CATALOGUE DATA MODEL BASED INNOVATIVE DATABASE SYSTEM FOR SHIP STRUCTURAL AND MACHINERY CONDITION MONITORING (MCM) DATA EXCHANGE PROTOCOL

A. Taheri[1], I. Lazakis[1] and T. Koch[2]

[1]Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, 100 Montrose Street, Glasgow G4 0LZ, UK, atabak.taheri.100@strath.ac.uk, iraklis.lazakis@strath.ac.uk
[2]Atlantec Enterprise Solutions (AES), Hamburg, Germany, thomas.koch@atlantec-es.com

ABSTRACT

This paper is about the machinery and overall ship database system used for ships to be used by the maintenance and inspection methodology and tools created in the INCASS (Inspection Capabilities for Enhanced Ship Safety) European Project. The aim is to minimise failures and increase the energy efficiency of the ship. Because of the complexity of the data pattern obtained on a condition based maintenance system in ships, it is crucial to create a customizable and effective central database system. This system should be able to combine three different data structures of historical data, continuously monitored data and analysed/manipulated data. Therefore, the central database in this paper is based on data models defined in ISO 10303 and ISO 13584 standards and it has an object-oriented graph database management structure. Base of the database architecture uses Topgallant® Information Server product. Major raw data inputs for this database are obtained from operators, sensors, classification societies, OEMs and other databases. More analytical clients of the database are INCASS Machinery Reliability Analysis (MRA) systems, Structural Reliability Analysis (SRA), FEM analysis tools and INCASS Decision Support (DSS) system. JAVA based APIs are used to connect the database to its analytical clients. A data exchange protocol has been created in order to connect main database with other available databases such as ship inventory database used by DANAOS Shipping Company. This has led to development of a unique data exchange format called MCM (Machinery Condition Monitoring) data format and updated previously introduced HCM (Hull Condition Monitoring) protocol. Data itself inside the database are categorised in several major sections of operational parameters including voyage information, general conditions used at the time of data collection, more static information type data such as general engine information based on manufacturers' data.

Keywords: database, P-LIB, ship machinery, parts catalogue, database programming, API, MCM

NOMENCLATURE

AES	Atlantec Enterprise Solutions
API	Application Programming Interface
DBMS	Database Management System
DSS	Decision Support System
ER	Entity Relationship
FEA	Finite Element Analysis
GUI	Graphical User Interface
HCM	Hull Condition Monitoring
INCASS	Inspection Capabilities for Enhanced Ship Safety
ISO	International Organisation for Standard
JDBC	Java Database Connectivity
MCM	Machinery Condition Monitoring
MRA	Machinery Reliability Analysis
OEM	Original Equipment Manufacturer
OBDC	Open Database Connectivity
POID	Persistent Object Identifier
RDMS	Relational Database Management System
SRA	Structural Reliability Analysis
XML	Extensible Mark-up Language
XSD	XML Schema Definition

1. INTRODUCTION

A standardised and structured maintenance system is vital for safe shipping and lowering the ship emissions and fuel consumption of vessels. Effective maintenance schedule can significantly decrease the number of breakdowns in a system and increase the reliability of the system. More advanced condition based maintenance systems such as the one used for INCASS (Inspection Capabilities for Advanced Ship Safety) European FP7 project can additionally monitor the performance of the different components and their influence on overall performance, emissions and fuel consumption of the vessel. Therefore, it can suggest inspection and repair jobs through cost-benefit analysis in order to lower emissions and fuel consumption.

In general, the maintenance methodology used in INCASS project has two major sub-sections of the structural and machinery analysis units with their own unique decision making tool. They both get their information from sensors, past maintenance schedules, analysis tools, expert knowledge and other relevant databases. Consequently, this would require a substantial data gathering, query and storage system in order to manage all the large amount of data that has been obtained from different sources. As a result, this paper represents a unique database management tool created in this project in order to manage all the different types of data gained from different sources for both structural and machinery analysis tools.

Main benefit of having this database system is to gather and store all the data analysis results in one place that can be easily be read by the maintenance managers and inspectors. It also standardises the communication strategy between different analysis tools, systems, sensors and other databases. In summary, this paper will give a brief literature on different database types and tools used in industry. Then it will discuss about the architecture, database model type used, the general data exchange protocol and client (stakeholder) interaction case studies used on both structural and machinery databases for the INCASS project. Subsequently, general database interfaces for both database types including the general database type selection process and effectiveness of the overall database is discussed in section 5. Finally, the paper will be concluded with the summary of the points represented and statement that will be performed in near future.

2. DATABASE TYPES

All databases are either analytical or operational. Analytical type databases are read-only type that is more useful to store historical data quickly with as little space or computational power as possible. On the other hand, are more dynamic databases that allow data manipulation and are more used with real-time dynamic type data input types (Sol, 2003). Operational databases would require special tools in order to transact, sort, query and update data. These tools are provided through Database Management Systems (DBMSs). All different types of DBMSs facilitate data modelling inside a database by developing unique linking systems with hardware data to software analysis. There are five major types of DBMS structures available: Hierarchical database, Network database, Relational database, Object Oriented and Graph database (Faircloth, 2014).

2.1 HIERARCHICAL DATABASE

In this type of database, data is structured in a tree like format. This means that different data records are connected to each other through a parent-child relationship pattern in top to bottom format for doing its query (Celko, 2014). It can be relatively easy to use and very efficient for simple data models. However, if you want have interconnections between different child from different parent records, it can create data repetition. For complex models also can make it rather difficult add totally new attributes.

2.2 NETWORK DATABASE

This type of database represents data links with network style connections which makes it possible to have child to child interconnection possibility as well as having parent to child relationships (Celko, 2014). It has more bottom-to-top style layout and it results in much easier way of adding new data layout as they can be simply connected through network connection to their related data records already existing inside the database. Though, this database can be rather difficult to model for a data records with too many interconnections. These data interconnections also require higher computational power to process, which can be problematic on large database structures.

2.3 RELATIONAL DATABASE

This database type uses relational tables instead of any kind of node patterns to organise the relationship between different data records. Rows of tables that have the data record contain their own unique code that can be used to connect or create a structure with other records from other rows of other tables. Columns of the tables represent the unique value and attribute types for the data record entered or present at each row. In order to create codes and connection between different rows of data unique software systems are used, which are called Relational Database Management Systems (RDMS) (Xu, et al., 2013). Due to lack of parent to child links used on first two main database types, this database system can be more efficient on managing complex data structures. However, it can sometimes be important to have a volume of parent to child relationships in order to have pattern history of the data query. Databases with high numbers incoming data it can also require more computational power to create tables and then codes for each row.

2.4 OBJECT-ORIENTED DATABASE

In this database management system, data are represented as objects unlike the relational database where they are represented by tables. They require object-oriented programming language and they are becoming a norm in industry on creating major databases (O'Brian & Marakas, 2009). It is highly efficient and object language relations created can allow illustration of any parent to child relations between data records. However, query between different data records can be rather difficult due to complex coding and connection types.

2.5 GRAPH DATABASE

Graph database models are usually used in conjunction with other database types such object-oriented database, in order to facilitate the query process. This is due to the fact that this database type represents data, objects, tables or information related to the data in forms of the edges of a graph. This makes it rather easy to relate different data records with each other or with certain information in more structured manner (Neubauer, 2010).

2.6 OFF-THE-SHELF DATABASE MANAGEMENT SYSTEMS

There are further readily designed database tools that facilitate the creation, recording and enquiring data for the users around the world. They can use any different style of the database can let people to subscribe in order to use their on-line storage spaces. However, they all have their own limitations. Additionally, they are not very flexible and may also create data security issues as end user has to save data inside an online platform. Table 1 demonstrates a sample of major off-the-shelf database management tools with full comparison of their capabilities and their relative weaknesses shown in form off their capabilities and missing areas.

Table 1 - Off the Shelf Database Management Systems Comparison

	OBDC	JDBC	Browse	wizard creation	auto completion	multi-server	Import	Export	Debugger	reverse engineering	Forward Engineering	Security	ER Diagram
Adminer	No	No	Fully Supported	Fully Supported	No	Some	SQL script, CSV, TSV and all in zip	SQL script, CSV, TSV and all in zip	No	Yes	No	Medium to Low	No
EMS SQL Manager Light	No	No	Fully Supported	Fully Supported	Yes	Yes	CSV,XML, MS Excel	CSV,XML, MS Excel	No	Yes	Yes	Medium	No
Firebird	Yes	Yes	Procedure and trigger not supported	Partial	Some	Some	SQL, ACID	SQL, ACID	No	Yes	No	Medium	No
Sqiuirrel SQL	Yes	Yes	Procedure and trigger not supported	Partial	Yes	Some	Mostly	Partial	No	Yes	No	Medium to Low	No
SQLite Database Browser	No	No	Fully supported	Fully supported	Yes	Yes	CSV,XML, MS Excel	CSV,XML, MS Excel	No	Yes	Yes	Medium	No
SQL Workbench	No	No	Fully supported	Fully supported	Yes	Yes	CSV,HTML,JSON, MS Excel, SQL, XML	CSV,HTML,JSON, MS Excel, SQL, XML	Yes	Yes	Yes	Medium	Yes

3. STRUCTURAL DATABASE

Structural database created for the INCASS project is advanced version of the structural database created on a previous sister project called RISPECT. Data storage engine used for this database is a product of Atlantec Enterprise Solutions (AES) called Topgallant® Information Server (INCASS, 2014a). Subsequently, this section will start with a brief description of the all the system components and their interactions inside a unique database architecture. Then it will represent the data model concept used for structural database with other complementary analysis tools integrated to it. Next, it will demonstrate the data exchange strategy used between database and its stakeholders. Finally, an INCASS project case study of the structural database with its client interaction will be illustrated.

3.1 DATABASE ARCHITECTURE

Main components of the INCASS structural database are Topgallant® Information Server, Data storage configuration tool, Data model definition tools, Application Programming Interface (API), Adapter Framework (ADK) for data import/export and INCASS database on-board/on-shore Graphical User Interface (GUI). Data model architecture used in this database follows ISO 10303-11 (EXPRESS) format (ISO10303-11, 2004) or as an XML schema definition (XSD) (Thomson et al., 2004) and (Biron and Malhotra, 2004). Based on these definitions, all API complementary codes are generated to provide a programmatic early-bound access layer for the complete model. Documentation can also be produced in the format of a data modelling oriented documentation as well as an API documentation.

3.2 DATA MODEL AND DATA ENTITIES

Data model used for structural database contains data entities that provide the base for modelling engineering type data used for the project. There are two major classes of objects used for the database: first-class objects and second-class objects. First-class objects are inherited abstract data entity class that add information on main data record such as `InformationObject`. These are the searchable persevered data entity types and they are the foundation objects for the other publicly visible objects. As a result, they are main query forms used within the database.

On the opposite side, second-class objects are more minor supporting objects that do not inherit information from `InformationObject` and can only exist along that first-class object. Both object types are working to implement `BoundObjectInstance` interface that offers POIDs (Persistent Object Identifier) as identifiers.

Whole database architecture is based on parent to child inheritance pattern. One of the parent objects is considered the primary parent, which classifies the rest of the hierarchy for that particular data entry. In general, various hierarchies of the data model are identified as below (INCASS, 2014a):

- FunctionalSystemNode
- HullDesignStructureNode
- Zone, Compartment
- ThicknessMeasurementCampaign,
- AnalysisScenario, AnalysisCase

The structure model can be represented in different forms of Physical structure, FEA (finite Element Analysis) model and Simplified analysis model. The physical structure contains the major physical components that are part of the ship hull and superstructure. The FEA-model is a an idealised model of the structural components that is more suitable for FEA. The simplified analysis model is more succinct representation of the even more simplified of structural components for analysis and illustration proposes.

3.3 DATA EXCHANGE

Data exchange protocol for the structural database follows the HCM (Hull Condition Monitoring) formatted data. HCM is an XML-based data exchange format that is being maintained by the OpenHCM consortium. HCM can be used to transfer data about:

- The hull structure details of a vessel (Ship Data and Ship Structure)

- The results of a Thickness Measurement Campaign, which describes the results of an on-board structural survey (the survey may have been performed by a human operator or by robotic equipment)
- The results of a Hull Survey, which describes the observed corrosion conditions and structural defects. (This is an extension to the current OpenHCM 2.0 definitions)

The further on-going work on the application of imaging technology for condition assessment will add more features and coverage to the overall data exchange framework. Due to variations on sensorial data types provided, they will be turned into a time series of data points relating to a particular property being detected with an extra inherited property being connected to another observed information object. For Example, a location in the ship's coordinate system can help to provide acceleration data and a point in the structure can provide location for stress measurements.

3.4 STRUCTURAL DATABASE CLIENTS AND CASE STUDY

In INCASS project data from structural section is provided from five major sources of historical/restored data such as OEM (Original Equipment Manufacturer) data, real time measured data from installed sensors, inspection reports, measured data from robotic systems, and other database sources such as classification society data. These data are then filtered in order to eliminate any repetition or unwanted data and then restored in the database. Successively, core structural analysis tools from the INCASS project (i.e. hydrodynamic analysis, FEM analysis and Structural Reliability Analysis (SRA)) extract the raw data stored in database for their analysis. Overall structural database to its clients interaction model is shown on Figure 1 (INCASS, 2014c).

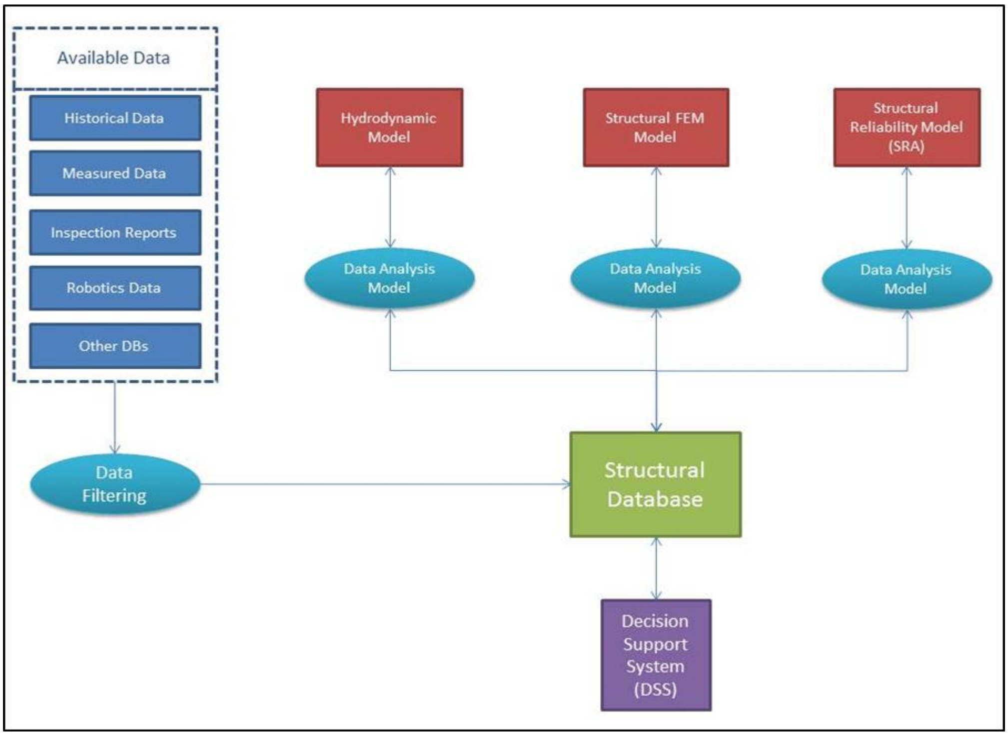

Figure 1 - Client Interaction Model for Structural Database (INCASS, 2014c)

These raw data can be classified as one of five major categories: voyage data, coating conditions, corrosion analysis data, crack growth data and other relevant structural information such as residual stress, yield stress, fracture toughness and surface imperfection measurements. Finally, structural DSS uses results obtained from all of the three analysis models, and combines it with cost-benefit data including spare parts prices and repair costs to create its decisions on maintenance and repair tasks. Database itself has two major data storages and GUIs for both on-board and on-shore uses.

4. MACHINERY DATABASE

Machinery database created for this project is totally new and ingenious database that uses catalogue data model and follows similar ISO standards and EXPRESS data schema. Therefore the next subsections will explain database platform including the catalogue data model used, the developed data exchange protocol and an example of the INCASS machinery database client interaction case study respectively.

4.1 CATALOGUE DATA MODEL

This type database stores data inside a catalogue style structure with different types of inheritance between catalogue and items inside the catalogues. The whole system is regulated by the catalogue dictionary has main sets of `CatalogueItemClass` and `CatalogueProperty` data objects. The hierarchical prent child relationship of items can be created through `CatalogueItemClasses`. The hierarchy of classes follow a tree like structure as multiple inheritances will not be possible. However, a way around this restriction can be identified only in important special cases (INCASS, 2014b).

Few entity types help to define the catalogue model such as catalogue object, which is the origin of a catalogue data hierarchy. Entities create a solid meta-data framework to create data record inheritances such as `information_objects` of `erm_object` and catalogue with attributes `of catalogue_item_classes` and `catalogue_properties`. further definitions. Table 2 shows a typical example of entity catalogue, where its attributes are shown. On next subsections of the table, inheritances of the entity catalogue and their attributes are illustrated. The second column of the table illustrates the type of the attribute input/output. The third column also defines if the attribute is a main attribute, derived attribute or an optional attribute and forth column provides further description on the attribute (INCASS, 2014b).

Table 2- An Example of Catalogue Entities Definition (INCASS, 2014b)

Attributes of catalogue_item_instance			
properties	LIST OF key_value		*the set of properties describing this item*
supplier_code	STRING	Opt	*a supplier specific code related to the item (e.g. product number)*
type_code	STRING	Opt	
Attributes of erm_object			
extended_attributes	LIST OF key_value		
native_data	SET OF parameter_set		
Attributes of information_object			
aliases	LIST OF key_value		
common_name	STRING		
contributors	LIST OF STRING		
description	STRING		
digest	BINARY	Opt	
parents	LIST OF information_object		
targets	SET OF instance		
thumbnail	BINARY	Opt	
path_name	STRING	Der	

APIs (Application Programming Interface) introduce a connection platform between different data clients and the database. This is created in JAVA language so it can be adaptable in most frameworks and operating systems. They also offer a systemised access to database functionalities. Additionally, it would facilitate the automatic and dynamic data storage and manipulation through database. First JAVA API type developed is called Connection Modes, where the user can define the connection types such as private or public, and even create a unique password for the connection. Subsequently, Object Creation Helper and Object Manager Queries API documentations would help the user to define data types (i.e. dynamic, static, inherited, etc.) and identify the query style they should have. Successively, POID Constructors create unique data identification IDs in order to facilitate data storage and queries. Finally, Transaction and Query APIs implement the previously defined data storage and query actions for the desired data record.

4.2 DATA EXCHANGE

In similar way to the HCM protocol, a Machinery & Equipment Condition Monitoring (MCM) protocol has also been introduced in order to standardise the data exchange between different stakeholders. This protocol contains the following data transfer scope:

- Definition of monitored equipment
- Observations and measurements from condition snapshots
- Measured sensor data

The MCM format is modelled along the same general concepts as HCM, utilizing as many common data definitions as possible. Typical data sets to be transferred will include:

- On-board inventory of equipment
- Sensor time series
- Imaging for documentation purposes

In addition, MCM will be prepared to transport imaging recognition data generated from post-processing similar to the capabilities provided by the HCM extensions defined in section 2. Since INCASS currently does not include any image post-processing for condition assessment concerning equipment, this will be tested as part of the INCASS implementation. The format is specified as an XML Schema [XSD-1, XSD-2].

4.3 MACHINERY DATABASE CLIENTS AND CASE STUDY

Data Clients or stakeholders for the machinery side of the INCASS database can be classified into two major areas of the raw data providers (i.e. ship operators, Classification societies and OEM reports) or the secondary data users for analysis proposes (i.e. Machinery Reliability Analysis (MRA) and Machinery DSS). Raw data provided by stake holders are filtered in order to eliminate unwanted data such as repetitions and some extraordinary circumstances. This is vastly important as it decreases data storage and increases data recovery speed. Overall machinery database to its clients interaction model is shown on Figure 2 (INCASS, 2014c).

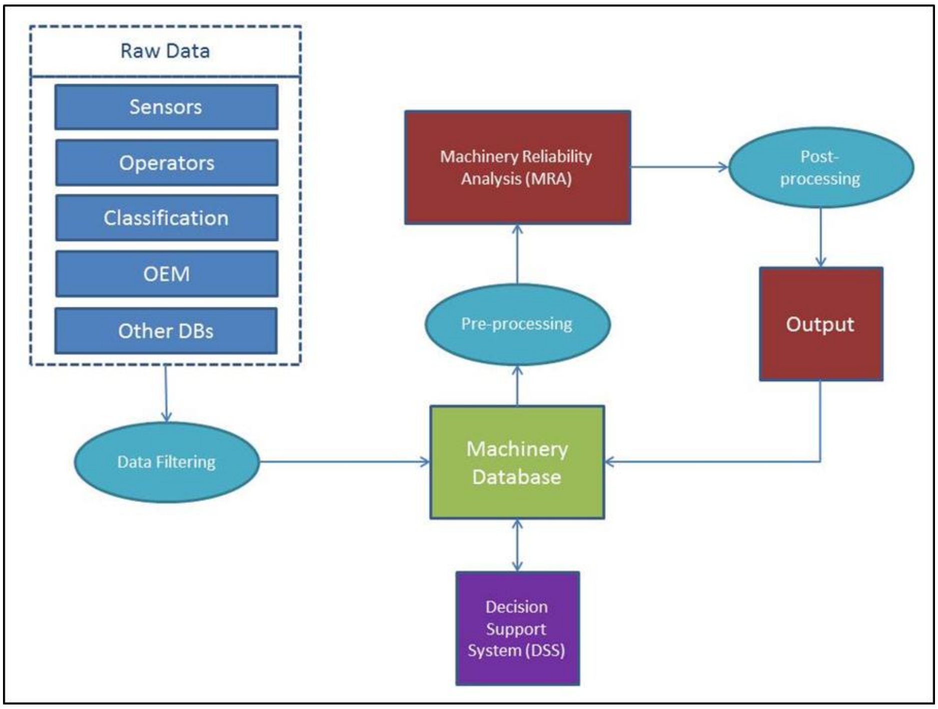

Figure 2 - Client Interaction Model for Structural Database (INCASS, 2014c)

Machinery Reliability Analysis (MRA) system then uses the raw data in order to accomplish a reliability analysis. These results in probabilistic changes in degradation pattern of the machinery components and can predict the approximate time to failure. Subsequently, MRA results are fed back into the database as processed data. MRA itself require some data pre-processing in order to turn raw data such as temperature changes into probabilistic/statistic data using predefined thresholds. Finally, results from MRA and other cost information including spare part prices, repair costs and consequences of a certain failure are fed into machinery DSS. Machinery DSS schedules maintenances and prioritises tasks, which is then fed back into the machinery database.

5. DATABASE INTERFACES AND DISCUSSION

This section looks into the different Graphical User Interfaces (GUIs) developed for both structural and machinery databases. These GUIs help end users to interact with data in order to use them or add new data manually. At the end of this section, a brief discussion of overall methodology used is provided.

5.1 STRUCTURAL INTERFACE

As explained in previous section, this the Project Manager application GUI facilitates any manual data manipulation, addition or query. This would also allow further inspection of any data added automatically through APIs from other stakeholders such as machinery DSS (Figure 3).

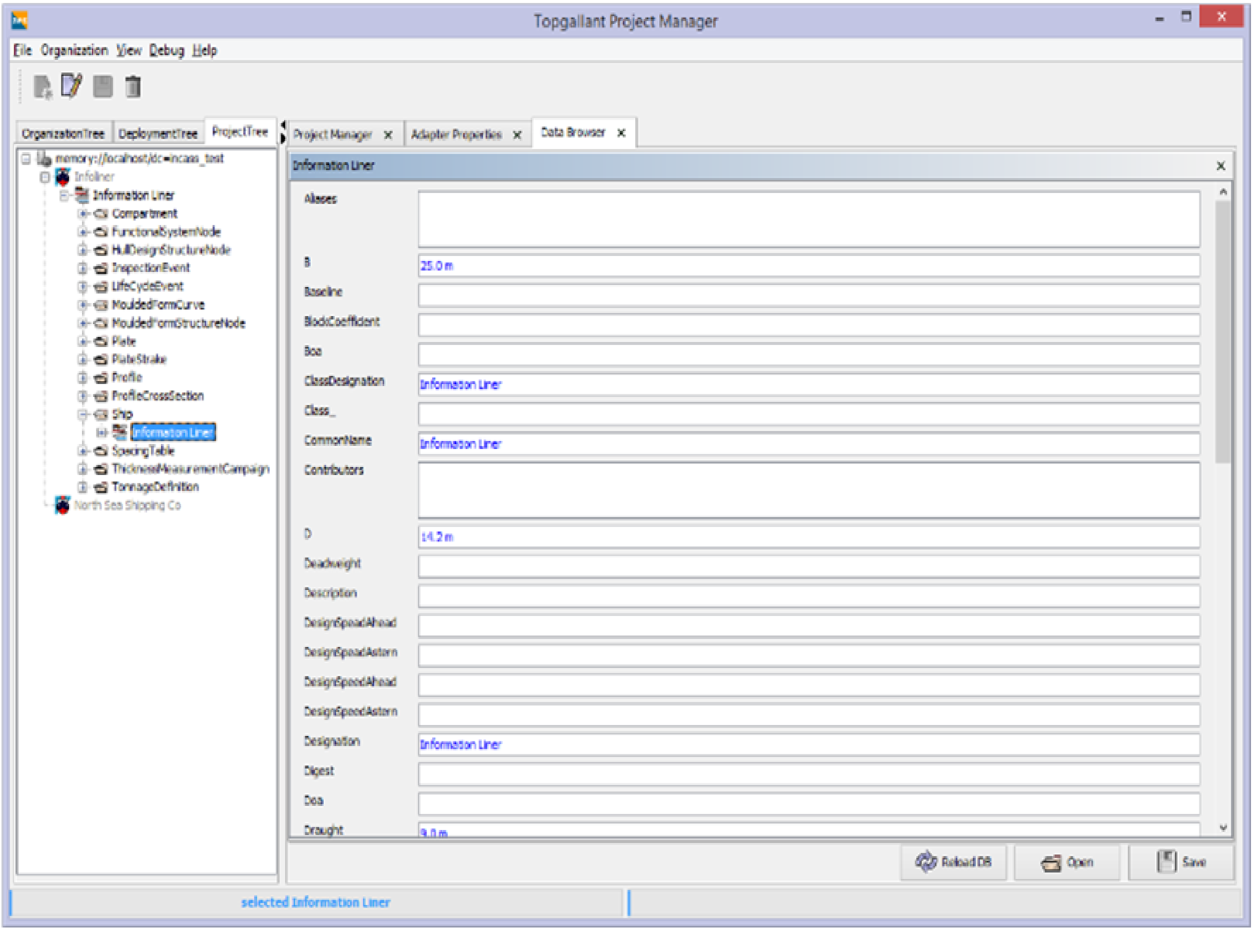

Figure 3 - Structural Database Project Manager User Interface (INCASS, 2014a)

Project Manager illustrates three different management levels in more graphical format. These management levels are:

- Organization Tree: Presents the server, server name and a hierarchically view of the organization, project and product and also shows basic definition of each entity
- Project Tree: Displays ship modelling procedure, organization of all included objects and comprehensive information on stored items
- Deployment Tree: Demonstrates detailed information about the adapter properties and data creation forms

Project Tree helps to browse through the storage structure using POIDs. Objects are classified by database settings through main object (e.g. a Ship), type of entity (e.g. a Plate), a slot or gap (which is defined by the fully-qualified pathname of the object) and version (Figure 4).

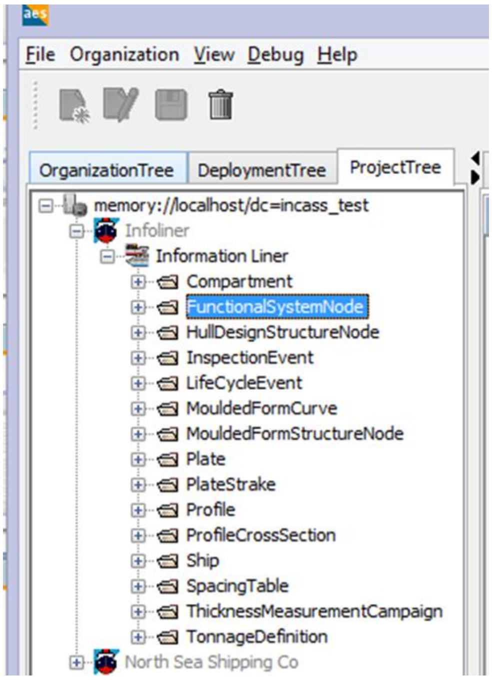

Figure 4 - Selecting Object Type for Browsing in Structural Database (INCASS, 2014a)

This navigation mode also helps on creating query of any stored object. Another possibility this tool creates is to be able to circumnavigate through supporting/dependent objects (Figure 5). This would also let the end user to export the XML representation of an object for closer inspection.

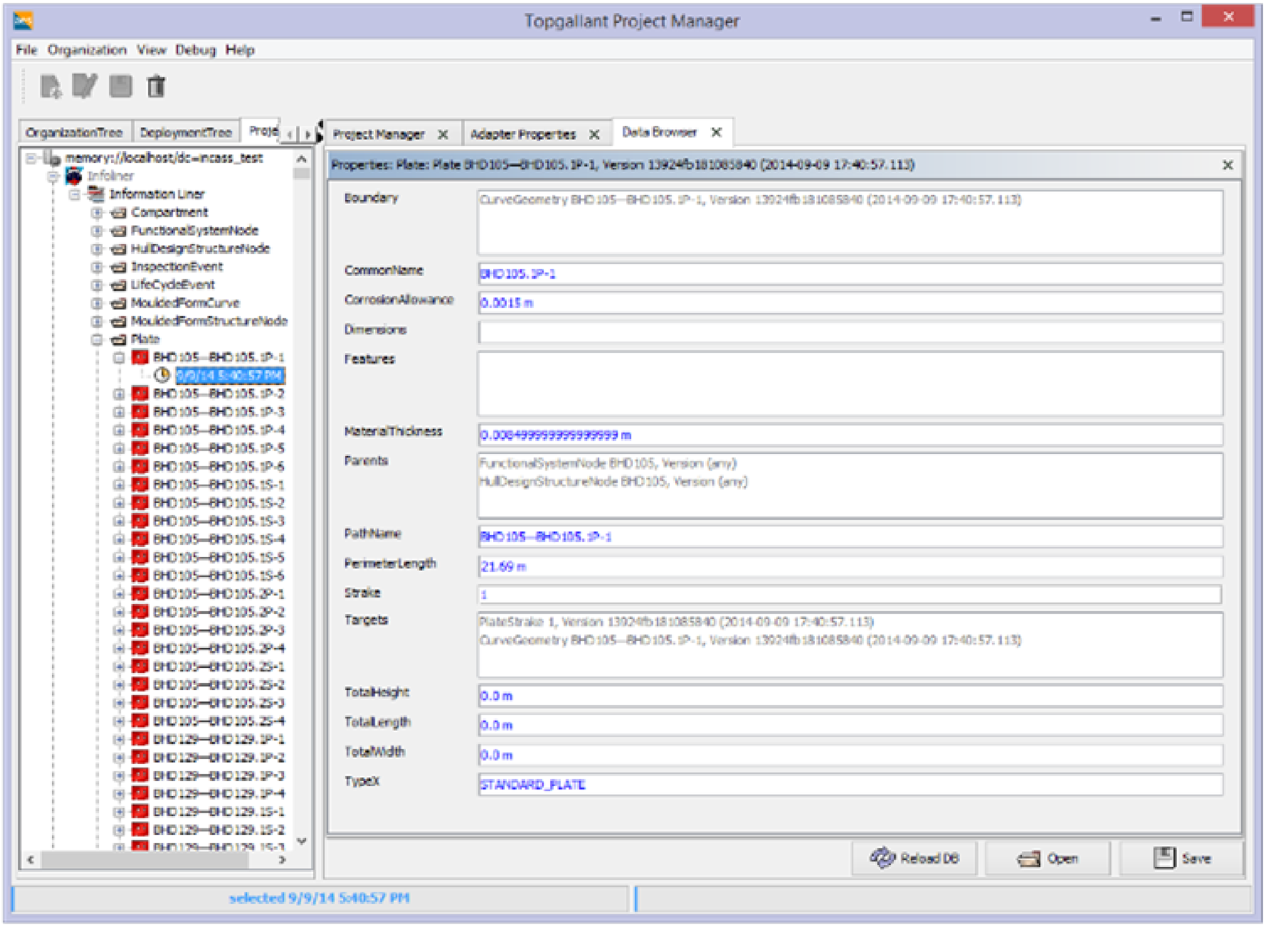

Figure 5 - Browsing an InformationObject in Structural Database (INCASS, 2014a)

3.2 MACHINERY INTERFACE

Machinery database manager lets end users to define catalogue properties in two major sections of static properties (i.e. Historical data) and dynamic properties (i.e. Sensorial data and MRA analysis data) (Taheri et al, 2015). Users can also organize the data inputs in different classes as defined by the catalogue items properties (Figure 7).

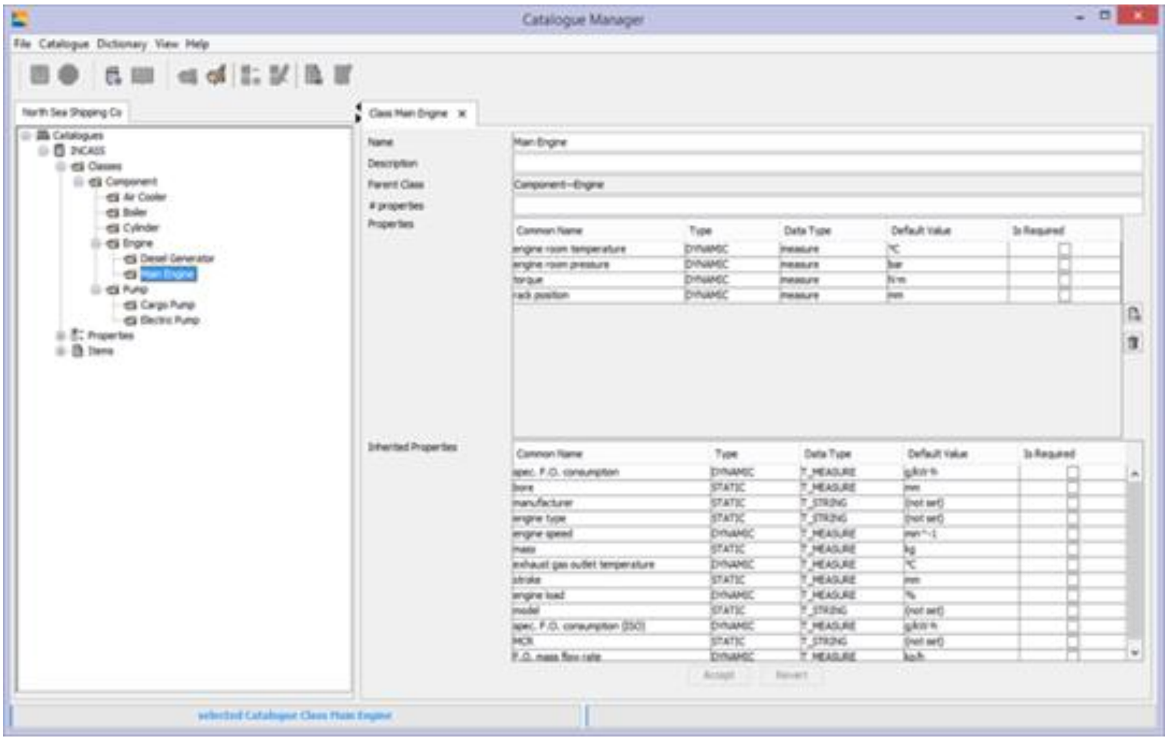

Figure 6 - Catalogue Data Classification (Taheri et al, 2015)

The next step is to connect different cataloque items and develop item data inheritances. Therefore, catalogue dictionary is used in conjunction with acquired meta-data to develop catalogue items. As a result, connection between different data records are established (Figure 8).

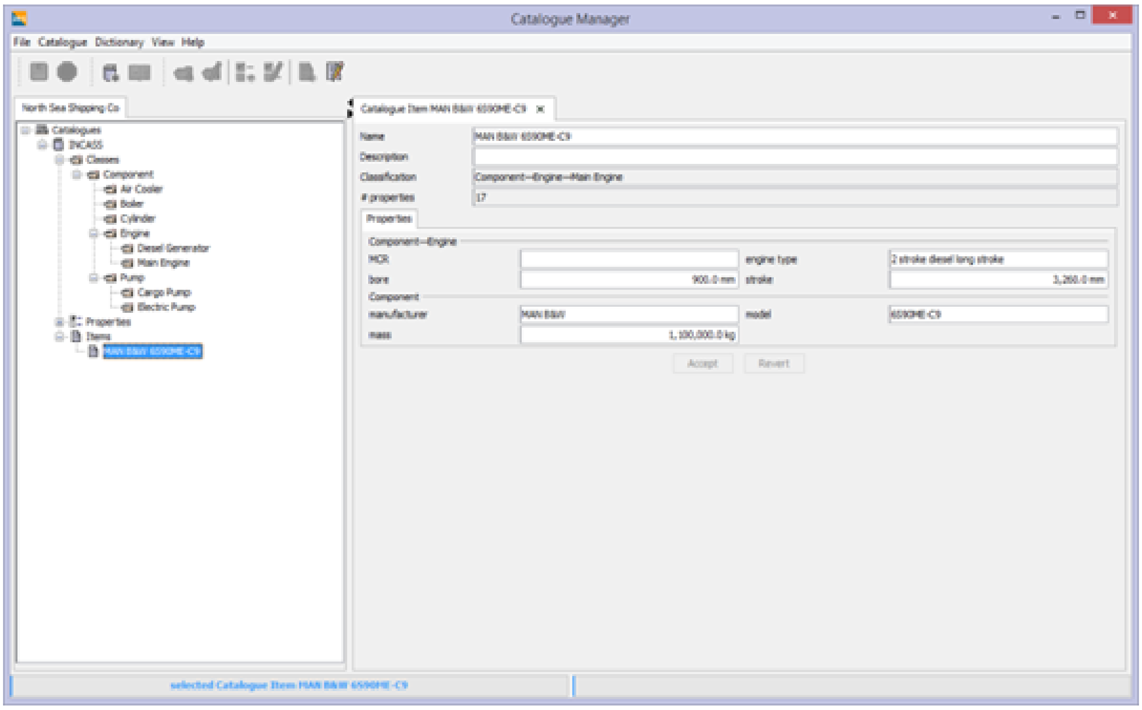

Figure 7 - Catalogue Property Definition (Taheri et al, 2015)

For more dynamic and highly fluctuating data, equipment editor section of the machinery database can record the full life-cycle of the data record in more graphical format. This would help the end users to easily obtain any old data in order to perform calculation and create further useful graphs (Figure 9).

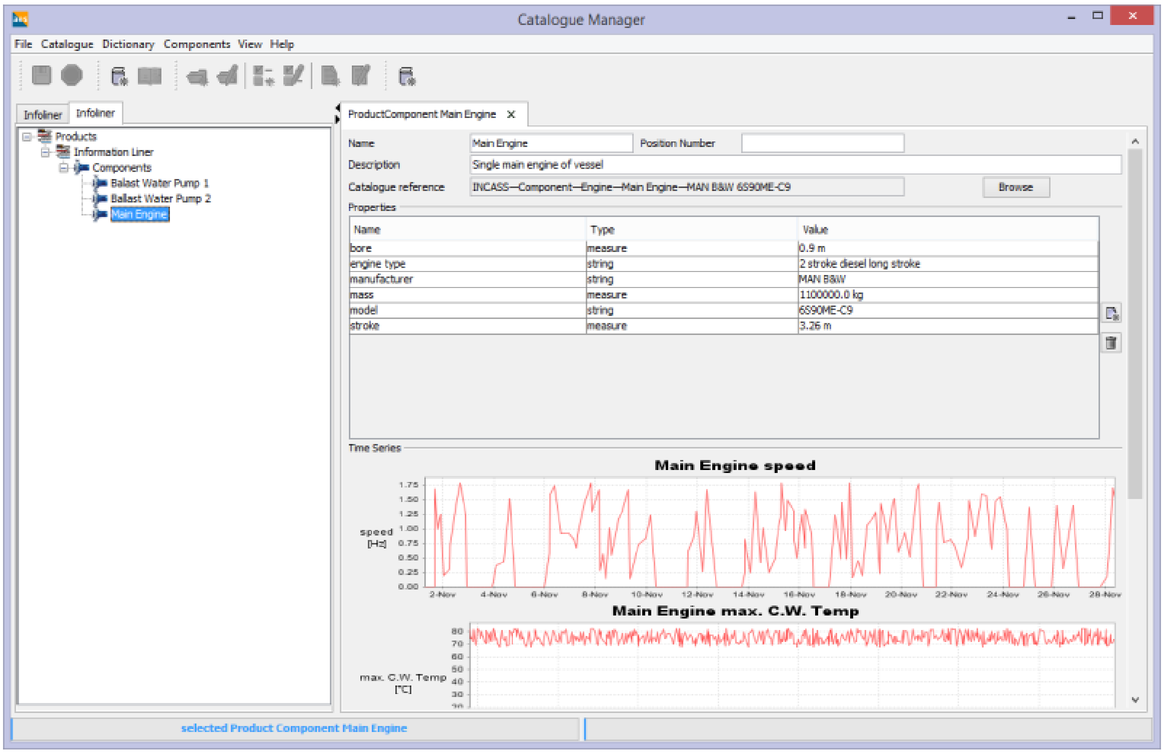

Figure 8 - Equipment Editor (Taheri et al, 2015)

3.3 OVERALL DISCUSSION

In brief, both structural and machinery databases created for INCASS project are fully described in this paper use object-oriented database structure. This creates a standard format for gathering data from both historical and on-board dynamic data gathering campaign in quickest and most efficient way possible. Graph database connections are also implemented in order to facilitate the data connections and queries within the database.

Structural database model can handle not just raw and calculated data, but can also save graphical results from both on-board inspection and robotic platforms. This is due to availability of connection with the image handling tool available in the project. Additionally, structural database has advantage of using ship structural views and coordinates in order to exactly identify the location of the data record related to any structural elements. Additionally, this database lets stakeholders to manually control and search for any data required through the GUI database management tool.

Moreover, machinery database model of this project can distinguish between raw static data such as voyage logs and OEM data and its dynamic counterparts such as on-line sensorial measurement campaigns. Additionally, dynamic data calculations can be automatically and manually (through database management GUI) to be added to the database. Machinery database also has a functionality of recording data in a dynamic pattern in graphical representation in order to show the full history of any dynamic data input types. In general, having comprehensive database tools on both structural and machinery sides of a ship helps engineers to more effectively monitor different aspects of the ship, which in turn can eliminate any unwanted failures, reduce fuel consumption, decrease emissions and increase profitability.

6. CONCLUSION

In conclusion, this paper represented the overall structural and machinery databases created for ships used in INCASS EC7 European project. Having a standard comprehensive database to record all ship conditions and maintenance/inspection actions can help to diminish unwanted failures and increase the overall efficiency of the vessel. This would optimise fuel consumption and emissions. The databases developed for this project follow STEP standard platform with EXPRESS data schema that follows ISO10303-11, 2004 standard. Database is based on object-oriented programming style of database and includes extra GUI applications in order to facilitate any manual manipulation and query of data records.

Both databases can handle static (i.e. ship structural components) and dynamic (i.e. Sensorial data) types of data which can also include images. In future, the overall database will be put together as one single unique central stochastic database that can store both structural and machinery data. Subsequently, overall data base will be tested on three major ship types of Tanker, Container ship and Bulk carrier in order to test its performance. Finally, further collaboration between partner classification societies(i.e. Lloyds' Register (LR), Bureau Veritas (BV) and Registero Italiano Navale (RINA)) in project will facilitate the introduction and approval of Machinery Condition Monitoring (MCM) protocol.

ACKNOWLEDGEMENTS

Work described in this paper has been supported with funding from the 7th Framework Programme of the European Commission under Grant Agreement FP7-SST.2013.4-2, CP-605200 (INCASS - Inspection Capabilities for Enhanced Ship Safety). It reflects only the author's views and European Union is not liable for any use that may be made of the information contained therein.

REFERENCES

Biron, P.V., et. Al. (eds.), 2004, “XML Schema Part 2: Data types Second Edition”, *W3C Recommendation 28 October 2004*, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/ (Accessed 11/20/2014).

Celko, J. (2014). Chapter 13 – Hierarchical and Network Database Systems. In Joe Celko’s Complete Guide to NoSQL. *What Every SQL Professional Needs to Know about Non-Relational Databases*. Morgan Kaufmann.pp. 185-201.

Faircloth, J. (2014). Chapter 4 – Databases. In Enterprise Applications Administration. *The Definitive Guide to Implementation and Operations.* Morgan Kaufmann. pp. 131-173.

INCASS (2014a) Deliverable D5.1 Structural Database. INCASS - Inspection Capabilities for Enhanced Ship Safety. *EC FP7 Project.*

INCASS (2014b) Deliverable D5.2 Machinery and equipment Database. INCASS - Inspection Capabilities for Enhanced Ship Safety. *EC FP7 Project*.

INCASS (2014c) Deliverable D5.3 Database: Structural and Machinery Interface. INCASS - Inspection Capabilities for Enhanced Ship Safety. *EC FP7 Project.*

INCASS (2014b) Deliverable D7.1 Specifications of individual case studies for each ship types. INCASS - Inspection Capabilities for Enhanced Ship Safety. *EC FP7 Project.*

Neubauer, P. (2010). Graph Databases, NOSQL and Neo4j. Retrieved 12 10, 2014, from InfoQ.

OBDMS. (2014.). *Operational Database Management Systems*: http://www.odbms.org/free-downloads-and-links/object-oriented-programming/ (Accessed 11/20/2014).

O'Brian, J. A., & Marakas, G. M. (2009). Management Information Systems. *Intelligent Data Engineering and Automated Learning.* Burgos, Spain: Springer. pp. 578-585.

Sherman, M. (2014). Your Database Is So Retro: Old Data, New Databases. *Retrieved 12 10, 2014, from Texas Enterprise*: http://www.texasenterprise.utexas.edu/2014/08/14/innovation/your-database-so-retro-old-data-new-databases (Accessed 11/20/2014).

Sol, S. (2003). The UK Web Design Company. Retrieved 5 29, 2014, from Types of Databases: http://www.theukwebdesigncompany.com/articles/types-of-databases.php (Accessed 11/20/2014)

Taheri, A., Lazakis, I. & Koch, T. (2015). An Innovative Machinery Data Management Systems for Ships using a Catalogue Data Model. *International Maritime Association of the Mediterranean (IMAM) 2015.* Pula, Croatia.

Thompson, H. S., et. al. (eds.), 2004, “XML Schema Part 1: Structures Second Edition”, *W3C Recommendation 28 October 2004*, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/ (Accessed 11/20/2014).

Xu, Y., Guan, J., and Ishikawa, Y. (2013). Scalable Top-k Keyword Search in Relational Databases, *Database Systems for Advanced Applications*. Vol 7239 (pp. 65-80).