

A WEB3D ENABLED INFORMATION INTEGRATION FRAMEWORK FOR FACILITY MANAGEMENT

Edvinas Rasys, Michael Hodds, Nashwan Dawood & Mohamad Kassem
Centre for Construction Innovation & Research, Teesside University, UK

ABSTRACT: *Managing capital oil and gas and civil engineering facilities requires a large amount of heterogeneous information that is generated by different project stakeholders across the facility lifecycle phases and stored in various databases and technical documents. The amount of information reaches its peak during the commissioning and handover phases when the project is handed over to the operator. The operational phase of facilities spans multiple decades and the way facilities are used and maintained have a huge impact on costs, environment, productivity, and health and safety. Thus, the client and the operator bear most of the additional costs associated with incomplete, incorrect or not immediately usable information.*

Web applications can provide a quick and convenient access to information regardless of user location. However, the integration and delivery of engineering information, including 3D content, over the Web is still at its infancy and is affected by numerous technical (i.e. data and tools) and procedural (i.e. process and people) challenges. This paper addresses the technical issues and proposes a WEB3D-enabled information integration framework that delivers engineering information together with 3D content without any plug-ins. In the proposed framework, a class library defines the engineering data requirements and a semi-structured database provides means to integrate heterogeneous technical asset information. This framework also enables separating the 3D model content into fragments, storing them together with the digital assets and delivering to the client browser on demand. Such framework partially alleviates the current limitations of the JavaScript based 3D content delivery such as application speed and latency. Hence, the proposed framework is particularly valuable to petroleum and civil engineering companies working with large amounts of data.

KEYWORDS: *Information integration, Facility Management, Class Library, Web3D, WebGL*

1. INTRODUCTION

Information delivers a significant business value to organizations in the oil and gas sector (Hawtin & Lecore, 2011). Oil and gas engineering sector covers multiple disciplines and includes both chemical process engineering (designing and applying the equipment for processing the liquids and gasses in the oil fields) and civil engineering (designing and building the infrastructure and the facilities). As a result, there are overlapping and shared challenges between the disciplines, and the development outlined in this paper could be transposed between the disciplines and any other sector with similar business processes and data modeling approaches.

This paper proposes an information integration framework developed as part of a larger industrial research project that aims to improve the information management and delivery across asset lifecycle phases (e.g. design, planning, operations, maintenance and decommissioning). The proposed information integration framework aims to form a single and homogeneous source of accessible and trusted engineering data that can be accessed via the Web from personal computers and mobile devices or integrated with other systems.

Challenges affecting the information management in asset capital projects were identified in the literature and discussed in semi-structured interviews with industry professionals from several large companies, involved in managing oil and gas capital assets. The following sections will present the findings from the literature review, industry interviews and current progress in the development of the proposed information integration framework.

2. LITERATURE REVIEW

Engineering information can be generated in the form of 2D and 3D drawings, process diagrams, technical datasheets, user manuals, etc. by diverse disciplines and applications. As a result data models often suffer from a

Citation: Rasys, E., Hodds, M., Dawood, N. & Kassem, M. (2013). A Web3D enabled information integration framework for facility management. In: N. Dawood and M. Kassem (Eds.), Proceedings of the 13th International Conference on Construction Applications of Virtual Reality, 30-31 October 2013, London, UK.

lack of a well-structured, standardized information representation (Wiesner, Morbach & Marquardt 2011; Bayer & Marquardt 2004). The ISO15926 standard has been created to address the issues related to different and proprietary import/export schemas (POSC Caesar 2011; Leal 2005). However, due to the complexity and slow acceptance of the ISO15926 standard, software solutions remain fragmented (Smith 2006). An earlier study executed by National Institute of Standards and Technology (NIST) (Gallaher & Chapman 2004) suggested that the capital facilities industry is losing billions of dollars because of the poor software interoperability.

Existing information integration architectures have been described in the literature (Zhang, Li, Zhang, *et al.* 2011; Haas, Miller, Kossmann, *et al.* 2010; Lenzerini, Vassiliou, Vassiliadis, *et al.* 2003):

- Federated architecture uses a mediator engine to present a virtual view of the distributed, heterogeneous data sources as if it was a single dataset. They transform the application queries, retrieve data from connected online services, combine the results in real time and present to the end user or the client application. In such systems the data services have to be available online at the time of query and the speed of the application depends on the complexity of the integration algorithms and the network access.
- Centralized architecture utilizes extract, transform, load (ETL) engines for importing, mapping and cleansing the heterogeneous datasets into a single data storage area. The process can be initiated from the central database (pull action, suitable for online services) or by a client calling a central database service (push action, suitable for loading offline files into the system).

Some sources also mention a third type of architecture – tiered data warehouse (Lenzerini, Vassiliou, Vassiliadis, *et al.* 2003), which aggregates data into a central database, but also stores the copies or summaries of the data at different tiers. There are engines which can combine the federation and centralization approaches and provide both capabilities (Zhang, Li, Zhang, *et al.* 2011; Fagin, Haas & Hern 2009). In such cases the system is populated first, importing the legacy data that still has value, and then additional online data sources are connected into the virtual data views.

Regardless of the approach, fundamental data or schema conflicts can occur when combining data from heterogeneous data sources (Ram & Park 2004). Data conflicts can occur because of the different units of measure, different interpretation of the values or different reliability of the values. Schema conflict encompasses identifier or naming mismatch, different attributes used to define the same data and other structural inconsistencies occurring from the different schema design. In such cases ontologies are used for resolving the schema conflicts and mapping the different project schemas (Wache, Vögele, Visser, *et al.* 2001; Batres, West, Leal, *et al.* 2007).

Besides the general interoperability issue common for all engineering disciplines, oil and gas projects have one specific challenge. Because of the necessity to shorten the project duration a lot of overlapping, concurrent engineering is being done (Wiesner, Morbach & Marquardt 2011). This causes the teams working on detailed design rely on preliminary data, which can substantially change at the beginning of the project causing rework and even project delays. A good change management system is required to control the impact of the changes, also making sure that all required project collaborators are informed when they occur.

Information accessibility and speed is another issue for an information integration framework. It is common for the project contractor in the AEC and Oil & Gas industries to share information and responsibilities with the owner and subcontractors (Schramm, Meißner & Weidinger 2010) while the project collaborators can often be geographically dispersed or located in remote areas (Reece, Hoefner, Seetharam, *et al.* 2008). Engineering web portals provide a single secure access to the project data and provide significant productivity gains due to increased collaboration (Samdani and Till, 2007).

Application speed, reflected by the time spent searching for or retrieving information, impacts the users decisions and productivity (Burda, Crompton, Sardoff, *et al.* 2007), however big enterprises are often reluctant to change the existing systems due to the project requirements (Stonebraker 2010). This is often the case with oil and gas companies, which value reliability and proven track record (Crompton & Gilman 2011). While most sources point to online transaction processing (OLTP) systems requiring the strict SQL (structured query language) data model, the different requirements for data warehouses are often ignored. Studies show that analyzing the requirements and choosing a storage engine optimized for a particular task can increase the performance of an application up to several orders of magnitude compared to a traditional relational database system (RDBMS) (Stonebraker & Çetintemel 2005; Stonebraker, Bear, Çetintemel, *et al.* 2007). There seems to be a mismatch between the increasing need for application speed, accessibility and flexibility, and the commitment to the old technologies, which often limits the performance of the overall solution.

3. INDUSTRY INTERVIEWS

A total of five expert interviews with project consultants and data managers working in large oil and gas projects for BP, ConocoPhillips and Total have been conducted. The interviewees were guided by the main questions; however they were free to elaborate on data management issues present in their projects. The interviews were aimed at confirming and clarifying the information management issues found in the literature. They also served the purpose of defining the guidelines for the required features of the information integration framework. The results from the interviews and expert opinions are summarized below.

The major questions that had been asked aimed to:

- Identify the information types and formats used when exchanging data between project stakeholders;
- Clarify the current issues present in information management processes;
- Define the industry expectations from a new generation of information management systems;
- Explain the use cases and the need for integrated 3D model functionality.

The results confirmed that issues regarding the poor interoperability of information management systems are still present in current oil and gas capital asset projects. There is no industry standard information exchange format – data is being transferred via various text based files also including Excel files, Access databases and project design documents. Very rarely applications are able to exchange information automatically, often email systems are used to drive the business workflows and exchange the information between the project collaborators.

With up to 500 companies and 1500 people collaborating on a single project, web portals can be an invaluable information resource providing each of the stakeholders with an access to the same data interfaces. Accessibility can often become an issue for mobile staff, such as engineers who operate far from their offices – from hotels or on site. Current remote access methods such as VPN connection are subject to strict security measures and depend on very good network access. Thus a lightweight web application could often result in higher productivity in remote areas.

Well defined workflows efficiently drive the processes in software systems. The interaction between people, processes and software is a complex relationship. The resistance to change could be exacerbated by poor software interfaces – “if the application feels sluggish, people start to lose interest”, noted one information manager from BP. There are also different cultures and different customer practices around generating engineering data, which make their integration a human issue as much as it is a technical issue. If people feel constrained by poor workflows or slow software, they tend to create their own data stores (e.g. spread sheets) because they are faster or more convenient to work with. In such cases people communicate the changes through emails, but managing the changes is very difficult in such a decentralized system. Once the original email have been forwarded, it becomes nearly impossible to check who has seen what information, and until the document has been submitted to the management system it is impossible for other collaborators to know of any changes that are occurring. Managing the changes in a timely manner is crucial. If missing or incorrect information is detected later in the project, it might be very costly and time consuming to replace that information as the engineers responsible may have already left the project.

The requirements for the 3D model integration are still maturing as not all companies have the 3D model representation as a requirement in the early project stages. It is expected however that the 3D model of the facility would be used in (remote) operations, maintenance planning and safety training, especially if the facility has not been yet completed or is difficult to access. It is worth noting that due to the file sizes, broadband limits and the different needs of various people working on the 3D model, there is a requirement to have two 3D model representations: a review link for a quick review of the model, and a download link, where engineers can access and download the actual model when there is a need for corrections.

It is clear from the above that a wide range of aspects relating to business processes, communication between people and software interoperability needs to be considered and they cannot all be addressed in this paper. In this paper, we focus on the integration and delivery of information and 3D content over the Web by proposing a framework and developing a proof-of-concept tool. The proposed framework and obtained results can be utilised by field researchers to advance the research and development in this key area and by industry developers to improve their systems.

4. INFORMATION INTEGRATION FRAMEWORK

4.1 Architecture overview

In order to achieve the aim of a single homogeneous data source and considering the results of the literature review and the interviews, objectives for the information integration framework have been defined. The integration system needs:

- To be web based and provide web service and client interfaces;
- Utilize an efficient storage layer for storing and retrieving different types of engineering data;
- To be driven by an ontology, which is called Class Library in the field of oil and gas projects;
- Has audit and change management capabilities;
- Integrates a current Web3D technology to deliver the 3D content to the client browser.

The framework proposed in this paper adopts a centralized data storage approach for the heterogeneous engineering information (Figure 1). The object data model in oil & gas sector has its own distinguishing requirements. Each object (called a *tagged item*) has a unique identifier within a facility, a name or description, type of the object (class), a set of possible attributes and associations. Handle identifiers (CNRI 2013) are used to make the objects globally identifiable and resolvable. While part of the data schema is fairly static, the number and values of the attributes can vary greatly, as well as the possible associations to the other tagged items or documents.

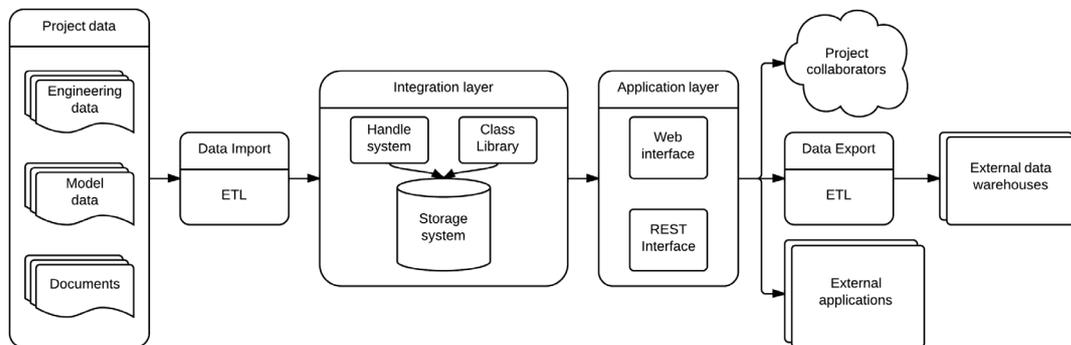


Fig. 1: Integration framework diagram

Data requirements are defined in the ontology called Class library. The role of the Class Library is to define the attribute data that is expected for a given object. Classification of the objects can be carried out by the Class Library through the use of class name mappings (to interpret given object classification into known classes in the Class Library), or through the use of an Engineering Numbering Specification (ENS), which interprets object classification based on the identifier codes. Class Library also provides validation rules for attribute values, such as accepted lookup values (typically for attributes with plain text values), or accepted units of measure for attribute values (typically for attributes with numeric values).

ETL processes use the predefined data requirements and mappings and import data from various data sources into the central data store. The web interface can then be used to search and browse the information. A representational state transfer (REST) based interface makes data available to applications able to consume this type of service. Such framework requires an efficient data model and data storage system and a semi-structured data model has been chosen for this task.

4.2 Semi-structured data storage

Document or semi-structured data models are used to describe data, which does not have a strict schema or it evolves over time (Papakonstantinou, Garcia-Molina & Widom 1995). Current popular formats for the semi-structured data are JavaScript Object Notation (JSON) and Extensible Markup Language (XML). Both of these formats are self-describing, but JSON is more compact and querying JSON objects does not need a separate language to access the attributes.

JSON has been chosen to store the data model in the database as it suits the domain model better than the key-value or relational database systems (Rasys & Dawood 2012). Such storage makes the planning and development of the database schema easier as it closely resembles the human readable representation of the data as shown in Table 1.

Variable number of attributes enables iterative system development, which can be executed using the agile project management (Larman 2004). The system can be started small and provide basic functionality initially; over time additional functions can be added as the database layer can be easily extended without taking the existing application offline.

With client JavaScript code already present in the majority of interactive web applications, usage of client side JavaScript libraries like Knockout.js (Sanderson 2013) simplifies the development of rich client interfaces. Having this format at the storage layer, combined with a JavaScript based application server such as Node.js (Joyent 2013), means that JSON format could be the only form of information needed in the whole path from the database to the client browser. This saves significant development time as data can be extracted from the database, manipulated and passed to the client using the same programming language constructs.

Table 1: Human readable and JSON representation of the data compared.

Human readable data	JavaScript Object Notation (JSON)
Pump (P-101)	{ "ID": " P-101",
Class: Pump	"Name": " Pump P-101",
ID: P-101	"Description": "Cold water pump, floor 2",
Description: Cold water pump, floor 2	"Class": "Pump",
Pressure Max: 80 psi	"Attributes": [
Type: electric	{ "Name": "Pressure Max", "Value": 80, "UOM": "psi"},
Voltage: 380 V	{ "Name": "Type", "Value": "electric"},
...	{ "Name": "Voltage", "Value": 380, "UOM": "V"}, ...]

As JSON data format seems suitable for the database layer, several JSON based databases were considered and one called MongoDB (10gen Inc., 2012) was chosen as the storage engine. MongoDB was selected over other JSON databases because it supports multiple indexes and has a query language similar to SQL. It is designed to be scalable and features like GridFS (for storing large files) and geospatial indexing are additional features worth exploring when developing 3D integration. MongoDB is actively maintained and developed, while clients choosing this database engine can get commercial support, which is rarely possible with other free solutions.

Performance benchmarks have been done on Amazon’s AWS cloud infrastructure - EC2 large machine instances, each with 2 cores and 7.5GB of memory. The scripts were run on the Ubuntu 11.10 64bit operating system. The database versions available at the time were 2.0.2 for MongoDB and 5.1.58 for MySQL. Every MongoDB collection and MySQL table had two indexes – one for the primary key and one secondary index.

Test scenarios were created to emulate a large data process job for importing *tagged items* and their *attributes* into the system with a huge number of write and/or read operations taking place sequentially. For these tests it was assumed that each *tagged item* has 100 *attributes*. As it is a prototype data store model test, the performance metrics should not be treated as absolute values, but rather as relative performance indicators.

To benchmark the performance of the database storage layer, write and read operation tests were developed (Rasys & Dawood 2012). For write operations MongoDB is consistently faster than MySQL implementation even as the number of records in the databases increases (Figure 2). Read operation performance (Figure 3) shows a different picture as it only takes one operation for MongoDB to retrieve the *tagged item* along with its *attributes*. MySQL on the other hand needs to join two tables together and this starts to become an issue once the database grows to 10 million *tagged item* records. From that point the performance of MySQL decreased a lot quicker than the MongoDB database, which showed a nearly linear scalability.

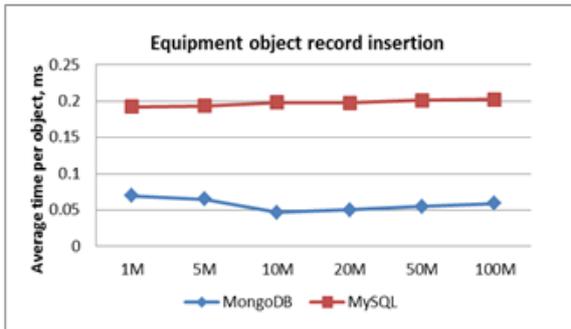


Fig. 2: Write Operations

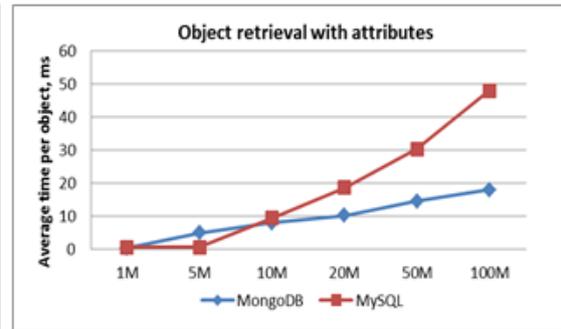


Fig. 3: Read Operations

The performance benchmarks indicate that JSON is a feasible way for storing project data in the database, and that MongoDB is faster than MySQL when the amount of database records is large. Other studies have also shown the superiority of MongoDB in their particular domains (Boicea, Radulescu & Agapin 2012; Jokić, Krčo, Dejanović, *et al.* 2012). As a result, MongoDB has been chosen as the database storage engine for the information integration framework.

4.3 Implementation of the web interfaces

4.3.1 REST interface integration

Some of the issues identified in the interview of industry experts concerned the linking relationships between the equipment items and the design documents. In this paper we demonstrate how a two-way communication between the 3D model components and additional engineering information, received from other stakeholders, can be achieved. Clicking the equipment item in the component list should highlight/display the selected object and vice versa – selecting an item in the 3D model should bring up the data, associated data with that equipment. The usual approach in the design review applications for showing a component of interest is to load up the whole model and then hide the unnecessary elements. For this type of usage, a REST web service and a sample plug-in for NavisWorks Manage has been developed (Figure 4). It allows retrieving associated data via REST interface and displaying it in the application plugin pane or opening the web portal in a separate browser window. The implication of this functionality is that the engineers who are conducting design review are no longer disconnected from the rest of the domain data.

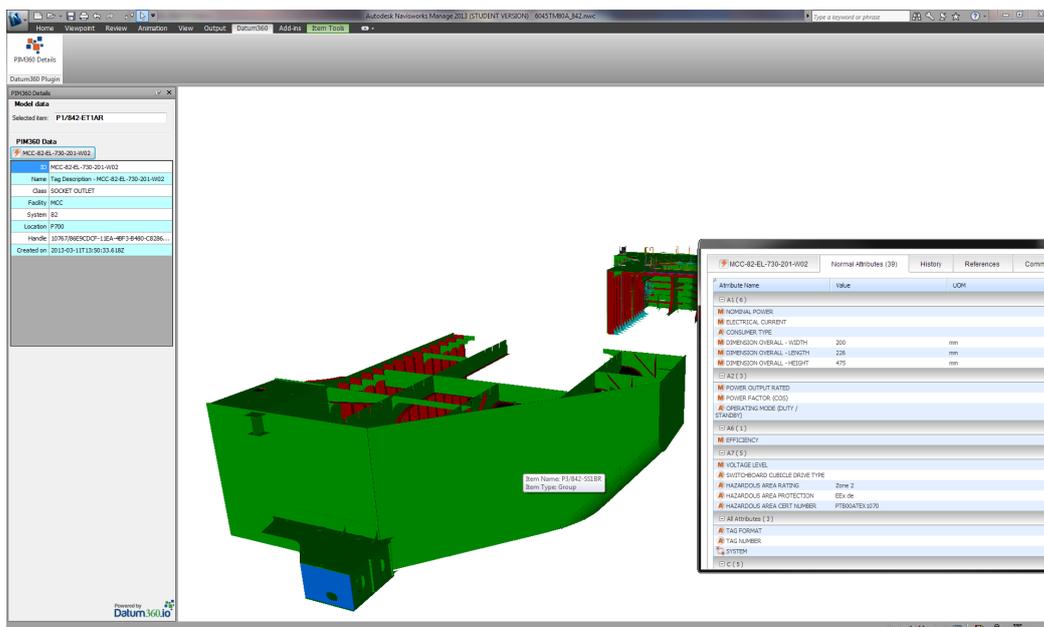


Fig. 4: NavisWorks Manage plug-in links 3D content with the extended engineering information

However, as some of the capital asset models can be very large, loading the full model is not a practical use of processing resources and bandwidth. While a multi-story building model can be sliced by floors, a complex FPSO (Floating Production, Storage and Offloading unit) or sea platform is much more challenging and complex to separate into areas or systems. Also a design review application may not be a viable option for the client operating systems (Linux, Mac OS, tablets). A web-based approach would make the 3D content much more available and accessible. It should also enable a more efficient method of loading only the component(s) of interest for each stakeholder. Such approach would make 3D visualization possible even on portable devices with low power 3D graphics hardware. To implement such a feature the 3D model elements have to be extracted, stored in a predefined structure and mapped to the individual *tagged items* stored in the database. These operations are discussed in the following paragraph.

4.3.2 Proposed Web 3D integration

A lightweight delivery of 3D content over the Web has been identified as one of the key requirements for enhancing the accessibility to engineering information. For a Web3D based model presentation multiple 3D frameworks have been considered. Indeed, there are multiple ways to display 3D content inside web browsers. Many of them are implemented using browser plugins (Behr, Eschler, Jung, *et al.* 2009; Wright & Madey 2008), which create a separate virtual environment within a browser, isolate the Document Object Model (DOM) tree, events and Cascading Style Sheets (CSS) features (Sons, Klein, Rubinstein, *et al.* 2010). WebGL is emerging as one of the most popular base frameworks for the 3D content presentation (Ortiz Jr., 2010), and is supported natively on some browsers (i.e. Chrome) and with certain requirements/exceptions on others (i.e. Firefox, Opera, Safari).

Microsoft Internet Explorer browsers do not support WebGL natively because of the security flaws in the previous WebGL implementations, identified by Context Information security in their report (Forshaw 2011; Microsoft 2011). Such flaws allow malicious code to be run, which either renders the machine unusable or puts “users’ data, privacy and security at risk”. Browser vendors have responded to the report with various means – blacklisting the video cards and drivers with known security issues, restricting the use of the WebGL textures as per cross-domain (CORS) policy (Khronos Group 2011). Such measures and the increasing popularity of WebGL applications forced Microsoft to reconsider their approach and the information on Windows Blue and Internet Explorer 11 indicates that Microsoft intends to add support for WebGL (Thurrott 2013).

WebGL is a low level Application Programming Interface (API), a JavaScript wrapper over OpenGL 2.0 ES functions. Its specification does not define any file formats and therefore it is not designed to natively support any 3D file standards. Web applications utilizing WebGL often define their own data structures and the lack of universal 3D format support makes the data exchange between the design and client applications difficult.

Numerous libraries that use WebGL as their rendering engine have been created. They abstract the low level functions into API calls, which deal with vertices, objects and scene graphs. While some of the libraries try to tie the existing standards, e.g., X3D (Behr, Eschler, Jung, *et al.* 2009), others deviate from that and focus on making the most of the underlying platform and the rendering engine. Several library implementations have data converters, which can convert 3D content from the more popular standards (IFC, FBX) to the internal structures that the libraries use. SceneJS is one of such libraries and it creates a JSON scene graph, which can be easily manipulated by a JavaScript application engine. It employs numerous optimizations to increase the rendering performance (SceneJS group, 2012). An open source BIM Surfer project, which uses SceneJS as the visualization engine, already exists (BIM network, 2011), but there are no known design and CAD applications that would produce SceneJS JSON file as the export format. However, an open source BIMserver project (BIM network, 2012) can be used as a transformation step. This allows taking a model from a design application (i.e. REVIT), exporting it to IFC format, importing it into BIM server and exporting it as the SceneJS JSON format, which can be used by an application using the SceneJS library.

Three.js is another high level JavaScript library. It is fast, object oriented, supports math operations and has an extensive support for the file formats in addition to its own JSON and binary formats (three.js 2013; Parisi 2012). Choosing between the libraries is not an easy task as they evolve very quickly. Initially SceneJS was chosen as the library for the framework, but it has been observed that some of the model data is being lost during the transformation through the BIM Server. While BIM Server is an open source project, it does not seem to have a large community and the support for the additional file formats seems difficult to implement. SceneJS project itself seems to have stalled as the latest available version is more than a year old. After considering these issues, three.js has been selected as the intermediate 3D framework.

Three.js has a native JSON file format support and content can be imported from engineering design review applications (e.g. NavisWorks) via FBX file converter. Since the output native JSON file is just a plain text file, it is trivial to write an extraction/mapping routine, associating the 3D coordinates, material, lighting, rotation of a component with the *tagged item* stored in the database. As the same *tagged item* can be present in multiple models, the following mapping schema has been proposed to link the 3D data with the *tagged item* identifiers:

```
"hdl" : "10797/a9a30061-76b0-4c15-8f2a-6efa74371612"
"src" : "FPSO 1"
"id" : "FLANGE 1 of BRANCH /I70-PIPE-TEST/B1"
"geometry" : {},
"material" : "",
"position" : [ 0, 0, 0 ],
"rotation" : [ 0, -0, 0 ],
"scale" : [ 1, 1, 1 ],
"visible" : true
```

The first property (hdl), which denotes the unique object identifier (handle), identifies the object within the integration database. The next two properties (src, id) represent the drawing or model and the identifier within that model. The geometry property needs to have the 3D data in accordance with the three.js data specification. The combination of these properties allows linking geometric shapes from different drawings or models to a single set of engineering tagged items. Such linking enables the delivery of the tagged item information alongside its context specific 3D representation to a web browser (Figure 5):

Possible usage scenarios include getting the data in a single discipline for a particular subsystem in a particular area. Querying can also be done on the attributes so it becomes possible to obtain the equipment from a particular

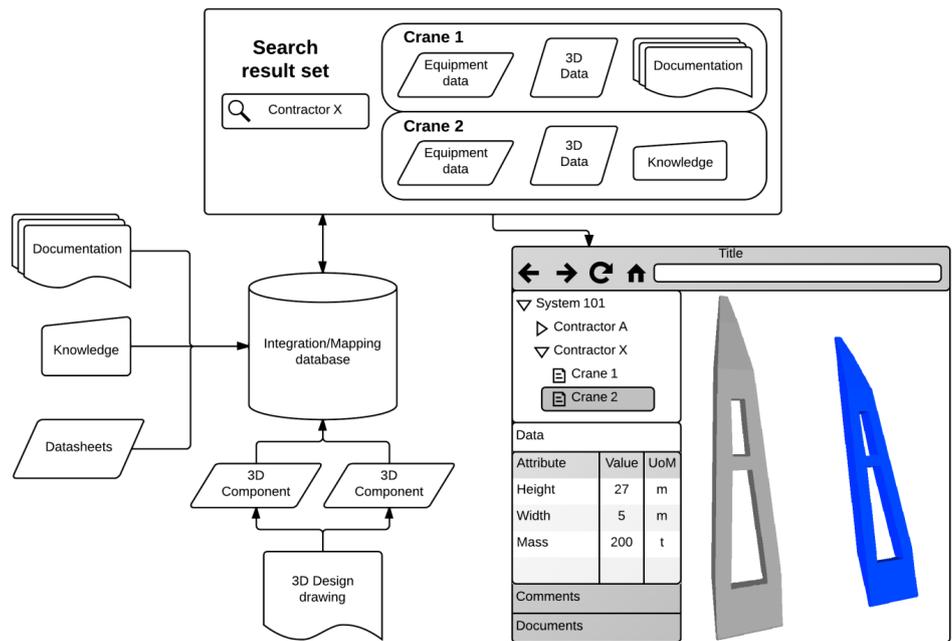


Fig. 5: Dynamic assembling of the scene, based on a query

supplier or the equipment for certain flow rates and display that information together in a web browser window. This proposed solution, while still under development, is promising in terms of its ability to present the slices of information relevant to each particular user.

5. CONCLUSIONS AND FUTURE WORK

This research identified the issues and challenges affecting the management of engineering information in the oil and gas engineering field using semi-structured interviews with industry experts from large oil and gas companies. To address some of those challenges, in particular those related to the heterogeneity of information and the speed and access to information, an information integration framework was proposed. The framework was prototyped in

a proof-of-concept tool that proved to be capable of delivering integrated engineering information to client machines from web services and therefore providing a remote access to this engineering information. By employing a semi-structured storage engine, the framework also guaranteed the query speed and scalability to accommodate large data sets of capital asset projects. Furthermore, this research presented the concept and discussed the feasibility of the web interface that can make engineering information available for both external applications and project collaborators directly via their web browsers. The proposed features of the concept enable the selective displaying of 3D model elements along with the corresponding data from other applications. As a result, this proposed framework can be considered as one step closer to a single trusted and 3D Web accessible source of project facility data.

The additional engineering data and the visualization of 3D model views, whether using a REST data interface or a full web portal, is expected to have a positive impact on the client productivity given the business value of managing information and data identified in both literature and expert interviews. The proposed framework is particularly beneficial for organizations involved in the oil and gas sector, dealing with large quantities of information that is spread over multiple systems. Future work will involve developing additional features to enhance the business workflows and user experience. Examples of such features are: the subscription to changes to better coordinate the collaboration between engineers; the enhanced audit and management of change capabilities, and the augmentation of visualization with heat maps or color coding to identify 3D components that have missing or inconsistent information.

6. REFERENCES

- 10gen Inc. (2012) *MongoDB*. [Online]. 2012. Available at: <http://www.mongodb.org/> (Accessed: 15 July 2012).
- Batres, R., West, M., Leal, D. and Price, D. (2007) *An Upper Ontology based on ISO 15926*. 31519–534.
- Bayer, B. and Marquardt, W. (2004) Towards integrated information models for data and documents. *Computers & Chemical Engineering*, Vol. 28, No. 8, 1249–1266.
- Behr, J., Eschler, P., Jung, Y. and Zollner, M. (2009) X3DOM - A DOM-based HTML5/ X3D integration model. In: *Proceedings of Web3D 2009 The 14th International Conference on Web3D Technology*. 2009 Association for Computing Machinery, 127–135.
- BIM network (2011) *BIM Surfer*. [Online]. 2011. Available at: <http://bimsurfer.org/> (Accessed: 15 July 2012).
- BIM network (2012) *BIMserver*. [Online]. 2012. Available at: <http://bimserver.org/> (Accessed: 15 July 2012).
- Boicea, A., Radulescu, F. and Agapin, L.I. (2012) MongoDB vs Oracle -- Database Comparison. *2012 Third International Conference on Emerging Intelligent Data and Web Technologies*, 330–335.
- Burda, B., Crompton, J., Sardoff, H., Falconer, J., et al. (2007) *Information Architecture Strategy for the Digital Oil Field*.
- Corporation for National Research Initiatives (CNRI) (2013) *Handle System*. [Online]. 2013. Available at: <http://handle.net/> (Accessed: 18 May 2013).
- Crompton, J. and Gilman, H. (2011) The Future of Integrated Operations. *SPE Economics & Management*, Vol. 3, No. 1, 45–51.
- Fagin, R., Haas, L.M. and Hern, M. (2009) *Clio : Schema Mapping Creation and Data Exchange*, 198–236.
- Forshaw, J. (2011) *WebGL - A New Dimension for Browser Exploitation*. [Online]. 2011. Available at: <http://www.contextis.co.uk/research/blog/webgl-new-dimension-browser-exploitation/> (Accessed: 19 May 2013).
- Gallaher, M.P. and Chapman, R.E. (2004) *Cost analysis of inadequate interoperability in the US capital facilities industry*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
- Haas, L.M., Miller, R.J., Kossmann, D. and Hentschel, M. (2010) A first step towards integration independence. *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)*, 147–150.

- Hawtin, S. and Lecore, D. (2011). The business value case for data management - a study. [Online]. Available at: <http://www.oilandgasuk.co.uk/cmsfiles/modules/publications/pdfs/OP055.pdf> (Accessed: 15 July 2013).
- Jokić, S., Krčo, S., Dejanović, I., Vučković, J., *et al.* (2012) Evaluation of a Document Oriented Resource Directory Performance. *Telfor Journal*, Vol. 4, No. 2.
- Joyent (2013) *node.js*. [Online]. 2013. Available at: <http://nodejs.org/> (Accessed: 18 May 2013).
- Khronos Group (2011) *WebGL Security*. [Online]. 2011. Available at: <http://www.khronos.org/webgl/security/> (Accessed: 19 May 2013).
- Larman, C. (2004) *Agile and iterative development: a manager's guide*. Addison-Wesley Professional.
- Leal, D. (2005) ISO 15926 "Life Cycle Data for Process Plant": An Overview. *Oil & Gas Science and Technology - Rev IFP*, Vol. 60, No. 4, 629–637.
- Lenzerini, M., Vassiliou, Y., Vassiliadis, P. and Jarke, M. (2003) *Fundamentals of data warehouses*. Springer.
- Microsoft (2011) *WebGL Considered Harmful - Security Research & Defense - Site Home - TechNet Blogs*. [Online]. 2011. Available at: <http://blogs.technet.com/b/srd/archive/2011/06/16/webgl-considered-harmful.aspx> (Accessed: 19 May 2013).
- Ortiz Jr., S. (2010) Is 3D Finally Ready for the Web? *Computer*, Vol. 43, No. 1, 14–16.
- Papakonstantinou, Y., Garcia-Molina, H. and Widom, J. (1995) Object exchange across heterogeneous information sources Philip S Yu and Arbee L P Chen (eds.). *Proceedings of the Eleventh International Conference on Data Engineering*, Vol. 67, No. 5, 251–260.
- Parisi, T. (2012) *WebGL: Up and Running*. O'Reilly Media, Incorporated.
- POSC Caesar (2011) *ISO15926 Primer*. [Online]. 2011. Available at: https://www.posccaesar.org/wiki/ISO15926Primer_Parts (Accessed: 11 May 2013).
- Ram, S. and Park, J. (2004) Semantic Conflict Resolution Ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts. *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 16, No. 2, 189–202.
- Rasys, E. and Dawood, N.N. (2012) Semi-Structured Data Modelling for a Web-Enabled Engineering Application. In: *Computing in Civil Engineering (2012)*. June 2012 American Society of Civil Engineers, 41–48.
- Reece, C.A., Hoefner, M.L., Seetharam, R. V and Killian, K.E. (2008) *Business Subsurface*.
- Samdani, K. and Till, A. (2007) Engineering portals add significant value to E&P project delivery capabilities. *World Oil*, Vol. 228, No. 11, 91-95.
- Sanderson, S. (2013) *Knockout : Home*. [Online]. 2013. Available at: <http://knockoutjs.com/> (Accessed: 18 May 2013).
- SceneJS group (2012) *3D Scene Graph Engine for WebGL*. [Online]. 2012. Available at: <http://scenejs.org/> (Accessed: 15 July 2012).
- Schramm, C., Meißner, A. and Weidinger, G. (2010) Contracting strategies in the oil and gas industry. *3R International*, Vol. 1, 33–36.
- Smith, B. (2006) *Against Idiosyncrasy in Ontology Development*.
- Sons, K., Klein, F., Rubinstein, D., Byelozorov, S., *et al.* (2010) XML3D: interactive 3D graphics for the web. In: Philipp Slusallek, Byoungyun Yoo, and Nicholas Polys (eds.). *Web3D 10 Proceedings of the 15th International Conference on Web 3D Technology*, 175–184. 2010 ACM.
- Stonebraker, M. (2010) *Why Enterprises Are Uninterested in NoSQL*. [Online]. 2010. Available at: <http://cacm.acm.org/blogs/blog-cacm/99512-why-enterprises-are-uninterested-in-nosql/fulltext> (Accessed: 21 July 2013).

Stonebraker, M., Bear, C., Çetintemel, U., Cherniack, M., *et al.* (2007) *One Size Fits All ? – Part 2 : Benchmarking Results*.

Stonebraker, M. and Çetintemel, U. (2005) “One Size Fits All”: An Idea Whose Time Has Come and Gone. In: *Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*. 2005 IEEE Computer Society, 2–11.

three.js (2013) *three.js - JavaScript 3D library*. [Online]. 2013. Available at: <http://threejs.org/> (Accessed: 19 May 2013).

Thurrott, P. (2013) *Blue's Clues: WebGL Support in IE 11*. [Online]. 2013. Available at: <http://winsupersite.com/windows-8/blue-s-clues-webgl-support-ie-11> (Accessed: 19 May 2013).

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., *et al.* (2001) Ontology-Based Integration of Information — A Survey of Existing Approaches. In: *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*. 2001 Seattle, 108–117.

Wiesner, A., Morbach, J. and Marquardt, W. (2011) Information integration in chemical process engineering based on semantic technologies. *Computers & Chemical Engineering*, Vol. 35, No. 4, 692–708.

Wright, T.E. and Madey, G. (2008) A Survey of Collaborative Virtual Environment Technologies. *University of Notre Dame-USA, Tech. Rep.*

Zhang, S., Li, J., Zhang, F. and Li, X. (2011) Design of Data Center and Implementation of Data Integration Based on SOA Framework, 267–271.