Semantic Mapping Between IAI ifcXML and FIATECH AEX Models for Centrifugal Pumps

E.F. Begley
M.E. Palmer
K.A. Reed

Semantic Mapping Between IAI ifcXML and FIATECH AEX Models for Centrifugal Pumps

E.F. Begley
M.E. Palmer
K.A. Reed
Building Environment Division
Building and Fire Research Laboratory

May 2005

TABLE OF CONTENTS

1. INTRODUCTION

This report describes a semantic mapping between two extensible markup language (XML)[1] specifications describing a single type of equipment that is used in heating, ventilation, air conditioning, refrigeration (HVAC/R) and other types of systems: the centrifugal pump. The first XML specification is called ifcXML and represents the information content of the *I*ndustry *F*oundation *C*lasses (IFC) as defined by the International Alliance for Interoperability (IAI)[2]. The second XML specification, AEX, is the product of a Fully Integrated and Automated Technology (FIATECH)[3] project to define a language for *A*utomating *E*quipment Information E*x*change.

The differences between these XML specifications for centrifugal pump information arise from the contrasting viewpoints – design, procurement and maintenance of engineered equipment vs. whole building project – of the FIATECH AEX and IAI IFC efforts. The semantic mapping between these specfcations offers potential benefits to the AEC/FM community as the basis for harmonization of the XML structures of the respective XML schemas as well as the identification and specification of a methodology for harmonizing schemas representing other types of equipment, system information, facility information, and project information. In the near term, the results can be used to establish exchange mechanisms between information systems that utilize the different XML applications in order to test cross-system exchange scenarios.

The rest of this introduction provides overviews of XML, IAI, IFC and ifcXML, FIATECH, AEX, uses cases for the IFC and AEX, and centrifugal pumps that should prove useful for understanding the semantic mapping presented later in this report.

1.1 What is XML?

XML is an acronym for *eXtensible Markup Language* and its roots stretch back to well before the Internet existed. In fact, XML is a descendant of the "mark up" process used by editors prior to the days of automated typesetting. An editor would write specific processing instructions on a document to delineate its organization and presentation style. A typesetter would use the markup to set the type in the manner appropriate for conveying the editor's vision to the reader. The markup process was retained but adapted when automation and typesetting converged. Markup was now included with the machine-readable document but the specific processing instructions were written in the language of the formatting program rather than the language of the typesetter.

While computerized typesetting represented a major advance, it also introduced a new problem: markup instructions differed on different typesetting systems. A file containing the marked up document could not be readily used on another automated typesetting system without revising the markup. The solution to this problem came in step-wise fashion from the development of generic coding to the Standard Generalized Markup Language (SGML)[4, 5]. SGML is a meta-markup language, meaning it is a language for describing languages and is well-suited for allowing one to design one's own markup language, called an SGML application. SGML is maintained by the International Organization for Standardization (ISO)[6].

From June through December 1980, CERN, the European Organization for Nuclear Research, employed an independent software engineer named Tim Berners-Lee[7]. During this time period, Berners-Lee developed a program for his own use called Enquire. The software was designed for information storage and included support for random associations among the information but was never published.

For approximately four years after contracting at CERN, Berners-Lee worked at John Poole's Image Computer Systems Ltd. on real time control firmware, graphics and communications software, and a generic macro language. During 1984, he returned to CERN under fellowship to develop distributed real-time systems for scientific data acquisition and control. In 1989, he proposed a software project based on Enquire that would enable international collaboration using a web of hyperlinked documents. By this time, SGML was widely used, including at CERN. Berners-Lee noticed that a small, select subset of tags from SGML could be used to markup the hyperlinked documents for distribution by server software. By the summer of 1991, that document distribution system became known globally as the World Wide Web and the small, select subset of tags used to create and link those documents was called the HyperText Markup Language (HTML)[8]. HTML is now maintained centrally by the World Wide Web Consortium (W3C)[9].

That HTML is a small set of tags in large part contributed to the explosion of the World Wide Web. Humans are naturally gregarious and HTML provides a simple, easy-to-learn language that allows anyone to establish a web presence. Yet, the very strength of HTML, a small easily learned tag set, can also be viewed as a weakness. HTML is a very specific and limited application of SGML. It does not support flexibility for those in a given community to establish their own language for their own purposes, i.e., HTML is not extensible. Furthermore, HTML is used for describing a document's form rather than its content. This is a key point since communities of web users rapidly reached the point where they wanted to move beyond the *dissemination* of information into the *exchange* of information.

Recognizing the shortcomings of HTML and the needs of the user community for extensibility, the W3C embarked on an effort to bring an abridged implementation of SGML to the Web. The result of the W3C's effort was the 10 February 1998 release of the version 1.0 specification of XML, the E*x*tensible *M*arkup *L*anguage, which is itself a meta-markup language.

XML brings the three principal features of SGML to the Web:

- Extensibility: users can define their own tags and attributes used in their documents;
- Structure: users can define their own document schema, which is the information model of a document describing how the tags and attributes are combined;
- Validation: users can test the conformance of their documents to the structure defined by the schema.

Additionally, XML intentionally does not include some of the complex functionality of SGML. It was designed to be easy to learn, easy to write, easy to interpret, and easy to implement; characteristics perfectly suited for use on the Web.

The above description of XML emphasizes its role in the world of structured documents. It turns out that XML also offers the possibility of creating truly platform- and program-independent data exchange formats. The syntactic rules for constructing an exchange file through the markup of instance data are simple, well documented, and widely supported in commonly available software tools. The resulting text-based file format is easy to parse. In any given exchange scenario, users achieve interoperability by agreeing to restrict their XML documents to some specific set of tags. These tag sets are known as XML applications. For historical reasons, the resulting exchange file is usually referred to as an XML document even though its content does not represent a document in the traditional sense.

There are now many applications of XML in use or being developed by different communities[10] to facilitate the exchange of data and improve the interoperability of their software systems. The ifcXML and AEX efforts are two such applications of interest to the architecture, engineering, and construction industries.

1.2 International Alliance for Interoperability and the Industry Foundation Classes

The widespread adoption of information technology (IT) over the last 50 years has had an extraordinary effect on how businesses operate. During the same period of time, a key problem also has evolved that affects how businesses interact. Initially, this difficulty was nearly imperceptible since computers were scarce and information including its underlying software systems infrastructure was proprietary and not shared. Eventually, the cost of microprocessors decreased while their capabilities increased. In turn, software applications gained in complexity as well as diversity. Interoperability, i.e., readily sharing information among these applications without tedious, error-prone, and redundant data entry, became a serious impediment. As the use of centralized mainframe computers largely gave way to personal computers and an enormously rich, complex, and ever-changing software environment arose, interoperability issues were exacerbated.

The architecture, engineering, construction, and facilities maintenance (AEC/FM) industry is very familiar with interoperability problems since it uses a broad array of software applications for design, planning, estimating, scheduling, engineering, construction, operation, and maintenance during the life-cycle of a building project. The International Alliance for Interoperability was formed to address the data exchange issues among these AEC/FM software applications. The IAI work builds on the lessons learned in two decades of AEC/FM efforts to leverage and extend several generations of CAD/CAM data exchange standards originating in the manufacturing industries, notably the automotive and aerospace sectors.

The IAI began in August 1994 with twelve companies in the United States collaborating on demonstrating AEC/FM software interoperability. The results of their efforts coupled with the central argument for their activities, a neutral standard for software interoperability would provide significant economic benefit, drew interest sufficient for broadening the level of participation by the AEC/FM community in North America and software vendors. This collaborative effort was formalized in October 1995 with the establishment of the Industry

Alliance for Interoperability in North America. By 1996, the collaboration had become international in scope and was renamed the International Alliance for Interoperability. Table 1 lists the current IAI chapters and provides their links on the World Wide Web.

The interoperability solution developed by the IAI is known as the Industry Foundation Classes (IFC). The information models contained within the IFC initially were derived from the Building Construction Core Model and other models developed within the ISO 10303 framework, known informally as the Standard for the Exchange of Product Model Data (STEP)[11],

North American	http://www.iai-na.org/
German Speaking	http://www.iai-ev.de/
Nordic	http://cic.vtt.fi/niai/
French	http://www.iai-france.org/
Singaporean	http://www.corenet.gov.sg/it_standards/iai/index.htm
Korean	http://italab.khu.ac.kr/iai_korea/
Australasian	http://www.interoperability.org.au/
Japanese	http://www.interoperability.gr.jp/
UK	http://cig.bre.co.uk/iai_uk/new/index.jsp
Iberian	http://www.labein.es/iai-iberica/

Table 1: IAI chapters and their URLs

with input from the AEC/FM industries. STEP was the first major data exchange standardization effort to define data semantics for a variety of application domains in terms of formal information models. For a number of technical and business reasons, the IAI members found that STEP was not a good solution for the interoperability problems they faced. However, their experiences led the IAI to retain a number of underlying STEP technologies to use in creating and implementing its own information models.

Since their initial release in January 1997, the IFC have evolved and now contain information models for nine domains within AEC/FM: 1. building controls; 2. plumbing and fire protection; 3. structural elements; 4. structural analysis; 5. heating, ventilation, and air conditioning systems; 6. electrical systems; 7. architecture; 8. construction management; and 9. facilities management.

Until the fifth release of the IFC in October 2000, the information models were provided solely as schemas represented in the EXPRESS[12] data definition language, the ISO language created for documenting STEP. Instance data are encoded in the STEP clear-text format[13] based on the EXPRESS schema and a set of encoding rules. EXPRESS was developed to be a general-purpose information modeling language and the companion encoding rules are complicated. The syntax of the resulting exchange files is complex and difficult to parse without using STEP technology. A number of STEP- and IFC-based software toolkits are available to application developers that greatly ease the implementation of translators and repositories.

1.3 ifcXML

In recognition of the impact of XML in other IT domains, the fifth and sixth releases of the IFC have also included XML schema definition language[14] representations, which are known as ifcXML1 and ifcXML2, respectively. These XML schemas are translated mechanically from the EXPRESS schemas using the evolving XML language-binding rules[15] defined by the ISO/STEP

organization and specialized[16] by the IAI. The goal of this work was to make it possible to exchange IFC data files alternatively as XML documents. Another, equally important goal was to enable the reuse of IFC content and structure within XML-based initiatives for data exchange and sharing in the AEC/FM industries.

These two goals impose different technical requirements on the ifcXML. As one consequence, ifcXML retains some of the information structure imposed by the underlying EXPRESS schema, and the resulting XML documents are more intricate and voluminous than would result from an information model optimized for XML representation. This problem has been known since the creation of ifcXML1 and work is on-going[17] to establish a leaner XML representation that is better suited to the creation of repositories and facilitating partial-model exchanges.

The semantic mapping discussed in this document utilizes ifcXML2, which will be referred to as ifcXML throughout.

1.4 IFC Use Cases

The International Alliance for Interoperability succinctly states[18] its vision, mission, and goal as:

> **Vision**
> Improving communication, productivity, delivery time, cost, and quality throughout the whole building life cycle.
>
> **Mission**
> Providing a universal basis for process improvement and information sharing in the construction and facilities management industries.
>
> **Goal**
> Building on the collective knowledge of the global construction and facilities management industries to define Industry Foundation Classes.

The breadth and depth of the IFC models are tangible reflections of these ideas and clearly indicate the broad, diverse extent of their intended use. The models provide data structures suitable for many applications beyond the expected CAD software exchange of building geometry and address all types of AEC/FM project information including:

- manufactured, supplied, or created products used in the project;
- processes that take place within a project such as the acquisition of products or services, construction, scheduling, and maintenance; controls or constraints applied to objects including guidelines, specifications, or regulations;
- resources or concepts describing the use of an object;
- actors, which are the human agents involved in a project;
- and project activities.

The IFC model architecture, discussed in section "2.1 IFC Design Principles and Pump-Related Modeling," has four layers. Even without examining their full content but, rather, simply

Resource Layer		
IfcActorResource	IfcMaterialPropertyResource	IfcProfileResource
IfcApprovalResource	IfcMaterialResource	IfcPropertyResource
IfcConstraintResource	IfcMeasureResource	IfcQuantityResource
IfcCostResource	IfcPresentationAppearanceResource	IfcRepresentationResource
IfcDateTimeResource	IfcPresentationDefinitionResource	IfcStructuralLoadResource
IfcExternalReferenceResource	IfcPresentationDimensioningResource	IfcTimeSeriesResource
IfcGeometricConstraintResource	IfcPresentationOrganizationResource	IfcTopologyResource
IfcGeometricModelResource	IfcPresentationResource	IfcUtilityResource
IfcGeometryResource	IfcProfilePropertyResource	

Core Layer	Interoperability Layer	Domain Layer
IfcKernel	IfcSharedBldgElements	IfcArchitectureDomain
IfcControlExtension	IfcSharedBldgServiceElements	IfcBuildingControlsDomain
IfcProcessExtension	IfcSharedComponentElements	IfcConstructionMgmtDomain
IfcProductExtension	IfcSharedFacilitiesElements	IfcElectricalDomain
	IfcSharedMgmtElements	IfcFacilitiesMgmtDomain
		IfcHvacDomain
		IfcPlumbingFireProtectionDomain
		IfcStructuralElementsDomain
		IfcStructuralAnalysisDomain

Table 2: High-level view of the layers within the IFC model architecture

focusing on just the names of the sub-models in each layer (Table 2), especially Resource and Domain, provides insight into the many use cases that might be addressed by applying the IFC.

A few examples of the variety of IFC use cases are:

- Import of an IFC-encoded preliminary space plan developed from a client brief regarding the basic rooms and spaces of a building into architectural design software;
- Web-based visualization of an IFC-encoded building model using the Virtual Reality Markup Language (VRML)[19];
- Export of architectural design to structural design and analysis software;
- Export of structural design to prefabricated component design software;
- Information exchange of supply specifications and availabilities among material suppliers, accessory suppliers, and designers;
- Export of architectural design to quantities takeoff cost estimating software;
- Export of architectural design to construction planning and scheduling software;
- Export of architectural design to building code compliance software for verifying fire egress, handicapped access, and other constraints;
- Export of HVAC system design to quantities takeoff cost estimating software;
- Export of architectural design to thermal load calculation software for HVAC system design;
- Export of HVAC design to energy code checking software;

- e-Business procurement wherein customers try to locate, search, and select from available Web-based commodity HVAC/R equipment catalogs deployed by information brokers, manufacturers, and suppliers that contain design information, performance properties, cost, installation instructions, and supplier details;
- Examination of IFC-encoded HVAC/R product catalogues to identify and select functionally equivalent replacements for equipment that has reached the end of its life-cycle;
- Examination of IFC-encoded HVAC/R product catalogues to identify and select new HVAC/R equipment during the facilities design process;
- Export of building design to facilities management and maintenance planning software;
- Export of building design to occupancy planning software;
- Export of planned maintenance activities to scheduling software;
- Export of planned maintenance activities to cost estimating and resource planning software;
- Export of asset information to facilities management software.

While tabulation of use cases is valuable for linking the abstract IFC data models to concrete applications, it is especially important to note that, internationally, assorted implementation activities are being conducted by industries, laboratories, and other organizations. An overview of some of these activities showing the status of IFC translators in a variety of AEC software products is available from the IAI-Implementation Support Group[20].

1.5 FIATECH and the AEX Project

FIATECH is a research, development, and deployment consortium established in January 2000. FIATECH's mission is to develop, deploy, and commercialize technologies that realize fully integrated and automated project processes, which will result in significant cycle time and lifecycle cost reductions and efficiencies in the delivery, operation, and maintenance of capital projects and facilities. FIATECH focuses on accelerating the deployment of existing technologies and works to enhance and implement industry-wide standards and guidelines for capital facilities projects. FIATECH's membership of over forty leading industry organizations includes a diverse sample of major facility owner-operators, both large and small facilities related engineering and construction service contractors, a wide variety of facilities related software vendors, and government and academic institutions.

In May 2001, FIATECH launched a Strategic Focus Area on eBusiness for Capital Facilities. This program group investigated the barriers to achieving successful eBusiness capabilities and conducted a detailed analysis of work processes, information flows, and data transfers to support the engineering, procurement, construction, operations and maintenance (EPCOM) life cycle for capital facilities. The investigation determined that there were significant impediments to automating and integrating the EPCOM work process and project management activities for configurable and engineered equipment.

In analyzing the detailed work and data flows for engineered equipment, it is clear that the data contained on engineered equipment data sheets and the procurement documents are involved in

many data transfers among people, software programs, and organizations. Developing and deploying industry standard XML schemas to support equipment data sheet information, equipment procurement information, and project information is a critical component of automating the EPCOM work flow, and represents a large potential benefit opportunity to reduce capital costs, shorten schedules, and reduce operating and maintenance costs associated with capital facilities and the equipment.

After investigating whether there were any available XML standards for this domain or industry initiatives working to develop the needed XML specifications, FIATECH started the "Automating Equipment Information Exchange" (AEX) Project in May 2002.

1.6 AEX Use Cases

From the list of potential equipment types important to the AEX Project sponsors, the project team chose centrifugal pumps and shell and tube exchangers as the initial scope, in recognition that these are some of the most common types of equipment in a typical capital facility. Over twenty transactions in the life of engineered equipment were identified as valuable for automating, and from this list of twenty, the AEX Project identified five as having high potential for significant economic benefits. These five scenarios include:

1. Request for Quote (RFQ)
2. Quote
3. Purchase Order (PO)
4. As-built
5. Bill of Materials

In making this assessment, the project noted that it was important to address usage scenarios that cross the organizational boundaries between the owner, EPC, and equipment suppliers. The rationale for this is that many companies are working on software interoperability and integration capabilities within their own companies, but they recognize that a common industry approach is needed to cross the organizational boundaries and offers the highest potential benefit for the industry. Figure 1 shows the equipment life cycle and the primary transactions, i.e., use cases, selected for the AEX Project.
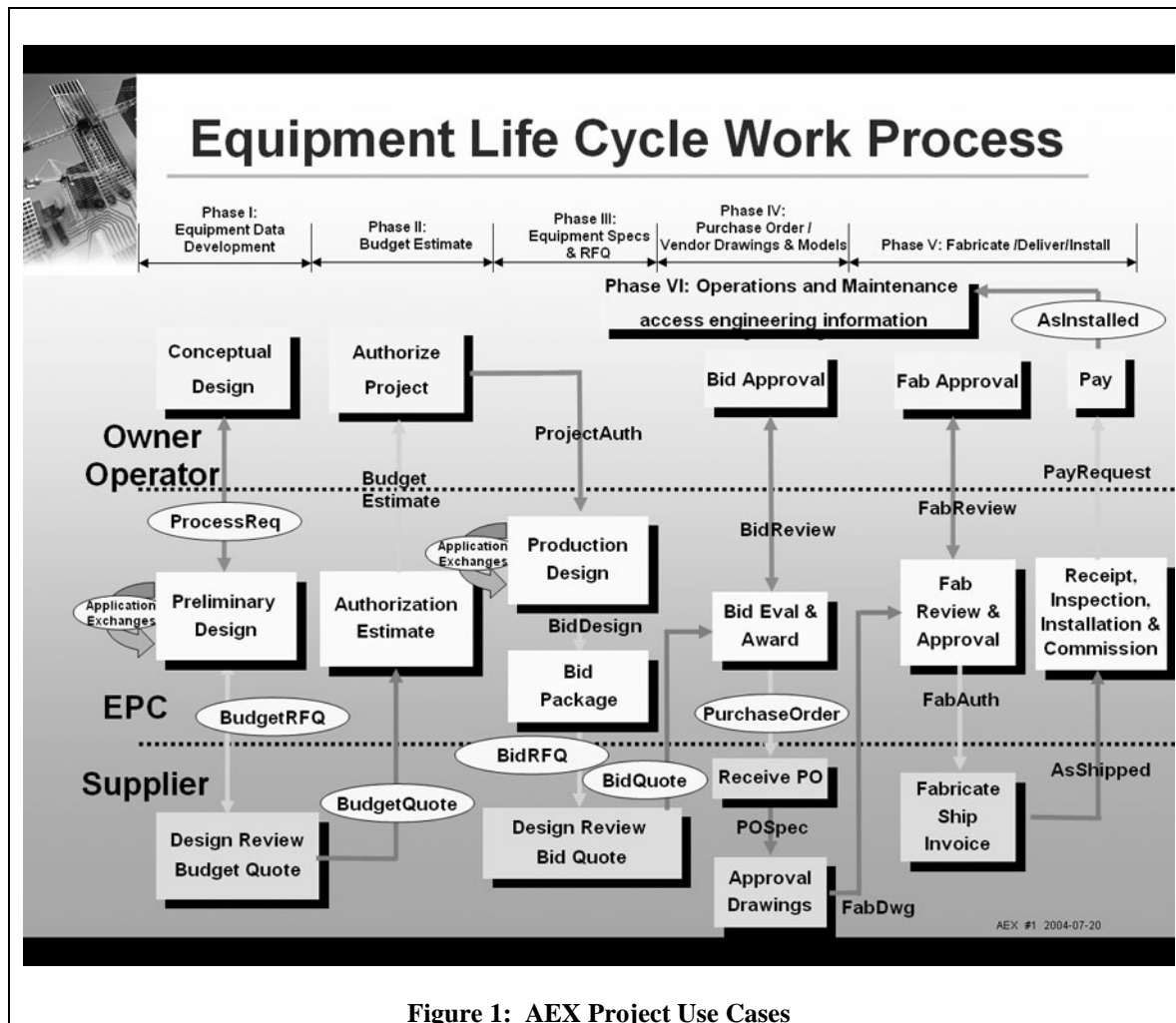
Figure 1: AEX Project Use Cases

In addition to identifying the above major usage scenarios, the AEX project did a limited survey of the work processes and the various software packages that produce and consume material properties and equipment information over the life cycle. In this analysis, the general types of software packages were identified, and the key documents and data packets that are used to transmit information from one software package to another were noted. As expected, the equipment datasheet is a key document for transmitting such information. The other key document type was the equipment list or bill of materials, which provides a subset of summary information about equipment items. In practice, equipment datasheets, equipment lists, and engineering drawings and models all need to be maintained in a synchronized state, but this task is difficult because different software packages are often used by multiple organizations to produce the various engineering documents. The key types of software and the key engineering information transmittal documents are shown in Figure 2.
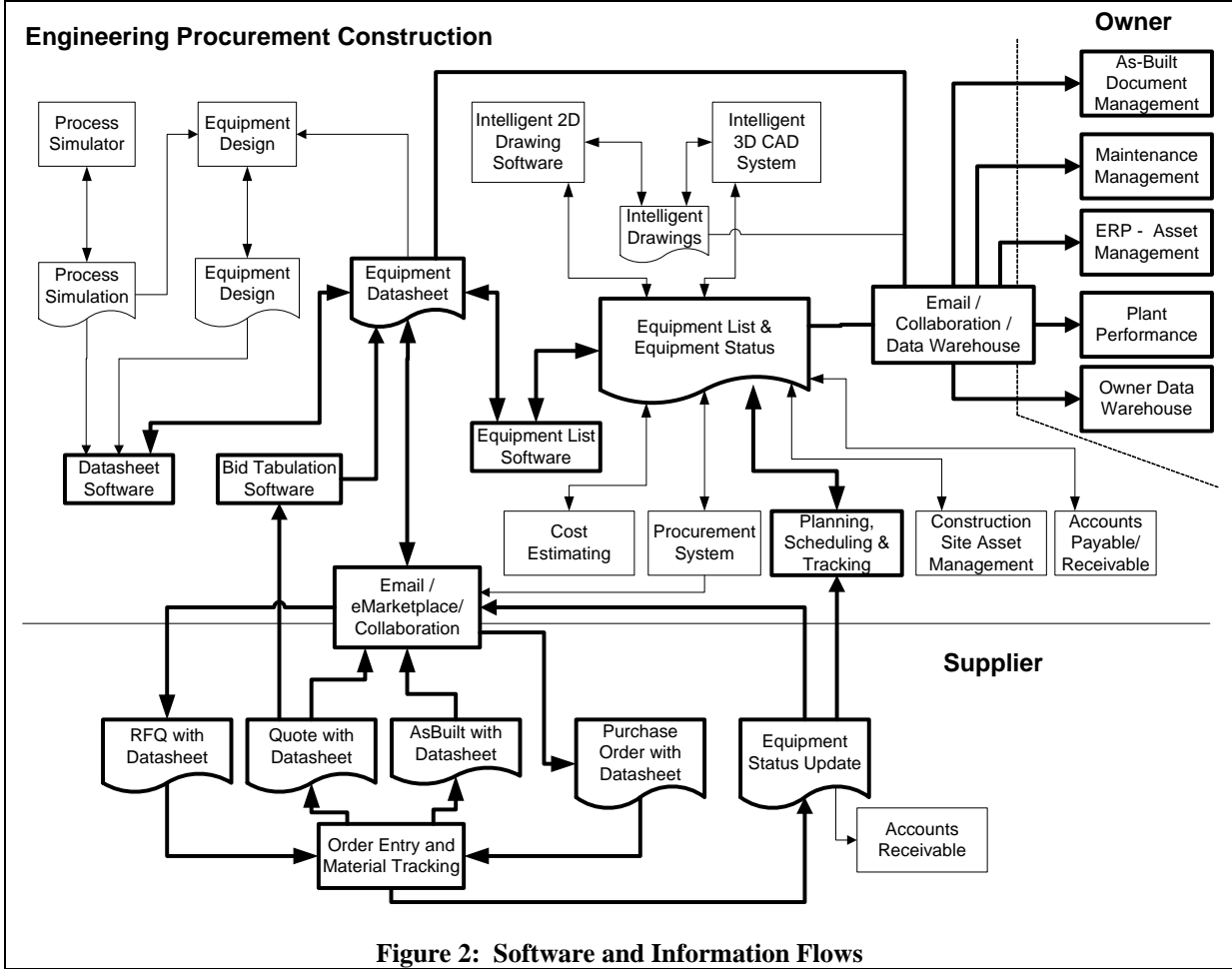
Figure 2: Software and Information Flows

1.7 Centrifugal Pumps

A centrifugal pump is a machine used to transfer liquids from one place to another through a piping system. Although centrifugal pumps contain a number of parts, the most important two are the impeller and its surrounding casing (Figure 3[21]). A power source rotates the impeller, and when the fluid enters the pump through a suction nozzle, it is directed toward the center of the rotating impeller. As the impeller rotates, liquid is accelerated from the eye of the impeller towards the impeller tip along the impeller vanes. The accelerating fluid causes a low-pressure area to develop that draws more fluid into the impeller eye. The fluid along the impeller vane is then affected by rotational force at the tips of the vanes, increasing the velocity of the fluid as it leaves the impeller. Larger increases in the velocity are achieved when the impeller's size or rotational speed are increased. Finally, the casing surrounding the impeller comes into play. It is designed to efficiently convert the kinetic energy
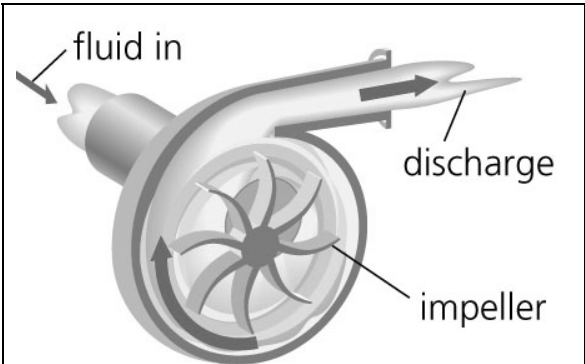
Figure 3: Simplified view of a centrifugal pump

of the accelerated liquid into pressure energy as the fluid velocity decelerates into the discharge piping, according to Bernoulli's Principle[22], as the fluid is discharged.

Centrifugal pumps were selected for examination in this semantic mapping report because of their economic significance in the construction supply chain stemming from their ubiquitous application. Among the most common machines used worldwide, pumps are second only to electric motors; and, among pumps, the centrifugal type is the most widely applied. In 2002 alone, U.S. manufacturers shipped 7.9 million centrifugal pumps with a total value of $1.53 billion dollars[23]. During the same year, more than 470 000 centrifugal pumps were exported with a total value of $3.92 hundred million and 13.3 million centrifugal pumps were imported with a total value of $3.91 hundred million[24]. The ubiquity of centrifugal pumps is a reflection of their versatility due to their broad pressure and load handling capabilities paired with relatively low maintenance costs. Centrifugal pumps can be found in nearly all applications where fluids are transferred such as HVAC/R systems, chemical processing, irrigation, sewage and sanitation plants, petroleum production and processing, electric power generation, and desalination plants.

2. BACKGROUND

The development of the IFC and AEX schemas, while differing in motivating factors, exhibit similarity in their design principles. This section discusses those principles and provides an overview of the pump schemas that are semantically mapped in this report.

2.1 IFC Design Principles and Pump-Related Modeling

The IFC model is meant to address the interoperability of a large, diverse collection of software applications used by AEC/FM practitioners throughout the life-cycle of a building project. As illustrated by the use cases described in section "1.4 ifcXML," an overarching goal of the IAI has been to improve the interoperability between software applications that represent different discipline and life-cycle views relevant to a project. So, for example, the organization and development of the IFC model emphasizes representing the common elements of a pump that would be useful to the architect who must reserve space for the pump, the civil engineer who must design the structural support for the pump, the mechanical or process engineer who must specify the pump characteristics based on system requirements, the project managers who deal with cost estimates, schedules, procurement and material management, and the facility staff who must operate and maintain the pump in service. Capturing and explicitly representing domain-specific information such as all possible performance characteristics relevant to a pump engineer is purposely deemphasized, although the IFC model has been designed to be extensible in such a way that domain-specific applications can define and use custom information elements that have no meaning in other domains.

2.1.1 IFC Model Architecture

In order to support coherent organization of the model as well as its maintenance and continued enhancement, the development of the IFC, from the outset, has adhered to a succinct set of design principles: modularity, reusability, and upward compatibility. These principles and

many other aspects of the IFC development process are described in several IAI documents[25, 26]. The design principles are introduced in Table 2, which depicts the data structures found in each of the four conceptual layers of the IFC model architecture. Figure 4, adapted from IAI documentation, clarifies the layered approach. Each of the boxes in this figure represents a topical IFC sub-model. The layer of sub-models at the lowest level is called Resource. The layer one level up is Core. Interoperability is the third layer and, finally, at the highest level, is the Domain layer. The data structures within each layer are allowed to reference data structures at the same or lower levels but not those from higher levels.
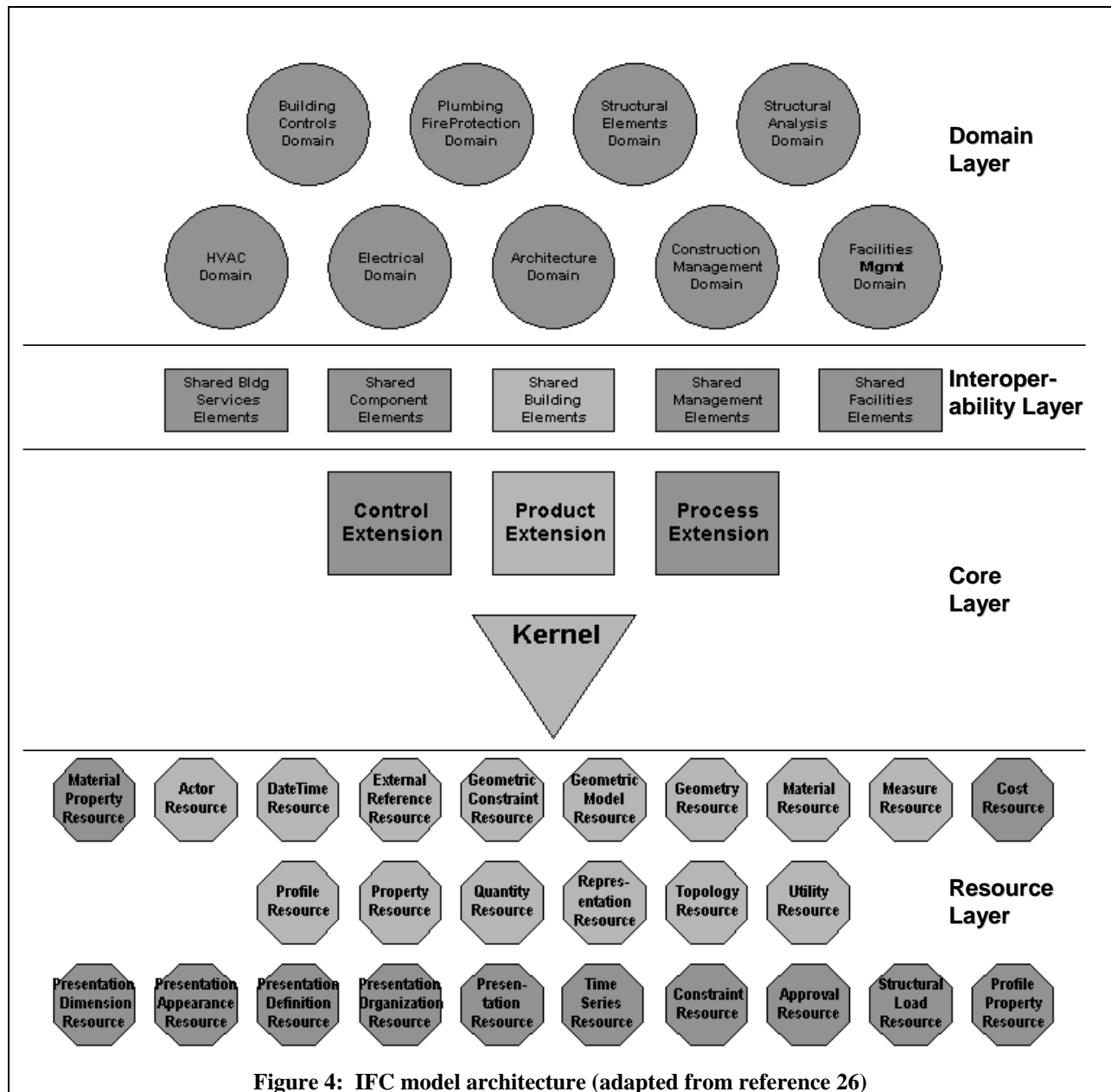

Figure 4: IFC model architecture (adapted from reference 26)

The Resource layer contains data structures that represent general purpose concepts that are widely used and are independent of domain. The IfcActorResource sub-model, for example,

contains data structures pertinent to the description of an actor, e.g., a person or an organization. In IfcActorResource one finds structures for encoding addresses and other personal and organizational information. Similarly, data models associated with the cost of an object can be found in the IfcCostResource sub-model, geometrical concepts are addressed within the IfcGeometryResource sub-model, etc. Complete documentation for all resources as well as the entire IFC model is available online[27]. It should be noted that, in accordance with the adopted STEP methodology, all of the data structures that compose the IFC model are defined in a single EXPRESS schema, irrespective of organizational layer or application domain (different sub-models are developed in different, cross referenced schemas, and are often spoken of separately, but the result is flattened into a single schema for use in information exchange). This is equivalent to being defined in a single XML namespace.

The Core layer, which delineates the basic structure and conceptual definitions of the IFC, is composed of two parts: the Kernel (IfcKernel) sub-model and the extensions to the Kernel called the Core Extensions (IfcControlExtension, IfcProcessExtension, IfcProductExtension). The Kernel sub-model defines data structures for describing low-level abstract concepts such as object, property, and relationship. These concepts are then used and specialized within the Core Extensions to provide higher-level but still abstract constructs such as product, process, and control, and the relationships between them. Overall, the Core layer represents the beginning of the amalgamation of basic concepts into those needed for representing AEC/FM-specific data structures defined in the Interoperability and Domain layers. As an illustration, IfcPumpType, whose schema is examined in this semantic mapping report, exists in the Domain layer IfcHvacDomain sub-model. A pump is a type of product and it has various properties. Therefore, the schema for IfcPumpType can reuse the general data structures defined in the Core layer for describing a product and its properties.

The IFC Core layer provides for the factoring of information about building objects into generic-, specific-, and occurrence-level data structures. This factoring capability is then propagated by inheritance to the relevant entities in the Interoperability and Domain layers. For example, generic information about pumps is captured in the structure of the IfcPumpType definition itself. Information about a specific pump type, e.g., information that would characterize a particular make and model of a pump, is captured in a populated instance of IfcPumpType. Information about an occurrence of a pump of a specific type, e.g., information that would characterize a particular pump uniquely identified by its tag or serial number, is captured in a populated instance of the companion IfcFlowMovingDevice. Instances of IfcRelationship constructs are used to interrelate the instances of the occurrence-level IfcFlowMovingDevice and specific-level IfcPumpType.

It should be noted that the meaning of occurrence-level data structures in the IFC model is overloaded in the sense that they can be applied both to occurrences of an object type in the abstract product model (e.g., a system design calls for one or more occurrences of a particular pump type, with each occurrence uniquely identified by its tag number) and to occurrences in the real world (e.g., one or more physical occurrences of the pump type will be procured, delivered, and installed on site to realize the occurrences of the pump type in the system design, with each occurrence uniquely identified by make, model, and serial number). It is likely that this

overloading will be disambiguated in some future version of the IFC as IFC information becomes more widely used in the operations and maintenance of facilities.

The factorization approach enables the exchange of information about product types without them being assigned to product instances, and enables the exchange of information about numerous product instances without repeating their type information or even without exchanging the type information itself by value rather than by reference (e.g., to a catalog item). Further, this factorization makes it possible not just to exchange multiple occurrences of a common product type but also to customize each occurrence by overriding attribute and property set values. This latter capability is a two-edged sword, since it easily can be abused.

The IFC Core layer also provides for the explicit representation of whole-to-part relationships, so that IFC data structures can be aggregated as needed in the instantiation of a complex object without defining it in advance in the IFC model. For example, a unitary pump assembly consisting of a pump, an electric motor, the base, and associated components such as a drain valve, sensors, and actuators, can be represented by a composite object that is an aggregation of representations of its individual parts.

The Interoperability layer contains a collection of data structures that define concepts or objects common to multiple AEC/FM domain models. These data structures are organized into five sub-models:

1. IfcSharedBldgElements, which define the principal architectural components of the building structure such as walls, beams, stairs, doors, and windows;
2. IfcSharedBldgServiceElements, which define concepts common among building services (e.g. HVAC, plumbing, electricity) such as flow and distribution systems;
3. IfcSharedComponentElements, which define different types of small parts such as fasteners and accessories. Also included within IfcSharedComponentElements are schemas for representing building element edge forms as features, such as chamfered and rounded edges;
4. IfcSharedFacilitiesElements, which define basic concepts used in facilities management such as assets, furniture types, service life, and occupants;
5. IfcSharedMgmtElements, which define basic common managements concepts used throughout the lifecycle of a building such as costs, project orders, and project order records.

Finally, the Domain layer contains categorized data structures for specific AEC/FM domains such as architecture, structure, plumbing, and HVAC.

2.1.2 IFC Pump Model

The IfcPumpType data model is defined in IfcHvacDomain in the Domain layer along with other HVAC-related equipment types (Table 3). Each of these data structures draws upon definitions supplied in the Resource, Core, and Interoperability layers as described above.

IfcAirTerminalBoxType	IfcCooledBeamType	IfcFanType	IfcPumpType
IfcAirTerminalType	IfcCoolingTowerType	IfcFilterType	IfcSpaceHeaterType
IfcAirToAirHeatRecoveryType	IfcDamperType	IfcFlowMeterType	IfcTankType
IfcBoilerType	IfcDuctFittingType	IfcGasTerminalType	IfcTubeBundleType
IfcChillerType	IfcDuctSegmentType	IfcHeatExchangerType	IfcUnitaryEquipmentType
IfcCoilType	IfcDuctSilencerType	IfcHumidifierType	IfcValveType
IfcCompressorType	IfcEvaporativeCoolerType	IfcPipeFittingType	IfcVibrationIsolatorType
IfcCondenserType	IfcEvaporatorType	IfcPipeSegmentType	

Table 3: Data Models within IfcHvacDomain

IfcFlowMovingDeviceType, for example, is defined in IfcSharedBldgServiceElements within the Interoperability layer. IfcFlowMovingDeviceType defines common properties of flow moving devices, such as fans, compressors, and pumps. Therefore, IfcPumpType, IfcFanType, and IfcCompressorType all inherit the general characteristics of flow moving devices defined in IfcFlowMovingDeviceType. This modular reuse is more efficient and less error-prone than tediously defining separate data models for fans, compressors, and pumps. Distinctions are drawn among different equipments types through the actual data entered for each element of the related data structure and through the use of property sets.

Figure 5 contains graphical representations of the ifcXML schemas for IfcPumpType and IfcFanType. Their data structures are the same because they have inherited their definitions from IfcFlowMovingDeviceType. That one schema describes a pump and the other a fan is not simply a matter of calling them different names. Within the schemas, portions of the inherited IfcFlowMovingDeviceType definition have been overridden. Specifically, PredefinedType in IfcPumpType is declared to be IfcPumpTypeEnum, an enumeration of different kinds of pumps whereas PredefinedType in IfcFanType is declared to be IfcFanTypeEnum, an enumeration of different types of fans. As described in section "1.3 ifcXML," these schemas have been translated mechanically from the EXPRESS schema for IFC using the ifcXML language binding for EXPRESS. They exist in a single namespace as a result of being translated from a single EXPRESS schema.

Figure 6 contains a graphical representation of the ifcXML schema for IfcFlowMovingDevice. Recall from the previous discussion that this element represents an occurrence of a pump defined by a related IfcPumpType, so many of the optional elements in this schema, notably Name, Description, ObjectType, and Representation, which appear by virtue of the original definition of the IFC entities in EXPRESS, do not add new information. The ObjectPlacement deals with the location of the pump in a facility model and has no counterpart in AEX. For the purpose of the current mapping study, only the Tag identifier is relevant.
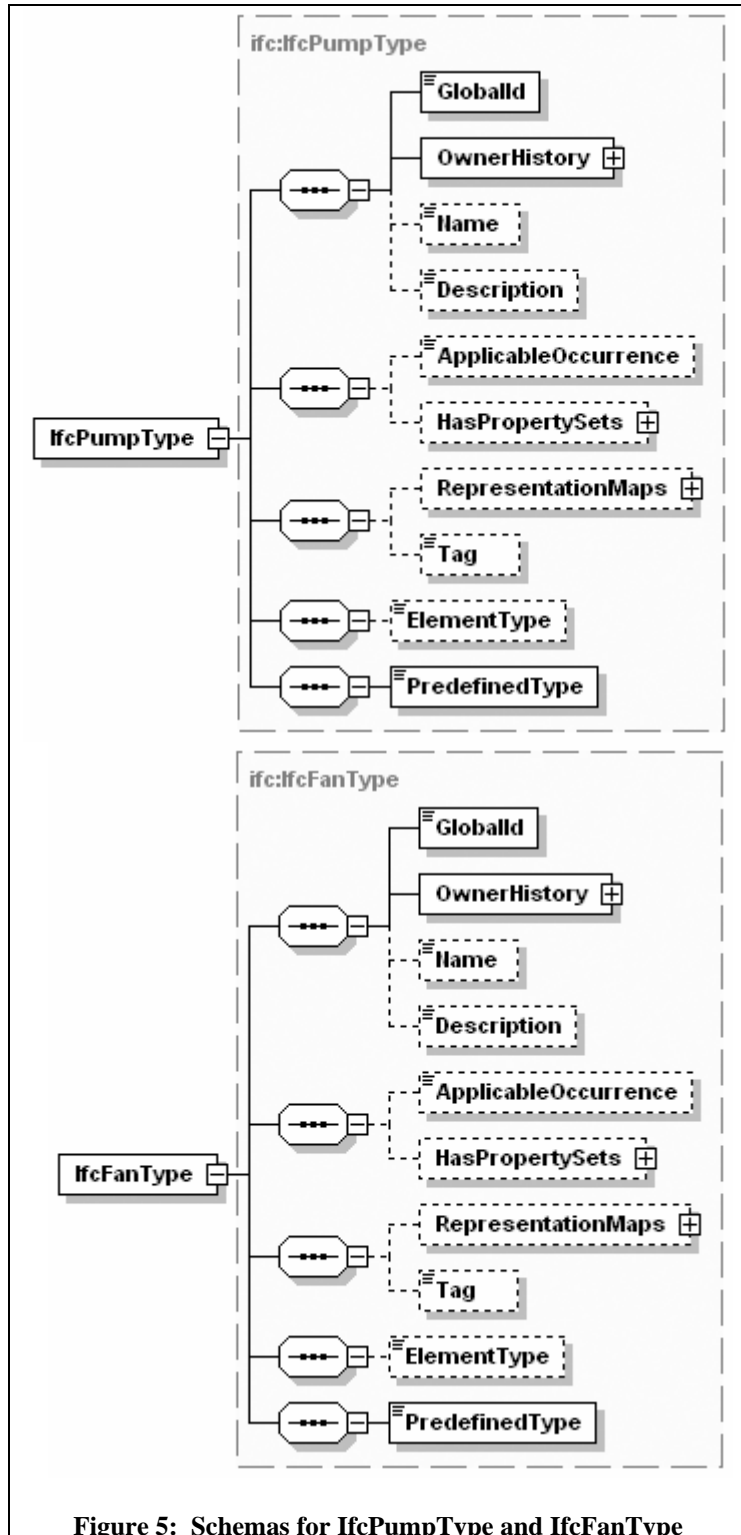
Figure 5: Schemas for IfcPumpType and IfcFanType

IfcPumpType and IfcFanType are further distinguished by their respective property sets, which would be included within HasPropertySets in an actual XML document containing data. Property sets are found in the Core, Interoperability, and Domain layers of the IFC model and provide a mechanism for expressing pertinent differences among similar concepts as well as a means of extending the IFC. The property set approach allows an application to exchange data definitions and data instances in the same file without requiring a change to the underlying EXPRESS schema. As the IFC work has progressed, certain sets of property data have been agreed by the IAI participants, with the result that each successive specification has included an expanding collection of "standard" property sets. In ifcXML2, for example, the standard property sets for IfcPumpType are Pset_PumpTypeCommon and Pset_FlowMovingDevicePump, while those available for IfcFanType are Pset_FanTypeCommon, Pset_FlowMovingDeviceFan, Pset_FanTypeSmokeControl, and Pset_FlowMovingDeviceFanCentri-fugal.

IfcFlowMovingDevice is further elaborated in ifcXML2 by two standard property sets that apply only to real-world occurrences. As implied by their names, Pset_PumpPHistory and Pset_FanPHistory allow the capture of performance history data for a pump or a fan, respectively.

The descriptions for each element of the IfcPumpType schema and its property sets as well as the related property set Pset_PumpPHistory are given in section "3.2 Results."

17

2.1.3 Units of Measure

The IfcMeasureResource sub-model provides extensive support for units of measure, allowing any unit to be defined and used. This support is taken directly from STEP, which draws, in turn, on the ISO standard[28] for SI units and the construction of other units from them using explicit dimensional analysis. This approach was developed in STEP to cope with the multiplicity of units in use around the world that varied by industrial sector, by discipline, and by locale.

All measures used in an ifcXML document must have their appropriate units explicitly declared

Figure 6: Schema for IfcFlowMovingDevice

in the same document. A unit is declared in terms of its construction from the base SI-units and not by selection from an enumerated list of predefined units. Where measures are expressed in terms of a unit within the SI system, this declaration is relatively simple; where Imperial or other units are used, this declaration is more complex. Once declared, a unit can be referenced as often as needed.

2.1.4 Documents

In the ifcXML world, paper documents and their electronic equivalents are "atomic" entities, which can be named, classified, referenced, approved, tracked, and so forth, but which are not modeled in terms of their information content. Hence, an ifcXML document may include data about a pump, say, as well as data about paper documents (or their equivalent, such as word-processor files) that pertain to the pump. The paper documents themselves, however, are opaque to ifcXML. The ifcXML does not model explicitly the content of a pump data sheet for example, and generating, verifying, or reading the content of such a document is out of scope for ifcXML.

This dichotomy grew out of the original IFC modeling philosophy that paper documentation, including drawings, specifications, and reports, represents particular views of an underlying information model. The IFC approach was to model the information and not the views of the information. This philosophy is subject to re-examination as the understanding and use of IFC matures. The ifcXML effort facilitates the possibility of interweaving model-based data and document-based views of the data because of XML's document-markup heritage.

2.1.5 Systematics

The ifcXML schema supports the identification and characterization of a system or subsystem and the composition of the (sub)system in terms of its constituent components. By this means, the functional requirements of a system can be related to the functional and physical specifications of its parts. This capability is not considered further in the current study.

2.1.6 External References

The IFC model includes mechanisms to make references between instance data sets, whether in exchange files or in repositories. These mechanisms were intended to facilitate the use of traditional catalogs and libraries, rather than the arbitrary distribution of information across multiple data sets. However, the emergence of distributed client-server technologies such as web services has triggered a rethinking of the subject. Future versions of ifcXML may well implement different mechanisms that facilitate referencing and exchanging partial sets of information. This capability is not considered further in the current study.

2.2 AEX Design Principles and AEX Centrifugal Pump

These XML schemas have been developed with an object-oriented schema framework to support equipment item data, process material data that appear on equipment datasheets, procurement data, and relevant project data. The initial subject focus of this work is:

- Basic equipment information found on various equipment lists and bill of materials documents;
- Equipment datasheets for centrifugal pumps and shell and tube exchangers, which include operating conditions and performance requirements;
- Process materials, associated properties, calculation methods and experimental property data;
- Equipment documents used over the life cycle of capital facilities.

2.2.1 Structural Overview of AEX Schemas

While the scope is initially limited to these types of information, schema design has been approached with the requirement that the base schema structures be capable of handling any equipment item, any engineering document, and any material property for any information exchange usage scenario. Therefore, it is possible to use the schema in its current form to describe any equipment item and any engineering document. At this point in time, in-depth attribution has been provided only for shell and tube exchangers, centrifugal pumps and their associated datasheets as well as an extensive physical properties model for materials, including provision for experimental data, calculation models, parameters and calculated properties, and property curves and property tables. Because of the initial effort to create a robust, reusable schema framework, it is anticipated that extension of the object-oriented capital AEX XML framework to handle other equipment items will be very straightforward and cost-effective.

There are four basic parts to the AEX XML architecture as illustrated in Figure 7:

1. *Core data type schemas* for extensions to basic data types to support engineering requirements (these are essential extensions added to the "XSD" base data types provided in the W3C XML Schema standard);
2. *Core engineering object schemas* for reusable base engineering objects that can be used by multiple engineering disciplines and subject domains;
3. *Subject engineering object schemas* that provide schemas related to specific equipment items;
4. *Collection-container schemas* that are used to model engineering document schemas. The collection-container schemas allow core and subject-specific engineering objects to be combined in various ways to support various data transactions and usage scenarios.
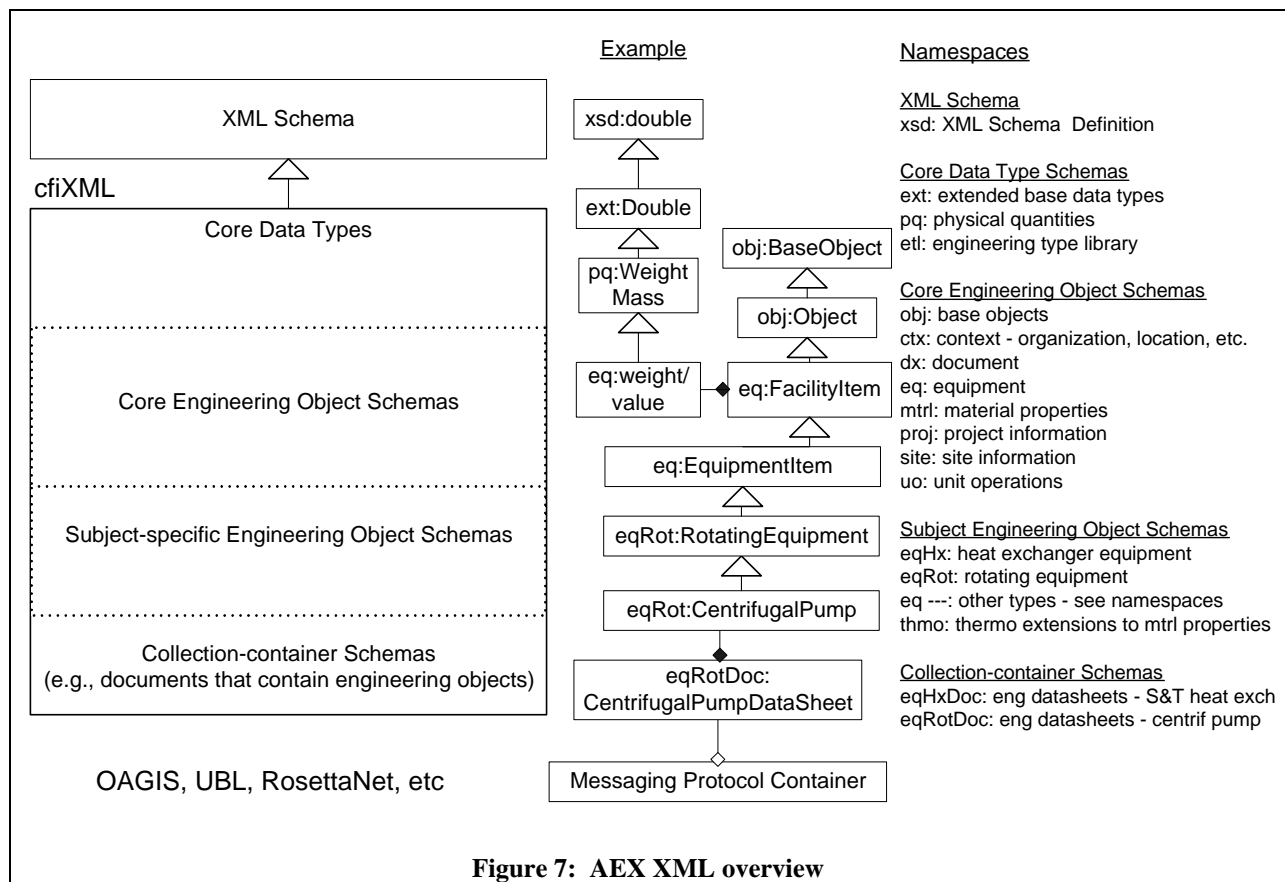

Figure 7: AEX XML overview

Figure 7 illustrates how these parts relate to each other, to standard XML schema definitions, and to various messaging protocol containers that are currently being developed by various industry groups. Subsequent sections will introduce each of these major parts of the AEX XML schema architecture.

2.2.2 Namespace Overview

XML documents need to have unique names for XML tags (elements) that have specific meanings. In a small schema, with relatively few tags, it is fairly easy to define and maintain unique tags. In domains, such as the capital facilities industry, where multiple collaborating groups working independently could potentially define thousands of element definitions, it becomes more difficult to ensure uniqueness across the various parts of the schema.

To accommodate this problem, XML schema provides for the use of "namespaces." Namespaces provide the ability to define a collection of conceptually related data elements, where uniqueness is only required within the namespace. This allows different namespaces to use the same tag in different ways. In order to accommodate global uniqueness, XML documents can use multiple namespaces in the same document. Element tags in the document have a shorthand namespace prefix, e.g., "pq:" which is shorthand for physical quantities. In the XML document, the combination of the namespace prefix and the element tag ensures that the tag is unique within the document. For ease of maintenance for large schemas, each namespace may also include one or more separate files that combine to provide all the types and elements required in a single namespace.

In order to ensure interoperable XML documents that use multiple namespaces, the namespaces themselves are required to be named uniquely. It is convention to use a common globally unique identifier string called a Uniform Resource Identifier (URI) that is "owned" by the organization developing the schema and is constructed like a Web URL.

For the AEX XML, all namespaces belong to a common "root" URI, specifically http://www.cfixml.org, referring to the capital facilities industry. To the end of this common root, a short identifier is added separated by a forward slash. For example, the "pq" namespace would have a full unique identifier of "http://www.cfixml.org/pq." This string then uniquely identifies the physical quantities namespace, which ensures there is no conflict with other industries and schema data definitions. By convention, the shorthand prefix tag "pq:" is typically assigned in a schema declaration to mean the same thing as "http://www.cfixml.org/pq" so that the XML files are much more human readable, yet maintain uniqueness for the XML parsing program. In this document the file name, namespace prefix and full namespace qualifier will often, for convenience and simplicity, be treated as synonymous. Figure 8 illustrates the capital facility industry namespaces and their relationships to each other.

The namespaces shown in Figure 8 were defined to meet the following general requirements:

- To enable conceptually-related schema elements to reside in the same namespace;
- To enable namespaces to be easily imported and reused in other derivative schemas;
- To separate domains that are likely to be developed and maintained by separate organizational groups;
- To anticipate the need for collection-container XML documents to use only the relevant portions of a potentially very large suite of XML schema.

In Figure 8 the dashed arrow lines indicate a usage dependency of a namespace upon the namespace that is pointed to. For example the "eq" namespace depends on the "mtrl" namespace to define the construction material of an equipment item. The open arrow lines indicate that complex types in a namespace extend complex types in the referenced namespace. For example, the "ext" extended data types are extension types from the base "xsd" namespace, and the "eq" EquipItem complex type extends from the "obj:Obj" complex type.
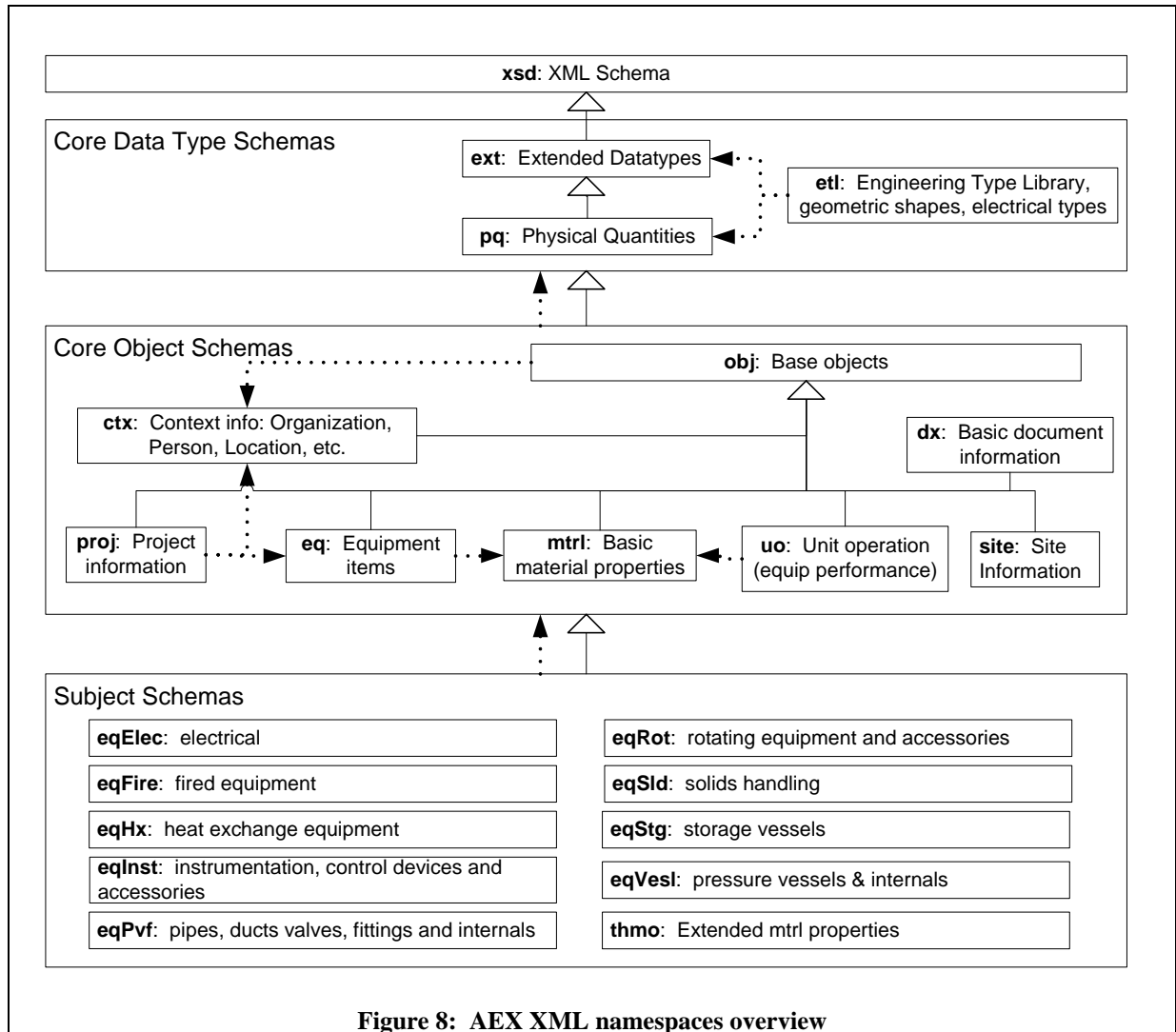
Figure 8: AEX XML namespaces overview

2.2.3 Core Data Types Overview

As an underlying basis, AEX XML extends the base XML Schema standard by providing a generalized framework for handling engineering data requirements for any engineering domain. These core domain-independent schemas are divided in two main parts – the core data types and the core objects (see next section).

22

The core data type schemas extend the standard XML Schema data types for engineering needs, including consistent means for handling units of measurement, handling of vector data and table data, and underlying mechanisms to provide annotation and revision tracking for any element in the schema. There are three core data type namespaces:

- ext: extends the XSD basic data types to include base attributes for annotation and revision tracking at the element level and to provide vector and table data types;
- pq: defines physical quantity base data types that associate the appropriate units of measurement labels, including enumeration lists of valid units of measure;
- etl: defines some reusable complex data types, which are not Objects (see next section), including simple shape geometries and electrical source and area classification specifications.

As shown in Figure 8, these extensions use the object-oriented mechanism of inheritance, extending the base data types provided in the XSD schema. Inheritance allows the extended, or derived, data types to include all the features of the base data type, with extensions to handle additional requirements. For example, starting with xsd:double and adding the first data type extension defines the data type "ext:Double" data type. The ext:Double data type includes three attributes for handling revision tracking information and associated annotation for each data element in the schema. Next, units of measurement are handled by providing another data type extension to "pq:WeightMass." The element pq:WeightMass extends the ext:Double data type by associating mass units of measurement with ext:Double. When extended finally to the engineering domain, any element in the domain schema that has mass weight units of measure can be defined simply by extending from the pq:WeightMass data type. At this final engineering object level, when creating schemas containing engineering domain elements in context, the underlying engineering data requirements are satisfied "for free" by using the inheritance mechanism without having to duplicate this information many times in the schema.
Similar data type extensions exist for all the XSD base data types including integers, Booleans, dates, strings, etc. In addition to base data type scalar values, extended base data types are defined for vectors and tables of XSD base data types. For units of measurement, in addition to weight mass illustrated above, definitions are provided for over 80 types of base physical quantities, including pressure, mass, mole, length, area, volume, flow rates, energy flows, etc.

2.2.4 Core Objects Overview

While defining the AEX XML framework, a number of core objects were discovered that could be reused and applied across multiple subject domains. These include:

- obj: defines base object data types that allow engineering objects to be uniquely identified and referenced by other objects, as well as providing facilities for recording object transactions, and tracking object type and status;
- ctx: objects that describe context information including organization, person, location;
- dx: objects that describe document information, identification, literature citations;
- eq: objects that describe equipment items and parts;
- mtrl: objects that describe construction and process materials and material properties;

- proj: objects that contain basic descriptions of projects and planned and actual status of activities;
- site: objects that describe site information, facilities, facility systems, environmental data;
- uo: objects that describe unit operations (equipment performance) and streams.

The key idea of a core object is that it can be thought of as a "container" which can be accessed by its name, like understanding the contents of a box without having to open the box. Providing this mechanism to uniquely identify objects anywhere in the engineering schema enables the schema (and resulting XML instance files) to make reference to and use these objects elsewhere in the schema or to find instance data elsewhere in the same file, or in different files on the network without having to replicate the schema or the data.

In general, engineering objects contain a large number of individual engineering data elements that are typically grouped together in logical ways. For example, in Figure 7, a FacilityItem is a type of Object that contains a number of elements that describe the facility item, such as the element weight/value. There are many types of items in a facility, but all of them can have the common characteristic of having a weight/value.

The rationale for separating unit operation data from equipment data should be noted briefly. Unit operations generally describe process functional performance requirements, while equipment data generally describe physical items of equipment. Also, in practice, a unit operation may have multiple associated items of equipment, e.g., a distillation column, and an equipment item may be modeled using multiple unit operations, e.g., a three-phase separator. Finally, the domain users of unit operations are different from the domain users of physical equipment information. This model separation also enables more convenient support of multiple business work process use cases – some focused on process simulation, and others focused on equipment design and specification. Of course, when defining engineering documents for equipment such as data sheets, there are embedded associations in the XML schema between unit operation and stream data for a particular item of equipment.

2.2.5 Subject Schemas Overview

The subject-specific schemas build on and extend from the core object schemas in a particular subject domain. For example, centrifugal pumps and shell and tube heat exchangers are types of Equipment Items. Shell and tube exchangers are part of the eqHx namespace while centrifugal pumps are part of the eqRot namespace. This reflects the fact that two different domain disciplines work with these different types of equipment. In the 'thmo' namespace, used by thermodynamic specialists and process engineers, the basic material properties core schemas are extended to describe the experimental or calculation methods, and to produce both simple and complex property tables associated with various experiments and calculations.

The currently identified subject schema namespaces are:

- eqElec: electrical equipment types;
- eqFire: fired equipment types;

- <u>eqHx</u>: heat exchanger equipment types;
- <u>eqRot</u>: rotating equipment types;
- <u>eqSld</u>: solids handling equipment types;
- <u>eqStg</u>: storage vessel equipment types;
- <u>eqVesl</u>: vessel equipment types;
- <u>eqPvf</u>: piping, valves and fittings equipment types;
- <u>eqInst</u>: instrumentation and control system equipment types;
- <u>thmo</u>: thermodynamic parameters and extended material properties data.

In Figure 7, it should be noted that eqRot:CentrifugalPump objects inherit characteristics from an EquipmentItem object, which in turns inherits characteristics from FacilityItem and from there the base characteristics of obj:Object, the base for all engineering objects in the AEX XML schemas. The FacilityItem object contains the weight/value element discussed earlier. The EquipmentItem object describes characteristics associated with all equipment items, such as the equipment tag. The CentrifugalPump object contains a number of additional elements that are specific to centrifugal pumps. The idea of an inheritance/extension/generalization hierarchy is depicted in Figure 9.

Figure 9: Inheritance/Extension mechanism illustrated conceptually for a centrifugal pump
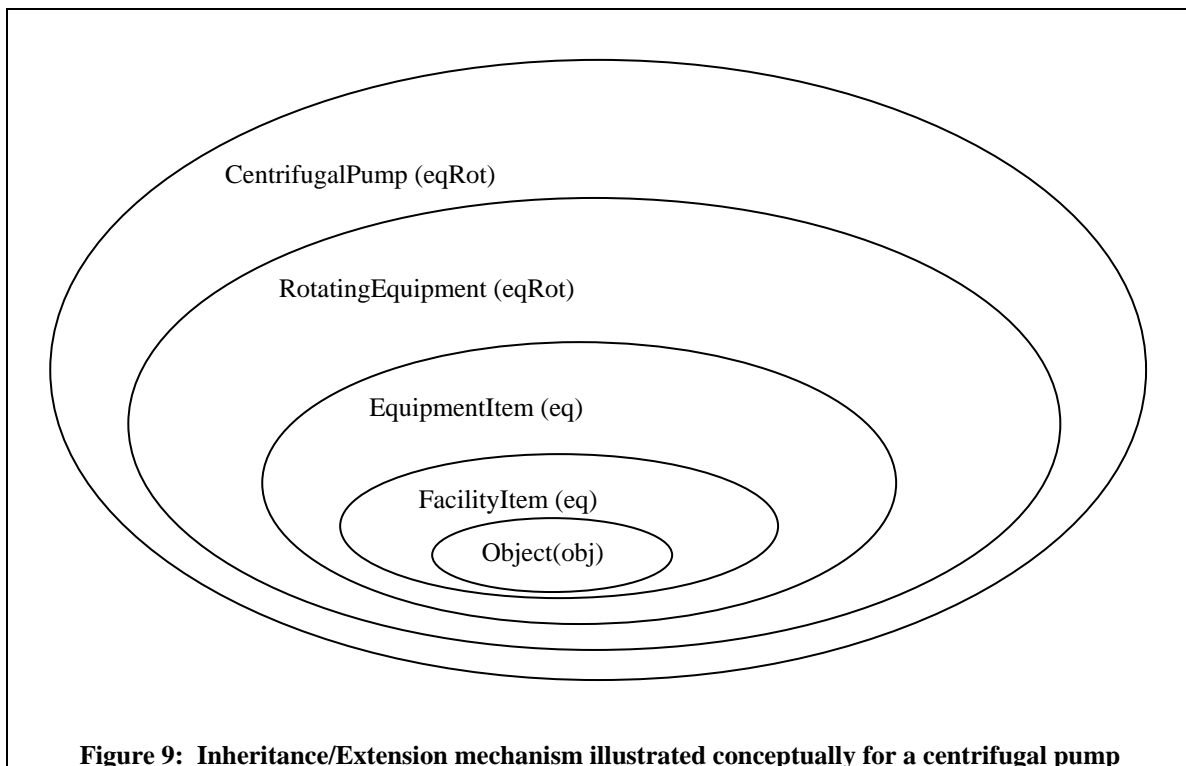
2.2.6 Equipment Types Classification

In terms of defining namespaces for the subject-specific schemas, similar equipment items have been grouped together – heat exchangers, rotating equipment, instrumentation etc. This enables a number of conceptually related equipment items that are likely to share data element structures

to be grouped together in a namespace. In usage, this means an XML document describing a pump does not have to be burdened with bringing in all the schemas associated with heat exchangers or instruments. The starter list of equipment types, grouped by equipment namespaces, is summarized in Figure 10.

Figure 10: Equipment Types Classification
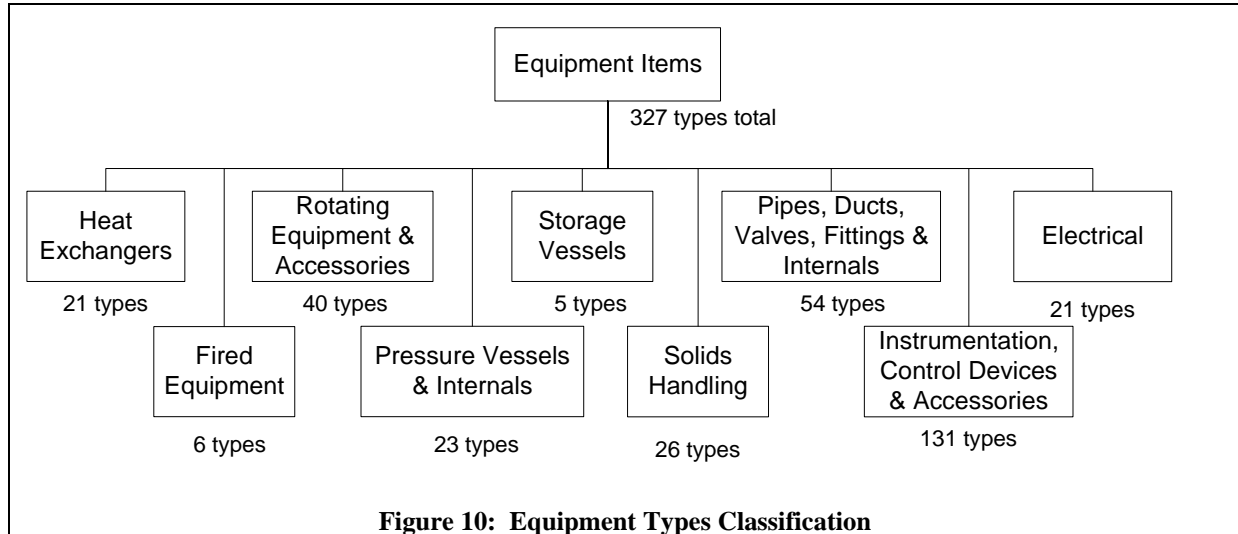
Using the equipment types that are identified by name only in the nine equipment namespaces above, it is possible to construct summary equipment lists or generic data sheets that contain basic information about the equipment item, e.g., tag, manufacturer, model number, cost, weight, etc. In addition to the summary information on any of the above equipment types, the following equipment types were developed in detail:

- Electrical – Motor (partial model for centrifugal pump driver support);
- Heat Exchangers – Shell and Tube Exchangers;
- Rotating Equipment in general including Rotating equipment common parts, e.g., bearing assemblies, shafts, seals, etc.;
- Centrifugal Pumps.

2.2.7 Container-Collector Overview

To the engineering object schema, some basic data models have been added to handle information associated with engineering, project and commercial documents.

Documents are modeled essentially as collector-container "views" of the underlying engineering and project object data. Therefore, they can be constructed more or less arbitrarily as needed to suit the purpose of any given data transaction and usage scenario.

In the initial set of the capital facilities engineering XML schemas, some standard engineering and project documents have been provided, such as a centrifugal pump datasheet, a shell and tube exchanger datasheet, and a material property data table document to serve as a document for

a set of experimentally measured or calculated properties. Also included is a simple "order" document that can be used for inquiries, bids, and purchase orders.

Many document types can be assembled using the same base underlying data including process simulation inputs and results, equipment design program inputs and results, and, in the future, supporting graphical documents such as process flow diagrams (PFDs), piping and instrumentation diagrams (P&IDs), isometrics, etc.

3. SEMANTIC MAPPING

The semantic mapping process included identifying an appropriate methodology, documenting the mapping results, and developing sample ifcXML and AEX documents from a test artifact.

3.1 Methodology

The IfcPumpType and IfcFlowMovingDevice schemas and their associated property sets address those data relevant to information exchanges in the context of a whole building project as determined by the IFC data model developers working in the HVAC domain. The AEX effort is focused on technical information that describes engineered equipment generally found on equipment data sheets and equipment lists along with the procurement information. Two consequences of these distinct viewpoints are immediately evident: the schemas differ in size and scope. The AEX centrifugal pump schema contains much more detail than the corresponding portions of the IFC schema and related ifcXML schema. The extensibility offered by the object aggregation and property set approaches taken in the IFC model means that these differences are more apparent than real, but using this extensibility also means that some of the semantic content of an IFC exchange may be defined by the sending application rather than by the IFC specification. Of course, harmonization of the IAI IFC and the AEX schemas could result in the migration of critical data elements into appropriate standard property sets in the IFC specification and, consequently, in ifcXML. This would require refinement of the use of generic, specific and occurrence in ifcXML to map to the AEX structures which are designed to support the specific and the occurrence.

The semantic mapping is worked from the IFC side, where the explicitly defined elements of IfcPumpType and its standard property sets, Pset_PumpTypeCommon and Pset_FlowMovingDevicePump, are mapped to their equivalents in AEX. Since IfcPumpType is actually a data model for a generic pump, no specific schema exists for a centrifugal pump. Rather, an XML document prepared using IfcPumpType distinguishes its use for a centrifugal pump by the data entered into ElementType and PredefinedType. Specifically, PredefinedType would be set to "userdefined" (<PredefinedType>userdefined</PredefinedType>) and ElementType would be set to "centrifugal" (<ElementType>centrifugal</ElementType>).

The use case implied by this mapping is one in which a design of a building or facility created using IFC-compliant software includes the preliminary functional specification of a pump (e.g., an instance of IfcPumpType) in the context of a system. This preliminary specification must be extracted from the IFC model and passed to AEX-compliant software used in detailing the pump design and procuring the resulting equipment. Given the current use of information systems for

equipment design, procurement, and installation processes in the capital facilities, an early use of the ifcXML to AEX mapping will likely include the export of an equipment design, both functional and physical specifications, to the capital facilities design system. This would benefit from further alignment of these specifications and documenting a mapping from AEX to ifcXML.

Because of its relevance to pumps in the operation phase of their life cycle, the standard property set Pset_PumpPHistory is also considered in the semantic mapping to AEX. As described in section "2.1.1 IFC Model Architecture," this property set is associated not with the specific-level IfcPumpType but with the occurrence-level IfcFlowMovingDevice. Since instances of the latter reference instances of the former, the property set is related, albeit indirectly, to IfcPumpType. The semantic mapping of IfcFlowMovingDevice itself is not considered because its explicit attributes do not contribute to the specification of a pump. Rather, its attributes deal with project-specific context information such as the location and orientation of a pump in a facility and its connections to other elements.

By default, every AEX semantic equivalent will rest within the centrifugalPump element and, therefore, "centrifugalPump" will not appear as part of the equivalency text. Because AEX uses namespaces, the namespace prefixes, e.g., "ctx:," have been included in the element mapping for clarity. For more details on namespaces, refer to section "2.2.2 Namespace Overview" and the documentation and schema definitions in the AEX Public Release Version 1.0[29]. In some cases in the mapping tables, there are XML Paths that navigate to a particular element in the AEX schemas and these are noted using XPath notation, e.g., "eq:id/tag" means in the 'eq' namespace, element id, subelement tag.

Lastly, ifcXML2[30] and AEX – Public Release Version 1.0[31] were used for the preparation of the semantic mapping.

3.2 Results

The mappings found among the tables in this section concentrate on the elements in the AEX centrifugal pump schema that have essentially the same descriptions, i.e., meaning, as the IFC elements. Each table contains five columns, which, from left to right, are: 1) the IFC element; 2) the IFC description of that element; 3) the AEX semantic equivalent of the IFC element, where available; 4) the AEX description of the equivalent element; and 5) remarks concerning the semantic mapping of the elements, where appropriate.

The semantic mapping occurs within eleven tables, as follows:

1. IfcPumpType
2. IfcPumpType: OwnerHistory
3. IfcPumpType: OwningUser
4. IfcPumpType: ThePerson
5. IfcPumpType: TheOrganization
6. IfcPumpType: Addresses
7. IfcPumpType: IfcPostalAddress

8. IfcPumpType: IfcTelecomAddress
9. Pset_FlowMovingDevicePump
10. Pset_PumpTypeCommon
11. Pset_PumpPHistory

The first eight tables map the specific product-related elements regarding a pump. The Pset_FlowMovingDevicePump table maps the common properties of a flow moving device as they relate to a pump. The Pset_PumpTypeCommon table maps the common descriptive properties of a pump and the Pset_PumpPHistory table maps the performance history characteristics of a pump.

IfcPumpType				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
GlobalID	Identifier that is unique throughout the software world. This is also known as a Globally Unique Identifier (GUID) or Universal Unique Identifier (UUID) by the Open Group.	**obj:externalGUID** attribute for **centrifugalPump** element or **obj**:**organizationID** attribute (Universal Unique Identifier) plus the **obj**:**objected**	Globally unique identifier for external objects.	**GlobalID** is a required element; **externalGUID** is an optional attribute, **objectID** is a required element
OwnerHistory	Assignment of the information about the current ownership of the object, including owning actor, application, local identification and information captured about the recent changes of the object.	**obj:transaction** element	Transaction table for the object.	**OwnerHistory** and **transaction** are composed of additional elements. See the detailed breakdown of **OwnerHistory** containing a lower-level mapping to **transaction** in the separate table "IfcPumpType: OwnerHistory."
Name	Optional name for use by the participating software systems or users.	**obj:name** element	Name of the object.	
Description	Optional description, provided for exchanging informative comments.	**obj:description** element	Description of the object.	
ApplicableOccurrence	Optional element for defining the data type of the object.	See remarks.		In general extensions to W3C standard data types are explicitly used in AEX schemas. The AEX **customCentrifugalPump** element, which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **ApplicableOccurrence**.
HasPropertySets	List of unique property sets.	See remarks.		AEX contains an extensive treatment of the properties of centrifugal pumps among its elements. Mappings to these elements for IFC property sets Pset_FlowMovingDevicePump, Pset_PumpTypeCommon, and Pset_PumpPHistory are provided in separate tables below.

IfcPumpType *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
RepresentationMaps	Representations are the structures that relate a collection of aspects, generally shape and topology, of product data to a context. A representation map is the identification of a representation, and of a representation item that is an element in the representation, for the purpose of mapping.	See remarks.		The IFC support multiple, CAD-quality geometry representations. Only some basic data about simple equipment shape geometries (cylinders and boxes) are included in the AEX schemas. The AEX **customCentrifugalPump** element, which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **RepresentationMaps**.
Tag	The tag (or label) identifier at the particular type of a product, e.g. the article number. It is the identifier at the specific level.	**eq:id / tag** element	Tag identification name.	The semantics are exact only if the IFC model contains a separate IfcPumpType instance for each IfcFlowMovingDevice instance, the pair corresponding to a AEX pump instance. The IFC Tag is an optional element.
ElementType	Denotes a particular type that indicates the object further. In particular, it holds the user defined type, if the enumeration of the 'PredefinedType' is set to USERDEFINED.	See remarks.		The AEX **type** element (below) enumerates two types of centrifugal pumps: Horizontal and Vertical. The AEX **customCentrifugalPump** element, which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **ElementType**.
PredefinedType	Enumerates a list of general types of pumps from which one must be selected. The selectable values are: CIRCULATOR ENDSUCTION SPLITCASE VERTICALINLINE VERTICALTURBINE USERDEFINED NOTDEFINED	**eqRot:centrifugalPumpType** and **eqRot:centrifugalPumpISO Classification** and **eqRot:casingMountingType** element and **eqRot:casingSplit** element and **casingType** element	Centrifugal pump type enumeration. The selectable values are: Horizontal Vertical Selectable Pump ISO classification types include: OH1, OH2, OH3, OH4, BB1, BB2, BB3, BB4, BB5, VS1, VS2, VS3, VS4, VS5, VS6, VS7, Other, Custom	AEX also provides additional information about casing types for example: **casingMountingType:** centerlilne, centerline-near,close coupled, foot, inline, inline-vert, other, separate mounting nplate, separate sole plate, sump, sump over plate, vert, unspecified **casingSplit**: axial, radial, other, unspecified **casingType**: barrel, diffuser, vert double casing, volute-multiple, volute-single, volute-staggered, other unspecified, custom

IfcPumpType: OwnerHistory				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
OwningUser	Direct reference to the end user who currently "owns" this object. Note that IFC includes the concept of ownership transfer from one user to another and therefore distinguishes between the Owning User and Creating User.	**obj:transaction /ctx:person**	Details about the person involved with the transaction.	**OwningUser** and **transaction person** are composed of additional elements. See the detailed breakdown of **OwningUser** containing a lower-level mapping to **transaction person** in the separate table "IfcPumpType: OwningUser." **LastModifyingUser** (below) and **OwningUser** are both mapped to AEX **transaction person** since AEX **transaction type** can be used to distinguish ownership and modification.
OwningApplication	Direct reference to the application which currently "Owns" this object on behalf of the owning user, who uses this application. Note that IFC includes the concept of ownership transfer from one application to another and therefore distinguishes between the Owning Application and Creating Application.	**obj:generatingSoftware**	Software information for the software that generated the object.	
State	Enumeration that defines the current access state of the object. The selectable values are: READWRITE READONLY LOCKED READWRITELOCKED READONLYLOCKED	**obj:transaction / type**	Enumeration that describes the type of transaction. The selectable values include: modifiedCopy readCopy viewEnabled viewProtected writeEnabled writeProtected	AEX **transaction** is a multiple occurrence structure wherein transactions against an object can be distinguished using **transaction type**. The AEX transaction type includes enumeration values for both IFC **state** and **ChangeAction** (see next element)

IfcPumpType: OwnerHistory *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
ChangeAction	Enumeration that defines the actions associated with changes made to the object. The selectable values are: NOCHANGE MODIFIED ADDED DELETED MODIFIEDADDED MODIFIEDDELETED	**obj:transaction / type**	Enumeration that describes the type of transaction. The selectable values are: approved archived checked created decrypted deleted encrypted issued modified modifiedCopy moved newRemark ownershipChanged readCopy renamed viewEnabled viewProtected writeEnabled writeProtected	
LastModifiedDate	Date and Time at which the last modification occurred.	**obj:transaction / dateTime**	Time and date of transaction.	
LastModifyingUser	User who carried out the last modification.	**obj:transaction / person**	Details about the person involved with the transaction.	**OwningUser** (above) and **LastModifyingUser** are both mapped to AEX **transaction person** since AEX **transaction type** can be used to distinguish ownership and modification.
LastModifyingApplication	Application used to carry out the last modification.	**generatingSoftware**	Software that generated the current values held in the object.	AEX **transaction** is a multiple occurrence structure wherein transactions can be distinguished using **transaction type**. The AEX **generatingSoftware** element is used to hold the information about the software used to carry out the last transaction, and objects can have an object history according to the recorded **transactions.**

IfcPumpType: OwnerHistory *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
CreationDate	Time and date of creation.	**obj:transaction / dateTime**	Time and date of transaction.	**LastModifiedDate** (above) and **CreationDate** are both mapped to AEX **transaction dateTime** since AEX **transaction** is a multiple occurrence structure wherein transactions can be distinguished using **transaction type**. In the case of **LastModifiedDate**, **transaction type** would be "modified." Similarly, for **CreationDate, transaction type** would be "created." For each **transaction type**, the appropriate **transaction dateTime** value would be entered.

IfcPumpType: OwningUser				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
ThePerson	The person who is related to the organization.	**ctx:person**	Details about the person involved with the transaction or an organization.	**ThePerson** is composed of additional elements. See the detailed breakdown of **ThePerson** containing a lower-level mapping to **ctx:person** in the separate table "IfcPumpType: ThePerson."
TheOrganization	The organization to which the person is related.	**ctx:organization**	Details about the organization to which the person belongs.	**TheOrganization** is composed of additional elements. See the detailed breakdown of **TheOrganization** containing a lower-level mapping to **organization** in the separate table "IfcPumpType: TheOrganization."
Roles	Roles played by the person within the context of an organization.	**ctx:person / obj:description**	Description of the object.	The AEX **person description** element could be used to describe the roles played by the person.

IfcPumpType: ThePerson				
Element(s)	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
Id	Identification of the person.	**obj:objectID** attribute or **obj:externalGUID** attribute	Unique identification for the person which is modeled as an object.	AEX **ctx:person** is an object and therefore can be uniquely identified like any other object.
FamilyName **GivenName** **MiddleNames**	FamilyName is the name by which the family identity of the person may be recognized. GivenName is the name by which a person is known within a family and by which he or she may be familiarly recognized. MiddleNames are additional names given to a person that enable their identification apart from others who may have the same or similar family and given names.	**ctx:person / obj:name**	Name of the person.	The AEX **person name** element is a single string and does not include a breakdown into separate entries for family, given, and middle names. **customPerson** element, which supports the declaration of any custom elements, any number of times, could be used to construct elements equivalent to **FamilyName, GivenName** and **MiddleName**.
PrefixTitles **SuffixTitles**	PrefixTitles is the word, or group of words, which specify the person's social and/or professional standing and appear before his/her names. SuffixTitles is the word, or group of words, which specify the person's social and/or professional standing and appear after his/her names.	See remarks.		The **PrefixTitle** and the **SuffixTitle** would be included with the single string of the AEX **transaction person customPerson** element, which supports the declaration of any custom elements, any number of times, could be used to construct elements equivalent to **PrefixTitles** and **SuffixTitles**.

IfcPumpType: ThePerson *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
Roles	Roles played by the person. The value is selected from an enumeration of 22 predefined roles or is user defined.	**ctx:person / obj:description**	Description of the object.	The AEX **transaction person description** element could be used to describe the roles played by the person. A person's role may show up in other context uses as well. For example, **contactPerson** for an **organization** is used in the context of a **project** definition and **transaction person** is used in the context of creating or editing objects. An **originatedBy** person and an **approvedBy** person are used in the context of creating and approving documents such as datasheets.
Addresses	Postal and telecommunication addresses of a person.	**ctx:person / location** corresponds to the IFC postal address; other AEX elements correspond to the information contained within an IFC telecommunication address. See remarks for additional information.	Location of the person's office.	**Addresses** and **person location** are composed of additional elements. Furthermore, **person / location** and **person / organization / facilityLocation** (below) share the same **Location** content model. See the detailed breakdown of **Addresses** containing a lower-level mapping to the AEX location content model in the separate table "IfcPumpType: Addresses."

IfcPumpType: TheOrganization				
Element	IFC Description	AEX Semantic Equivalent	AEX Description	Remark(s)
Id	Identification of the organization.	**obj:objectID** attribute or **obj:externalGUID** attribute	Unique identification for the organization which is modeled as an object	AEX **organization** is an object and therefore can be uniquely identified like any other object.
Name	The word, or group of words, by which the organization is referred to.	**ctx:organization / obj:name**	Name of the organization.	
Description	Text that relates the nature of the organization.	**ctx:organization / obj:description**	Description of the organization.	
Roles	Roles played by the organization. The value is selected from a list of 22 predefined roles or is user defined.	See remarks.		In AEX, the organization role is embedded within the context of other things. For example in a **project** an organization's role in the context of a **project** can be defined. Similarly for an item of equipment there are **fabricatingCompany**, **manufacturerCompany** and **supplierCompany** all of which share the O**rganization** content model. The AEX **organization customOrganization** element, which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **Roles**.
Addresses	Postal and telecommunication addresses of an organization.	**ctx:Location** corresponds to the IFC postal address; other AEX elements correspond to the information contained within an IFC telecommunication address. See remarks for additional information.	Location information for organization facilities.	**Addresses** and **Location** are composed of additional elements. Furthermore, **organization location** and **person location** (above) share the same location content model. See the detailed breakdown of **Addresses** containing a lower-level mapping to the AEX location content model in the separate table "IfcPumpType: Addresses."

IfcPumpType: Addresses				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
Purpose	Enumeration that identifies the logical location of the address. The selectable values are: OFFICE SITE HOME DISTRIBUTIONPOINT USERDEFINED	See remarks.		In the AEX model **Location** may be in a particular context, e.g., **siteFacility Location** or for example, refer to a **person location** or an **organization location** where each of these has a description field which can be used to elaborate **Purpose**. The AEX **customLocation** element**,** which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **Purpose**.
Description	Text that relates to the nature of the address.	**ctx:location / obj:description**	Description of the location.	
UserDefinedPurpose	Allows for the specification of a user specific purpose of the address beyond the enumeration values provided by Purpose. When a value is provided for UserDefinedPurpose, in parallel Purpose shall have enumeration value USERDEFINED.	See remarks.		The AEX **customLocation**, which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **UserDefinedPurpose**.
IfcPostalAddress	The address for delivery of paper based mail.	**ctx:location**		**IfcPostalAddress** and the AEX **location** content model used for **person location** as well as **siteFacilityLocation** are composed of additional elements. See the detailed breakdown of **IfcPostalAddress** containing a lower-level mapping to the AEX location content model in the separate table "IfcPumpType: IfcPostalAddress."

IfcPumpType: Addresses *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
IfcTelecomAddress	Address to which telephone, electronic mail and other forms of telecommunications should be addressed.	Various AEX elements correspond to the IfcTelecomAddress content model. See remarks for additional information.		**IfcTelecomAddress** is composed of additional elements, many of which correspond to AEX elements used to markup similar information. See the detailed breakdown of **IfcTelecomAddress** containing a lower-level mapping to these AEX elements in the separate table "IfcPumpType: IfcTelecomAddress."

IfcPumpType: IfcPostalAddress				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
Purpose	Enumeration that identifies the logical location of the address. The selectable values are: OFFICE SITE HOME DISTRIBUTIONPOINT USERDEFINED	See remarks.		In the AEX model, **Location** may be in a particular context, e.g., **siteFacility Location** or for example, refer to a **person location** or an **organization location** where each of these has a description field, which can be used to elaborate **Purpose**. The AEX **customLocation** element**,** which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **Purpose**.
Description	Text that relates to the nature of the address.	**location description**	Description of the location.	

IfcPumpType: IfcPostalAddress *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
UserDefinedPurpose	Allows for the specification of a user specific purpose of the address beyond the enumeration values provided by Purpose. When a value is provided for UserDefinedPurpose, in parallel Purpose shall have enumeration value USERDEFINED.	See remarks.		In the AEX model, **Location** may be in a particular context, e.g., **siteFacility Location** or for example, refer to a **person location** or an **organization location** where each of these has a description field that can be used to elaborate **Purpose**. The AEX **customLocation** element**,** which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **UserDefinedPurpose**.
InternalLocation	An organization defined address for internal mail delivery.	**ctx:location / mailStopNumber**	The identification number associated with a physical location where mail is delivered.	
AddressLines	The postal address.	**ctx:location / addressLine**	An occurrence of a generally accepted formatted line of data such that when combined with the entire address structure there is sufficient information for delivery or location purposes.	
PostalBox	An address that is implied by an identifiable mail drop.	**ctx:location / mailBoxNumber**	The identity of a physical box within a Mail delivery facility to accept and receive mail.	
Town	The name of a town.	**ctx:location / city**	The name of the city or municipality.	

IfcPumpType: IfcPostalAddress *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
Region	The name of a region.	**ctx:location / region**	The name of the geographic region that contains the location.	
PostalCode	The code that is used by the country's postal service.	**ctx:location / postalCode**	The code assigned by local postal authorities to delivery addresses.	
Country	The name of a country.	**ctx:location / country**	Name of the country.	

IfcPumpType: IfcTelecomAddress				
Element	IFC Description	AEX Semantic Equivalent	AEX Description	Remark(s)
Purpose	Enumeration that identifies the logical location of the address. The selectable values are: OFFICE SITE HOME DISTRIBUTIONPOINT USERDEFINED	See remarks.		In the AEX model **Location** may be in a particular context, e.g., **siteFacility Location** or for example, refer to a **person location** or an **organization location** where each of these has a description field that can be used to elaborate **Purpose**. The AEX **customLocation** element**,** which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **Purpose**.
Description	Text that relates to the nature of the address.	**ctx:location/description**	Description of the location.	
UserDefinedPurpose	Allows for the specification of a user specific purpose of the address beyond the enumeration values provided by Purpose. When a value is provided for UserDefinedPurpose, in parallel Purpose shall have enumeration value USERDEFINED.	See remarks.		In the AEX model, **Location** may be in a particular context, e.g., **siteFacility Location** or for example, refer to a **person location** or an **organization location** where each of these has a description field that can be used to elaborate **Purpose**. The AEX **customLocation** element**,** which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **UserDefinedPurpose**.
TelephoneNumbers	The list of telephone numbers at which telephone messages may be received.	**ctx:person / phone**	Telephone number.	The **person phone** element is composed of additional elements, one of which is called **phoneType**. **phoneType** enumerates specific entries for differentiating phone numbers: business fax home mobile

IfcPumpType: IfcTelecomAddress *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
FacsimileNumbers	The list of fax numbers at which fax messages may be received.	**ctx:person / phone**	Telephone number.	The AEX **transaction person phone** element is composed of additional elements, one of which is called **type**. **type** enumerates specific entries for differentiating phone numbers: business fax home mobile The mapping of IFC **FacsimileNumbers** to AEX **transaction person phone** would include setting AEX **transaction person phone type** to "fax."
PagerNumber	The pager number at which paging messages may be received.	**ctx:person / phone**	Telephone number.	The **phoneType** of mobile could be used, or alternatively **phoneType** can be extended to have 'pager' as a valid choice.
ElectronicMailAddresses	The list of Email addresses at which Email messages may be received.	**ctx:person / email**	Email address.	
WWWHomePageURL	The world wide web address at which the preliminary page of information for the person or organization can be located.	**ctx:person/ organization / webSiteURL**	URL to the organization web site.	

Pset_FlowMovingDevicePump				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
ImpellerDiameter	Diameter of pump impeller - used to scale performance of geometrically similar pumps.	**eqRot:impeller ratedDiameter**	impeller rated diameter.	
BaseType	Enumeration that identifies general types of pump bases. The selectable values are: FRAME BASE NONE OTHER NOTKNOWN UNSET	**eqRot:baseplate type** and **eqRot:baseplateMounting-Type**	Enumeration that identifies the baseplate type. The selectable values are: open beam top decking – full top decking – partial other unspecified baseplateMountingType choices include: bolted on steelwork free-standing free-standing – 3 pt free-standing – 4 pt free-standing – stilt grouted grouted-epoxy grouted-modified regular grouted-regular other unspecified	
DriveConnectionType	Enumeration that identifies the way the pump drive mechanism is connected to the pump. The selectable values are: DIRECTDRIVE BELTDRIVE COUPLING OTHER NOTKNOWN UNSET	**eqRot:driver / eqElec:motor / motorConnectionType**	Enumeration that identifies the type of connection between a motor driver and the shaft. The selectable values are: belt drive coupling direct drive other unspecified	

Pset_PumpTypeCommon				
Element	IFC Description	AEX Semantic Equivalent	AEX Description	Remark(s)
FlowRateRange	Allowable range of volume of fluid being pumped against the resistance specified.	eqRot:performance / operatingRegion / minAllowableFlowVolume and eqRot:performance / operatingRegion / maxAllowableFlowVolume	Minimum allowed volumetric flow rate and maximum allowed volumetric flow rate, respectively.	
FlowResistanceRange	Allowable range of frictional resistance against which the fluid is being pumped	See remarks.		Unclear from definition what FlowResistanceRange means in this case. AEX provides a number of parameters supporting this idea including specification of min and max suction and discharge pressures, min and max pressure difference or head. The AEX **performance operatingRegion customOperatingRegion** element, which supports the declaration of any custom elements, any number of times, could be used to construct an element equivalent to **FlowResistanceRange**.
ConnectionSize	The connection size to and from the pump	eq:equipmentConnection / endDescription / flangedConnection / nominalDiameter and eq:equipmentConnection / endDescription / threadedConnection / nominalDiameter	Nominal diameter for flanged end connection and nominal diameter for threaded connection, respectively.	
CasingMaterial	Material from which the pump casing is constructed	eqRot:casing / mtrl:constructionMaterial	Casing construction material	
ImpellerMaterial	Material from which the impeller is constructed.	eqRot:impeller / mtrl:constructionMaterial	Impeller construction material.	
ImpellerSealMaterial	Material from which the impeller shaft seal is constructed.	eqRot:seal / detail / mtrl:constructionMaterial	Shaft seal construction material.	

Pset_PumpTypeCommon *continued*				
Element	**IFC Description**	**AEX Semantic Equivalent**	**AEX Description**	**Remark(s)**
TemperatureRange	Allowable operational range of the fluid temperature.	**eq:requirement / mechanicalDesign / minDesignT** and **eq:requirement / mechanicalDesign / maxDesignT**.		
NetPositiveSuctionHead	Minimum liquid pressure at the pump inlet to prevent cavitation.	**eqRot:operatingCondition / condition / availableNPSH** and **eqRot:operatingCondition / condition / requiredNPSH**.	Available net positive suction head in height of fluid and required net positive suction head in height of fluid, respectively.	The IFC description for **NetPositiveSuctionHead** does not distinguish between "available" and "required."
NominalRotationSpeed	Pump rotational speed under nominal conditions.	**eqRot:performance / ratedPumpSpeed**	Rated pump speed.	

Pset_PumpPHistory				
Element	IFC Description	AEX Semantic Equivalent	AEX Description	Remark(s)
MechanicalEfficiency	The pump's operational mechanical efficiency, expressed as a time series (using IfcTimeSeries).	See remarks.		The AEX **performance customPerformance** element could be used to construct an element equivalent to **MechanicalEfficiency**. As in the following elements, however, the notion of a time series of data would still be lacking.
OverallEfficiency	The pump and motor overall operational efficiency, expressed as a time series.	**eqRot:performance / efficiency**	Efficiency.	The AEX element lacks the notion of a time series.
PressureRise	The developed pressure, expressed as a time series.	**eqRot:operatingCondition / condition / pressureDifference**	Pressure difference.	The AEX element lacks the notion of a time series.
RotationSpeed	Pump rotational speed, expressed as a time series.	See remarks.		The AEX **performance customPerformance** element could be used to construct an element equivalent to **RotationSpeed**, but it would still lack the notion of a time series.
Flowrate	The actual operational fluid flowrate, expressed as a time series.	**eqRot:operatingCondition / condition / volumetricFlow**	Volumetric flow.	The AEX element lacks the notion of a time series.
Power	The actual power consumption of the pump, expressed as a time series.	See remarks.		The AEX **performance ratedPower** element relates to the rated rather than actual power of the pump. However, the AEX **performance customPerformance** element could be used to construct an element equivalent to **Power,** albeit lacking the notion of a time series.

3.3 Test Artifact

Example XML instance files for the following data were created to give a direct comparison of the IFC and AEX approaches to modeling pump information. Section 3.4 contains the IFC example instance and section 3.5 contains the AEX example instance. In this example, an object creation transaction is recorded, undertaken by a particular person in a specific organization using a designated software package. There are several pump properties which have been included, such as min and max volumetric flow rates, impeller diameter, rotation speed, materials of construction, and so forth. Manufacturer identification has been masked (Xxxxxx) and GlobalIds have been constructed arbitrarily rather than algorithmically by a GUID/UUID generator[32].

Manufacturer	Xxxxxx
Type of Pump	endsuction
Type of Pump Base	not specified, i.e., unset
Type of Pump Drive Connection	not specified, i.e., unset
Casing Material	Cast iron
Impeller Material	Cast iron
Impeller Seal Material	Rotating face – carbon, Stationary face – Ni-Resist, Elastomer – Buna-N, Hardware – stainless steel
Impeller Diameter	12.09 in
Minimum Flow Rate	0 gal/min
Maximum Flow Rate	560 gal/min
Maximum Operatonal Fluid Temperature	180 F
Nominal Rotation Speed	1800 rpm
Suction End Connection Size	4.0 in
Discharge End Connection Size	3.0 in

3.4 Sample IfcPumpType Document

```
<?xml version = "1.0" encoding = "UTF-8"?>
<ifc:uos xmlns:ifc = "http://www.iai-international.org/ifcXML2/RC1/ifcXML_for_IFC2x2" xmlns:xsi =
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation = " http://www.iai-
international.org/iai_international/Technical_Documents/files/ifcXML_for_IFC2x2.xsd" id = "">

    <ifc:IfcPumpType id="Xxxxxx">
        <GlobalId>Xxxxxx_____Sample001</GlobalId>
        <OwnerHistory>
            <ifc:IfcOwnerHistory  href = "oh1" xsi:nil = "true"/>
        </OwnerHistory>

        <HasPropertySets>
            <ifc:IfcPropertySet>
                <GlobalId>PFMDP_Xxxxxx_Sample001</GlobalId>
                <OwnerHistory>
                    <ifc:IfcOwnerHistory href = "oh1" xsi:nil = "true"/>
                </OwnerHistory>
                <Name>Pset_FlowMovingDevicePump</Name>
                <Description>Property set for a pump occurrence of a flow moving device.</Description>
                <HasProperties>
```

```
<ifc:IfcPropertySingleValue>
    <Name>ImpellerDiameter</Name>
    <Description>
    Diameter of pump impeller - used to scale performance of geometrically similar pumps
    </Description>
    <NominalValue>
        <ifc:IfcPositiveLengthMeasure>12.09</ifc:IfcPositiveLengthMeasure>
    </NominalValue>
    <Unit>
        <ifc:IfcConversionBasedUnit href = "in" xsi:nil = "true"/>
    </Unit>
</ifc:IfcPropertySingleValue>

<ifc:IfcPropertyEnumeratedValue>
    <Name>BaseType</Name>
    <Description>Defines general types of pump bases.</Description>
    <EnumerationValues>
        <ifc:IfcText>UNSET</ifc:IfcText>
    </EnumerationValues>
</ifc:IfcPropertyEnumeratedValue>

<ifc:IfcPropertyEnumeratedValue>
    <Name>DriveConnectionType</Name>
    <Description>The way the pump drive mechanism is connected to the pump.</Description>
    <EnumerationValues>
        <ifc:IfcText>UNSET</ifc:IfcText>
    </EnumerationValues>
</ifc:IfcPropertyEnumeratedValue>

</HasProperties>
</ifc:IfcPropertySet>

<ifc:IfcPropertySet>
    <GlobalId>PPTC_Xxxxxx__Sample001</GlobalId>
    <OwnerHistory>
        <ifc:IfcOwnerHistory href = "oh1" xsi:nil = "true"/>
    </OwnerHistory>
    <Name>Pset_PumpTypeCommon</Name>
    <Description>Property set for the common properties of a pump.</Description>
    <HasProperties>

        <ifc:IfcPropertyBoundedValue>
            <Name>FlowRateRange</Name>
            <Description>
            Allowable range of volume of fluid being pumped against the resistance specified.
            </Description>
            <UpperBoundValue>
                <ifc:IfcMassFlowRateMeasure>560</ifc:IfcMassFlowRateMeasure>
            </UpperBoundValue>
            <LowerBoundValue>
                <ifc:IfcMassFlowRateMeasure>0</ifc:IfcMassFlowRateMeasure>
            </LowerBoundValue>
            <Unit>
                <ifc:IfcConversionBasedUnit href = "gpm" xsi:nil = "true"/>
            </Unit>
        </ifc:IfcPropertyBoundedValue>

        <ifc:IfcPropertySingleValue>
            <Name>ConnectionSize</Name>
            <Description>The connection size to the suction end of the pump.</Description>
```

```
<NominalValue>
    <ifc:IfcPositiveLengthMeasure>4.0</ifc:IfcPositiveLengthMeasure>
</NominalValue>
<Unit>
    <ifc:IfcConversionBasedUnit href = "in" xsi:nil = "true"/>
</Unit>
</ifc:IfcPropertySingleValue>


<ifc:IfcPropertySingleValue>
    <Name>ConnectionSize</Name>
    <Description>The connection size to the discharge end of the pump.</Description>
    <NominalValue>
        <ifc:IfcPositiveLengthMeasure>3.0</ifc:IfcPositiveLengthMeasure>
    </NominalValue>
    <Unit>
        <ifc:IfcConversionBasedUnit href = "in" xsi:nil = "true"/>
    </Unit>
</ifc:IfcPropertySingleValue>


<ifc:IfcPropertyReferenceValue>
    <Name>CasingMaterial</Name>
    <Description>Material from which the casing of the pump is constructed.</Description>
    <PropertyReference>
        <ifc:IfcMaterial>
            <Name>Cast Iron</Name>
        </ifc:IfcMaterial>
    </PropertyReference>
</ifc:IfcPropertyReferenceValue>


<ifc:IfcPropertyReferenceValue>
    <Name>ImpellerMaterial</Name>
    <Description>Material from which the impeller of the pump is constructed.</Description>
    <PropertyReference>
        <ifc:IfcMaterial>
            <Name>Cast Iron</Name>
        </ifc:IfcMaterial>
    </PropertyReference>
</ifc:IfcPropertyReferenceValue>


<ifc:IfcPropertyReferenceValue>
    <Name>ImpellerSealMaterial</Name>
    <Description>Material from which the impeller shaft seal of the pump is constructed.</Description>
    <PropertyReference>
        <ifc:IfcMaterial>
            <Name>
            Rotating face - carbon, Stationary face - Ni-Resist, Elastomer - Buna-N, Hardware - stainless steel
            </Name>
        </ifc:IfcMaterial>
    </PropertyReference>
</ifc:IfcPropertyReferenceValue>


<ifc:IfcPropertyBoundedValue>
    <Name>TemperatureRange</Name>
    <Description>Allowable operational range of the fluid temperature.</Description>
    <UpperBoundValue>
        <ifc:IfcThermodynamicTemperatureMeasure>180</ifc:IfcThermodynamicTemperatureMeasure>
    </UpperBoundValue>
    <Unit>
        <ifc:IfcConversionBasedUnit href = "F" xsi:nil = "true"/>
    </Unit>
</ifc:IfcPropertyBoundedValue>
```

```xml
            <ifc:IfcPropertySingleValue>
                <Name>NominalRotationSpeed</Name>
                <Description>Pump rotational speed under nominal conditions.</Description>
                <NominalValue>
                    <ifc:IfcRotationalFrequencyMeasure>1800</ifc:IfcRotationalFrequencyMeasure>
                </NominalValue>
                <Unit>
                    <ifc:IfcContextDependentUnit href = "rpm" xsi:nil = "true"/>
                </Unit>
            </ifc:IfcPropertySingleValue>

        </HasProperties>
    </ifc:IfcPropertySet>
</HasPropertySets>

<PredefinedType>endsuction</PredefinedType>
</ifc:IfcPumpType>

<ifc:IfcRelAssignsToActor>
    <GlobalId>RAA_Xxxxxx___Sample001</GlobalId>
    <OwnerHistory>
        <ifc:IfcOwnerHistory href = "oh1" xsi:nil = "true"/>
    </OwnerHistory>
    <Name>Manufacturer</Name>
    <Description>Pump Manufacturer</Description>
    <RelatedObjects>
        <ifc:IfcFlowMovingDevice href="Xxxxxx" xsi:nil = "true"/>
    </RelatedObjects>
    <RelatedObjectsType>product</RelatedObjectsType>
    <RelatingActor>
        <ifc:IfcActor>
            <GlobalId>RA__Xxxxxx___Sample001</GlobalId>
            <OwnerHistory>
                <ifc:IfcOwnerHistory href = "oh1" xsi:nil = "true"/>
            </OwnerHistory>
            <TheActor>
                <ifc:IfcOrganization>
                    <Id>Xxxxxx</Id>
                    <Name>Xxxxxx</Name>
                    <Description>Xxxxxx pump manufacturer</Description>
                </ifc:IfcOrganization>
            </TheActor>
        </ifc:IfcActor>
    </RelatingActor>
    <ActingRole>
        <ifc:IfcActorRole>
            <Role>manufacturer</Role>
            <Description>assembly or fabrication</Description>
        </ifc:IfcActorRole>
    </ActingRole>
</ifc:IfcRelAssignsToActor>

<ifc:IfcSIUnit id = "m">
    <UnitType>lengthunit</UnitType>
    <Name>metre</Name>
</ifc:IfcSIUnit>

<ifc:IfcSIUnit id = "s">
    <UnitType>timeunit</UnitType>
    <Name>second</Name>
```

```
        </ifc:IfcSIUnit>

        <ifc:IfcSIUnit id = "litre">
            <UnitType>volumeunit</UnitType>
            <Prefix>milli</Prefix>
            <Name>cubic_metre</Name>
        </ifc:IfcSIUnit>

        <ifc:IfcSIUnit id = "K">
            <UnitType>thermodynamictemperatureunit</UnitType>
            <Name>kelvin</Name>
        </ifc:IfcSIUnit>

        <ifc:IfcConversionBasedUnit id = "min">
            <Dimensions>
                <ifc:IfcDimensionalExponents>
                    <LengthExponent>0</LengthExponent>
                    <MassExponent>0</MassExponent>
                    <TimeExponent>1</TimeExponent>
                    <ElectricCurrentExponent>0</ElectricCurrentExponent>
                    <ThermodynamicTemperatureExponent>0</ThermodynamicTemperatureExponent>
                    <AmountOfSubstanceExponent>0</AmountOfSubstanceExponent>
                    <LuminousIntensityExponent>0</LuminousIntensityExponent>
                </ifc:IfcDimensionalExponents>
            </Dimensions>
            <UnitType>timeunit</UnitType>
            <Name>minute</Name>
            <ConversionFactor>
                <ifc:IfcMeasureWithUnit>
                    <ValueComponent>
                        <ifc:IfcInteger>60</ifc:IfcInteger>
                    </ValueComponent>
                    <UnitComponent>
                        <ifc:IfcSIUnit href = "s" xsi:nil = "true"/>
                    </UnitComponent>
                </ifc:IfcMeasureWithUnit>
            </ConversionFactor>
        </ifc:IfcConversionBasedUnit>

        <ifc:IfcConversionBasedUnit id = "gallon">
            <Dimensions>
                <ifc:IfcDimensionalExponents>
                    <LengthExponent>3</LengthExponent>
                    <MassExponent>0</MassExponent>
                    <TimeExponent>0</TimeExponent>
                    <ElectricCurrentExponent>0</ElectricCurrentExponent>
                    <ThermodynamicTemperatureExponent>0</ThermodynamicTemperatureExponent>
                    <AmountOfSubstanceExponent>0</AmountOfSubstanceExponent>
                    <LuminousIntensityExponent>0</LuminousIntensityExponent>
                </ifc:IfcDimensionalExponents>
            </Dimensions>
            <UnitType>volumeunit</UnitType>
            <Name>gallon</Name>
            <ConversionFactor>
                <ifc:IfcMeasureWithUnit>
                    <ValueComponent>
                        <ifc:IfcReal>3.785412</ifc:IfcReal>
                    </ValueComponent>
                    <UnitComponent>
                        <ifc:IfcSIUnit href = "litre" xsi:nil = "true"/>
                    </UnitComponent>
```

```xml
            </ifc:IfcMeasureWithUnit>
        </ConversionFactor>
</ifc:IfcConversionBasedUnit>

<ifc:IfcDerivedUnit id = "gpm">
    <Elements>
        <ifc:IfcDerivedUnitElement>
            <Unit>
                <ifc:IfcConversionBasedUnit href = "min" xsi:nil = "true"/>
            </Unit>
            <Exponent>-1</Exponent>
        </ifc:IfcDerivedUnitElement>
        <ifc:IfcDerivedUnitElement>
            <Unit>
                <ifc:IfcConversionBasedUnit href = "gallon" xsi:nil = "true"/>
            </Unit>
            <Exponent>1</Exponent>
        </ifc:IfcDerivedUnitElement>
    </Elements>
    <UnitType>volumetricflowrateunit</UnitType>
</ifc:IfcDerivedUnit>

<ifc:IfcConversionBasedUnit id = "in">
    <Dimensions>
        <ifc:IfcDimensionalExponents>
            <LengthExponent>1</LengthExponent>
            <MassExponent>0</MassExponent>
            <TimeExponent>0</TimeExponent>
            <ElectricCurrentExponent>0</ElectricCurrentExponent>
            <ThermodynamicTemperatureExponent>0</ThermodynamicTemperatureExponent>
            <AmountOfSubstanceExponent>0</AmountOfSubstanceExponent>
            <LuminousIntensityExponent>0</LuminousIntensityExponent>
        </ifc:IfcDimensionalExponents>
    </Dimensions>
    <UnitType>lengthunit</UnitType>
    <Name>inch</Name>
    <ConversionFactor>
        <ifc:IfcMeasureWithUnit>
            <ValueComponent>
                <ifc:IfcReal>0.0254</ifc:IfcReal>
            </ValueComponent>
            <UnitComponent>
                <ifc:IfcSIUnit href = "m" xsi:nil = "true"/>
            </UnitComponent>
        </ifc:IfcMeasureWithUnit>
    </ConversionFactor>
</ifc:IfcConversionBasedUnit>

<ifc:IfcConversionBasedUnit id = "F">
    <Dimensions>
        <ifc:IfcDimensionalExponents>
            <LengthExponent>0</LengthExponent>
            <MassExponent>0</MassExponent>
            <TimeExponent>0</TimeExponent>
            <ElectricCurrentExponent>0</ElectricCurrentExponent>
            <ThermodynamicTemperatureExponent>1</ThermodynamicTemperatureExponent>
            <AmountOfSubstanceExponent>0</AmountOfSubstanceExponent>
            <LuminousIntensityExponent>0</LuminousIntensityExponent>
        </ifc:IfcDimensionalExponents>
    </Dimensions>
    <UnitType>thermodynamictemperatureunit</UnitType>
```

```xml
            <Name>degrees Fahrenheit</Name>
            <ConversionFactor>
                <ifc:IfcMeasureWithUnit>
                    <ValueComponent>
                        <ifc:IfcText>1.8(K-273)+32</ifc:IfcText>
                    </ValueComponent>
                    <UnitComponent>
                        <ifc:IfcSIUnit href = "K" xsi:nil = "true"/>
                    </UnitComponent>
                </ifc:IfcMeasureWithUnit>
            </ConversionFactor>
</ifc:IfcConversionBasedUnit>

<ifc:IfcContextDependentUnit id = "rpm">
        <UnitType>frequencyunit</UnitType>
        <Name>rpm</Name>
</ifc:IfcContextDependentUnit>

<ifc:IfcPerson id = "p1">
        <Id>efb</Id>
        <FamilyName>Begley</FamilyName>
        <GivenName>Ed</GivenName>
        <PrefixTitles>
            <ifc:IfcLabel>Mr</ifc:IfcLabel>
        </PrefixTitles>
        <Roles>
            <ifc:IfcActorRole>
                <Role>userdefined</Role>
                <UserDefinedRole>Principal Investigator</UserDefinedRole>
                <Description>Project Principal Investigator</Description>
            </ifc:IfcActorRole>
        </Roles>
        <Addresses>
            <ifc:IfcPostalAddress href = "pa1" xsi:nil = "true"/>
            <ifc:IfcTelecomAddress href = "ta1" xsi:nil = "true"/>
        </Addresses>
</ifc:IfcPerson>

<ifc:IfcOrganization id = "o1">
        <Name>NIST</Name>
        <Description>National Institute of Standards and Technology</Description>
        <Roles>
            <ifc:IfcActorRole>
                <Role>userdefined</Role>
                <UserDefinedRole>Standards research and development</UserDefinedRole>
            </ifc:IfcActorRole>
        </Roles>
        <Addresses>
            <ifc:IfcPostalAddress href = "pa1" xsi:nil = "true"/>
        </Addresses>
</ifc:IfcOrganization>

<ifc:IfcPersonAndOrganization id = "pao1">
        <ThePerson>
            <ifc:IfcPerson href = "p1" xsi:nil = "true"/>
        </ThePerson>
        <TheOrganization>
            <ifc:IfcOrganization href = "o1" xsi:nil = "true"/>
        </TheOrganization>
</ifc:IfcPersonAndOrganization>
```

```xml
<ifc:IfcPostalAddress id = "pa1">
    <AddressLines>
        <ifc:IfcLabel>100 Bureau Drive Stop 8630</ifc:IfcLabel>
    </AddressLines>
    <Town>Gaithersburg</Town>
    <Region>Maryland</Region>
    <PostalCode>20899</PostalCode>
    <Country>United States</Country>
</ifc:IfcPostalAddress>

<ifc:IfcTelecomAddress id = "ta1">
    <TelephoneNumbers>
        <ifc:IfcLabel>(301) 975-6118</ifc:IfcLabel>
    </TelephoneNumbers>
    <FacsimileNumbers>
        <ifc:IfcLabel>(301) 975-5433</ifc:IfcLabel>
    </FacsimileNumbers>
    <ElectronicMailAddresses>
        <ifc:IfcLabel>begley@nist.gov</ifc:IfcLabel>
    </ElectronicMailAddresses>
    <WWWHomePageURL>http://www.nist.gov</WWWHomePageURL>
</ifc:IfcTelecomAddress>

<ifc:IfcOwnerHistory id = "oh1">
    <OwningUser>
        <ifc:IfcPersonAndOrganization href = "pao1" xsi:nil = "true"/>
    </OwningUser>
    <OwningApplication>
        <ifc:IfcApplication>
            <ApplicationDeveloper>
                <ifc:IfcOrganization href = "o1" xsi:nil = "true"/>
            </ApplicationDeveloper>
            <Version>v2004.06.09</Version>
            <ApplicationFullName>SIMA Product Data Standards for HVAC_R Systems</ApplicationFullName>
            <ApplicationIdentifier>SIMA HVAC_R</ApplicationIdentifier>
        </ifc:IfcApplication>
    </OwningApplication>
    <ChangeAction>added</ChangeAction>
    <CreationDate>1086764760</CreationDate>
</ifc:IfcOwnerHistory>

</ifc:uos>
```

3.5 Sample AEX Centrifugal Pump Document

```xml
<?xml version="1.0" encoding="UTF-8"?>
<centrifugalPump xmlns="http://www.cfixml.org/eqRot" xmlns:ctx="http://www.cfixml.org/ctx"
xmlns:dx="http://www.cfixml.org/dx" xmlns:etl="http://www.cfixml.org/etl" xmlns:eq="http://www.cfixml.org/eq"
xmlns:eqElec="http://www.cfixml.org/eqElec" xmlns:eqRot="http://www.cfixml.org/eqRot"
xmlns:ext="http://www.cfixml.org/ext" xmlns:mtrl="http://www.cfixml.org/mtrl" xmlns:obj="http://www.cfixml.org/obj"
xmlns:pq="http://www.cfixml.org/pq" xmlns:proj="http://www.cfixml.org/proj" xmlns:site="http://www.cfixml.org/site"
xmlns:thmo="http://www.cfixml.org/thmo" xmlns:uo="http://www.cfixml.org/uo"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.cfixml.org/eqRot
../schema/eqRot.xsd" objectID="eqRot.CentrifugalPump.Xxxxxx" unitSet="USEngineering"
externalGUID="Xxxxxx____Sample001">
    <obj:generatingSoftware>
        <obj:name comment="SIMA Product Data Standards for HVAC_R Systems">SIMA HVAC_R</obj:name>
        <obj:version>v2004.06.09</obj:version>
    </obj:generatingSoftware>
    <obj:transaction>
        <obj:dateTime>2001-12-17T09:30:47-05:00</obj:dateTime>
        <ctx:person objectID="ctx.Person.EFBegley" unitSet="USEngineering">
            <obj:name>Mr. Ed. Begley</obj:name>
            <ctx:shortID>efb</ctx:shortID>
            <ctx:email>begley@nist.gov</ctx:email>
            <ctx:phone>
                <ctx:fullNumber>(301) 975-6118</ctx:fullNumber>
                <ctx:phoneType>business</ctx:phoneType>
            </ctx:phone>
            <ctx:phone>
                <ctx:fullNumber>(301) 975-5433</ctx:fullNumber>
                <ctx:phoneType>fax</ctx:phoneType>
            </ctx:phone>
            <ctx:organization objectID="ctx.Organization.NIST">
                <obj:name>NIST</obj:name>
                <obj:description>National Institute of Standards and Technology</obj:description>
                <ctx:facilityLocation objectID="ctx.FacilityLocation.NISTGaithersburg">
                    <obj:customBaseObject/>
                    <ctx:addressLine>100 Bureau Drive Stop 8630</ctx:addressLine>
                    <ctx:mailStopNumber>8630</ctx:mailStopNumber>
                    <ctx:city>Gaithersburg</ctx:city>
                    <ctx:stateProvince>MD</ctx:stateProvince>
                    <ctx:country>United States</ctx:country>
                    <ctx:postalCode>20899</ctx:postalCode>
                </ctx:facilityLocation>
                <ctx:webSiteURL>http://www.nist.gov</ctx:webSiteURL>
                <ctx:customOrganization>
                    <role>Standards research and development</role>
                </ctx:customOrganization>
            </ctx:organization>
            <ctx:customPerson>
                <role>Principal Investigator</role>
                <roleDescription>Project Principal Investigator</roleDescription>
            </ctx:customPerson>
        </ctx:person>
        <obj:transactionType>created</obj:transactionType>
    </obj:transaction>
    <eq:id>
        <ctx:manufacturerCompany objectID="ctx.manufacturerCompany.Xxxxxx"
externalGUID="RA__Xxxxxx__Sample001">
            <obj:name>Xxxxxx Manufacturer</obj:name>
            <obj:description>Xxxxxx pump manufacturer</obj:description>
        </ctx:manufacturerCompany>
```

```xml
    </eq:id>
<eq:equipmentConnection objectID="eq.equipmentConnection.Suction.Sample001">
    <obj:name>Suction</obj:name>
    <eq:endDescription>
        <eq:flangedConnection objectID="eq.Flange.Suction">
            <eq:nominalDiameter symbol="in">4</eq:nominalDiameter>
        </eq:flangedConnection>
    </eq:endDescription>
    <eq:connectionType>suction</eq:connectionType>
</eq:equipmentConnection>
<eq:equipmentConnection objectID="eq.equipmentConnection.Discharge.Sample001">
    <obj:name>Discharge</obj:name>
    <eq:endDescription>
        <eq:flangedConnection objectID="eq.Flange.Discharge">
            <eq:nominalDiameter symbol="in">3</eq:nominalDiameter>
        </eq:flangedConnection>
    </eq:endDescription>
    <eq:connectionType>discharge</eq:connectionType>
</eq:equipmentConnection>
<eq:requirement>
    <eq:mechanicalDesign>
        <eq:maxDesignT symbol="degF">180</eq:maxDesignT>
    </eq:mechanicalDesign>
</eq:requirement>
<baseplate objectID="eqRot.baseplate.Sample001">
    <baseplateType>unspecified</baseplateType>
</baseplate>
<eqRot:casing>
    <casingSplit>unspecified</casingSplit>
    <mtrl:constructionMaterial objectID="mtrl.Material.CastIron">
        <obj:name>CastIron</obj:name>
    </mtrl:constructionMaterial>
</eqRot:casing>
<eqRot:driver>
    <eqElec:motor objectID="eqElec.Motor.Sample001">
        <eqElec:motorConnectionType>unspecified</eqElec:motorConnectionType>
    </eqElec:motor>
</eqRot:driver>
<eqRot:impeller objectID="eqRot.Impeller.Sample001">
    <mtrl:constructionMaterialProperty objectID="mtrl.MaterialProperty.CastIronProperty">
        <mtrl:material objectID="mtrl.Material.CastIron" objectState="refLocal"/>
    </mtrl:constructionMaterialProperty>
    <eqRot:ratedDiameter symbol="in">12.09</eqRot:ratedDiameter>
</eqRot:impeller>
<eqRot:operatingCondition>
    <eqRot:condition>
        <eqRot:volumetricFlow symbol="US_gal/min">0</eqRot:volumetricFlow>
        <eq:operatingConditionType>min</eq:operatingConditionType>
    </eqRot:condition>
    <eqRot:condition>
        <eqRot:volumetricFlow symbol="US_gal/min">560</eqRot:volumetricFlow>
        <eq:operatingConditionType>max</eq:operatingConditionType>
    </eqRot:condition>
</eqRot:operatingCondition>
<eqRot:performance>
    <eqRot:operatingRegion>
        <eqRot:maxAllowableFlowVolume symbol="US_gal/min">560</eqRot:maxAllowableFlowVolume>
        <eqRot:minAllowableFlowVolume symbol="US_gal/min">120</eqRot:minAllowableFlowVolume>
    </eqRot:operatingRegion>
    <eqRot:ratedPumpSpeed symbol="rev/min">1800</eqRot:ratedPumpSpeed>
</eqRot:performance>
```

```
    <eqRot:sealOrPacking>
        <eqRot:mechanicalSeal objectID="eqRot.MechSeal.Sample001">
            <mtrl:constructionMaterialProperty objectID="mtrl.MaterialProperty.Seal001">
                <mtrl:material objectID="mtrl.Material.Seal001">
                    <obj:name>Impeller Seal Material Composite</obj:name>
                    <obj:description>Rotating face - carbon, Stationary face - Ni-Resist, Elastomer - Buna-N, Hardware -
stainless steel</obj:description>
                </mtrl:material>
            </mtrl:constructionMaterialProperty>
        </eqRot:mechanicalSeal>
    </eqRot:sealOrPacking>
    <customRotatingEquipment>
        <ifcPredefinedPumpType>endsuction</ifcPredefinedPumpType>
    </customRotatingEquipment>
</centrifugalPump>
```

4. DISCUSSION

For the example worked in this study, the AEX instance document is roughly one-third the size of the ifcXML instance document for the same content. The IFC instance file contained 386 lines of XML versus the AEX instance file, which contained 125 lines of XML. This is partly due to the structure of the ifcXML that is inherited from the underlying IFC EXPRESS model and partly due to the fact that the ifcXML document contains data definitions (e.g., schemata as data) inside the same file as the data, whereas the AEX document is based on an optimized XML schema and contains only the instance data. Notably, the ifcXML document includes explicit, voluminous definitions of the units of measure and of the property data sets used in the document. The AEX schemas have a much larger set of tags and enumeration values that are explicitly defined in the schema definition files, and therefore do not have to be defined in the instance documents. In particular, units of measure are enumerated in the AEX schema. This disparity in document size is likely always to occur because of the derivation of ifcXML from the IFC EXPRESS model. As discussed in section "1.3 ifcXML," the IAI is addressing this issue.

It is worth examining the different approaches to extensibility taken by the IAI and FIATECH efforts. In the IAI approach, the XML schema is fixed, but designed to allow user-defined extensions to the schema to occur as data in the exchange file along with population data that are consistent with the fixed schema and the user-defined extensions. This approach is analogous to extending a fixed language by adding a macro language. Just as introducing a macro language requires the addition of a macro processor to deal with the macro instances, taking the IAI approach requires the addition of a post-processor that can deal with the in-line extensions to the schema. Validating XML documents containing such extensions requires extra computation after parsing is complete to determine the schema against which validation will take place.

This approach is a direct result of the IFC (and before it, STEP) heritage of ifcXML. EXPRESS schemas must be preprocessed into code stubs to which must be added application code that deals with the semantics of the related data structures. Combined with the facts that EXPRESS is not expressed in EXPRESS and the syntax of EXPRESS structures is different from the exchange file structures, this makes it difficult to accommodate user-defined extensions as add-on EXPRESS schemata.

The AEX approach, on the other hand, is unencumbered by any such legacy considerations. It allows user-defined schema extensions to be added directly to the baseline AEX schema. Assuming the augmented schema is accessible to the post-processing software, then substantial validation can be done by the normal XML parser. Of course, the receiving application must still understand the extended semantics.

The AEX schema definitions also allow for user extensions to be exchanged as non-validated, custom elements placed directly in an instance document, in which case additional application-level processing is required to validate the file, just as in the IAI approach. The use of such non-validated custom elements is illustrated in the example instance document where there were cases that AEX did not match IFC element definitions exactly, e.g., the pump predefined type. Not so apparent from the instance documents are issues that arose in creating them. The concept of role, for example, is rather loosely developed in IFC. Optional lists of roles can be assigned to an IfcPerson, to an IfcOrganization, and to an IfcPersonAndOrganization, but it is not easy to disambiguate the specific role played by a person and/or organization in a particular context. The AEX model, however, explicitly recognizes that they may play different roles in different contexts.

The concept of an equipment tag needs to be explored more fully than was possible in this study. Although both IFC and AEX models contain tag attributes, their meanings are not the same. In the AEX model, a tag can represent only the occurrence of a pump in a design, because of its focus on the information exchanges that take place during design and procurement. In the IFC model, the tag attribute is overloaded: it can represent either an occurrence of a pump in a design or an occurrence of a pump in the real world (see related discussion in "2.1.1 IFC Model Architecture") because of its focus on the lifecycle of a facility. This dual meaning needs to be disambiguated in the IFC model. At the same time, the AEX community needs to consider the ramifications of post-design needs for the capture and use of information about equipment, particularly in operations and maintenance. In the current study, only design-occurrence tags were mapped.

The content models for person differ. In ifcXML, a person is represented by separate title, first, middle, and last name elements, whereas name is represented by only a simple element in AEX. It seems that the AEX model should be expanded to facilitate the interchange of information with ifcXML-based systems as well as facilitate operations such as search by last name within an AEX-based system.

The AEX generatingSoftware element could be improved by expanding it to include 'description' or alternatively it could become derived from baseObject, which provides name and description. In the instance document, the generating software description had to be placed in the 'comment' attribute associated with the 'name' element, which is not as good as having a dedicated element for this purpose.

The ifcXML content model for address, used in several places, includes the notion of a logical location expressed by the Purpose element whose value is to be selected from an enumeration of OFFICE, DISTRIBUTIONPOINT, etc. In the AEX model, Objects have an equivalent 'function' element, but this element is not currently in the baseObject and so does not show up in

the mapping for Location. The AEX project could consider moving the 'function' element from the Object to the baseObject, which would enable mapping without adding a customBaseObject element. The AEX content model for Location also does not have a 'locationType' element, e.g., distinguishing between office, home, distribution point, etc. If one is introduced, and its allowable values are strings rather than an enumeration, then interoperability will be impaired.

The ifcXML content models for IfcPumpType and IfcFlowMovingDevice include a number of optional elements, for example, the Description element, whose value may be a string. The free-form use of these elements by implementers to convey additional information must be carefully controlled if interoperability between ifcXML-based and AEX-based applications is to be achieved.

The ability to distribute data related to a single pump across multiple documents was not tested in this study. The instance documents that were created using the ifcXML and AEX schemas were self contained. In any case, an ifcXML example would be rather artificial for the reason discussed in section "2.1.6 External References," whereas the AEX schema have been designed to handle external references, as discussed in section "2.2.4 Core Objects Overview," so the test would not be very illuminating.

5. CONCLUSIONS AND FUTURE WORK

A summary-level conclusion from the work completed is that it is possible to map from the semantic content of the ifcXML representation of a centrifugal pump to that of the AEX representation. This means, in turn, that it is possible to create software that translates and exchanges centrifugal pump data from an ifcXML environment to an AEX environment. Even in this limited context, however, it should be apparent from the backround discussion and from the tables presented in section "3.2 Results" that additional work is needed to approach the goal of seamless interoperability. Examples of this additional work include reconciling enumeration types (on both sides) and agreeing on the use of custom elements (on the AEX side) and user-defined property sets (on the ifcXML side). For attributes that map directly, for example, organization ID, there is still the need for implementer or user agreements regarding acceptable values.

Several aspects of the semantic mapping problem that were not considered here suggest needed future work. The capability in ifcXML to define and instance composite objects, inherited from the IFC and described in section "2.1.1 IFC Model Architecture," should be used to expand the centrifugal pump example to include its drive, base, and associated components, so that a more complete semantic mapping to AEX can be undertaken. With this mapping in hand, one or more use cases that deal with the exchange of information from AEX to ifcXML should be defined and the reverse mappings developed. This latter mapping likely will entail substantial use of the extensible IFC property-set capability. The capability in ifcXML to capture information about physical occurrences of equipment during operation and maintenance should be used as a starting point to consider how to bring forward information received via AEX-enabled software to support operation and maintenance and possibly how to expand AEX to include the kinds of physical occurrence-specific, time-dependent operational data that ifcXML can capture. An

example of a business case for this capability would be the transmission of performance data measured on pumps in service back to the pump manufacturer for diagnostic purposes.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] "World Wide Web Consortium Extensible Markup Language (XML) home page," web page at http://www.w3.org/XML/, accessed 09 April 2004.

[2] "International Alliance for Interoperability home page," web page at http://www.iai-international.org/iai_international/, accessed 09 April 2004.

[3] "FIATECH home page," web page at http://www.fiatech.org/, accessed 09 April 2004.

[4] "A Brief History of the Development of SGML," web page at http://www.sgmlsource.com/history/sgmlhist.htm, accessed 09 April 2004.

[5] GCA Standard 101-1983 Document Markup Metalanguage: GENCODE™ And The Standard Generalized Markup Language (SGML), Graphics Communications Association, Arlington, Virginia, 1983, ISBN: 0897402248.

[6] International Organization for Standardization, Geneva, Switzerland, web page at http://www.iso.ch, accessed 20 September 2004.

[7] "World Wide Web Consortium People: Tim Berners-Lee," web page at http://www.w3.org/People/Berners-Lee/, accessed 09 April 2004.

[8] "World Wide Web Consortium HyperText Markup Language (HTML) home page," web page at http://www.w3.org/MarkUp/, accessed 09 April 2004.

[9] "World Wide Web Consortium home page," web page at http://www.w3.org, accessed 09 April 2004.

[10] "XML Cover Pages: About XML Applications and Initiatives," web page at http://xml.coverpages.org/xmlApplications.html, accessed 09 April 2004.

[11] STEP: The Grand Experience, Kemmerer, S.J., editor, National Institute of Standards and Technology, Gaithersburg, Maryland, NIST SP 939, 1999.

[12] Information Modeling: the EXPRESS Way, Schenck, D. and Wilson, P., Oxford University Press, New York, New York, 1994, ISBN: 0195087143.

[13] ISO/IS 10303-21:2002 Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear-text encoding of the exchange structure, International Organization for Standardization, Geneva, Switzerland, 2002.

[14] "World Wide Web Consortium XML Schema," web page at http://www.w3.org/XML/Schema, accessed 09 April 2004.

[15] ISO/TS 10303-28:2003 Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data, International Organization for Standardization, Geneva, Switzerland, 2003.

[16] "ifcXML Language Binding of Express," Liebich, T., IAI International, 11 December 2001, web page at http://www.iai-international.org/iai_international/Technical_Documents/documentation/IFCXML/ifcXML_language_binding_V1-02.pdf, accessed 20 September 2004.

[17] "Flat ifcXML – A Discussion Paper," Liebich, T., AEC3 Ltd., undated, web page at http://www.iai-na.org/technical/flat_ifcXML_draft.pdf, accessed 20 September 2004.

[18] "International Alliance for Interoperability Information: Mission," web page at http://www.iai-international.org/iai_international/Information/Mission.html, accessed 09 April 2004.

[19] "VRML: Virtual Reality Markup Language home page," web page at http://www.w3.org/MarkUp/VRML/, accessed 13 April 2004.

[20] "International Overview of IFC-Implementation Activities," web page at http://www.bauwesen.fh-muenchen.de/iai/ImplementationOverview.htm, accessed 13 April 2004.

[21] "Centrifugal Pump Illustration," The American Heritage Dictionary of the English Language, Fourth Edition, web page at http://www.bartleby.com/61/imagepages/A4pump.html, accessed 20 April 2004.

[22] Daniel Bernoulli, Hydrodynamics, and Johann Bernoulli, Hydraulics, Translated from the Latin by Carmody, T. and Kobus, H., Dover Publications, New York, New York, 1968.

[23] "U.S. Census Bureau Current Industrial Reports: MA333P – Pumps and Compressors: 2002," [Table 1], web page at http://www.census.gov/industry/1/ma333p02.pdf, accessed 22 April 2004.

[24] Ibid. [Table 3].

[25] "IFC Technical Guide," Liebich, T., ed., International Alliance for Interoperability, October, 2000.

[26] "IFC2X Edition 2 Model Implementation Guide," Version 1.6, Liebich, T., ed., International Alliance for Interoperability, June, 2003.

[27] "Industry Foundation Classes IFC2x Edition 2 Documentation," web page at

http://www.iai-international.org/iai_international/Technical_Documents/R2x2_final/index.html, accessed 18 May 2004.

[28] ISO 1000:1992 SI units and recommendations for the use of their multiples and certain other units, International Organization of Standardization, Geneva, Switzerland, 1992.

[29] "AEX Public Release Version 1.0," web page at http://www.fiatech.org/projects/idim/aexresources.htm, accessed 19 August 2004.

[30] "ifcXML2," web page at http://www.iai-international.org/iai_international/Technical_Documents/IfcXML2.htm, accessed 25 May 2004.

[31] "AEX Public Release Version 1.0," web page at http://www.fiatech.org/projects/idim/aexresources.htm, accessed 19 August 2004.

[32] "Open Management Group UUID and GUID," web page at http://www.opengroup.org/dce/info/draft-leach-uuids-guids-01.txt, accessed 19 August 2004.