DYNAMIC WEB3D VISUALISATION OF OIL & GAS FACILITY ASSETS[1]

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

ABSTRACT: *Efficient information management in the oil and gas industry is crucial to provide all of the data flows, required to support the business and safe operation of facilities. The cost of creating digital facility asset data sets increases, if it is performed late in the delivery process, as it becomes increasingly difficult and expensive to retrieve missing information. There is a need to develop approaches and tools for the management of such data sets throughout the whole project lifecycle. This includes the harvesting of heterogeneous data from disparate applications, its integration, sharing and use across the project phases.*
This paper is part of a large industrial research project, aimed to create an efficient engineering information integration framework. This framework will create a single, accessible and trusted data source, combining operational equipment data and its 3D representation. Previous papers published by the authors have presented the overall framework and the proposed technology to deliver the operational asset data. This paper focuses on linking that data to the 3D model elements, harvested from design applications, and delivering configurable and dynamic 3D scenes to the client's web browser. Performance benchmarks show the usability of the implemented Web3D prototype, based on WebGL, in a project environment.

KEYWORDS: *Digital facility asset, engineering information integration, Web3D, WebGL*

❖ INTRODUCTION

Information management in oil and gas industry covers both the creation of data sets and enabling data flows that are crucial to supporting the business processes (Hawtin and Lecore, 2011). The cost of creating or replacing missing information in the data set increases as the project progresses. It becomes very difficult and expensive to retrieve or recreate the missing information once the contractors have handed over the facility and associated information to the operation and owner teams (Rasys et al. 2014).
Equipment in an oil and gas facility is part of a very complex and interconnected asset. In addition to this challenge, projects in this industry suffer from information fragmentation. Data is generated by multiple applications, stakeholders and at different project phases. Due to the wide range of disciplines involved and applications used, data models often lack a well-structured and standardized information representation (Wiesner et al. 2011; Bayer and Marquardt, 2004).
Integrating data, including 3D model representation, in an information integration framework can help reduce this complexity. However, equipment data in this sector has a large number of operational attributes. 3D models in design applications are not suitable to store such amount of attributes from various disciplines. Project data warehouses are used to collect and store the equipment data; and 3D models include only equipment identifiers, referencing the model data in other systems.
Accessibility and the speed of access to information are important factors in the information integration framework (Rasys et al. 2014). The information needs to be available to multiple project stakeholders as project contractors in oil and gas industry regularly share information and responsibilities with the owner and subcontractors (Schramm et al. 2010). Project collaborators are often geographically dispersed or located in remote areas (Reece et al. 2008). Engineering web portals are deemed to provide a single immediate access to shared project information, enabling improved collaboration (Samdani and Till, 2007). To provide a single source of trusted engineering information, including 3D data, these web portals require an integration framework to help overcome the key challenges associated with the business processes in this sector:

- **Data mapping**. In greenfield projects - i.e. new oil and gas asset developments (Bell, 2012), which include populating new IT systems (Hopkins and Jenkins, 2008) - various parts of the overall dataset are made available at different phases of the project. In such projects, there is often a significant pressure on project schedules and design tasks are concurrently performed (Wiesner et al. 2011). As a result, 3D models are

[1] Citation: Rasys, E., Dawood, N., Kassem, M. & Scott, D. (2014). Dynamic WEB3D visualization of oil & gas facility assets. In: N. Dawood and S. Alkass (Eds.), Proceedings of the 14th International Conference on Construction Applications of Virtual Reality, 16-18 November 2014, Sharjah, UAE.

rarely available from the start and project collaborators are basing their work on preliminary data, which can change significantly over the course of the project.
In brownfield projects - i.e. existing, mature oil and gas assets, legacy IT infrastructure - data is present in various heterogeneous systems, which are populated by various companies. Collecting and aggregating data from such systems to assemble a single source of information is challenging, as conflicts in the data values, naming schemes and units of measure are common and need to be resolved.

- **3D content extraction**. Content creation applications are not suitable as the basis for information integration solutions because they are usually proprietary applications with closed data models. System developers do not have full control over the data models, which might be subject to changes with every new release. Therefore 3D content extraction into an application neutral format is needed to enable the successive reuse of data by other applications in the integration chain. An open file format, such as the Industry Foundation Classes (IFC) standard (Dibley et al. 2011; Schevers et al. 2007) adopted in the building industry, does not exist in the oil and gas sector.
- **3D content rendering.** Web3D (3D on the Web) technologies have not fully matured yet and web portals have limited graphic capabilities. Also mobile devices have limited network capabilities and their hardware resources (memory, processor and 3D support,) are relatively low. Desktop computers can be powerful and can have suitable network connectivity to provide a good web experience.
WebGL (Parisi, 2012), which allows 3D visualisation in web browsers without the need of plug-ins, is becoming the new Web3D standard. As WebGL is a wrapper API (Application Programming Interface) over low level OpenGL functions, the 3D content extracted from the design applications cannot be directly rendered by a page just using WebGL. Therefore, to display the 3D content over the Web, additional libraries are needed to bridge the gap between the content generation and web applications. Also when considering web applications the data size of the 3D model have to be taken into account, as it can hinder the speed of the application and consequently its acceptance by industry practitioners.

While various research projects have tried to address specific aspects of the above challenges, an overall solution is still absent. In the remainder of this paper, we will present the findings from the review of related studies and propose a Web3D delivery framework and tool, as part of our overall information integration framework.

LITERATURE REVIEW

Integration approaches

In the process industry there have been several research projects addressing the challenge of information integration. Brandt et al. (2008) proposed an ontology based Process Data Warehouse system using knowledge integration. The system is based on KAON server (Oberle et al. 2004) and extended by OntoCAPE ontology (Braunschweig et al. 2004) to model the relationships between entities and enable decision support via semantic interpretation of queries. The authors chose scalability over the descriptive capabilities, available in description-logic based languages (Baader and Nutt, 2003). However the authors accept that the system is not efficient for large datasets, as it needs to load the entire information into memory.
Wiesner et al. (2011) further extended the use of OntoCAPE ontology to create a Comprehensive Information Base system for information and knowledge integration. They demonstrated with a prototype that the framework is suitable for processing industrial information and can offer modelling power via the hybrid ontology model. Their system, however, suffers from similar efficiency drawbacks to those encountered in its earlier version. A medium-sized dataset described in the benchmarks (521 items) took eight hours to process four integration queries. Only when the machine was upgraded to a 64 CPU (Central Processing Unit) one with 128 GB of RAM (Random Access Memory), the four queries took two minutes. Such an approach is not practical for data sources over 50000 items, which is often the norm in oil and gas projects.
Kim et al. (2011) presented an ISO 15926 based data repository (façade) to store equipment data throughout the nuclear power plant lifecycle. The framework prototype also includes the use of semantic web technologies and web services, so that other applications can interact with the project database. The authors suggest that the data sizes start in the gigabyte range in the beginning of the project, which increase to the terabyte range at the end of the project. In this case the selection of MySQL as the database engine is questionable, as the project does not show how the overall system performs with such datasets. Overall none of the proposed systems have a module for integrating and visualising the 3D data of the oil and gas facility.

Web 3D delivery and Building Information Modelling (BIM)

3D data is becoming a common BIM dataset in architectural, engineering and construction industry. The integration and delivery of 3D data have attracted significant attention in recent years. Chuang et al. (2011) recognised that existing proprietary BIM systems (e.g. Autodesk Revit, Bentley Architecture, Tekla Structures, etc.) are based on standalone frameworks and the information is difficult to access from different sites. To overcome this challenge, they proposed a Software as a Service (SaaS) approach using a server-based custom integration on the cloud. The solution incorporated RealityServer - a web service, integrating NVIDIA Tesla GPUs (Graphics Processing Unit) and 3D web services software (migenius, 2013). RealityServer was designed to deliver photorealistic 3D images to devices that do not have the power to render such images. The project chose a browser plug-in based technology - Microsoft Silverlight (Microsoft, 2014b), which enables the development of Rich Internet Applications (RIA). This, however, limits the compatibility of the solution, as it requires a desktop Operating System (OS) and a supported version of a web browser (Microsoft, 2014a).

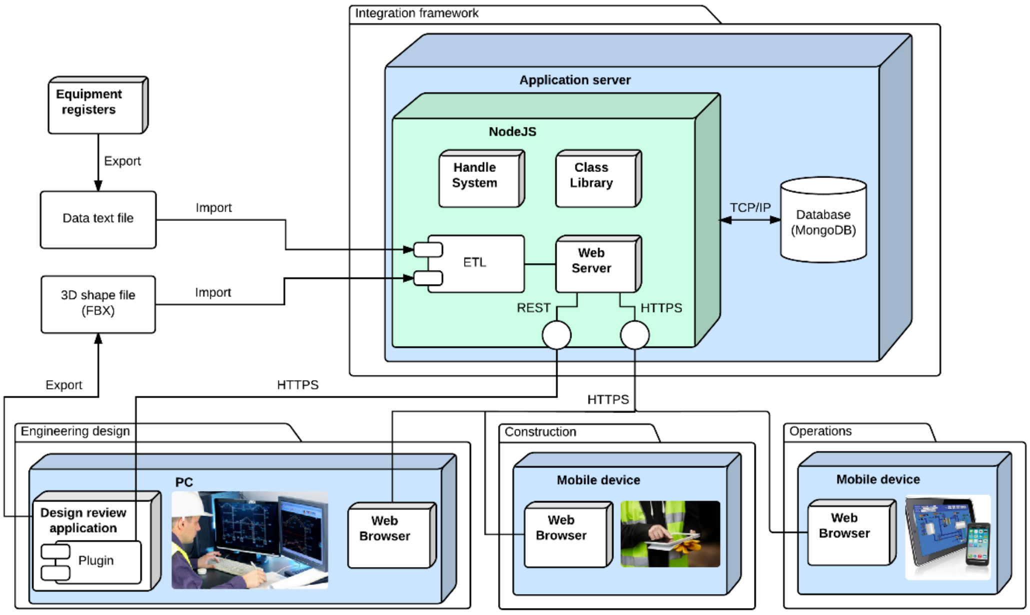
Hagedorn and Döllner (2007) presented a framework for integrating BIM models in the virtual city 3D models (CityGML). This framework assumed that BIM and 3D geodata were made available as web services. IFC was taken as a source format for BIM data and mappings between BIM and CityGML data were added to extend the CityGML model. The implementation of the visualisation was based on the LandXplorer technology – software system that enables presentation, exploration and analysis of geovirtual 3D environments (Autodesk, 2009). The complex 3D rendering was also done on the server side and the end result was then transferred to the client as image. While this approach enables photorealistic rendering of sites, it limits the interactivity as the user is not able to manipulate the scene.

A different approach has been presented by Beetz et al. (2010) in their BIMserver system. The IFC based framework was using an EXPRESS parser (Lubel, 2001) to produce in-memory data models in a Java based solution. The data was backed on a key-value store, and the solution included a prototype visualisation based on O3D – JavaScript API for building interactive 3D applications in the browser (Google, 2010). Subsequently BIMSurfer project (BIM network, 2011) has added a WebGL based visualisation client. Lack of documentation and dependency on the BIMserver makes it difficult to reuse the visualisation component of BIMSurfer in a custom project, not based on BIMserver.

To conclude, the research projects in the process industry seem to be ignoring the need for Web3D enabled portals. BIM solutions seem to acknowledge this gap, but the existing prototypes either fail to address the specific requirements of the oil and gas industry or are not sufficiently generic to be reusable in this context.

INTEGRATION FRAMEWORK OVERVIEW

The ultimate outcome of this research and development project is an engineering information integration framework which serves as a basis for a commercial SaaS solution. The framework uses a central information database, which is being populated by loading in files in a plain text (tab delimited) format (Fig. 1). It is difficult to achieve an automatic system integration without an extensible format (e.g. XML or JSON), but this is not a requirement due to the heterogeneity of the applications involved (design/review software, Microsoft office tools, ERP systems). Such an approach allows end users to extract and transform the data from text files using common data processing software - e.g. Microsoft Excel, Talend Open Studio (talend, 2014), etc. It also makes it possible to utilise data from systems that are not always online, or provide access only to snapshots of the data due to security reasons.



The system utilises a simplified ontology called Class Library. Class Library defines the equipment attribute requirements, which change during the project. It also defines a set of mappings for various data sources (e.g. text based datasheets, 3D model extracts, etc.), which are used to resolve naming conflicts and map data into unique entities with an extended set of attributes within the database.

As JavaScript is a de facto programming language used in modern web browsers, the framework has been implemented on a JavaScript based Node.js server engine (Joyent, 2014). Such implementation eases the system development as the data model only needs to be expressed in the constructs of one programming language (JavaScript). The system delivers data over a web portal and REST (Representational State Transfer) API. Web interface allows a convenient and immediate access to information on various devices, without the need to install any additional software. The web application can also be integrated into a company's web portal. The REST API interface provides data for the plug-ins in desktop applications (e.g. NavisWorks), which can then retrieve additional project information for the selected geometry shapes within the desktop application. Cloud based solution is suggested by default, as this provides the best availability and scalability options, but it can also be deployed inside an internal company network.

Such a system requires an efficient storage engine. A semi-structured database system has been chosen for this project (Rasys and Dawood, 2012). It provides a balance between performance and functionality, providing an adequate level of reliability with the possibility of scaling up for large deployments. The database is capable of storing and querying the JSON objects directly; therefore the manipulation of the data becomes much easier in a JavaScript based framework.

Scene extraction and segmentation

In the prototype mapping implementation (Fig. 2) the model is exported from Autodesk NavisWorks into a text based FBX file. The 3D scene is triangulated – parametric shape definitions are converted into a polygon mesh. This increases the size of the 3D dataset and reduces the accuracy of representation, but greatly simplifies the processing logic, when providing data to the client rendering engine. The FBX file is processed by a Python script, which triangulates the scene objects using the FBX SDK Python bindings (Autodesk, 2014).

A single piece of equipment is usually modelled as a group of simple geometric shapes. Therefore the shapes

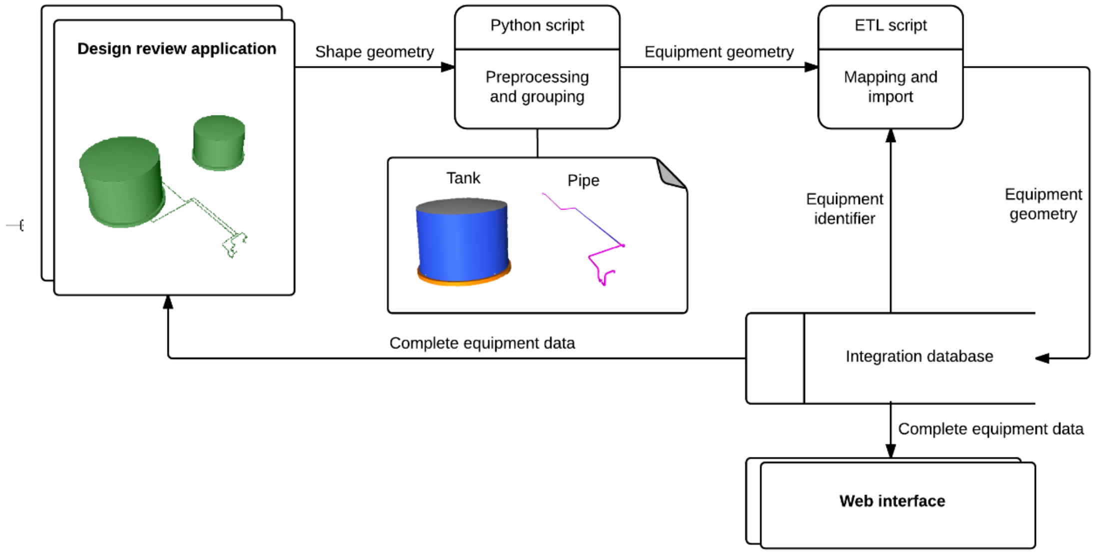
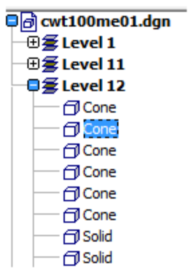


Fig. 2: mapping, aggregation and integration of 3D model elements

need to be grouped into equipment shape sets by identifier data present in the model. A tree node structure is used to organise the overall scene and usually no node or shape contains the actual equipment identifier. The identification data depends on the design application used, project naming, modelling conventions, etc., as shown in Fig. 3 and 4. As the data is often produced by different contractors it is often impossible to produce reliable 3D model equipment mappings to the equipment data supplied by other stakeholders. Therefore the mapping is usually a labour intensive process, which requires engineer input.
In the first example (Fig. 3) a pipeline is modelled by multiple shapes in the scene tree, none of which matches the actual pipe identifier in the database (*4"-VA-032-348-CB03-NI*). In the second example (Fig. 4), the shapes are not grouped in the scene tree by equipment; each of them has equipment identification data, which also does not match the equipment identifier in the database (*KHA-T-440003*). Due to such inconsistencies in the modelling practices, the mapping procedure cannot be fully automated and has to be configured specifically for each project.



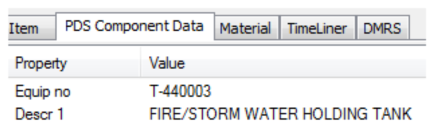


Fig. 3: Model with the part of an identifier in the node tree

Fig. 4: Model with the part of an identifier in the additional attributes

During the ETL (Extract, Transform, and Load) import process the shapes are grouped into collections, representing the same equipment item. This is done based on the information present in the node tree (Fig. 3). As the FBX file format does not contain the additional shape attributes, visible in Fig. 4, the integration is currently not possible with this type of modelling practice. After the shapes are grouped to parent nodes, their names are processed – the string */VA-032248-06-B1* is transformed into *VA-032-248* and then a partial match query is executed on the equipment identifiers in the database. As the match is found (*4"-**VA-032-348**-CB03-NI)*, the equipment geometry is associated with this particular database entity.
The full data model of an Oil & Gas facility is usually a large and complex 3D object, which is best viewed on screens with large resolution. Also the triangulated 3D scene takes significantly more space than the original

model; hence its delivery to mobile devices with slow network throughput could be challenging. Instead of scene simplification, a dynamic scene delivery is proposed to improve the user experience.
The user is usually interested in seeing the visualisation of a particular area of interest e.g. a location in the plant, equipment in a particular work package or change order, items with a certain attribute value, etc. Instead of loading the whole model and then hiding the unnecessary elements, the framework only fetches the data that is relevant for a particular query. Such scene delivery also allows applying access control rules over certain item data, including the 3D representation.

Dynamic scene rendering using Web3D

Web3D object visualisation implementation is based on WebGL technology. As WebGL is a wrapper API, representing OpenGL functions, a JavaScript scene graph library, three.js (three.js, 2014), is used to simplify the 3D content display. It abstracts the more complex mathematical calculations needed (e.g. matrix multiplications for projections, lighting and visibility calculations, etc). The library is able to dynamically construct a scene or add objects to a scene. When incorporated into a web portal, the approach allows dynamic scene composition based on a configurable view. The system retrieves only those 3D shapes that are needed to visualise the items, selected in the browser, and renders them in a custom scene (Fig. 5 and 6).
It has been observed that for larger scenes the HTTP requests can time out because of the size of the data that needs to be downloaded from the server. To overcome this issue a 3D streaming prototype based on WebSocket protocol (Fette and Melnikov, 2011) has been implemented. WebSocket protocol is designed for an efficient bi-directional communication between a client and a server application. Current versions of the most popular browsers (i.e. Chrome, Firefox, and Internet Explorer) natively support this technology. A socket.io library (Open-Source, 2014) is used to abstract the complexity of the protocol; it also provides compatibility with older browsers, where the software automatically falls back to using an XMLHttpRequest (W3C, 2014).

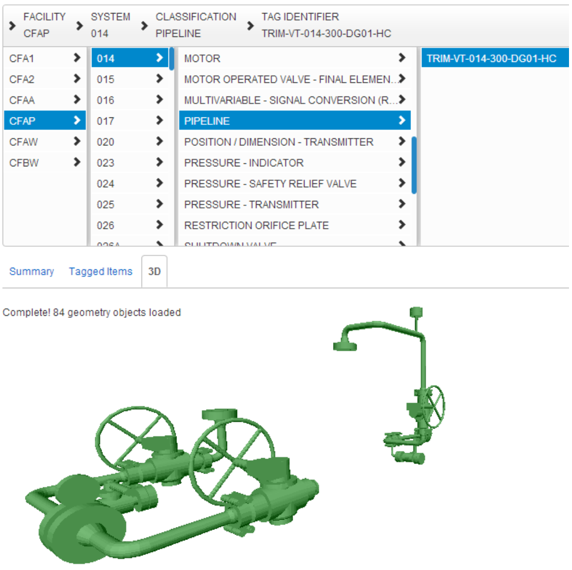


Fig. 5: visualising a single complex pipeline

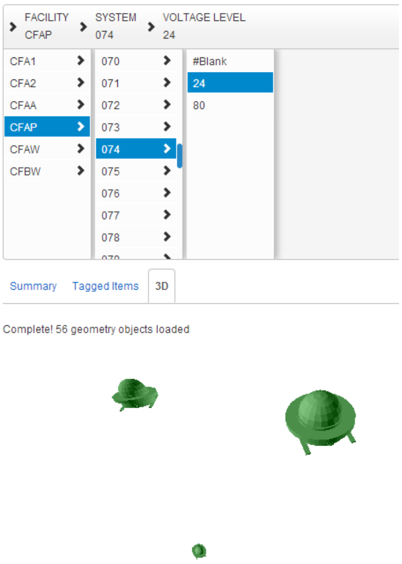


Fig. 6: visualising all items, which have a VOLTAGE LEVEL = 24V

The speed of the 3D scene delivery is a function of network latency (a delay between requests and responses), network throughput (speed of data transfer), web server and database performance, browser performance, complexity and accuracy of shape geometry (which affects the size of the data that the users have to download), the number of simultaneous users, etc. Additionally, because of the implementation of the socket based streaming, it is problematic to automate the system benchmark using the existing testing tools, such as Apache JMeter (Apache Software Foundation, 2013). Because of this, the metrics were collected in the web browser (Google Chrome v35), measuring the time it takes for the system to render the scene – starting from the submission of the initial request and ending with the moment, the last socket message is received (Fig. 7).

As testing in multiple networks with predefined speeds is impractical and results would not be comparable, tests were executed on a single fast network with around 90Mbit throughput to allow ample room for unexpected network activity. To simulate slower networks a bandwidth limiting software NetBalancer was used with various speed settings, which represent the real world scenarios.

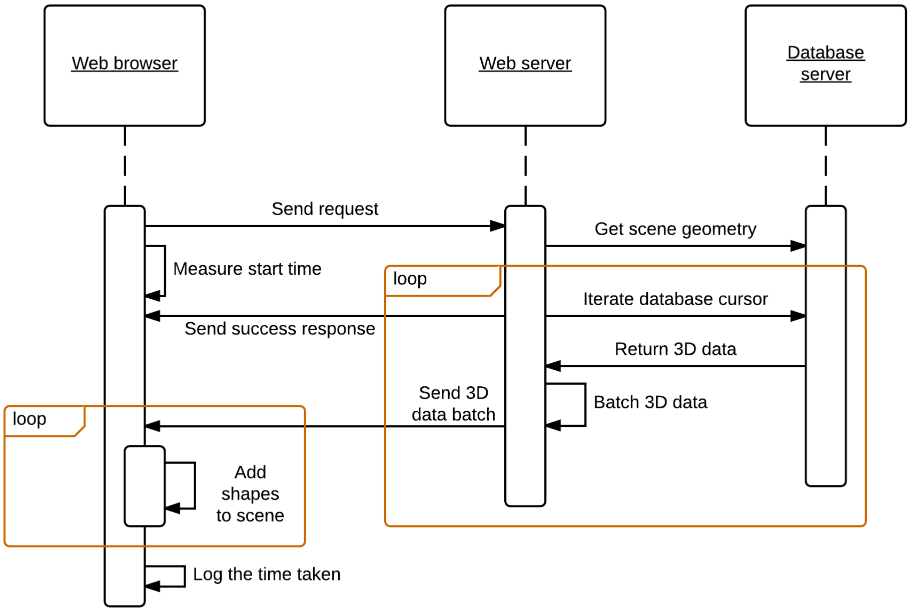
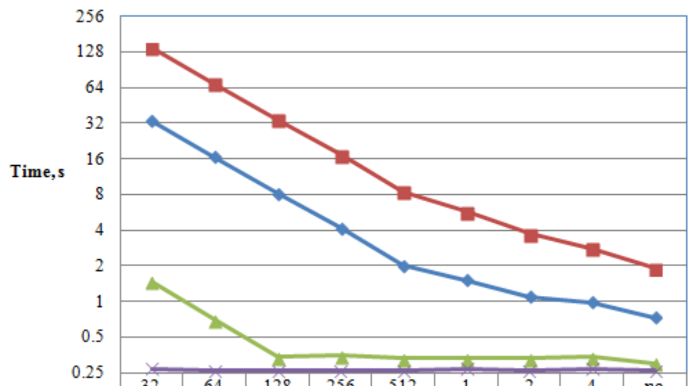


Fig. 7: The scheme for testing the speed of the 3D streaming service

The database/web server was based on Amazon AWS cloud infrastructure in Ireland, which also has ample reserve bandwidth available, and the measured round-trip network latency (ping) to Ireland was 25ms. All tests have been executed five times, the highest and lowest values were discarded and then an average result was calculated from the remaining three values. The performance testing results are presented in the Figure 8.



	32 KB/s	64 KB/s	128 KB/s	256 KB/s	512 KB/s	1 MB/s	2 MB/s	4 MB/s	no limit
4.26 MB (loading time, s)	138.06	68.99	34.44	17.22	8.51	5.7	3.74	2.82	1.9
1.03 MB (loading time, s)	33.26	16.66	8.18	4.14	2.02	1.51	1.1	0.98	0.73
48 KB (loading time, s)	1.46	0.7	0.34	0.35	0.33	0.33	0.33	0.34	0.3
3.3 KB (loading time, s)	0.27	0.26	0.26	0.26	0.26	0.27	0.26	0.27	0.26

Fig. 8: Loading times depending on various network speeds

RESULT ANALYSIS

The result analysis shows that the overall system latency on average is around 0.26 seconds – this is the duration of the whole request-data retrieval-response-render cycle with the minimal amount of data (3.3KB) and a very simple geometric shape (cube). The bottom two lines represent very small scenes and show that different baselines exist for different scene sizes. Increasing the network speed beyond those baselines does not result in a faster scene delivery. For larger scene sizes and when the scene loading time is higher than the baseline, the graphs show approximately a linear dependency between the loading time and the speed of the network. For slower speeds a 50% increase in network throughput also decreased by loading time by 50%. For faster speeds the dependency is not linear as other factors (speed of the server, browser capability to load a big scene, etc.) also become apparent.

Literature disagrees on the acceptable loading times for web applications. Some data shows that delay of up to 10 seconds is acceptable for the users to keep their attention and interest (Nielsen, 2014), provided a loading indicator is present, showing when the system is going to finish the task. Other researchers came to varying conclusions, although it seems that many have observed a change in behaviour between two and four seconds (Nah, 2004). In the tested configuration a medium scene with 4MB of data (132 equipment items) could be usable on a 512KB/s (4Mbit) internet connection, while the recommended network bandwidth is at least 2MB/s (16Mbit). As the loading time largely depends on the scene size, it is obvious that the web application accessed on a computer inside an office could deliver a larger scene within an acceptable timeframe than on a laptop with only mobile broadband available.

The size of the full oil and gas facility mesh model is at least multiple gigabytes. The prototype system reaches the limits of the recommended latency with the 4MB scene, therefore benchmark tests with larger scenes were not attempted.

DISCUSSION AND CONCLUSIONS

The backend of the integration framework has proven to be able to operate on large quantities of data (millions of items) (Rasys and Dawood, 2012). As the client side views are based on Web technologies, it offers an immediate access to the project data on a multitude of devices, although the web site usability is limited on devices with small screens. This can be improved by designing a separate, mobile version of the website or writing specific applications for mobile devices.

As the JavaScript engines inside the browsers abstract the hardware resources of the client machines, the performance of the client JavaScript code is unlikely to achieve the performance level of highly optimized, platform specific software written in a low level language (e.g. C). However, the JavaScript engine performance is constantly improving and there are several projects addressing the optimisation of the JavaScript compilers (Resig, 2013).

The issue with the data size of the 3D content is also apparent as the system is capable to deliver a certain amount of items that can be streamed in a reasonable time. Transferring a gigabyte of data over a Fast Ethernet network (100Mbit/s) takes more than a minute. In order to transfer large oil and gas facility model scenes and potentially the full model, the size of the dataset has to be addressed first. One potential solution is to investigate the parametric shape visualisation, which is considered to be more compact than the polygon mesh dataset. This will reduce the amount of data needed to display 3D objects and consequently increase the possible scene sizes.

Another potential solution is data compression and mesh simplification. A number of projects exist which address the issue (e.g. Li et al. 2007). While this potentially reduces the fidelity of the visual representation, it might be necessary to achieve a balance between the user expectations and the system usability.

The relatively low possible object count in a scene (up to hundreds of items) prevents the usage of the system in some of the more complex scenarios (e.g. clash detection, full facility visualisation and simulation, etc.). However, the proposed framework and tool is considered adequate for visualising a limited set of equipment or a small assembly of objects. Because of the dynamic nature of the generate view, the system can assemble views on any attribute stored in the database.

Such functionality is valuable for data mangers and maintenance engineers. The system adds a 3D representation into the custom data views for data managers. Maintenance engineers can visualise the equipment and their attributes before and during a work order execution.

REFERENCES

Apache Software Foundation, 2013. *Apache JMeter - Apache JMeter™*. [online] Available at: http://jmeter.apache.org/ [Accessed 25 Aug. 2014].

Autodesk, 2009. *The Autodesk LandXplorer Studio*. [online] Available at: http://www.3dgeo.de/landx.aspx [Accessed 25 Aug. 2014].

Autodesk, 2014. *FBX Software Development Kit*. [online] Available at: http://usa.autodesk.com/adsk/servlet/pc/item?siteID=123112&id=10775847 [Accessed 25 Aug. 2014].

Baader, F. and Nutt, W., 2003. Basic description logics. In: *Description logic handbook*. pp.43–95.

Bayer, B. and Marquardt, W., 2004. Towards integrated information models for data and documents. *Computers & Chemical Engineering*, 28(8), pp.1249–1266.

Beetz, J., Berlo, L. Van, Laat, R. de and Helm, P. van den, 2010. BIMSERVER.ORG – AN OPEN SOURCE IFC MODEL SERVER. In: *Proceedings of the CIP W78 conference*. Cairo.

Bell, R., 2012. Mentoring in the oil and gas industry. *Proceedings of the ICE - Management, Procurement and Law*, [online] 165(Volume 165, Issue 2), pp.95–101(6). Available at: http://www.icevirtuallibrary.com/content/article/10.1680/mpal.10.00013.

BIM network, 2011. *BIM Surfer*. [online] Available at: http://bimsurfer.org/ [Accessed 21 Jul. 2013].

Brandt, S.C., Morbach, J., Miatidis, M., Theißen, M., Jarke, M. and Marquardt, W., 2008. An ontology-based approach to knowledge management in design processes. *Computers & Chemical Engineering*, [online] 32(1-2), pp.320–342. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0098135407001032 [Accessed 25 Feb. 2014].

Braunschweig, B., Fraga, E., Guessoum, Z., Marquardt, W., Nadjemi, O., Paen, D., Piñol, D., Roux, P., Sama, S., Serra, M., Stalker, I. and Yang, A., 2004. CAPE web services: The COGents way. In: *European Symposium on Computer-Aided Process Engineering-14, 37th European Symposium of the Working Party on Computer-Aided Process Engineering*, Computer Aided Chemical Engineering. [online] Elsevier, pp.1021–1026. Available at: http://www.sciencedirect.com/science/article/pii/S1570794604802360 [Accessed 24 Aug. 2014].

Chuang, T.-H., Lee, B.-C. and Wu, I.-C., 2011. Applying cloud computing technology to BIM visualization and manipulation. In: *28th International Symposium on Automation and Robotics in Construction*. pp.144–149.

Dibley, M.J., Li, H., Miles, J.C. and Rezgui, Y., 2011. Towards intelligent agent based software for building related decision support. *Advanced Engineering Informatics*, [online] 25(2), pp.311–329. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1474034610001047 [Accessed 29 Mar. 2014].

Fette, I. and Melnikov, A., 2011. *The WebSocket Protocol*. [online] RFC 6455. Available at: http://tools.ietf.org/html/rfc6455 [Accessed 25 Aug. 2014].

Google, 2010. *o3d - WebGL implementation of O3D - Google Project Hosting*. [online] Available at: https://code.google.com/p/o3d/ [Accessed 25 Aug. 2014].

Hagedorn, B. and Döllner, J., 2007. High-level web service for 3D building information visualization and analysis. *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems - GIS '07*, [online] p.1. Available at: http://portal.acm.org/citation.cfm?doid=1341012.1341023.

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 Jul. 2013].

Hopkins, R. and Jenkins, K., 2008. *Eating the IT elephant: Moving from greenfield development to brownfield*. Addison-Wesley Professional.

Joyent, 2014. *node.js*. [online] Available at: http://nodejs.org/ [Accessed 25 Aug. 2014].

Kim, B.C., Teijgeler, H., Mun, D. and Han, S., 2011. Integration of distributed plant lifecycle data using ISO 15926 and Web services. *Annals of Nuclear Energy*, [online] 38(11), pp.2309–2318. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0306454911003161 [Accessed 8 Jul. 2012].

Li, W.D., Cai, Y.L. and Lu, W.F., 2007. A 3D simplification algorithm for distributed visualization. *Computers in Industry*, [online] 58(3), pp.211–226. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0166361506000935 [Accessed 8 Jul. 2012].

Lubel, J., 2001. *Open Source EXPRESS Parser*. [online] Available at: http://osexpress.sourceforge.net/ [Accessed 25 Aug. 2014].

Microsoft, 2014a. *Get Silverlight*. [online] Available at: http://www.microsoft.com/getsilverlight/Get-Started/Install/Default.aspx [Accessed 25 Aug. 2014].

Microsoft, 2014b. *Microsoft Silverlight*. [online] Available at: http://www.microsoft.com/silverlight/ [Accessed 25 Aug. 2014].

migenius, 2013. *RealityServer*. [online] Available at: http://www.migenius.com/products/realityserver/overview [Accessed 25 Aug. 2014].

Nah, F.F.-H., 2004. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3), pp.153–163.

Nielsen, J., 2014. *Response Times: The 3 Important Limits*. [online] Excerpt from Chapter 5 of Usability Engineering by Jakob Nielsen. Available at: http://www.nngroup.com/articles/response-times-3-important-limits/ [Accessed 31 Aug. 2014].

Oberle, D., Volz, R., Staab, S. and Motik, B., 2004. An extensible ontology software environment. In: *Handbook on ontologies*. Springer, pp.299–319.

Open-Source, 2014. *Socket.IO*. [online] Available at: http://socket.io/ [Accessed 25 Aug. 2014].

Parisi, T., 2012. *WebGL: Up and Running*. O'Reilly Media, Incorporated.

Rasys, E. and Dawood, N.N., 2012. Semi-Structured Data Modelling for a Web-Enabled Engineering Application. In: R.R. Issa and I. Flood, eds., *International Conference on Computing in Civil Engineering*. Clearwater Beach, Florida, United States: American Society of Civil Engineers, pp.41–48.

Rasys, E., Hodds, M., Dawood, N. and Kassem, M., 2014. A Web3D Enabled Information Integration Framework for Facility Management. In: *Australasian Journal of Construction Economics and Building-Conference Series*. pp.1–12.

Reece, C.A., Hoefner, M.L., Seetharam, R. V and Killian, K.E., 2008. An Enterprise-Wide Approach to Implementing "Digital Oilfield."In: *Intelligent Energy Conference and Exhibition*. Amsterdam, The Netherlands: Society of Petroleum Engineers.

Resig, J., 2013. *Asm.js: The JavaScript Compile Target*. [online] Available at: http://ejohn.org/blog/asmjs-javascript-compile-target/ [Accessed 19 May 2013].

Samdani, K. and Till, A., 2007. Engineering portals add significant value to E&P project delivery capabilities. *World Oil*, 228(11), pp.91–95.

Schevers, H., Mitchell, J., Akhurst, P., House, S.O., Point, B., Marchant, D., Bagot, W., Bull, S., Mcdonald, K., Drogemuller, R., Linning, C. and Coordinator, T.I., 2007. TOWARDS DIGITAL FACILITY MODELLING FOR SYDNEY OPERA HOUSE USING IFC AND SEMANTIC WEB TECHNOLOGY Space Room. 12(February), pp.347–362.

Schramm, C., Meißner, A. and Weidinger, G., 2010. Contracting strategies in the oil and gas industry. *3R International*, 1(Special), pp.33–36.

talend, 2014. *Talend Open Studio | Talend*. [online] Available at: http://www.talend.com/products/talend-open-studio [Accessed 25 Aug. 2014].

three.js, 2014. *three.js - JavaScript 3D library*. [online] Available at: http://threejs.org/ [Accessed 25 Aug. 2014].

W3C, 2014. *XMLHttpRequest Level 1*. [online] Available at: http://www.w3.org/TR/XMLHttpRequest/ [Accessed 25 Aug. 2014].

Wiesner, A., Morbach, J. and Marquardt, W., 2011. Information integration in chemical process engineering based on semantic technologies. *Computers & Chemical Engineering*, 35(4), pp.692–708.