned by the MIMOSA REST API as ElasticSearch documents. Full contents of the (chosen subset of) MIMOSA tables are thus available for search and visualization. On the other hand, the data is raw and thus requires knowledge of MIMOSA data model (or, in general, the specific data model used). This can be seen in Figure 6.11, which shows a Kibana dashboard with three different graphs from a MANTIS use case[17] – alarm types and environmental measurements are labelled by raw MIMOSA field values, because that is how they are stored in ElasticSearch.

Kibana individual visualizations or whole dashboards can be embedded in any HTML document using the <iframe> tag. Figure 6.12 shows a single visualization inside the MANTIS generic HMI prototype, based on the data from a MANTIS use case. Note that the embedded graph is interactive and allows the user to change the time range shown or apply custom filter, as shown in this example. In cases when this is not desired, static snapshots of visualizations can be embedded instead.

To conclude, Kibana is particularly useful for (but not limited to) exploring large datasets of time-based data, such as sensor measurements and

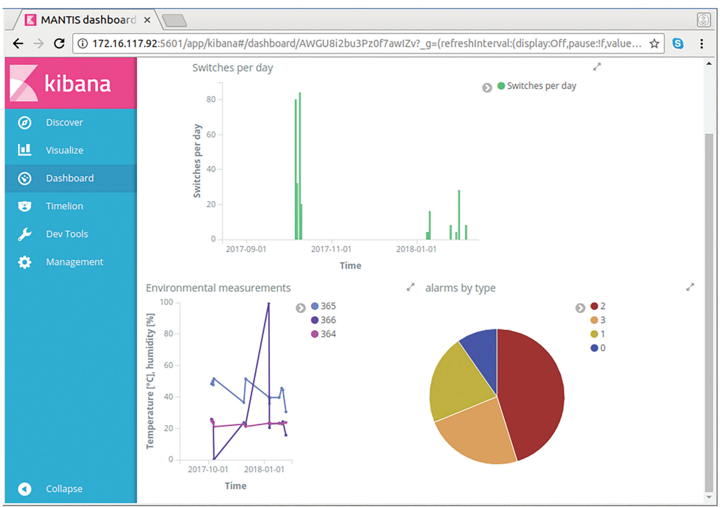


Figure 6.11 MANTIS data visualization in Kibana.

[17]Please note that the time range includes periods when the MIMOSA database contained test data rather than real sensor measurements, therefore, the values shown here may not be realistic.

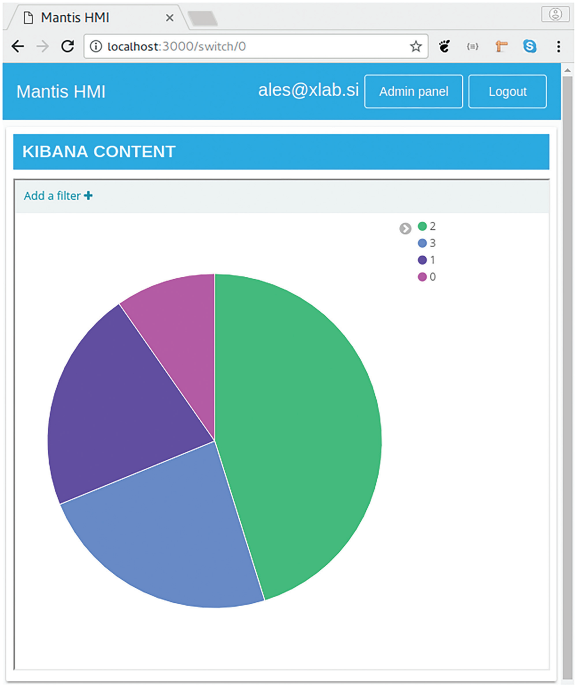


Figure 6.12 A Kibana graph embedded in the MANTIS HMI prototype.

alarms. It is being widely used for monitoring assets in cloud computing and should thus perform equally well in industrial asset monitoring.

6.3.1.3 Scenario 2: Textual and graphical data representation

Data on the machine is collected by means of sensors that are part the machine's control systems or from sensors which were added specifically for maintenance purposes. In the MANTIS sheet metal industry use case [Ferreira et al., 2017], raw data visualization includes the visualization of both historical and streamed data.

Historical Data

Historical data is related with the values detected by sensors installed in the machine and is stored in the centralized database. It can then be consulted in a table or graphical format.

Figure 6.13 shows the data collected from the machine in a table format. This data includes all the variable values that were measured during the machine operation.

Logs M00000017

List of logs received from M00000017 machine.

Show 10 entries

Search:

MachineID	Date_Time	Y1	Y2	B.X1	B.X2	B.R	B.Z1	B.Z2	LazerSafe	Y1.s	Y2.s	B.X1.s	B.X2.s	B.R.s	B.Z1.s	B.Z2.s	PedalD	PedalUp	Start	Stop	ProdSeq	FirstSeq	Auto	Bend
M00000017	2/19/2018 17:37:40.877	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.813	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.750	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.687	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.627	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.563	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.500	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.437	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.377	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
M00000017	2/19/2018 17:37:40.313	217419	217204	597590	-25000	95998	1500000	1999998	16545	0	0	0	0	1	0	0	1	0	0	0	6	1	0	0
Search Machi	Search Date_	Search	Search	Search	Searc	Searc	Search	Search	Search L	Sea	Sea	Searc	Searc	Sear	Searc	Searc	Search	Search	Sea	Sea	Search	Search	Sea	Sear

Showing 1 to 10 of 2,030,730 entries

Previous 1 2 3 4 5 … 203073 Next

Figure 6.13 Machine Data Visualization (Table).

Streamed Data

Streamed data is also related with sensor values detected however the data is received directly from the message broker in the communication middleware and is presented to the user immediately after being consumed. For this the HMI subscribes to the appropriate machine queue and as soon as new data is loaded into that queue, it is transmitted to the HMI and is converted to a graphical format.

Figure 6.14 shows the data collected from the machine in a graphic format. After the user selects the data interval and the variables to be analysed the corresponding plot will be generated with all the values recorded from the machine within the selected interval. The user may then perform different analysis operations such as variables comparison, zoom in/out, define offsets, etc.

Figure 6.15 shows data obtained from Maintenance Sensors which are placed on the machine, usually communicating over an independent channel (e.g., a wireless network). These sensors only acquire specific maintenance related information. In this case the sensor is acquiring the values regarding machine's moving parts.

6.3.2 Augmented and Virtual Reality

Virtual reality and augmented reality were chosen as the technologies to be used in the advanced HMI approaches for the Finnish conventional energy

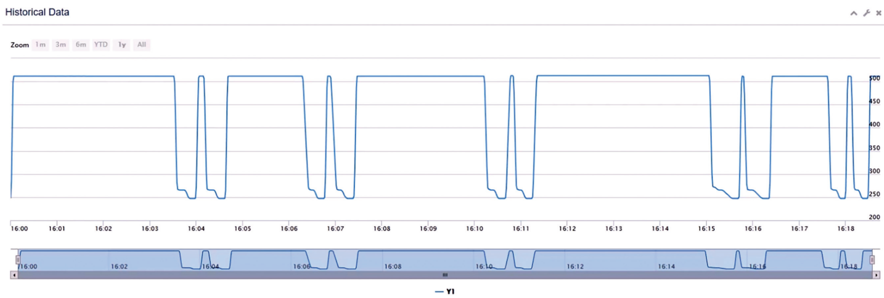


Figure 6.14 Graphic Visualization of Machine Data.

Figure 6.15 Maintenance Sensor Data Visualization.

production use case. They are an emerging market especially on consumer side and will most likely have an impact on maintenance in the future in one form or another. There is also a lot of innovation potential in these technologies.

The distinction between industrial maintenance related usage of VR and AR approaches can be roughly defined between factory-floor and back-office, where AR is more applicable for factory-floor and on the field maintenance tasks and guidance. VR is inherently more suited for back-office and other at office activities such as training and planning. VR's reliance on raw graphical computing power and, depending on the hardware solution used, external location and position hardware eliminates any possibility of it being mobile.

HTC Vive and its direct competitor Oculus, both rely on external hardware for position and location functionalities and are considered to be outside-in tracking solutions, meaning that the location and position data for the headset on the user's head comes from external beacons. However, hardware solutions such as the Microsoft Mixed Reality platform, which uses cameras attached to the headset itself to orient themselves and the controllers, are considered inside-out tracking solutions. This negates the need for external beacons and makes the VR more portable. The Microsoft MR platform also enables mixing in live, real-world stereo video using stereo cameras for a through-camera AR like approach, hence the name mixed reality.

Initially interest in the use case was placed on the AR approach, as it was more suitable for use in maintenance monitoring on the field or factory floor. The AR approach was done on the Google Tango platform that consists of a comprehensive Unity compatible AR SDK and a special hardware platform that consists of an IR dot matrix projector and a special camera capable of measuring the time-of-flight of the independent dots projected onto a shape. The combination of the SDK and the hardware platform is used to mitigate inherent drift in any pure IMU based positioning solution.

The AR application named AHMI (Advanced HMI) would enable users to create a virtual representation of the flue gas blower in the Järvenpää plant (see Figure 6.16) and retrieve real-world measurement data onto the measurement points attached to the 3D model. It also supports adding virtual measurement points to real-world objects using real-world measurement data retrieved from MIMOSA (see Figure 6.17). It also had additional features such as disassembling of 3D models (where the model allows it) and virtual post-it notes for leaving virtual messages onto the factory floor. The texts contained on the notes are stored in MIMOSA and thus could be used to gather tacit knowledge.

Development of the AR version of AHMI was halted as it became apparent that Google was halting support and development of the Tango platform. With the emergence of Apple's own proprietary augmented reality ARKit SDK, Google announced their own, mostly hardware-independent AR SDK the AR Core, which then quickly superseded the Tango platform that relied on specific hardware to be present on the device. On 15th of December 2017 Google announced that the Google Tango platform will be deprecated on 1st of March 2018[18], and that finalized the cessation of AR development for the MANTIS platform as AR Core was still not ready for release.

[18]https://twitter.com/projecttango/status/941730801791549440

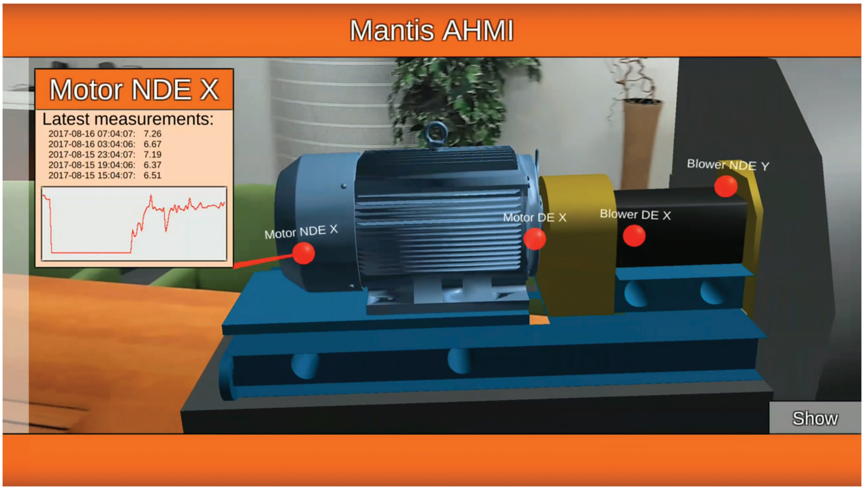


Figure 6.16 3D model of flue gas blower placed at meeting room table.

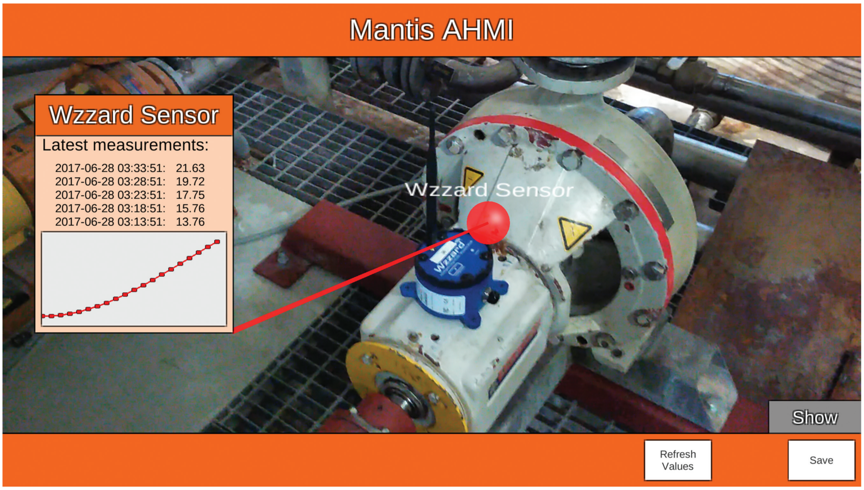


Figure 6.17 Marker object representing data from sensor.

As the fate of the Tango Platform was already quite apparent during development, it was decided to develop a VR solution based on the work done with AR. In fact, the VR application was very similar to the AHMI one as it re-used most of code, as both were developed in Unity and were therefore mostly compatible. Some additional features were introduced such as more in-depth measurement windows and the possibility of viewing spectrums and even listening to the raw vibration measurement data. The VR AHMI application was built solely on the HTC Vive platform, however it could be transferred to other VR hardware platforms such as Oculus, possibly even the Microsoft mixed reality platform.

Figure 6.18 shows the windows displaying real-world data from the Järvenpää plant. The users can open and reposition these windows to their own preferences. They also have a snap functionality that allows for neat alignment of all windows. It is possible to load FFTs and raw vibration data for any data point by moving a red indicator. It is also possible to move the indicator on all windows at the same time by enabling the indicator lock visible on the right hand corner of the window. Figure 6.19 shows these measurement windows opened to a selected data point, where the raw measurements are shown in the top windows and the FFT are shown in the lowest windows. Figure 6.20 shows closer inspection of the FFT data using gesture control enabled by Leap Motion.

Both VR and AR solutions could be utilized as a part of collaborative decision-making. The VR could be used to convene and observe anomalies online over wide geographic distances. Experts around the world could communicate with each other using avatars in a 3D space. AR could be used, for instance locally to observe machinery status. It would allow the users to load the 3D model onto a conference room table and it could be visible to all users with AR capable devices. This could allow for new business

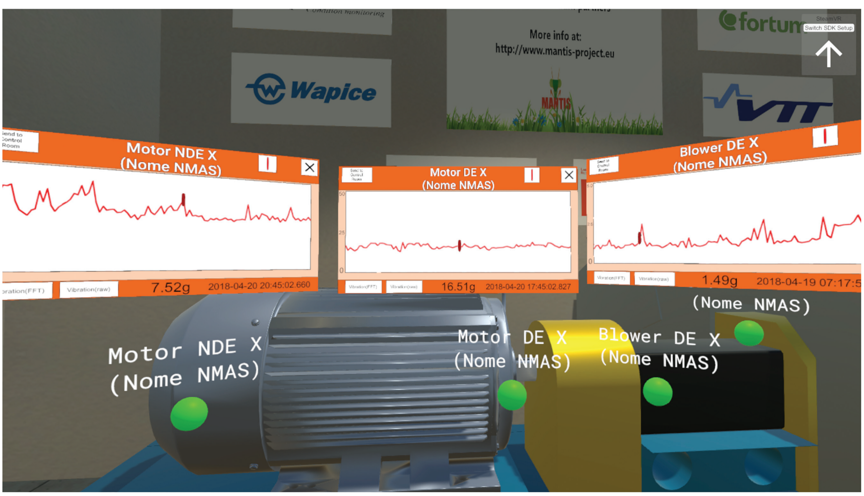


Figure 6.18 Screen capture of the VR demonstrator displaying real measurement data.

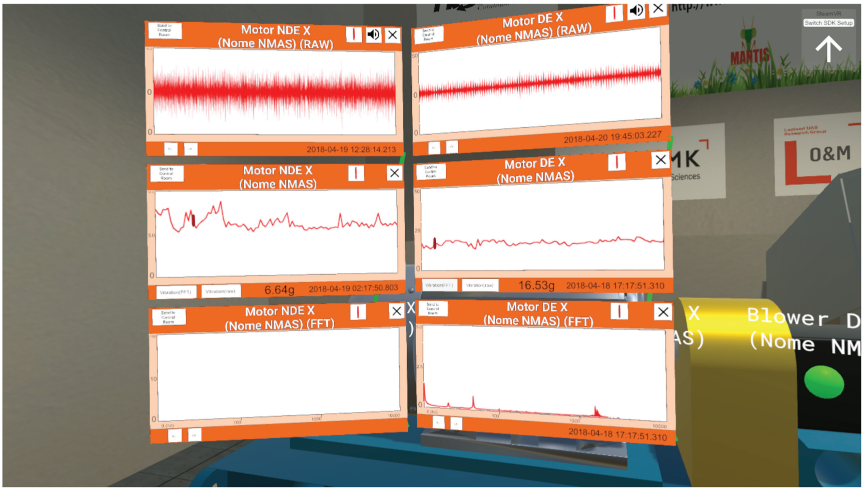


Figure 6.19 FFT and raw vibration data opened for a measurement.

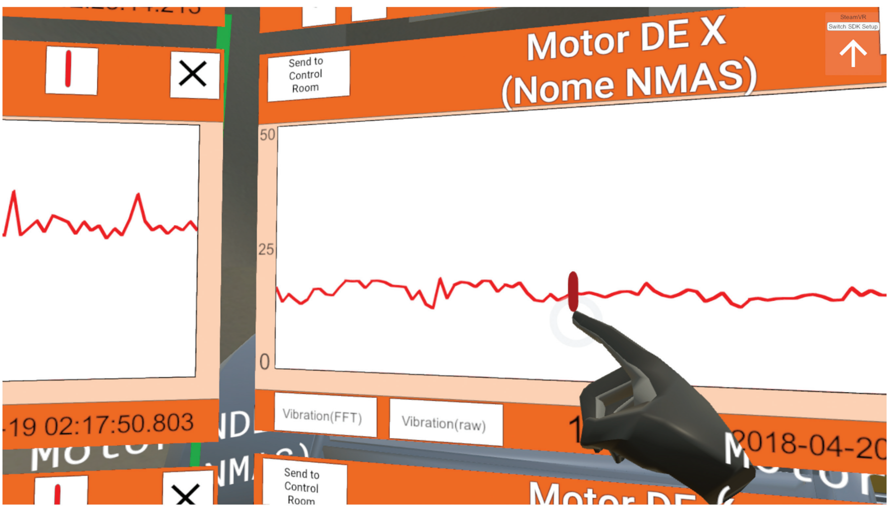


Figure 6.20 FFT data inspection with Leap Motion controls.

opportunities in maintenance related decision support and collaboration. A separate "control room" was created as a part of the VR application. It's an empty VR space where collaborators can "send" measurement windows from the VR factory floor. This enables collaborativeness in the VR world between experts.

During testing and demonstration of the VR system, utilizing the HTC Vive included controllers, observations were made that the controllers themselves makes the VR less approachable and less intuitive. The controllers were then replaced with Leap Motion controller, which is a structured light based IR projection camera system intended for recognizing hand positions and gestures. This was incorporated into the VR demo to replace the hindrance of the controllers allowing for an immediately more intuitive approach using the user's hands as control tools. The Leap Motion unit came with a cradle that allowed it to be attached to the HTC Vive headset. The application was then modified to support the Leap Motion SDK and functionalities.

In Use Case 3.3 two separate scenarios related to HMIs were identified in the requirement specifications; automated vibration monitoring and condition and incoming maintenance alert for plant operators. Each of these scenarios were implemented in the Finnish consortium partners' own HMI approaches. These scenarios were also implemented in the advanced visualization approaches, as described in the following subsections.

6.3.2.1 Scenario 1: Automated vibration monitoring

In the VR implementation, the system level is represented by the virtual models of the equipment. The component levels are represented by the virtual sensor nodes. Drilling down to the measurements is enabled by opening the individual sensor positions and accessing the measurements by picking the point in time the operator would like to examine. At its current state the VR implementation does not allow for automated opening of measurement levels, it depends on user interaction to drill down to the FFT and raw data.

All the measurement data displayed within the VR environment is obtained from the REST interface and the MIMOSA database. Due to the way the measurement data is inserted into the database, it is not real-time. However, were the data stored in the database in real-time, it could be displayed near real-time within the VR environment.

6.3.2.2 Scenario 2: Condition and incoming maintenance alert for plant operators

The VR demonstrator supports alarms via MIMOSA's alarm related tables. They are implemented in a manner that makes the alarms visible to the people using the VR application and the headset. The alarms appear as hovering items in the field of view of the operator and does not go away until the

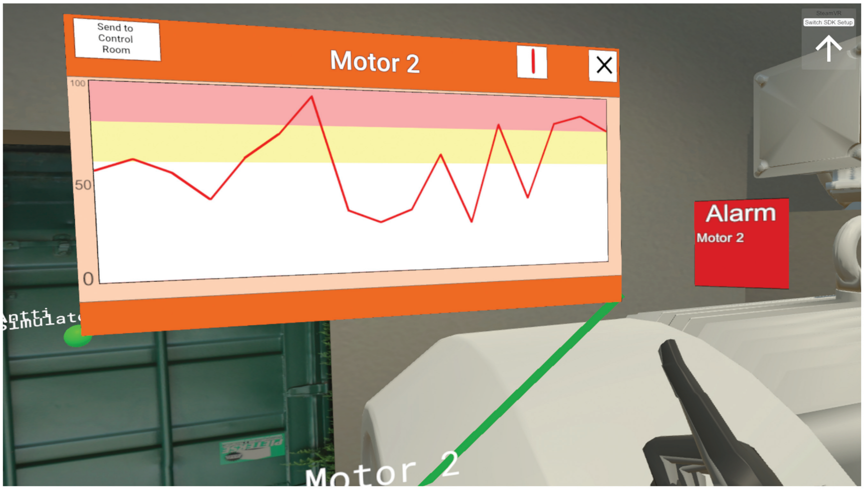


Figure 6.21 Screenshot showing a super-imposed alarm and the synthetic data that has caused it.

operator has confirmed the alarm by interacting with it with either the HTC Vive controllers or by hands.

The alarm item does also show the measurement that has caused the alarm and can provide the operator a virtual representation of the equipment or measurement point that has caused the alarm. From there the operator can observe the cause of the alarm using the normal VR demonstrator tools. Figure 6.21 shows two active alarms and the measurements that has caused the alarm. Using the VR controller to click and dismiss the alarm will teleport the user to the equipment the measurement point is attached to, which is visible in Figure 6.21 as the electric motor with the red dot attached. There are plans to incorporate user recognition to direct the alarms at correct personnel.

6.4 Usability Testing Methodology for Industrial HMIs

Usability aspects are becoming increasingly important and a common criterion associated with performance is the quality of service. Quality of service can be studied from both the system point of view and the user point of view. In this context, quality of service from the system point of view means, that the system is capable of offering and doing required functionalities which users are observing or commanding through the HMI. Quality of service from the user point of view means, that the users are capable of reading or doing those required functionalities through the HMI. Even if there are divergences for the evaluation approaches of these services we shall follow the best practice in individual aspects such as input performance, interpretation performance, context appropriateness (at the system side) and perceptual effort, cognitive workload, physical effort on the user side. In addition, quality of experience covering user perception and satisfaction issues is also gaining importance [MANTIS Consortium, 2015].

Before proceeding to the usability evaluation of specific HMI implementation, first refer to the issue of usability as defined by the standards (ISO) dealing with human computer interaction (HCI), focusing on ergonomics of human-system interaction and associated product quality.

The website of the UsabilityNet, a project funded by the European Union to promote usability and user centred design (Usability Net), provides an extensive list of standards related to HCI and usability. They are categorised into four groups, as those primarily concerned with:

- the use of the product (effectiveness, efficiency and satisfaction in a particular context of use);
- the user interface and interaction;
- the process used to develop the product;
- the capability of an organization to apply user centred design.

6.4.1 Human-system Interaction – Usability Standards

As regards usability definition, ISO 9241-11, ISO 9241-210, ISO/IEC 9126 standards and ISO/TR 16982 technical report are exposed. In the following, we borrow some parts of their descriptions in order to reveal their specific features for the purposes of the MANTIS project.

ISO 9241-11: Guidance on Usability (1998)

In this standard, usability is defined as the "*extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use*".

Hereby:

- effectiveness denotes the accuracy and completeness with which the users achieves specified goals;
- efficiency measures the resources spent in relation to the accuracy and completeness with which users achieve their goals;

- satisfaction designates the freedom from discomfort, and positive attitudes towards the use of the product;
- context of use includes a description of users, their tasks, employed equipment and physical/social environment.

In order to specify or measure usability, it is necessary to identify the goals and to decompose effectiveness, efficiency and satisfaction and the components of the context of use into sub-components with measurable and verifiable attributes.

Guidance is given on how to describe the context of use of the product and the measures of usability in an explicit way. The document also explains how measures of user performance and satisfaction can be used to measure how any component of a work system affects the quality of the whole work system in use [ISO, 1998].

ISO 9241-210: Ergonomics of Human-system Interaction – Part 210: Human-centred Design for Interactive Systems (2010)

As the title suggests, this standard deals with human-centred design for interactive systems and explains the activities required for user centred design. It is intended to be used by those managing design processes.

Principles of human-centred design are listed and described. Next, the main steps of planning human-centred design are surveyed. Once the planning is done, the following design activities are foreseen and described in more details:

- understanding and specifying the context of use;
- specifying the user requirements;
- producing design solutions;
- evaluating the design.

User-centred evaluation should involve:

- properly allocating resources, such as properly selected users, description of product functionalities and context of use (in an early stage) to obtain feedback that can be used to improve or redesign the product and (later) to determine whether the requirements have been met;
- planning the user-centred evaluation;
- carrying out comprehensive testing to provide meaningful results for the system as a whole;
- analysing the results;
- communicating the results so that they can be used by the design team.

Two widely used approaches to user-centred evaluation are

- user-based testing;
- inspection-based evaluation.

Both are described at the appropriate level of details considering the fact that the standard is intended primarily for managers and not for product developers [ISO, 2010].

ISO/IEC 9126: Software Product Evaluation – Quality Characteristics and Guidelines for their Use (1991)

This standard was developed separately as a software engineering standard. It defined usability as one relatively independent contribution to software quality associated with the design and evaluation of the user interface and interaction. This standard has been replaced by a new four-part standard ISO/IEC FDIS 9126 (2000) [ISO, 1991].

ISO/IEC FDIS 9126-1: Software Engineering – Product Quality – Part 1: Quality Model (2000)

This is the first of the four parts of ISO/IEC FDIS 9126 (2000) and describes a two-part model for software product quality:

- internal quality and external quality;
- quality in use.

Internal Quality Requirements specify the level of required quality from the internal view of the product and address implementation issues (such as employed models, source code, etc.). Internal quality represents the internal characteristics of the software product and can be evaluated against the internal quality requirements. External Quality Requirements specify the required level of quality from the external view and are derived from user quality needs. External quality thus represents the characteristics of the software product from an external view.

The quality model of the internal quality and external quality specifies six categories of software quality that are relevant during product development: functionality, reliability, usability, efficiency, maintainability and portability. Each of them is further divided in sub-categories.

Quality in use is the user's view of quality. It is specified in four categories: effectiveness, productivity, safety and satisfaction. Hereby the effectiveness, productivity and satisfaction somehow correspond to the notions of effectiveness, efficiency, and satisfaction defined in ISO 9241-11 [ISO, 2001].

ISO/IEC FDIS 9126-2: Software Engineering – Product Quality – Part 2: External Metrics and ISO/IEC FDIS 9126-3: Software Engineering – Product Quality – Part 3: Internal Metrics

As the title suggests, these two parts of the standard define metrics for quantitative measuring software quality. Each of the two parts contains an explanation of how to apply software quality metrics and a basic set of metrics for each sub-characteristic of the stated six categories. Additional explanations, such as considerations when using metrics (i.e., interpretation of measures, validation of metrics, etc.) and explanation of metric scale types and measurement types are given in informative annexes [ISO, 2003; ISO, 2003].

SO/IEC FDIS 9126-4: Software Engineering – Product Quality – Part 4: Quality in Use Metrics

Similar to the above two, this part of the standard defines metrics for quantitative measuring of software quality related to the four categories defined by the quality model for quality in use [ISO, 2004].

SO/TR 16982: Ergonomics of Human-system Interaction – Usability Methods Supporting Human-centred Design

In reference to ISO 9241-11 and ISO 9241-210 this technical report provides an overview of existing usability methods which can be used on their own or in combination to support design and evaluation. Each method is described with its advantages, disadvantages and other factors relevant to its selection and use. The purpose of this technical report is to help project managers make informed decisions about the choice of usability methods to support human-centred design principles as described in ISO 9241-210.

In order to incorporate usability requirements, the following four human-centred design activities are suggested:

- understanding and specifying the context of use;
- specifying user requirements;
- producing designs and prototypes;
- performing user-based assessment.

The described usability methods are focused either on design or evaluation. In the first case data-gathering techniques that are applied early in the design phase and are used to guide the design. The second case refers to the evaluation of design such as assessment of interface features, expected task completion time, expected use pattern, etc. The methods that are presented in this technical report are those that are most frequently used. The methods are listed in a table and divided into two broad categories:

- methods that imply the direct involvement of users;
- methods that imply the indirect involvement of users, which are used either when it is not possible to gather usage data due or where they provide complementary data and information.

After the description of individual usability methods, the preferred choice of usability methods some general guidelines for their application during the different design phases are given. The last part is devoted to the choice of usability methods depending on life-cycle process, constraints of project environment, user characteristics, characteristics of the task to be performed, the product used and the abilities required for the designer or evaluator [ISO, 2002].

While ISO 9241-11 and ISO/IEC FDIS 9126-1 address the usability in a slightly different way, the approach described in ISO 9241-11 is more closely related to the issues of the MANTIS project and we shall adopt it as the starting point of usability testing. According to ISO 9241-11, the following information is needed when measuring usability:

- a description of intended goals;
- a description of the components of the context of use including users, tasks, equipment and environments;
- target or actual values of effectiveness, efficiency, and satisfaction for the intended contexts.

According to ISO 9241-11, measures of usability should be based on data, which reflect the result of user interaction with the system. Effectiveness, efficiency and satisfaction can be measured as follows:

- Effectiveness is defined as accuracy and completeness with which the users achieve specified goals. Hereby, the accuracy can be measured to which extend the quality of the implemented HMI corresponds to the specified criteria. For example, how consistent are the implemented functionalities. The completeness can be measured as the extent of the achieved target quantity. For example, how many of the specified functionalities have actually been implemented;

- Efficiency is measured by relating the level of effectiveness achieved to the resources used. For example, for a given functionality, how long does it take to perform given task and achieve the result complying to the stated goal;
- Satisfaction can be assessed by objective and subjective measures. Objective measures are based on observation of the behaviour of the user. Subjective measures comprise data expressing user's opinions, attitudes and reactions. These data can be obtained by asking users to express their feeling when performing specific task, or by using an attitude scale based on a questionnaire.

In order to properly measure effectiveness, efficiency and satisfaction, appropriate metrics should be applied. In this regard, metrics for software product quality described in the four-part standard ISO/IEC FDIS 9126 (2000) can serve as an example or a guideline. In particular, metrics described in ISO/IEC FDIS 9126-2 and ISO/IEC FDIS 9126-3 can be applied to measuring effectiveness and efficiency of the implemented HMI; and metrics described in ISO/IEC FDIS 9126-4 for measuring user satisfaction when performing tasks on the implemented HMI, respectively.

Usability tests differ in the way that usability measurements are performed, depending on the phase of the design process in which they are applied. In ISO/TR 16982, advantages and disadvantages of each type of usability methods are described.

While all standards described previously deal with usability issues, none of them gives explicit guidelines for performing usability tests. To some minor extent, the usability testing is described in ISO 9241-210. Two widely used approaches to user-centred evaluation are: user-based testing and inspection-based testing.

User-based testing can be undertaken at any stage in the design. At a very early stage, users can be presented with models, scenarios or sketches of the design concepts and asked to evaluate them in relation to a real context. Such early testing can provide valuable feedback on the acceptability of the proposed design. At a later stage in the development, user-based testing can be carried out to assess whether usability objectives, including measurable usability performance and satisfaction criteria, have been met in the intended context.

Inspection-based evaluation complements user testing. It can be used to eliminate major issues before user testing and hence make user testing more cost-effective. Usually two to three analysts evaluate the system with reference to established guidelines or principles, noting down their observations and often ranking them in order of severity. The analysts are usually experts in human factors or HCI. However, inspection does not always find the same problems that would be found in user-based testing. The greater the difference between the knowledge and experience of the inspectors and the real users, the less reliable are the results.

For detailed description, how to perform usability testing one should rather rely upon the information and guidelines provided in (Usability Net) (Usability.gov), or in [Rubin, et al., 2008]. Based on the above resources we describe in the next section the usability testing methods.

6.4.2 Usability Testing Methodology for MANTIS

In the case of MANTIS HMI, usability testing was performed primarily as user-based testing. Usability test is performed by representative users under guidance of a skilled moderator. Users are observed when performing given tasks (e.g., the most frequent or the most critical). The collected qualitative and quantitative data related to the performed tasks serve for the improvement of the product.

Basic elements of usability testing can be summarised as follows:

- development of research questions or test objectives;
- use of a representative sample of end users;
- presentation of the actual work environment and usability tasks;
- observation of users who use the product in order to perform given task;
- interviewing the users by the test moderator;
- collection of quantitative and qualitative performance and preference measures;
- recommendation of improvements to the design of the product.

As described in [Rubin, et al., 2008], three types of usability testing are distinguished depending on the phase of a development cycle:

- exploratory study;
- assessment or summative test;
- validation or verification test.

The exploratory study is conducted early in the development cycle, when a product is still in the preliminary stages of design when its basic concept and functionalities are being defined. Some typical user-oriented questions that an exploratory study would attempt to answer might include: what do users think about using the product, are product's functionalities useful to the user, how easily can the user learn to use the product, etc.

The main objective of the exploratory study is to examine the effectiveness of preliminary design concepts. Exploratory tests are characterised by extensive interaction between the participant and the test moderator. Since the product is in the early design phase, only preliminary versions (in a form of prototype) are available for user's evaluation. During the test of such a prototype, the user would attempt to perform representative tasks or just simply express his feeling about the product under test. The user is encouraged to "think aloud" and his comments and remarks are collected for subsequent analysis and improvement of the product. Collected data is qualitative. Its benefits are twofold:

- Potential usability problems can be detected at an early stage before development is complete;
- A deeper understanding of the users' expectations and impressions of the system.

Assessment tests are performed early or midway into the product development cycle, usually after the fundamental concept and functionalities of the product have been established. The product is developed with specified functionalities but probably requires optimization and polishing. The purpose of the assessment test is to examine how effectively the concept has been implemented. Assessment test thus reveals how well a user can actually perform a typical realistic task and identifies possible usability deficiencies that manifest during the task completion. In contrast to exploratory study:

- the user will always perform tasks rather than simply walking through and commenting;
- the communication with the test moderator is less concerned about user's feeling and comments and more focused on the actual task execution.

Qualitative and quantitative data are collected. Its benefits are:

- identified deficiencies can be improved since the product is still in a development phase, with all development tools and development team available;
- possible missing functionalities can be implemented.

The validation test, also referred to as the verification test, is usually conducted late in the development cycle and, as the name suggests, it should confirm that the problems identified in the earlier phases have been solved and that the product under test operates without faults. It may also be used to measure usability of a product against established benchmarks. This test typically takes place close to the release of the product.

- Usability goals are stated in terms of performance criteria. The effectiveness and efficiency metrics are defined in accordance with the stated goals;
- Participants perform tasks without (or very little) interaction with the test moderator;
- Quantitative data related to the stated effectiveness and efficiency metrics is collected.

Exploratory test will be made in an early design phase. The main purpose of exploratory test is to help HMI developers to improve their HMI already in the design phase. The advantage of the exploratory test is that the shortcomings are noticed early enough before the first version of HMI has actually been built. The exploratory test can be performed independently for every HMI related use case.

Assessment test will be made when the basic HMI functionalities are implemented in use-case but not yet optimized. Some general guidelines how to perform assessment test for the first version of implemented HMI are given in the following section. The purpose is to help use-case owners and HMI developing partners to improve their implemented HMIs towards the final version and validation testing. The same way as exploratory testing, the assessment test can be made independently among use-case owners and HMI developing partners based on the given guidelines.

Validation is the final testing phase. The purpose of the validation test is to provide valuable feedback how to improve individual HMIs in use-cases. The main idea is that validation tests are done in each use-case according to the common guideline. In this way, the results of different use-cases can be compared and possible deviations and deficiencies removed.
The following guidelines have been adopted from (Usability Net).

Planning:

- Define goals of the performed usability testing;
- Define metrics that will be used to assess or measure to what extent the goals have been achieved;
- Select representative users. The same users can be employed in all stages of usability tests;
- Select the most important tasks (e.g., the most frequent or the most critical) related to the stated goals;

- Produce task scenarios and input data and write instructions for the user (tell the user what to achieve, not how to do it);
- Plan sessions allowing time for giving instructions, running the test, and a post-test interview;
- Invite developers to observe the sessions if possible. An alternative is to videotape the sessions, and show developers edited clips of the main issues.

Running sessions:

- Welcome the user, and give the task instructions. Ask for their consent do gather, keep and use the collected data (according to GDPR rules);
- Ask user to perform the task and record and/or measure task implementation (e.g., record user's comments and questions, measure task completion time);
- Do not give any hints or assistance unless the user is unable to complete the task;
- Observe the interaction and note any problems encountered;
- During exploratory study and in some cases also during assessment test, the user may be prompted for their impressions of the design, what they think different elements may do, and what they expect the result of their next action to be. The user may also be asked to suggest how individual elements could be improved;
- Interview the user to gain general opinions, and to ask about specific problems encountered.

Output:

- Report of the conducted usability tests, including employed metrics and achieved results;
- Based on the above results produce a list of usability problems, categorised by importance and an overview of the types of problems encountered;
- Arrange a meeting with the designers to discuss whether and how each problem can be fixed.

While the above guidelines are quite general and could be applicable to any type of usability testing, some additional commends and recommendations specific to exploratory study, assessment test, and validation or verification test in the case of MANTIS HMI given in the following paragraphs might be helpful.

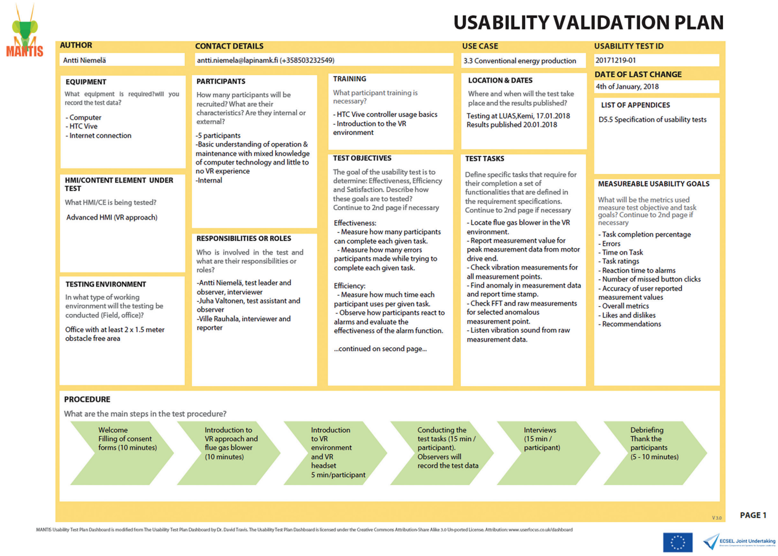
Since the initial steps toward HMI design in individual use cases have already started and each use case addresses specific user requirements, the approaches in exploratory studies and assessment tests are likely to be divergent to some extent. Nevertheless, it would be prudent to collect and document the acquired usability test data for possible exchange of "lessons learned" among use cases and for reporting.

In the early design phase, preliminary concepts are evaluated with representative users. The focus group research employs simultaneous involvement of more than one user and deals with the target product in a very preliminary form (i.e., paper drawing, screen-based prototypes) and explores users' judgements and feelings. In this way, basic functionalities of the target product are explored, and possible missing items can be identified. Such proof of concept may include also surveys and questionnaires employed in order to understand the preferences of a broader base of users. For these procedures no formal templates are foreseen in the frame of MANTIS project but use case owners are advised to document the data in an appropriate way.

Assessment test and verification test differ in the goals and the corresponding metrics. Assessment test typically establishes if the functionalities stated in the requirement specifications have been implemented. On the other hand, verification test checks the efficiency of the implemented product and typically measures the time and resources required to accomplish given tasks. For example, the assessment test can be employed to check the functional implementation completeness in the following way: count the number of missing functions detected in evaluation and compare with the number of functions described in the requirement specifications. The goal of a verification test could be, for example, to verify that 70 percent of the users could meet the established successful completion criteria for each major task scenario.

When conducting usability testing it is good to have a validation plan. In MANTIS one of the methods to plan this activity was to fill in a two-page, interactive dashboard-like planning template. This offers a concise yet rather comprehensive way to plan the usability testing procedure. It is based on David Travis' single page usability testing dashboard document. The original document was expanded and modified slightly with additional space for objectives, test tasks and measureable usability goals descriptions. This was done in order to make it more suitable for MANTIS and industrial HMI testing purposes. The idea was to provide a comprehensive dashboard for testing procedures.

Figures 6.22 and 6.23 are screen captures for a validation plan done for a VR based advanced HMI approach developed by Lapland University of Applied Sciences. This document, together with the report template, was



USABILITY VALIDATION PLAN

AUTHOR: Antti Niemelä

CONTACT DETAILS: antti.niemela@lapinamk.fi (+358503232549)

USE CASE: 3.3 Conventional energy production

USABILITY TEST ID: 20171219-01

DATE OF LAST CHANGE: 4th of January, 2018

EQUIPMENT
What equipment is required?will you record the test data?
- Computer
- HTC Vive
- Internet connection

HMI/CONTENT ELEMENT UNDER TEST
What HMI/CE is being tested?
Advanced HMI (VR approach)

TESTING ENVIRONMENT
In what type of working environment will the testing be conducted (Field, office)?
Office with at least 2 x 1.5 meter obstacle free area

PARTICIPANTS
How many participants will be recruited? What are their characteristics? Are they internal or external?
-5 participants
-Basic understanding of operation & maintenance with mixed knowledge of computer technology and little to no VR experience
-Internal

RESPONSIBILITIES OR ROLES
Who is involved in the test and what are their responsibilities or roles?
-Antti Niemelä, test leader and observer, interviewer
-Juha Valtonen, test assistant and observer
-Ville Rauhala, interviewer and reporter

TRAINING
What participant training is necessary?
- HTC Vive controller usage basics
- Introduction to the VR environment

TEST OBJECTIVES
The goal of the usability test is to determine: Effectiveness, Efficiency and Satisfaction. Describe how these goals are to tested? Continue to 2nd page if necessary

Effectiveness:
- Measure how many participants can complete each given task.
- Measure how many errors participants made while trying to complete each given task.

Efficiency:
- Measure how much time each participant uses per given task.
- Observe how participants react to alarms and evaluate the effectiveness of the alarm function.

...continued on second page...

LOCATION & DATES
Where and when will the test take place and the results published?
Testing at LUAS,Kemi, 17.01.2018
Results published 20.01.2018

TEST TASKS
Define specific tasks that require for their completion a set of functionalities that are defined in the requirement specifications. Continue to 2nd page if necessary
- Locate flue gas blower in the VR environment.
- Report measurement value for peak measurement data from motor drive end.
- Check vibration measurements for all measurement points.
- Find anomaly in measurement data and report time stamp.
- Check FFT and raw measurements for selected anomalous measurement point.
- Listen vibration sound from raw measurement data.

LIST OF APPENDICES
D5.5 Specification of usability tests

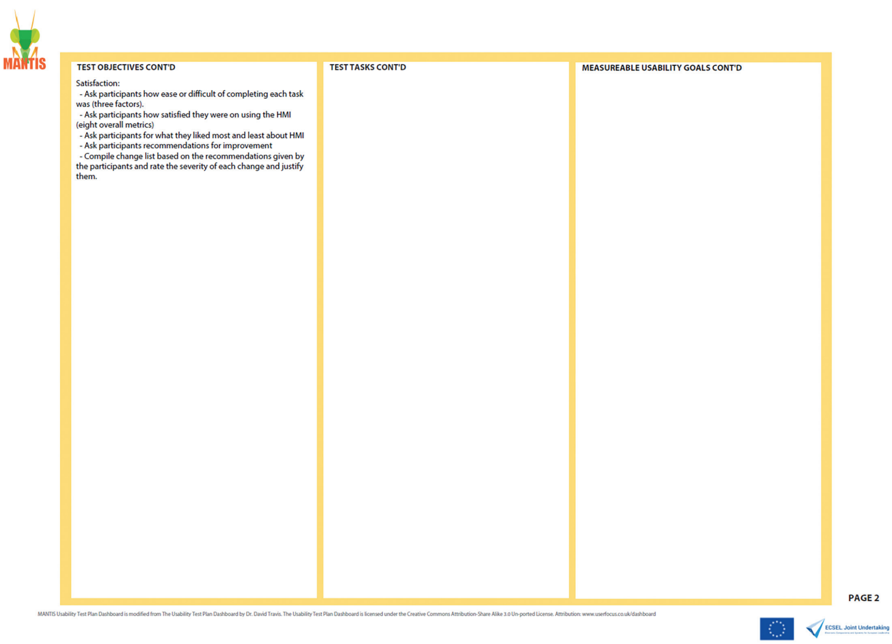
MEASUREABLE USABILITY GOALS
What will be the metrics used measure test objective and task goals? Continue to 2nd page if necessary
- Task completion percentage
- Errors
- Time on Task
- Task ratings
- Reaction time to alarms
- Number of missed button clicks
- Accuracy of user reported measurement values
- Overall metrics
- Likes and dislikes
- Recommendations

PROCEDURE
What are the main steps in the test procedure?
Welcome Filling of consent forms (10 minutes) → Introduction to VR approach and flue gas blower (10 minutes) → Introduction to VR environment and VR headset 5 min/participant → Conducting the test tasks (15 min / participant). Observers will record the test data → Interviews (15 min / participant) → Debriefing Thank the participants (5 - 10 minutes)

PAGE 1

MANTIS Usability Test Plan Dashboard is modified from The Usability Test Plan Dashboard by Dr. David Travis. The Usability Test Plan Dashboard is licensed under the Creative Commons Attribution-Share Alike 3.0 Un-ported License. Attribution: www.userfocus.co.uk/dashboard

Figure 6.22 An example of a filled Usability Validation Plan document page 1.



TEST OBJECTIVES CONT'D
Satisfaction:
- Ask participants how ease or difficult of completing each task was (three factors).
- Ask participants how satisfied they were on using the HMI (eight overall metrics)
- Ask participants for what they liked most and least about HMI
- Ask participants recommendations for improvement
- Compile change list based on the recommendations given by the participants and rate the severity of each change and justify them.

TEST TASKS CONT'D

MEASUREABLE USABILITY GOALS CONT'D

PAGE 2

MANTIS Usability Test Plan Dashboard is modified from The Usability Test Plan Dashboard by Dr. David Travis. The Usability Test Plan Dashboard is licensed under the Creative Commons Attribution-Share Alike 3.0 Un-ported License. Attribution: www.userfocus.co.uk/dashboard

Figure 6.23 An example of a filled Usability Validation Plan document page 2.

used to conduct the usability testing of the developed VR HMI. The same validation plan templates were also used in other use cases successfully.

The usability validation plan was done in a dashboard form factor. This was then released as a PDF document template with two pages and multiple prepared form fields that allow for insertion of text. Each text field also had a tooltip that would guide how to fill each field accordingly. The tooltip was a mouse-over based event and would appear once the user moves the mouse over the form field similar to those used in webpages. Each usability validation plan document was then saved as its own PDF document for the use case usability testing.

The document is divided into 12 separate sections that further explain the testing procedure and requirements. There are also MANTIS specific fields that tie the planned test to the use case and related content or HMI element. Text intensive fields such as the test objectives, tasks and measurable usability goals have been extended to the second page in order to fit more content. It is advised however to try to keep the content as short as possible to keep the plan as concise and easy to approach as possible.

The ultimate goal of MANTIS HMI is to provide interaction facilities that comply to the established usability goals: effectiveness, efficiency, and satisfaction. In order to help use case owners to perform usability tests and report the results in a way that would allow easy comparison and exchange of good practices.

References

Abowd G. D. [et al.] (1999) *Towards a Better Understanding of Context and Context-Awareness*, Handheld and ubiquitous computing (pp. 304–307). Springer Berlin Heidelberg.

Ahmadzadeh F., Ghodrati, B. and Kumar U. (2012) 'Mean Residual Life Estimation Considering Operating Environment International Conference on Quality, Reliability, Infocom Technology and Industrial Technology Management. Delhi, India.

Atterer Richard, Wnuk Monika and Schmidt Albrecht (2006) 'Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction,' *Proceedings of the 15th international conference on World Wide Web*, pp. 203–212.

Barkhuus L. and Dey A. (2003) 'Is context-aware computing taking control away from the user? Three Levels of Interactivity Examined,' *Proceedings of the Fifth Annual Conference on Ubiquitous Computing (UBICOMP 2003)*, Springer Berlin Heidelberg, pp. 149–156.

Boy G. A. (2011) The Handbook of human–machine interaction: A human-centered approach [Book]. [s.l.]: Aldershot, 2011.

Chantzara M. and Anagnostou M. (2005) Evaluation and selection of context information, In Second International Workshop on Modeling and Retrieval of Context.

Coutand O. [et al.] (2006) A Case-based Reasoning Approach for Personalizing Location-aware Services Workshop on Case-based Reasoning and Context Awareness.

Dev Himel and Zhicheng Liu (2017) 'Identifying Frequent User Tasks from Application Logs,' *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pp. 263–273.

Diaper Dan and Stanton Neville (2004) *The Handbook of Task Analysis for Human-Computer Interaction*. London: Lawrence Erlbaum Associates, Publishers.

Dunlop Mark and Brewster Stephen (2002) The challenge of mobile devices for human computer interaction, *Personal and Ubiquitous Computing*, pp. 235–236.

Ehlert Patrick (2003) Intelligent user interfaces. *Introduction and survey*.

Endres C. [et al.] (2010) Cinematic analysis of automotive personalization, In Proceedings of the International Workshop on Multimodal Interfaces for Automotive Applications (MIAA'10) in Conjunction with IUI, pp. 1–6.

Feng J. and Liu Y. (2015) Intelligent context-aware and adaptive interface for mobile LBS, *Computational Intelligence and Neuroscience*.

Ferreira, L. L., Albano, M., Silva, J., Martinho, D., Marreiros, G., di Orio, G., Maló, P., and Ferreira, H. (2017) 'A Pilot for Proactive Maintenance in Industry 4.0,' In *13th IEEE International Workshop on Factory Communication Systems (WFCS 2017)*. Trondheim, Norway.

Ferreiro Susana [et al.] (2012) 'Application of Bayesian networks in prognostics for a new Integrated Vehicle Health Management concept,' *Expert Systems with Applications*, pp. 6402–6418.

Ferscha A., Vogl S. and Beer W. (2001) 'Context sensing, aggregation, representation and exploitation in wireless networks,' *Scalable Computing: Practice and Experience*.

Gardner Brett S. (2011) 'Responsive web design: Enriching the user experience,' *Sigma Journal: Inside the Digital Ecosystem*, pp. 13–19.

Garzon S. R. and Poguntke M. (2011) The personal adaptive in-car HMI: Integration of external applications for personalized use, In *International Conference on User Modeling, Adaptation, and Personalization*. Springer Berlin Heidelberg, pp. 35–46.

Hashemi Mohammad and Herbert John (2016) User interaction monitoring and analysis framework, *Proceedings of the International Conference on Mobile Software Engineering and Systems*, pp. 7–8.

Hashiguchi T., Matsuo H. and Hashizume A. (2003) 'Healthcare dynamics informatics for personalized healthcare,' *Hitachi Review*, pp. 183–188.

Hynes G., Reynolds V. and Hauswirth M. (2009) 'A Context lifecycle for web-based context management services,' *Smart Sensing and Contextser. Lecture Notes in Computer Science*, Heidelberg: Springer Berlin, pp. 51–65.

ISO ISO 9241-11:1998 (1998) Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability.

ISO ISO 9241-210: (2010) Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems.

ISO ISO International Organization for Standardization. 10 April 2017. https://www.iso.org/standards.html.

ISO ISO/IEC 9126: Software product evaluation – Quality characteristics and guidelines for their use. 1991.

ISO ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model. 2001.

ISO ISO/IEC TR 9126-2:2003 Software engineering – Product quality – Part 2: External metrics. 2003.

ISO ISO/IEC TR 9126-3:2003 Software engineering – Product quality – Part 3: Internal metrics. 2003.

ISO ISO/IEC TR 9126-4:2004 Software engineering – Product quality – Part 4: Quality in use metrics. 2004.

ISO ISO/TR 16982:2002 Ergonomics of human-system interaction – Usability methods supporting human-centred design. 2002.

Jämsä-Jounela S. L. [et al.] (2013) 'Outline of a fault diagnosis system for a large-scale board machine,' *The International Journal of Advanced Manufacturing Technology*.

Kiseleva Julia (2013) 'Context mining and integration into predictive web analytics,' *Proceedings of the 22nd International Conference on World Wide Web.*, ACM, pp. 383–388.

Kobsa A. (2011) 'Generic user modeling systems,' *User modeling and user-adapted interaction*, pp. 49–63.

Koutkias V. G., Meletiadis S. L., and Maglaveras N. (2001) 'WAP-based personalized healthcare systems' *Health Informatics Journal*, pp. 183–189.

Krueger M. [et al.] (2014) 'A data-driven maintenance support system for wind energy conversion systems, *IFAC Proceedings Volumes*, pp. 11470–11475.

Langley Pat (1997) 'Machine learning for adaptive user interfaces,' *Annual Conference on Artificial Intelligence*, Springer, pp. 53–62.

Lathia N. [et al.] (2012) 'Personalizing mobile travel information services,' *Procedia-Social and Behavioral Sciences*, pp. 1195–1204.

Lee E. A. (2008) 'Cyber physical systems: Design challenges,' In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, IEEE, pp. 363–369.

Ma T. [et al.] (2005) 'Context-aware implementation based on CBR for smart home,' *International Conference on Wireless And Mobile Computing, Networking And Communications*, IEEE.

MANTIS Consortium D1.5 MANTIS Platform Requirements V2. 2016.

MANTIS Consortium D5.1 List and Description of Different Maintenance Scenarios. 2016.

MANTIS Consortium MANTIS - Cyber Physical System based Proactive Collaborative Maintenance Full Project Proposal (FPP). 2015.

Marcotte Ethan On Being "Responsive" [Online]. 14 09 2010: http://unstoppablerobotninja.com/entry/on-being-responsive.

Market Share Statistics for Internet Technologies [Online]. 2016. 17 08 2016. https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qptimeframe=Y.

Maybury Mark T. and Wahlster Wolfgang (1998) *Readings in Intelligent User Interfaces*, Morgan Kaufmann.

Miller Michael (2008) *Cloud computing: Web-based Applications that Change the Way you Work and Collaborate Online*, Que publishing.

Nagamatsu T. [et al.] (2003) 'Information support for annual maintenance with wearable device,' In *Proceedings of HCI International*, pp. 1253–1257.

Nguyen Phong [et al.] (2017) A Visual Analytics Approach for User Behaviour Understanding through Action Sequence Analysis.

Nicolai T. [et al.] (2006) 'Wearable computing for aircraft maintenance: Simplifying the user interface, In *Applied Wearable Computing (IFAWC) 3rd International Forum*, VDE, pp. 1–12.

Nicolai T., Sindt, T., Kenn, H., & Witt, H. (2005) 'Case study of wearable computing for aircraft maintenance,' In *IFAWC-International Forum on Applied Wearable Computing*, VDE VERLAG GmbH.

Niggemann O. [et al.] (2015) 'Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control, *26th International Workshop on Principles of Diagnosis*, Paris.

Norcio A. F. and Stanley J. (1989) 'Adaptive human-computer interfaces: A literature survey and perspective,' *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 399–408.

Perera C., Zaslavsky, A., Christen P. and Georgakopoulos D. (2014) 'Context aware computing for the internet of things: A survey,' *IEEE Communications Surveys & Tutorials*, pp. 414–454.

Poklukar, Špela, Papa, G., and Novak, F. (2018) 'A formal framework of human machine interaction in proactive maintenance - MANTIS experience,' *Automatika*, 58(4), pp. 450–459. DOI: 10.1080/00051144.2018.1465226

POWER-OM, Lulea Technology Univ. Product Reliability Evaluation Tools(Power consumption driven Reliability, Operation and Maintenance optimisation) Internal deliverable report D7.3 [Report]. 2015.

Rubin Jeffrey and Chisnell Dana (2008) *Handbook of Usability Testing*, Willey Publishing Inc.

Schilit B., Adams N., and Want R. (1994) 'Context-aware computing applications,' In *Mobile Computing Systems and Applications, WMCSA 1994. First Workshop on IEEE*, pp. 85–90.

Soh Harold [et al.] (2017) 'Deep sequential recommendation for personalized adaptive user interfaces,' *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pp. 589–593.

Stiefmeier T. [et al.] (2008) 'Wearable activity tracking in car manufacturing,' *IEEE Pervasive Computing*, pp. 42–50.

Swanson D. C., Spencer J. M., and Arzoumanian S. H. (2000) 'Prognostic modelling of crack growth in a tensioned steel band,' *Mechanical systems and signal processing*, pp. 789–803.

Techopedia Techopedia [Online]. 2011. 10 January 2017. https://www.techopedia.com/definition/12829/human-machine-interface-hmi.

Thaduri A., Kumar U., and Verma A. K. (2014) 'Computational intelligence framework for context-aware decision making,' *International Journal of System Assurance Engineering and Management*, pp. 1–12.

Usability Net Usability Net [Online] // Usability Net. 10 April 2017. http://www.usabilitynet.org/home.htm.

Usability.gov Usability.gov [Online]. 10 April 2017. https://www.usability.gov/.

Wigdor Daniel and Wixon Dennis (2011) *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*, Elsevier.

Wikipedia User Interface [Online]. 2017. https://en.wikipedia.org/wiki/User_interface.

Wrona K. and Gomez L. (2006) 'Context-aware security and secure context-awareness in ubiquitous computing environments,' In *Annales UMCS, Informatica*, pp. 332–348.

Zhang D., Yu Z., and Chin C. Y. (2005) 'Context-aware infrastructure for personalized healthcare,' *Studies in health technology and informatics*, pp. 154–163.

Zhang Z., He X., and Kusiak A. (2015) 'Data-driven minimization of pump operating and maintenance cost,' *Engineering Applications of Artificial Intelligence*, pp. 37–46.

Zouhaier L., Hlaoui Y. B., and Ayed L. J. B. (2013) 'Building adaptive accessible context-aware for user interface tailored to disable users,' In *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*.