Collaborative Management of Inspection Results in Power Plant Turbines

Daniel Gonzalez-Toledo, Maria Cuevas-Rodriguez and Susana Flores-Holgado

Abstract This chapter presents an industrial case study that investigates a collaborative tool for use in the fossil and nuclear power plant industries. The tool makes the results of technical inspections on fossil and nuclear power plants available to all stakeholders and assists in the post-inspection decision-making process by highlighting decisions that minimise the outage duration and prolong the turbine's service life. Before development commenced, an actor-product-service (APS) model was employed to establish the problem, the process of which is presented in this chapter. This model describes the relationships between the elements that the system must store and manage. In this particular industrial case, the APS model defines the product as the power plant turbine and the service as technical inspections. Henceforth, this model describes the relation between the inspection tasks and results and the turbine parts that are inspected. In addition, the APS model allows the application to work jointly with the product and the service, representing the information in a way closer to the mental model of each user profile, which should result in an improvement in productivity.

Keywords Collaborative tools · 3D interaction · 3D viewer · Web-based application · Inspection services · Turbine maintenance

D. Gonzalez-Toledo (✉) · M. Cuevas-Rodriguez
DIANA Research Group, Departamento de Tecnología Electrónica,
ETSI Telecomunicación, Universidad de Málaga, Malaga, Spain
e-mail: dgonzalezt@uma.es

M. Cuevas-Rodriguez
e-mail: mariacuevas@uma.es

S. Flores-Holgado
Materials and Life Management, Tecnatom, San Sebastián de los Reyes, Spain
e-mail: sflores@tecnatom.es

S.N. Grösser et al. (eds.), *Dynamics of Long-Life Assets*,
DOI 10.1007/978-3-319-45438-2_11

1 Introduction

Tecnatom is an engineering company that provides services to a number of markets, including the nuclear energy, combined cycle and thermal, aircraft and aerospace, transport, and petrochemical markets. The company's main activity is performing inspection services and training operation personnel by means of full-scope simulators to support plant operations.

As part of the evaluation of the structural integrity of nuclear power plants' components and industrial facilities in general, Tecnatom performs inspections and tests based on applicable standards. Once an inspection has been carried out, the results are recorded, transmitted and evaluated. When defects are detected, the flow of information becomes crucial because there are many actors involved, including the companies that design, supervise and manufacture the turbines; the companies that perform the inspections; engineering companies; the power plant managers; and maintenance and repair companies. It is essential that those involved understand the problem, share information, analyse the results and propose a solution in the shortest amount of time possible.

Within the UIW-context, this industrial case is centred on the power plant steam turbine (Fig. 1). Turbines are long-lasting, high-investment components whose operation directly affects power generation and hence productivity. A collaborative tool that manages the inspections carried out on turbine components and the results has the potential to contribute to improving company services.

1.1 *Company Necessities*

During the last 15 years, Tecnatom has developed and successfully established a software tool to manage inspections and testing plans and the results of such inspections in several Spanish power plants. The system also stores all the

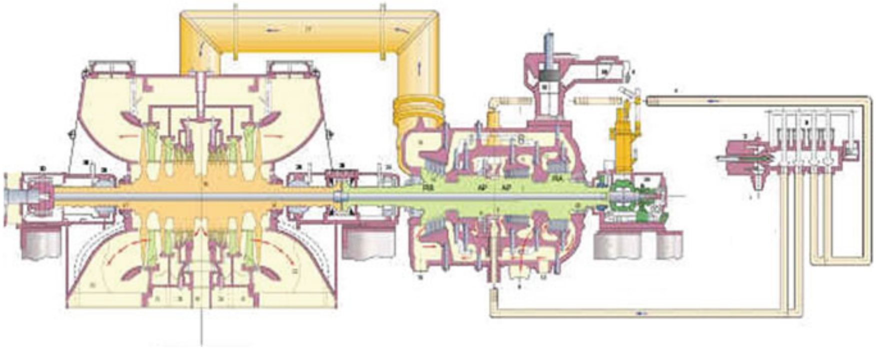


Fig. 1 Typical turbine-generator set scheme

information required by technicians to fulfil inspection tasks, including inspection and maintenance procedures; information regarding inspection areas, techniques, and frequencies for each component; and 2D drawings of systems and components. Figure 2 shows a schematic diagram of the information flow managed by the system.

Although the system has had prolonged success, some users have suggested that improvements are needed to obtain a more reliable tool and provide added-value services in a collaborative environment by supporting the decision-making process throughout the life cycle of power plants components. This updated application would allow designers and engineers to analyse the problem, propose a definitive solution and even modify the design to avoid similar problems in future versions of the product.

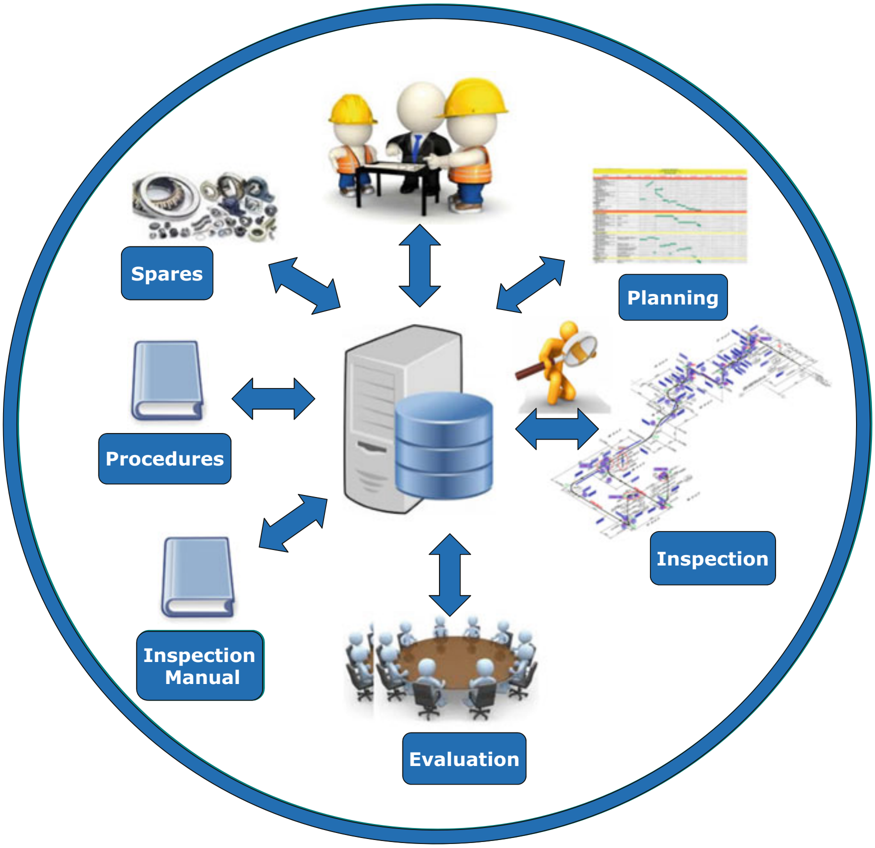


Fig. 2 Flow of information and working team

In relation to the Tecnatom industrial case, added-value services would be the 3D visualisation of the whole turbine and 3D interaction with turbine components (Bowman et al. 2004), along with relevant information about the turbine, linked to the 3D model (Elmqvist and Tsigas 2008). These services would allow the user to more quickly assess the situation when a problem arises, resulting in the optimal solution in a short period of time. In addition, the system could create an environment that allows decisions to be made among all the stakeholders in a collaborative way while registering and storing the comments and the agreed solution for future use. This would allow comparisons to be made when issues arise with similar components by taking into account lessons learned or previously stored operating experiences.

In summary, the main goal of this industrial case is to investigate the different possibilities and technologies for the development of an innovative collaborative prototype system that will work as a decision support tool for the life-cycle management of power plant steam turbines. Taking into account the company requirements, the aim of this industrial case study is to provide (1) interactive 3D models of the turbines, (2) visualisation of augmented information in the 3D models to understand the structure and issues, (3) information linked to the 3D model regarding the inspection results and (4) a discussion management tool to share information and comments related to inspection results.

1.2 Industrial Case Approach in the Use-It-Wisely Project Context

In Chapter "The Use-It-Wisely (UIW) Approach" of this book, common challenges of industrial cases were identified and organised within a framework (see Chapter "The Use-It-Wisely (UIW) Approach", Fig. 7) that contains three different domains: (1) market and data analysis, (2) collaboration management and (3) actor-product-service (APS) modelling. These three domains cover the challenges that manufacturing industries such as Tecnatom experience in providing services for high value, long-life products related to the upgrade initiating process.

Considering the company needs presented in the previous subsection, the challenges to be addressed can be allocated into two framework domains: APS modelling and collaboration management. The APS modelling domain organises all the information related to the turbine (the product), the inspections tasks and results (the service), whereas the collaboration management domain includes a discussion management tool to assist in optimal decision-making and a 3D application that depicts the inspection process, allowing for the visualisation and management of a turbine's technical information in a 3D interactive environment.

2 Modelling the Problem, from Theory Towards Implementation

To improve the management of inspection results in power plant turbines using the collaborative tool, the industrial case problem must be modelled. Therefore, this section aims to describe the specifications related to the industrial case problems, including a brief summary of the system use cases and requirements in the first sub-section, the APS model in the second sub-section, and the proposed implementation approach and system architecture in the third subsection. A conceptual prototype of the tool and a description of the industrial case can also be found in Reyes-Lecuona et al. (2014).

2.1 *Requirements and Use Cases*

This section describes an industrial use-case model of the system that can solve the problem of information flow among actors involved in the overall management of turbine inspections. In this industrial case, these actors are an inspection team (in charge of planning and performing inspections and informing on the results), an engineering team (in charge of analysing results and inputting them into the model), a plant team (representing different technicians and workers from the power plant) and administrators (technicians who are in charge of managing the model to create, edit, complete and adjust instantiations of each turbine with the Tecnatom databases).

A wide list of requirements has been defined to specify the system in a technical way. These include requirements related to how the system represents the inspection results and links them to the 3D model of the turbine, as well as the requirement for visualisation and discussion management tools. These requirements are associated with a set of system use cases, which are listed below.

- *Activity login*. The system identifies every actor before giving the actor access to the system to control which information and functionalities are available.
- *Inspection result input*. The Tecnatom inspection team inputs the results of an inspection into the system. If needed, the actor opens and prepares a discussion related to these results.
- *Visualising information*. The actors navigate through the turbine model, obtaining information about the different parts of the turbine, specific inspection data (defects/flaws, repairs performed, etc.) or information related to the corrective maintenance of a specific part. The actors visualise this information supported by the 3D geometry of the turbine.
- *Input location and size of defects*. When flaws are discovered during an inspection, their position and size can be registered into the model. Inspectors are able to graphically sketch the location of the defects using the system.

- *Management of discussions*. The actors can collaborate in a discussion related to a registered inspection point. The actors may add proposals to the discussion and relate it to other discussions until it is closed by an authorised actor.

2.2 Actor Product Service Model

From a high-level perspective, the APS model aims to describe the different business elements of the company and the relationships between them. These elements refer to both human and non-human factors. In other words, the APS model aims to detail the relationships between the workers and/or departments of the company and third-party customer companies and to model the information needed to manage the work and information flows among the relevant stakeholders.

In this industrial case, the APS model design was built based on the model-based systems engineering methodology defined within the context of the project Virtual Spacecraft Design (Rey 2013), which is described in Chapter "Extending the System Model". The model focuses on identifying the structure of any relevant information that the system has to store and manage to provide the needed functionalities. It was produced through a functional analysis based on the requirements and use cases shown in the previous subsection. After several revisions, the model shown in Fig. 3 was reached.

The model consists of two elements: the product, which involves the power plant turbine, and the service, which involves the inspection of the turbine for maintenance purposes. Actors can also be classified into two categories: (1) those who approach the problem from the point of view of the product and (2) those who approach the problem from the point of view of the service. The first category of

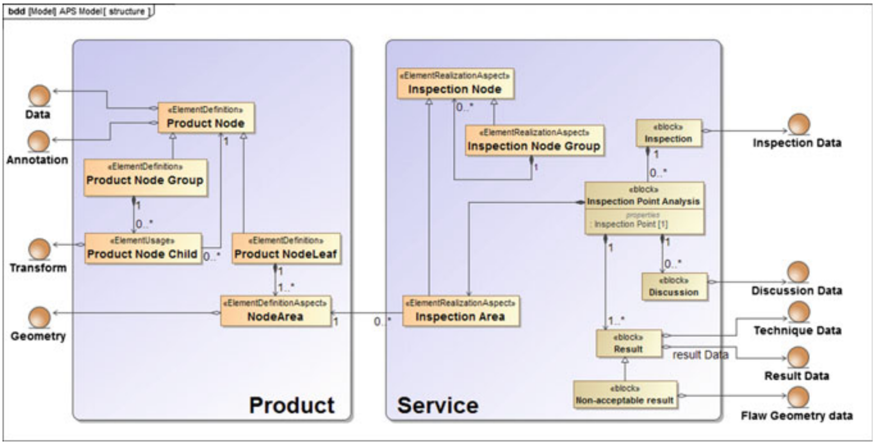


Fig. 3 Actor-product-service model diagram

actors consists of technicians and workers from the power plant who are interested in the product and its operation as a whole. The second category consists of technicians from the maintenance company who are interested in the planning, execution and analysis of the inspection results and who will be entering these data into the model.

The APS diagram presented in Fig. 3, based on SysML modelling languages (Friedenthal et al. 2014), represents both the product and service. These two models are organised in hierarchical trees and are connected to each other by their leaf nodes. After several attempts, this structure arose as the best way to organise the information.

In the product model, each node represents a component of the turbine in such a way that every component consists of the assembly of its children. The product model represents how the whole product is formed or assembled by the sub-systems, which in turn are formed by more basic sub-systems and so on, until the basic components are reached. The leaves of the product model tree are formed by the basic parts of the product, which does not mean they are small or simple; rather, they are simply pieces that are not formed by others. These blocks are shown in Fig. 3 as *NodeLeafs*.

Hanging from the leaves are areas that are not part of the product itself but are areas of interest within the *NodeLeaf* components. These areas are an important concept because different areas of a component have different physical requirements and are not subject to the same conditions. Therefore, the model specifies each of these areas (called *NodeArea* in Fig. 3).

The service model is formed by the inspection tree and the results. Whereas the product tree was intended to show the hierarchical structure of the product, the service tree has been created to express the way in which inspections are performed. Thus, the nodes of this tree represent not the physical parts of the product but parts or layers of the inspection. In this way, the model represents the structure and organisation of the whole turbine inspection. Just as the product structure is stable, so too is the service model structure throughout the turbine life-cycle.

The leaf nodes of the inspection tree are the *Inspection Areas*, which are the points that are going to be inspected. The service model represents turbine sections that are inspected at the same time (inspection areas), regardless of whether they are part of the same physical component of the product. It is important to keep together parts that are physically close or under the same operation/environmental conditions. As shown in Fig. 3, the two models are related by the areas formed by one or more products.

The model presents a block called *Inspection Point Analysis* that is associated with each inspection area. This is where new results and the associated data are stored after every inspection. In addition, the model allows for discussions related to each inspection point. This feature allows the technicians to analyse the results in a collaborative way. In addition, by keeping the entire history of decisions over the product stored together with its model, the risk of information fragmentation is avoided. In the service model, the discussion block hangs from either the Inspection

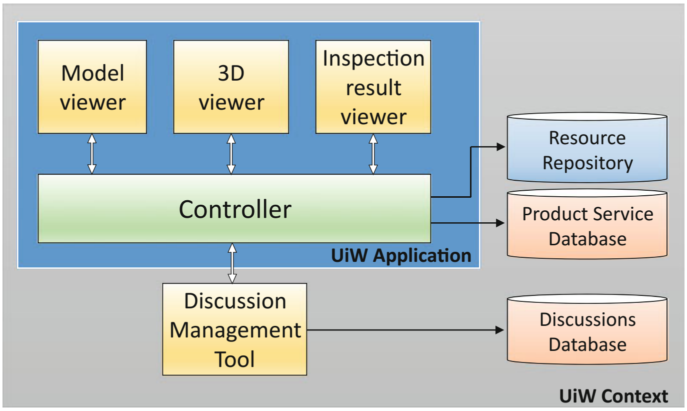


Fig. 4 Block architecture of the system

Point Analysis or from each result. However, discussions for decision-making are often based on several results, which is why this discussion block is associated with the Inspection Point Analysis block.

All the hierarchical information regarding products and services is stored in a database called *Product Service database* (Fig. 4). This database also contains all the data regarding the service (inspections, techniques, results and flaw geometry) and the product (technical specifications, annotations and geometrical transformations). However, the 3D models files, inspections result pictures, technical documents and other additional information are stored in a repository called *Resources repository*. All the information regarding discussion management is stored in its own database.

2.3 Implementation Approach

To meet the industrial case requirements, a system architecture composed of four main modules was designed (described in more detail in Sect. 3):

- *Model viewer*. This module is an interactive viewer in which the user can navigate the different elements of the product and service models using a hierarchical tree (presented in the previous subsection).
- *3D viewer*. This module consists of an interactive viewer that shows the three-dimensional geometry model of the product (the power plant turbine) and presents information about the service (inspection results) linked to the 3D product model.
- *Interactive Inspection result viewer*. This third module is a viewer that shows the inspection tasks and results stored in the database (*Product Service database*) and in the repository (*Resources repository*).

- *Discussion management tool*. This module is an external application that allows users to discuss the inspection results, make proposals, make comparisons with other results, etc. The purpose of these discussions is to achieve a final decision about how to proceed after carrying out inspections. This module is a customisation of an existing opensource tool called Redmine (Redmine 2006).

Figure 4 shows a simplified representation of the system architecture, in which the four modules are connected and managed by a controller that is also in charge of the communication with the Product Service database and the Resource repository. The database contains information regarding the product and service models, following the structure of the APS model. Currently, the company database contains only some of the product information and is organised following a model based solely on the service. One of the contributions of this industrial case is to extend the company database to the APS model so that it is organised according to a model that takes into account both products and services.

The repository stores different resources, such as the 3D model files, inspection result pictures, technical documents and additional information. The discussion management tool does not use the controller to access the discussion database but rather has direct access.

The physical architecture of the system is shown in Fig. 5. This architecture also presents the decisions adopted regarding the platforms and tools used to develop the system. The implemented system has been designed as a web-based client-server distributed architecture that allows the user to access the system through a web browser. The 3D viewer runs on Web Graphics Library (WebGL 2001).

The 3D viewer module has been developed using the Unity (Unity 2005) platform and integrated in the ASP project using WebGL. Originally, the 3D viewer module was built and integrated using the Unity plugin for web navigators (Unity 2015), whose operation was based on the Netscape Plug-in API (NPAPI). However, the majority of web browsers have now disabled support from this API (Google Chrome did so in its version 42, April 2015) because, according to web browser companies, it has become a leading cause of hangs, crashes, security incidents, and code complexity (Chromium Blog 2013).

The modules *inspection result viewer and model viewer* were implemented using ASP.NET (ASP 2002). The server side also contains a set of databases and repositories that form the Product Service database and Resources repository. These repositories store all the information and data that the application needs. Finally, the discussion management module is based on an existing tool named Redmine, an opensource project management web application written using Ruby on Rails (Ruby 2005). This module has its own database.

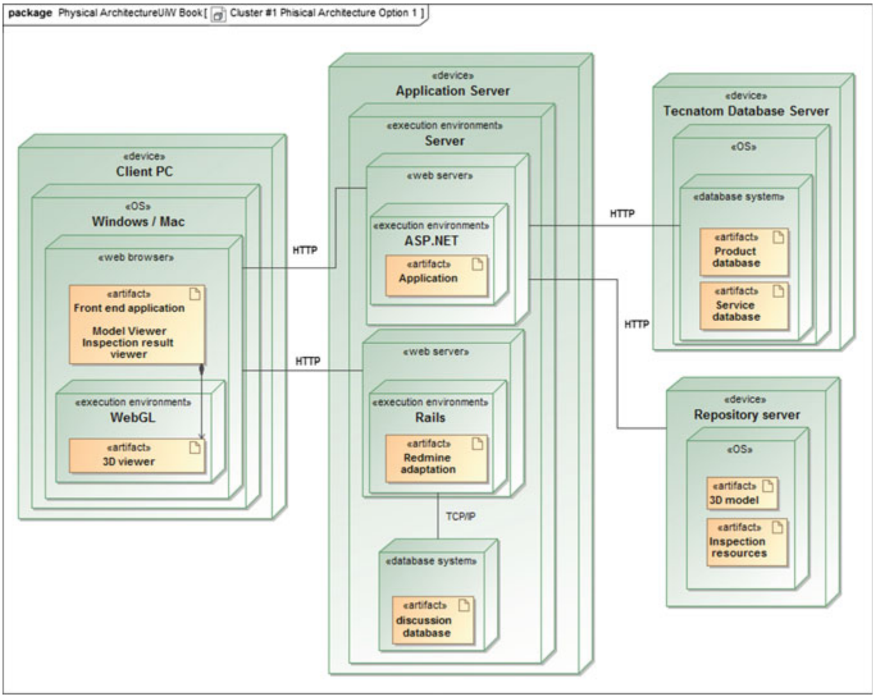


Fig. 5 Physical system architecture diagram

3 Contributions and Implementation, Virtual Reality in a Web Context

A layout of the implemented application, which is divided into four modules presented in the architecture (*model viewer, 3D viewer, interactive inspection result viewer and discussion management*), is shown in Figs. 6 and 7. Figure 6 presents the interactive views and Fig. 7 presents the discussion management tools, which will be describe hereafter. The implementation was carried out taking into account the company needs, the system requirements and the use cases. More details about the system can be found in Gonzalez-Toledo et al. (2015).

3.1 Model Viewer Module

In this industrial case, the product consists of a power plant turbine, whereas the service consists of the instructions for planning the inspections and the inspections carried out in the turbine. The model viewer allows actors to gain access to all the information stored in the system in an efficient and collaborative way.

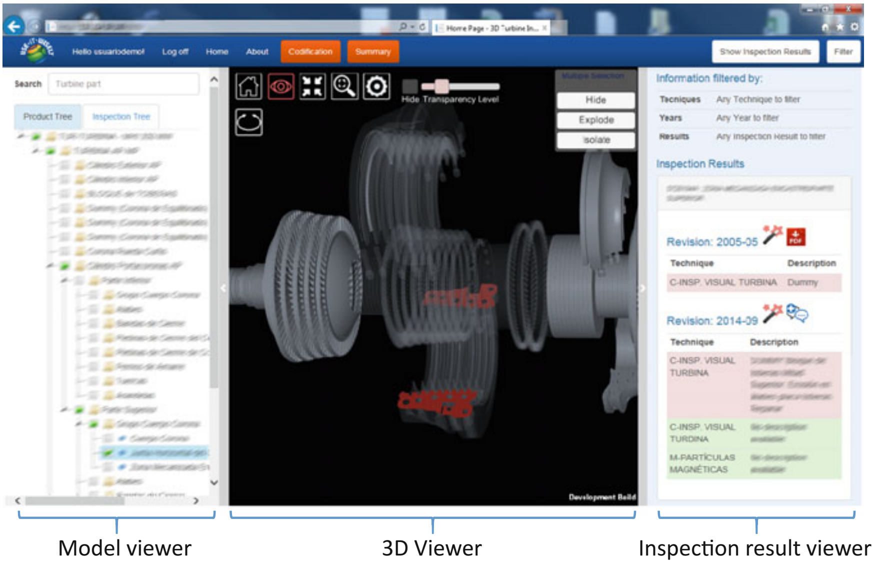


Fig. 6 Application user interface. *From left to right* model viewer, 3D viewer and inspection result viewer

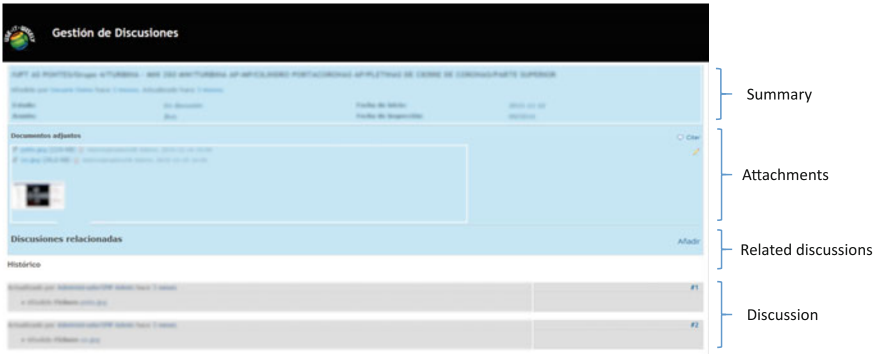


Fig. 7 Discussion management tool

The system has two types of potential users: the staff of the inspection and maintenance company and the staff of the power plant owner company. The first group might be more interested in navigating through the inspection tree model to conduct the inspection, whereas the latter might be more interested in navigating through the information from the point of view of the product.

When using the model viewer, users can choose which of the two navigation trees they want to use and can easily change between them by clicking on the corresponding tab. This allows the users to navigate through the product or inspection model interchangeably. The system is in charge of keeping the displayed information consistent and allows different users to use the application and collaborate so that they can analyse the product (turbine) state and make decisions together.

Once the user reaches the node of interest and selects the appropriate options in the tree, the system will show the requested information with the support of the other viewers.

3.2 3D and Inspection Result Interactive Viewer Modules

Both the 3D and inspection result viewers (Fig. 6) work together to show information about the product and the service. The 3D model of the turbine allows users to visualise and interact with the product model and understand information about the service, whereas the inspection result viewer allows users to investigate the inspection results in depth.

The module 3D viewer graphically represents the 3D geometry of the turbine model. In addition, the application allows users to interact with the product by navigating around the turbine 3D model. To help the user with the visualisation of hidden parts, mechanisms have been implemented, such as an advanced navigation system, identification and selection of different parts and occlusion management:

- *Navigation around the 3D model*. The user can navigate around the turbine model using the mouse and the keyboard to visualise the turbine from different points of view (Fig. 8). The user can select each part of the turbine with the mouse and access detailed information on the selected part.
- *Occlusion Management*. Because there are turbine parts that are occluded by others, the system provides mechanisms to make them visible. When the user is interested in a selected part, the viewer is able to provide a complete view of the part without losing its spatial relationship between the other parts of the turbine. Several mechanisms have been studied and classified in Elmqvist and Tsigas (2008) and Tominski et al. (2014) that could be employed to allow for this function, such as the cutaway views (Burns et al. 2008), 3D Magic Lens (Ropinski et al. 2004), transparency techniques (Burns 2011) and exploded views (Li et al. 2008). The last two techniques are both implemented within this system.

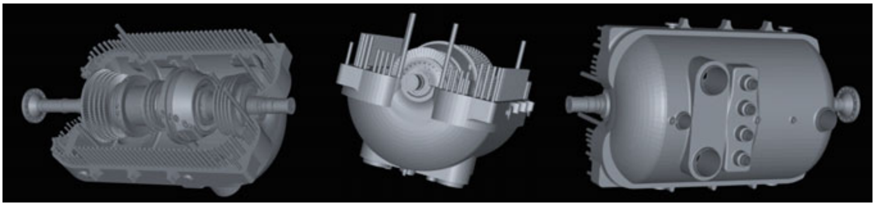


Fig. 8 The user navigates around the turbine, obtaining different points of view

The *adaptive transparency* mechanism makes each part that occludes the selected part transparent. In addition, when the user navigates around the turbine, the transparent objects change according to the user's point of view, as shown in Fig. 9.

The exploded view allows the user to discover and access the internal parts of the turbine (Fig. 10). The explosion refers to the simultaneous separation of parts in an explosive way and takes into account the assembly information stored in the product model.

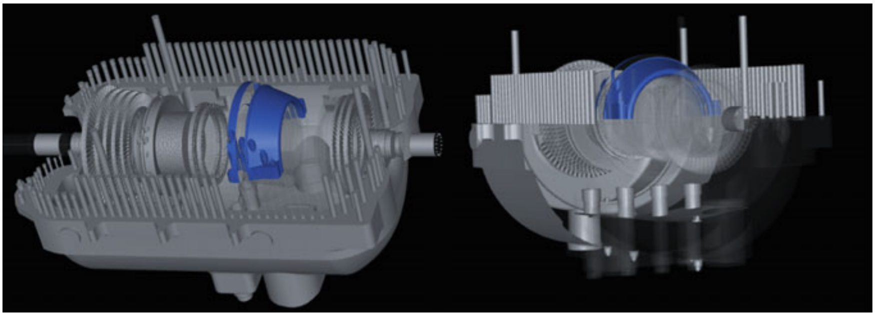
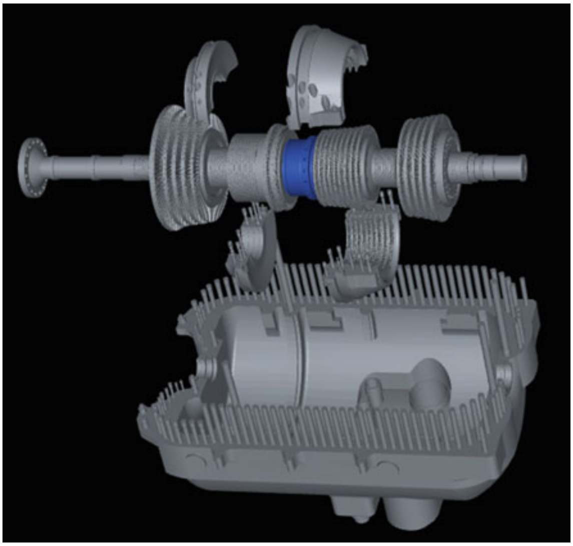


Fig. 9 Adaptive transparency view

Fig. 10 Exploded view



Regarding to the module inspection result viewer module, the inspection data can be shown in two ways:

- Together with the 3D model. Different mechanisms have been implemented to present the inspection result information in the 3D viewer. One of the mechanisms consists of overlaying the turbine graphical model with symbols and colours and relevant information regarding inspections, flaws and repairs. To achieve this, the application uses different symbols and colours that correspond to specific information, for example, whether the inspection result is acceptable (the turbine part does not need to be repaired) or not (the turbine part must be repaired). Another mechanism consists of pointing and tagging inspections results in a specific location of the turbine (over the 3D turbine geometry).
- Within the *Inspection Result Information panel*. The inspection results can also be presented in the Inspection Result Information Panel. This panel shows a summary of the inspection results carried out on a specific turbine part, for example, the technique used in the inspection, the sum of all flaws found at a selected part or their description. The shown information can be selected and filtered by the user. This panel also provides a set of links through the discussion management tool that connects the inspection results with the associated discussion.

3.3 Discussion Management Tool

The discussion management tool allows for analyse of the inspection results in a collaborative way by allowing the different users (power plant operators, inspection service engineers, power plants manager, etc.) to be involved in the final decision-making process, such as re-scheduling future inspections or scheduling maintenance activities to repair or replace an affected turbine part. This tool increases the amount of communication among actors, makes documentation easily accessible and enables the sharing of past experiences. Discussions that are associated to specific inspection results are accessible from the *Inspection Result Information panel*. Once the user has selected a discussion, the tool is opened (Fig. 7).

The discussion can be at two different states: open and close. If the discussion is open, the tool provides a set of controls for registering new contributions (text or attached files) and setting connections between different discussions. Different users will have access to different topics of discussion depending on their role. Once the users have a make a decision and conclude their discussion, it is closed and the system provides a report of the discussion and the decision.

4 Conclusions

The main aim of this chapter was to report on the development of a UIW-collaborative tool that assists in the collaborative management of inspection results. By decreasing the decision-making time and the amount of time taken for repair and maintenance procedures, the tool optimises the activities and increases the productivity of the power plants.

The UIW-methodology has made it possible to identify challenges that must be addressed in an effective way. The problem was modelled by use cases, requirements and a system architecture. In addition, an APS model was used to identify the structure of the relevant information that the system stores and manages to provide the needed functionalities.

As result of the work developed and the experience accumulated in this industrial case, it seems appropriate that the information systems of a company that has a problem such as the two hierarchical trees should be consistent with the APS presented. The APS model has been presented as an important output of this project; using a system that follows this structure would yield important benefits for the company.

Finally, this chapter demonstrates the implementation of a tool that offers a web-based application for the visualisation of the product and the data regarding inspection results, such as inspection data, techniques used and information about flaws found in a specific part of the turbine. This information is shown in two ways: (1) through a classic web app, that is, with hypertext, using plain text, tables, lists, photographs, 2D planes, etc. and (2) through a 3D module. The latter allows users to see information in a three-dimensional model of the turbine geometry and to navigate through the different parts of the turbine. The tool also includes a collaborative decision-making application to manage all stakeholders' proposals, annotations and discussions to assist in the decision-making process.

References

ASP. (2002). Microsoft ASP.NET. http://www.asp.net/. Accessed March 11, 2016

Bowman, D. A., Kruijff, E., LaViola Jr, J. J., & Poupyrev, I. (2004). *3D user interfaces: Theory and practice*. USA: Addison-Wesley.

Burns, M. S. (2011). *Efficient and comprehensible visualization of complex 3-D scenes* (Ph.D. thesis). Princeton University.

Burns, M., & Finkelstein, A. (2008). Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM Transactions on Graphics, 27*(5), 1.

Chromium Blog. (2013). *Saying goodbye to our old friend NPAPI*. http://blog.chromium.org/2013/09/saying-goodbye-to-our-old-friend-npapi.html. Accessed March 16, 2016

Elmqvist, N., & Tsigas, P. (2008). A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics, 14*(5), 1095–1109.

Friedenthal, S., Moore, A., & Steine, R. (2014). *A practical guide to SysML, the System modeling language*. Burlington: Morgan Kaufmann.

Gonzalez-Toledo, D., Cuevas-Rodriguez, M., Garre, C., Molina-Tanco, L., & Reyes-Lecuona, A. (2015). A tool for collaborative decision making on service information linked to 3D geometry of complex hierarchical products. In *Presented at EuroVR2015 Conference, Campus Politecnico Milano, Lecco (Italy)*, October 15–16, 2015.

Li, W., Agrawala, M., Curless, B., & Salesin, D. (2008). Automated generation of interactive 3D exploded view diagrams. In *ACM Transactions on Graphics (TOG)* (vol. 27, No. 3, p. 101). ACM.

Redmine. (2006). *Redmine*. http://www.redmine.org/. Accessed March 11, 2016

Rey, J. (2013). *Modeling with VSEE: Definition of guidelines and exploitation of the models*. YGT Final Report. Available via http://www.vsd-project.org/download/documents/YGT%20final%20report%20Rey%20V2.pdf. Accessed March 16, 2016.

Reyes-Lecuona, A., Molina-Tanco, L., González-Toledo, D., Flores, S., Frutos, E., Patel, H., et al. (2014). *Design, maintenance and refurbishment of turbines in a collaborative environment*. Presented at AHFE 2014, Kraków, Poland, July 19–23, 2014.

Ropinski, T., & Hinrichs, K. (2004). Real-time rendering of 3D magic lenses having arbitrary convex shapes.

Ruby. (2005). *Ruby on rails*. http://rubyonrails.org/. Accessed March 11, 2016.

Tominski, C., Gladisch, S., Kister, U., Dachselt, R., & Schumann, H. (2014). *A survey on interactive lenses in visualization*. EuroVis State-of-the-Art Reports (pp. 43–62).

Unity. (2005). *Unity technologies*. https://unity3d.com/. Accessed March 11, 2016.

Unity. (2015). *Web publishing following chrome NPAPI deprecation*. http://blogs.unity3d.com/2015/05/28/web-publishing-following-chrome-npapi-deprecation/. Accessed 16 March 2016.

WebGL. (2001). *Web graphics library*. https://www.khronos.org/webgl/. Accessed March 11, 2016.