DEVELOPING A FRAMEWORK TO SUPPORT DATA EXCHANGE FROM HETEROGENEOUS SOURCES VIA IFC AND WEB SERVICES

Mark O. Danso-Amoako[1], Raja R.A. Issa[2], and Robert Cox[3]

ABSTRACT

The nature and complexity of projects in the Architectural, Engineering and Construction (AEC) industry makes the industry very fragmented. This is because these complexities necessitates the use of specialized firms such as mechanical, electrical, structural steel, networking and heating, ventilation and air-conditioning to complete projects. Although specialization in general is a good concept, it comes with a price. The downside of this multiparty concept is the problem of coordination, data and information sharing between the various parties. Standards and models have been developed to minimize the adverse effects of fragmentation. One of such models is the Industry Foundation Classes (IFC) developed by the International Alliance for Interoperability (IAI). Unfortunately most AEC applications do not comply with these standards and models. This paper reports on an ongoing research and development of a system to translate non-IFC compliant AEC applications into IFC and then share the data in real time with other parties over the Internet with the use of web services.

KEY WORDS

data exchange, integration, Industry Foundation Classes IFC, web services, data translation.

INTRODUCTION

For the last decade the total spending on the Architectural, Engineering and Construction (AEC) industry in the United States has been steadily increasing from 490 billion dollars in 1993 to approximately one trillion dollars in 2004 representing an average of 6-10 percent of the United States' Gross Domestic Product (GDP) (U.S. Census Bureau 2005). The U.S. construction industry employed 8.4 million people in 2001 and an additional one million people worked in architecture and engineering (Constructionweblinks 2005). These figures underscore the importance of construction on the national economy. However despite these encouraging figures, a lot of problems have been identified in the AEC industry that make the industry less competitive compared to other industries. One of the major problems is data exchange and information sharing, which is the focus of this paper.

---

[1] Ph.D. student, Rinker School of Building Construction, University of Florida, Gainesville, FL 32611, Phone (352) 273 1150, markdan@ufl.edu

[2] Rinker Professor, Rinker School of Building Construction, University of Florida, Gainesville, FL, 32611, Phone (352) 273 1152, raymond-issa@ufl.edu

[3] Associate Professor, Rinker School of Building Construction, University of Florida, Gainesville, FL, 32611, USA, Phone (352) 273-1153, robcox@ufl.edu

The construction industry in general is highly fragmented with the degree of fragmentation unparalleled to any other industry sector, not even the manufacturing industry (Dawood et al. 2002). Gone are the days of the "master builder" who took charge of the entire project from inception to completion. The complexity of current projects has increased in multiple folds over the years and has necessitated the use of specialized firms such as mechanical, electrical, structural steel, networking and heating, ventilation and air-conditioning to complete projects. This makes the AEC industry multiparty in nature with a high number of stakeholders each of whom has to contribute significantly to the successful realization of the project (Deng et al. 2001). The downside of this multiparty concept is the problem of coordination, data and information sharing among the various parties. To overcome these challenges a framework for real-time data exchange and information sharing between AEC industry applications, by means of Industry Foundation Classes (IFC) as the common denominator and web services technology is proposed.

The main objectives of this paper are to explore the:

- possibility of the development of IFC data translators for AEC applications, especially legacy systems that are currently heavily in use but do not comply with contemporary data exchange models.
- use of web services to transport the translated data between firms in real time.

The search for real time data and information sharing solutions especially between non-IFC compliant applications (legacy applications) is underlying goal of this research. A translator system is proposed to map native application data from non-IFC standard to IFC and then share that information in real time over the internet through the use of web services. With this combination it is envisioned that data integration between AEC applications can be achieved. IFC was chosen due to its scalability and maturity in terms of overall development and testing. Almost all construction concepts can be represented in the IFC format. For example, application A's data can be reduced to IFC and then accessed by application B with little or no loss in semantics.

CHARACTERISTICS OF THE AEC INDUSTRY

Fragmentation is one of the distinct characteristics of the construction industry (Howard 1989). A typical construction project involves many stakeholders that include owners, general and sub-contractors, suppliers, project managers, architects and engineers spread over sometimes mutually exclusive work sections. With such a complex network, one can easily envision the likely difficulties that will be encountered in coordinating and exchanging data between the various parties on a typical project. Furthermore the extent of this fragmentation is compounded by the fact that the various parties use different computer applications resulting in incompatible data format (Zhu and Issa, 2003). This current high level of fragmentation and the subsequent lack of adequate measures to help in coordination have resulted in low productivity, cost and time overruns, conflicts and disputes, claims and time-consuming litigation (Dawood et al. 2002, Deng et al. 2001). It has been reported that about two-thirds of the construction problems are a result of inadequate coordination and inefficient means of communication of project information and data (Cornick 1990).

DATA EXCHANGE AND INTEGRATION STANDARDS THE AEC INDUSTRY

In a quest to minimize the adverse effect of fragmentation in the AEC industry many software applications, data models and standards have been developed and tried within the past two to three decades. Some of these early models include Information Reference Model for AEC (IRMA), Building Product Model (BPM), Information/Integration for construction (ICON), Unified Approach Model, General construction object model (GenCOM), ATLAS Large-Scale Engineering Project Type Model, COMBINE and Standard for the Exchange of Product Model Data (STEP) (see Anumba and Amor 1999, Eastman 1999, Froese 1996 for more details). One of such more contemporary standards which is quickly becoming a *de facto* standard of the AEC industry is the Industry Foundation Classes (IFC) developed by the International Alliance for Interoperability (IAI) is briefly discussed in the next section.

INDUSTRY FOUNDATION CLASSES (IFC)

The Industry Foundation Classes (IFC) is perhaps the largest and most ambitious effort that is being undertaken to develop an integrated building model (Eastman 1999) with the hope of achieving the goal of Computer Integrated Construction (CIC). The development of the IFC was based on all the experience and successes of earlier projects especially Standard for the Exchange of Product Model Data (STEP) (Rönneblad and Olofsson 2003). The IFC model architecture (IAI 2000) is built up of data model schemata organized under four main layers namely, resource layer, core layer, interoperability layer and domain layer.

The lowest layer contains the resource classes that are used by classes in the upper layers. These classes are general, low-level, domain-independent and even not AEC-specific such as date and time.

The core layer comprises the kernel and core extensions (control, product and process extensions). The kernel provides the basic abstract part within the IFC architecture. Similar to the resource layer the concepts in the kernel are general and non-AEC-specific such as object, property and relationship but they are required for all other higher level models. The purpose of the core extensions is to serve as the first line of specialization of the kernel objects towards AEC specific constructs. For instance the core process extension provides information that supports the concept of process in the AEC context and the core product extension helps to define the properties of the product (building component).

There are some objects that are shared by multiple domains. Such objects are captured by the interoperability layer. Major building elements like wall, beam, column, slab, roof and stair are not unique to any one particular domain and thus are captured by the "shared building elements" data model.

The final and topmost layer is the domain layer. As a result of successive refinements, the model at this layer provides domain specific support. The models that are currently contained in the domain layer of IFC2.2 are HVAC, electrical, architecture, construction management, building controls, plumbing, fire protection, structural element, structural analysis and facilities management.

The primary goal of IFC is to enable interoperability between AEC and facilities management applications from different software developers (IAI, 2000). The fundamental idea and concept of IFC is a great one and if implemented by all AEC industry software developers and subsequently used by all construction companies will totally eliminate data exchange and information sharing problems. Unfortunately, this is currently not the case, about ten years since the inception of IFC. The construction industry is relatively slow to accept technological changes compared to its competitors like the manufacturing industry (Alshawi and Ingirige 2003, Brandon and Betts 1997). Although currently the IAI can boast about a number of applications that are IFC-compliant, unfortunately, there are more AEC applications out today that are non-compliant to the IFC model.

WEB SERVICES

Web services (sometimes referred to as XML Web services) are software components that provide some type of service over the internet. The major similarity between conventional websites and a web service is that they are both reachable through a public URL and are subject to the same security restrictions of an HTML-based web site (Esposito 2003). However there are a number of properties that make this exciting new technology different from conventional websites. Web pages are targeted at human users whereas web services are developed mainly for access by other applications. In plain words, Web services are about machine-to-machine communication whereas web pages are about human-to-machine communication (Papazoglou and Dubray 2004).

The infrastructure of Web services makes use of existing open internet standards and transport protocols such as Hypertext Transfer Protocol (HTTP), Extensible Markup Language (XML), Simple Object Access Protocol (SOAP) as well as Web Services Description Language (WSDL) and Universal Discovery Description Integration (UDDI). It is the combination of such standards that make web services platform-independence, accessible and consumable from any client or internet-enabled device (Esposito 2003).

COMPLEXITIES IN DATA SHARING

Prior to the use of models it was impossible for heterogeneous applications to effectively share data and information. For instance, a scheduling application TYPE A had no way to share the data produced by another scheduling application TYPE B since the files were incompatible. Where access could be made to the files, translators must be provided for each application's link to the other as shown in Figure 1A. This means for ***n*** number of heterogeneous applications there is the need for **n(n-1)** or $\boldsymbol{n^2-n}$ number of translators to be developed. This is a function upper bounded by big-oh $\boldsymbol{O(n^2)}$ as n gets large. For example an industry with fifty heterogeneous applications that must share application data will require 2450 translators to be developed. This is a very laborious task if not impossible to achieve.

With the advent of data exchange model such as IFC and CIMSteel Integration Standards (CIS/2) the number of translators drops to ***2n***. For each application two translators are required Figure 1B:

- The first translator to convert the data from the native application data structure to the model (neutral) format

- A second one to read the neutral format and reconvert it to the native application data structure.

This function is linear and upper bounded by ***O(n)*** as n gets large. A graph showing these complexities is shown in Figure 2.

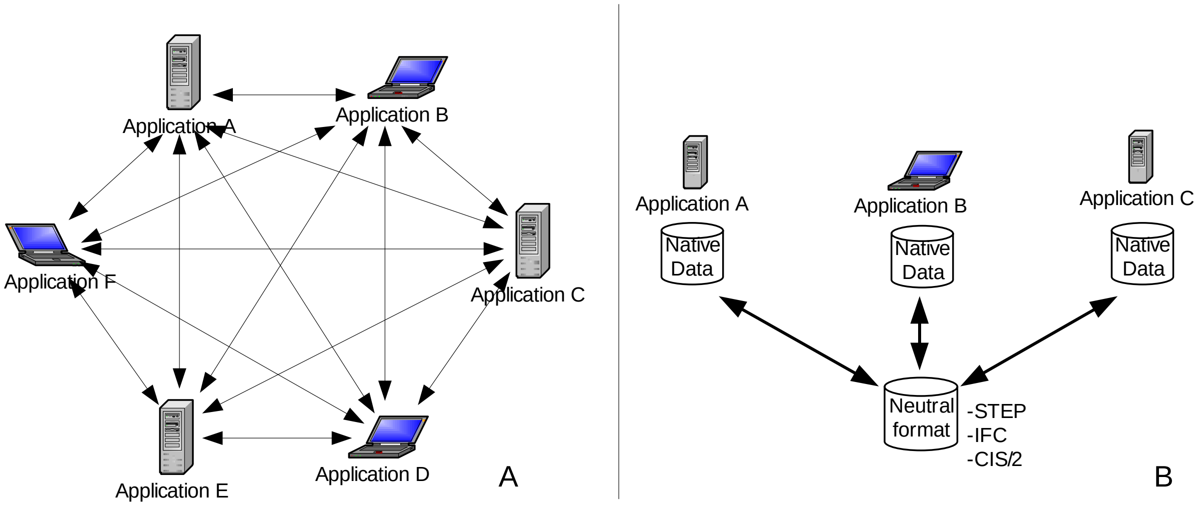


Figure 1: Complexity of creating translators (A) without models and (B) with models

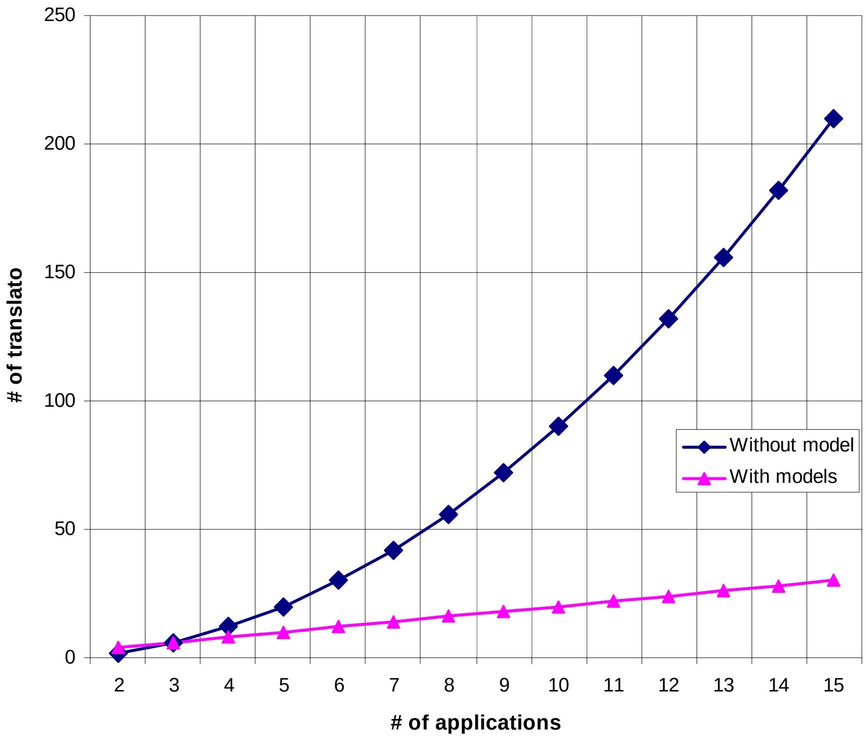


Figure 2: Complexity Analysis of Developing Translators

THE NEED TO SUPPORT LEGACY SYSTEMS

If all construction application developers agree to make their products compliant with a given data model standard say "X" and all construction application users agree to purchase this new application which is X-compliant, then we are guaranteed to have a lower bound of the number of translators required which is ***2n***. From past experience in the construction industry, we know this may take a very long period of time to materialize. In this modern day, paper based transactions are still a norm. It is not uncommon to hear companies claim they have advanced in technology when in fact they are sending email attachments of documents that have been scanned. Most of the time these documents must be printed out and then manually inputted at the receiving end into a similar application but of a different vendor (Zhiliang et al. 2004). Taking scheduling as an example, a common practice is to post or hand deliver hard copies of the schedule to another party to be manually entered into their scheduling application. The so-called hi-tech companies email these schedules, but its intended use is just like that of a hard paper copy.

The development of models facilitates data integration but the models by themselves in isolation do not provide a silver bullet for the problem. The early AEC models such as GenCOM, ATLAS, ICON, AEC Core Model and even contemporary ones like IFC and CIS/2 all depend on file-based data exchange which normally limits real time data and information sharing. Secondly in order to share data with these model systems, all the applications involved should be in compliance with that specific standard. Compounded by the fact that AEC industry has a history of reluctance in accepting technological change and depends heavily on legacy systems that do not comply with current data standards it is reasonable to say that, in one way or the other, legacy system must be accounted for when considering data exchange and information integration.

PROPOSED SYSTEM

This section describes the proposed system for real-time data integration between heterogeneous AEC applications which is currently under development by the authors. The idea is based on the fact that current AEC models, specifically IFC, is matured and can implement most if not all common AEC concepts. The principle as shown in Figure 3 is to map data from an application TYPE A to a common base, IFC model.

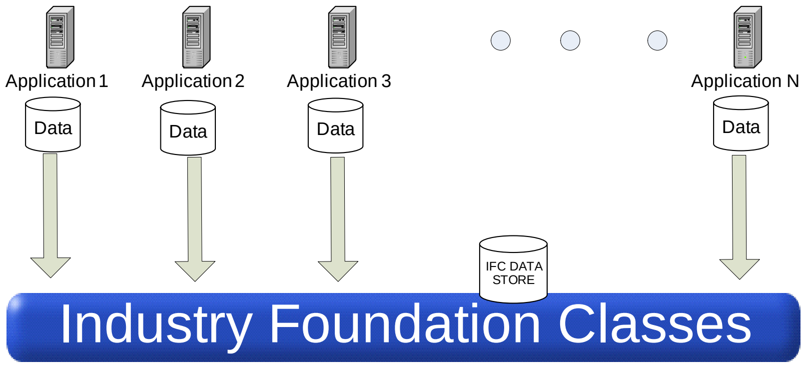
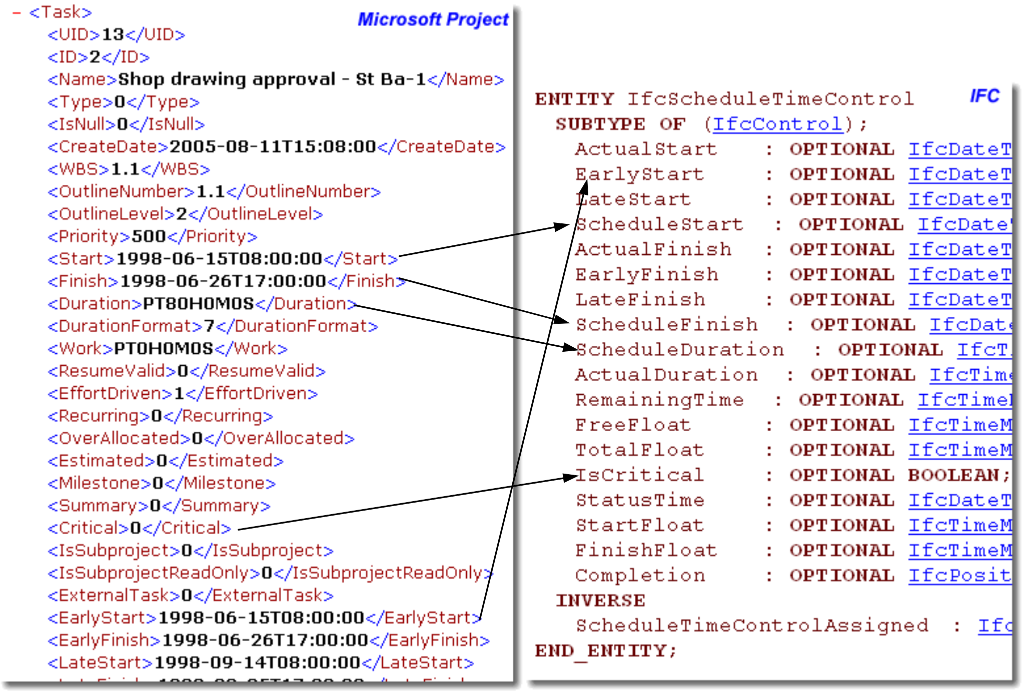


Figure 3: Principle Behind System Architecture

Picking an AEC domain such as scheduling for instance, different applications in the scheduling domain may have different names for representing the same concept. In other words they may have different ways of representing the same information. Typical scheduling concepts include activity/task, duration, precedence, scheduled start times, actual start times, critical path and resources. Fortunately the base concepts in this AEC domain are all accommodated in models such as IFC. Figure 4 shows a partial schema of an activity referred to as <*task*>, stored in Microsoft Project and its corresponding partial representation in IFC. Primavera Project Planner uses the same domain concepts but with different names and representations.



```
- <Task>
    <UID>13</UID>
    <ID>2</ID>
    <Name>Shop drawing approval - St Ba-1</Name>
    <Type>0</Type>
    <IsNull>0</IsNull>
    <CreateDate>2005-08-11T15:08:00</CreateDate>
    <WBS>1.1</WBS>
    <OutlineNumber>1.1</OutlineNumber>
    <OutlineLevel>2</OutlineLevel>
    <Priority>500</Priority>
    <Start>1998-06-15T08:00:00</Start>
    <Finish>1998-06-26T17:00:00</Finish>
    <Duration>PT80H0M0S</Duration>
    <DurationFormat>7</DurationFormat>
    <Work>PT0H0M0S</Work>
    <ResumeValid>0</ResumeValid>
    <EffortDriven>1</EffortDriven>
    <Recurring>0</Recurring>
    <OverAllocated>0</OverAllocated>
    <Estimated>0</Estimated>
    <Milestone>0</Milestone>
    <Summary>0</Summary>
    <Critical>0</Critical>
    <IsSubproject>0</IsSubproject>
    <IsSubprojectReadOnly>0</IsSubprojectReadOnly>
    <ExternalTask>0</ExternalTask>
    <EarlyStart>1998-06-15T08:00:00</EarlyStart>
    <EarlyFinish>1998-06-26T17:00:00</EarlyFinish>
    <LateStart>1998-09-14T08:00:00</LateStart>
```

```
ENTITY IfcScheduleTimeControl
  SUBTYPE OF (IfcControl);
    ActualStart      : OPTIONAL IfcDateT
    EarlyStart       : OPTIONAL IfcDateT
    LateStart        : OPTIONAL IfcDateT
    ScheduleStart     : OPTIONAL IfcDate
    ActualFinish     : OPTIONAL IfcDateT
    EarlyFinish      : OPTIONAL IfcDateT
    LateFinish       : OPTIONAL IfcDateT
    ScheduleFinish    : OPTIONAL IfcDate
    ScheduleDuration    : OPTIONAL IfcT
    ActualDuration    : OPTIONAL IfcTim
    RemainingTime     : OPTIONAL IfcTime
    FreeFloat        : OPTIONAL IfcTimeM
    TotalFloat       : OPTIONAL IfcTimeM
    IsCritical       : OPTIONAL BOOLEAN;
    StatusTime       : OPTIONAL IfcDateT
    StartFloat       : OPTIONAL IfcTimeM
    FinishFloat      : OPTIONAL IfcTimeM
    Completion       : OPTIONAL IfcPosit
  INVERSE
    ScheduleTimeControlAssigned  : Ifc
END_ENTITY;
```

Figure 4: Mapping an Application Schema to IFC Model

Figure 5 shows a detailed architecture of the proposed system. It consists of four layers namely the 1) application layer, 2) native application data store, 3) IFC translators and the 4) data transport layer. This research focuses on developing the IFC translators and the service to transport the data.

Most AEC applications have the *export* or *save as* functionality that allow users to save a project to other structured formats such as HTML, XML, text file, spreadsheet or even to a data base. The purpose of the first translator is to read, analyze and then convert these formats using a set of pre-defined rules into IFC format. The second type of translator also capitalizes on the fact that, similar to the export function most applications also have the import function. This translator converts the IFC-generated file to a format that can be imported by the requesting application again using a set of predefined rules.

The web service component of this proposed architecture is to enable the real-time publishing, transporting and sharing of the translated data. A *requesting* application can use this service to connect to the *supplying* application and transaction can proceed in real time.

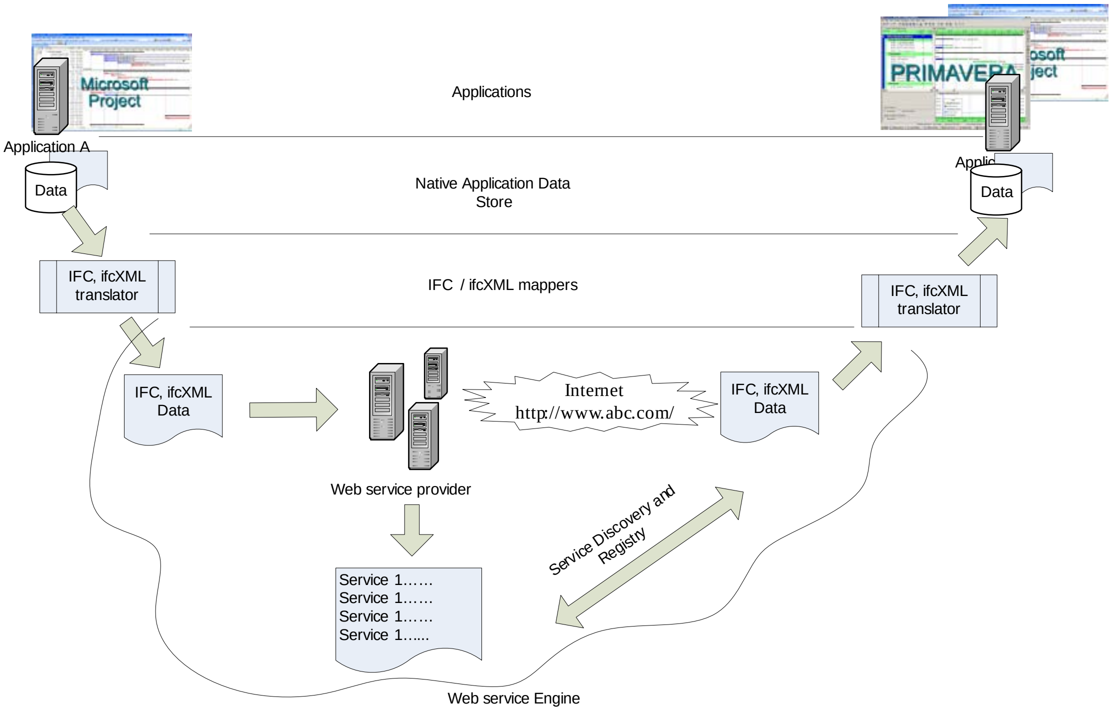


Figure 5: Proposed System Architeture

Currently the translators are being manually developed by comparing and contrasting the various data sources. As with any semantic data integration there is bound to be heterogeneity conflicts some of which have been illustrated in Figure 6. Research thus far with selected applications as a case study, the major problems encountered are entity definition conflicts, more specifically naming conflicts and missing data item conflict. Currently this is being handled by the researchers on a case-by-case basis with the hope that there will be some generalization after a number of translators have been developed to help in the automation of this manual process.

RESEARCH PATH FORWARD

Having identified the initial roadblocks to this research, the next steps are:

- continue with working on the translator with the case by case mapping to IFC model.
- build a service that will recieve and transport the translated data and
- develop and inverse translator from IFC model to the application requiring the service

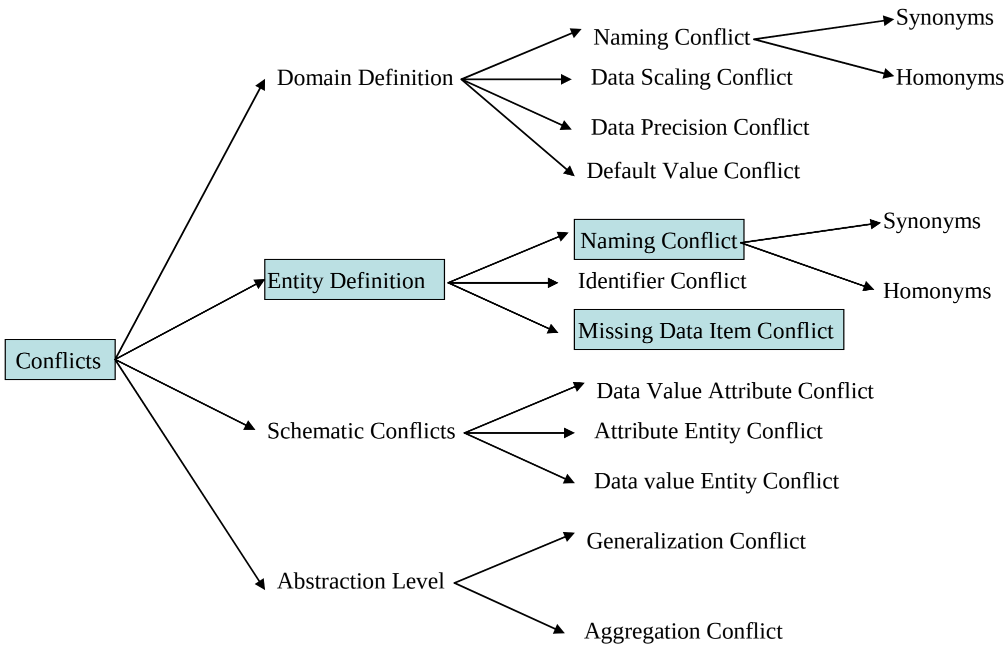


Figure 6: Heterogeneity Conflicts

CONCLUSIONS

This paper reports on an on-going research to develop IFC data-translators for non-IFC AEC applications and consequently share the translated data over the internet with the help of web services. The necessity for such a system was outlined as well as the brief overview of the system architecture. Currently the IFC translator portion of the system for some selected applications is under development and the problems so far encountered have been outlined.

REFERENCES

Alshawi, M., Ingirige, B., (2003). “Web Enabled Project Management: An Emerging Paradigm.” *Construction Automation in Construction* 12(4) 349-364.

Anumba, C.J., Amor, R. (1999). A Survey and Analysis of Integrated Project Databases. Proceedings of Concurrent Engineering in Construction'99, Espoo, Finland, August 25-27, pp. 217-228.

Brandon, P., and Betts, M. (1997). “Creating a framework for IT in construction.” The Armathwaite initiative, The formation of a global construction IT network, Construct IT Centre of Excellence, Salford, U.K.

Constructionweblinks, (2005). U.S. Construction, Architecture and Engineering Industry: An Overview for International Investors http://www.constructionweblinks.com/, Date accessed, 4/4/2005.

Cornick, T. (1990). Quality Management for Building Design, Butterworth Architecture Management Guides, 1990.

Dawood N., Akinsola A., and Hobbs B., (2002). "Development of automated communication of system for managing site information using internet technology" *Automation in Construction*, 11(5) 557-572.

Deng, Z. M., Li H., Tam C. M., Shen, Q. P., Love, P. E. D. (2001). "An application of the Internet-based project management system", *Automation in Construction*, 10(2) 239-246

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

Esposito, D., (2003), Programming Microsoft ASP.NET, Microsoft Press, Redmond

Froese, T., (1996). "Models of Construction Process Information", *Journal of Computing in Civil Engineering*, 10(3), 183-193.

Howard, H.C., Levitt, R.E., Paulson, B.C., Pohl, J.G., Tatum, C.B., (1989). "Computer integration: reducing fragmentation in the AEC industry", *Journal of Computing in Civil Engineering* 3, 18–31.

IAI. (2000). "IFC Technical Guide" International Alliance for Interoperability, http://www.iaiinternational.org/iai_international/Technical_Documents/documentation/IFC_2x_Technical_Guide.pdf, (Date accessed 09 April 2004).

Papazoglou, M. P., Dubray, J., (2004). A Survey of Web service technologies, Technical Report DIT-04-058, Informatica e Telecomunicazioni, University of Trento.

Rönneblad A and Olofsson T (2003). Application of IFC in Design and Production of Precast Concrete Constructions, ITcon Vol. 8, Special Issue IFC - Product Models for the AEC Arena, 167-180.

U.S. Census Bureau (2005). http://www.census.gov/const/C30/total.pdf, Date accessed, 4/4/2005.

Zhiliang M., Heng, L., Shen Q. P. and Jun, Y. (2004). "Using XML to support information exchange in construction projects" *Automation in Construction*, 13 (5), September 2004, 629-637

Zhu Y., Issa, R. R. A. (2005). "Viewer controllable visualization for construction document processing." *Automation in Construction*, (12) 255-269.