buildingSMART alliance &
The Open Geospatial Consortium, Inc.

OGC Technology Office
4899 North Old State Road 37
Bloomington, IN 47408
Telephone: +1-812-334-0601
Facsimile: +1-812-961-2053

Request for Quotation (RFQ)
and
Call for Participation (CFP)

AECOO Testbed – Phase 1
(AECOO-1)
buildingSMART alliance Project 2008-STP-01

Annex B

Testbed Architecture

RFQ Issuance Date: May 2, 2008
Proposal Due Date: May 30, 2008

Table of Contents

1 Introduction and Summary

This Annex is directed toward product development teams, software engineers, systems architects, and rich Internet application developers.

This Annex presents the AECOO-1 Testbed Architecture. This Annex B document is an integral part of the AECOO-1 RFQ/CFP.

The IPTeam used results from previous and ongoing OGC Interoperability Program initiatives, the AECOO Testbed Request for Technology, buildingSMART alliance and buildingSMART International activities, publicly available documentation from related standards initiatives, and elsewhere.

- Section 2 provides an overview of the AECOO-1 development threads
- Section 3 presents the technology standards baseline for AECOO-1
- Section 4 discusses architectural approaches and patterns
- Sections 5-7 present requirements for each of the Testbed's development threads

The AECOO Testbed is a short-term, intensive, multi-participant "spiral engineering" activity to develop, test, and promote the use of open standards for building information. This activity is designed to be iterative and to build on accomplishments of past work completed by previous testbeds, and infused from projects now underway under the umbrella of buildingSMART alliance, buildingSMART International and research accomplishments from industry and academia.

The AECOO Testbed is a joint initiative of the OGC and buildingSMART alliance and sponsored by leading AEC (Architect Engineering Construction) software user organizations. OGC and buildingSMART alliance believe OGC's Interoperability Program and its method for defining market driven interoperability solutions through global IT standards can positively influence technical and market transformation issues that hinder the industry's efficiency and growth.

The architectures presented in this Annex are based upon a collaborative effort between Sponsors of the AECOO Testbed – Phase 1 (AECOO-1), OGC's Interoperability Program Team (IPTeam), and buildingSMART alliance staff.

The OGC portal provides an online Glossary of Terms at the following URL that may be useful to aid in understanding and interpretation of terms and abbreviations contained throughout this RFQ:

http://www.opengeospatial.org/resources/?page=glossary

We also provide a glossary of buildingSMART terms and definitions are provided at:

ftp://ftp.opengis.org/pub/aecoo-08/.

The ftp library is organized by categories and comprises 19 folders. The glossary is located within BIM – GENERAL Folder and titled "BuildingSMART_Glossary_V05.pdf.

1.1 Information Architecture for Interoperable Information Technology

Information Architecture is the fundamental organization of an information system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.

Despite significant efforts to improve engineering practices and technologies, software-intensive systems continue to present formidable risks and difficulties in their design, construction, deployment, and evolution. Recent attempts to address these difficulties have focused on the earliest period of design decision-making and evaluation, increasingly referred to as the architectural level of system development. A key premise of this metaphor is that important decisions may be made early in system development in a manner similar to the early decision-making found in the development of civil and capital facility architecture and construction projects.

The Testbed architecture presented in this Annex is organized into viewpoints. Each viewpoint addresses one or more of the concerns of the system stakeholders – developers, users, integrators and content suppliers. The term viewpoint is used to refer to the expression of a system's information and communication architecture with respect to a particular design concern.

1.2 Viewpoints in the AECOO Architecture

The Reference Model of Open Distributed Processing, ISO/IEC 10746-1 to ISO/IEC 10746-4 is the approach used for defining the Testbed architecture; it is based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification.

There are five viewpoints defined in RM-ODP:

- *Enterprise viewpoint:* Software systems must support the enterprise or business. Whether they are initially driven by an organizational need or they arise from a technology opportunity, software systems exist to support the business of that organization. The Enterprise Viewpoint, therefore, defines the business problems and associated processes in clear and consistent terms that can be realized as system requirements and articulated the same way throughout the other viewpoints.
- *Information viewpoint:* A viewpoint on the system and its environment that focuses on the semantics of the information and information processing that is to be performed to address the identified problems.
- *Computational viewpoint:* A viewpoint on the system and its environment that presents the system as a functional decomposition of services that interact at interfaces. This viewpoint captures the details of these components and interfaces without regard to distribution.
- *Engineering viewpoint:* A viewpoint on the system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system.
- *Technology viewpoint:* A viewpoint on the system and its environment that focuses on the choice of technology in that system.

The first three viewpoints are provided in this document as a guideline for preparation of responses. The Engineering and Technology viewpoints are the work to be accomplished during the course of the Testbed.

2 AECOO-1 Initiative Threads

2.1 OGC's Interoperability Program

The OGC operates an Interoperability Program that is a global, hands-on and collaborative rapid prototyping program designed to develop and deliver proven candidate specifications into OGC's Specification Program that can then be formalized for public release as standards. In OGC's Interoperability Initiatives, international technology developers and providers team together to solve specific geo-processing interoperability problems posed by the initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments, and interoperability support services – all designed to encourage rapid development, testing, validation and adoption of open, consensus based standards specifications.

2.2 Adapting OGC's Interoperability Program for buildingSMART Standardization

The buildingSMART alliance and OGC have agreed to the Interoperability Program approach to encourage broad international participation on well-defined sets of AECOO community problems. It is expected that testbeds dealing with AECOO requirements will yield accelerated alignments of industry on open standards solutions. The AECOO Testbed, more than anything else, is positioned to help the AECOO community understand how to plan and conduct successful standards development using market driven approaches; to bridge cooperation among and between standards bodies whose mission crosses boundaries for information sharing and dissemination; and to achieve outcomes greater than what can be achieved alone.

OGC's goal is to develop joint initiatives with the AECOO community because we believe collectively we all can be more efficient in the future in addressing issues of geospatial and AEC information convergence.

For AECOO-1 the policies and procedures that define testbed operation are found here:

http://www.opengeospatial.org/ogc/policies/ippp

In October of 2007, the OGC issued a call for sponsors for a Testbed to advance open frameworks for interoperability in the AECOO industry. A series of meetings were conducted during the period November 2007 to January 2008 with Sponsors to review technical baseline documents, and identify Testbed requirements.

In February of 2008, buildingSMART alliance and OGC jointly released a Request for Technology (RFT) to obtain additional input from the industry at large. The testbed architecture, and the following work threads, result from those efforts, and describe a set of cohesive AECOO-1 work items that can feasibly be accomplished in the given time frame.

In this initial Testbed, Sponsors seek to advance standards in the following areas:

1. Quantity Takeoff for Cost Estimation (QTO)
2. Building Performance and Energy Analysis (BPEA)
3. Communications, Project Delivery, and Decision Support (CPD)

An introduction to each of these threads is given below, followed by a detailed discussion of the architectural implications of the initiative threads.

2.3 Quantity Takeoff for Cost Estimation (QTO)

The role of the cost estimator is to facilitate the design process by systematic application of cost criteria so as to maintain a sensible and economic relationship between cost, quantity, utility and appearance which thus helps in achieving the client's requirements within an agreed budget. It commences when little is known about the project other than its overall size, probable location and type (or design intent). Ideally, building modelers and cost estimators and specifiers need to continually work together and share relevant information throughout their processes and individual activities.

Cost estimators and designers undertake cost modeling progressively throughout the design and construction of a project, and make use of the best information that is available at the time. Since this may be unbalanced, in that some areas may be more accurate than others, metadata is required as to the level of confidence there is in each number. It starts at the earliest stage as historically based when information may be available only about the type of building required together with its expected overall size and location. As more detail is added to the design, cost modeling can be refined based on area measurement of spaces typically using a Uniformat based approach, until estimates can be developed based on a more complete knowledge of the elements to be incorporated within the project in a MasterFormat based estimate.

Unfortunately, there has been no simple correlation between building elements modeled in early design stages, and building elements modeled much closer to construction. However, with the new version of Uniformat a cross over to MasterFormat will soon be available. This non-hierarchical level of detail issue complicates the process of quantity takeoffs at different stages of design. This necessitates the need for metadata so that an estimating continuum can be established and not perpetuate the cost information disconnects that currently plague the process

The sponsors recognize there is considerable work to do at the conceptual level to resolve information exchange issues in the design-quantify takeoff-cost estimation continuum. The goal of this thread of the Testbed is to enable better cost estimates by achieving better interoperability in geometry and design level of detail for quantity takeoffs.

2.4 Building Performance and Energy Analysis (BPEA)

Connecting building performance[1] and energy analysis with building design is now a critical step in planning a building. Interactions between the architect and the engineers, the owner, the construction company and their HVAC&R contractor are critical to the success in the design, construction and operation of the building; there is simultaneous demand for applicable energy analysis tools for design that can connect with cost estimating.

This Testbed will focus on two information flows that define the interactions between the stakeholders. First, we will explore interoperability between the expression of architectural design intent as captured by a BIM and the building's thermal loads. Second, we will explore interoperability issues between architectural models and annual energy performance with focus on interoperability between information within a BIM and EnergyPlus and/or DOE2.

2.5 Communications, Project Delivery, and Decision Support (CPD)

There are numerous types of project management, narrative, legal, contractual, and financial communications that take place between architects, owners and construction companies during the planning, design and construction phases. Many of these communications are the information building blocks for decisions, decision tracking, submittal progress, cost and budget issues, and follow the trails of questions and answers. Many of these communications may refer, or otherwise relate, to a building information model's general project properties (e.g. geospatial origin, address, building type, building infrastructure, etc.), geometry, and/or object metadata.

This thread seeks to increase interoperability in two primary areas:

1. Define and develop ways in which existing decision support, communications, and project management tools can interface with non-native software and information more efficiently.

---

[1] The term performance is very broad. There is energy performance, as well as daylighting performance, thermal comfort performance, circulation performance, etc.). To make things even more complicated, one can refer to "thermal performance" as encompassing energy, daylighting, and thermal comfort. Daylighting and thermal comfort, both of which address "energy flows", are generally not considered by industry to fall under "energy analysis". There are separate programs/engines for evaluating energy (e.g. EnergyPlus and DOE2), daylighting (e.g. Radiance/AGi32), and thermal comfort (computational fluid dynamics, or CFD), though EnergyPlus can address daylighting and thermal comfort in a simplistic manner. This RFQ focuses on interoperability with energy simulation tools that evaluate peak load and annual energy performance.

2. Define and develop decision support, communications, and project management requirements for the work done in other threads that cover specific business processes

3 AECOO-1 Standards Baseline

3.1 buildingSMART standards Baseline

The buildingSMART standards baseline can be accessed at the URLs provided below or at: ftp://ftp.opengis.org/pub/aecoo-08/. The latest IFC releases can be obtained at the urls provided. The ftp library is organized by categories and comprises 19 folders.

3.1.1 Industry Foundation Class (In IFC Folder or at the URLs indicated)

- IFC 2x3 at http://www.iai-tech.org/groups/msg-members/news/ifcXMLl2x3-final-published
- IFC2x3 Technical Corrigendum 1 - documentation and constraint only corrigendum to update of IFC2x3, published in Jul 2007 at http://www.iai-tech.org/products/ifc_specification/ifc-releases/ifc2x3-tc1-release/summary
- IFC2x3g at http://www.iai-tech.org/downloads/ifc/ifc2x3g?searchterm=ifc2x3g
- ifcXML for IFC2x Edition 3 Technical Corrigendum 1 at http://www.iai-tech.org/products/ifc_specification/ifcxml-releases
- PM-4 Quantity take-off from the IFC model at http://www.iai-tech.org/projects/ifc_extension_projects/current
- IDM for Costing and Energy at http://idm.buildingsmart.no/confluence/display/IDM/home
- Using IFD at http://dev.ifd-library.org/index.php/Ifd:Ifc_naming_convention_in_Ifd and http://dev.ifd-library.org/index.php/Ifd:ImportExportIfcPropertyset

3.1.2 IDM and MVD Process Methods for Information Organization

- IDM for Costing and Energy
- Draft ASHRAE Guideline 20P– XML Definitions for HVAC&R
- Workflow Management Coalition Process Definition Interface – XML Process Definition Language
- Architectural to Structural Model View for IFC2x3
- Model View Definition Architectural to Structural
- Model View Definitions Quantity Take-off Levels 1,2, and 3
- Model View Definition Architectural Design to Thermal Simulation
- Architectural Design to Thermal Simulation Model View
- IFC Project PM 4: Quantity take-off from the IFC model
- IDM Process Maps for HVAC, Piping and Electrical Engineering
- International Framework for Dictionaries API

3.1.3 General References and Literature for Building Information Management

- US GSA BIM Guide
- US GSA P120 and P100 – Public Buildings Project Estimating and Facilities Standards
- US GSA Richard Bolling Federal Office Building Scope of Work (SOW) for BIM Services
- Arto Kiviniemi, Requirements Management Interface to Building Product Models
- Thomas Froese, Kevin Yu, Kathleen Liston, Martin Fischer, System Architectures For AEC Interoperability; Published in Construction Information Technology 2000
- Integrated Project Delivery: A Guide, 2007 Version 1; American Institute of Architects and AIA California Council
- Staub-French, Sheryl, Feature-based Construction Cost Estimating
- Staub-French Sheryl; Fischer, Martin; Kunz, John, A Feature Ontology to Support Construction Cost Estimating
- Staub-French Sheryl; Fischer, Martin; Kunz, John, An Ontology For Relating Features of Building Product Models with Construction Activities to Support Cost Estimating
- The Pankow Report: BIM and Interoperability for Precast Concrete
- European and European Commission Funded Projects and Platforms:
- European Construction Research Network (E-CORE)
- European Construction and Technology Platform
- Open Information Environment for Knowledge-Based Collaborative Processes throughout the Lifecycle of a Building (INPRO)
- Standards for Innovative Construction and FM
- The buildingSMART Glossary of Terms
- United States Building Information Modeling Standard
- CSI: OmniClass, UniFormat and MasterFormat
- NIBS: Building Information Modeling (BIM) For Precast Concrete (Pankow)
- IAI: DRAFT Information Delivery Manual: Guide to Components and Development Methods
- AIA: Integrated Project Delivery Guide
- AGC Contractors Guide to BIM
- CURT, Collaboration, Integrated Information and the Project Lifecycle in Building Design, Construction and Operation
- NIST General Buildings Information Handover Guide: Principles, Methodology and Case Studies

3.2 OGC standards Baseline

The OGC standards Baseline, at any point in time, is the set of all Adopted Specifications plus all other technical documents that have been made available to the public by the OGC Technical and

Planning Committees. The standards Baseline includes all member-approved standards and best practices documents. These specifications and other documents are available at:

http://www.opengeospatial.org/specs/?page=baseline

3.2.1 OpenGIS Reference Model (ORM)

The OpenGIS Reference Model (ORM) provides an architecture framework for the ongoing work of the OGC. Further, the ORM provides a framework for the OGC Standards Baseline. The OGC Standards Baseline consists of the member approved Implementation/Abstract Specifications as well as for a number of candidate specifications that are currently in progress. The ORM is available on the OGC Member's Portal here:

http://portal.opengeospatial.org/files/?artifact_id=3836

The ORM is a living document that will be revised on a regular basis to continually and accurately reflect the ongoing work of the Consortium. It is encouraged that respondents to this RFQ understand the concepts that are presented in the ORM.

The structure of the ORM is based on the Reference Model for Open Distributed Processing (RM-ODP). This Annex of the AECOO-1 RFQ will deal with the upper three views; Enterprise, Information, and Computational. Each thread of the initiative will be described in the annex using these views.

3.2.2 OGC Standards

Several documents were approved at the recent OGC TC/PC meeting and are being processed for public release. These documents will become available over the next few weeks. If a proposer needs a specific document that has not yet been published to the public link above, contact OGC (techdesk@opengeospatial.org).

The Table 1 lists the approved OGC Specifications that are relevant to the Testbed.

Table 1: Relevant OGC Standards

Title	Version	Document #	Date
Coordinate Transformation Service	1.0	01-009	2001-01-12
Web Coordinate Transformation Service	0.4.0	07-055r1	2007-10-09
Access Control & Terms of Use (ToU) "Click-through"	1.0.0	07-004	2006-05-09
Geolinked Data Access Service	0.4.0	04-010r1	2004-05-04
Web 3D Service	0.3.0	05-019	2005-02-02
Web Processing Service	1.0.0	05-007r7	2007-06-08
Geography Markup Language (GML)	3.2.1	07-061	2007-08-27
Web Feature Service (WFS)	1.1	04-094	2005-05-03
Request for Comment on CityGML Standard	1.0.0	08-007	2008-01-20
KML 2.2 Reference	2.2	0 7-113r1	2007-11-23

3.2.3 OGC Best Practices

Best Practice Documents contain discussion of best practices related to the use and/or implementation of an adopted OGC document and for release to the public. Best Practices Documents are an official position of the OGC and thus represent an endorsement of the content of the paper. These Best Practice Documents have been made available at: http://www.opengeospatial.org/standards/bp.

Table 2: Relevant OGC Best Practices

Title	Version	Document #	Date
(BXML) Encoding Specification	0.0.8	03-002r9	2006-01-18
Definition identifier URNs in OGC namespace	1.1.0	06-023r1	2006-08-08
OpenGIS® web services architecture description	0.1.0	05-042r2	2005-11-21
Specification best practices	1.0.0	06-135r1	2007-01-29

3.2.4 OGC Discussion Papers Baseline

OGC Discussion Papers are documents that present technology issues being considered in the Working Groups of the Open Geospatial Consortium Technical Committee. Their purpose is to create discussion in the geospatial information industry on a specific topic. These papers do not represent the official position of the Open Geospatial Consortium nor of the OGC Technical Committee. These discussion papers have been made available at: http://www.opengeospatial.org/standards/dp.

Table 3: Relevant OGC Discussion Papers

Title	Version	Document #	Date
OGC Web Services Architecture for CAD GIS and BIM	0.9	07-023r2	2007-05-16
A URN namespace for the Open Geospatial Consortium (OGC)	2	06-166	2007-01-30
OGC Web Services (OWS) 3 UGAS Tool	0.0.3	05-118	2006-04-28
Schema Maintenance and Tailoring	0.0.7	05-117	2006-05-02
OWS-4 GeoDDS Mass Market (formerly GeoRSS) Interoperability Program Report	0.0.1	07-004	2007-05-07
GeoDRM Engineering Viewpoint and supporting Architecture (OWS-4 GeoDRM Interoperability Report)	1.0.0	06-184	2007-02-09
GML Implementation of simple solids, planes and lines	0.1.0	07-001	2007-01-15
Trusted Geo Services IPR	1.0.0	06-107	2007-01-22

4 AECOO-1 Cross-Thread Architecture

4.1 Cross-Thread Information Viewpoint

The Information Viewpoint for the AECOO Test Bed specifies the modelling approach for all categories of information the architecture deals with, including their thematic, spatial and temporal characteristics as well as their meta-information. This section details a number of information models and encodings that participants may employ in the Testbed as appropriate.

4.1.1 Overview

Actors in the AEC world are converging on standards for structuring and exchanging highly detailed information about buildings across the building project lifecycle. The development of BIM standards are being coordinated by the buildingSMART International through their development of the exchange specification, Industry Foundation Classes (IFC), and other ontological and narrative requirements as they are defined in Information Delivery Manuals (IDM) and International Framework Dictionaries (IFD). These classification models and ontologies are the platform for developing Model View Definitions (MVD) that are designed to provide software developers and ultimately users with composite viewpoints of the data according to their domain specialty or government agency requirement or phase of the project. Each of these is explained below.

4.2 Industry Foundation Classes (IFC)

The Industry Foundation Classes (IFC) system is a data representation standard and file format used to support data exchange and sharing of project life-cycle data of the construction facilities. It enables electronic communication among different disciplines that participate the project life-cycle, such as architecture, building services, building automation, structural analysis and engineering, quantity take-off and cost estimation, construction planning and scheduling, project handover and facilities management. A main realization if the ability to provide architectural CAD users with the ability to exchange data between complementary applications such as building services and structural modeling CAD applications and estimating, energy calculation, or facility management tools. The IFC specification is developed by the buildingSMART International

http://www.iai-international.org/Model/IFC(ifcXML)Specs.html

The Industry Foundation Classes for GIS (IFG) project intended to provide support for geographic information within the IFC model to support the integration of CAD and GIS data. A task of the IFG was data transformation between IFC and GML. The outcome of this project will form an integral part of the upcoming IFC2x4 specification. A description of the IFG project is available here:

http://ce.vtt.fi/iaiIFCprojects/ShowProjectInfo.jsp?project_id=89&status_id=2

Table 4: Selected IFC2 IFC2x Specification Schemas Relevant to the Testbed

IFC Data Model	Description
IfcKernel	The most abstract part within the IFC architecture. It captures general constructs that are founded by their different semantic meaning in common understanding of an object model, like object, property and relationship. Those are then specialized into non-AEC/FM specific constructs, like product, process, control and resource, which form the main entry points for the next level, the Core Extension layer.
Ifc Product Extension	Further, specialize the concepts of a (physical) product, i.e. a component likely to have a shape and a placement within the project context. The product information is provided for individual product occurrences as subtypes of IfcProduct, and for common specific product types as subtypes of IfcTypeProduct. Both definitions are rooted in supertypes provided within the IfcKernel.
Ifc Geometry Resource	The schema IfcGeometryResource defines the resources used for geometric representations. The primary application of this resource is for representation of the shape or geometric form of a product model.
Ifc Representation Resource	This schema defines the representation of shape and topology as important definitional properties for products defined within the IFC Object Model. The representations characterize certain properties of a product, and zero, one, or many of those properties can define any product.

The IFC specification is structured in various schemas, however current IFC implementations are based on a long-form schema (i.e. where all schemas are combined into a single form). The definition of subsets of the long-form schema suitable for a particular viewpoint, or exchange scenario, is a Model View Definition.

IfcKernel defines the most general definition of semantics in IFC. The IfcKernel serves a role for IFC that is similar to the General Feature Model for GIS. The GFM can be found in GML and

ISO 19109. For example, in the IfcKernel Kernel is most abstract class is IfcRoot, which is the super type of IfcObjectDefinition, IfcPropertyDefinition and IfcRelationship.

IFCs have several methods to extend the basic types of the IFC specification to specific industry domains: IfcTypeProduct and Domain Data Models.

- The product type (IfcTypeProduct) defines a list of property set definitions of a product and an optional set of product representations. It is used to define a product specification (i.e. the specific product information that is common to all occurrences of that product type). A product type is used to define the common properties of a certain type or style of an object that may be applied to instances of those products to assign a specific style to them. Product types may be exchanges without being already assigned to products.
- Several Domain Data models are defined in the non-platform part of the IFC specification, e.g., Architecture Domain, Facilities Management Domain, and Structural Analysis Domain. For example, the IfcConstructionMgmtDomain schema defines concepts in the construction management (CM) domain. Together with the IfcProcessExtension and IfcSharedMgmtElementschema it provides a set of models that can be used to exchange information between construction management applications.

An area for study is to apply the previously developed GML Application Schema tools to the support development of domain specific and business process application practices and product specific extensions of IFC.

4.3 IFC and ifcXML

In 2004, buildingSMART International began a project to recast IFC from STEP to XML. A methodology for generating .xsd from EXPRESS definitions was developed in collaboration with other industry groups in ISO TC184/SC4, it became on source for developing ISO10303-28 "XML representation of EXPRESS schemas and data". It is thought that ifcXML is suitable for a variety of uses like those being addressed in AECOO-1.

IFC or ifcXML are established as requirements for testing. Specification access for IFC STEP and ifcXML are provided in Section 3.1.1. Participants have the choice of each or proposing a hybrid.

4.4 Conceptual Information Models

4.4.1 Geometry Models

One of the core information modeling issues in this domain relates to how geometry is represented. Geometry is concerned with questions of size, shape, and relative position of figures and with properties of space. In this field, there are various approaches to using geometry to model the key aspects of objects. The IFC 2x model defines three different types of solid model representations:

1. Boundary model representation (b-rep)
2. Swept area solid representation
3. Constructive solid geometry (CSG)

In addition to geometric set representation (set of 0D, 1D, and 2D geometric representation items) and non-solid 3D representations (face based and shell based surface models). The geometric model of IFC is routed in ISO10303 part 42 and part 43 that also form part of all STEP standards.

OGC's GML presents another approach to geometry, which is defined in GML 3 in the following schemas:

- geometryBasic0d1d.xsd
- geometryBasic2d.xsd
- geometryPrimitives.xsd
- geometryAggregates.xsd
- geometryComplexes.xsd

While there have been comparisons between the IFC and GML geometry models the tradeoffs between the two are unclear. For example, GML has better support for spatial reference systems, but the need for spatial reference systems in the engineering context is debatable (see Liebich, Comparison of gml3.0 and IFC2x(2)). Clearly, there is a role for different geometry models in different applications, but therefore the ability to seamlessly move data between models is a key aspect of interoperability.

4.4.2 Spatial Reference Systems

The OpenGIS® Abstract Specification Topic 2, Spatial Referencing by Coordinates, defines Coordinate Reference Systems (CRS) using UML. Topic 2 is the basis of a joint project with ISO TC211 to revise *ISO 19111:2003, Spatial referencing by coordinates*. OGC's GML implements the CRS abstract model in OGC Topic 2 in XML. GML is also a joint project with ISO TC211 where it is known as project ISO 19136. Consistent with Topic 2, GML defines six schema documents for encoding the definitions of coordinate reference systems and coordinate operations, explaining their contents, structure, and dependencies. Those six schemas as defined in Clause 12 of the GML specification are:

- referenceSystems.xsd
- coordinateReferenceSystems.xsd
- datums.xsd
- coordinateSystems.xsd
- coordinateOperations.xsd
- dataQuality.xsd

These schema documents are available at http://schemas.opengis.net/gml/3.2.1/.

A fundamental concept of *OGC Topic 2 (ISO 19111)* is that of a Coordinate Reference System (CRS) which is composed of a Coordinate System and a Datum. Examples of Coordinate Reference Systems include: Geographic, Geocentric, Projected, and Engineering. Examples of Coordinate Systems include: Cartesian, ellipsoidal, spherical, cylindrical, and linear. A datum defines the position of the origin, the scale, and the orientation of a coordinate reference system. Examples of datum include geodetic, vertical, and engineering. Coordinate systems and datum are constrained for use with only certain CRSs. For example the Cartesian coordinate system is used only with Geocentric, Projected, and Engineering CRSs.

For IFC information models, a fundamental assumption is that all geometry shall be defined in a right-handed rectangular Cartesian coordinate system with the same units on each axis. A common scheme has been used for the definition of both two-dimensional and three-dimensional geometry. Points and directions exist in both a two-dimensional and a three-dimensional form, these forms are distinguished solely by the presence, or absences, of a third coordinate value. Complex geometric entities are all defined using points and directions from which their space dimensionality can be deduced. (See IFC2x Edition 3: IFCGEOMETRYRESOURCE). IFCs do not directly represent datum. The IfcSite object does, however, contain reference to latitude, longitude and other methods to geo-locate the site.

4.4.3 Metadata

Metadata is used to facilitate the understanding, use and management of data. The metadata required for effective data management varies with the type of data and context of use. One type of metadata is provenance metadata, which describes the person or organization responsible for the data, and details about its creation. Metadata may also be used for providing fuller, narrative descriptions of the meaning of property, or attribute names. Another class of metadata is often machine-generated, and can capture the history of changes to the data, or computational operations performed on the data to move it from one state to another.

In IFC, the main entity connected with provenance issues is the ifcOwnerHistory entity, which has sub-entities such as ifcAuditTrail, ifcApplication, ifcOrganization, ifcAddress, ifcPersonAndOrganization, and ifcPerson.

ISO 19115:2003 defines the schema required for describing geographic information and services. It provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data. *ISO 19115:2003* defines: mandatory and conditional metadata sections, metadata entities, and metadata elements; the minimum set of metadata required to serve the full range of metadata applications (data discovery, determining data fitness for use, data access, data transfer, and use of digital data); optional metadata elements – to allow for a more extensive standard description of geographic data, if required; a method for extending metadata to fit specialized needs.

ISO 19115 does not provide any guidance on how the metadata records should be built and formatted. To address this need an additional standard, *ISO 19139*, was undertaken to create an XML schema that prescribes the format of the metadata record. The *ISO 19139*standard incorporates metadata elements referenced, but not defined, in *ISO 19115* such as the entity and attribute descriptions addressed by the *ISO 19109* geospatial data standard *ISO 19139*provides an encoding schema for describing, validating, and exchanging metadata about geographic datasets, dataset series, individual geographic features, feature attributes, feature types, feature properties, etc.

4.5 IFC Model View Definitions (MVD)

IFC is a generic container for structuring and exchanging building information specific applications are expected to use subsets of the IFC schema defined in Information Delivery Manuals (IDMs). Information Delivery Manuals are human readable specifications of information needs. These are instantiated in a particular subset of the full IFC specification as Model View Definitions (MVD). Specific information for describing and encoding MVDs and IDMs is provided in Heitanen, 2006.[2]

IFC Model View Definitions document how the IFC Model Specification is applied in the data exchange between different application types. They include:

- **Format:** The type of data that needs to be captured and how that data is structured
- **Content:** The data that is needed in a specific case. For example the IFC Schema is content that is captured using the EXPRESS format and an IFC Model View Definition is content that is captured using the IFC Model View Definition format.

[2] Model View Definition Format; Jiri Heitanen, International Alliance for Interoperability, http://www.iai-international.org/software/mvd.shtml.

- **Process:** The roles and responsibilities of different involved parties, for example how a model view definition becomes official and how certification is organized.
- **Tools:** The tools used for creating content, e.g. Process Maps and Exchange Requirements, and managing the process of creating content. Tools are highly important, but the format itself must be independent from any specific tools.
- **IDM (Information Delivery Manual):** provides the integrated reference for process and data required by BIM by identifying the discrete processes undertaken within building construction, the information required for their execution and the results of that activity. It will specify:
    - Where a process fits, and why it is relevant
    - Who are the actors creating, consuming and benefiting from the information
    - What is the information being created and consumed
    - How the information should be supported by software solutions
- **IFD (International Framework of Dictionaries):** describes what the content and relationships of objects are in terms of properties, units and values and also help to serve definition of the contents IDM.

Model View Definitions are expected to be integral to the Testbed in that they describe the marriage of information, software, and business processes to achieve a particular business goal. This topic is discussed in more detail below, in thread-specific Information Viewpoint sections.

4.6 CityGML

CityGML is a common information model for the representation of 3D urban objects. It defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic and appearance properties. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. This thematic information goes beyond graphic exchange formats and allow to employ virtual 3D city models for sophisticated analysis tasks in different application domains like simulations, urban data mining, facility management, and thematic inquiries.

CityGML is realized as an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is implemented as an application schema for the Geography Markup Language 3 (GML3), the extensible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211. CityGML is an open standard and therefore can be used free of charge.

CityGML does not only represents the graphical appearance of city models but especially takes care of the representation of the semantic response, thematic properties, taxonomies and aggregations of Digital Terrain Models, sites (including buildings, bridges, tunnels), vegetation, water bodies, transportation facilities, and city furniture. The underlying model differentiates five consecutive levels of detail (LOD), where objects become more detailed with increasing LOD regarding both geometry and thematic differentiation.

CityGML files can – but are not required to – contain multiple representations for each object in different LOD simultaneously.

4.7 Cross-Thread Computational Viewpoint – Service Architecture

IFC and Web Services have begun to be implemented within the digital tools of the AEC professional. Several leading CAD applications now import and export IFC data to *various simulation, quantity estimation and management applications. A framework is required that binds individual data-sources to enable efficient searching and exchange of AEC data across unrelated organizations and countries. This framework should implement the technologies already present within emerging AEC software tools and information systems. Just as Web-based search engines spurred the growth of the Internet, such a framework would foster information exchange and encourage binding of the AEC knowledge base.*

Applying web services within the AEC industry
D. Harrison, M. Donn and H. Skates
School of Architecture, Victoria University

Since Web Services, by its very nature, promotes interoperability, it will be particularly critical to the IFC interoperability effort. IFC compatible applications can then access this data in real time and communicate dynamically with each other, using the Web Services Technology. The power of the building information model can be substantially extended with this concept. The building data will no longer just reside in a file that has to be exchanged between applications, with confusion about which is the most recent version; instead, it will be live, up-to-date building data that reflects the latest state of the model. So, for example, a change made by the architect to the building design will be immediately seen by an energy-analysis application working with the model, without a manual IFC export from the architectural application and a manual IFC import into the energy analysis application.

The Web Services Technology and its Implications for AEC
AECbytes Newsletter #8, April 29, 2004

Sponsor sentiments and industry feedback from the AECOO Testbed Request for Technology confirm the view that the industry must move from its current, file-based information exchange paradigm to a more web-based, service oriented approach. In fact, many newer firms, and newer products from traditional firms, are already moving to embrace this approach. This Testbed, therefore, will focus on the exchange of information through web protocols and standards. While we adopt a service-oriented architecture (SOA) approach, this in no way implies, that we adopt or endorse the formal SOA "stack" of technologies (e.g. UDDI, WSDL, SOAP, WS-*) for this Testbed.

SOA is a set of business, process, organizational, governance and technical methods to reduce or eliminate frustrations with IT, quantifiably measure IT's business value, and create an agile business environment for competitive advantage.

From a slightly technical perspective: SOA is a way for different computers, from different vendors, with different programs, from different functional areas of the business to (or externally to customers, suppliers or vendors) intelligently talk and exchange data with each other in a highly automated, efficient manner.

See the Wikipedia entry on service-oriented architecture for an introduction to the topic and the distinctions between SOA as a concept, and various approaches to SOA:

http://en.wikipedia.org/wiki/Service-oriented_architecture

4.7.1 A Business-centric View of Information Exchange

The purpose of aligning the information technology for AECOO around services is an economic business case. By aligning the firms business with interactions through services, the IT infrastructure has the advantages of agility and reuse. Return on investment and cost savings for pursuing a SOA has been demonstrated in multiple industries.[3] SOA has a history of successful implementations. The reason the term SOA has become so visible now is because of SOA implementation with web services. Web services are important because it breaks the proprietary barrier between vendors and software programs. Major vendors have agreed to standards that allow their hardware and software to share information and data.

Enterprises have multiple applications that are being built independently, with different languages and platforms. These enterprises need to share data and process shared information in a responsive way. The challenge is Application Information Interoperability.

In the popular discussion, people have tended to jump to the conclusion that there is a single BIM model in the middle of the process. This reflects an early vision for BIM in the AECOO industry. Unfortunately, it has been hard to find ways to make this work. As a result, we have evolved a pattern of using multiple models, generally divided according the professional discipline areas of responsibility.

The use of multiple models creates an interoperability issue, namely, how to move object data between the models or make them usable as if they were one model.

In this context, each office (or even each project team) defines a strategy – even if they don't really realize that this is what they are doing and what the implications are. It may involve a combination of the following options.

Project teams Firm "A" (Architecture Firm) and Firm "C" (Construction Firm) are working to define a conceptual and criteria design for a building. Firm A has standardized on Tools "TA". Firm C has standardized on Tools "TC". TA and TC exist in mutually exclusive network space constrained by their organizational computing environments and security regimes. TA is unaware of TC, and TC is unaware of TA.

The computation viewpoint defines methods to meet the following objectives:

1. Interoperability between Firm A and Firm C can be achieved by transferring model information files.
2. Firm A and C shall communicate with each other by exchanging models created with their respective authoring tools, TA and TC.
3. Firm A and C shall exchange Models that conform to industry standards.
4. Firm A and C shall assimilate the exchanged models from one another in their respective Tools TA and TC.
5. Protect the intellectual property rights of Firm A and Firm C on their proprietary tools TA and TC.
6. Integrating applications shall not be coupled to underlying hardware and software technologies.
7. Enterprises shall agree on fixed, strongly typed integrating data formats (preferably based on MVD).
8. Enterprises shall exchange relatively small fragments of critical time sensitive information and larger chunks of batch data.
9. Integrating applications shall exchange information asynchronously.

---

[3] "Service Oriented Architecture - Business Value Proposition; Customer Version 2, "October 2004, Randy Lange, IBM Software Group Enterprise Integration Solutions, http://www.ibm.com/webservices/eis.

4.7.2 File Transfer

Current implementations of IFC-based integration rely almost exclusively on the exchange of IFC files. This mode of transferring data is simple and effective. However, IFCs are limited in their ability to manage a large pool of shared project information that is accessed concurrently by many users, or to enable transactional forms of data exchange between project parties and applications. So, an approach that collects the common data and their semantics in distributed databases and offers a variety of data management services are required for team based, integrated model-based systems.

Integrating applications, also known as enterprise end points, shall produces data files for others to consume, and may in turn, consume files others have produced.

File transfer can be achieved via one of the following methods:

- Direct access to secured location on collaborating enterprise network. This can be a secured FTP location or a location accessible via VPN.
- Indirect transfer to a secured location where the receiving and transferring enterprise can exchange integrating data files.

4.7.3 Service-Oriented Architecture Overview

4.7.3.1 Basic Definitions and Standards

Service Oriented Architecture (SOA) is defined as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. An SOA approach provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations."[4]

- **Service:** The means by which the needs of a consumer are brought together with the capabilities of a provider.
- **Service Provider:** An entity (person or organization) that offers the use of capabilities by means of a service.
- **Service Consumer:** An entity that seeks to satisfy a particular need through the use capabilities offered by means of a service.
- **Service Description:** The information needed in order to use, or consider using, a service.
- **Service Interface:** The means by which the underlying capabilities of a service are accessed.
- **Offer:** An invitation to use the capabilities made available by a service provider in accordance with some set of policies.
- **Visibility:** The capacity for those with needs and those with capabilities to be able to interact with each other.
- **Behavior Model:** The characterization of (and responses to, and temporal dependencies between) the actions on a service.
- **Information Model:** The characterization of the information that is associated with the use of a service.

[4] OASIS Reference Model for Service Oriented Architecture 1.0 (SOA-RM, 2006).

- **Semantics:** A conceptualization of the implied meaning of information that requires words and/or symbols within a usage context.

Web Services are an implementation of an SOA using the standards of the web. The World Wide Web Consortium has defined a Web Service as, "…a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols."[5]

Web services are based upon a set of mature standards. The basic Web Services Platform consists of the following standards:

- **URI:** A Uniform Resource Identifier (URI) is a compact string of characters used to identify or name a resource. The main purpose of this identification is to enable interaction with representations of the resource over a network.[6]
- **HTTP:** Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol for many tasks beyond its use for hypertext, such as name servers and distributed object management systems.[7]
- **XML:** The Extensible Mark-up Language (XML) is a general-purpose mark-up language. Its primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.[8]

4.7.3.2 Publish-Find-Bind Pattern

A key the agility of a SOA is the use of the Publish-Find-Bind pattern. This pattern of interaction between several services is vital to the loose coupling between service consumers and service providers. Services are made available for any authorized consumer. The visibility of the service to the consumer occurs at the time that the consumers express interest in such a service. The key elements of the pattern are:

- Service providers advertise their resources (**publish**)
- Service consumers discover resources that they need at run-time (**find**)
- Service consumers and their applications access and exercise resources at run-time (**bind**)

Figure 1 shows a basic, generic service interaction pattern for publish-find-bind. An organization wanting to share information in an SOA does two things: create a data access service, and **publish** (step 1: PUT) its holdings and capabilities to a discovery service. An organization wanting to use that information also is responsible for two things: be able to **find** the information by being able to query the discovery service (step 2: GET), and be able to **bind** to the information source by developing software to work with the data access service's access interface (step 3: GET).

---

[5] "Web Services Architecture," W3C Working Draft 14 May 2003, http://www.w3.org/TR/2003/WD-ws-arch-20030514/#id2608426

[6] "Uniform Resource Identifier (URI): Generic Syntax," IETF Standard: 66, http://tools.ietf.org/html/rfc3986.

[7] "Hypertext Transfer Protocol -- HTTP/1.1," IETF Standards track, Request for Comments: 2616, http://www.ietf.org/rfc/rfc2616.txt

[8] "Extensible Markup Language (XML)," World Wide Web Consortium (W3C) http://www.w3.org/XML/

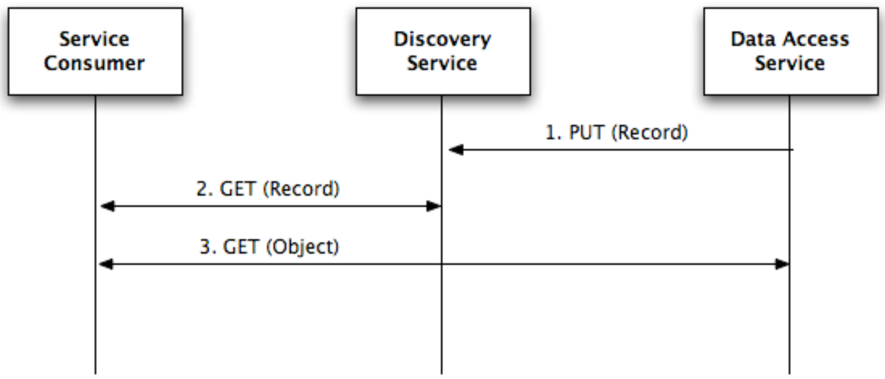

Figure 1: Basic Service Interaction Pattern

4.7.3.3 Workflows

The previous section demonstrated several basic interactions between service consumers and providers. As interactions add more steps, they can be scripted – thereby automating common workflows for reuse by many consumers.

BPEL (Business Process Execution Language for Web Services), XPDL (XML Process Definition Language), and related services can be used across the Building Life Cycle provide a means to formally specify business processes and interaction protocols.

4.7.3.4 Service Tiers

As services become more complex, it will be valuable to organize the services into tiers. A typical pattern is a three-tier model of service consumers and providers.

1. A top tier provides the human-computer interface and is the only one with which people with deal directly
2. A middle tier embodies business processes required to respond to requests issued by consumers. The services in general embody everything from authentication to complex processing on data from various repositories and from generation of analytical results that the client gets back at the end of the process
3. The lower tier provides read and/or write access to data stored in any of a dozen different types of access services.

This concept of tiers will likely not be developed in this Testbed, but may become useful in future Testbed phases.

4.7.4 Prior OGC Work on AEC SOA

The OGC has applied the Web Service Platform to geospatial information, and OGC Web Services have been previously applied to the spatial topics of AECOO. Initial development of web service based interoperability of IFC information was achieved in the OGC Web Services, Phase 4 (OWS-4) Testbed.[9]

[9] OGC 07-023r2, OGC Web Services Architecture for CAD GIS and BIM, 2007-05-16

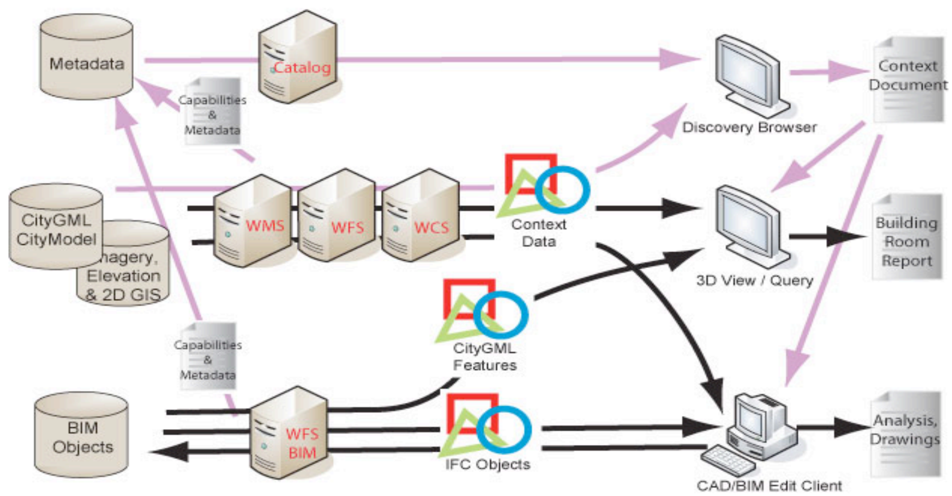

Figure 2: OWS-4 Architecture and Components for IFC

The activities of OWS-4 resulted in the development of several new types of components that demonstrate the integration of BIM with web services architecture. Figure 2 depicts a new type of Transactional Web Feature Service (WFS-T for BIM). This service serves features from BIM in both IFC and CityGML; New client capabilities for three-dimensional thematic viewing and analysis of building information in CityGML; New capabilities in BIM authoring clients that consume CityGML from WFS, and images from Web Map services to allow the development of BIM in geographic context. The OWS-4 architecture enabled an interactive scenario for quickly responding to an emergency situation by establishing a temporary hospital in an existing airport hangar.

While OWS-4 focused on the spatial aspects of IFC, the basic paradigm of web services for IFC exchange was established and can serve as a basis for the AECOO Testbed. The OWS-4 experiments identified several items that are needed in AECOO Web Services architecture. The OWS-4 architecture identifies services for discovery, access and visualization in a tiered architecture.

4.7.5 Resource-Oriented Architecture (ROA)

A resource-oriented approach to SOA differs from the traditional one described above in that the focus is on data sets, or information resources, rather than the computational operations, or services, which provide access to the data. Both design patterns can provide information through SOA — if SOA is a sentence, operations are the verbs, and resources are the nouns. In traditional Web services SOA, software architects concern themselves greatly with designing the operations, while in ROA, the operations, or verbs, are given – HTTP's GET, PUT, POST, and DELETE – and the designers concern themselves solely with the nouns, or representing the underlying data sets as web-accessible resources.

Here are some useful definitions in the ROA space:

- **Resource:** anything important enough to be referenced as a thing.
- **Resource identifier:** a URL (or URI or URN) to uniquely identify the resource.
- **Resource name:** identifies the resource. It has meaning to a human user and indicates the intended purpose of the resource. It has to be distinguished from the identifier of the

representations of the resources that also provide an address (a path) to access the resource.

- **Resource representation:** a resource may have any number of representations. A representation is a concrete instantiation of a resource.

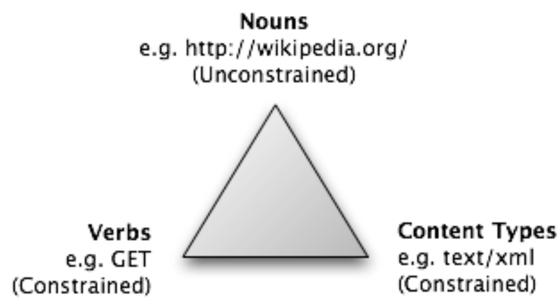

Figure 3: REST Triangle of nouns, verbs, and content types

4.7.6 IFC Access with Files, SOA and ROA

To illustrate the differences between exchanging files over a network, traditional SOA, and ROA, let's look at an example of sharing an IFC model.

Scenario: There is a building information model (BIM), with a subset that represents the building's enclosure (BIM-E) at a particular level of detail (BIM-E-LOD1). These data must be provided in their native format, let's call it ABC, as well as DXF and ifcXML.

A *file exchange* system would put these 9 identified items (3 BIMs in 3 formats) on an FTP server or a "portal" file sharing system, with no standards for access, naming, update, or other crucial communications and metadata requirements.

A *Web Services SOA* system would provide web-based access to these data, and might define data access service requests that look something like these:

BIM metadata:

http://ex.com?request=GetCapabilities&projectid=3324
information on ownership, provenance, file formats, updates, etc.)

BIM access:

http://ex.com?request=GetModel&projectid=3324
(returns data in ABC format)

http://ex.com?request=GetModel&projectid=3324&format=DXF
(returns data in DXF format)

http://ex.com?request=GetModel&projectid=3324&format=XML
(returns data in ifcXML format)

BIM-E access:

http://ex.com?request=GetModel&projectid=3324&section=enclosure
(returns data in ABC format)

http://ex.com?request=GetModel&projectid=3324&section=enclosure&format=DXF
(returns data in DXF format)

http://ex.com?request=GetModel&projectid=3324&section=enclosure&format=XML
(returns data in ifcXML format)

BIM-E-LOD1 access:

http://ex.com?request=GetModel&projectid=3324&section=enclosure&LOD=1
(returns data in ABC format)

http://ex.com?request=GetModel&projectid=3324&section=enclosure&LOD=1&format=DXF
(returns data in DXF format)

http://ex.com?request=GetModel&projectid=3324&section=enclosure&LOD=1&format=XML
(returns data in ifcXML format)

The system would also specify some catalog, or registry that users could query to figure out (find) what data was available in the system (bind).

An *ROA approach* would consider the BIM as a resource, and major subsets of the BIM, such as the enclosure) would also be resources. The different formats would be representations of the resource.

An ROA design might provide the following URLs to access these resource representations:

BIM metadata:

http://www.example.com/project/3324

BIM access:

http://www.example.com/project/3324/bim
(returns data in default format or a list of available formats)

http://www.example.com/project/3324/bim.dxf
(returns data in DXF format)

http://www.example.com/project/3324/bim.xml
(returns data in ifcXML format)

BIM-E access:

http://www.example.com/project/3324/bim/enclosure
(returns data in default format or a list of available formats)

http://www.example.com/project/3324/bim/enclosure.dxf
(returns data in DXF format)

http://www.example.com/project/3324/bim/enclosure.xml
(returns data in ifcXML format)

BIM-E-LOD1 access:

http://www.example.com/project/3324/bim/enclosure/lod/1
(returns data in default format or a list of available formats)

http://www.example.com/project/3324/bim/enclosure/lod/1.dxf
(returns data in DXF format)

http://www.example.com/project/3324/bim/enclosure/lod/1.xml
(returns data in ifcXML format)

4.7.7 Computational Viewpoint Summary

This section presented a variety of approaches to exchanging information between collaborating organizations. We started with a discussion of the current state of the industry, and its deep roots in file-based information exchange. We then laid out the landscape of alternatives, starting with file exchange and moving into web services, with a generic definition of the concepts and terms. We finished with a short description of a particular type of web service architecture, ROA, which is well suited to adoption by the AECOO industry, due to its close relationship to file-based access over the web.

In this Testbed, we plan to use ROA as the guiding principle behind our service designs. This approach will be sketched out roughly in the following thread-specific sections, and further developed in the Testbed kickoff.

5 Quantity Takeoff for Cost Estimation (QTO) Thread

This thread of the Testbed will advance interoperability by defining a Model View Definition that properly captures the information requirements for performing a quantity takeoff especially on the building enclosure for use with an energy analysis tool. The same concepts can then expand to the rest of the facility. Participants working with architectural software will export their designs to this MVD, and participants working with quantity takeoff software will import from this MVD, and write a set of to-be-determined results back into this MVD along with the appropriate metadata and visualization information to identify what work has been completed. All information exchanges will be lossless, and will not result in degradation of coordinate information.

5.1 Deliverables

The following Engineering Reports, services, tools and data instances will be developed in the QTO thread, tested in Technology Integration Experiments, and invoked for cross-thread scenarios in demonstration events.

In all cases, proposers should indicate their preference for using IFC STEP physical file, ifcXML or a hybrid for defining model views and/or implementing software components.

Deliverable	Type
IDM for QTO of Building Enclosures (process maps, exchange requirements, functional parts, etc.)	Engineering Report
Model View Definition for QTO	Engineering Report
QTO Model View service	Software Component
QTO Model View client	Software Component
Model View Catalog	Software Component

5.2 Enterprise Viewpoint

5.2.1 Problem Statement

Building information models need to capture systematic constructs and information about the building so that sufficient quantities and supporting model information may be passed to estimators without having to re-enter information or make laborious preparation efforts, employing ad hoc methods that may cause errors to cascade, or otherwise result in inefficiencies and inconsistencies.

There are as yet no defined standardized process models that any architecture or construction organization can use to define workflows and rules for capturing information coming from architectural building models and use in construction building modes that are then used to provide cost estimates. Put another way, IFC objects and IDM derived model views need to account for differences between designers' BIMs and construction companies' BIMs. Specifications are usually template documents that are currently standalone and static.

Enabling accurate and reliable transfers of quantities necessary for cost estimating is thought to be a first step in making this process work efficiently. This requires a well established definition of the quantity definitions that are well understood by the designer, the contractor, and the BIM application, the estimation package and the cost recipes. An estimator should receive model information from a BIM, be able to link it to its cost recipes, and then provide a return of appropriate cost information along with linked attribute data for IFC objects.

5.2.2 QTO Use Cases

Table 5: Quantity Takeoff Use Case

Name

QTO-1

Objective

Exchange building enclosure design information between architects and cost estimators. Support consistent, predictable quantity takeoffs with QTO Model View, which then can enable cost and constructability analysis, and ultimately inform detailed design.

Description

This Use Case addresses the ability for architects to provide cost engineers with design information appropriate for quantity takeoff generation. While the primary roles in this Use Case are architects related roles include the contractor and engineering disciplines. The scenario for this use case is a quantity takeoff from a model that is reasonably well populated at a full design development phase. The concept will be to be able to involve all the disciplines to include designer, engineers, specifier, contractor, sub-contractor, fabricator, and supplier. However, the focus is on having information that can be used to support the energy analysis as part of the Testbed. This would therefore include materials and functional aspects of the exterior closure as well as information bout the mechanical and control systems. The ideal situation would be that as the energy analysis suggested alternatives, the cost would be reflective of those changes.

Actors

Key Actor: Cost Engineer

Data Suppliers: Owner/Occupant/Developer, Architect, Engineer Specifier and Material Supplier(s).

Data Consumers: Fabricator, Architect, Engineer, and Owner/Developer.

Data Requirements

Project Data (identifying information; revision documentation).

Building Description: Building Element Geometry and Parameter, Building Element Properties and Quantities, Spatial Elements with Geometry, Parameter, Quantities and Properties.

Workflow Process and Information Exchanges

Step	*Activity*	*Primary Actor*	*Actor that is Source of Data*	*Actor that is Destination of Data*
1	Architect CREATES IFC or ifcXML conformant to an "enclosure" Model View (to be developed in the thread) out of their proprietary design software.	Architect	Architect	Cost Engineer and Mechanical Engineer
2	Architect PUBLISHES Enclosure Model View.	Architect	Architect	Cost Engineer and Mechanical Engineer
3	Architect UPDATES Model View catalog to advertise the publication of above.	Architect	Architect	Cost Engineer and Mechanical Engineer

4	Fabricator QUERIES catalog for location (resource endpoint) of the Enclosure Model View.	Fabricator	Architect	Cost Engineer and Mechanical Engineer
5	Cost Engineer and Mechanical Engineer RETRIEVES the Enclosure Model View.	Fabricator	Architect	Cost Engineer and Mechanical Engineer
6	Cost Engineer and Mechanical Engineer EXECUTES quantity takeoffs. (quality component)	Fabricator	Fabricator	Cost Engineer and Mechanical Engineer
7	Cost Engineer and Mechanical Engineer UPDATES Enclosure Model View with the above calculations.	Fabricator	Fabricator	Architect
8	Cost Engineer and Mechanical Engineer repeats all or some of the previous steps based on gaining additional knowledge as the building design becomes more specific.	Fabricator	Fabricator	Architect, Building Owner

5.3 Information Viewpoint

The key information components in this thread are:

- Geometric representation of the building enclosure
- Building construction elements such as walls, windows, doors, floors, roof, mechanical, control systems, etc.
- Space plans

These information components must be captured in an information model that is shared between the design and quantity takeoff teams, and those components must be incorporated into an MVD.

The following resources may be adopted, or used as input to this work.

- IFC2x3 QTO Functional Parts
- IFC2x3 QTO Requirements Definition - Part 1 Base Quantities
- Model View Definitions at http://www.blis-project.org/IAI-MVD/
  - Architectural design to quantity take-off - level 1 (VBL-004)
  - Architectural design to quantity take-off - level 2 (GSC-002)
  - Architectural design to quantity take-off - level 3 (VBL-006)
  - Architectural design to Structural design (VBL-002)

5.4 Computational Viewpoint

The computational viewpoint for this thread is straightforward. It is developed out of the use case and the MVDs discussed above in the Information viewpoint. There will be three interoperability components required to perform the work in this thread:

- *BIM Access Service:* provides BIM data conforming to specific MVDs developed for quantity takeoffs. This service should also be able to write (**publish**) updates regarding its holdings to the Discovery Service.
- *Discovery Service:* a basic catalog of information available through the BIM Access Service.

- *Engineering Service Client:* software component that can read information from the Discovery service (**find**), use that to read information from the BIM Access Service (**bind**), and put this information into native software. Finally, this or a related component, shall also write cost estimation information back to the MVD (**update**).

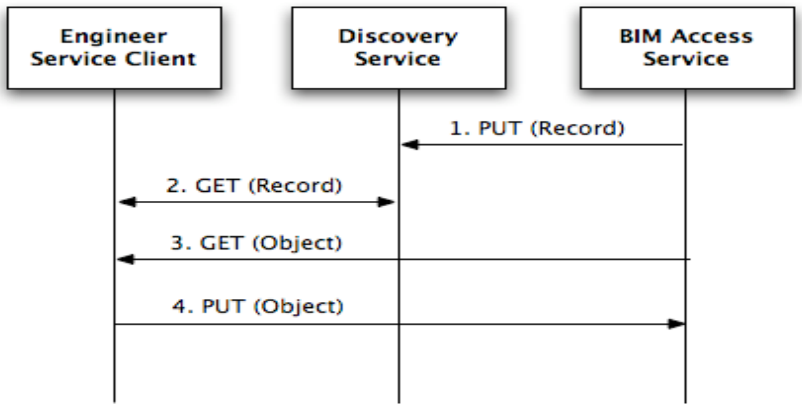

Figure 4: Quantity Takeoff Interaction Diagram

Step 1 – PUT (Record): Architect adds a record to the Discovery Service, advertising the existence of the required Model View.

Step 2 – GET (Record): Cost Engineer and Mechanical Engineer receives notification that the work product is ready (by phone or email), and queries the Discovery Service for details about accessing the Model View.

Step 3 – GET (Object): Cost Engineer and Mechanical Engineer review objects of the BIM Model View by performing a GET operation on the BIM Access Service. The Fabricator may be using an application that is a client of the Access Service, and provides data inspection and visualization tools. The Analyst concludes that the BIM objects are suitable for quantity takeoffs and cost estimation.

Step 4 – PUT (Object): Cost Engineer and Mechanical Engineer append information to the Model View.

The role of the cost estimator is to facilitate the design process by systematic application of cost criteria so as to maintain a sensible and economic relationship between cost, quantity, utility and appearance which thus helps in achieving the client's requirements within an agreed budget. It commences when little is known about the project other than its overall size, probable location and type (or design intent). Ideally, building modelers and cost estimators and specifiers need to continually work together and share relevant information throughout their processes and individual activities.

Cost estimators and designers undertake cost modeling progressively throughout the design and construction of a project, and make use of the best information that is available at the time. Since this may be unbalanced, in that some areas may be more accurate than others, metadata is required as to the level of confidence there is in each number. It starts at the earliest stage as historically based when information may be available only about the type of building required together with its expected overall size and location. As more detail is added to the design, cost modeling can be refined based on area measurement of spaces typically using a Uniformat based approach, until estimates can be developed based on a more complete knowledge of the elements to be incorporated within the project in a MasterFormat based estimate.

Unfortunately, there has been no simple correlation between building elements modeled in early design stages, and building elements modeled much closer to construction. However, with the new version of Uniformat a cross over to MasterFormat will soon be available. This non-hierarchical level of detail issue complicates the process of quantity takeoffs at different stages of design. This necessitates the need for metadata so that an estimating continuum can be established and not perpetuate the cost information disconnects that currently plague the process

The sponsors recognize there is considerable work to do at the conceptual level to resolve information exchange issues in the design-quantify takeoff-cost estimation continuum. The goal of this thread of the Testbed is to enable better cost estimates by achieving better interoperability in geometry and design level of detail for quantity takeoffs.

6 Building Performance and Energy Analysis (BPEA) Thread

This thread of the Testbed will advance interoperability by defining IDMs, Exchange Requirements, and Model View Definitions that properly capture the information requirements parts of the building life cycle where energy analysis plays important roles. Emphasis is place on enabling interoperable information flows for the analysis tools *EnergyPlus* and/or *DOE2*. Participants working with architectural software will export their designs to these MVDs, and participants working with energy analysis software will import from these MVDs, and write analysis results back into these MVDs. All information exchanges will be lossless, and will not result in degradation of coordinate information.

6.1 Deliverables

The following Engineering Reports, services, tools and data instances will be developed in the BPEA thread, tested in Technology Integration Experiments, and invoked for cross-thread scenarios in demonstration events.

In all cases, proposers should indicate their preference for using IFC STEP physical file, ifcXML or a hybrid for defining model views and/or implementing software components.

Deliverable	Type
IDMs for peak loading analysis (process maps, exchange requirements, functional parts, etc.)	Engineering Report
IDMs for annual energy performance analysis (process maps, exchange requirements, functional parts, etc.)	Engineering Report
Model View Definitions for peak loading analysis	Engineering Report
Model View Definitions for annual energy performance analysis	Engineering Report
BPEA Model View service(s)	Software Component
BPEA Model View client(s)	Software Component

6.2 Enterprise Viewpoint

6.2.1 Problem Statement

Whole building energy analysis is an integral part of designing sustainable/green buildings. Energy analysis is performed to help design the building, define building systems and for code checking.

Decisions regarding the reduction of energy consumption and carbon footprint, to take advantage of passive energy and lighting sources, and to incorporate renewable energy sources, should begin to be addressed during schematic design. The potential for the BIM to support this analysis is significant.

The abstract notion of a BIM somewhat obscures the issue of how energy analysis programs manage data by suggesting that the analysis is performed on the same object model that is being used for other purposes. In fact, the analysis programs typically have their own representation of the building geometry, which is mapped to the BIM. This implies a workflow that includes extracting geometry from the BIM, transforming it in some way, analyzing the new representation, adding data to it, and then returning data to the BIM. (This is not to say that it is not possible to perform energy analysis in the context of the BIM, but this implies a very proprietary solution and this is not under consideration.)

There are two general kinds of barriers to this workflow: first is that there are characteristics of the geometry and the spatial data stored in the BIM that make it unsuitable for energy analysis; and second is that there is no way to seamlessly exchange data between the energy analysis software and the BIM.

For this Testbed we are focusing on energy simulation in design and the definition of building systems, specifically the information exchanges that must happen between BIM software and energy analysis software for building design teams to clarify design intent and the loads imposed on HVAC building systems.

There are no defined and interoperable processing services to facilitate information exchange for performance systems, that connect building information models and energy analysis tools, or that facilitate information exchanges between the disparate analysis tools. Some software applications do offer a one-way export of IFC data related to HVAC, which can be used by analysis applications. Having such kinds of services would well serve the interests of architects, engineers, construction companies and many of their tradecraft partners and owners. The output from such analysis applications would inform other building specification activities about the performance of energy related building components and cost, and ultimately enable more seamless mechanisms for code checks.

When engineers calculate peak load conditions, they frequently do it with software other than EnergyPlus and DOE2 for the actual sizing of equipment. They use load calculation software such as Trane Trace and Carrier HAP. Neither of these use EnergyPlus or DOE2. EnergyPlus and DOE2 execute peak design day calculations, but that is only used to determine relative impact of designs, typically earlier in the design process, not to actually size the mechanical equipment, ducts, piping, etc. When we say the emphasis is placed on EnergyPlus and DOE2 in this Testbed, we are saying the simulation engine/software specific efforts of this Testbed are addressed in use case 2 only, described below. Use case 1, for detailed peak load analysis, doesn't reference any specific software or engine, therefore it is a bit more broad. Testbed work for Use Case 1is about how best to go about preparing BIM data for use in load analysis software, and we are interested in novel approaches to this problem. The sizes of the HVAC systems, piping and duct layouts, etc. (the building systems that will have elements within a BIM model), are based off the load calculation results.

6.2.2 BPEA Use Cases

Use cases drive software developers' ability to meet a project's business requirements. Therefore, detailed use cases are presented here in the enterprise viewpoint. This section defines the data exchange requirements and workflow scenarios for a building structure where there are required communications between a design firm and a firm analyzing building performance from an energy perspective.

The use cases explore the relationships of BIM information for informing:

- The impacts of building material alternatives, building envelope configurations, and internal loads on building peak load requirements, and

- The impact of daily and seasonal variations in weather, occupancy, and equipment operation (lighting, computers, HVAC, etc.) on annual energy performance.

Table 6: Peak Loading Use Case

Name

BPEA-Peak Loading 1

Objective

Model building peak heating and cooling load requirements during the design of a new facility. Exchange building enclosure and other relevant design information between architects and mechanical engineers. Link thermal loading calculations to design at all stages of the design process.

Description

Peak design values must be established to determine the maximum heat addition or extraction required to maintain specified conditions in all spaces in a building during annual temporal extremes. These values are termed "peak loads." Peak loads are calculated using thermal modeling procedures driven by suitably chosen assumptions for weather and operation (the "design conditions"). The Mechanical Engineer is the central actor in the load calculation process and assembles from other actors design requirements, building information (location, geometry, and construction specifics), weather data, operation data (occupancy schedules, internal lighting and equipment power profiles), and regulatory requirements (e.g. mandated fresh air rates). These values are inputs to loads calculation software. Frequently many of these items are uncertain, especially early in the design process. The Mechanical Engineer must supply assumed values based on judgment.

The loads calculation process is iterative and may be refined several times as the design evolves. The final load calculation results ultimately become a part of the building life cycle database for future facility renovation and operation needs (e.g. equipment replacement).

Actors

Key Actor: Mechanical Engineer

Data Suppliers: Owner/Occupant/Developer, Architect, Mechanical Engineer, Lighting Designer, Electrical Engineer, Equipment Supplier(s), Material Supplier(s), Regulatory Requirements Source(s), Weather Data Source(s), Engineering Data Source(s), Commissioning Authority, Facility Operations and Maintenance Personnel.

Data Consumers: Mechanical Engineer, Architect, Electrical Engineer, Commissioning Authority, Regulatory Authorities, Owner/Developer, Facility Operations and Maintenance Personnel.

Data Requirements

Site Data: Location, terrain, surrounding environmental and land use characteristics that may effect exposure and lighting and shading

Project Data: Identifying information, revision documentation.

Design Requirements: Required thermal and comfort conditions at peak design condition.

Regulatory Requirements: Relevant Energy/ mechanical codes (prescriptive and performance based)

Site Data: Location (coordinates, elevation), surrounding environment (exposure, shading).

Weather Data: Heating and cooling design conditions.

Operation Data: e.g., Occupancy type, floor area breakdown, occupancy profiles for lighting, plug loads, by space during peak conditions

Building Description: Thermal view geometry, construction elements (e.g., walls, windows, floors, etc), construction material thermal characteristics, HVAC systems type, etc

Workflow Process and Information Exchanges

Step	*Activity*	*Primary Actor*	*Actor that is Source of Data*	*Actor that is Destination of Data*
1	Architect CREATES IFC conformant to an "enclosure" Model View (to be developed in the thread) out of their	Architect	Architect	Architect and Mechanical

	proprietary design software.			Engineer
2	Architect PUBLISHES Enclosure Model View.	Architect	Architect	Architect and Mechanical Engineer
3	Architect UPDATES Model View catalog to advertise the publication of above.	Architect	Architect	Architect and Mechanical Engineer
4	Mechanical Engineer QUERIES catalog for location (resource endpoint) of the Enclosure Model View.	Mechanical Engineer	Architect	Mechanical Engineer
5	Mechanical Engineer RETRIEVES the Enclosure Model View.	Mechanical Engineer	Architect	Mechanical Engineer
6	Mechanical Engineer EXECUTES thermal loading calculations.	Mechanical Engineer	Mechanical Engineer	Mechanical Engineer
7	Mechanical Engineer UPDATES Enclosure Model View with the above calculations.	Mechanical Engineer	Mechanical Engineer	Architect
8	Mechanical Engineer repeats all or some of the previous steps based on gaining additional knowledge as the building design becomes more specific.	Mechanical Engineer	Mechanical Engineer	Architect, Building Owner

Table 7: Annual Energy Performance Use Case

Name

BPEA- Energy Performance 1

Objective

This Use Case describes building annual energy performance simulation requirements during design of a new facility. Simulation results are to be archived and updated as necessary to support decision-making during design and construction, and also during operations and maintenance (O&M) of the occupied facility over time.

Description

At various stages of the design cycle, the project team may want to predict the energy performance of design iterations for a variety of reasons, including, code compliance, rating system certification (e.g., LEED), and energy-efficiency design analysis. Though frequently simulation services are obtained from a 3rd party energy/LEED/simulation consultant, the majority of the time annual energy performance is evaluated by the Mechanical Engineer on record. The Mechanical Engineer receives building design data from multiple design team players. The design data includes information about the current physical configurations and use patterns to consider. The Mechanical Engineer produces a series of alternative design configurations, all referenced against a common baseline design, and reports results back for the design team's consideration.

The final design energy model produced by the Mechanical Engineer ultimately becomes a part of the building life cycle database for future facility renovation and operation needs.

Actors

Key Actor: Mechanical Engineer

Data Suppliers: Architect, Mechanical Engineer, Electrical Engineer, Simulation Consultant, Equipment Manufacturer, Material Supplier, Codes and Standards, Various Databases (e.g., Energy Performance Benchmark databases), Commissioning Authority, Facility Operations and Maintenance Personnel.

Data Consumers: Architect, Mechanical Engineer, Electrical Engineer, Simulation/LEED Consultant, Commissioning Authority, Owner/Occupant/Developer, Facility Operations and Maintenance Personnel, HVAC equipment and systems specifiers and suppliers, lighting equipment and systems specifiers and suppliers.

Data Requirements

Site Data: Location, terrain, surrounding environmental and land use characteristics that may effect exposure and lighting and shading

Operational Requirements: Required thermal and comfort conditions throughout the year, part load performance of HVAC systems, control strategies, etc.

Physical Building Data: Thermal view geometry, construction elements (e.g., walls, windows, floors, etc.), construction material thermal characteristics, etc.

Operational Data: Occupancy type, floor area breakdown, internal peak loads (e.g., people, lighting, equipment/plug, process), internal load schedules/profiles.

Physical building data: Thermal view geometry, construction elements (e.g., walls, windows, floors, etc,), construction materials thermal characteristics, HVAC systems type, etc.

Regulatory Requirements: Relevant Energy/ mechanical codes (prescriptive and performance based)

Design Strategies: e.g., Daylighting, solar PV, natural ventilation, etc.

Simulation Purpose and Supporting Data: e.g., rating system certification data, and energy performance benchmarks.

Workflow Process and Information Exchanges

Step	*Activity*	*Primary Actor*	*Actor that is Source of Data*	*Actor that is Destination of Data*
1	Architect CREATES IFC conformant to an "enclosure" Model View (to be developed in the thread) out of their proprietary design software.	Architect	Architect	Architect and Mechanical Engineer
2	Architect PUBLISHES Enclosure Model View.	Architect	Architect	Architect and Mechanical Engineer
3	Architect UPDATES Model View catalog to advertise the publication of above.	Architect	Architect	Architect and Mechanical Engineer
4	Mechanical Engineer QUERIES catalog for location (resource endpoint) of the Enclosure Model View.	Mechanical Engineer	Architect	Mechanical Engineer
5	Mechanical Engineer RETRIEVES the Enclosure Model View.	Mechanical Engineer	Architect	Mechanical Engineer
6	Mechanical Engineer EXECUTES thermal loading calculations.	Mechanical Engineer	Mechanical Engineer	Mechanical Engineer
7	Mechanical Engineer UPDATES Enclosure Model View with the above calculations.	Mechanical Engineer	Mechanical Engineer	Architect
8	Mechanical Engineer repeats all or some of the previous steps based on gaining additional knowledge as the building design becomes more specific.	Mechanical Engineer	Mechanical Engineer	Architect, Building Owner

6.3 Information Viewpoint

The key information components in this thread are:

- Geometric representation of the building enclosure
- Building construction elements such as walls, windows, doors, etc.
- Other inputs to energy analysis such as weather, utility rates, building usage, etc.

Many of these information components must be captured in an information model that is shared between the design and analysis teams, and those components must be incorporated into one or more MVDs.

A significant input to the design of these MVDs is Green Building XML (gbXML). The gbXML schema describes building information needs between standalone CAD and energy modeling programs. The relationship between IFC/ifcXML and gbXML to produce MVDs will need to be further clarified during the work on this thread.

It will also be necessary to develop a shared handling of Cartesian geometry and earth geometry to support the types of common analyses desired. OGC's OWS-4 Testbed prototype implementations did achieve the exchange and proper registration of geo-referenced BIM, OGC did not implement the geo-referencing model of the proposed draft standard IFC2x3g. Yet we understand that this geo-referencing model instantiates the full geo-referencing specification of the European Petroleum Survey Group (EPSG) and we presume, ISO19111. We believe that this indicates a fairly complete specification of the sorts of coordinate referencing systems expected in the world of AEC and portends a high level of interoperability with other geospatial service architectures.

IFC2x3g along with CityGML (see section 4.6) should be considered as a means to achieve this goal.

The following additional resources may be adopted, or used as input to this work.

- Architectural Design to Thermal Simulation Model View at http://www.blis-project.org/IAI-MVD/MVDs/VBL-007/Generic/index.html
- IDM Process Maps for HVAC, Piping and Electrical Engineering

6.4 Computational Viewpoint

The objective of the scenario is to perform an energy analysis of the BIM using "black box[10]" analytic tools. It is assumed that the BIM and the analysis tools are available as web-accessible resources that are known to the Energy Analyst. The Energy Analyst is the service consumer in this use case. The Analyst may have discovered the services either through previously searching a Discovery Service or information about the services may have been provided through some other communication, e.g., e-mail.

[10] In computing in general, a black box program is one where the user cannot see its inner workings (perhaps because it is a closed source program) or one which has no side effects and the function of which need not be examined, a routine suitable for re-use. Also in computing, a Black Box refers to a piece of equipment provided by a vendor, for the purpose of using that vendor's product. It is often the case that the vendor maintains and supports this equipment, and the company receiving the Black Box typically are hands-off.

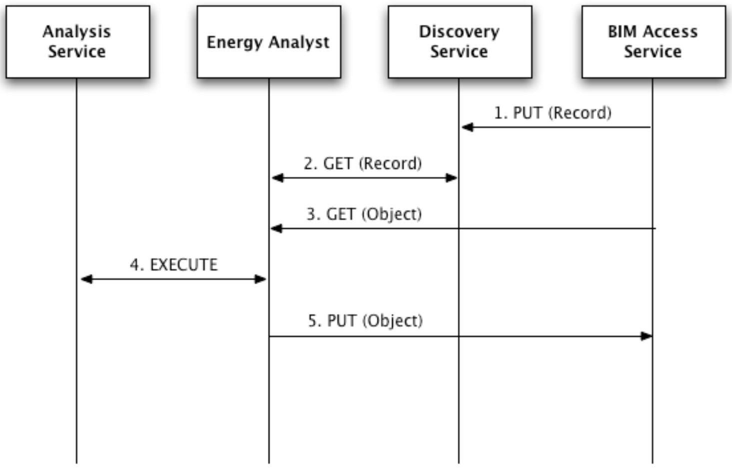

Figure 5: Energy Analysis Interaction Diagram

Step 1 – PUT (Record): Architect adds a record to the Discovery Service, advertising the existence of the required Model View.

Step 2 – GET (Record): Analyst receives notification that the work product is ready (by phone or email), and queries the Discovery Service for details about accessing the Model View.

Step 3 – GET (Object): Analyst reviews objects of the BIM Model View by performing a GET operation on the BIM Access Service. The Analyst may be using an application that is a client of the Access Service, and provides data inspection and visualization tools. The Analyst concludes that the BIM objects are suitable for analysis by the Analysis Service.

Step 4 – EXECUTE: The Analyst's application requests an Execute operation on the Analysis Service.

Step 5 – PUT (Object): Analyst appends information to the Model View.

7 Communications, Project Delivery and Decision Support (CPD) Thread

7.1 Deliverables

The following Engineering Reports, services, tools and data instances will be developed in the CPD thread, tested in Technology Integration Experiments, and invoked for cross-thread scenarios in demonstration events.

In all cases, proposers should indicate their preference for using IFC STEP physical file, ifcXML or a hybrid for defining model views and/or implementing software components.

Deliverable	Type
Harmonizing agcXML with IFC and MVD	Engineering Report
IDM for RFIs	Engineering Report
CPD information server	Software Component
CPD information viewer	Software Component

7.2 Enterprise Viewpoint

7.2.1 Problem Statement

The preceding work threads on quantity takeoffs and energy analysis address important tasks that performed in the development of a building. The intent of this thread is to capture and share the ***meta-information*** about these tasks. In other words, the CPD thread addresses the issue of bringing transparency to the management of all the work that occurs in the design and construction lifecycle.

Here we define open interfaces for sharing the progress of a project across disparate organizations and software packages, and tying that project information to the MVDs developed in the other threads.

7.2.2 Use Cases

This use case describes a transaction between two parties that is commonly known as a Request for Information. The purpose of the use case is to provide a framework for defining the project-related information that is commonly exchanged in a simple RFI transaction. The complete business process for a typical RFI often includes many information exchanges or transactions between multiple parties. Some of these exchanges or transactions may occur simultaneously. This use case is not intended to be exhaustive or encompass all possible use cases.

These use cases are adopted largely from "agcXML Use Case" (Thomas Froese), "agcXML Example Use Case: RFI" (Thomas Froese), and "agcXML RFI Data Requirements and Sample Schemas" (Yimun Zhu). However, it is applied in a slightly modified context here, as the intent is to capture the transactions and business processes occurring in the other threads as much as it is to capture traditional RFIs.

Table 8: RFI Exchange Use Case

Name
CFD – RFI Exchange 1
Objective
Automate the RFI process in a software-agnostic manner so that all parties in a project may use different software systems and still increase efficiency and reduce errors.
Description
An RFI is typically a request from a prime contractor to the owner's prime design consultant (typically, but not always, an architect) for additional or clarifying design information. RFIs are typically regarded as formal project communications and a part of the project record, and typically have contractual implications related to the timeliness of a satisfactory response.
Actors
Key Actor: Architect
Data Suppliers: Architect, Contractors
Data Consumers: Architect, Contractors, Owner, Operator
Data Requirements
Design clarifications revisions, additions
Workflow Process and Information Exchanges

Step	*Activity*	*Primary Actor*	*Actor that is Source of Data*	*Actor that is Destination of Data*
1	Contractor prepares an RFI Request that identifies an issue requiring	Contractor	Contractor	Architect

	additional information from the Architect.			
2	Contractor links RFI narrative to other project information such as model views, spreadsheets, etc.	Contractor	Contractor	Contractor
3	Contractor submits the RFI to Architect	Architect	Contractor	Architect
4	Architect prepares a response and distributes it.	Architect	Architect	Contractor
5	Contractor receives response and verifies it is satisfactory, or re-submits the RFI for additional clarification	Contractor	Contractor	Architect

Table 9: Unit of Work Information Capture Use Case

Name
CFD – Unit of Work Information Capture 1
Objective
Track all significant work products that occur in the building lifecycle.
Description
Actors
Key Actor: Architect
Data Suppliers: Architect, Contractors
Data Consumers: Architect, Contractors, Owner, Operator
Data Requirements
Work process of other threads
Workflow Process and Information Exchanges
• The impacts of building material alternatives, building envelope configurations, and internal loads on building peak load requirements, and • The impact of daily and seasonal variations in weather, occupancy, and equipment operation (lighting, computers, HVAC, etc.) on annual energy performance. Table 6: Peak Loading Use Case**. The objective of this use case is to track all other use cases**
Not applicable. The objective of this use case is to track all other use cases and develop the ability to capture a record of the activities that occur (instead of developing the activities themselves).

7.3 Information Viewpoint

The key information components in this thread are:

- RFIs
- MVDs developed in other threads

In this thread, it may not make sense for the resulting information model to be one or more MVDs. Instead, it may be more fruitful to use agcXML and the work being performed in that group. The following documents are available at http://www.buildingsmartalliance.org/agcxml/doc_review.php

- agcXML Technical Framework (v0.6) 2007-08-20

- agcXML Generic Document Distribution Use Case
- agcXML RFI Comparative Analysis–Rev 1
- agcXML RFI Data Requirements and Sample Schemas 2007-08-16
- agcXML Example Use Case: RFI 2007-06-21

In either case, a goal of this thread will be to understand the relationship between IDM, MVD and agcXML and suggest a strategy going forward for the use of these information standards.

7.4 Computational Viewpoint

The following is a simplified view of the computational requirements for RFI exchange. We hope to develop a more realistic, sophisticated scenario based on the work of the other threads, but those details are not known at this point.

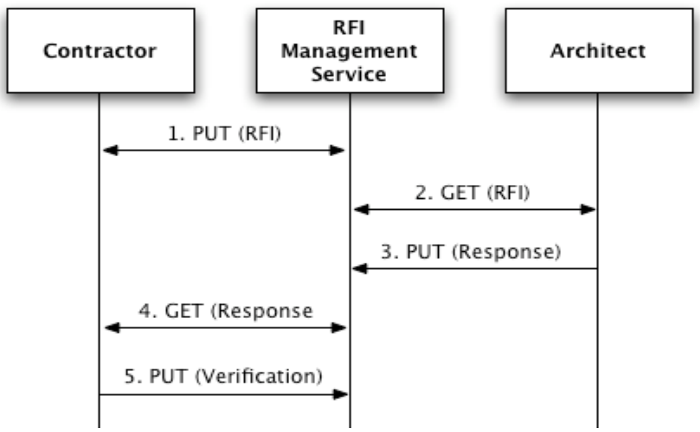

Figure 6: RFI Interaction Diagram

Step 1 – PUT (RFI): Contractor submits an RFI to the RFI Management Service.

Step 2 – GET (RFI): Architect queries RFI Management Service for new RFIs.

Step 3 – PUT (Response): Architect responds to the RFI.

Step 4 – GET (Response): Contractor gets Architect’s response.

Step 5 – PUT (Verification): Contractor verifies the RFI has been resolved.

8 Bibliography

Fielding, Roy, “Architectural Styles and the Design of Network-based Software Architectures”, University of California, Irvine doctoral dissertation, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

Harrison, D., M. Donn and H. Skates, “Applying web services within the AEC industry,” School of Architecture, Victoria University, http://itc.scix.net/data/works/att/w78-2003-145.content.pdf.

Khemlani, Lachmi, “The Web Services Technology and its Implications for AEC,” AECbytes Newsletter #8, (April 29, 2004), http://www.aecbytes.com/newsletter/2004/issue_8.html.

Richardson, Leonard, Sam Ruby, *RESTful Web Services*, O’Reilly, Sebastopol, CA, 2007.