Interface Specification Tool for Maritime and Oil & Gas

Statoil, Fred. Olsen Energy, Dolphin Drilling, Rolls-Royce, PanAsia, Hyundai Heavy Industries, IBM, and Genus

Report No.: 2014-1152, Rev. 1
Document No.: SICT-FINAL-TR
Date: 30 Dec. 2014
Project name: Joint Industry Project – System Test Strategies and Software Interface Specification
Report title: Interface Specification Tool for Maritime and Oil & Gas
Customer: Statoil, Fred. Olsen Energy, Dolphin Drilling, Rolls-Royce, PanAsia, Hyundai Heavy Industries, IBM, and Genus
Contact person: Jingyue Li
Date of issue: 30 Dec. 2014
Project No.: 913A0109
Organisation unit: DNVGL Strategic Research and Innovation
Report No.: 2014-1152, Rev. 1
Document No.: SICT-FINAL-TR
Applicable contract(s) governing the provision of this Report:

Objective:

Prepared by: Jingyue Li
Verified by: Torbjørn Skramstad
Approved by: [Name] [title]

Thierry Coq
Principle Consultant

Copyright © DNV GL 2014. All rights reserved. This publication or parts thereof may not be copied, reproduced or transmitted in any form, or by any means, whether digitally or otherwise without the prior written consent of DNV GL. DNV GL and the Horizon Graphic are trademarks of DNV GL AS. The content of this publication shall be kept confidential by the customer, unless otherwise agreed in writing. Reference to part of this publication which may lead to misinterpretation is prohibited.

DNV GL Distribution:
☒ Unrestricted distribution (internal and external)  ☐ Integrated Software Dependent System (ISDS)
☐ Unrestricted distribution within DNV GL  ☐ Interface management
☐ Limited distribution within DNV GL after 3 years  ☐ System engineering
☐ No distribution (confidential)  ☐ Maritime and Offshore
☐ Secret

Rev. No.  Date   Reason for Issue  Prepared by  Verified by  Approved by
1  2014-12-30  First issue  Jingyue Li  Torbjørn Skramstad  Thierry Coq
Table of contents

Contents

1 EXECUTIVE SUMMARY ............................................................................................................. 1
2 INTRODUCTION .......................................................................................................................... 2
3 INTERFACE SPECIFICATION TOOLS ...................................................................................... 6
  3.1 Detailed tool requirements ........................................................................................................ 6
  3.2 IBM’s own descriptions of its tool
      3.2.1 Features and benefits of IBM tool ..................................................................................... 9
      3.2.2 Reference price model ........................................................................................................ 12
  3.3 Genus’ own descriptions of Genus Nexus
      3.3.1 Features and benefits of Genus Nexus ................................................................................. 13
      3.3.2 Reference price model ........................................................................................................ 15
  3.4 Descriptions of Smartsheet based on information of its website
      3.4.1 Features and benefits of Smartsheet .................................................................................... 15
      3.4.2 Reference price model ........................................................................................................ 16
  3.5 Comparison of tools from DNVGL viewpoints
      3.5.1 The IBM tool and Genus Nexus compared to Smartsheet .................................................. 19
      3.5.2 Genus Nexus and the IBM tool compared .......................................................................... 21
  3.6 Discussions
      3.6.1 How to apply the tools into Maritime and Oil & Gas businesses? ....................................... 22
      3.6.2 Way forward ......................................................................................................................... 24
4 CONCLUSIONS ............................................................................................................................ 26
5 ACKNOWLEDGEMENT .................................................................................................................. 27
6 REFERENCES .................................................................................................................................... 28
7 APPENDIX A. DETAILED FUNCTIONS OF IBM TOOL ............................................................... 29
8 APPENDIX B. DETAILED FUNCTIONS OF GENUS NEXUS .................................................... 34
9 APPENDIX C. DETAILED PRICE MODELS .................................................................................. 45
1 EXECUTIVE SUMMARY

For advanced ships and Mobile Offshore Units (MOUs), several systems from many different suppliers are integrated. To increase safety while reducing downtime and commissioning delays, it is important that integrator (yard) and suppliers specify and follow up interfaces more efficiently. This will enable suppliers to build and configure their systems consistently and thereby ease integration of systems from the different suppliers. Currently, suppliers and integrator (typically a shipyard) specify and exchange interface information mainly through Excel or Pdf files. Such practices make it is difficult to exchange and coordinate interface information. The inconsistency of interfaces can cause project delay and control system quality problems.

This report summarizes results of the "interface specification tool work package" of the Joint-Industry-Project (JIP) “System test strategies and Software interface specification tool”. The JIP has partners representing different roles of building and operating MOUs, including Statoil (Operator), Fred. Olsen Energy and Dolphin Drilling (Owner), Hyundai Heavy Industry (integrator), Rolls-Royce (supplier), DNVGL (Class society), PanAsia (supplier), IBM (software tool provider), and Genus (software tool provider). The main task of the “interface specification tool work package” was to develop a web-based interface management tool to help operator, owner, suppliers, class society, and integrator exchange and coordinate interface information in new-building. The tool is also expected to be feasible to be used in operation, if interface information is transferred to owners and operators after new-building.

In this JIP, first, we communicated with MOU operator, owner, suppliers, integrator, third party testers, and class society to collect their requirements on the interface specification tool. Then, we made an interface management prototype to clarify the requirements further. After that, we worked together with two software companies, i.e. IBM and Genus, and helped them develop their interface specification tools for demonstration. The tools from IBM and Genus were then demonstrated to partners of the JIP. The demonstration was also recorded and sent to partners for replaying and for giving comments.

In this report, first, we summarize expected functions of the tools. We then include IBM’s and Genus’ own descriptions of the high-level functions, benefits, and price models of their tools respectively. We then present comparisons of the JIP related functions of the IBM and Genus tools from DNVGL’s viewpoint through our evaluation in the JIP. In addition, we compare IBM and Genus tools with a tool (i.e. Smartsheet) introduced by a MOU owner. We conclude that IBM and Genus have implemented several useful functions in their tools to fit the purpose of managing interface information in Maritime and Oil & Gas industries. We also conclude that IBM and Genus tools can outperform Smartsheet if the tools are to be used in large engineering projects. Finally, we propose and discuss how to integrate IBM and Genus tools into new building and operation practices of Maritime and Oil & Gas industries.

As tools from IBM and Genus have been developed from this JIP, the possible next step is to pilot and evaluate the tools in JIP partners’ engineering projects, in order to collect more feedbacks and to improve the tools further.
2 INTRODUCTION

To our knowledge, most interface specifications in maritime and O&G industries are typed in using Excel, Word, or Pdf files, which makes it difficult to check the completeness and consistency of interface specification, to analyse system dependencies, to support coordination of interface specification, and to synchronize changes. To improve the interface specification and analysis practice, we decided to develop a tool using a web-based GUI (Graphic User Interface) with DB support to help suppliers and/or system integrators fill in interface specification in an easier and more uniformed way.

A study to standardize the interface specification, and the development of a web-based tool to formalize inputting and analysing interface information, will benefit the yard, the suppliers and the owner/operator (end user). Interface specifications and their dependency information in the tool can also be used to support change impact analyses.

The expected benefits of having such a software interface specification tool are:

- For Owners and Operators, the tool will support safe and efficient operation of advanced ships and offshore units, especially when managing changes and upgrades.
- For Integrators, the tool will help integrators analyse the system dependencies during unit design and system integration.
- For suppliers, the tool can help the interface definition during the design, interface testing, and change impact analysis.
- For DNVGL, the tool will help DNVGL to provide a more cost-effective service to safeguard system and software integrity throughout the systems’ lifetime.

To implement such an interface specification tool, DNVGL together with HHI (Hyundai Heavy Industries) have initialized a JIP (joint-industry-project) to study the tool concept and to develop a tool. Besides DNVGL and HHI, the JIP also includes operator, suppliers, and software tool providers. The JIP was initially planned to last from Q4 of 2013 until Q2 of 2014. Due to contractual issues at the beginning of the JIP, the start of the project was delayed. The interface specification tool was finished in Q4 of 2014.

In the JIP project proposal, we planned (as specified in the contract) to develop the tool in three Phases.

- Phase A is to investigate interface specification provided by different suppliers in current new building projects. We study similarities, differences, and possibly inter-connections of interface specifications from different suppliers.
- Phase B is to collect software requirements of the tools, and to develop tool prototypes.
- Phase C is to execute an extended project with the purpose of building full-scale tools for commercial use, if the prototype tools are reviewed by the partners as useful.

Only Phase A and Phase B are under the financial framework of the JIP presented in this report. For functions proposed in Phase C, we initially planned to apply for a larger and longer-term (2-3 years) follow-up JIP (probably also apply for funding from the Norwegian Research Council) to implement them.

The proposed functional and non-functional requirements of Phase B and Phase C are shown in Figure 1 and Figure 2. The implementation statuses of the requirements are in Table 1. Detailed explanations of the requirements are in Table 2.
All tasks specified in Phase A and Phase B have been implemented.

In phase A, to collect detailed software requirements of the interface specification tool, we had interviews or meetings with partners of the JIP, who were possible end users of the tool, to understand their requirements. We decided to use an iterative process. This was because it was difficult for partners to describe the software requirements of the tool using free text. In addition, different partners might have different understandings and opinions on the functions to be included in the tool. We therefore performed several iterations.
To facilitate the discussion of software requirements, instead of writing the software requirement using formal specification (e.g. UML), we decided to make the Graphic User Interface (GUI) mock-ups of the interface specification tool to show the high level architecture and functions to be included in the tool. We started with a very preliminary version of the mock-ups, and then sent the mock-ups to all partners for comments. Then we revised the mock-ups based on comments and sent it again for further comments. After 3-4 rounds iterations, we believed that we had understood the partners’ requirements for the tool.

In phase B, we have implemented all functional and non-functional requirements of the tool specified in the contract with IBM and Genus. We have also implemented some additional functional and non-functional requirements specified for Iterations 1, 3, and 5 of Phase C, although Phase C is supposed to be financed in a follow-up project.

### Table 1. Implementation statuses of the requirements specified in contract

<table>
<thead>
<tr>
<th>ID of high level requirements</th>
<th>Name of the high-level requirements</th>
<th>Iterations to be implemented</th>
<th>Implementation status</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>Function tree</td>
<td>Iteration 1</td>
<td>Implemented</td>
</tr>
<tr>
<td>B2</td>
<td>System tree</td>
<td>Iteration 1</td>
<td>Implemented</td>
</tr>
<tr>
<td>B3</td>
<td>Function to system allocation matrix</td>
<td>Iteration 1</td>
<td>Implemented</td>
</tr>
<tr>
<td>B4</td>
<td>Integrator view (interface matrices, interface to function and system allocation)</td>
<td>Iteration 2</td>
<td>Implemented</td>
</tr>
<tr>
<td>B5</td>
<td>Supplier view (signal management, interface to signal trace)</td>
<td>Iteration 2</td>
<td>Implemented</td>
</tr>
<tr>
<td>B6</td>
<td>Support multi-user coordination</td>
<td>Iteration 1 and 2</td>
<td>Implemented</td>
</tr>
<tr>
<td>B7</td>
<td>Visibility control (security)</td>
<td>Iteration 1 and 2</td>
<td>Implemented</td>
</tr>
<tr>
<td>B8</td>
<td>Version control</td>
<td>Iteration 1 and 2</td>
<td>Implemented</td>
</tr>
<tr>
<td>C1</td>
<td>Integrate with interface management sys. of integrator and suppliers</td>
<td>Iteration 1</td>
<td>Not implemented</td>
</tr>
<tr>
<td>C2</td>
<td>Support reuse (totally and partially) of previous project interface spec. of integrator and suppliers</td>
<td>Iteration 1</td>
<td>Implemented.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible for suppliers and integrators to import information from previous Excel files.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible for suppliers and integrators to check “difference” between different versions of information.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>Make an agreed and standard interface or signal template</td>
<td>Iteration 2</td>
<td>Not implemented, because not enough suppliers are involved in the JIP.</td>
</tr>
<tr>
<td>C4</td>
<td>Support interface and signal completeness and consistency checking</td>
<td>Iteration 3</td>
<td>Implemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible to add rule to define and check completeness and consistency of interface and signal information.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td>Support description of alternative interface and signal</td>
<td>Iteration 4</td>
<td>Not implemented</td>
</tr>
<tr>
<td>----</td>
<td>--------------------------------------------------------</td>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>C6</td>
<td>More advanced functionalities, e.g.:</td>
<td>Iteration 5</td>
<td>Implemented functions:</td>
</tr>
<tr>
<td></td>
<td>Visualize dependency</td>
<td></td>
<td>Traceability from operational scenario to interface and to test cases</td>
</tr>
<tr>
<td></td>
<td>Traceability from operational scenario to interface and to test cases</td>
<td></td>
<td>Not implemented functions:</td>
</tr>
<tr>
<td></td>
<td>Functions to be able to calculate end-to-end RAMS (Reliability, Availability, Maintainability, and Safety)</td>
<td></td>
<td>Functions to be able to calculate end-to-end RAMS</td>
</tr>
</tbody>
</table>
3 INTRFACE SPECIFICATION TOOLS

In Chapter 3.1, we explain the detailed tool requirements we have collected through interviews and meetings with JIP partners. In Chapter 3.2 and 3.3, we present IBM’s and Genus’ own description of high-level features, benefits, and price models of their tools respectively. In Chapter 3.4, the high-level features and benefits of Smartsheet are briefly described based on information we collect from the tool website by the end of 2014. In Chapter 3.5, we describe the comparisons of the JIP relevant features of the tools from DNVGL’s viewpoint. At the end, we discuss how to integrate IBM and Genus tools into daily business of the different actors in the Maritime and Oil & Gas industries.

3.1 Detailed tool requirements

To collect high-level requirements of the tool, we had interviews or meetings with partners of the JIP (who are potential end users of the tool) to understand their requirements.

- HHI twice (all face-to-face)
- Dolphin Drilling (email)
- PanAsia twice (telephone and face-to-face)
- Rolls Royce twice (telephone and face-to-face)
- Statoil three times (one face-to-face and two video meetings)

The interview results show that the tools need to include the following high-level features:

- Access control
- Interface data management
- Test management
- Traceability and change management
- Technical query management

The requirements we collected are in line with high-level requirements defined in the contract, and are shown in Table 1. In Table 2, we show detailed descriptions of the requirements. DNVGL has introduced and implemented ISDS class notation [1] in Oil & Gas industries to improve the engineering practices. Interface information is one part of the information that actors in this industry must manage well, in order to be compliant with ISDS. Some of the detailed requirements of the tools are therefore closely linked to ISDS class notation concepts and practices. In Table 2, we also show the links between these detailed requirements and the high-level requirements specified in Table 1.

After interviews and meetings with integrator, suppliers, and owners, we had more than 10 face-to-face meetings with IBM and Genus to discuss the customers’ requirements, to translate the customers’ high-level requirement to software requirements, and to guide them to develop and customize the tool.
Table 2. Detailed requirements lists

<table>
<thead>
<tr>
<th>Detailed req. name</th>
<th>Description of detailed req.</th>
<th>Trace to ID of high level req. in Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional requirements:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational scenario and use cases</td>
<td>In ISDS class notation of DNVGL [1], high-level software requirements are formatted as operational scenarios. Low-level ISDS software requirements are formatted as use cases. The operational scenario and use cases are following similar templates. The tool should be able to store and manage operational scenario and use case info.</td>
<td>B1</td>
</tr>
<tr>
<td>System and sub-system lists</td>
<td>ISDS system list is a sheet including several columns: • System type (ISDS/external) • System ID • System Name • Sub-System ID • Sub-system Name • System/sub-system vendor • System function index (SFI) The tool should be able to store and manage system and sub-system info.</td>
<td>B2</td>
</tr>
<tr>
<td>Function-system allocation matrix</td>
<td>The rows of function-system allocation matrix are lists of ISDS functions with System Function Index (SFI). The first column of the matrix is the ID of the drawings of the SFI. The rest of the columns of the matrix are lists of names of the ISDS systems. The data (X/empty) in the cell indicates whether the function is relevant to a specific ISDS system. The tool should be able to store and manage the function-system allocation matrix info.</td>
<td>B3</td>
</tr>
<tr>
<td>Interface matrix</td>
<td>Both rows and columns of the interface matrix are lists of ISDS systems. The data in the cell indicate how the systems interact with each other: • X: External (Direct) Interface • I: Interface internal to one subsystem, described by vendor in vendor design • -: Not within scope • E: End-to-end interface (not direct) • V: interface between subsystems of same vendor, to be described by vendor • 0: No interface identified The verification status of the interface information is indicated using different colours: • Green: interface info. verified • Yellow: interface info. implemented, not verified • Red: interface info proposed. The tool should be able to store and manage the interface matrix info.</td>
<td>B4</td>
</tr>
</tbody>
</table>
### Visualize Interface Interactions
The interface matrix data indicate interactions between systems. The interactions of the systems can be visualized in a graphic format.

### Interface Register
The interface register data is an ISDS data sheet that includes high-level interface description. The sheet includes several columns, such as:
- Interface ID
- Name of the function
- Source system
- Destination system
- Protocol
- Data type
- Validity domain
- Etc.

The tool should be able to store and manage the interface register info.

### Signals
For each row in the interface register, there will be a table showing the signals of the interface in detail.

The detailed signals are often in Excel format and includes several columns, such as:
- ID
- Function Number
- Location
- Etc.

Each supplier may have his/her own template of the detailed signals.

The tool should be able to store and manage the signal info and link the signal info back to interface register.

### Coordinate TQ Processing
Users can fill in Technical Query (TQ), discuss TQ, and track the status of TQ.

### Alert and Reminder
When TQ is created / changed or interface information is updated, the related users should be notified by email. Administers can also set up reminder to let users remember to submit.

### TQ Status Reporting
Some users need to have overviews of the status of TQ.

### Record History
The tool should be able to record the changes made on the interface information, and record when and who have made the changes.

### Configuration Management
The tool should support managing versions/releases, compare differences between versions/releases of artefacts.

### Lifecycle Management
The tool should support engineering process and lifecycle management, such as submission, review, revision, agree, and approve upon individual signals and/or complete interfaces.

### Validate Completeness
Although different companies and different interfaces may have different columns to list the interface details, it is necessary to ensure all mandatory information is filled in.

The tool should be able to automatically validate the completeness of the input information to ensure that the mandatory fields are filled in.

### Validate Consistency
In some cases, the interface information needs to be complied with certain rules. For example, some value cannot be more than a certain limit.

The tool should be able to validate the consistency of the value with respect to the rules.

### Traceability
The interface signals need to be traced from requirements (Operational scenario), to systems, sub-systems, interface register, and tests.

### Test Management
The tool should support managing interface test information.
Non-functional requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Document(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability: Possible to manage large number of interfaces and signals</td>
<td>The tools should support storing and manipulating more than several thousands (or more) interfaces and signals.</td>
<td>B4, B5, B6</td>
</tr>
<tr>
<td>Performance: support multi-user access in parallel</td>
<td>The tool should support 20-30 users to access data concurrently with decent response time.</td>
<td>B6</td>
</tr>
<tr>
<td>Usability: Easy and intuitive GUI</td>
<td>The GUI of the tool should be either in Web or Desktop. The GUI should be easy for users to understand and use</td>
<td>B6</td>
</tr>
<tr>
<td>Usability: Easy to learn and adopt</td>
<td>The tool should not introduce too many changes to the current practice of the users. The tool should still allow people to use the current engineering tools, such as Excel, Word, Pdf, and AutoCAD.</td>
<td>B6</td>
</tr>
<tr>
<td>Usability: Easy to change template</td>
<td>If necessary, suppliers may revise their signal template in the future. It should be easy to implement such changes.</td>
<td>B6</td>
</tr>
<tr>
<td>Security: Authorization</td>
<td>Super users should be able to set up access control of various artefacts. It is critical that one supplier should not be able to see the information of another supplier without authorization. The information includes TQ and interface details.</td>
<td>B7</td>
</tr>
<tr>
<td>Security: Authorization</td>
<td>Users need to register and log in to get access to the system</td>
<td>B7</td>
</tr>
<tr>
<td>Security: Data storage and privacy</td>
<td>The data need to be stored at the server of owner or integrator to protect the data to be accessed by people irrelevant to the project.</td>
<td>B7</td>
</tr>
<tr>
<td>Reusability: Possible to reuse info. from previous projects</td>
<td>For all the information, e.g. operation scenario, system lists, interface register, and signals, that need to be input to the systems, it should be possible to let people import data from Excel file, which may be reused from previous projects, given the Excel follows a template.</td>
<td>C2</td>
</tr>
</tbody>
</table>

3.2 IBM’s own descriptions of its tool

3.2.1 Features and benefits of IBM tool

The IBM tool implemented in this JIP is based on the IBM Rational solution for SSE. IBM Rational solution for SSE provides collaborative, integrated systems engineering and embedded software development capabilities on an open platform, as shown in Figure 3. It supports system engineering approaches to unify software engineering and other disciplines such as electrical and mechanical engineering. The IBM Rational solution enables stakeholders and contributors to collaborate closely and effectively. The main advantages of IBM Rational solution are:

- Full asset lifecycle management support through various tools and process templates. It includes all major lifecycle activities: requirements management, architecture & interface design, development, collaboration and change management, configuration management, and quality management. It also provides traceability and change management from proposal and impact analysis to decision sign-off and execution.

- Smoothly integrated multiple tools that can help bring together cross-discipline engineering artefacts and their relationships into views for analysis and decision optimization. It provides near real-time cross-lifecycle views of related artefacts and relationships.
• Web-based sharing (view, mark-up) of designs for cross-discipline and cross-team collaboration.

The IBM tool implemented in this JIP is an integrated tool set to support all lifecycle activities. IBM DOORS is integrated for requirements management. IBM Rational Rhapsody and IBM Design Manager are integrated for architecture & interface design. IBM Rational Team Concert is integrated for collaboration and change management, and configuration management. IBM Quality Manager is integrated for test management.

Figure 3. Rational Solution for Systems and Software Engineering: Best practices, tools and services on an open platform

3.2.1.1 Full asset lifecycle management and traceability

IBM Rational solution for SSE offers full support for lifecycle traceability and change management. The lifecycle traceability, as shown in Figure A.1 of Appendix A, establishes relationships between various software artefacts, e.g. requirements, design, interface, signals, and test cases, and provides views for insightful decision-making. Queries and views can be defined to view the traceability over the entire design and tool-chain. It is also possible to work at different levels (i.e. real signals and "logical signals") and show traceability between the different levels. It is possible to visualize and analyse the signal flow. This allows the designers to focus on just the signals of interest.

Through full traceability between artefacts, the engineering team can answer questions about project status. Lifecycle traceability improves quality by:

• Establishing relationships between software artefacts raising awareness between team members
• Helping project managers identify and close artefact gaps across disciplines
• Providing practitioners access to related artefacts so that they can make fully informed decisions
• Providing a clear view of completeness from requirement through release

3.2.1.2 Configurable change and coordination management

In IBM Rational solution for SSE, one can use configurations to group a specific set of versioned artefacts. Every artefact in the solution keeps track of changes through its history. Figure A.2 of Appendix A shows change management of a design artefact. Configurations commonly identify one version, or reference-able state, of each artefact in the set. A team, for example, might group all the requirements for a specific component into a configuration making it easier to identify and work with all the requirements associated with that component.

The main feature of change management is work items, which track and coordinate tasks, including change requests, defects, plan items, and ordinary tasks. Work items and the workflow process they follow can be customized to suit the project.

A Technical Query (TQ) is an example of a work item type. Technical Queries are created and shared by relevant stakeholders. An example of Technical Query is shown in Figure A.3 in Appendix A. TQs are captured and organized on dashboards where stakeholders collect their own relevant TQs for visibility and follow up. TQs are highly configurable and allow for collaboration between suppliers, integrators and operators & owners. An example of dashboard for capturing and acting on TQs are shown in Figure A.4 in Appendix A. Subscribers of TQs get email notifications whenever updates take place. Workflow management is part of each individual TQ, for example Propose, Approve, or Reject.

3.2.1.3 Model-based interface data management

The interface data will mainly be managed by IBM Rational Rhapsody and Rational Design Manager. IBM Rational Rhapsody is a model-based system engineering tool. Using custom scripts of Rational Rhapsody to some extent, it will be possible to read and validate interface matrix, interface register data, and signals from Excel files. A custom input validator script or model checker can be created to detect missing data and verify that all required fields are filled in. In addition, input validators can be created to detect inconsistency data and verify that all fields are correctly filled in.

Interfaces can be viewed in an Excel like table format or in a diagram. With a custom script, it will be possible to output data to an Excel file. The interfaces need to follow a template (profile) so that the suppliers can enter the data in the same format. Templates can be created for each interface type with each type having its own information fields and colour as desired, as shown in Figure A.5 in Appendix A.

System interfaces can be viewed graphically as to how the systems are interacting with each other, as shown in Figure A.6 in Appendix A. The drawings follow the SysML and UML standards. Multiple drawing diagrams can be created for the same interface with the contents filtered on specific signals of interest. For instance, one drawing can be created to only show signals related to test. Colours can be used to show the type of interface as well as the verification status of the interface. It is also possible to have drawings that focus on specific signals for a specific purpose. For example, sequence diagrams that show the interaction between a set of signals for a specific purpose, i.e. only the signals that are involved in a system start-up (as shown in Figure A.7 in Appendix A) can be created. This may make it easier to discover "race conditions" or block conditions earlier in the design process.

Although it is possible to read interface information from Excel files and export the information to Excel files, entering the interface information directly in the IBM tool, i.e. IBM Rational Rhapsody, may provide better functionality and quality.
3.2.1.4  Role-based access control

Interface details are under change- and configuration control and can be made visible and shared with integrators or other suppliers on a need to know basis. In order to access project data, one has to login and be authenticated by the tool. Access to artefacts can be set on project level where suppliers work in different project areas, or in some cases on a lower level for example for TQs, where access groups can be created to segregate suppliers from each other and provide visibility for only selected stakeholders. Server administrators and project administrators can control access to project area artefacts through the use of client access licenses (CALs) [2], repository group permissions, team areas, roles and role-based permissions, and access control settings.

3.2.1.5  Requirement-driven quality management

Quality Management integrates with the requirement management tool, and supports requirement driven testing. By linking test plan to a collection of requirements, it will streamline the process of keeping test cases in sync with requirements and design artefacts, as they evolve, as shown in Figure A.8 and Figure A.9 in Appendix A.

It is possible to keep track of the test progression with real time reporting and dashboards and to gather information about who has done what on validation and verification. All testers involved in running a test are displayed in table views. Test results show the workflow states of the referenced test case, test script, and where applicable, test plan and test suite at the start of the test execution. It is also possible to automatically create test cases based on requirements, to ensure complete requirement test coverage. The test automation strategy is also flexible that facilities reuse of test artefacts.

3.2.2  Reference price model

IBM has chosen to present three potential solutions that clients could use as a part of their service offering, as show in Figure 4:

- IBM cloud – IBM Softlayer
- Clients private cloud
- Purchase perpetual licenses for on-premise installation

The reference pricing for key components of the solution for a monthly basis with a deployment cost for supporting a team of 15 - 20 persons is proposed in Table C.1 of Appendix C (As the number can be confidential, the number will be excluded for external distribution of this report. However, the number can be provided on mutual agreement between IBM and clients).

![Figure 4. IBM proposed commercial solutions for corporations](image-url)
3.3 Genus’ own descriptions of Genus Nexus

3.3.1 Features and benefits of Genus Nexus

Genus® Nexus is made using model-driven development in Genus Application Framework®. The model-driven approach delivers flexible, scalable and highly usable applications much faster than traditionally programmed systems and much more tailored to customer’s needs than off-the-shelf products.

Genus Nexus is a working application that demonstrates how to meet the requirements in this JIP project. The functionality is split into a desktop system for the owner and yard, and a web portal for the suppliers in the following way:

Suppliers (Web Portal):

- Submittal of signal lists
- Receive feedback on submitted signals
- Send and receive TQs and answers
- Receive input and procedures for interface testing (released interface specifications)
- Secure login with access only to relevant information sanctioned by the owner/yard

Owner and Yard (Desktop):

- Create and maintain interface specifications
- Review submitted signals from suppliers
- Handle signal changes and revisions
- Create, answer, forward, and close TQs
- Create and maintain operational scenarios
- Release official signal revisions for interface testing
- Create and maintain interface test procedures and results
- Handle system, supplier and vessel data
- Manage user access rights

All functions are fully implemented and running in a demonstration application using data from a fictional vessel’s new build project. Elaboration on the different functions and examples, and screen snapshots from Genus Nexus are shown in Figures B.1 to B.15 in Appendix B.

Building complex offshore vessels is very expensive. It is important that new tools and processes do not add to the total cost, but are able to present a positive return on investment from the first project. Based on first-hand experience from the oil and gas industry in general, and work on the ISDS Class notation [1] specifically, Genus emphasises the following benefits as key to success.

3.3.1.1 Implementing the needed functionality, and nothing else

All parties in the industry have existing tools, processes and standards that support consistency and quality across the different disciplines and projects. This JIP has identified some areas where tool support could improve. Genus proposes to focus on these areas as a starting point and avoid the ambition of implementing a large tool or suite of tools that cover much more than needed. By orchestrating thoroughly tested building blocks in Genus Application Framework, Genus is able to
implement tailor made applications much faster than traditionally programmed systems and much better customized than off-the-shelf products. The flexibility of model-driven development also ensures that Genus Nexus rapidly evolves with the changing needs in a project.

Genus acknowledges that a software tool must be based on the industry’s current concepts and ways of working, and not introduce significant changes in engineering practices. A tool must adapt to and enhance the business processes, not the other way around. A successful software tool provides value from day one. Integrating to existing production and documentation systems such as eRoom, RigDoc, SmartPlant and DNVGL NPS gives a better return on investment than trying to replace them, and is a key strategy when implementing Genus Nexus in a real project.

3.3.1.2 Minimizing extra effort from the suppliers

In a new build project, any variation orders and extra requirements that are passed on to the suppliers are very expensive. In fact, changes in requirements are among the main drivers of delays and budget overruns in these projects. A successful Interface Specification Tool must introduce as little extra effort from the suppliers as possible. Any additional costs will come back to the yard with a multiplied price tag.

Genus Nexus minimizes this by:

- Allowing the suppliers to continue using the existing formats for their signal lists. The idea of having standard signal and interface formats for all suppliers is very good, but not feasible to implement from the first project. Genus Nexus adapts to the existing formats.

- Introducing no tool installation or changes in the existing supplier engineering processes. The suppliers know best how to manufacture their systems and should be allowed to continue doing that in the way they see appropriate.

- Having the suppliers pay no license fee for using the tool. The supplier web portal is a service provided as part of the yard/owner license.

Should the suppliers’ needs change in a way that the suppliers need more tool support, i.e. when an interface originates at a supplier and the yard submits the signals, Genus Nexus is highly flexible and adaptable.

3.3.1.3 Protecting intellectual property

Genus Nexus comes with a solid built-in security model tested and used by demanding customers, such as The Norwegian Police and The Norwegian Ministry of Justice and Public Security. All access to data and objects passes through the security model built on these concepts:

- No data from one supplier is accessible by the other suppliers except when needed, i.e. for interface testing and relevant technical queries.

- The yard fully controls the access control and authorisation.

- The yard controls what is passed on to the owner when delivering the vessel, enabling use of the tool for sensitive data that should never leave the yard.

- Single-sign-on is possible through existing user account handling at the yard/owner with inherited group access configurations (i.e. LDAP or Active Directory).

- The yard or a partner trusted by the yard does the hosting. No cloud service is necessarily to be involved.
3.3.1.4 Supporting software processes

A software tool is not a guarantee for compliance to software process standards such as ISDS [1], but a tool may support and encourage certain elements from these standards.

Genus Nexus has built-in mechanisms to:

- Facilitate review processes
- Track changes and make sure verification is performed on common revisions (releases)
- Trace verification activities to requirements and operational scenarios
- Implement rules and triggers based on the project’s verification and validation strategy

There are possibilities in Genus Nexus to implement rules and triggers based on process standards and quality plans. Genus Nexus has also flexibilities to adapt to changes in requirements in a fast manner.

3.3.2 Reference price model

The Genus Nexus price model is based on two project phases.

- **Phase 1**: Software delivery and setup during a vessel’s engineering, construction and acceptance stages, normally constituting a period of up to 4 years.

- **Phase 2**: Software maintenance during a vessel’s operations stage. A period of up to 20 years or more.

Detailed numbers of the proposed reference prices are shown in Table C.2 of Appendix C (As the number can be confidential, the number will be excluded for external distribution of this report. However, the number can be provided on mutual agreement between Genus and clients).

3.4 Descriptions of Smartsheet based on information of its website

3.4.1 Features and benefits of Smartsheet

Smartsheet is a generic cooperation tool based on Excel sheet and cloud computing. Here, we will present only high level features of Smartsheet based on information we collect from its website by the end of 2014. We do not present details of the features of this tool, because the detailed descriptions can be updated after this report. Readers can go to Smartsheet’s websites (http://www.smartsheet.com/) to get more detailed information of Smartsheet.

The coordination features of Smartsheet include:

- File sharing,
- Alert & reminders
- Gantt charts
- Calendars
- Google App,
- Mobile,
- Web form,
Cell linking, Reporting, Resource management.

Smartsheet is applicable to various management and coordination based on Excel files, such as project management, sales management, or marketing management. The website front page states the benefits of Smartsheet are:

- **Spreadsheet Easy**
  Instantly familiar, so everyone on your team "gets it" and hits the ground running. With one centralized, always-available tool, you’ll work smarter, not harder.

- **Works Your Way**
  Flexible to manage any kind of work — from simple task lists to complex processes. Attach files, set alerts, automate workflows, view Gantt charts and more.

- **Scales Quickly**
  Smartsheet delivers the security, integrations, and controls that exceed corporate requirements. User embraced, IT approved.

### 3.4.2 Reference price model

Smartsheet has three levels of price models, i.e. basic, advanced, team, and Enterprise. The features included in each price model vary. The price models of basic, advanced, and team are fixed price per month. However, the Enterprise price is not listed in the website and is up to negotiation. The detailed price of each price model by the end of 2014 is listed in Figure C.2 of Appendix C.

### 3.5 Comparison of tools from DNVGL viewpoints

In this section, we compare the features of the IBM tool, Genus Nexus, and Smartsheet against the detailed requirements lists in Table 2. The comparison results come from DNVGL researchers’ trial of the IBM tool, Genus Nexus, and Smartsheet in the JIP. As few JIP partners have piloted the tools in their engineering projects, their feedbacks on the tools have not been collected, and therefore are not presented here.

The comparison results are shown in Table 3. Although we have presented reference price models of the tools in Section 3.2, 3.3, and 3.4, we do not compare price in this report, because detailed pricing numbers are up to negotiation.
**Table 3. Comparison of tool features**

<table>
<thead>
<tr>
<th>Detailed req. name</th>
<th>IBM</th>
<th>Genus Nexus</th>
<th>Smartsheet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operational scenario and use cases</strong></td>
<td>Support Table, Web Form, and Word input format. Possible to compare and merge Word files.</td>
<td>Support Table and Web Form input format. Possible to export to Word, Excel and Pdf for reporting and signing.</td>
<td>Limited to Table input format.</td>
</tr>
<tr>
<td><strong>System and sub-system lists</strong></td>
<td>Support inputting and listing systems and sub-systems.</td>
<td>Support inputting and listing systems and sub-systems.</td>
<td>Possible to make a system and sub-system template to support inputting and listing such information.</td>
</tr>
<tr>
<td><strong>Function-system matrix</strong></td>
<td>Support inputting and listing functions and showing their relationship with systems in a matrix.</td>
<td>Support inputting and listing functions and showing their relationship with systems in a matrix.</td>
<td>Possible to make a function list template to input and list the functions and then make a function-system matrix from it.</td>
</tr>
<tr>
<td><strong>Interface matrix</strong></td>
<td>Support showing interface interactions in a matrix.</td>
<td>Support showing interface interactions in a matrix.</td>
<td>Possible to show the interface interactions in a matrix.</td>
</tr>
<tr>
<td><strong>Visualize interface interactions</strong></td>
<td>Support one-way synchronization. When dependency is changed in the interface matrix, it will be reflected in the graph.</td>
<td>Support two-way synchronizations. When dependency is changed in interface matrix, it can be reflected in the graph. When adding/deleting dependency in graph, it will be reflected back to the interface matrix.</td>
<td>Not possible.</td>
</tr>
<tr>
<td><strong>Interface register</strong></td>
<td>Support inputting and listing the interface register data in a Table format.</td>
<td>Support inputting and listing the interface register data in a Table format.</td>
<td>Possible to input and list the interface register data in a Table format.</td>
</tr>
<tr>
<td><strong>Signals</strong></td>
<td>Support inputting and listing the signal data in Table format.</td>
<td>Support inputting and listing the signal data in a Table format.</td>
<td>Possible to input and list the signal data in a Table format.</td>
</tr>
<tr>
<td><strong>Coordinate TQ processing</strong></td>
<td>Support creating and coordinating TQ.</td>
<td>Support creating and coordinating TQ.</td>
<td>Possible to coordinate TQ through coordinate features of the tool.</td>
</tr>
<tr>
<td><strong>Alert and reminder</strong></td>
<td>Support sending alert and reminder of unprocessed TQ.</td>
<td>Support sending alert and reminder of unprocessed TQ.</td>
<td>Possible to support sending alert and reminder of unprocessed TQ.</td>
</tr>
<tr>
<td><strong>TQ status reporting</strong></td>
<td>Support reporting TQ status.</td>
<td>Support reporting TQ status.</td>
<td>Possible to implement functions to support reporting TQ status.</td>
</tr>
<tr>
<td><strong>Record history</strong></td>
<td>Support recording changes of various artefacts, such as requirements, design, interface, and tests.</td>
<td>Possible to support recording changes of various artefacts, such as requirements, design, interface, and tests. Genus Nexus developed in the JIP has implemented functions to record changes of interfaces and signals.</td>
<td>Only support recording change history of an Excel cell.</td>
</tr>
<tr>
<td>Configuration management</td>
<td>Full support of version and release management. Also support &quot;diff&quot; two versions of requirement documents or code.</td>
<td>Support version and release management. Also support &quot;diff&quot; two versions of signal lists.</td>
<td>No support</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Lifecycle management</td>
<td>Support various development management processes. Can create own process through making a template.</td>
<td>Support the engineering process of submitting, reviewing, and approving of interface and signals.</td>
<td>No explicit support on specific engineering process. Only provide the coordination features.</td>
</tr>
<tr>
<td>Validate completeness</td>
<td>A custom input validator script or model checker can be created to detect missing data, and to verify that all fields are filled in.</td>
<td>Web input validator and DB data validator can be created to detect missing data, and to verify that all fields are filled in.</td>
<td>Completeness validation is limited to few Marco functions provided by the tool.</td>
</tr>
<tr>
<td>Validate consistency</td>
<td>A custom input validator script or model checker can be created to detect inconsistency data, and to verify that all fields are correctly filled in.</td>
<td>Web input validator and DB data validator can be created to detect inconsistency data, and to verify that all fields are correctly filled in.</td>
<td>Consistency validation is limited to few Marco functions provided by the tool.</td>
</tr>
<tr>
<td>Traceability</td>
<td>Support traceability from requirements, to systems, functions, interfaces, and test cases.</td>
<td>Possible to support traceability from requirements, to systems, functions, interfaces, and test cases. Genus Nexus developed in the JIP has shown traceability from operational scenario, to interface matrix, signals, and tests.</td>
<td>Can use cell link to link cells in Excel Sheets. However, it is very difficult or even impossible to establish complex traceability from requirement, to interface, and test cases.</td>
</tr>
<tr>
<td>Test management</td>
<td>Support creating test plans and test cases and capture the test process and other dimensions of the testing activities.</td>
<td>Support inputting and listing test case and test results. Genus Nexus developed in the JIP supports interface and signal test management.</td>
<td>No explicit support on test management.</td>
</tr>
<tr>
<td>Non-functional requirements</td>
<td>Support large number of data, because the system is based on industrial large DB.</td>
<td>Support large number of data, because the system is based on industrial large DB.</td>
<td>Can be difficult to manage large amount of data, because the data storage is based on Excel.</td>
</tr>
<tr>
<td>Performance: support multi-user access in parallel</td>
<td>Can support &gt;30 users access in parallel</td>
<td>Can support &gt;30 users access in parallel</td>
<td>Can support &gt;30 users access in parallel</td>
</tr>
<tr>
<td>Usability: Easy and intuitive GUI</td>
<td>Web based GUI. Easy to understand.</td>
<td>Genus Nexus developed in the JIP implemented desktop GUI at integrator or owner side, and Web based GUI at suppliers’ side. Web-based GUI can also be accommodated for</td>
<td>Web based Excel like GUI.</td>
</tr>
</tbody>
</table>
Usability: Easy to learn and adopt

A tool set includes several tools. Need moderate training to start using it. Adding or changing the function needs support from developers.

The solution is DB based. It is possible to be integrated with existing IT infrastructures of owner, integrator, and suppliers.

Usability: Easy to change template

Possible to deal with changed template of interface or signal.

Need to create or revise custom scripts.

Security: Authorization

Solid access control using client access licenses (CALs). Need to configure and coordinate security setup of tools in the tool set.

Security: Authentication

Single sign-on

Security: Data storage and privacy

Have IBM cloud based solutions.

Also provide local server-based solutions.

Security: Data storage and privacy

Prove local and third party server based solutions.

Cloud support may be added on demand.

Limited to public cloud based data storage. It means all data should be stored at Smartsheet cloud, rather than local servers of integrator or owner.

Reusable: Possible to reuse info. from previous projects

Support import information from existing Excel files. Need to create custom scripts.

Support import information from existing Excel files. Need to specify data model.

Support import information from existing Excel files. Need to make Excel templates.

Usability: Easy to learn for users with Excel knowledge. Adding or changing the functions needs support from developers.

However, it can be difficult to be integrated with existing IT infrastructures. It is because its data is stored in public cloud.

Usability: Possible to deal with changed template of interface or signal.

Need to create or revise data model.

Usability: Possible to deal with changed template of interface or signal.

Need to create or revise Excel templates.

Access control are limited to:
- Row level access control of Excel sheet.
- Define who to see a file/information in the coordination features.

In summary, although Smartsheet is applicable to manage Excel based interface information for small projects, the IBM tool and Genus Nexus are better than Smartsheet in terms of flexibility, engineering process support, data privacy, security, and scalability:

- **Flexibility.** Smartsheet supports only Excel sheet format, IBM and Genus Nexus support more input formats of requirements, interfaces, and tests, e.g., web form or Word format. In addition, both the IBM tool and the Genus Nexus support more visualization of various reports, e.g. TQ status report and graphic visualization of system dependency.
- **Engineering process support.** An artefact of a project usually needs to go through several review and approve statuses, such as submitted, under review, request revisions, acceptance, create report for signature, release with a new release number or version number. To facilitate management of a large engineering project with many suppliers for a long time, the tool should provide solid version control, configuration management, and engineering process support. Smartsheet is not developed with the purpose to support such system engineering processes.

Genus Nexus has implemented functions to support storing, comparing, and sharing different releases and versions of interface information. In addition, Genus Nexus has functions to help integrators, third-party class society, and owners to review, comment on, and approve interface and TQ submissions and revisions.

The IBM tool is designed from beginning to support distributed development with several hundred developers. The IBM tool has solid support on version control and configuration management. The IBM tool is also flexible to allow users making a process template to customize review and approval process of interface information and TQs.

The IBM tool and Genus Nexus provide functions to trace from requirements, to design, interface matrix, interface test cases and test results, and backwards. Although Smartsheet has provided cell link functions, it is insufficient or even impossible to implement such complex two-way traceability.

- **Data privacy.** Smartsheet stores data in cloud of Google, Box, or DropBox. However, interface and other engineering data are very business critical for suppliers, integrators, and owners. Storing project engineering data in public cloud may not be preferable. All suppliers, integrators, and owners have their own IT infrastructures to manage various engineering data. Interface management tool must be smoothly integrated with the existing IT infrastructures to support existing engineering process. Public cloud-based storage like Smartsheet will make it difficult to perform such integration.

- **Security.** It is possible to specify receivers in Smartsheet when sharing and discussing messages and files. However, such authorization is insufficient for managing interface information and other engineering data in complex engineering projects. In a new building rig project, it is likely that suppliers are also competitors in other projects. Thus, suppliers do not want to disclose unnecessary information to each other. In addition, there are different roles, such as supplier, integrator, owner, and third-party verifier, in a project. Different roles need different access authorities to different artefacts of the project. Thus, the authorization must be role-based.

The IBM tool and Genus Nexus provide much more authorization functions than Smartsheet to support finer granularity of authorization. Genus Nexus security is based on Microsoft active directory. It is therefore possible to grant different permissions (e.g. find and list, read and execute, create, modify, delete, read event history, read permission, and set search criteria) to different roles on different artefacts (e.g. requirements, interfaces matrix, signal, test cases, and test results). To process TQ, in Genus Nexus, it is also possible to specify who is responsible to answer a TQ. IBM security is based on client access licenses (CALs), which is also role-based and it is possible to specify similar granularity of access control as Genus Nexus.

- **Scalability.** Both IBM tool and Genus Nexus are based on large industrial DB. The IBM tool has been used to support the IBM’s internal global distributed development. So, the tools have been applied and tested in large scope engineering projects, which include several hundred developers.
work together and coordinate. The Genus Application Framework® has also been applied to develop large scope business applications, which include millions of data items in DB. As Smartsheet is based on Excel files, it is probably efficient only for small projects. IBM and Genus would scale well and provide an efficient solution for medium to large projects with hundreds and thousands of interfaces and signals.

3.5.2 Genus Nexus and the IBM tool compared

Genus Nexus is developed from scratch in this JIP using Genus Application Framework®. The key success factors of Genus Nexus, as explained in Section 3.3, are:

- **Flexibility.** Genus Application Framework® is a business process model-based development environment. It is therefore faster and cheaper to develop new functionalities, and to extend and modify existing functionalities of business and information management systems than traditional ways of coding. Genus joined the JIP at a very late stage. However, by using model-based development, it took the company very limited number of person-hours to develop an interface management tool in a short time to provide functions listed in Table 3. The Genus Application Framework® is easy to learn and requires only limited programming knowledge to master. It is therefore easy for end users to add or revise interface specification tool functionalities themselves when needed.

- **Tailor made.** As Genus Nexus is based on a model-based development environment, customers can ask for tailor made functions and cover only those functions that are needed, and not unnecessary ones. The tool can also easily be adapted to project specific requirements, processes, and concepts (terms), due to the flexibility mentioned above. The Genus Application Framework® utilizes some third-party components integrated into the Framework, like a component to present graphics, to provide advanced functions. In this JIP, Genus Nexus has shown capability to present the interface matrix in graphs through integration with a third-party graph presentation tool. In addition, Genus Nexus is based on relational DB. It is therefore easy to be integrated with companies’ existing IT infrastructure through DB data sharing.

- **Solid security.** The Genus Application Framework has been used to develop applications for The Norwegian Police and The Norwegian Ministry of Justice and Public Security. As a result, the security model in this Framework is strong. It can provide strict access control and make sure “no access to wrong data.”

As the Genus Nexus is usually tailor made, it can provide an affordable solution with a possible high return on investment. However, Genus Nexus has not implemented more system engineering functions than those listed in Table 3. It means that development effort, although the effort can be moderate, is probably needed, if users require new functions.

The IBM tool developed in this JIP is an adaption of an engineering tool set that being in use in IBM internally for supporting the IBM’s distributed software development. As explained in Section 3.2, the tool set includes a set of engineering tools (e.g. requirement management, design management, test management, and configuration management) that are running on a common framework. The tool set has more functions than specified in Table 2. The activities of IBM in this JIP is to pick up relevant functions from the tool set and customize those to make them fit the practices (terms) of Maritime and Oil & Gas industry. The success factors of the IBM interface specification tool are:

- **Systematic system engineering.** The tool set of IBM has been developed following systematic system engineering theory and practices. Some of the tools in the tool set, e.g. IBM DOORS, Rational Rhapsody, and test management, have been used in several other industries, and in
some companies of Maritime and Oil & Gas industry. The functions in these tools are therefore based not only on scientific theories, but also on extensive industrial experience.

- **Ready to use functionalities.** The IBM tool set has many ready-made functions that users can pick up, customize, and apply for interface specification and management. The tool set has also flexibility to allow users setting up own engineering process through making a process template.

- **Advanced requirement management.** IBM DOORS is the tool in the tool set for requirement management. The advantage of IBM DOORS is that it can manage requirements in various formats, such as Word, Excel, and Pdf. In addition, it is possible to store and present various requirements, e.g. functional requirements, technical specification, regulations, in a structured and hierarchal manner. The link and traceability between different tools in the tool set are well integrated. It is therefore possible to establish forward and backward traceability between requirements, design, interface, signal, and tests.

- **Advanced version control and configuration management.** As the IBM tool has been used to support distributed software development of thousands of developers, the version control and configuration management functions of the tool set are well developed. It is therefore possible for several suppliers to work on the same interface or signal lists in parallel without overwrite the work of each other.

- **Easy to use.** Another advantage of IBM tool set is that, except the Rational Rhapsody, most tools in the tool set are web-based. It means that it will be easy for participants in an engineering project to get access to the information from any place through only web browsers.

The IBM tool set includes several system engineering tools. The benefit is that if users want more system engineering functions than interface specification in the future; it is very likely that the functions are already ready made for use. On the other hand, to start with only interface specification and management, the users need to pick up and choose those relevant functions as a starting point. One possible difficulty of using a tool set rather than a single tool is that complex security configurations may be needed. It is because the access control and security setup of different tools need to be coordinated.

### 3.6 Discussions

#### 3.6.1 How to apply the tools into Maritime and Oil & Gas businesses?

For Maritime and Oil & Gas industry, one important challenge is to integrate control systems from different suppliers in a cost-efficient manner. In some cases, control systems or components to be integrated are reused from previous projects. The integrator just chooses control systems from a catalogue as COTS (commercial-off-the-shelf) and then asks suppliers to customize or configure systems to make them work together. In such “reuse” cases, the main challenge today is that different suppliers need to coordinate the interfaces between systems to make sure that the systems can communicate correctly and work together. In other cases, the control systems are “yard made”, meaning that suppliers make control of a system that is custom-designed by the yard for each individual vessel. In the “yard made” cases, the main challenge today is that the details about the interfaces to the actuators and sensors are decided too late, and are often with numerous changes along the way. Thus, in a vessel new building phase, one key element is to help suppliers, integrator, owner, and third-party verifier to communicate and coordinate interfaces and signals between systems.

Although suppliers, yard, and owners may already have certain IT systems to support their daily business, few such systems are dedicated to manage interface information. The IBM tool and Genus Nexus developed in this JIP are mainly targeted at helping integrator and owners to collect interface
information and manage the information in a cost-efficient way. To apply the tool in daily business, it is important for users to integrate the tool with their engineering and verification processes and with their existing IT infrastructure. The tools implemented in this JIP have much more functions than just interface specification. We therefore refer activities specified in DNVGL ISDS class notation [1] as examples to explain how these tools can help improve engineering practices in different engineering phases of Maritime and Oil & Gas industry.

- **Basic engineering phase:** In this phase, owner needs to define mission, objectives, and share vision for all involved roles, and define operational modes and scenarios to capture expected behaviour. The functions of the IBM tool and Genus Nexus, e.g. operational scenario and use case management, can help users organize various requirements and regulations in a systematic manner and trace the requirements to high-level engineering design. Integrator in this phase needs to establish a baseline of the requirements for the unit, allocate functions and requirements to systems, and establish traceability of requirements. Again, the IBM tool and Genus Nexus have provided sufficient functions to support such activities. The IBM DOORS can compare different versions of requirements, e.g. comparing the new version of requirements with the baseline of requirements. The function-system matrix functions make it easier to allocate functions and requirements to systems. Traceability functions of IBM tool and Genus Nexus can help integrator establish traceability of requirements from day one.

- **Engineering phase:** In this phase, integrator needs to establish and implement configuration management, coordinate inter-system interfaces, make detailed operational scenarios, review design with respect to requirements and design rules, review consistency between design and operation scenarios, review interface specification, validate critical or novel user-system interactions, and so on. The version control and configuration management functions provided by the IBM tool and Genus Nexus can help integrator implement configuration management. The requirement management functions, function-system matrix, interface matrix, system dependency visualization, system modelling tools (e.g. IBM Rational Rhapsody) can help integrator coordinate inter-system interfaces, review designs and validate system and user-system interactions. In this phase, suppliers need to be involved in several activities, such as refine system requirements into software component requirements, provide detailed operational scenarios, review the design with respect to requirement and design rules, review consistency between design and operational scenarios, review interface specification, and so on. In these activities, the IBM tool and Genus Nexus have functions to facilitate suppliers’ input requirements and detailed operational scenarios, review designs (especially interactions between systems), and review interface specification in interface matrix. These functions can also be a part of the solutions to address the problems with automation of the “yard made” systems.

- **Construction phase:** In this phase, integrator needs to establish procedures for problem resolution and maintenance activities and arrange independent testing. The TQ management functions of IBM tool and Genus Nexus can help integrator establish problem resolution procedure and follow the status of all the raised problems in the project. The version control, configuration management, and change history recording functions will help integrator store, follow, and share history of various changes. To test the interfaces, the interface management and test management functions of the IBM tool and Genus Nexus can help integrator pick up the right interface signals from the right version, manage traceability between signal and test cases, and monitor test status. Suppliers have many activities in this phase as well. Requirements, design, and interface information managed by the IBM tool and Genus Nexus can help suppliers develop and configure the software components,
perform software component testing, review software parameterisation data, qualify reused software, and perform FAT tests.

- **Acceptance phase:** In the acceptance phase, owner needs to perform validation with operational scenario and analyse validation results with respect to targets. The traceability functions of IBM tool and Genus Nexus provide information of the traceability from operational scenario, through design and interface, to test cases and test results. Such traceability information can help owner measure and evaluate test coverage. In this phase, integrator needs to manage software change during commissioning, establish a release note for the systems in ISDS scope, transfer responsibility for system configuration management to owner, and perform systems integration tests. The version control, configuration management, and TQ management functions of the IBM tool and Genus Nexus can help integrators perform such activities. In addition, storing the requirements, design, interface, signals, and test data in a DB rather than in many Excel and Pdf files can improve the cost-efficiency of transferring responsibility for configuration management from integrator to owner.

- **Operation phase:** In operation phase, most activities are owners’ responsibilities. Owner needs to manage change requests during operation, perform configuration audit, define procedure for problem resolution, change handling, and maintenance activities, perform validation testing after changes in the systems in operation, and so on. To perform these activities, it will be much easier for owner to use IBM tool and Genus Nexus to store all data in DB, and to manage software versions, configurations, and change systematically.

There are activities that are across several engineering phases, e.g. review intermediate deliverables, track and control changes to the baselines, establish a release note for the delivered system, monitor project status against plan, perform joint project milestone reviews, and control procedures. To facilitate such activities, the version control, configuration management, lifecycle management, engineering process support functions of the IBM tool and Genus Nexus can help all partners in the project to establish and agree on the baseline, review the changes against baseline, and monitor and share project progress.

### 3.6.2 Way forward

The purpose of this JIP project is to elaborate requirements of an interface specification tool for Maritime and Oil & Gas industry. Based on requirements, Genus and IBM have made tools to satisfy the requirements. As explained in Section 3.5, both tools are flexible and powerful, and are well suited for further extension and customization to provide more functionalities that are needed. To apply these tools in daily Maritime and Oil & Gas business, one necessary way forward for players, especially owners and integrators, is to contact IBM and Genus to pilot the tools in projects, and to extend or customize the tools to fit their purposes and engineering processes.

To adapt Genus Nexus in cooperation with integrator and owner, possible detailed ways forward are:

- To integrate with existing IT infrastructures and document handling systems (e.g. e-room).
- To integrate with reporting channels of third-part verifier or class society (e.g. DNVGL NPS).
- To facilitate the handover process from yard to owner.
- To include functions to deal with C&E diagrams and FMECA analyses.
- To enable signal submittals "both ways", i.e. allow integrators to submit signals as well. This is to address the issues of developing the "yard made" systems.
- Train super users and establish local experts.
The IBM tool set has many functions to support system engineering practices. The tools keep evolving along new theories and technologies, and along the companies’ own need for distributed development. Recently, IBM is adding new features to IBM DOORS and Configuration management tools to enhance configuration management and lifecycle management of requirements. To adapt IBM tool in cooperation with integrator and owner, possible detailed ways forward are:

- To scope functions to be applied and adapted step-by-step.
- To integrate with existing IT infrastructures and document handling systems (e.g. e-room).
- To integrate with reporting channels of third-part verifier or class society (e.g. DNVGL NPS).
- To establish local expert to configure and customize the tools

Applying these tools may lead to change of the traditional way of doing business in Maritime and Oil & Gas industry. To avoid dramatic changes, currently, the tools are developed and customized to be able to process interface and signal information in Excel files. However, different companies or even different teams in the same company have different Excel templates of interface and signal lists. Although the IBM tool and Genus Nexus are resilient to deal with different Excel templates, the variations of Excel template may add unnecessary effort to create custom scripts or data models. An ideal situation would be to have a standardized template of the interface and signal list across suppliers. The standardized interface and signal list template can also be used to facilitate more advanced drilling operations, e.g. autonomous drilling. However, due to limited number of suppliers in this JIP, the standardization is not achieved in this JIP and is left as a possible future work.

Requirements of a complex system includes functional requirements and non-functional requirements, such as RAMS (Reliability, Availability, Maintainability, and Safety), and security. RAMS and security specification and validation are one of the key elements in DNVGL ISDS class notation [1]. In this JIP, the tools focus mainly on managing and verifying functional aspects of MOU. Another possible way forward would be to extend the tools to be able to calculate end-to-end RAMS from design, interface, signal, and tests.
4 CONCLUSIONS

Maritime and Oil & Gas industry build more and more complex and advanced control systems to support operations. One challenge of developing complex systems using components from different suppliers is to manage and coordinate interfaces and signals between these components. In this JIP, we have collected requirements on interface specification, and have developed two tools with IBM and Genus to help owner, integrator, and suppliers to manage interface information. The tools have also functions to facilitate system engineering practices in new-building and operation phases of complex control systems.

In this report, we have summarized functions of the IBM tool and Genus Nexus and compared them. We have also compared these two tools with an Excel based project management tool called Smartsheet. Although IBM and Genus have proposed price models, we have limited the comparison and discussion in this report on the technical aspects of the tools without going into details of the possible cost of buying and using the tool.

In summary, we believe the IBM tool and Genus Nexus fit better than Smartsheet to manage interface and system integration of complex control systems. It is because the IBM tool and Genus Nexus have better flexibility, engineering process support, data privacy, security, and scalability than Smartsheet.

The IBM tool is a customization of a collection of well-established system engineering tools. The advantages of the IBM tool are: many ready-made solid systems engineering functions, advance requirement management, configuration management, and test management, and most functions are implemented in web-based GUI.

Genus Nexus is developed from scratch within this JIP based on the Genus Application Framework@. The advantages of Genus Nexus are: flexible, easy to be tailored and extended, integrated with several good libraries or features to support fast tailoring and extension, and most functions are possible to be implemented in a web-based GUI.

The IBM tool and Genus Nexus have been demonstrated in a final demonstration, where most partners of the JIP have presented. The demonstration is recorded and is available for sharing on request. However, we have received little feedback from the partners after the demonstration. One of the ways forward for partners of the JIP is to pilot-test the tools in real projects. The IBM tool and Genus Nexus can be available on the internet for JIP partners for trials for a time. Another way forward is to standardize interface and signal templates and to focus more on calculating end-to-end RAMS from information collected in the tools.
5 ACKNOWLEDGEMENT

We would like to thank all JIP partners for their financial support, for contributing information to specify the requirements of tool, for attending meetings and demonstrations, and for giving feedback. We would also like to thank IBM and Genus for their effort on developing the tools. We will give special thanks to HHI for their valuable time.
6 REFERENCES

/1/ **ISDS Class Notation:** DNV OS-D203, Integrated Software Dependent Systems, December, 2012.

/2/ **IBM access control:** https://jazz.net/help-dev/clm/topic/com.ibm.jazz.repository.web.admin.doc/topics/c_understand_user_access_control.html
7 APPENDIX A. DETAILED FUNCTIONS OF IBM TOOL

The IBM tool can be available on the internet for JIP partners for trials for a time. Please contact Peter Asp (peter.asp@se.ibm.com) for details.

Figure A.1. Traceability View

Figure A.2. Design to Change Management relationship.
Figure A.3. Example Technical Query

Figure A.4. Example Dashboard for capturing and act on Technical Queries

Figure A.4 shows how changes to interface information is linked to a Task capturing information on when and who made the changes, and what changes were made. The individual versions of the interface information can be compared with each other to show the differences for audit purposes.
Figure A.5. Example of templates

Figure A.6. Example of relationship graphs
Figure A.7. Example of sequence diagram

Figure A.8. Test Case Traceability View
Figure A.9. Test Case to Requirements Coverage graph

For further information please follow the link to our solution web site and read about our best practices that support Systems Engineering projects in more detail.

8 APPENDIX B. DETAILED FUNCTIONS OF GENUS NEXUS

Genus Nexus can be available for trial use. Please contact kristian.aaslund@genus.biz for demonstration and setup.

Object data model

Figure B.1. An example of visualisation of the object model in Genus Studio®, the model-driven environment used to develop Genus Nexus.
High level overview

Figure B.2. The yard’s system overview showing the latest submittals, open TQs and state of interfaces.

Interface matrix

Figure B.3. The interface matrix showing the interfaces between the systems. New interfaces are created by the yard and the underlying signals are submitted to each interface by the suppliers.
Detailed overview

Figure B.4. The yard’s system overview with detail view of one system.

Project overview

Figure B.5. The project information and supplier list.
Signal submittal

Figure B.6. Submittal of a new signal list in the supplier web portal. The supplier uses its own Excel format.

Figure B.7. The list of submittals presented to the yard in the desktop system.
Signal review

Figure B.8. A signal review in progress by the yard. The submitted signals are compared to the existing signals in the database and presented in a diff view with highlighted changes. The reviewer accepts/rejects/comments on the signals and gives a total feedback on the entire submittal.

- Top left highlight: Submittal information and link to the original signal list. Work flow for the signal review process.
- Top right highlight: Legend explaining the colour coding.
- Bottom left highlight: Grouping of signals based on comparison to the existing signals in the database.
- Bottom right highlight: Indication of the actual changes in each signal.
Figure B.9. The result of a signal review presented to the supplier in the web portal.
Figure B.10. The yard’s overview of the technical queries (TQ). The details of the selected TQ are displayed.

The yard can perform multiple operations on the TQs:

- Close, reject and put the TQ on hold
- Add and remove recipients
- Link the TQ to existing systems and interfaces
- Reply to the TQ
- Create a printable Pdf report of the TQ
- Create a new TQ
Figure B.11. A supplier’s reply to a technical query with a file attachment. The supplier can also create new TQs in the web portal.
Operational scenarios, releases, testing and traceability

Figure B.12. The yard’s management of the vessel’s operational scenarios
Figure B.13. An official release of the current signal revisions of an interface. Released by the yard.

Figure B.14. An interface test created by the yard, based on a released interface and published to the involved suppliers. An interface test can be traced to steps in an operational scenario.
Figure B.15. Interface procedures published to each involved supplier in the web portal.
9 APPENDIX C. DETAILED PRICE MODELS

Chapter removed for confidentiality reasons.

About DNV GL
Driven by our purpose of safeguarding life, property and the environment, DNV GL enables organizations to advance the safety and sustainability of their business. We provide classification and technical assurance along with software and independent expert advisory services to the maritime, oil and gas, and energy industries. We also provide certification services to customers across a wide range of industries. Operating in more than 100 countries, our 16,000 professionals are dedicated to helping our customers make the world safer, smarter and greener.