BUILDING INFORMATION MODELS: A REVIEW ON STORAGE AND EXCHANGE MECHANISMS

Umit Isikdag, Ghassan Aouad, Jason Underwood, Song Wu
School of the Built Environment, University of Salford, UK

ABSTRACT: The concept of Building Information Modelling (BIM) has its roots in 2D digital drawings and product modelling. Early modelling efforts emerged in order to provide a solution to data exchange problems between several CAD and analysis systems. BIMs which used STEP as their base standard, adopted STEP data exchange and sharing mechanisms, enabling data exchange in three levels; file exchange, working form and shared database. In contrast to the traditional and accepted view of BIM as the central project model, some software vendors today define BIM as the management of a federated data layer, which consists of multiple models. After a brief summary on the history of information modelling in the construction industry, this paper reviews the storage and exchange mechanisms of BIMs and presents the role of model views as an information sharing and exchange method.
KEYWORDS: building information models, storage, sharing, exchange.

1 INTRODUCTION

Fragmentation is a key feature of the construction industry structure and client base. The traditional nature of the industry involves bringing together multi-disciplines/practitioners in a one-of-a-kind project that requires a tremendous amount of coordination. A typical construction project is extremely complex, involving a variety of organisations and disciplines being brought together for the duration of the project to form the "project team" or "virtual enterprise". These organisations vary in terms of size, physical location and IT capabilities. Furthermore, the industry is essentially an information intensive and processing industry. The traditional nature of the industry is extremely 'document-centric' with construction project information being captured predominately in documents, i.e. 2D CAD drawings, specifications, etc. Although project information may be produced in an electronic form, in essence it is distributed among the various multi-disciplinary teams involved in the project as documents. The format of such information is also rich and multi-dimensional. Such nature of the industry has resulted in significant barriers to communication between the various stakeholders, which in turn has significantly affected the efficiency and performance of the industry. Gallaher *et al.* (2004) indicated that US$15.8B is lost annually in the U.S Capital Facilities Industry due to the lack of interoperability. In recent years, Building Information Modelling has become an active research area in order to tackle the problems related to information integration and interoperability. Today, Building Information Models (BIMs) are promising to be the facilitators of integration, interoperability and collaboration in the future of the construction industry. This paper provides a review of the storage and exchange mechanisms in terms of BIMs, as they have evolved in addressing the issue of the seamless communication and transition of information between software applications/stake holders and across the project life-cycle phases. The paper first summarise the evolution of BIMs from the early initiatives of facilitating the exchange of CAD information between different CAD applications, leading to the emergence of the concept of product models which incited the development of BIMs. Later in the paper, the storage and exchange mechanisms of BIMs are reviewed and the role of model views is discussed.

2 BACKGROUND

2.1 *Storage and exchange of information in early CAD applications*

The construction industry is highly fragmented and there is a variety of different information systems that are used in each organisation. Thus, the transfer of information between different systems is and continues to be an apparent need. As Eastman (1999) explained, during the 1970s CAD companies realised the need for data exchange which was being expressed by industry. In response, they developed some low level methods to read and write data from applications. Some of these methods were:

1. *Writing parts of a project out to a file,* translating into a textual format and conversely reading such a text file and parsing it.
2. *Writing standalone application* that reads the file format in which project data is stored on disk.

3. *Providing subroutine calls* that can be executed from other programs to extract data from a file format.

As explained in Eastman (1999), Autodesk offered its proprietary file format DXF as an implementation of method 1, which then became a de-facto standard, while other methods were also implemented by some other CAD vendors. The drawback of these proprietary file formats were that:

1. They were controlled by the companies who had implemented them therefore they could change at any time.
2. Another problem to emerge around the same time was when several different applications had a need to exchange data. When a new file format is introduced a number of N*(N-1) data translators needed to be written (or 2*(N-1) more needs to be added to existing translators) in order to exchange data between N applications.

If a common format from which all applications can exchange data could be defined, the number of translators needed would decrease to two, i.e. one for writing the common format and one for reading it.

The drawbacks of the proprietary file formats and the advantage of using a common file format have forced communities of several knowledge domains to develop non-proprietary/vendor neutral file formats such as Interim Graphics Exchange Specification (IGES).

2.2 *The emergence of standard for exchange of product data*

The emergence of STEP - *ISO 10303: Industrial Automation Systems – Product Data Representation and Exchange* - was a result of the issues associated with the shortcomings of CAD data translation and the recognition by a number of industry-based research groups for a new generation of standards effort. In 1984, the International Standards Organisation (ISO) initiated a Technical Committee (TC184) to initiate a Sub Committee (SC4) to develop the STEP standard with the long-term ambition of improving the communication of engineering information and enabling integration through the co-ordination of open standards for data exchange and data sharing. In parallel, in the US the first Product Data Exchange Specification (PDES) report was issued in July 1984 and the second report was issued in November 1984 (Kemmerer, 1999). These reports laid the background for the PDES Initiation effort, a proof of concept project for the IGES organisation. In 1988, these two different efforts (TC184/SC4 and IGES) and information models developed for them merged into a single model called Integrated Product Information Model (IPIM). IPIM used emerging EXPRESS language for its data definition and specification. Following these developments, SC4 Resolution 68 established the first edition of STEP- *ISO 10303* in June 1990. The distinction between data sharing and exchange is clearly identified during STEP development efforts and in addition the STEP standard identified four implementation levels for data storage and exchange. These will be explained further in the next sections.

2.3 *Data exchange and sharing in STEP context*

ISO TC 184/SC 4 defined data exchange as the transfer of information from one software system to another via a medium that represents the state of information at a single point in time. This information snapshot is encoded digitally, typically in an ASCII or binary representation. For a given software system (System A) to generate a data exchange file, the system must implement specific functionality, generating a neutral file that represents the information to be exchanged. Another software system (System B) that receives the neutral exchange file created by System A has to provide functionality in its implementation for accessing the neutral file, interpreting its contents and creating an internal representation of that information (Kemmerer, 1999). On the other hand, according to ISO TC 184/SC 4 data sharing provides a single logical information source to which multiple software systems have access. In data sharing, the information source may be realised as a database management system, a specialised file system or a combination of the two. The characteristics that distinguish data sharing from data exchange are the centrality of the data and ownership of that data. In the exchange model, one software system maintains the master copy of the data internally and exports a snapshot of the data for others to use. Other software systems that import the exchange file has effectively assumed the ownership of the data. In the sharing model, there is a centralised control of ownership and there is a known master copy of the data, i.e. the copy maintained by the information resource. In theory, the data-sharing model alleviates the revision control problems associated with the data exchange model.

2.4 *STEP implementation levels*

STEP has four different implementation levels derived from PDES implementation levels. Wilson (1990) (cited in Loffredo, 1998) presented these four levels as:

1. *File exchange level:* EXPRESS-defined product data is passed between applications using flat files. The STEP Part 21 format has been defined for this purpose and at this level for an application to simply read and write files. An application may read the EXPRESS-defined data file using a dedicated parser and immediately convert the instance data into some other data structure.
2. *Working form level:* The software in working form level has all features of level one in addition to the ability to manipulate data. When an application in this level reads the data into its memory the data should be made available to the code, in a form organised and described by the EXPRESS model. Standard Data Access Interface (SDAI) is developed as a standard API for level two. The SDAI functions allow the product data to be manipulated.
3. *Database level:* This level has all features of level two along with the ability to work with the data stored in a database.
4. *Knowledgebase Level*: Implementations of this level will have all features of level three and should be able to reason about the contents of the database. This level has never been implemented.

In this section, we would like to note that BIMs (such as IFC, CIS2) that are defined with STEP description methods such as EXPRESS can be exchanged and shared using the first three implementation levels (as the fourth level has never been implemented). Today, the sharing and exchange methods of BIMs are not limited to these three levels. The next section looks at STEP implementation levels from a different perspective in order to identify where STEP implementation levels fit into the bigger picture of software/system integration. Further sections then expand these three levels in light of the state of art in data sharing and exchange.

2.5 *Data storage and exchange in software integration perspective*

Software integration can be defined as making disparate applications work together to produce a unified set of functionality. There is no standard classification for approaches to software integration, as approaches change by emerging technologies and technological achievements. Well known textbooks on integration technologies and their applications include Erl (2004), Hophe and Woolf (2003), Fowler et al. (2002), Chappell (2004), and Linthicum (2003). Erl (2004) identified three levels of integration, which are *data level integration* (where an application logic of an application reaches the data layer of another application or data from an application's data layer is replicated to another application's data layer), *application level integration* (where the integration is between application logic layers) and *process level integration* (which is process oriented). Linthicum (2003) explained four different approaches to integration, namely *information oriented* (which includes data replication by file exchange and database replication, data federation by the integration of multiple databases into a single unified view of them, and interface processing by using APIs), *business process oriented*, *service oriented* (where a composite application is created by using several components, application interfaces or wrapped legacy applications, which provide methods that can be invoked by standard messages over the web) and *portal oriented* (where all information is made available through a web browser). Hophe and Woolf (2003) provided four styles for integrating different systems. These styles are file transfer, shared database, remote procedure invocation and messaging. Table 1 provides information on the STEP implementation levels and corresponding integration levels, and approaches and styles, to demonstrate where the STEP implementation levels fits in the bigger picture of software integration.

Table 1. STEP implementation levels vs. Software Integration Approaches.

STEP Implementation Level	Focus	Corresponding Integration		
		Level (Erl, 2004)	Approach (Linthicum, 2003)	Style (Hophe and Woolf, 2003)
File exchange	Data Exchange	Data Level	Information Oriented Approach	File Transfer
Working Form	Data Sharing	Data Level	Information Oriented Approach	Remote Procedure Invocation
Database	Data Sharing	Data Level	Information Oriented Approach	Shared Database

3 BUILDING INFORMATION MODELS: A BRIEF HISTORY

BIMs of today have emerged as a result of an evolution from de-facto drawing exchange formats such as DXF through semantic AEC information models (which in the main are based on STEP technologies).

Tolman (1999) provided an historical overview of semantic AEC information models. Early models in the area include General AEC reference model GARM (Gielingh,1988), Integration Core Model (ICM) and the Integration Reference Model Architecture (IRMA). These efforts continued with the development of the Building Construction Core Model (BCCM) which was later approved as Part 106 of the STEP standard (ISO 10303). Another early effort in the area includes COMBINE project, which is explained in Sun and Lockley (1997) and Eastman (1999).

Other important efforts in the area include Computer Integrated Manufacturing of Constructional Steelwork (CIMSteel and CIS/2) explained in Eastman (1999), NIST CIS2 Web Site (2005), and Eastman *et al.* (2005), Engineering Data Model (EDM) as explained by Eastman *et al.* (1991), Semantic Modelling Extension (SME) explained in Zamanian and Pittman (1999), models developed in the Integrated Design Environment (IDEST) project as explained in Kim *et al.* (1997), Kim and Liebich (1999), RATAS and STEP Part 225 as explained in Eastman (1999).

Eastman (1999) grouped these models as Building Aspect Models and Building Framework Models. The author indicated that Building Aspect Models are similar to the ISO 10303 models developed for middle level 'application domains' that addressed the data exchange needs related to the organisational level of engineering departments. On the other hand, according to Eastman (1999), Building Framework Models are more comprehensive and appeared as the result of efforts to develop a Building Model structure capable of capturing the information needed to represent an overall building.

Most of these semantic models also adopted product modelling concepts by being enablers of communication, interpretation and processing of information, thus many of them are also known as Building Product Models. Similarly, most of them such as CIS2, BCCM, etc. are defined using STEP (ISO 10303) description methods (EXPRESS), thus these models are also referred to as BIMs, deriving from the term Information Modelling that is used in STEP resources such as Schenk and Wilson (1994). The next section will focus on the recent definitions of BIMs.

4 BUILDING INFORMATION MODELS: THE RECENT DEFINITIONS

NBIMS (2007) defines three important elements of building lifecycle as the process helix, the knowledge core and the external suppliers of products and services. The knowledge core is defined as the information backbone which provides historical and current data about the lifecycle processes. Information exchanges occur throughout the processes between all the stakeholders that take part in these processes. In this context, Building Information Modelling can be defined as a new way of creating, sharing, exchanging and managing the information throughout the entire building lifecycle. The NBIMS initiative categorises Building Information Modelling in three ways as a:

- *Product*: an intelligent digital representation of a building.
- *Collaborative process:* which covers business drivers, automated process capabilities, and open information standards use for information sustainability and fidelity.
- *Facility*: of well understood information exchanges, workflows, and procedures which teams use as a repeatable, verifiable, transparent, and sustainable information based environment used throughout the building lifecycle.

According to NBIMS (2006), a BIM is a computable representation of all the physical and functional characteristics of a building and its related project/life-cycle information, which is intended to be a repository of information for the building owner/operator to use and maintain throughout the life-cycle of a building.

AGC (Associated General Contractors) Guide (2006) defined BIM as a data-rich, object-oriented, intelligent and parametric digital representation of the facility, from which views and data appropriate to various users' needs can be extracted and analyzed to generate information that can be used to make decisions and improve the process of delivering the facility.

CRC Fact Sheet (2005) defined BIM as a three dimensional database designed specifically for built facilities, and indicated that BIM integrates a digital description of a building with all the elements that contribute to its ongoing function such as air conditioning, maintenance, cleaning or refurbishment along with describing the relationship between each element.

US General Services Administration BIM Guide (2006) indicated that the information in a BIM model catalogues the physical and functional characteristics of the design, construction, and operational status of the building. This information may span a number of disciplines and application types. BIM integrates this information in one database in a consistent, structured, and accessible way. The importance of BIMs stems from having an open interchange of information across platforms and a transferable record of building information throughout a building lifecycle.

Autodesk (2002) explained that BIM solutions have three characteristics:

1. They create and operate on digital databases for collaboration.
2. They manage change throughout those databases so that a change to any part of the database is coordinated in all other parts.
3. They capture and preserve information for reuse by additional industry-specific applications.

Today, the current key efforts in the area of BIM are the Industry Foundation Classes (IFC) and the CIMSteel Integration Standards (CIS), both of which are defined using STEP description methods, and can be shared and exchanged in the three implementation levels explained earlier.

IFC is the effort of IAI/buildingSMART whose goal is to specify a common language for technology to improve the communication, productivity, delivery time, cost, and quality throughout the design, construction and maintenance life cycle of buildings. Each specification (called a 'class') is used to describe a range of things that have common characteristics. These IFC-based objects aim to allow AEC/FM professionals to share a project model, while allowing each profession to define its own view of the objects contained within the model. In 2005, IFC became an ISO Publicly Available Specification (as ISO 16739)

CIS are one of the many results to emerge from the CIMsteel pan-European Eureka 130 project. The project involved some 70 organisations in 10 countries to improve the efficiency and effectiveness of the European Constructional Steelwork Industry both through the harmonisation of design codes and specifications, and through the introduction of Computer Integrated Manufacturing techniques for the design, analysis, detailing, scheduling, fabrication, erection and management functions. The CIS are open standards for the digital exchange and sharing of the engineering information relating to a structural steel framework.

In our research we identified several definitive characteristics of the BIMs by analysing the recent definitions of them and by investigating the key Building Information Modeling efforts in the area. As a result, the definitive characteristics of the BIMs were identified as follows:

1. *Object Oriented:* Most of the BIMs are defined in an object-oriented nature.
2. *Open/Vendor Neutral:* BIMs are developed with the aim of effective information exchange and sharing, therefore being open/non-proprietary and vendor-neutral is recognised as an important characteristic.
3. *Enables Interoperability:* BIMs are developed to overcome the problem of insufficient interoperability, thus this is recognised as a natural characteristic. As explained in NBIMS (2007), software interoperability is enabled by seamless data exchange among diverse applications using a shared universal information model.
4. *Data-rich / Comprehensive:* BIMs are data rich and comprehensive as they cover all physical and functional characteristics of the building.
5. *Extensible:* BIMs can be extended to cover different aspects of their information domain. For example, the IAI in the development and release of IFC 2x marked a major change in the way that IFCs had been previously developed/released. The IAI created a framework for the development of models to progressively extend the range and capability of IFCs in a modular way, i.e. projects developing models using the platform and issuing models independently as work is completed.
6. *Three dimensional:* BIMs always represent the geometry of the building in three dimensions.
7. *Covers various phases of the project life-cycle:* State of the art BIMs cover various phases of the project life-cycle. The objects of the model can be in different states in different phases of the lifecycle in order to

represent the N dimensional information about the building.

8. *Spatially-related:* Spatial relationships between building elements are maintained in the BIMs in a hierarchical manner.
9. *Rich in semantics:* BIMs store a high amount of semantic (functional) information about the building elements.
10. *Supports view generation:* The model views are subsets or snapshots of the model that can be generated from the base information model. BIMs therefore support view generation.
11. *Stored, shared and exchanged*: BIMs can be stored as a file or in a database, be shared in databases or by the help of APIs (when it is a physical file) and can be exchanged in form of physical files.

In many resources, one of the main characteristics of the BIM is defined as their ability to be stored, shared and exchanged. In the next section the ways and methods of storing, sharing and exchanging the BIMs are expanded on.

5 BUILDING INFORMATION MODELS: STORAGE, SHARING AND EXCHANGE

A BIM is defined by its object model. The object model of the BIM is the logical data structure (or data model) that defines all entities, attributes and relationships in the BIM. The object model is physically implemented in the form of schemas. The model data is created by an application and stored in physical files or databases. The model data must be consistent with the object model of the BIM. As mentioned previously, it is possible to share and exchange BIMs by using three implementation levels of STEP, if the model is defined by using STEP description methods. If not, (as the common practice in the other information domains indicate) then the BIM will possibly be defined and populated as a model in a relational or object database, and the data sharing will be realised by using the database interfaces. On the other hand, as the IFCXML implementation points out, the structure of the physical file will most probably be defined by using an XSD schema and the physical file will be exchanged as an XML file. The following sections present five different methods for storage and exchange of BIMs that were identified during the research. The first three methods are very similar to the three implementation levels defined by STEP, but in our approach we expanded these levels to cover XML based data sharing and exchange approaches. The fourth method is identified as a result of discussions between academics in our department with regard to Bentley's view of BIM. The fifth method is identified by observing the successful results of SABLE, a proof-of-concept project that demonstrated the use of Web Services to interact with the BIMs.

5.1 *Data exchange by using physical files*

Physical file exchange is carried out by creating and sharing a physical file of the BIM through the transfer of the file either by using physical mediums (e.g. DVD media), or using computer networks (e.g. Intranets and Internet). The physical file is created by a CAD application and can be exchanged among various applications. The BIM physical file can be an XML file or an ISO 10303 P21 file if the BIM is defined by using ISO 10303 description methods. For example, currently IFC P21, CIS2 P21, and IFCXML physical files are exchanged among various applications.

5.2 *Data sharing through application programming interfaces*

The BIM physical file can be accessed using an Application Programming Interface (API). This approach focuses on data sharing rather than exchange, and generally occurs in a two-tier architecture where a tier is formed by an API. The API can be proprietary for the BIM it is defined for, or it can be the Standard Data Access Interface (SDAI) API (if the BIM is defined by using ISO 10303 description methods). If the physical file is an XML file, then the model can be shared using the XML interfaces (i.e. APIs supporting DOM). The commercial APIs for IFC BIMs include BSPRo Server, IFC Active Toolbox, IFCsrv ActiveX Component, TNO IFC Engine. The SDAI APIs include StepCase and JSDAI.

5.3 *Data sharing through central project database*

Central project databases can store data that covers many aspects of the engineering life cycle. The advantage of storing the BIM in a central shared database is that multiple applications can access the product data, and make use of the database features such as query processing and business objects creation. The BIMs can be stored in a relational or object database system. Commonly the database entities are created (populated) from the object model of the BIM, which can either be carried out manually, by a database script or some specific database management systems have the ability to automatically create related entities from the object model schema. These object model schemas can be XML schemas (XSD) or ISO 10303 schemas (defined in EXPRESS language). The database can be populated by importing a physical file of the model (i.e. an XML file or a P21 file), or manually creating entity instances by using standard (i.e. ODBC) or proprietary (i.e. SDAI) database interfaces (which is not a common method). The data then can be queried using the database interfaces. One of the most well known databases in the field is the EDM Model Server.

5.4 *Data sharing through federated project database*

Linthicum (2003) defined data federation as the integration of multiple databases into a single unified view of them. This approach views the BIM as a composite information model in a federated database that is distributed but synchronised. The term federated database is used for the logical database and in physical terms it indicates a federated data layer, which consists of loosely coupled databases and files, coordinated by an application. As Linthicum (2003) explained, database federation software places a layer of software (middleware) between the physical distributed databases, files and the applications that view the data. This layer connects to the back-end databases and files using available interfaces and maps the physical databases to a virtual database model that exists only in the software. The data is then shared by using this virtual database. In this approach, the software application together with the database federation software are the main actors which will allow the use of various formats and keep the consistency of data throughout the entire life-cycle of the building. Bentley White Paper (2003) indicated that federated database is characterised by a system that allows users to continue to transact locally using methods, tools and data formats they find most productive and also provides central controls to manage global connectivity and broader transactions. The federated databases offer tightly/loosely coupled transaction mix and local/global load balance to make it more scaleable. On the other hand, the central database provides tightly coupled transactions where all changes to the database are immediately synchronised with the central copy. Bentley White Paper (2003) argued that the building life cycle requires a careful mix of tightly and loosely coupled transactions and the federated database approach will be a better solution than the central database approach.

5.5 *Data sharing by web services*

Web Services can be defined as data, software or system interfaces that are accessible over the Internet. He (2003) indicated that two constraints exist for implementing the Web Services:

1. Interfaces must be based on Internet protocols such as HTTP, FTP, and SMTP.
2. With the exception of binary data attachment, messages must be in XML.

BIMs can be shared through Web Services in two ways. The web service interface can provide access to:

1. The central project database where the BIM is stored.
2. An API, which in turn provides access to a BIM physical file or to the domain specific views of the model.

The SABLE (Simple Access to the Building Lifecycle Exchange) project is an example which demonstrated the use of Web Services to interact with BIMs. The project aimed to provide IFC web services by creating and using domain specific interfaces to an IFC model. SABLE Web Site (2005) indicated that IFC have the following shortcomings:

- IFC model is difficult to understand and to implement.
- Compatibility between different releases is difficult to maintain.
- EXPRESS technology is not main stream.

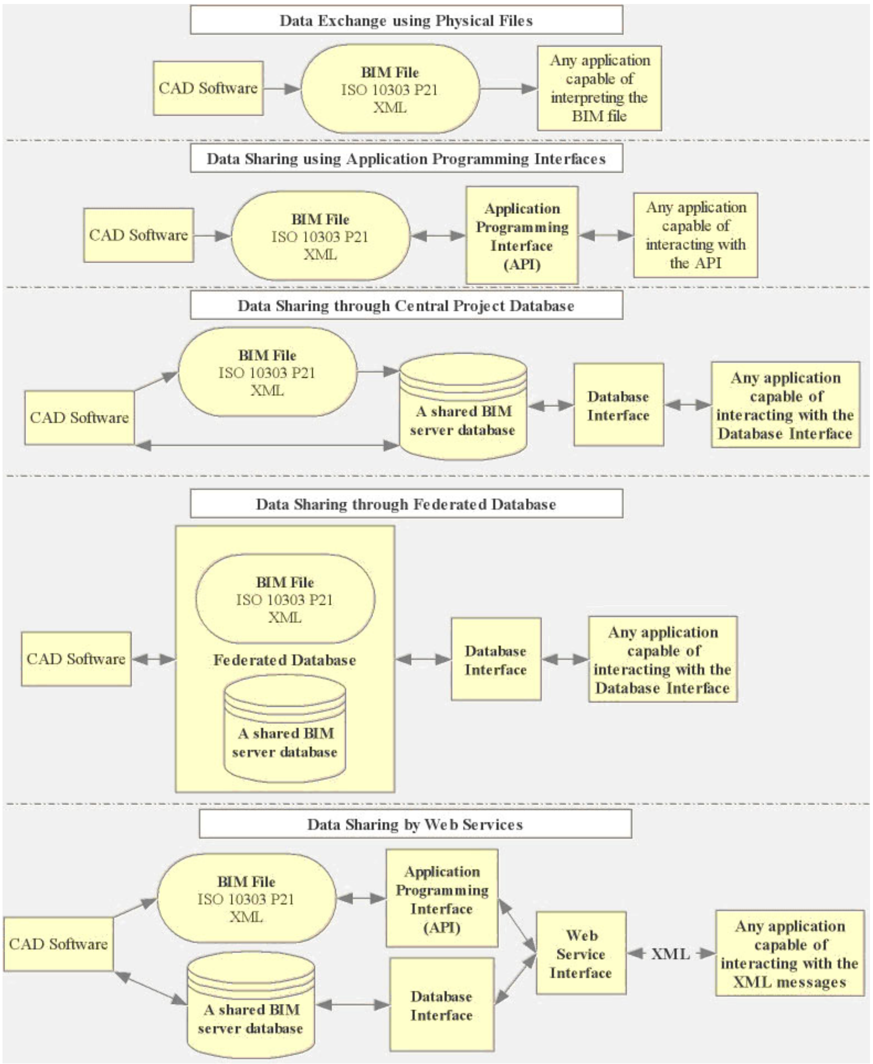

Figure 1. Methods for Storage and Exchange of Building Information Models.

The project participants mentioned that trying to fix any of them separately may have an adverse effect on the other one. The project proposed a solution to these shortcomings of IFC as:

1. *Creating low level web services:* these services are web interfaces to IFC processing components and model server databases.
2. *Creating high level web services:* these are domain specific web interfaces to an application server.
3. *Creating an application server:* that will use the data from low level services, process it and serve it for high level services.

According to the SABLE proposal, an application would either use low level web services to reach the data in the model server databases and IFC processing components, or use high level interfaces (over web services) to reach domain specific data. Figure 1 presents an overview of all the storage, sharing and exchange methods reviewed.

Along with the five methods presented here, the model views also have a significant importance in sharing and exchange of the BIMs. The next section will review the role of the model views.

6 THE ROLE OF MODEL VIEWS

In order to support several phases and the stakeholders of the construction life cycle, several views of the BIM needs to be generated. These views can be generated from files or databases by using application, database and web interfaces. These views can either be transient or persistent depending on the need. Eastman and Jeng (1999) indicated that model evolution is the transition of a model with one structure or schema to the same or different model with a different structure or schema. The authors described four types of model evolution as *model translation*, *view generation*, *modification of a single integrated model*, *evolution of a product model based on multiple application views* and *mapping between them and the central project model*. In this section, as the focus is on the model views, we will focus on the first two of the approaches, i.e. model translation and view generation. Eastman and Jeng (1999) explained that model translation involves a mapping from one complete model to another, where each model schema is defined statically. On the other hand, the model views are generated by a declaration of a model that is a subset of another model, or by declaration of a model that is derivable from another model. The original model is called the base model and the new model is called the view. The entities of the view are populated from the base model.

The persistent model views are generated by model translation, and under the following conditions the (translated) model can be called as the model view:

1. The view should not be a superset of a predefined (base) information model. The view can be a subset of the model or the model itself.
2. The view should provide a snapshot of the information model (or its subset).

If the model view is persistent then it will be stored in a physical file or a database, otherwise if it is transient the physical storage of the view is not necessary. The persistent model views can be used whenever there is a need to exchange a subset of BIM between various different domains or when there is a need to exchange a snapshot of the BIM in one stage of the project. For example, Eastman and Jeng (1999) indicated that model translations are most appropriate between the phases of the construction life cycle, i.e. at the end of the construction to generate a maintenance model. The transient views are commonly used when an application requires a subset of BIM entities from a model server database. Another type of model view is application/system specific view. Eastman and Jeng (1999) named this view as the application view. The application/system specific view does not have to be a subset of the base information model. In contrast, this view is an information model on its own. It is defined according to the needs of the application/system it is working with. Under the following conditions, an information model can be called as an application/system specific view:

1. The model should be interacting with a base information model.
2. The model should address the specific data needs of an application/system it is developed for.
3. The model should address a similar information domain with the base information model.
4. The model should address the same information domain with the application/system it is developed for.

The application/system specific views are used when the information domains of an application and the base information model are slightly different. An application (before processing the information) might need to eliminate some entities from the base information model, or conversely some information produced by an application might be redundant for the base information model. In such situations, the application will work with the application/system specific view, and the base information model will be updated by mapping information from the application specific view to the base model. In order to maintain consistency when the base model is updated, the mapping can also be in the other direction (i.e. from the base model to the application specific view). Isikdag (2006) demonstrated the use of application specific views while mapping information from the IFC model to an application specific view, which is developed to interact with a GIS. In the common practice, the model views (transient, persistent and application specific) are updated using EXPRESS-X and XSL languages.

7 SUMMARY AND CONCLUSION

The paper reviewed the ways and methods of information sharing and exchange using BIMs. The fragmented and document-centric nature of the construction industry continues to create barriers to effective exchange of information and integration. The lack of interoperability between information systems and software applications has caused great financial losses in the industry. Today, Building Information Modelling is a very active research area in order to tackle these problems related to information integration and interoperability. The recent collaborative effort in the US towards defining a national BIM standard and the development of Industry Foundation Classes, which later became an ISO Publicly Available Specification (16739), are just two examples that demonstrate the importance given to Building Information Modelling by the industry. Data sharing and exchange over the entire building life cycle can be enabled by using BIMs, which is recognised as potentially providing substantial benefits to the industry.

As most BIMs are being defined by using STEP description methods, three implementation levels of STEP are generally accepted as methods for storage and exchange of BIMs. In parallel with the developments of web technology, XML has become an important facilitator for file and message exchange in recent years. In addition, Web Services have also become an important method for data sharing and exchange. On the other hand, in line with the developments in the database technologies, the federated database approach appeared as a valuable method for data sharing. In the review, we expanded the three implementation levels of STEP to cover XML storage and exchange methods, and also highlighted the role of federated databases and Web Services as new storage and exchange mechanisms.

As explained earlier, file exchange methods are focused towards the data exchange where there is no centralised control of ownership, while the other four methods focus on data sharing where there is a centralisation and control of ownership. Three of these data sharing methods (APIs, Central Database, and Web Services) use tightly coupled transactions, and this provides advantages for maintaining the consistency of the base information model. On the other hand, the federated database approach offers tightly/loosely coupled transaction mix, where the centrality of the information is not clear as the other three approaches. In the federated database approach, it is the responsibility of the database federation application and software that manages the federated database to keep the consistency of data throughout building life-cycle. Otherwise, as there is no centrality in terms of a single shared file or database in this approach, the consistency can be lost and versioning problems between the different models can easily occur. It should also be noted that the Web Services can also be used together with federated databases, and can be a part of loosely coupled transactions.

The use of sharing and exchange methods explained here will facilitate the effective exchange of information throughout the building lifecycle. These data sharing and exchange methods need to be jointly used to create and manage the knowledge core (information backbone) of the lifecycle along the process helix. It is difficult to suggest that one method has a greater advantage over any other as every method has its own pros and cons. In terms of the file exchange and federated database approaches, problems on data consistency and versioning appear, while the other approaches that are built upon model and information centrality are less scaleable and load balancing becomes a need. It would be the joint decision of the project and the IT manager to use either a shared central BIM or a BIM that is exchanged among various applications, depending on the needs of the project and also on the phase of the building life-cycle. A mix-and-match approach where several of these methods are used in triangulation is also possible and reasonable. However, in this situation the data consistency should be carefully monitored. It should also be noted that none of these data sharing and exchange methods will work unless mature and robust BIM compliant software is offered by the software industry. Whatever method is used, the model views will always be important facilitators of information sharing and exchange and further research in the area should focus more on generating and using these model views.

REFERENCES

Associated General Contractors Guide (2006) “AGC Contractors’ Guide to BIM”, (Available online at: http://www2.agc.org/eCRM/scriptcontent/ProductDetail.cfm?pc= 2926E)

Autodesk White Paper (2002) “Building Information Modelling”, (Available online at: http://images.autodesk.com/apac_sapac_main/files/4525081_BIM_WP_Rev5.pdf)

Bentley White Paper (2003) “Does the Building Industry Really Need to Start Over?”, (Available online at: ftp://ftp2.bentley.com/dist/collateral/BIM_20White_20Paper_ 2C_20Low_20 Respdflo.pdf)

Chappell, D. (2004) Enterprise Service Bus, O'Reilly

CRC Fact Sheet (2005) “Sydney Opera House FM Exemplar project: Fact Sheet 2-BIM”, (Available online at: http://www.construction-innovation.info/images/pdfs/SOH-FH_Fact_Sheet_2.pdf)

Eastman, C. (1999) Building Product Models: Computer Environments Supporting Design and Construction, CRC Press, USA

Eastman, C., Bond, A. H., Chase, S. C. (1991) “Application and evaluation of an engineering data model”, Research in Engineering Design, Vol.2, No.4, pp. 185–208.

Eastman, C., Jeng, T. S. (1999) “A database supporting evolutionary product model development for design”, Automation in Construction, Vol.8, No.3, pp. 305-323.

Eastman, C., Wang, F., You, S. F., Yang, D. (2005) “Deployment of an AEC industry sector product model”, Computer Aided Design, Vol.37, No.12, pp. 1214-1228.

Erl, T. (2004) Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall

Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., Stafford, R. (2002) Patterns of Enterprise Architecture, Addison Wesley

Gallaher, M. P., O’Connor, A. C., Dettbarn Jr., J. L., Gilday, L. T. (2004) “Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry” , NIST Publication GCR 04-867, (Available online at: http://www.bfrl.nist.gov/oae/publications/gcrs/04867.pdf)

Gielingh, W. F. (1988) “General AEC Reference Model GARM, CIB Seminar Conceptual Modelling of Buildings”, Lund, Sweden

He, H. (2003) “What Is Service-Oriented Architecture”, (Available online at: http://webservices.xml.com/pub/a/ws/2003/09/30/soa. html)

Hophe, G., Woolf, B. (2003) Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Addison Wesley

Isikdag, U. (2006) “Towards the implementation of Building Information Models in Geospatial Context”, PhD Thesis, University of Salford, 2006

Kemmerer, S. J. (1999) “STEP: The Grand Experience”, NIST Special Publication 939, (Available online at: http://www.nist.gov/msidlibrary /doc/stepbook.pdf)

Kim, I., Liebich, T. (1999) “A data modelling framework and mapping mechanism to incorporate conventional CAD systems into an integrated design environment”, The International Journal of Construction Information Technology, Vol.7, No.2, pp 17-33.

Kim, I., Liebich, T., Maver, T. (1997) “Managing design data in an integrated CAAD environment: a product model approach” Automation in Construction, Vol.7, No.1, pp. 35-53.

Linthicum, D. S. (2003) Next Generation Application Integration: From Simple Information to Web Services, Addison Wesley

Loffredo, D. (1998) “Efficient Database Implementation of EXPRESS Information Models”, PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York, (Available online at: http://www.steptools.com /~ loffredo/papers/expdb_98.pdf)

NBIMS (2006) “National BIM Standard Purpose”, US National Institute of Building Sciences Facilities Information Council, BIM Committee, (Available online at: http://www.nibs.org/ BIM/ NBIMS_Purpose.pdf)

NBIMS (2007) “National Building Information Modeling Standard Part-1: Overview, Principles and Methodologies”, US National Institute of Building Sciences Facilities Information Council, BIM Committee, (Available online at: http://www.facilityinformationcouncil.org/bim/publications.php)

NIST CIS2 Web Site (2005), Web Site, (Available online at: http://cic.nist.gov /vrml/ cis2.html)

SABLE Web Site (2005),Web Site, (Available online at: http://www.blis-project .org/ ~sable)

Schenk, A. D., Wilson, R. P. (1994) Information Modelling: The EXPRESS Way, Oxford University Press

Sun, M., Lockley, S. R. (1997) “Data exchange system for an integrated building design system”, Automation in Construction, Vol.6, No.2, pp. 147-155.

Tolman, F. (1999) “Product modelling standards for the building and construction industry: past, present and future”, Automation in Construction Vol.8, No.3, pp. 227-235

US General Services Administration BIM Guide (2006) “GSA BIM Guide Series 01”, (Available online at: http://www.gsa.gov/bim)

Wilson, P. R. (1990) “Information Modelling and PDES/STEP”, Technical Report 90017, Rensselaer Design Research Center, Rensselaer Polytechnic Institute, Troy, New York

Zamanian, K. M., Pittman, J. H. (1999), “A software industry perspective on AEC information models for distributed collaboration”, Automation in Construction, Vol.8, No.3, pp. 237-248.