



# **National BIM Standard - United States® Version 3**

## **4 Information Exchange Standards**

### **4.2 Construction Operation Building information exchange (COBie) – Version 2.4**

#### **CONTENTS**

4.2.1 Scope .....	2
4.2.2. Normative References .....	17
4.2.3 Terms and definitions .....	22
4.2.4 Symbols and abbreviated terms .....	45
4.2.5 Business process documentation.....	47
4.2.6. Exchange requirement documentation.....	92
4.2.8 Conformance testing procedures .....	221
4.2.9 Implementation resources .....	236
4.2.10 Revision plans .....	243
Bibliography .....	245
Table of Figures .....	248
Annex A – COBie Mapping Rules .....	252
Annex B – Life Cycle information exchange (LCie) for Product and Product Type data exchanges, a technical Annex to COBie – Verion 2.4 .....	252

## 4.2.1 Scope

### 4.2.1.1 Business Case Description

*Author's Editorial Comment: Business process models supporting COBie were first documented in a U.S. Army Technical Report (East 2007). These business models were developed from a series of industry advisory panel meetings conducted in 2006 and 2007. Additional clarification of the scope of processes pertaining to the life-cycle of facility assets were proposed in East 2010. At the time of submission of this document another update to COBie-related business processes is being finalized for editing as part of a U.S. Army Technical Report (Fallon 2013). The objective of the research contained in this report was to use business process modeling in conjunction with "lean" management approaches, as in East 2011, to compare current with COBie-based business processes. As part of the NBIMS-US™ V3 COBie standard, an updated set business models developed in the latest report is provided for direct inclusion in the standard. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised information exchange submissions, the information provided in this section should be considered a Moderate Change.*

#### 4.2.1.1.1 Life-Cycle Phase List

- Study and Define Needs
- Develop Design Criteria
- Study Technical Feasibility
- Communicate Results Decision
- Develop Program – Space Program
- Develop Program – Product Program
- Prepare Invitation to Bid and Receive Proposal (Pre-Design)
- Explore Concepts – Design Early
- Develop Design – Design Schematic
- Develop Design – Design Coordinated
- Finalize Design – Design Final
- Prepare Invitation to Bid and Receive Proposals (Post Design)
- Respond to Pre-Proposal Inquiries
- Develop Pre-Construction Plan
- Identify Discrepancies
- Prepare Submittal Information – Product Type Selection
- Prepare Submittal Information – System Layout
- Organize Submittal Information
- Perform Submittal Review – Submittal Issue
- Provide Resources
- Execute Construction Activities
- Perform Equipment Testing

- Inspect and Approve Work
- Define, Record and Certify Discrepancies
- Closeout

While the use of shared, structured information, such as COBie, can be shown to have a direct have an impact on the speed and quality of value-added tasks, the business cases described below do not consider these beneficial effects. This is because in a generic model, such effects were too complex to model in the first study of the life-cycle impact of having COBie data. As a result, the business cases described here are conservative in their estimates. COBie 2.4 business cases are based on the value-added analysis technique published in East 2010. This approach organizes work operations into tasks that add value to the final product and those that do not. Work operations that do not add value are targeted for elimination in the streamlined business process.

The following opportunities for streamlining the production, exchange, use, and maintenance of COBie data through the elimination of non-value added operations are characterized by the following classification:

1. VALIDATION savings from the ability to programmatically check the space and equipment data for completeness, conformance to standards and conformance to requirements.
2. COPYING savings from reliance on electronic documents and data as the project record.
3. HANDLING savings from the adoption of managed project collaboration and management systems for transmittal and automated logging of project documents.
4. SEARCHING savings from the ability to electronically compare product data to product specifications.
5. REFORMATTING savings from adoption of a single, open standard data format for information relating to managed assets.
6. RECREATING savings from the use of a standard, structured data format for moving space and equipment information through the project process and into facility management, eliminating the need for data re-entry. In addition, rework is a form of recreation.

Savings can be achieved in non-value added tasks to different degrees. Some tasks can be completely eliminated, while others can be automated or streamlined. Of the 25 life cycle processes studied in Fallon 2013, 19 of these would obtain a savings from the expected approach. The following processes are defined and include the analysis of possible savings from the use of COBie-based information exchanges.

#### **4.2.1.1.1 Study and define needs**

Standard facility information must be available in order to determine the basic requirements for a potential project. The Owner identifies the need and either develops technical criteria for the facility if none exist or utilizes existing technical criteria if available. If it does exist, this information must be checked for relevancy every five years to remain consistent with overall needs. Potential savings in this process include, but may not be limited to, the following:

- Reproduction savings from reliance on electronic documents and the elimination of paper.

#### 4.2.1.1.1.2 Develop design criteria

Specification information for equipment based on facility criteria is generated early in the planning process by the Owner. This information must be checked for relevancy every five years to remain consistent with overall needs. Potential savings in this process include, but may not be limited to, the following:

- Reproduction savings from reliance on electronic documents and the elimination of paper.

#### 4.2.1.1.1.3 Study technical feasibility

The Feasibility Study allows the Owner to evaluate different options (typically three) based on the identified requirements before finalizing specific information about a project. The Architect or Planner develops the study based on the information contained in the Facility Criteria and Discipline Specification information exchanges. Potential savings in this process include, but may not be limited to, the following:

- Reproduction savings from reliance on electronic documents and the elimination of paper
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents.

#### 4.2.1.1.1.4 Communicate results decisions

Initial criteria about a project must be established in order to evaluate the project feasibility. The Owner evaluates the Facility Criteria, Discipline Specifications, and Feasibility Study to determine whether or not to move forward with the project. Potential savings in this process include, but may not be limited to, the following:

- Reproduction savings from reliance on electronic documents and the elimination of paper.

#### 4.2.1.1.1.5 Develop program – space program

Once the Project Definition has been established and approved, further development of the project requirements can occur. The Architect or Planner evaluates information contained in the Project Definition information exchange to identify space needs based on the facility type. Space requirements, based on facility type, are located online in electronic document format and must be downloaded. These documents are typically printed by the end user. If no standard facility space criteria exist, it must be created by referencing similar facility types. Potential savings in this process include, but may not be limited to, the following:

- Design professionals typically re-enter the Owner's space requirements into the system they use for space programming. COBie-formatted data permits data to be transferred directly from the Owner to the Architect or Planner's system
- Requirements associated with each space are typically gathered and then documented on Room Data Sheets. COBie format would either eliminate the need to produce room data sheets or support automation of their production
- The Architect/Planner sends the Space Program to the Owner's Representative for review. Currently, this is done by comparing 2 documents. Use of COBie format would permit automated checking.
- If the Architect/Planner could automate checking of his work product against the Owner's requirements, then a rework/re-review cycle could be eliminated.

- Reproduction savings from reliance on electronic documents and the elimination of paper
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents

#### **4.2.1.1.1.6 Develop program – product program**

The Architect or Planner evaluates information contained in the Project Definition information exchange to identify product needs based on the facility type. Requirements for products based on facility type are located online in electronic document format and must be downloaded. These documents are typically printed by the end user. If no standard facility product criteria exist, it must be created by referencing similar facility types. Potential savings in this process include, but may not be limited to, the following:

- Design professionals typically re-enter the Owner's product standards into the system they use for design. COBie-formatted product standards would permit direct transfer from the Owner to the Architect or Planner's system
- If the Owner's Representative returns the Product Program because it does not meet the Owner's product requirements, the Architect/Planner must recreate the Product Program.
- The Architect/Planner sends the Product Program to the Owner's Representative for review. Currently, this is done by comparing 2 documents. Use of COBie format would permit automated checking.
- If the Architect/Planner could automate checking of his work product against the Owner's requirements, then a rework/re-review cycle could be eliminated.
- Reproduction savings from reliance on electronic documents and the elimination of paper
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents

#### **4.2.1.1.1.7 Prepare invitation to bid and receive proposals (pre-design)**

Once the major criteria have been determined, the Owner's Representative prepares and distributes a Request for Proposal (RFP). Potential savings in this process include, but may not be limited to, the following:

- Reproduction savings from reliance on electronic documents and the elimination of paper in both soliciting and submitting proposals
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. A managed electronic collaboration system with a "bidding" module can handle distribution of Requests for Proposal, receiving questions, issuing addenda and receiving and securing the proposals submitted by design firms.

#### **4.2.1.1.1.8 Explore concepts – design early**

The Architect utilizes the specific information produced during pre-design to develop a solution that reflects the requirements stated in the Project Definition, Space Program, and Product Program. Currently, the owner requires 6 hard copies to be submitted for each review cycle. The Architect performs a QA/QC check before submitting to the Owner's Representative. After receiving the submission, the Owner's Representative validates the documents (reviews) and provides comments to the Architect. The Architect and Consultants are then required to update the documents based on the comments. After

revisions are made, the Architect resubmits. Potential savings in this process include, but may not be limited to, the following:

- Although the Owner's requirements might be provided as e-documents, the design team typically reformats the information to be compatible with their design systems. COBie-formatted requirements data permits direct transfer from the Owner to the design consultants' systems.
- If the Owner's Representative rejects the Concept Design because it does not meet the Owner's space requirements, the Architect must recreate the Concept Design.
- COBie would permit the Architect to automate checking of his Concept Design against the Owner's space requirements, saving the Architect time and potentially eliminating a rework/re-review cycle.
- The Architect sends the Design Early documents to the Owner's Representative for review. Currently, this review is done manually. Use of COBie format would permit automated checking of space program at this phase.
- Reproduction savings from reliance on electronic documents and the elimination of paper
- In a paper-based process, review comments often need to be transferred to multiple document copies.
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents

#### **4.2.1.1.1.9 Develop design – design schematic**

The Architect further develops the approved Design Early deliverable documents to produce the Design Schematic documents. Currently, the owner requires 6 hard copies to be submitted for each review cycle. The Architect performs a QA/QC check before submitting to the Owner's Representative. After receiving the submission, the Owner's Representative validates the documents and provides comments to the Architect. The Architect and Consultants are then required to update the documents based on the comments. After revisions are made, the Architect resubmits. Potential savings in this process include, but may not be limited to, the following:

- Design Schematic phase requires quantity take-offs (QTOs) for cost estimating. QTOs are a recreation of information because the items have already been documented in the drawings or BIM. COBie addresses spaces and products/equipment. It provides space areas and product types and counts.
- If the Owner's Representative rejects the Design Schematic documents because the design does not meet the Owner's space or product requirements, the Architect must recreate the design.
- Although the Owner's requirements might be provided as e-documents, the design team spends considerable time developing product type templates (or BIM content), as well as specifications. COBie-formatted requirements data could be used directly.
- If the Architect and his Consultants could automate checking of their design against the Owner's space and product requirements, they would save checking time and a rework/re-review cycle could potentially be eliminated.
- The Architect sends the Design Schematic documents to the Owner's Representative for review. Currently, this review is done manually. Use of COBie format would permit automated checking of space and product program at this phase.

- Reproduction savings from reliance on electronic documents and data and the elimination of paper
- In a paper-based process, review comments often need to be transferred to multiple document copies.
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents

#### **4.2.1.1.1.10 Develop design – product type template, product type candidate**

As the design progresses, performance characteristics and suitable products for the building systems are identified. System types and equipment are identified by the Specifier based on the facility requirements. Six copies are required to be submitted for review. Some products are defined in more detail by identifying manufacturers and model numbers which meet requirements (Basis of Design). In these cases, 3 qualifying products should be listed. Potential savings in this process include, but may not be limited to, those identified in the previous section.

#### **4.2.1.1.1.11 Develop design – design coordinated**

The Architect further develops the approved Design Schematic deliverable documents to produce the Design Coordinated documents. In addition, the building systems are coordinated to eliminate spatial interferences. This is the major coordination submittal before the final delivery package. The owner requires 6 hard copies to be submitted for each review cycle. Due to the higher level of coordination and increase in number of interested reviewing parties, more copies are sometimes needed. The Architect performs a QA/QC check before submitting to the Owner's Representative. After receiving the submission, the Owner's Representative validates the documents and provides comments to the Architect. The Architect and Consultants are then required to update the documents based on the comments. After revisions are made, the Architect resubmits. Potential savings in this process include, but may not be limited to, the following:

- Design Coordinated phase requires quantity take-offs (QTOs) for cost estimating. QTOs are a recreation of information because the items have already been documented in the drawings or BIM. COBie addresses spaces and products/equipment. It provides space areas and product types and counts.
- If the Owner's Representative rejects the Design Coordinated documents because the design does not meet the Owner's space or product requirements, the Architect must recreate the design.
- Although the Owner's requirements might be provided as e-documents, the design team spends considerable time developing product type templates (or BIM content), as well as specifications. COBie-formatted requirements data could be used directly.
- Candidate Products (typically 3 qualifying products) are identified for each product type template. This is done through reviewing product literature. Standard, structured product data available in COBie format would allow automated product selection based on the product type templates.
- If the Architect and his Consultants could automate checking of their design against the Owner's space and product requirements, they would save checking time and a rework/re-review cycle could potentially be eliminated.
- The Architect sends the Design Coordinated documents to the Owner's Representative for review. Currently, this review is done manually. Use of COBie format would permit automated checking of space areas and product data against Owner requirements.

- Reproduction savings from reliance on electronic documents and data and the elimination of paper
- In a paper-based process, review comments often need to be transferred to multiple document copies.
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents.

#### **4.2.1.1.1.12 Develop design – product type candidate**

The performance requirements of building systems and equipment are further refined and documented by the Specifier during this phase. Any equipment, products, or systems not selected previously are identified. Specific manufacturers and model numbers are noted. Three qualifying products are identified. Six copies are submitted for each review cycle. Potential savings in this process include, but may not be limited to, those identified in the previous section.

#### **4.2.1.1.1.13 Finalize design – design final**

The Design Final package is the final set of contract documents ready for bid solicitation by the Owner. This final design deliverable does not require another review by the Owner's Representative. Potential savings in this process include, but may not be limited to, the following:

- Design Final phase requires quantity take-offs (QTOs) for cost estimating. QTOs are a recreation of information because the items have already been documented in the drawings or BIM. COBie addresses spaces and products/equipment. It provides space areas and product types and counts.
- Design Final requires a Quality Control Review to evaluate both technical accuracy and discipline coordination. COBie supports automate checking of the design against the Owner's space and product requirements, saving checking time.
- Reproduction savings from reliance on electronic documents and data and the elimination of paper.
- In a paper-based process, review comments often need to be transferred to multiple document copies.
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. Managed electronic collaboration systems will notify intended recipients when e-documents are released and automatically log both the issuing and viewing of those documents.

#### **4.2.1.1.1.14 Finalize design – product type candidate**

At this phase of the project all equipment and system types must be identified by the Specifier. Product information from the Design Coordinated phase is incorporated into to this phase. Three (3) qualifying products for each type required must be listed. Potential savings in this process include, but may not be limited to, those identified in the previous section.

#### **4.2.1.1.1.15 Prepare invitation to bid and receive proposals (post-design)**

Once the design is complete, the Owner packages the Design Final Documents information together with other owner-supplied information (e.g., contractual terms) and creates a Request for Proposals (RFP)



Package. This becomes the official bid set. Potential savings in this process include, but may not be limited to, the following:

- Reproduction savings from reliance on electronic documents and the elimination of paper in both soliciting and submitting proposals
- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. A managed electronic collaboration system with a “bidding” module can handle distribution of Requests for Proposal, receiving questions, issuing addenda and receiving and securing the bids submitted by Contractors.

#### **4.2.1.1.16 Respond to pre-proposal inquiries**

Before finalizing a bid proposal, the Contractor typically requests additional information or clarification of some bid documents. Potential savings in this process include, but may not be limited to, the following:

Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. A managed electronic collaboration system with a “bidding” module can handle receipt of bidder questions and issuing addenda.

#### **4.2.1.1.17 Develop pre-construction plan**

The Contractor is required to develop a Pre-Construction Plan that describes how the Contractor will make provisions for managing the construction of the facility. This is sent as a submittal package. Refer to the Submittal Package exchange for detailed requirements related to transmitting and handling Pre-Construction Plan submittals. Savings from COBie exchanges were not explicitly identified in the referenced study.

#### **4.2.1.1.18 Identify discrepancies**

The Contractor submits a Request for Information (RFI) to ask for clarification during the construction process. These questions may be due to but not restricted to ambiguities or contradictions in the drawings or to site conditions. Potential savings in this process include, but may not be limited to, the following:

- Elimination of administrative costs associated with handling paper document transmittals as well as the delivery expense. A managed electronic collaboration system with a “bidding” module can handle receipt of bidder questions and issuing addenda.

#### **4.2.1.1.19 Prepare submittal information – product type selection**

The Contractor and Sub-Contractors gather information for products identified in the Design Final documents and prepare submittals. Refer to the Submittal Package exchange for detailed requirements related to transmitting and handling Product Type Selection submittals. Savings from COBie exchanges were not explicitly identified in the referenced study. Note that benefits from having standard product template data through the Life-Cycle information exchange for Equipment Assets and Specifiers’ Properties information exchange (SPie) projects are expected to address many issues related to streamlining the delivery of facility asset information.

#### **4.2.1.1.20 Prepare submittal information – system layout**

The Contractor and Sub-Contractors review information for products identified in the Design Final documents and prepare shop drawings. Refer to the Submittal Package exchange for detailed requirements related to transmitting and handling System Layout submittals. Savings from COBie

exchanges were not explicitly identified in the referenced study. Note that benefits from having standard system layout data through the HVAC information exchange (HVACie), Water System information exchange (WSie), Electrical System information exchange (SPARKie), and Building Automation Management information exchange (BAMie) are expected to be addressed directly in those projects.

#### **4.2.1.1.1.21 Organize submittal information**

The Contractor organizes the required submittal information and creates Submittal Packages to be reviewed by the Owner's Representative and/or Architect. Six hard copies are required to be submitted for review. Potential savings in this process include, but may not be limited to, the following:

- Contractors and Subcontractors must extract product requirements from the specifications. COBie provides product requirements in a concise, computable form.
- Contractors must compile disparate product data formats into Product Submittal Items and Submittal Packages for the Architect's approval. COBie formats product data consistently.
- Contractors must validate product data against the specifications before including them in a Submittal. COBie supports automate checking of the data against the product specifications, saving time and reducing the number of Product Submittals rejected. This reduces rework.(See Recreating above.)
- Reproduction savings from reliance on electronic documents and data and the elimination of paper
- In a paper-based process, review comments often need to be transferred to multiple Submittal copies.
- Elimination of administrative costs associated with handling paper document Submittals as well as the delivery expense. These costs are high, due to the large number of documents and the requirement for multiple copies. Managed electronic collaboration systems will notify reviewers when Submittal Packages are uploaded, automatically log both the release and the reviewing of those documents and track ball-in-court responsibility and due dates.

#### **4.2.1.1.1.22 Perform submittal review – submittal issue**

The Architect and/or Sub-Consultants validate the submittals provided by the Contractor and provide comments. Multiple hard copies are required. Potential savings in this process include, but may not be limited to, the following:

- If a Submittal Item is rejected by the reviewer (typically the Architect and the Architect's Consultants), the Contractor or Subcontractor must redo the Submittal. COBie supports automated validation product characteristics against the specification, lowering the number of Product Submittals rejected.
- Submittal reviewers (typically the Architect and the Architect's Consultants) must also check Product Submittal data against the specifications. COBie supports automated checking, saving time.
- Reproduction savings from reliance on electronic documents and data and the elimination of paper
- In a paper-based process, review comments often need to be transferred to multiple Submittal copies.
- Elimination of administrative costs associated with handling paper document Submittals as well as the delivery expense. These costs are high, due to the large number of documents and the requirement for multiple copies. Managed electronic collaboration systems will notify reviewers

when Submittal Issues are uploaded, automatically log both the release and the reviewing of those documents and track ball-in-court responsibility and due dates.

#### **4.2.1.1.1.23 Provide resources**

The Contractor contacts a Supplier to order equipment and materials. The Supplier then provides a price quote to the Contractor for the equipment and/or materials. The Contractor verifies the specifications of the equipment and/or materials in the quote against approved submittal documentation and then submits them to the Owner's Representative and/or Architect for approval. Savings from COBie exchanges were not explicitly identified in the referenced study.

#### **4.2.1.1.1.24 Execute construction activities**

The Contractor installs the building equipment, materials, and systems using the design final drawings, approved shop drawings, product data, and manufacturer's instructions. Potential savings in this process include, but may not be limited to, the following:

- While the project is on-going, the Contractor must continually prepare a Product Installation report that describes the status of installed components and corresponding data. The Contractor then spends time in the office processing these notes and compiling the Report. The COBie worksheet would be a vehicle for field data entry, as well as a reference to components. This would allow the Contractor to reduce office time.
- Elimination of administrative costs associated with handling paper documents as well as the delivery expense.

#### **4.2.1.1.1.25 Perform equipment testing**

After the Contractor completes the installation process, the equipment/systems must be tested by activating the equipment. This testing must be completed with the Owner's Representative and Manufacturer's representative present. Savings from COBie exchanges were not explicitly identified in the referenced study.

#### **4.2.1.1.1.26 Inspect and approve work**

When the Contractor has completed installation of equipment or systems, a notification is sent to the Architect indicating the installed item is ready for inspection/observation. The Architect conducts regular inspections of the installed construction work. The findings of the inspections including any deficiencies with the installation of the construction work are documented in a report. If deficiencies are identified in the inspection report, the Contractor corrects them and then requests a re-inspection. Potential savings in this process include, but may not be limited to, the following:

- The Architect must validate each Contractor Pay Request through a site visit to determine work progress. Typically, the Architect takes drawings to the site to check that items billed have been put in place. The Architect also notes any defects in workmanship. The Architect then spends time in the office composing field notes and quantifying work put in place to support or refute the Pay Request. COBie would provide a definitive list of items required per room or floor that could be "checked off" and automatically totaled. This would allow the Architect to reduce office time.
- Elimination of administrative costs associated with handling paper documents as well as the delivery expense. Managed electronic collaboration systems can notify the Contractor if the Pay Request has been accepted or rejected and deliver the Observation Field Report with tracking.

#### **4.2.1.1.1.27 Define, record and certify discrepancies**

The Architect creates a final punch-list based upon a survey of the completed construction work. The Contractor corrects the deficiencies identified in the punch-list. The Architect verifies that the Contractor has corrected the deficiencies in the punch-list by performing a final walkthrough. Savings from COBie exchanges were not explicitly identified in the referenced study.

#### **4.2.1.1.1.28 Closeout**

The Contractor gathers all as-built information related to the project and forwards the information to the Owner. Four copies are typically required. Potential savings in this process include, but may not be limited to, the following:

- Contractor must assemble the Turnover Package. A managed electronic collaboration system stores and indexes all documents submitted as they are uploaded. This greatly reduces the time required to find the necessary documents and assemble the Turnover Package, saving the Contractor time, improving the completeness and quality of the Turnover Package, and making the Turnover Package available to the Owner at an earlier date.
- Reproduction savings from turnover of electronic documents and data and the elimination of paper. Typically four sets of Turnover documents are required.
- Elimination of administrative costs associated with handling paper documents as well as the delivery expense.

#### **4.2.1.1.2 Business Case Analysis**

Three business cases were developed to support the COBie 2.4 specification. The first two were developed for the commonly used Office Building and Medical Clinic models East 2012a). These models were used as a representation of commercial and medical facilities. The third business case, a Chicago Transit Authority (CTA) station construction program, was also analyzed. The inclusion of the CTA construction program allowed the study authors to evaluate: (1) the assumptions regarding savings and values used on the experimental BIM models and (2) to demonstrate how mixed paper/electronic approaches are included in the business process analysis.

The business process analysis conducted contains 210 variables related to the specification, documentation and fulfillment of managed asset requirements across the complete life-cycle of COBie-related information exchanges. These variables included information about numbers of drawing sets, count and complexity of space inventories, extent of specifications, numbers of equipment schedules and required copies of product data sheets. The analysis assumes that there were no changes to the requirements in the contracts other than a change from traditional to COBie-based business processes.

The baseline, or Current Process, used in analyzing the Office and Clinic projects assumed a paper-based communication and documentation system. The Expected Process assumes an electronic collaboration communication and documentation system and use of standard, structured data (COBie) for data exchanges. The COBie Calculator determined the expected values of the variables affected. The analysis conducted evaluated considered which of the previously classified non-value added activities could be eliminated or streamlined, when compared to a traditional process.

The Office project was based on a federal typical Army standard office building. The total square footage was 40,053 SF. There were 50 product types and 1,706 scheduled components in the sample model. The table below shows the results of the total analysis across each of the new COBie business processes, and all types of users.

**Table 1 Business Case - Office Project**

OmniClass™ Project Phase (Table31)	Current Cost (\$)	Expected Cost (\$)	Savings (\$)	Savings (%)
Facility Criteria	-	-	-	0%
Design Specification	\$15.00	-	\$15.00	100%
Feasibility Study	\$560.00	\$25.00	\$535.00	96%
Project Definition	\$10.00	-	\$10.00	100%
Space Program	\$615.00	\$10.00	\$605.00	98%
Product Program	\$790.00	\$5.00	\$785.00	99%
Request for Proposal	\$800.00	\$100.00	\$700.00	88%
Design Early	\$5,300.00	\$90.00	\$5,210.00	98%
Design Schematic	\$9,900.00	\$600.00	\$9,300.00	93%
Design Coordinated	\$26,300.00	\$4,100.00	\$22,200.00	85%
Design Final	\$12,900.00	\$900.00	\$12,000.00	93%
Request for Proposal	\$930.00	\$30.00	\$900.00	97%
Inquiry Issue	\$1,210.00	\$30.00	\$1,180.00	98%
Pre-Construction Plan	-	-	-	0%
Inquiry Issue (RFI)	\$1,280.00	\$10.00	\$1,270.00	95%
Product Type Selection	-	-	-	0%
System Layout	-	-	-	0%
Submittal Package	\$34,400.00	\$3,000.00	\$31,400.00	91%
Submittal Issue	\$73,500.00	\$400.00	\$73,100.00	99%
Purchase Order	-	-	-	0%
Product Installation	\$41,005.00	\$5.00	\$41,000.00	100%
Start-Up	-	-	-	0%
Product Inspection	\$15,900.00	\$600.00	\$15,300.00	96%
Punch-list Issue	-	-	-	0%
Turnover Package	\$6,300.00	\$100.00	\$6,200.00	98%
<b>Total</b>	<b>\$232,000.00</b>	<b>\$10,000.00</b>	<b>\$222,000.00</b>	<b>96%</b>

The Medical Clinic project is based on an actual government medical clinic whose design drawings and operations and maintenance manuals have been provided, in redacted form. The total square footage was 49,571 SF. There were 155 product types and 3,950 components in the sample model. The table below shows the results of the total analysis across each of the new COBie business processes, and all types of users.

**Table 2 Business Case - Medical Clinic**

OmniClass™ Project Phase (Table31)	Current Cost (\$)	Expected Cost (\$)	Savings (\$)	Savings (%)
Facility Criteria	-	-	-	0%
Design Specification	\$135.00	-	\$135.00	100%
Feasibility Study	\$670.00	\$10.00	\$660.00	99%
Project Definition	-	-	-	0%
Space Program	\$840.00	\$10.00	\$830.00	99%
Product Program	\$1,900.00	\$5.00	\$1,895.00	99%
Request for Proposal	\$900.00	\$100.00	\$800.00	89%
Design Early	\$20,840.00	\$300.00	\$20,900.00	98%
Design Schematic	\$33,400.00	\$1,900.00	\$31,500.00	94%
Design Coordinated	\$91,100.00	\$12,600.00	\$82,700.00	86%

OmniClass™ Project Phase (Table31)	Current Cost (\$)	Expected Cost (\$)	Savings (\$)	Savings (%)
Design Final	\$30,400.00	\$2,200.00	\$28,200.00	93%
Request for Proposal	\$1,990.00	\$30.00	\$1,960.00	98%
Inquiry Issue	\$1,210.00	\$30.00	\$1,180.00	98%
Pre-Construction Plan	-	-	-	0%
Inquiry Issue (RFI)	\$1,800.00	-	\$1,800.00	100%
Product Type Selection	-	-	-	0%
System Layout	-	-	-	0%
Submittal Package	\$60,700.00	\$5,000.00	\$55,700.00	92%
Submittal Issue	\$214,600.00	\$500.00	\$214,100.00	99%
Purchase Order	-	-	-	0%
Product Installation	\$94,800.00	-	\$94,800.00	100%
Start-Up	-	-	-	0%
Product Inspection	\$29,100.00	\$1,200.00	\$27,900.00	96%
Punch-list Issue	-	-	-	0%
Turnover Package	\$14,100.00	\$100.00	\$14,000.00	99%
<b>Total</b>	<b>\$598,000.00</b>	<b>\$24,000.00</b>	<b>\$574,000.00</b>	<b>96%</b>

The third business case was conducted on the Chicago Transit Authority (CTA) station program. Unlike the Office and Clinic analysis, described above, the CTA utilized a web-based managed collaboration system with automated workflow and some paper-based communication in executing its projects. CTA processes combined the exchange of paper and e-paper documents. CTA did not make use of standard, structured data for exchanges, i.e. COBie. Changes to the business process model variables were made to take into account the use of CTA's web-based collaboration tools. This result of comparing the current CTA processes with a process that captured asset information, via COBie is provided in the table below.

**Table 3 - Business Case Transit Authority**

OmniClass™ Project Phase (Table31)	Current Cost (\$)	Expected Cost (\$)	Savings (\$)	Savings (%)
Facility Criteria	-	-	-	0%
Design Specification	-	-	-	0%
Feasibility Study	\$10.00	\$10.00	-	0%
Project Definition	-	-	-	0%
Space Program	\$510.00	\$10.00	\$500.00	98%
Product Program	\$890.00	\$30.00	\$860.00	97%
Request for Proposal	\$80.00	\$80.00	-	0%
Design Early	\$11,800.00	\$300.00	\$11,500.00	97%
Design Schematic	\$31,800.00	\$1,500.00	\$30,300.00	95%
Design Coordinated	\$68,000.00	\$7,600.00	\$60,400.00	89%
Design Final	\$28,200.00	\$1,900.00	\$26,300.00	93%
Request for Proposal	\$20.00	\$10.00	\$10.00	50%
Inquiry Issue	\$100.00	\$100.00	-	0%
Pre-Construction Plan	-	-	-	0%
Inquiry Issue (RFI)	\$3,200.00	\$300.00	\$2,900	91%
Product Type Selection	-	-	-	0%
System Layout	-	-	-	0%
Submittal Package	\$32,700.00	\$1,800.00	\$30,900.00	94%
Submittal Issue	\$61,400.00	\$500.00	\$60,900.00	99%
Purchase Order	-	-	-	0%
Product Installation	\$12,900.00	-	\$12,900.00	100%

OmniClass™ Project Phase (Table31)	Current Cost (\$)	Expected Cost (\$)	Savings (\$)	Savings (%)
Start-Up	-	-	-	0%
Product Inspection	\$326,800.00	\$13,100.00	\$313,700.00	96%
Punch-list Issue	-	-	-	0%
Turnover Package	\$15,400.00	\$300.00	\$15,100.00	98%
<b>Total</b>	<b>\$594,000.00</b>	<b>\$28,000.00</b>	<b>\$566,000.00</b>	<b>95%</b>

In comparing the savings projected from first two projects, developed from the buildingSMART alliance® Common BIM File repository and the savings projected in the third project, based on real life-cycle information collected through direct observation the conclusion be reached when evaluating these business cases is that the predicted savings are surprisingly consistent. In hindsight this result is consistent with the model that been developed to evaluate these business cases. This is because a consistent percentage of savings can be achieved for projects of any size when changing from paper-based document processing to using web-based tools that exchange COBie-based information. This consistent percent savings across all types of projects was directly due to the types of business operations contained in the business process models that could be eliminated as a result of no longer having to create, recreate, transmit and manually check data provided in COBie versus document-based formats.

#### 4.2.1.2 Participants and stakeholders

*Author's Editorial Comment: Information about participants and stakeholders were not explicitly provided in NBIMS-US™ V2 COBie documentation although the information could be found in originally referenced project reports. As part of the NBIMS-US™ V3 COBie submission, an updated list of participants and stakeholder is provided for direct inclusion in the standard. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange documents, the information provided in this section should be considered a moderate change.*

##### 4.2.1.2.1 Participants List

4Projects	Graphisoft SE
AEC 3 Ltd	Hitchcock Consulting, Inc
AssetWORKS	International Business Machines, Inc
ARCHIBUS	Kristine Fallon and Associates
Asite Solutions Ltd	LATISTA
Autodesk, Inc	Onuma
Bentley Facilities	Organization for the Advancement of Structured Information Standards, Open Building Information Exchange (oBIX)
Bentley Systems	Technical Committee
City of Chicago, Chicago Transit Authority	
CxAlloy	National Information Exchange Model (NIEM)
DRofus	Nemetschek Vectorworks, Inc.
DRRW, Inc	Plannon
EagleCMMS	TC9, Inc
EcoDomus, Inc	TMA Systems
FM: Systems	University of Chicago

United States Army, Corps of Engineers,

Engineer Research and Development Center

**4.2.1.2.2 Stakeholders List**

Owners	Manufacturers
Planners	Commissioning Agents
Architects	Facility Operators
Consulting Engineers	Facility Managers
Construction Managers	Asset Managers
Contractors	Software Developers
Sub-Contractors	System Analysts
Fabricators	Systems Integrators
Suppliers	

**4.2.1.2.3 Stakeholders coverage analysis**

The table below briefly describes how each COBie constituents' concerns are addressed through the COBie standard.

**Table 4 Stakeholder Coverage Analysis**

Constituent	Coverage Analysis
Owners	The format for the specification of requirements is provided by COBie. A generic guide to the contents of COBie, the COBie Guide, should be updated for a given owner's requirements before specifying COBie.
Planners	An extended version of the COBie format, defining specific requirements for the planning and programming stage has been defined as the BPie project.
Architects	Quality standards required by the COBie Guide ensure that designers create COBie data that matches the content of all scheduled assets.
Consulting Engineers	Quality standards required by the COBie Guide ensure that designers create COBie data that matches the content of all scheduled assets. Merging rules described in this standard provide implementation support required.
Construction Managers	COBie business case analysis supports streamlining those activities that have non-value added operations.
Contractors	COBie business case analysis supports streamlining those activities that have non-value added operations.
Sub-Contractors	COBie business case analysis supports streamlining those activities that have non-value added operations.



Constituent	Coverage Analysis
Constituent	Coverage Analysis
Fabricators	COBie business case analysis supports streamlining those activities that have non-value added operations.
Suppliers	COBie business case analysis supports streamlining those activities that have non-value added operations.
Manufacturers	COBie business case analysis supports streamlining those activities that have non-value added operations.
Commissioning Agents	COBie business case analysis supports streamlining those activities that have non-value added operations.
Facility Operators	Allows handover information to be imported prior to operating the facility.
Facility Managers	Allows handover information to be imported prior to managing the facility.
Asset Managers	Allows handover information to be imported prior to managing the facility's assets.
Software Developers	Improved IFC specification, and support for IFC 4, assist those using IFC. Updated XML formats will decrease the cost of XML-based exchanges and use in web-services.
System Analysts	Improved IFC specification, and support for IFC 4, assist those using IFC. Updated XML formats will decrease the cost of XML-based exchanges and use in web-services.
Systems Integrators	Improved IFC specification, and support for IFC 4, assist those using IFC. Updated XML formats will decrease the cost of XML-based exchanges and use in web-services.

#### 4.2.2. Normative References

*Author's Editorial Comment: This section updates the "Criteria Reference Standards (Other)" included in the COBie NBIMS-US™ V2 documentation, section A.8. New reference standards, Schematron and NIEM, were added to this document to reflect changes necessary for NBIMS-US™ V3. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange submissions, the information provided in this section should be considered a moderate change.*

##### 4.2.2.1 References and standards

###### 4.2.2.1.1 Reference standards list

###### 4.2.2.1.1.1 Industry Foundation Class (IFC) Model, Version 2x3

###### 4.2.2.1.1.2 Industry Foundation Class (IFC) Model, Version 4 (ISO 16739)

#### 4.2.2.1.1.3 Standard for Product Exchange (STEP) (ISO 10303)

ISO 10303-11, Industrial automation systems and integration — Product data representation and exchange — Part 11: description methods: The EXPRESS Language Reference Manual. This is the format of figures found in the referenced MVD.

ISO 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure. This is typically referred to as the STEP Physical File Format.

ISO 10303-28, Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas. This is typically referred to as ifcXML.

#### 4.2.2.1.1.4 Schematron (ISO/IEC 19757-3:2006)

*Author's Editorial Comment: The open source COBie Tool Kit has been since March 2012 the tool used to test files against the COBie format requirements. The rules used in the COBie Tool Kit, provided as an Annex to this standard, were developed and are maintained in Schematron format.*

An international standard format for the development of testing rules for COBie files is based on ISO/IEC 19757-3:2006 Information technology -- Document Schema Definition Language (DSDL) -- Part 3: Rule-based validation – Schematron. Schematron is widely used and available from ISO free of charge. Schematron has proven easy to use since its rules are based on XML tree searching and uses the XPATH programming language.

The use of Schematron is optional. Any other testing tool may be used provided that those tools test their algorithms to equivalence with the Schematron rules provided in this standard and included in the COBie Tool Kit application.]

#### 4.2.2.1.1.5 National Information Exchange Model (NIEM), version 2.1

*Author's Editorial Comment: NBIMS-US™ V2 COBie 2.26 allows three alternative exchange formats. These formats are the STEP Physical File Format, the ifcXML format, and SpreadsheetML format. For the purposes of the COBie standard there is no difference between these formats. The format selection criterion has been left up to the interest of the market place. The most widely used of these three formats, to date, has been the SpreadsheetML format. SpreadsheetML is an XML schema reflecting the organization of spreadsheet data in Microsoft Excel 2003.*

While the spreadsheet view of COBie data has been essential to COBie's understanding and success to date, software vendors have expressed concern with the spreadsheet format. The first concern is that the SpreadsheetML schema is a semi-proprietary XML specification directly tied to Microsoft Excel 2003. Eventually such a format will be depreciated by Microsoft and NBIMS-US™ will have to select an alternate format. European users of COBie have also expressed concern since they are restricted and in some cases prohibited adopting proprietary data standards.

The second concern with the SpreadsheetML XML is the complexity of that schema. The structure of SpreadsheetML is closely tied to the structure of a spreadsheet. One issue with this spreadsheet-based structure is that the COBie in spreadsheet form is a “referential” schema rather than a “nested” schema. The “nested” approach is more likely to be used by software developers as it is more compact and requires less testing than the nested schema.

The third concern expressed by many software firms and system integrators has been a concern that buildingSMART standards based on STEP or the SpreadsheetML are inconsistent with streamlined XML schema used to support system-to-system direct web-service interfaces. Such a streamlined XML schema would also more readily support the development and adoption of sub-schema that could be used to spark a productivity revolution in the construction industry.

As a result of these concerns, the COBie project team developed an alternative XML schema to support a broader use of the facility asset information found in COBie files. The new XML schema format for COBie data, whose technical details are discussed in detail in later sections of this standard, is called COBieLite. An essential design criterion in the development of COBieLite was that COBieLite conform to existing, freely available, highly-visible, United States national XML schema standards. The National Information Exchange Model (NIEM) a United States Government standard was selected as the basis from which the COBieLite schema would be developed.

NIEM is a United States Government standard developed by the U.S. Department of Homeland Security and the U.S. Department of Justice. NIEM currently contains a variety of schemas used at the federal and state levels of United States government. Domains currently included in this schema that relate to the buildingSMART mission include justice, public safety, emergency and disaster management, and homeland security enterprises. Schemas for building interiors, such as those that could be provided through buildingSMART alliance® standards are not currently included in the NIEM catalogue; however, COBieLite will facilitate such a discussion and further the exchange of COBie data into communities that use information about facilities, not just those who design, build, and operate them.

#### **4.2.2.1.2. (Not used)**

##### **4.2.2.1.2.1 OmniClass™ Construction Classification System**

*Author's Editorial Comment: OmniClass™ has been used in COBie example files in lieu of using contract-specific or customer-specific classification systems.*

The use of classification is required for the following COBie information: Contacts, Facility, Space, Type, and Zone. The specification of specific taxonomies must be specified by contract based on regional, local, or owner specification. From the point of view of the standard the selection of a classification system is, therefore arbitrary. Examples developed for COBie files have been developed using OmniClass taxonomy. Given that OmniClass has been used in COBie examples, the following OmniClass tables are referenced in this standard:

Table 11 – Construction Entities by Function

Table 13 – Spaces by Function

Table 21 – Work Results

Table 23 – Products

#### **4.2.2.1.3 Referenced programs and projects**

##### **4.2.2.1.3.1 Specifiers' Properties information exchange (SPie)**

The objective of the SPie project is to create set of product templates that can be used by manufacturers to export product data into an open-standard format consumed by designers, specifiers, builders, owners, and operators. This project extends manufacturers' efforts in Building Information Modeling, e-marketplaces, and standard identification tagging and delivers value through the entire supply chain. For

more information on SPie please see [http://www.nibs.org/?page=bsa\\_spie](http://www.nibs.org/?page=bsa_spie) for more information on this project.

#### **4.2.2.1.3.2 Operators' Properties information exchange (OPie)**

The objective of OPie was to identify properties needed by facility operators and maintenance technicians beyond those properties identified by the SPie project. This project was initiated by the U.S. Army Corps of Engineers, Construction Engineering Research Laboratory in 2011. During this project survey forms to validate information commonly used by U.S. Army Installation Command personnel working on military installations was created and distributed to the National Institute of Building Sciences, Facility Maintenance and Operations Committee for review. An insufficient number of responses (three partial responses) were received. The project was closed due to lack of industry interest. There not being a sponsor to move this project forward, an official buildingSMART alliance® project was never initialized.

Following 2012 conversations between Dr. East and members of the Norwegian government it may be possible that information being collected on operators properties in Norway could be used to validate the information gleaned from U.S. Army sources. At the time of this standard no further information is available on the OPie project.

#### **4.2.2.1.3.3 Life-Cycle information exchange (LCie)**

LCie is a refinement of the processes required to exchange complete or partial COBie data sets. While COBie is focused on the major contractual exchanges of complete model files, LCie is focused on the individual transactions needed to build those complete sets of COBie data. First proposed in East 2010, LCie has been further developed since that time and is now the basis for the process models found in this standard. An additional NBIMS-US™ V3 standard defining the product supply chain, developed from the LCie project, is expected to be submitted. Additional information about the LCie project may be found here: [http://www.nibs.org/?page=bsa\\_lcie](http://www.nibs.org/?page=bsa_lcie).

#### **4.2.2.1.3.4 Heating, Ventilating, and Air Conditioning information exchange (HVACie)**

To model a complete building system more than COBie information is required. At the time of the COBie NBIMS-US™ V2 submission it was proposed by the author that a generic Equipment Layout information exchange (ELie) project would be sufficient for this purpose. Further investigation revealed that domain-specific versions of ELie would be needed to implement domain-specific business rules governing each system. As a result, the ELie project was depreciated in 2011 and a series of new projects was undertaken by the U.S. Army, Corps of Engineers, Engineer Research and Development Center.

A system model must describe the complete set of components that make up those systems, the assemblies of those components, such as chillers and electrical distribution boards, and the logical (systems, zones, circuits) and physical connections (feeders, piping, ductwork) that allow those systems to perform their function. The geometry of the system is also critical to understanding their function so full geometry is also required. The format for the proposed IFC MVD must be IFC. A subset of the system model that contains scheduled assets can be represented in COBie.

The first demonstration of the HVACie format was held at the buildingSMART alliance® January 2013 Challenge event. More information on this specific project may be found here: [http://www.nibs.org/?page=bsa\\_hvacie](http://www.nibs.org/?page=bsa_hvacie).

#### **4.2.2.1.3.5 Electrical System information exchange (SPARKie)**

To model a complete building system more than COBie information is required. At the time of the COBie NBIMS-US™ V2 submission it was proposed by the author that a generic Equipment Layout information

exchange (ELie) project would be sufficient for this purpose. Further investigation revealed that domain-specific versions of ELie would be needed to implement domain-specific business rules governing each system. As a result, the ELie project was depreciated in 2011 and a series of new projects was undertaken by the U.S. Army, Corps of Engineers, Engineer Research and Development Center.

A system model must describe the complete set of components that make up those systems, the assemblies of those components, such as chillers and electrical distribution boards, and the logical (systems, zones, circuits) and physical connections (feeders, piping, ductwork) that allow those systems to perform their function. The geometry of the system is also critical to understanding their function so full geometry is also required. The format for the proposed IFC MVD must be IFC. A subset of the system model that contains scheduled assets can be represented in COBie. More information on this specific project may be found here: [http://www.nibs.org/?page=bsa\\_sparkie](http://www.nibs.org/?page=bsa_sparkie).

#### **4.2.2.1.3.6 Water System information exchange (WSie)**

To model a complete building system more than COBie information is required. At the time of the COBie NBIMS-US™ V2 submission it was proposed by the author that a generic Equipment Layout information exchange (ELie) project would be sufficient for this purpose. Further investigation revealed that domain-specific versions of ELie would be needed to implement domain-specific business rules governing each system. As a result, the ELie project was depreciated in 2011 and a series of new projects was undertaken by the U.S. Army, Corps of Engineers, Engineer Research and Development Center.

A system model must describe the complete set of components that make up those systems, the assemblies of those components, such as chillers and electrical distribution boards, and the logical (systems, zones, circuits) and physical connections (feeders, piping, ductwork) that allow those systems to perform their function. The geometry of the system is also critical to understanding their function so full geometry is also required. The format for the proposed IFC MVD must be IFC. A subset of the system model that contains scheduled assets can be represented in COBie. More information on this specific project may be found here: [http://www.nibs.org/?page=bsa\\_wsie](http://www.nibs.org/?page=bsa_wsie).

#### **4.2.2.1.3.7 Building Automation Management information exchange (BAMie)**

To model a complete building system more than COBie information is required. At the time of the COBie NBIMS-US™ V2 submission it was proposed by the author that a generic Equipment Layout information exchange (ELie) project would be sufficient for this purpose. Further investigation revealed that domain-specific versions of ELie would be needed to implement domain-specific business rules governing each system. As a result, the ELie project was depreciated in 2011 and a series of new projects was undertaken by the U.S. Army, Corps of Engineers, Engineer Research and Development Center.

A system model must describe the complete set of components that make up those systems, the assemblies of those components, such as chillers and electrical distribution boards, and the logical (systems, zones, circuits) and physical connections (feeders, piping, ductwork) that allow those systems to perform their function. The geometry of the system is also critical to understanding their function so full geometry is also required. The format for the proposed IFC MVD must be IFC. A subset of the system model that contains scheduled assets can be represented in COBie. More information on this specific project may be found here: [http://www.nibs.org/?page=bsa\\_bamie](http://www.nibs.org/?page=bsa_bamie).

#### **4.2.2.1.3.8 Building Programming information exchange (BPie)**

There have been several buildingSMART and related projects aimed at the capture of architectural programming information to support both the architectural programming effort, delivery of standard Request for Proposal documents to more clearly communicate owners' requirements, and to perform the automated assessment of spatial compliance of later design documents. These projects include the

Portfolio and Asset management – Performance Requirements (PAMPeR), International Alliance for Interoperability's AR-5 Project, the buildingSMART international Room Data Sheet 'aquarium' project, the United States General Services Administration project, the buildingSMART Spatial Compliance information exchange (SCie, pronounced 'ski') project, and a new Norwegian effort whose aims are similar to the projects above. While each of these projects project have explored some aspects of the contracted information exchanges needed to create an open standard for architectural programming, none of these projects have achieved a critical mass to be recognized as national standards and be widely implemented internationally. The objective of the BPie project is to develop a consolidated international specification for the contracted delivery of architectural programming information by the evaluation and consolidation of previous project results.

Of critical concern to those attempting to use contracted information exchanges is the need to repeat the results of testing conducted under controlled conditions. This requires the precise specification of both data format and content constraints. In addition, common test models must be used in the context of automated testing programs. Finally, software companies shall be required to provide native versions of test models, configuration guides, and usage instructions to ensure repeatability by those not directly participating in these efforts.

The first demonstration of the BPie format was conducted in January 2013 at the buildingSMART alliance® Challenge. At this event BPie information was exported using the COBie SpreadsheetML format. More information on this project may be found here: [http://www.nibs.org/?page=bsa\\_bpie](http://www.nibs.org/?page=bsa_bpie).

### 4.2.3 Terms and definitions

*Author's Editorial Comment: The listing of terms and definitions was not required in NBIMS-US™ V2 submissions. The information provided here reflects the compilation of the terms and definitions found in NBIMS-US™ V2 as updated for the current COBie version. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange submissions, the information provided in this section should be considered a moderate change.*

#### 4.2.3.1 Normative terms and definitions

The following terms and definitions are the applicable terms derived from the Industry Foundation Class Model View Definition. These terms are listed in alphabetical order.

##### 4.2.3.1.1

##### **actor**

person, an organization, or person acting on behalf of an organization. A specialization of the general term object. See COBie contact.

##### 4.2.3.1.2

##### **attribute**

unit of information within an entity, defined by a particular type or reference to a particular entity. There are three kinds of attributes: direct attributes, inverse attributes and derived attributes. See COBie attribute.

##### 4.2.3.1.3

##### **building**

a building represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, story, and space). See COBie facility.

**4.2.3.1.4****building story**

The building story has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound. See COBie floor.

**4.2.3.1.5****constraints on attributes**

data type restricting the values of attributes. The most general constraint is about the existence of attribute values. There are basically two types: mandatory and optional attributes. Values of mandatory attributes must be provided whereas values of optional attributes may be omitted. For aggregation data types such as Set, List, or Array, the existence constraint is often refined by a minimal and maximal number of elements, which is also known as cardinality.

**4.2.3.1.6****derived attribute**

unit of information computed from other attributes using an expression defined in the schema.

**4.2.3.1.7****direct attribute**

scalar values or collections including Set (unordered, unique), List (ordered), or Array (ordered, sparse) as defined in [ISO 10303-11] Similar to the term "field" in common programming languages.

**4.2.3.1.8****element**

tangible physical product that can be described by its shape representation, material representations, and other properties. A specialization of the general term product. See COBie.Component.

**4.2.3.1.9****element occurrence**

element's position within the project coordinate system and its containment within the spatial structure.

**4.2.3.1.10****entity**

class of information defined by common attributes and constraints as defined in [ISO 10303-11]. Similar to the term "class" in common programming languages but describing data structure only (not behavior such as methods).

**4.2.3.1.11****enumeration**

construct that allows an attribute value to be one of multiple predefined values identified by name. Similar to the "Enumeration" construct as defined in [ISO 10303-11]. Similar in concept to "enum" in common programming languages.

**4.2.3.1.12****external reference**

link to information outside the data set, with direct relevance to the specific information the link originates from inside the data set.

**4.2.3.1.13****feature**

parametric information and additional property information modifying the shape representation of an element to which it applies.

**4.2.3.1.14****group**

collection of information that fulfils a specified purpose. A specialization of the general term object. See COBie.Zone & COBie.System.

#### **4.2.3.1.15**

##### **identification**

capability to find, retrieves, report, change, or delete specific instances without ambiguity.

#### **4.2.3.1.16**

##### **instance**

occurrence of an entity. Similar to the term "instance of a class" in object oriented programming.

#### **4.2.3.1.17**

##### **inverse attribute**

unit of information defining queries for obtaining related data and enforcing referential integrity. Similar to the term "navigation property" in entity-relational programming frameworks.

#### **4.2.3.1.18**

##### **library**

catalogue, database or holder of data, that is relevant to information in the data set. It is information referenced from an external source that is not copied into the data set.

#### **4.2.3.1.19**

##### **object**

anything perceivable or conceivable that has a distinct existence, albeit not material.

#### **4.2.3.1.20**

##### **object occurrence**

characteristics of an object as an individual. Similar to "object", "instance", "individual," "component" in other publications.

#### **4.2.3.1.21**

##### **object types**

common characteristics shared by multiple object-occurrences. Similar to "class", "template", and "type" in other publications.

#### **4.2.3.1.22**

##### **process**

object-occurrence located in time, indicating "when". See COBie job.

#### **4.2.3.1.23**

##### **process occurrence**

conceptual object that may occur at a particular time.

#### **4.2.3.1.24**

##### **process type**

common characteristics shared by multiple process occurrences.

#### **4.2.3.1.25**

##### **product**

physical or conceptual object that occurs in space. It is specialization of the general term object. See COBie.Component.

#### **4.2.3.1.26**

##### **product occurrence**

physical or conceptual object that may have a location in space and shape characteristics.



**4.2.3.1.27****product type**

common characteristics shared by multiple product occurrences. See COBie.Type.

**4.2.3.1.28****project**

encapsulation of related information for a particular purpose providing context for information contained within. Context information may include default measurement units or representation context and precision.

**4.2.3.1.29****property**

unit of information that is dynamically defined as a particular entity instance. Similar to "late-bound" or "run-time" in programming terminology. See COBie.Attribute.

**4.2.3.1.30****property occurrence**

unit of information providing a value for a property identified by name.

**4.2.3.1.31****property template**

metadata for a property including name, description, and data type. Similar in concept to "extension property" in common programming languages.

**4.2.3.1.32****property set occurrence**

unit of information containing a set of property occurrences, each having a unique name within the property set.

**4.2.3.1.33****property set template**

set of property templates serving a common purpose and having applicability to objects of a particular entity. Similar in concept to "extension class" in common programming languages.

**4.2.3.1.34****proxy**

object that does not hold a specific object type information. A specialization of object occurrence.

**4.2.3.1.35****quantity**

measurement of a scope-based metric, specifically length, area, volume, weight, count, or time.

**4.2.3.1.36****quantity occurrence**

unit of information providing a value for a quantity.

**4.2.3.1.37****quantity set**

unit of information containing a set of quantity occurrences, each having a unique name within the quantity set.

**4.2.3.1.38****relationship**

unit of information describing an interaction between items.

**4.2.3.1.39****representation**

unit of information describing how an object is displayed, such as physical shape or topology.

**4.2.3.1.40****resource**

entity with limited availability such as materials, labor, or equipment. A specialization of the general term object. The "resource definition data schemas" section is unrelated to this concept. See COBie resource.

**4.2.3.1.41****resource occurrence**

entity with inherent financial cost, which may be passed onto processes, products, and controls to which it is assigned.

**4.2.3.1.42****resource type**

common characteristics shared by multiple resource occurrences.

**4.2.3.1.43****select**

construct that allows an attribute value to be one of multiple types or entities. Similar to the "Select" construct as defined in [ISO 10303-11]. Similar to a "marker interface" in common programming languages.

**4.2.3.1.44****space**

area or volume bounded actually or theoretically. See COBie space.

**4.2.3.1.45****system**

organized combination of related parts within an AEC product, composed for a common purpose or function or to provide a service. System is essentially a functionally related aggregation of products. See COBie system.

**4.2.3.1.46****transaction**

a discrete exchange of specific a defined subset of a larger information exchange specification.

**4.2.3.1.47****type**

basic information construct derived from a primitive, an enumeration, or a select of entities. Similar to the "Type" construct as defined in [ISO 10303-11]. Similar in concept to "typedef" or "value type" in common programming languages. See COBie type.

**4.2.3.1.48****zone**

a zone is a group of spaces, partial spaces or other zones. See COBie zone.

**4.2.3.2 Informative terms and definitions**

The terms and definitions below list column headings and common terms found in the COBie spreadsheet implementation format.

#### **4.2.3.2.1 Common COBie spreadsheet terms and definitions**

There are two sets of information present in each worksheet in a COBie spreadsheet. These are author information and originating system information. Author information allows those producing a specific COBie data set to identify if they, or someone before, created or changed the information provided. Author information is mandatory on all COBie data rows. Author information is comprised of:

##### **4.2.3.2.1.1**

##### **CreatedBy**

the contact email of the person or company creating or updating a row of COBie data.

##### **4.2.3.2.1.2**

##### **CreatedOn**

the date on which the information provided in a row of COBie data was created or updated by the person or company identified in the created by field. Originating system information is information that allows COBie a row of COBie data to be referenced back to the computer software that initially generated that information. If supported by the computer software, this information allows COBie data to be synchronized with the originating computer software. Originating system information is only required if the information provided in a given row of COBie data was produced through computer software. Originating System information is comprised of:

##### **4.2.3.2.1.3**

##### **ExternalSystem**

the name of the computer system generating the row of COBie data.

##### **4.2.3.2.1.4**

##### **ExternalObject**

the name of the data object within the computer system that holds the data provided in a given row of COBie data. The default values for external object are the associated IFC entities that are mapped (and those that are excluded) when transforming IFC data to a given row of COBie data.

##### **4.2.3.2.1.5**

##### **ExternalIdentifier**

the unique identifier of the identified external object that would allow COBie data to be matched back to the data from which it was developed in the named external system.

#### **4.2.3.2.2 Specific COBie spreadsheet terms and definitions**

The following terms and definitions are provided in the order that the COBie worksheets are presented in the COBie spreadsheet form. Common author and Originating system information, described above, are excluded.

##### **4.2.3.2.2.1**

##### **contact**

A worksheet in the COBie spreadsheet. Each row in the Contact worksheet identifies a person or company referenced elsewhere in a COBie spreadsheet. Contact.Email is the primary key for this worksheet.

##### **4.2.3.2.2.2**

##### **Contact.Email**

A well-formed email address used to identify this specific Contact.

##### **4.2.3.2.2.3**

##### **Contact.Category**

the category of business in which the specific Contact is engaged. If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 34.

#### **4.2.3.2.2.4**

##### **Contact.Company**

the name of the company for the Contact.

#### **4.2.3.2.2.5**

##### **Contact.Phone**

the telephone number for the Contact.

#### **4.2.3.2.2.6**

##### **Contact.Department**

the name of the department for the Contact.

#### **4.2.3.2.2.7**

##### **Contact.OrganizationCode**

the organizational code for the Contact.

#### **4.2.3.2.2.8**

##### **Contact.GivenName**

if the Contact is a person, the given name of the Contact

#### **4.2.3.2.2.9**

##### **Contact.FamilyName**

the Contact is a person, the family name of the Contact

#### **4.2.3.2.2.10**

##### **Contact.Street**

the street address for the Contact.

#### **4.2.3.2.2.11**

##### **Contact.PostalBox**

the postal box address for the Contact.

#### **4.2.3.2.2.12**

##### **Contact.Town**

the city or town address for the Contact.

#### **4.2.3.2.2.13**

##### **Contact.StateRegion**

the state or regional address for the Contact.

#### **4.2.3.2.2.14**

##### **Contact.PostalCode**

the zip, or postal code, address for the Contact.

#### **4.2.3.2.2.15**

##### **Contact.Country**

the country for the Contact.

#### **4.2.3.2.2.16**

##### **Facility**

a worksheet in the COBie spreadsheet. There may only be one row in the facility worksheet. Projects composed of multiple facilities, shall be described by separate COBie spreadsheets. Facility.Name is the primary key for this worksheet.

**4.2.3.2.2.17****Facility.Name**

the name of the Facility as it appears on design drawings, or as otherwise specified by contract.

**4.2.3.2.2.18****Facility.Category**

the category of facility described by the COBie data set. If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 11.

**4.2.3.2.2.19****Facility.ProjectName**

the name of the Project as it appears on design drawings, or as otherwise specified by contract.

**4.2.3.2.2.20****Facility.SiteName**

the name of the Site as it appears on design drawings, or as otherwise specified by contract.

**4.2.3.2.2.21****Facility.LinearUnits**

the associated unit of measurement applied to all COBie information other than that found in the Attribute and Impact worksheets.

**4.2.3.2.2.22****Facility.AreaUnits**

the associated unit of measurement applied to all COBie information other than that found in the Attribute and Impact worksheets.

**4.2.3.2.2.23****Facility.VolumeUnits**

the associated unit of measurement applied to all COBie information other than that found in the Attribute and Impact worksheets.

**4.2.3.2.2.24****Facility.CurrencyUnit**

the associated unit of measurement applied to all COBie information other than that found in the Attribute and Impact worksheets.

**4.2.3.2.2.25****Facility.AreaMeasurement**

the associated measurement method used to calculate spatial area measurement applied to all COBie information other than that found in the Attribute worksheet.

**4.2.3.2.2.26****Facility.ExternalSiteObject**

for information originating in IFC models this information is an additional object identifier, allowing information about this Facility to be referenced back to the computer software that initially generated that information.

**4.2.3.2.2.27****Facility.ExternalSiteIdentifier**

for information originating in IFC models this information is an additional object identifier, allowing information about this Facility to be referenced back to the computer software that initially generated that information.

**4.2.3.2.2.28****Facility.ExternalFacilityObject**

for information originating in IFC models this information is an additional object identifier, allowing information about this Facility to be referenced back to the computer software that initially generated that information.

**4.2.3.2.2.29****Facility.ExternalFacilityIdentifier**

for information originating in IFC models this information is an additional object identifier, allowing information about this Facility to be referenced back to the computer software that initially generated that information.

**4.2.3.2.2.30****Facility.Description**

a general text description of the facility.

**4.2.3.2.2.31****Facility.ProjectDescription**

a general text description of the project.

**4.2.3.2.2.32****Facility.SiteDescription**

a general text description of the site.

**4.2.3.2.2.33****Facility.Phase**

the designation of the phase of the project reflected in a given COBie data set.

**4.2.3.2.2.34****Floor**

a worksheet in the COBie spreadsheet. Floors are the vertical levels of a vertical facility. Geographic areas, such as a facility's site, may also be identified on the floor worksheet. Floor.Name is the primary key for this worksheet.

**4.2.3.2.2.35****Floor.Name**

the name of the floor as it appears on other contract documents, such as design drawings.

**4.2.3.2.2.36****Floor.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Floor", "Roof", and "Site".

**4.2.3.2.2.37****Floor.Description**

a general text description of the floor.

**4.2.3.2.2.38****Floor.Elevation**

elevation at the top of the floor structure. If allowable values are not specified by contract, the default value is measured as a relative value compared to the facilities datum.

**4.2.3.2.2.39****Floor.Height**

distance between top of floor structure to bottom of structure above. Typically applicable to rows having the Floor.Category of “Floor”.

#### 4.2.3.2.2.40

##### **Space**

a worksheet in the COBie spreadsheet. Spaces are the horizontal decomposition of floors into areas which have common functional purpose and user. Vertically, spaces run from top of floor to bottom of slab above. Occupied space runs to bottom of the ceiling as expressed by Space.UsableHeight. Large Spaces which have more than a single functional purpose or user may be separated into individual spaces. Spaces may also be used on floors of type “Roof” or “Site” to identify spatial regions outside a facility enclosure. Space.Name is the primary key for this worksheet.

#### 4.2.3.2.2.41

##### **Space.Name**

the Space.Name must match the value found on design drawings at the equivalent project stage of the current deliverable.

#### 4.2.3.2.2.42

##### **Space.Category**

the category of space described by the COBie data set. If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 13.

#### 4.2.3.2.2.43

##### **Space.FloorName**

a foreign key identifying the COBie.Floor.Name.

#### 4.2.3.2.2.44

##### **Space.Description**

the description of the space found on design drawings at the equivalent project stage of the current deliverable.

#### 4.2.3.2.2.45

##### **Space.RoomTag**

signage provided for each space. Signage applied to doors, if different, is identified as Attribute records related to each applicable door. The information here is equivalent to that found in construction signage submittals.

#### 4.2.3.2.2.46

##### **Space.UsableHeight**

distance from top of finished floor to bottom of ceiling. If there is no ceiling then this value must match Floor.Height.

#### 4.2.3.2.2.47

##### **Space.GrossArea**

total space area as specified in the design contract and calculated by the identified Facility.AreaMeasurement.

#### 4.2.3.2.2.48

##### **Space.NetArea**

usable space areas as specified in the design contract and calculated by the identified Facility.AreaMeasurement

#### 4.2.3.2.2.49

##### **Zone**

a worksheet in the COBie spreadsheet. Zones are aggregations of spaces that provide some common purpose. Zone.Name, Zone.Category, and Zone.SpaceName provide the compound key for this worksheet.

#### **4.2.3.2.2.50**

##### **Zone.Name**

the name of the specific function performed by a group of spaces within a given

#### **4.2.3.2.2.51**

##### **Zone.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Circulation Zone", "Lighting Zone", "Fire Alarm Zone", "Historical Preservation Zone", "Occupancy Zone", and "Ventilation Zone".

#### **4.2.3.2.2.52**

##### **Zone.SpaceNames**

a foreign key identifying the COBie.Space.Name. The plural designation of "SpaceNames" indicating use of a delimited list of Space.Name values is depreciated in COBie version 2.4.

#### **4.2.3.2.2.53**

##### **Zone.Description**

a general text description of the zone.

#### **4.2.3.2.2.54**

##### **Type**

a worksheet in the COBie spreadsheet. Types are the different products and equipment to be installed in the facility. Type.Name is the primary key for this worksheet.

#### **4.2.3.2.2.55**

##### **Type.Name**

the Type.Name must match the value found on design drawing schedules at the equivalent project stage of the current deliverable.

#### **4.2.3.2.2.56**

##### **Type.Category Type**

the category of type described by the COBie data set. If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 23.

#### **4.2.3.2.2.57**

##### **Type.Description**

a general text description of the type. If description is present on desging drawings schedules, this value must match.

#### **4.2.3.2.2.58**

##### **Type.Asset**

type A classification of the asset within one row. If allowable values are not specified by contract, the default values are: "Fixed" and "Movable".

#### **4.2.3.2.2.59**

##### **Type.Manufacturer**

during construction and handover phase: the Contact.Email of the installed product. During planning and design phase: not applicable.

#### **4.2.3.2.2.60**

##### **Type.ModelNumber**



during construction and handover phase: the manufacture's model number of the installed product.  
During planning and design phase: not applicable.

#### 4.2.3.2.2.61

##### **Type.WarrantyGuarantorParts**

during construction and handover phase: the Contact.Email of the party, or parties, responsible for replacement parts during the warranty period. During planning and design phase: not applicable.

#### 4.2.3.2.2.62

##### **Type.WarrantyDurationParts**

during construction and handover phase: the length of the warranty period for replacement parts provided by the product manufacturer. During planning and design phase: not applicable.

#### 4.2.3.2.2.63

##### **Type.WarrantyGuarantorLabor**

during construction and handover phase: the Contact.Email of the party, or parties, responsible for labor costs during the warranty period. Typically the same as Type.WarrantyGuarantorParts. During planning and design phase: not applicable.

#### 4.2.3.2.2.64

##### **Type.WarrantyDurationLabor**

during construction and handover phase: the length of the warranty period for labor repairs provided by the product manufacturer. Typically the same as Type.WarrantyDurationParts. During planning and design phase: not applicable.

#### 4.2.3.2.2.65

##### **Type.WarrantyDurationUnit**

the identification of the unit of measure associated with values found in Type.WarrantyDurationParts and Type.WarrantyDurationLabor. If allowable values are not specified by contract, the default values are: "month" and "year".

#### 4.2.3.2.2.66

##### **Type.ReplacementCost**

during construction and handover phase: The manufacture's suggested retail price for this type. During planning and design phase: not applicable.

#### 4.2.3.2.2.67

##### **Type.ExpectedLife**

during handover phase: the expected service life of the product type given the level of service within the facility. During planning, design, and construction phase: left blank unless conducting total cost of ownership studies.

#### 4.2.3.2.2.68

##### **Type.DurationUnit**

the identification of the unit of measure associated with value found in Type.ExpectedLife. If allowable values are not specified by contract, the default values are: "month" and "year".

#### 4.2.3.2.2.69

##### **Type.Warranty**

description a general text description of the warranty.

#### 4.2.3.2.2.70

##### **Type.NominalLength**

an approximate measure of the bounding box surrounding the product type.

**4.2.3.2.2.71****Type.NominalWidth**

an approximate measure of the bounding box surrounding the product type.

**4.2.3.2.2.72****Type.NominalHeight**

an approximate measure of the bounding box surrounding the product type.

**4.2.3.2.2.73****Type.ModelReference**

during construction and handover phase: the manufacturer's catalog or reference resource, such as website, where information about the installed product type may be found. During planning and design phase: not applicable.

**4.2.3.2.2.74****Type.Shape**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.75****Type.Size**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.76****Type.Color**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.77****Type.Finish**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.78****Type.Grade**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.79****Type.Material**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.80****Type.Constituents**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturer's product data sheet.

**4.2.3.2.2.81****Type.Features**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.

**4.2.3.2.2.82****Type.AccessibilityPerformance**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.

**4.2.3.2.2.83****Type.CodePerformance**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.

**4.2.3.2.2.84****Type.SustainabilityPerformance**

during planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.

**4.2.3.2.2.85****Component**

a worksheet in the COBie spreadsheet. Components are the individual instances of the products identified by Type. Component.Name is the primary key for this table.

**4.2.3.2.2.86****Component.Name**

the Component.Name must match the value found on design drawing schedules at the equivalent project stage of the current deliverable. For equipment scheduled by Type (and not Component) naming requirements to ensure unique names for every component shall be specified by contract.

**4.2.3.2.2.87****Component.TypeName**

a foreign key identifying the COBie.Type.Name.

**4.2.3.2.2.88****Component.Space**

a foreign key identifying the COBie.Space.Name. For components not contained in a single space, the value shall refer to the space from which the equipment is most likely to be maintained.

**4.2.3.2.2.89****Component.Description**

a general text description of the component.

**4.2.3.2.2.90****Component.SerialNumber**

the serial number of the installed equipment. Component.SerialNumber must match the value found on installed equipment nameplates. During planning and design phase: not applicable.

**4.2.3.2.2.91**

**Component.InstallationDate**

the date on which the product or equipment was placed in its final location. During planning and design phase: not applicable.

**4.2.3.2.2.91****Component.WarrantyStartDate**

the date on which the product or equipment was first powered. During planning and design phase: not applicable.

**4.2.3.2.2.92****Component.TagNumber**

if a tag is required to be affixed to the product or equipment during construction, this value is the designation for this product or equipment found on the associated tag. During planning and design phase: not applicable. During planning and design phase: not applicable.

**4.2.3.2.2.93****Component.BarCode**

the bar code found on the installed product name plate. During planning and design phase: not applicable.

**4.2.3.2.2.94****Component.AssetIdentifier**

an alternative identifier for the specific component to be used as defined by associated contract.

**4.2.3.2.2.95****System**

a worksheet in the COBie spreadsheet. Systems are aggregations of components that provide some common function. System.Name, System.Category, and System.ComponentNames provide the compound key for this worksheet.

**4.2.3.2.2.96****System.Name**

the name of the specific function performed by a group of components within a given System.Category.

**4.2.3.2.2.97****System.Category**

the category of the system described by the COBie data set. If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 21.

**4.2.3.2.2.98****System.ComponentNames**

a foreign key identifying the COBie.Component.Name. The plural designation of "ComponentNames" indicating use of a delimited list of Component.Name values is depreciated in COBie version 2.4 in favor of the use of a single row for each System.Component.

**4.2.3.2.2.99****System.Description**

a general text description of the system.

**4.2.3.2.2.100****Assembly**

a worksheet in the COBie spreadsheet. Assemblies are types (or components) that contain other types or (components). Assembly.Name is the primary key for this table.

**4.2.3.2.2.101****Assembly.Name**

the name of the assembly identifies a type or component that owns or encloses other types or components.

#### **4.2.3.2.2.102**

##### **Assembly.SheetName**

specifies references in this row as pertaining to COBie data in either the “Type” or “Component” worksheet.

#### **4.2.3.2.2.103**

##### **Assembly.ParentName**

identifies the the COBie.Type.Name or COBie.Component.Name of the object enclosing Assembly.ChildNames.

#### **4.2.3.2.2.104**

##### **Assembly.ChildNames**

identifies the COBie.Type.Name or COBie.Component.Name of the objects owned or enclosed by the Assembly.ParentName.

#### **4.2.3.2.2.105**

##### **Assembly.AssemblyType**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: “Fixed”, “Optional”, “Included”, “Excluded”, “Layer”, “Patch”, and “Mix”.

#### **4.2.3.2.2.106**

##### **Assembly.Description**

a general text description of the assembly.

#### **4.2.3.2.2.107**

##### **Connection**

a worksheet in the COBie spreadsheet. Connections describe the logic connections between COBie data in either the “Type” or “Component” worksheet. Connection.Name is the primary key for this table.

#### **4.2.3.2.2.108**

##### **Connection.Name**

the name of the connection between the two parts identified a COBie data row.

#### **4.2.3.2.2.109**

##### **Connection.ConnectionType**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: “Control”, “Flow”, “Return”, “Supply”, and “Structural”.

#### **4.2.3.2.2.110**

##### **Connection.SheetName**

specifies references in in this row as pertaining to COBie data in either the “Type” or “Component” worksheet.

#### **4.2.3.2.2.111**

##### **Connection.RowName1**

a foreign key identifying the upstream COBie.Type.Name or COBie.Component.Name.

#### **4.2.3.2.2.112**

##### **Connection.RowName2**

a foreign key identifying the downstream COBie.Type.Name or COBie.Component.Name.

#### **4.2.3.2.2.113**

**Connection.RealizingElement**

the object, if appropriate, that creates the connection.

**4.2.3.2.2.114****Connection.PortName1**

the name of the output connection port on Connection.RowName1

**4.2.3.2.2.115****Connection.PortName2**

the name of the input connection port on Connection.RowName2

**4.2.3.2.2.116****Connection.Description**

a general text description of the connection

**4.2.3.2.2.117****Spare**

a worksheet in the COBie spreadsheet. Spare records identify the on-site spare parts, replacement parts, and lubricants needed to maintain the facility. Spare.Name and Spare.TypeName form the compound key for this worksheet.

**4.2.3.2.2.118****Spare.Name**

the name of the item.

**4.2.3.2.2.119****Spare.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Part", "PartSet", "Lubricant", "Other", "Spare", and "SpareSet".

**4.2.3.2.2.120****Spare.TypeName**

a foreign key identifying the COBie.Type.Name.

**4.2.3.2.2.121****Spare.Suppliers**

a foreign key identifying the COBie.Contact.Name of the companies supplying spare parts.

**4.2.3.2.2.122****Spare.Description**

a general text description of the spare.

**4.2.3.2.2.123****Spare.SetNumber**

a general text description, if applicable, identifying the manufacturer's spare part set number.

**4.2.3.2.2.124****Spare.PartNumber**

a general text description, if applicable, identifying the manufacturer's spare part number.

**4.2.3.2.2.125****Resource**

a worksheet in the COBie spreadsheet. Resource records identify the tools, materials, and training needed to maintain the facility. Resource.Name is the primary key for this table.

**4.2.3.2.2.126****Resource.Name**

the name of the resource that is required to be used on the project. This resource may be shared across multiple Jobs.

**4.2.3.2.2.127****Resource.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Labor", "Material", "Tools", and "Training".

**4.2.3.2.2.128****Resource.Description**

a general text description of the resource.

**4.2.3.2.2.129****Job**

a worksheet in the COBie spreadsheet. Job records identify the variety of work that is required to operate, maintain, startup, shutdown, troubleshoot a given COBie.Type. Job.Name, Job.TypeName, and Job.TaskNumber create the compound key for this worksheet.

**4.2.3.2.2.130****Job.Name**

the name of the job

**4.2.3.2.2.131****Job.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Adjustment", "Calibration", "Emergency", "Inspection", "Operation", "PM", "Safety", "ShutDown", "StartUp", "Testing", and "Trouble".

**4.2.3.2.2.132****Job.Status**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Not Yet Started", "Started", and "Completed".

**4.2.3.2.2.133****Job.TypeName**

a foreign key identifying the COBie.Type.Name.

**4.2.3.2.2.134****Job.Description**

a general text description of the job. There are typically three uses of this field. First, the Job.Description may contain a general description of the job, with all steps outlined in an associated document. Second, the Job.Description may contain the complete set of all numbered steps. To assist CMMS/CAFM vendors these steps should be delimited with a semicolon. Finally the Job.Description may contain one of several linked steps using the Job.TaskNumber and Job.Priors.

**4.2.3.2.2.135****Job.Duration**

length of time required to perform the job.

**4.2.3.2.2.136****Job.DurationUnit**

unit of time associated with Job.Duration.

**4.2.3.2.2.137****Job.Start**

the length of time, from the delivery of the COBie data set when the next occurrence of the Job should take place.

**4.2.3.2.2.138****Job.TaskStartUnit**

the unit of time associated with Job.Start.

**4.2.3.2.2.139****Job.Frequency**

the length of time between each service period

**4.2.3.2.2.140****Job.FrequencyUnit**

the unit of time associated with Job.Frequency.

**4.2.3.2.2.141****Job.TaskNumber**

if Job.Description contains a series of individual operations, this is the identification nonzero integers used to reference each step. Job.TaskNumber becomes the third part of the compound key, if Job.Description contains a series of individual operations; otherwise it is ignored. The first Job.Description in the series shall have a value of "0" and provide the general information about the job whose tasks follow.

**4.2.3.2.2.142****Job.Priors**

if Job.Description contains a series of individual operations, this is a comma delimited list of the Job.TaskNumber for all prior jobs. The first Job.Description in the series shall have a value of "0".

**4.2.3.2.2.143****Job.ResourceNames**

a comma delimited list of foreign keys containing Resource.Name values.

**4.2.3.2.2.144****Impact**

a worksheet in the COBie spreadsheet. Impact records identify the health, safety, economic, or environmental impacts related to the assets found in the facility. Impact.Name is the primary key of this worksheet.

**4.2.3.2.2.145****Impact.Name**

the name of the impact.

**4.2.3.2.2.146****Impact.ImpactType**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Cost", "ClimateChange", "PrimaryEnergyConsumption"

**4.2.3.2.2.147****Impact.ImpactStage**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Production", "Installation", "Maintenance", "Replacement", "Use", and "Reuse".

**4.2.3.2.2.148**



**Impact.ImpactStage**

a classification of the life cycle stage of information contained within one row. If allowable values are not specified by contract, the default values are: "Production", "Installation", "Maintenance", "Replacement", "Use", and "Reuse".

**4.2.3.2.2.149****Impact.SheetName**

specifies a required reference to COBie data in any other COBie worksheet.

**4.2.3.2.2.150****Impact.RowName**

a foreign key identifying the row in the associated Impact.SheetName.

**4.2.3.2.2.151****Impact.Value**

a quantification of the impact described in a given row.

**4.2.3.2.2.152****Impact.ImpactUnit**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "currency", "kgCO2e", "MJ", "other", or "n/a".

**4.2.3.2.2.153****Impact.LeadInTime**

a description of the length of time before the impact is expected.

**4.2.3.2.2.154****Impact.Duration**

a description of the length of time over which the impact occurs.

**4.2.3.2.2.155****Impact.LeadOutTime**

a description of the length of time after the impact before the cycle resumes.

**4.2.3.2.2.156****Impact.Description**

a general text description of the impact.

**4.2.3.2.2.157****Document**

a worksheet in the COBie spreadsheet. Document records identify external files that provide information associated with data in a given COBie deliverable. Document.Name, Document.SheetName, Document.RowName, Document.Stage comprise the primary key for this worksheet.

**4.2.3.2.2.158****Document.Name**

the name of the document.

**4.2.3.2.2.159****Document.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Preconstruction Submittals", "Shop Drawings", "Product Data", "Samples", "Design Data", "Test Reports", "Certificates", "Manufacturer Instructions", "Manufacturer Field Reports", "Operation and Maintenance", "Closeout Submittals", "Contract Drawings", "Design Review

Comment”, “Specifications”, “Request for Information”, “Client Requirements”, “Contract Specifications”, “Contract Drawings”, “Requests for Information”, “Contract Modifications”, and “Punch List Items”.

#### **4.2.3.2.2.160**

##### **Document.ApprovalBy**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: “Owner Approval”, “Contractor Certified”, and “Information Only”/

#### **4.2.3.2.2.161**

##### **Document.Stage**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: “As Built”, “Submitted”, “Approved”, “Exact Requirement”, “Maximum Requirement”, “Minimum Requirement”, and “Requirement”.

#### **4.2.3.2.2.162**

##### **Document.SheetName**

specifies references in in this row as pertaining to COBie data in any other COBie worksheet.

#### **4.2.3.2.2.163**

##### **Document.RowName**

a foreign key identifying the row in the associated Document.SheetName.

#### **4.2.3.2.2.164**

##### **Document.Directory**

the path name to the file. If COBie is delivered on a portable media the directory should be the relative path name from the COBie file to the referenced document. If the COBie file references documents that are available on the World Wide Web, the complete path name to the document including the internet protocol used shall be included. In all cases the document directory should include the trailing “/” of all path names.

#### **4.2.3.2.2.165**

##### **Document.File**

the name of the file, with file extension, that contains the associated information.

#### **4.2.3.2.2.166**

##### **Document.Description**

a general text description of the description.

#### **4.2.3.2.2.167**

##### **Document.Reference**

if different from the Document.Directory/Document.File this is a reference to documents provided from manufacturers’ catalogs or websites

#### **4.2.3.2.2.168**

##### **Attribute**

a worksheet in the COBie spreadsheet. Attribute records identify external files that provide information associated with data in a given COBie deliverable. Attribute.Name, Attribute.SheetName, Attribute.RowName, Attribute.Category comprises the compound key for this worksheet.

#### **4.2.3.2.2.169**

##### **Attribute.Name**

the name of the attribute.

#### **4.2.3.2.2.170**

##### **Attribute.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: “As Built”, “Submitted”, “Approved”, “Exact Requirement”, “Maximum Requirement”, “Minimum Requirement”, and “Requirement”.

#### **4.2.3.2.2.171**

##### **Attribute.SheetName**

specifies references in in this row as pertaining to COBie data in any other COBie worksheet.

#### **4.2.3.2.2.172**

##### **Attribute.Row Name**

a foreign key identifying the row in the associated Attribute.SheetName.

#### **4.2.3.2.2.173**

##### **Attribute.Value**

the value of the attribute.

#### **4.2.3.2.2.174**

##### **Attribute.Unit**

the unit defining the value of the attribute.

#### **4.2.3.2.2.175**

##### **Attribute.Description**

a general text description of the attribute.

#### **4.2.3.2.2.176**

##### **Attribute.AllowedValues**

a comma delimited list of one or more allowed values.

#### **4.2.3.2.2.177**

##### **Coordinate**

a worksheet in the COBie spreadsheet. Coordinate records identify simple geometric information associated with data in a given COBie deliverable. Coordinate.Name is the primary key for this worksheet.

#### **4.2.3.2.2.178**

##### **Coordinate.Name**

the name of the coordinate.

#### **4.2.3.2.2.179**

##### **Coordinate.Category**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: “point”, “lineendone”, “lineend-two”, “box-lowerleft”, and “box-upperright”.

#### **4.2.3.2.2.180**

##### **Coordinate.SheetName**

specifies references in in this row as pertaining to COBie data in a COBie.Facility, COBie.Floor, COBie.Space, COBie.Type or COBie.Coordinate worksheet.

#### **4.2.3.2.2.181**

##### **Coordinate.RowName**

a foreign key identifying a specific row in the Coordinate.SheetName.

#### **4.2.3.2.2.182**

##### **Coordinate.CoordinateXAxis**

relative coordinate of the referenced object.

**4.2.3.2.2.183****Coordinate.CoordinateYAxis**

relative coordinate of the referenced object.

**4.2.3.2.2.184****Coordinate.CoordinateZAxis**

relative coordinate of the referenced object.

**4.2.3.2.2.185****Coordinate.ClockwiseRotation**

rotation of the referenced object around the identified point, if applicable.

**4.2.3.2.2.186****Coordinate.ElevationalRotation**

rotation of the referenced object around the identified point, if applicable.

**4.2.3.2.2.187****Coordinate.YawRotation –**

rotation of the referenced object around the identified point, if applicable.

**4.2.3.2.2.188****Issue**

a worksheet in the COBie spreadsheet. Issue records identify allow the exchange of business process and exception reporting information related to other parts of a given COBie deliverable. Issue.Name, Issue.SheetName, Issue.RowName comprise the compound key for this worksheet.

**4.2.3.2.2.189****Issue.Name**

the name of the coordinate.

**4.2.3.2.2.190****Issue.Type**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Change", "Claim", "Coordination", "Environmental", "Function", "IndoorAirQuality", "Installation", "RFI", "Safety", and "Specification".

**4.2.3.2.2.191****Issue.Risk**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Very High", "High", "Moderate", "Low", and "Unknown".

**4.2.3.2.2.192****Issue.Chance**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Has Occurred", "High", "Moderate", "Low", and "Unknown".

**4.2.3.2.2.193****Issue.Impact**

a classification of the type of information contained within one row. If allowable values are not specified by contract, the default values are: "Very High", "High", "Moderate", "Low", and "Unknown".

**4.2.3.2.2.194****Issue.SheetName1**

specifies references in in this row as pertaining to COBie data in any other COBie worksheet.

**4.2.3.2.2.195****Issue.RowName1**

a foreign key identifying a specific row in the Issue. SheetName1.

**4.2.3.2.2.196****Issue.SheetName2**

specifies references in in this row as pertaining to COBie data in any other COBie worksheet.

**4.2.3.2.2.197****Issue.RowName2**

a foreign key identifying a specific row in the Issue. SheetName2.

**4.2.3.2.2.198****Issue.Description**

a general text description of the issue.

**4.2.3.2.2.199****Issue.Owner**

a foreign key identifying the COBie.Contact.

**4.2.3.2.2.200****Issue.Mitigation**

a general text description of the mitigation.

**4.2.4 Symbols and abbreviated terms**

*Author's Editorial Comment: The listing of symbols and abbreviated terms was not required in NBIMS-US™ V2 submissions. The information provided here reflects the compilation of the terms and definitions found in NBIMS-US™ V2 as updated for the current COBie version. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange submissions, the information provided in this section should be considered a moderate change.*

**4.2.4.1****AEC**

architecture, engineering, and construction

**4.2.4.2****BAMie**

building automation management information exchange format

**4.2.4.3****BIM**

building information modelling

**4.2.4.4****bSa**

buildingSMART alliance®

**4.2.4.5****bSi**

buildingSMART International

**4.2.4.6**

**COBie**

construction-operations building information exchange format

**4.2.4.7****COBieLite**

NIEM compliant COBie XML schema, files conforming to that schema

**4.2.4.8****FM**

facility management

**4.2.4.9****GUID**

globally unique identifier

**4.2.4.10****IFC**

industry foundation classes

**4.2.4.11****ifcXML**

an XML schema for IFC Step files based on STEP XML transformation rules

**4.2.4.12****IFD**

international framework for dictionaries

**4.2.4.13****HVACie**

heating, ventilating, and air conditioning information exchange format

**4.2.4.14****LCie**

life-cycle information exchange format

**4.2.4.15****MVD**

model view definition

**4.2.4.16****NBIMS-US™**

National BIM Standard-United States®

**4.2.4.17****NIBS**

National Institute of Building Sciences

**4.2.4.18****NIEM**

national information exchange model

**4.2.4.19****Sparkie**

electrical system information exchange format

**4.2.4.20****SPF**

STEP physical file

**4.2.4.21****SpreadsheetML**

an XML schema for Microsoft Excel spreadsheet 2003

**4.2.4.22****STEP**

standard for the exchange of product data

**4.2.4.23****URI**

uniform resource identifier

**4.2.4.24****US**

United States of America

**4.2.4.25****UUID**

universally unique identifier

**4.2.4.26****WSie**

water system information exchange format

**4.2.5 Business process documentation**

Author's Editorial Comment: Business processes supporting COBie were first documented in a U.S. Army Technical Report (East 2007). These business models were developed from a series of industry advisory panel meetings conducted in 2006 and 2007. Additional clarification of the scope of processes pertaining to the life-cycle of facility assets were proposed in East 2010. At the time of submission of this standard another update to COBie-related business processes is being finalized for editing as part of a U.S. Army Technical Report (Fallon 2013). The objective of the research contained in this report was to use business process modeling in conjunction with “lean” management approaches, as in East 2011, to compare current with COBie-based business processes. As part of the NBIMS-US™ V3 updated COBie standard, an updated set business models developed in the latest report is provided for direct inclusion in the standard. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange standards, the information provided in this section should be considered a moderate\_change.

**4.2.5.1 Process models provided****4.2.5.1.1 Business process list**

- |                                   |   |
|-----------------------------------|---|
| • Study and Define Needs          | • Develop Program – Product Program                           |
| • Develop Design Criteria         | • Prepare Invitation to Bid and Receive Proposal (Pre-Design) |
| • Study Technical Feasibility     | • Explore Concepts – Design Early                             |
| • Communicate Results Decision    | • Develop Design – Design Schematic                           |
| • Develop Program – Space Program |   |

- Develop Design – Design Coordinated
- Finalize Design – Design Final
- Prepare Invitation to Bid and Receive Proposals (Post Design)
- Respond to Pre-Proposal Inquiries
- Develop Pre-Construction Plan
- Identify Discrepancies
- Prepare Submittal Information – Product Type Selection
- Prepare Submittal Information – System Layout
- Organize Submittal Information
- Perform Submittal Review – Submittal Issue
- Provide Resources
- Execute Construction Activities
- Perform Equipment Testing
- Inspect and Approve Work
- Define, Record and Certify Discrepancies
- Closeout

#### **4.2.5.1.2 Business process descriptions**

##### **4.2.5.1.2.1 Study and define needs**

Definition: Standard facility information must be available in order to determine the basic requirements for a potential project. The Owner identifies the need and either develops technical criteria for the facility if none exist or utilizes existing technical criteria if available. If it does exist, this information must be checked for relevancy every five years to remain consistent with overall needs.

##### **4.2.5.1.2.2 Develop design criteria**

Specification information for equipment based on facility criteria is generated early in the planning process by the Owner. This information must be checked for relevancy every five years to remain consistent with overall needs.

##### **4.2.5.1.2.3 Study technical feasibility**

Definition: The Feasibility Study allows the Owner to evaluate different options (typically three) based on the identified requirements before finalizing specific information about a project. The Architect or Planner develops the study based on the information contained in the Facility Criteria and Discipline Specification information exchanges.

##### **4.2.5.1.2.4 Communicate results decisions**

Initial criteria about a project must be established in order to evaluate the project feasibility. The Owner evaluates the Facility Criteria, Discipline Specifications, and Feasibility Study to determine whether or not to move forward with the project.

##### **4.2.5.1.2.5 Develop program – space program**

Once the Project Definition has been established and approved, further development of the project requirements can occur. The Architect or Planner evaluates information contained in the Project Definition information exchange to identify space needs based on the facility type. Space requirements, based on facility type, are located online in electronic document format and must be downloaded. These



documents are typically printed by the end user. If no standard facility space criteria exist, it must be created by referencing similar facility types.

#### **4.2.5.1.2.6 Develop program – product program**

The Architect or Planner evaluates information contained in the Project Definition information exchange to identify product needs based on the facility type. Requirements for products based on facility type are located online in electronic document format and must be downloaded. These documents are typically printed by the end user. If no standard facility product criteria exist, it must be created by referencing similar facility types.

#### **4.2.5.1.2.7 Prepare invitation to bid and receive proposals (pre-design)**

Once the major criteria have been determined, the Owner's Representative prepares and distributes a Request for Proposal (RFP).

#### **4.2.5.1.2.8 Explore concepts – design early**

The Architect utilizes the specific information produced during pre-design to develop a solution that reflects the requirements stated in the Project Definition, Space Program, and Product Program. Currently, the owner requires 6 hard copies to be submitted for each review cycle. The Architect performs a QA/QC check before submitting to the Owner's Representative. After receiving the submission, the Owner's Representative validates the documents (reviews) and provides comments to the Architect. The Architect and Consultants are then required to update the documents based on the comments. After revisions are made, the Architect resubmits.

#### **4.2.5.1.2.9 Develop design – design schematic**

The Architect further develops the approved Design Early deliverable documents to produce the Design Schematic documents. Currently, the owner requires 6 hard copies to be submitted for each review cycle. The Architect performs a QA/QC check before submitting to the Owner's Representative. After receiving the submission, the Owner's Representative validates the documents and provides comments to the Architect. The Architect and Consultants are then required to update the documents based on the comments. After revisions are made, the Architect resubmits.

#### **4.2.5.1.2.10 Develop design – product type template**

As the design progresses, performance characteristics and suitable products for the building systems are identified. System types and equipment are identified by the Specifier based on the facility requirements. Six copies are required to be submitted for review. Some products are defined in more detail by identifying manufacturers and model numbers which meet requirements (Basis of Design). In these cases, 3 qualifying products should be listed

#### **4.2.5.1.2.11 Develop design – design coordinated**

The Architect further develops the approved Design Schematic deliverable documents to produce the Design Coordinated documents. In addition, the building systems are coordinated to eliminate spatial interferences. This is the major coordination submittal before the final delivery package. The owner requires 6 hard copies to be submitted for each review cycle. Due to the higher level of coordination and increase in number of interested reviewing parties, more copies are sometimes needed. The Architect performs a QA/QC check before submitting to the Owner's Representative. After receiving the submission, the Owner's Representative validates the documents and provides comments to the

Architect. The Architect and Consultants are then required to update the documents based on the comments. After revisions are made, the Architect resubmits.

#### **4.2.5.1.2.12 Develop design – product type candidate**

The performance requirements of building systems and equipment are further refined and documented by the Specifier during this phase. Any equipment, products, or systems not selected previously are identified. Specific manufacturers and model numbers are noted. Three qualifying products are identified. Six copies are submitted for each review cycle.

#### **4.2.5.1.2.13 Finalize design – design final**

The Design Final package is the final set of contract documents ready for bid solicitation by the Owner. This final design deliverable does not require another review by the Owner's Representative.

#### **4.2.5.1.2.14 Finalize design – product type candidate**

At this phase of the project all equipment and system types must be identified by the Specifier. Product information from the Design Coordinated phase is incorporated into to this phase. Three (3) qualifying products for each type required must be listed.

#### **4.2.5.1.2.15 Prepare invitation to bid and receive proposals (post-design)**

Once the design is complete, the Owner packages the Design Final Documents information together with other owner-supplied information (e.g., contractual terms) and creates a Request for Proposals (RFP) Package. This becomes the official bid set.

#### **4.2.5.1.2.16 Respond to pre-proposal inquiries**

Before finalizing a bid proposal, the Contractor typically requests additional information or clarification of some bid documents.

#### **4.2.5.1.2.17 Develop pre-construction plan**

The Contractor is required to develop a Pre-Construction Plan that describes how the Contractor will make provisions for managing the construction of the facility. This is sent as a submittal package. Refer to the Submittal Package exchange for detailed requirements related to transmitting and handling Pre-Construction Plan submittals.

#### **4.2.5.1.2.18 Identify discrepancies**

The Contractor submits a Request for Information (RFI) to ask for clarification during the construction process. These questions may be due to but not restricted to ambiguities or contradictions in the drawings or to site conditions.

#### **4.2.5.1.2.19 Prepare submittal information – product type selection**

The Contractor and Sub-Contractors gather information for products identified in the Design Final documents and prepare submittals. Refer to the Submittal Package exchange for detailed requirements related to transmitting and handling Product Type Selection submittals.

**4.2.5.1.2.20 Prepare submittal information – system layout**

The Contractor and Sub-Contractors review information for products identified in the Design Final documents and prepare shop drawings. Refer to the Submittal Package exchange for detailed requirements related to transmitting and handling System Layout submittals.

**4.2.5.1.2.21 Organize submittal information**

The Contractor organizes the required submittal information and creates Submittal Packages to be reviewed by the Owner's Representative and/or Architect. Six hard copies are required to be submitted for review.

**4.2.5.1.2.22 Perform submittal review – submittal issue**

The Architect and/or Sub-Consultants validate the submittals provided by the Contractor and provide comments. Six hard copies are required.

**4.2.5.1.2.23 Provide resources**

The Contractor contacts a Supplier to order equipment and materials. The Supplier then provides a price quote to the Contractor for the equipment and/or materials. The Contractor verifies the specifications of the equipment and/or materials in the quote against approved submittal documentation and then submits them to the Owner's Representative and/or Architect for approval.

**4.2.5.1.2.24 Execute construction activities**

The Contractor installs the building equipment, materials, and systems using the design final drawings, approved shop drawings, product data, and manufacturer's instructions.

**4.2.5.1.2.25 Perform equipment testing**

After the Contractor completes the installation process, the equipment/systems must be tested by activating the equipment. This testing must be completed with the Owner's Representative and Manufacturer's representative present.

**4.2.5.1.2.26 Inspect and approve work**

When the Contractor has completed installation of equipment or systems, a notification is sent to the Architect indicating the installed item is ready for inspection/observation. The Architect conducts regular inspections of the installed construction work. The findings of the inspections including any deficiencies with the installation of the construction work are documented in a report. If deficiencies are identified in the inspection report, the Contractor corrects them and then requests a re-inspection.

**4.2.5.1.2.27 Define, record and certify discrepancies**

The Architect creates a final punch-list based upon a survey of the completed construction work. The Contractor corrects the deficiencies identified in the punch-list. The Architect verifies that the Contractor has corrected the deficiencies in the punch-list by performing a final walkthrough.





**4.2.5.1.2.28 Closeout**

The Contractor gathers all as-built information related to the project and forwards the information to the Owner. Four copies are typically required.

#### 4.2.5.1.3 Business process model diagrams

Business process models were created at a lower-level of detail than is typically required. This additional level of detail was included to demonstrate the value of data-centric information exchange, versus document-centric exchanges typical of practice in the design and construction industry today. This allows the models themselves to directly address the question of business case since, with data-centric exchanges; all non-value added tasks required for working with documents may be eliminated. Each business process diagram begins with the definition of the Current Process workflow. On top of that current process color codes that describe where tasks were eliminated, automated, or streamlined are shown in the COBie-based Expected Process.

When using these business process models, please consult the following legend.

- Eliminated Tasks 
- Automated Tasks 
- Streamlined Tasks 
- Unchanged Tasks from Current LCie Process 

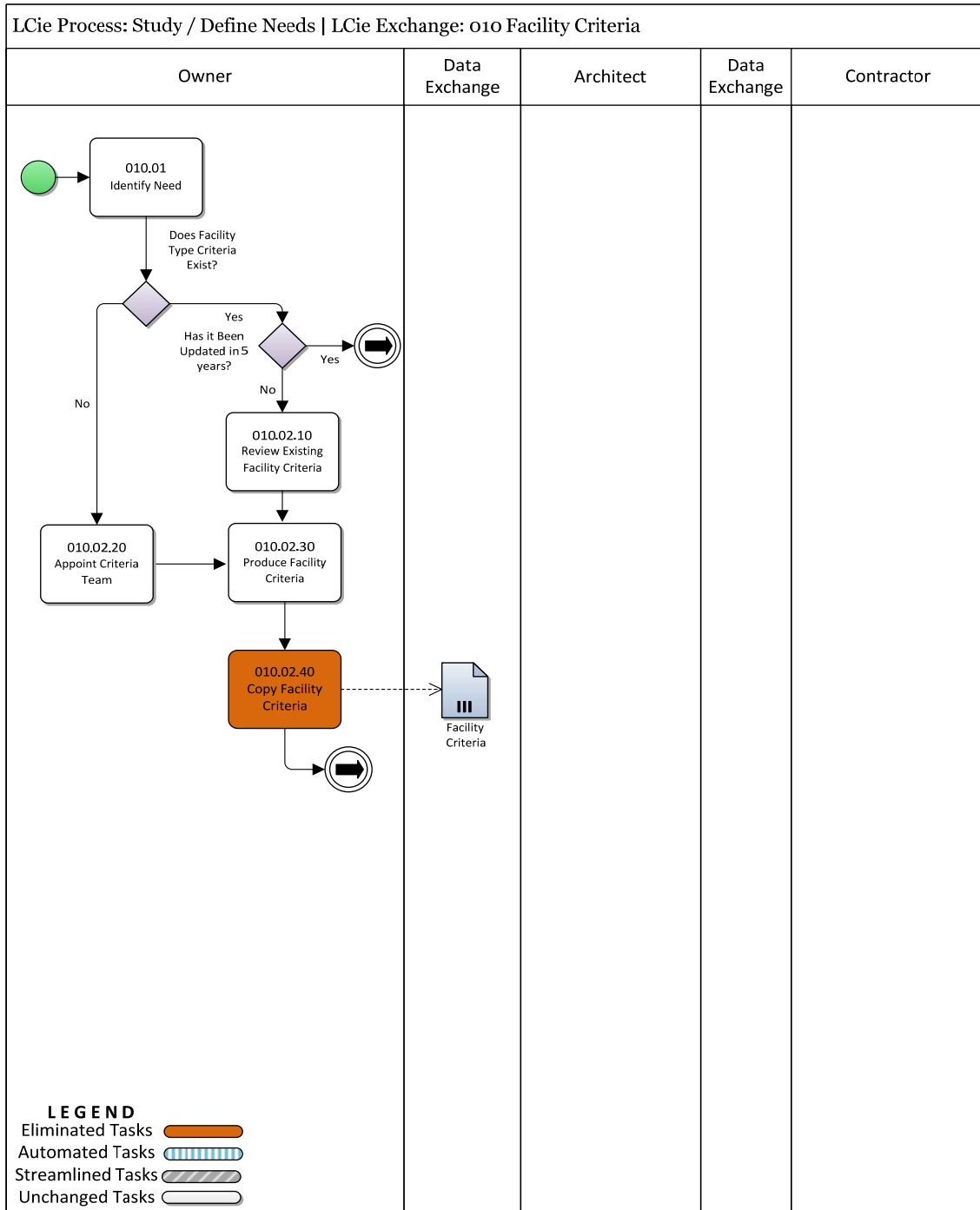


Figure 1 Study/Define Needs Process Model

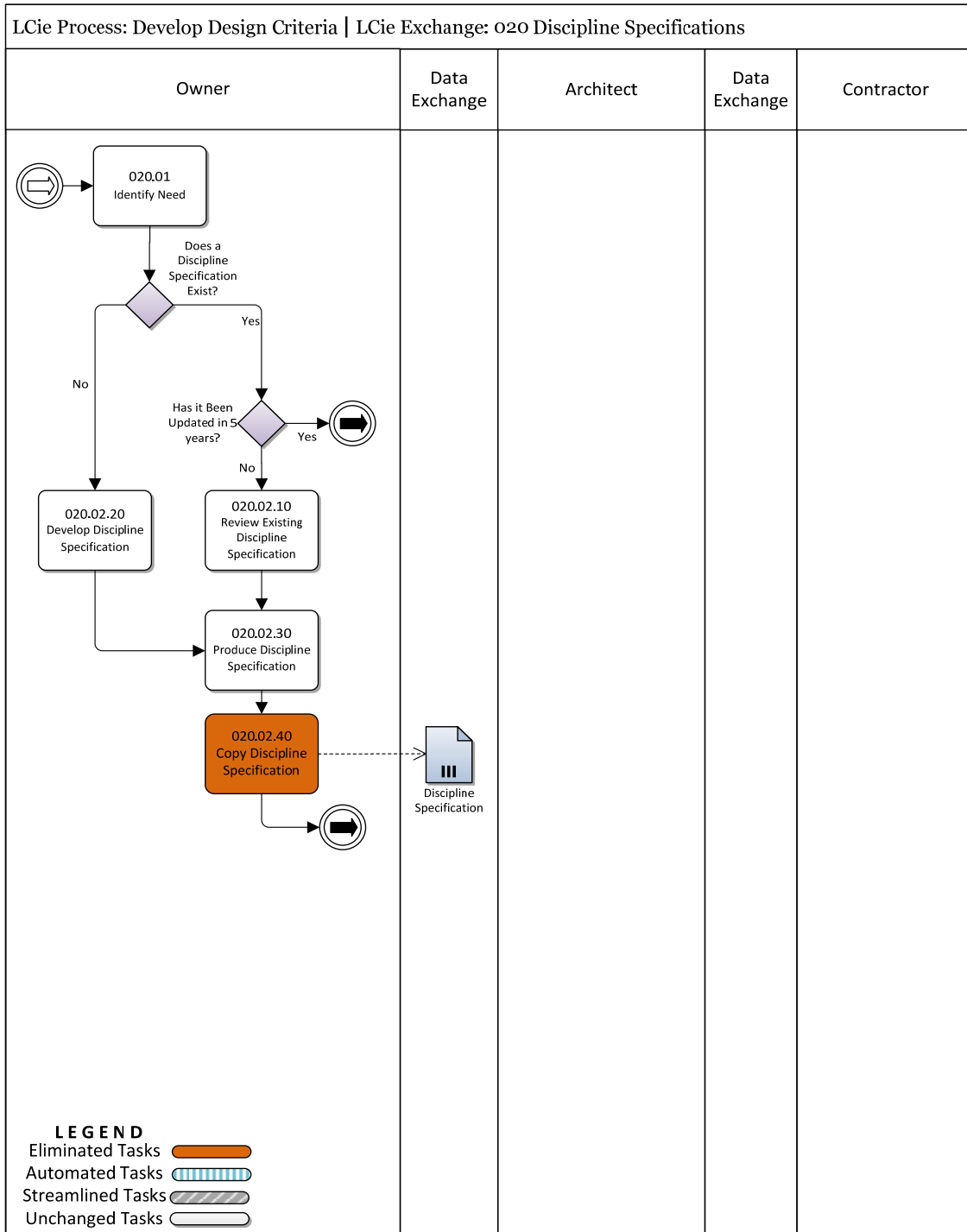


Figure 2 Develop Design Criteria Process Model

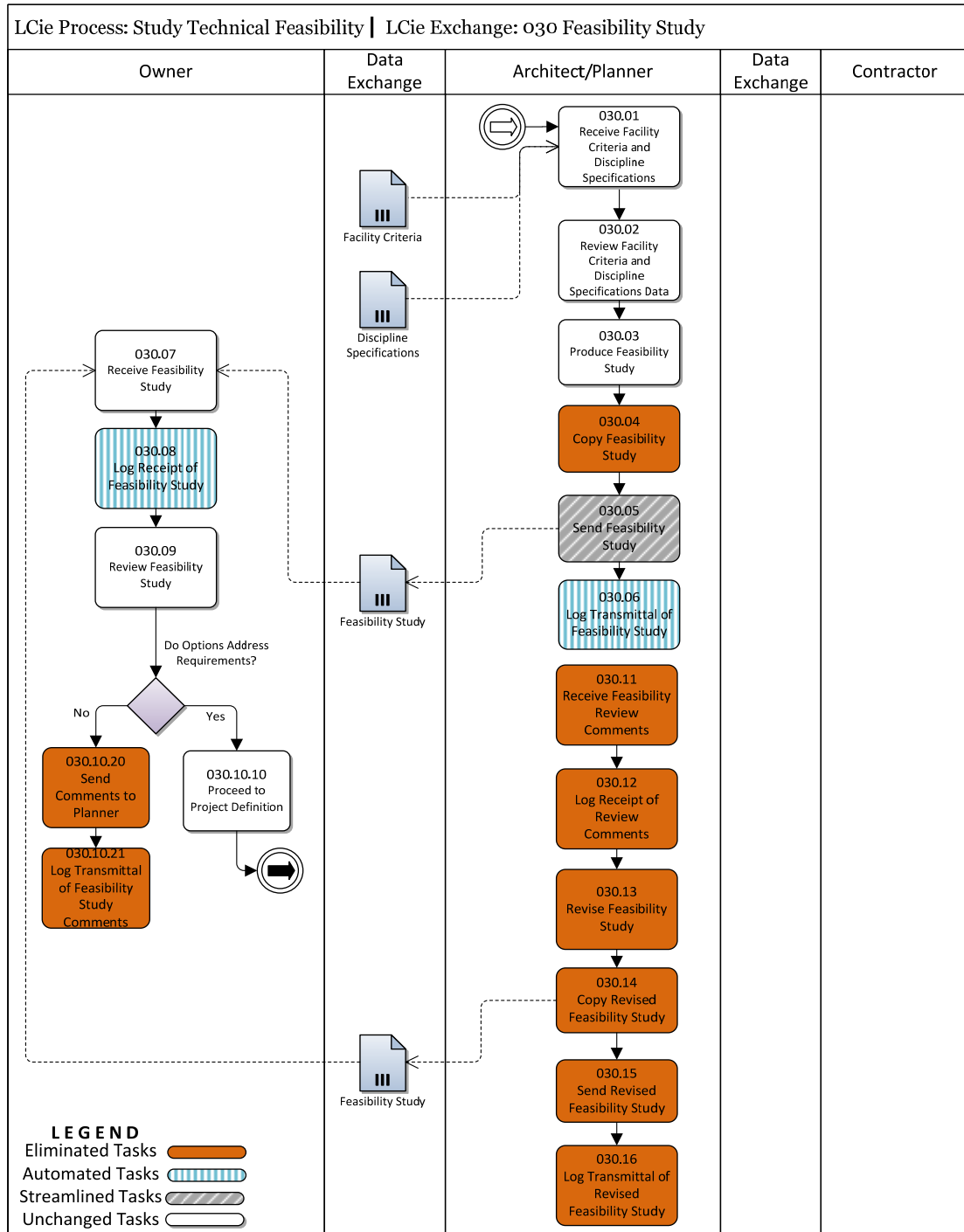


Figure 3 Study Technical Feasibility Process Model

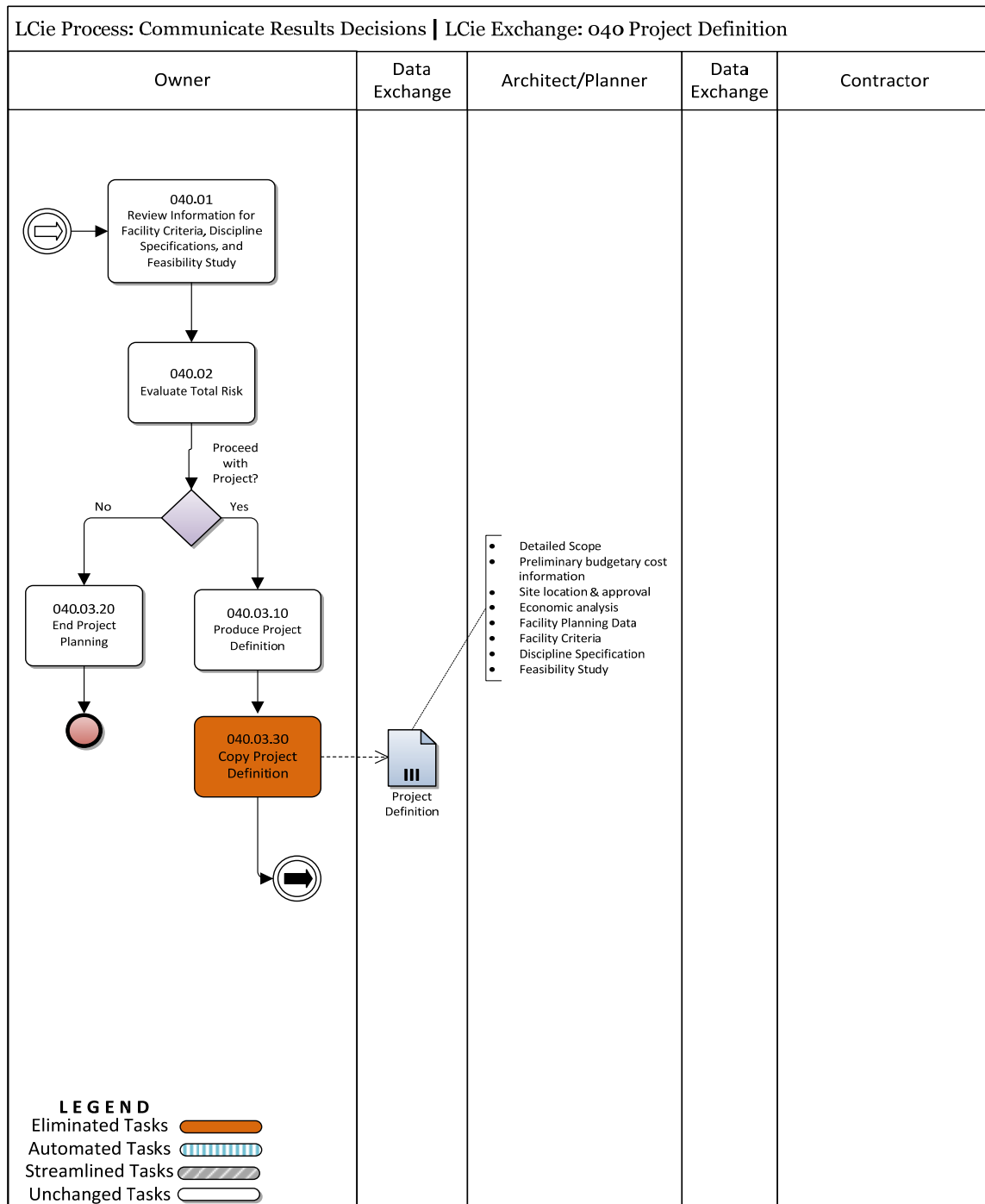


Figure 4 Communicate Results Decisions Process Model



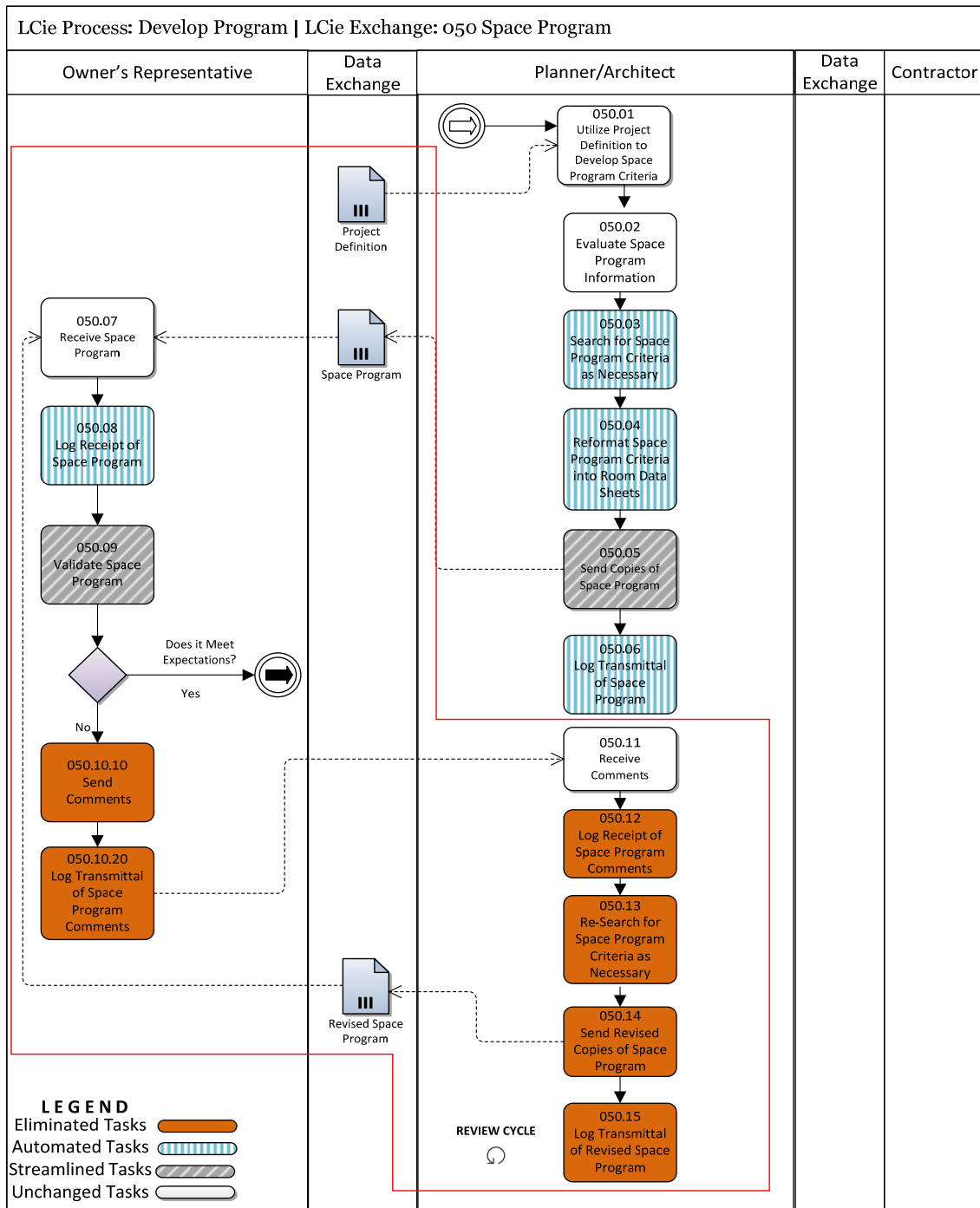


Figure 5 Develop Program - Space Program Process Model

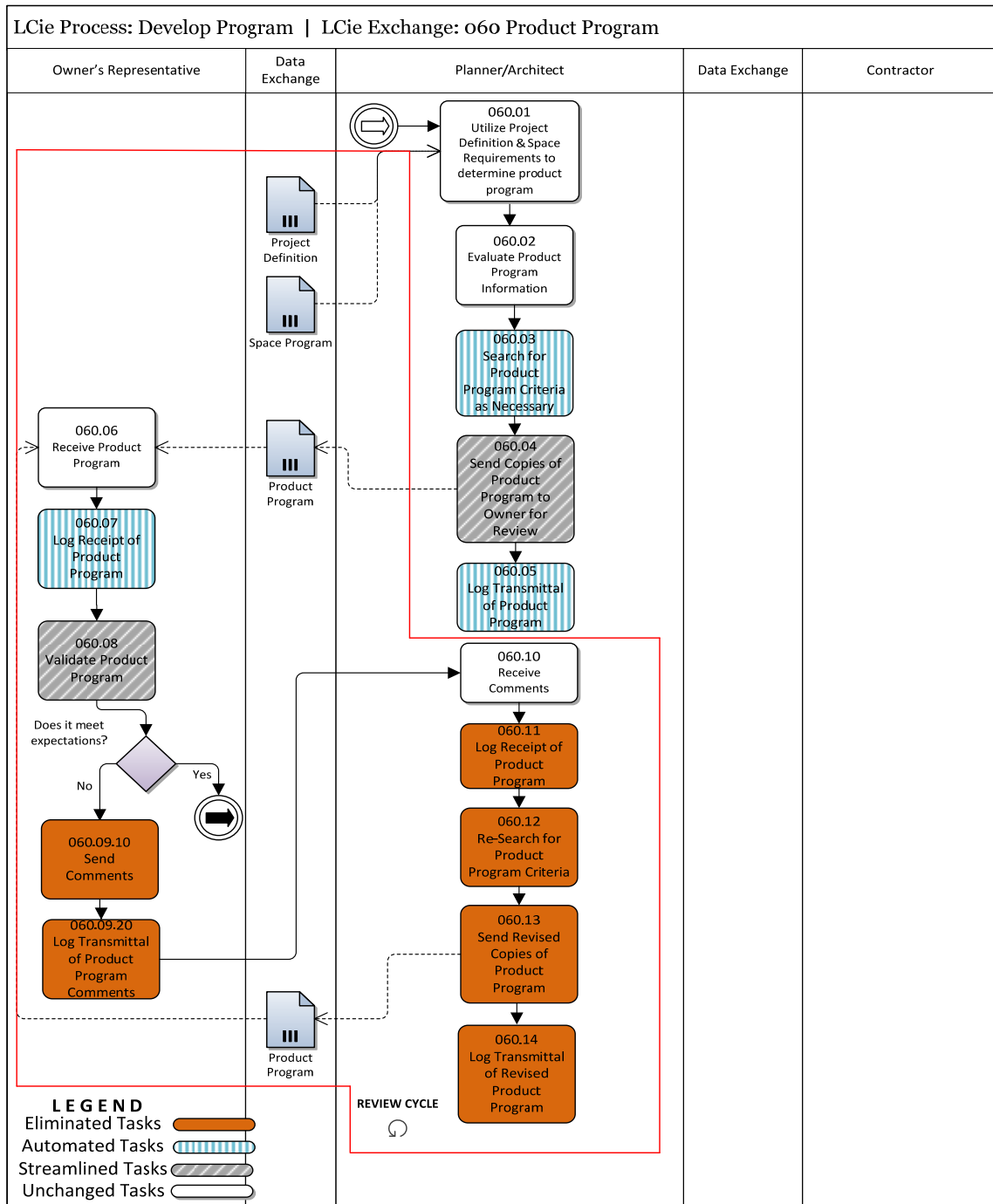


Figure 6 Develop Program - Product Program Process Model

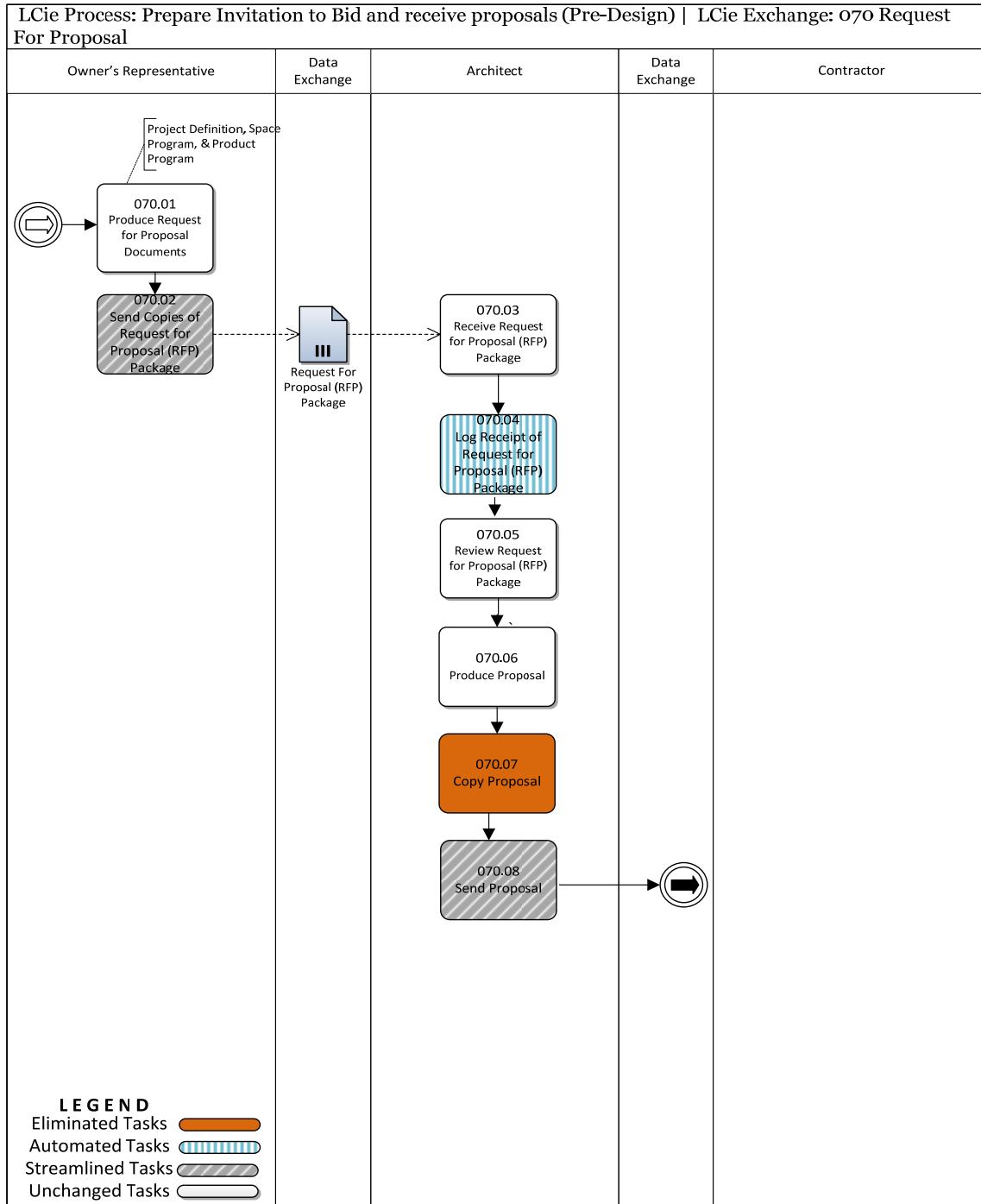


Figure 7 Prepare Invitation to Bid and Receive Proposals (Pre-Design) Process Model

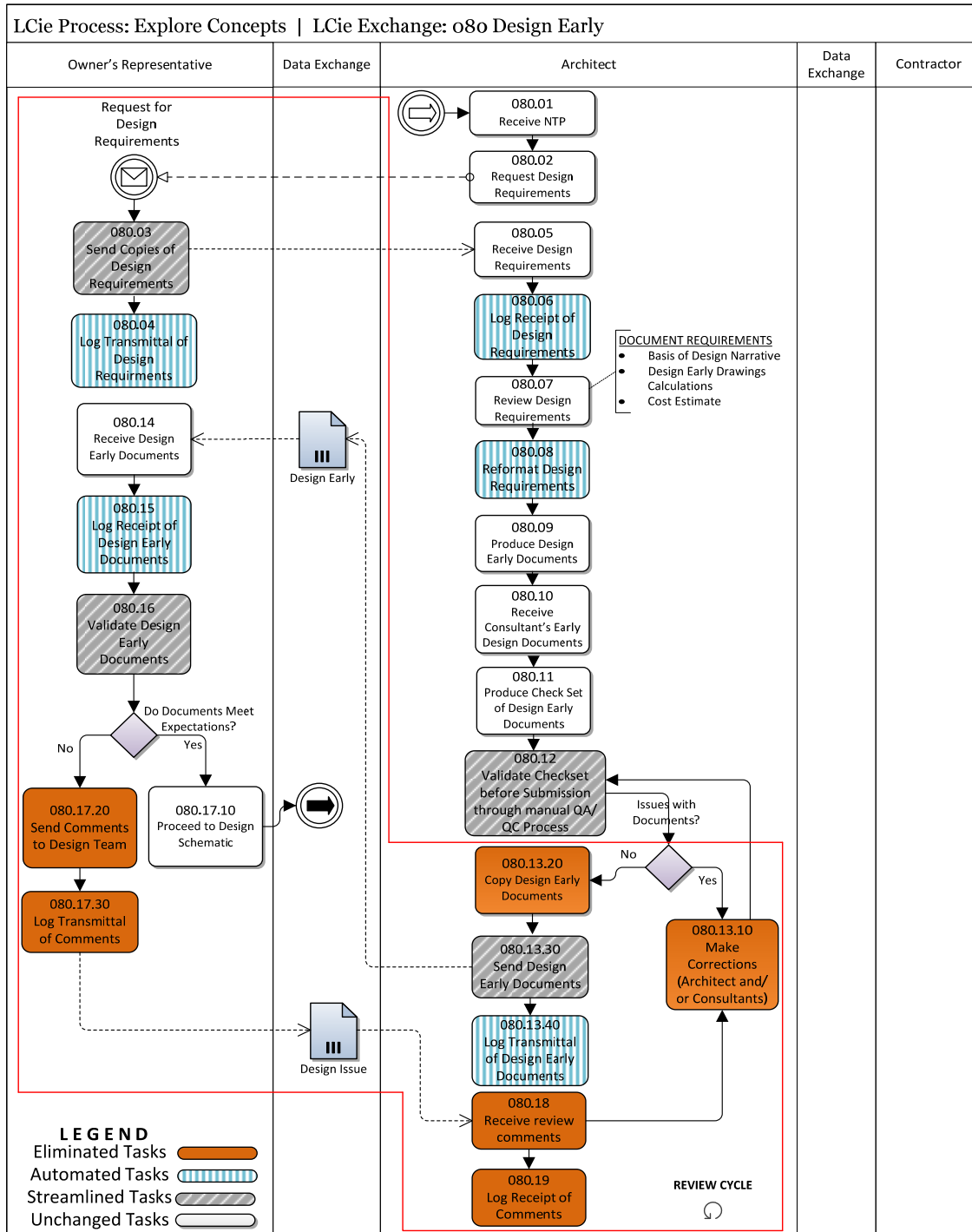


Figure 8 Explore Concepts - Design Early Process Model

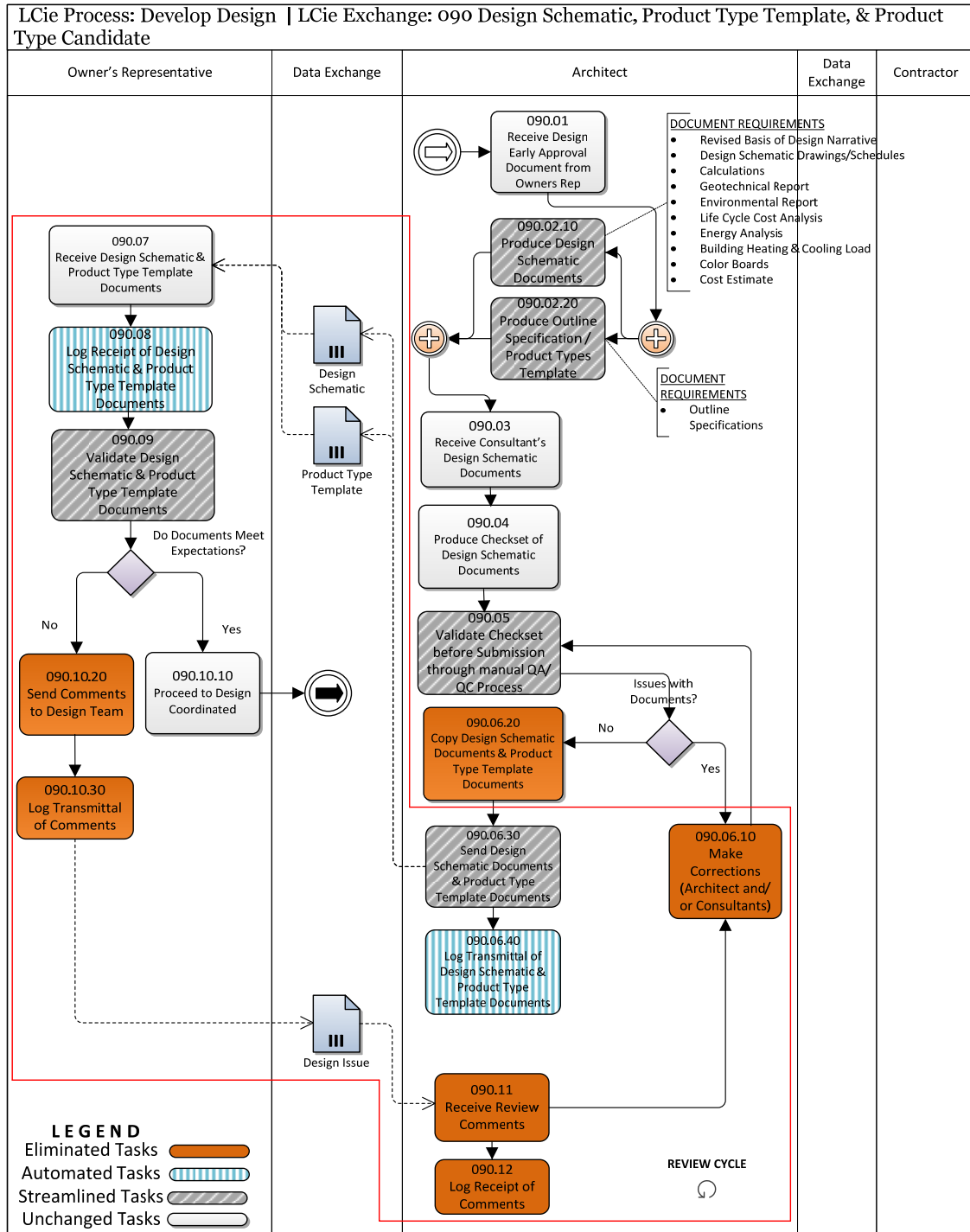
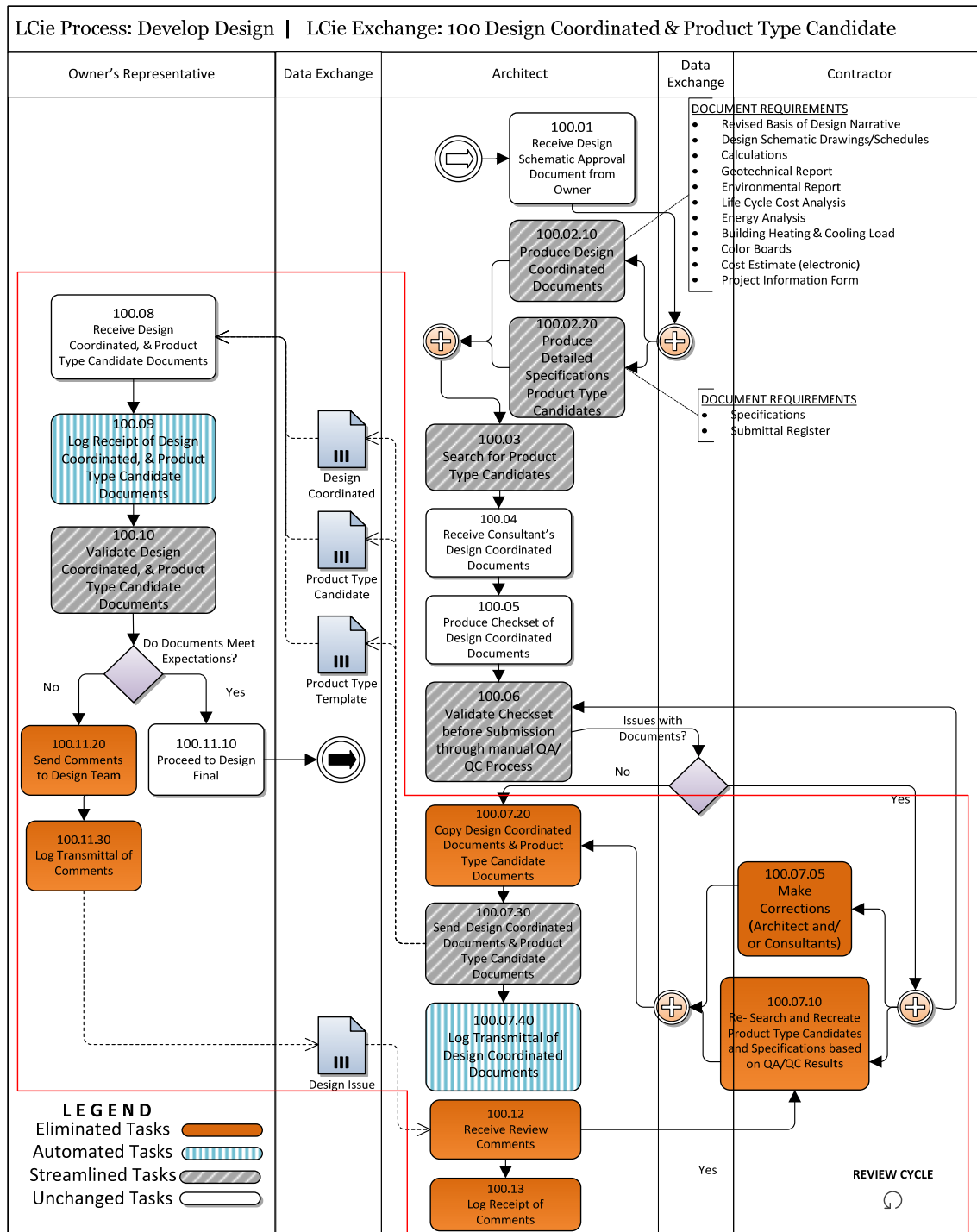


Figure 9 Develop Design - Design Schematic Process Model



**Figure 10 Develop Design - Design Coordinated Process Model**

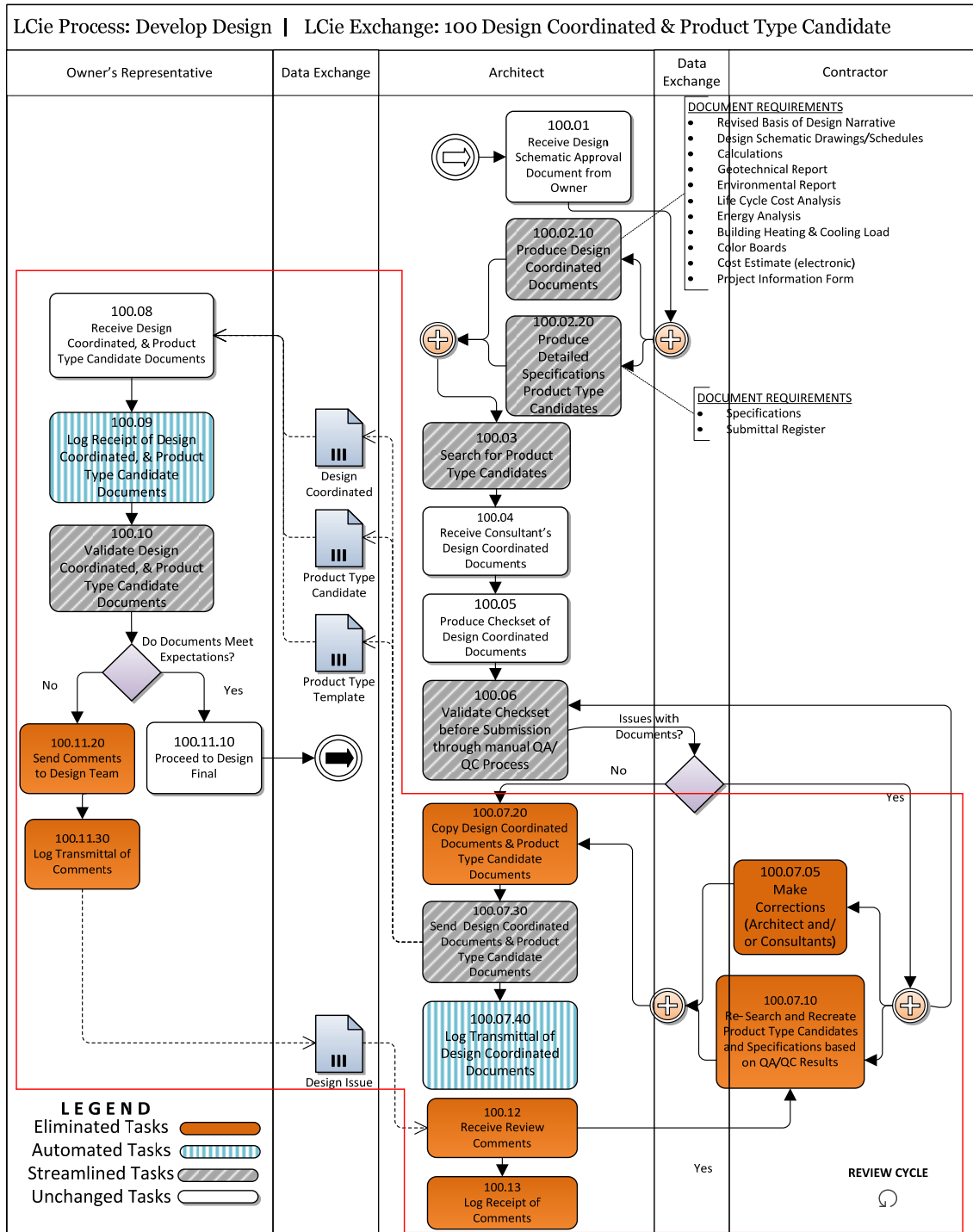
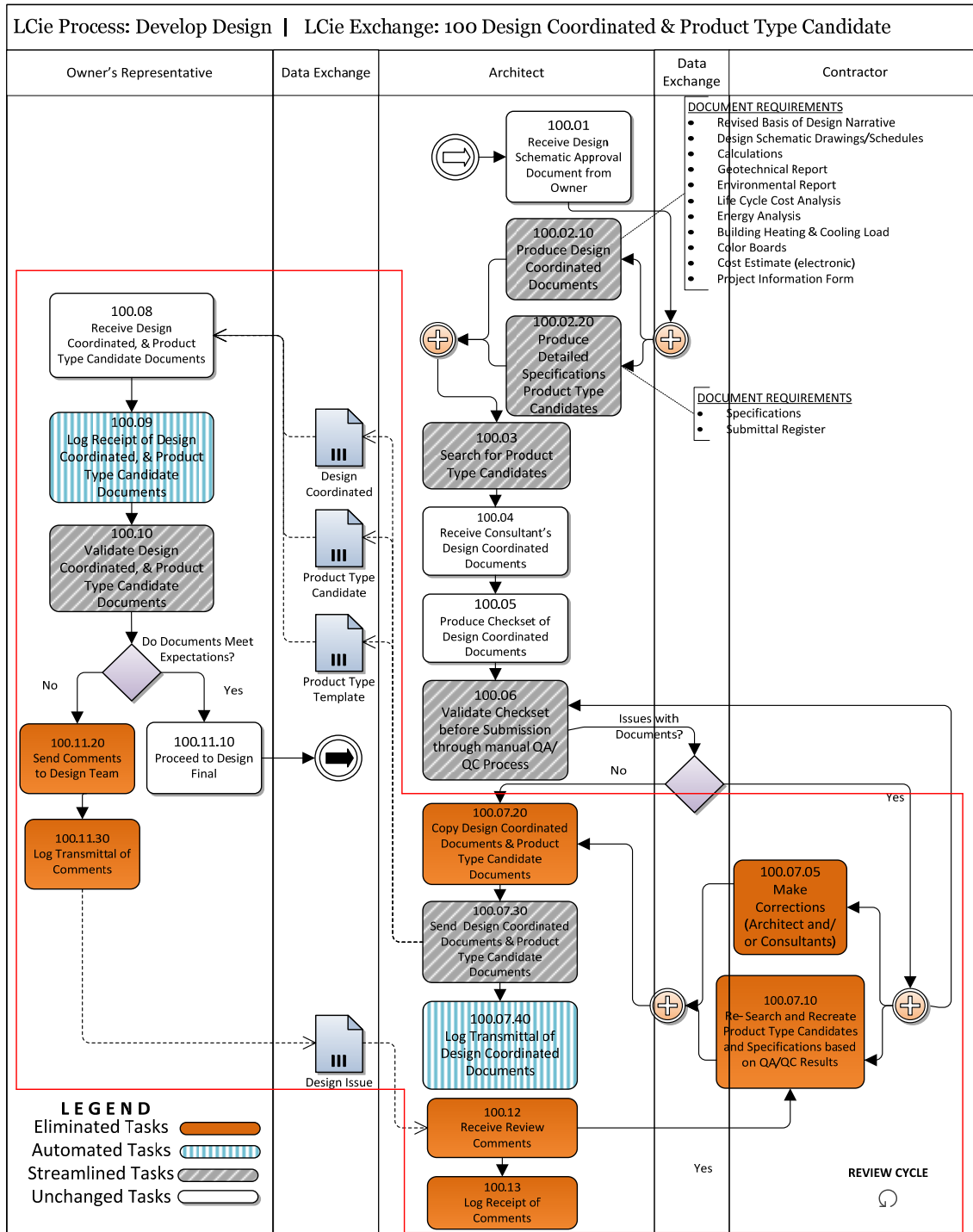


Figure 11 Product Type Template Process Model



**Figure 12 Develop Design – Product Type Candidate Process Model**



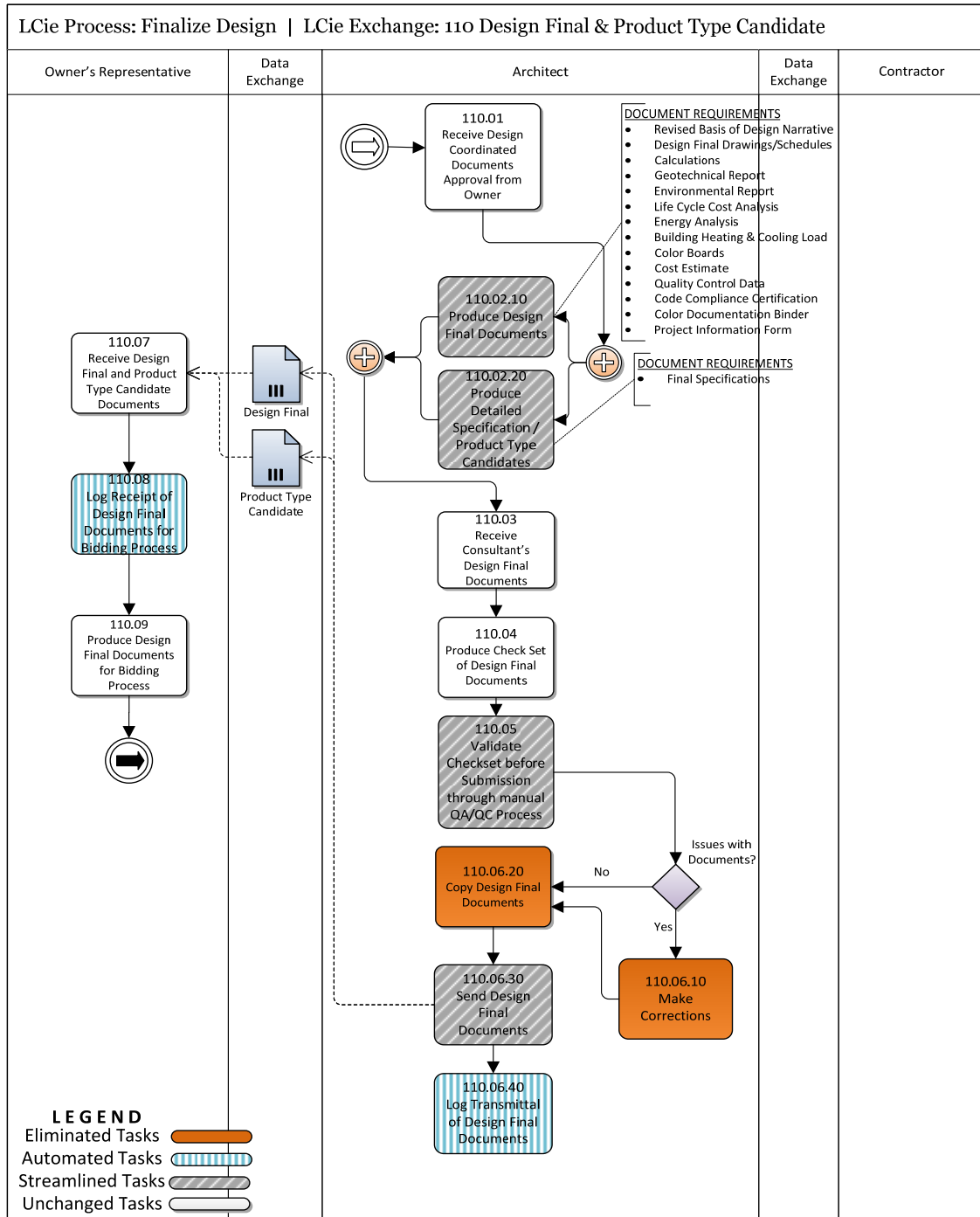
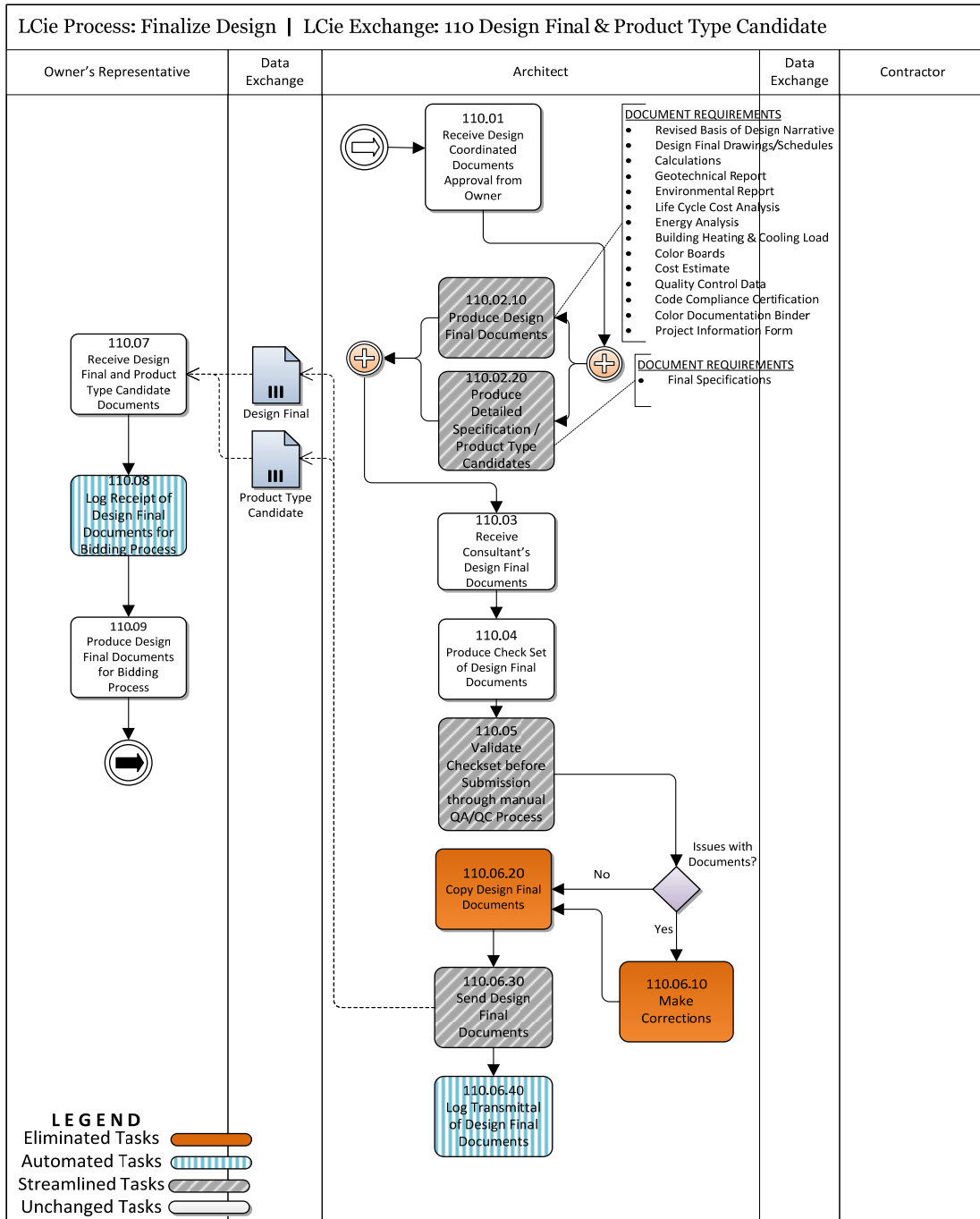


Figure 13 Finalize Design - Design Final Process Model



**Figure 14 Finalize Design – Product Type Candidate Process Model**

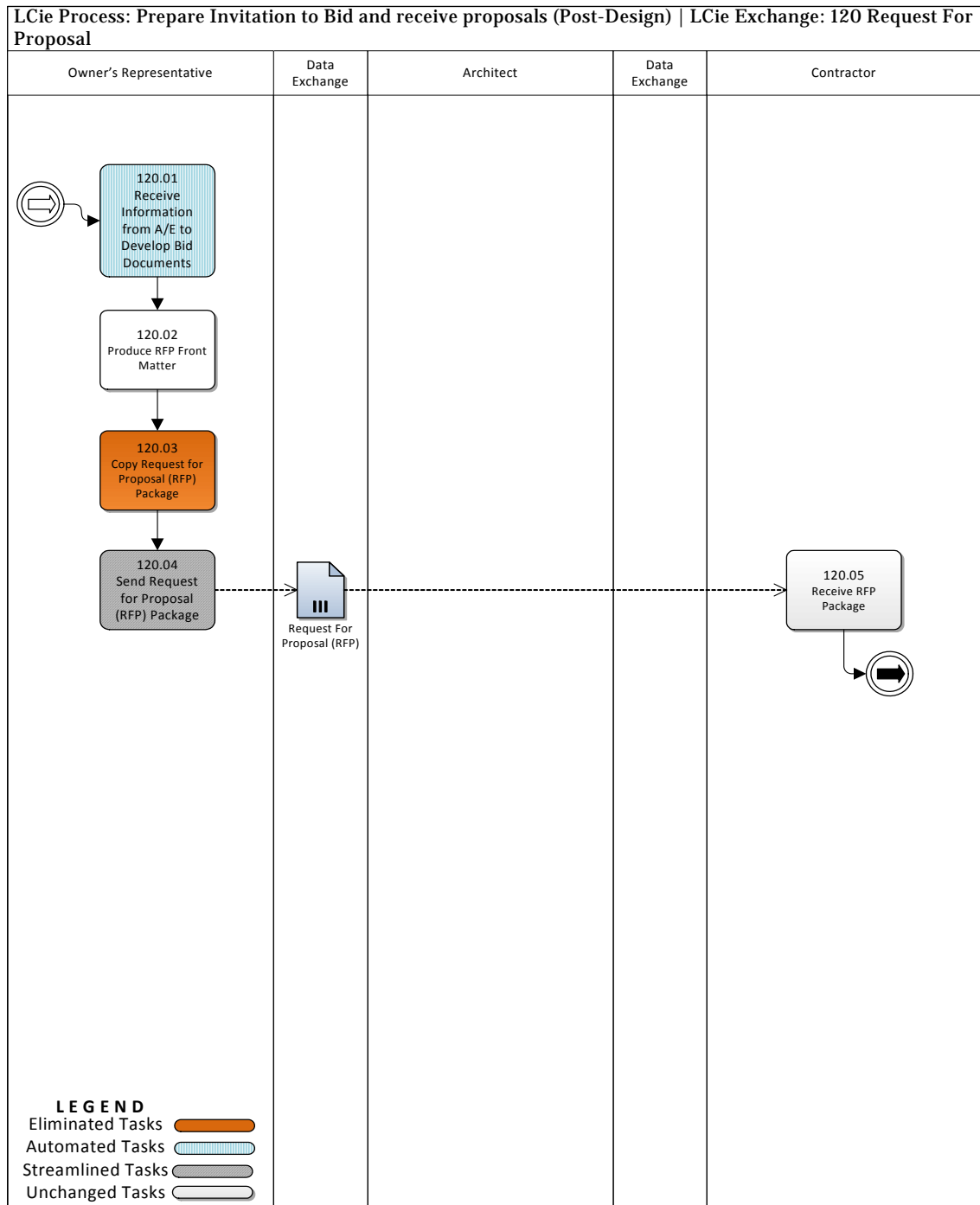


Figure 15 Prepare Invitation to Bid and Receive Proposals (Post-Design) Process Model

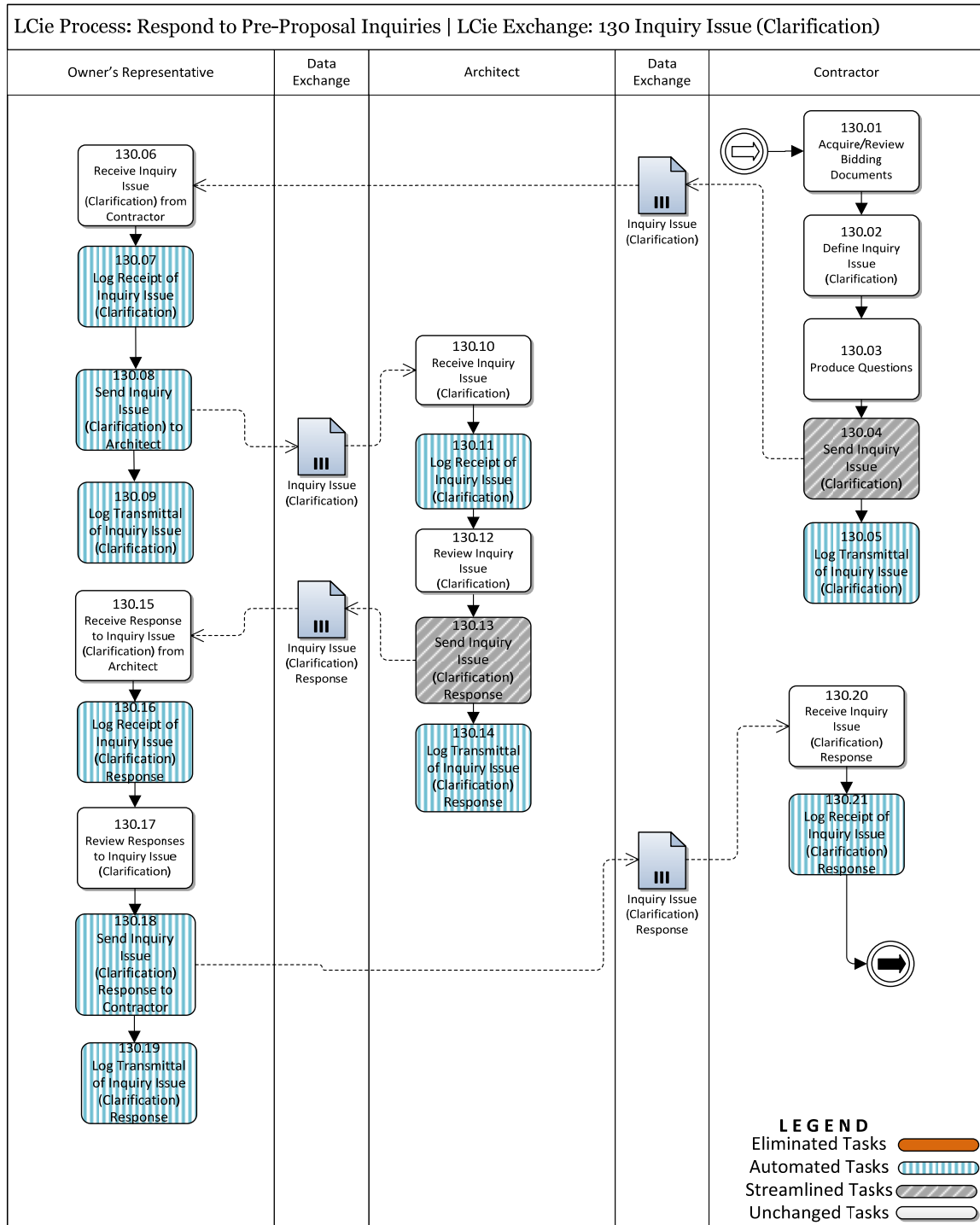


Figure 16 Respond to Pre-Proposal Inquiries Process Model

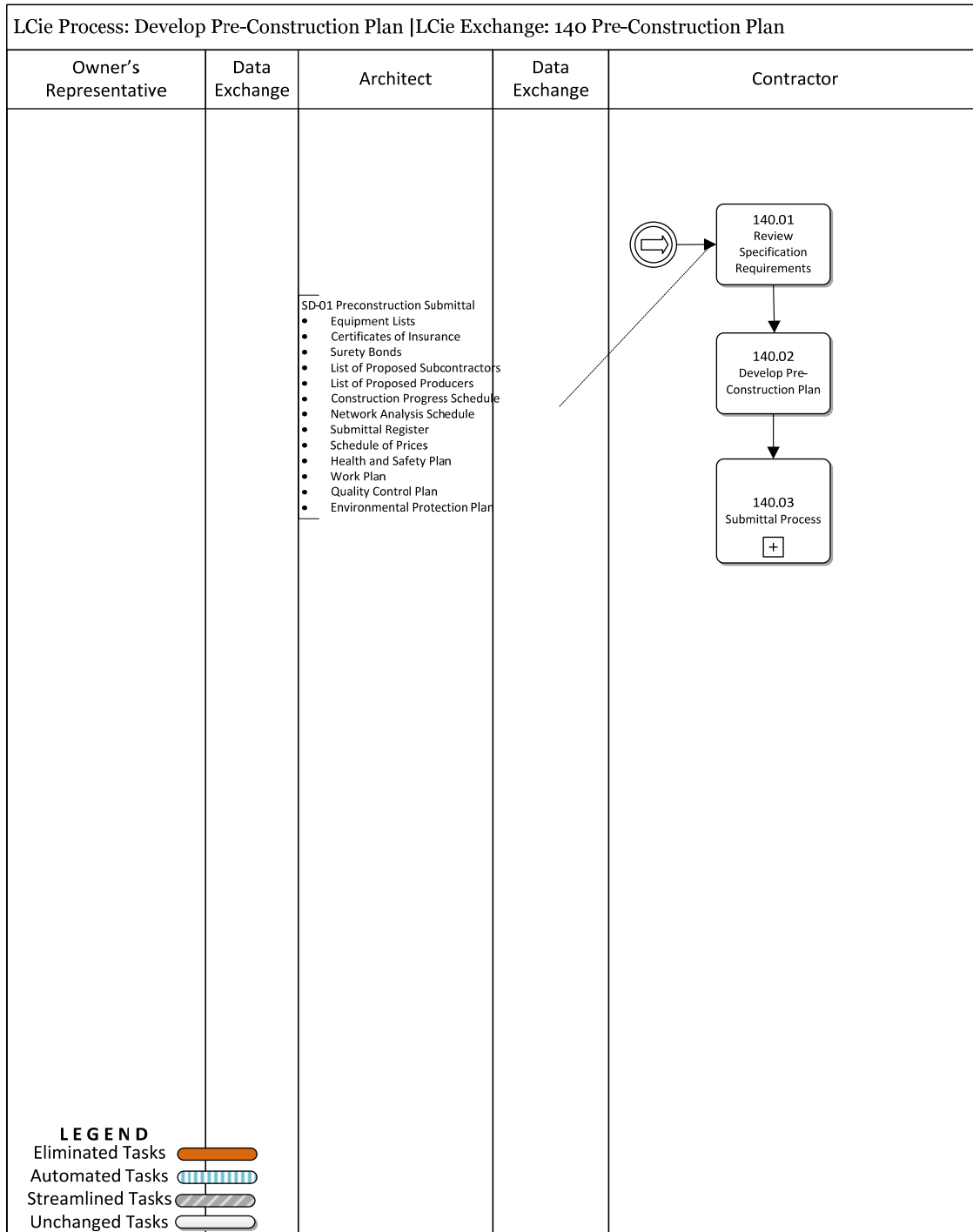


Figure 17 Develop Pre-Construction Plans Process Model

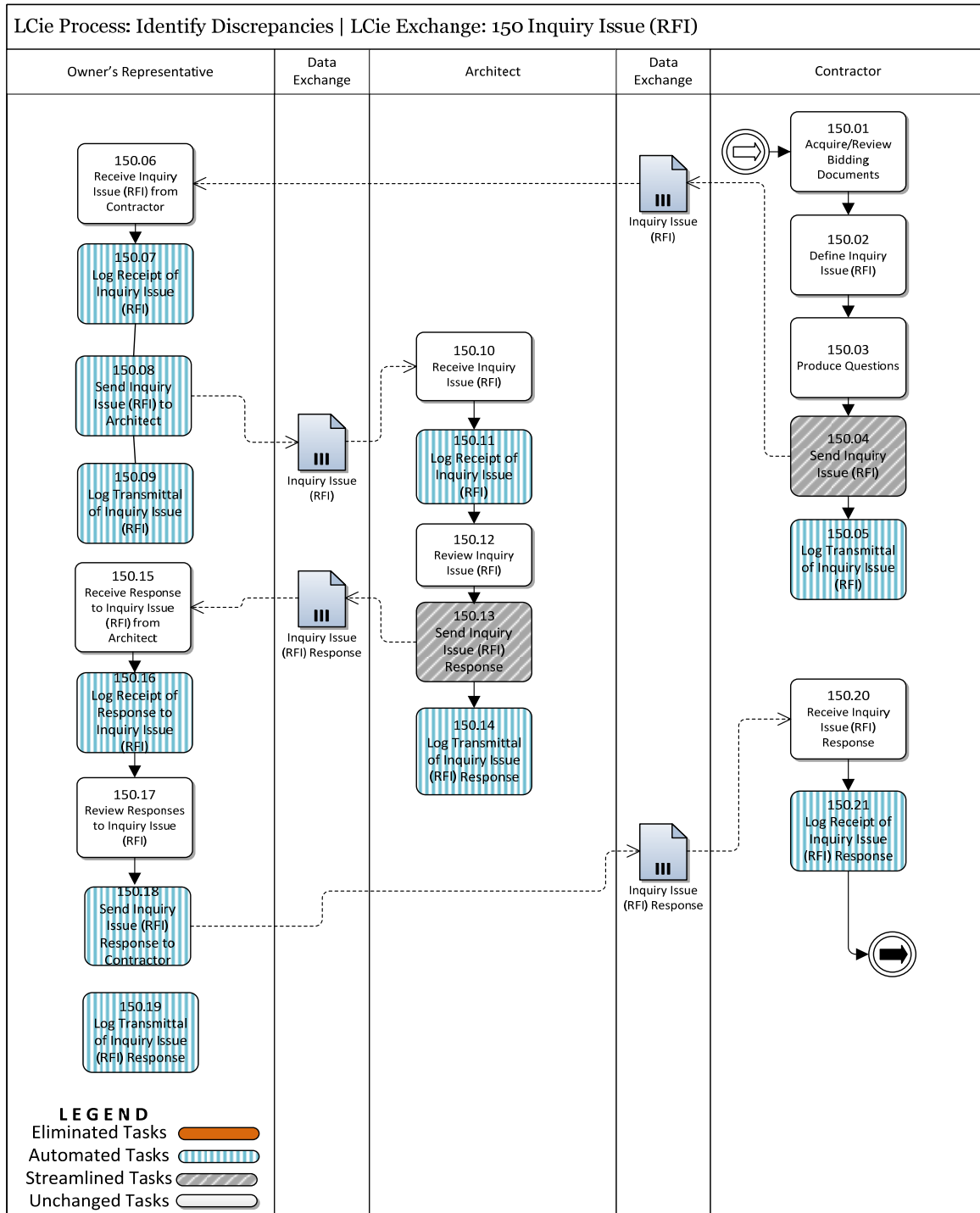
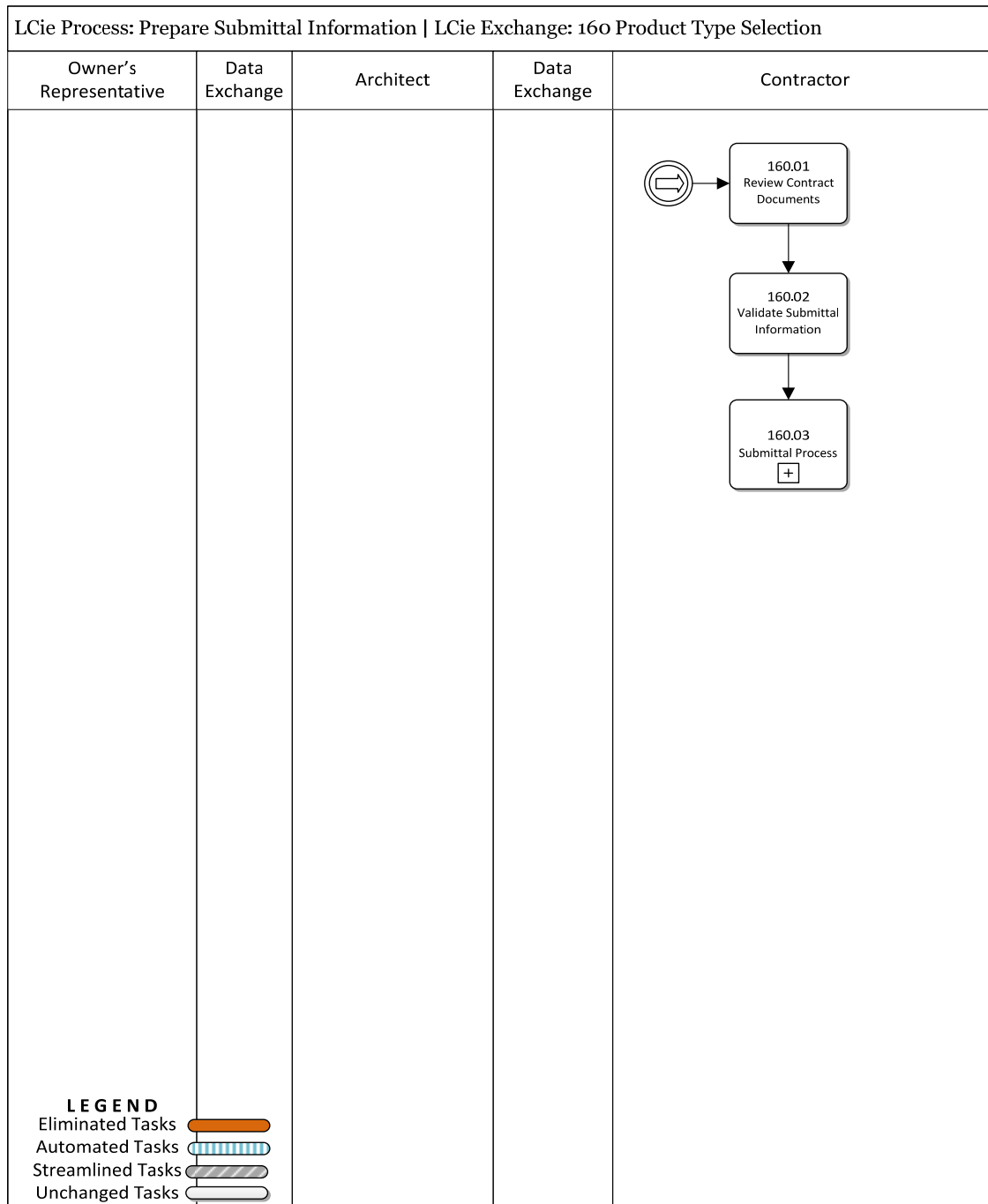


Figure 18 Identify Discrepancies Process Model



**Figure 19 Prepare Submittal Information Product Type Selection Process Model**

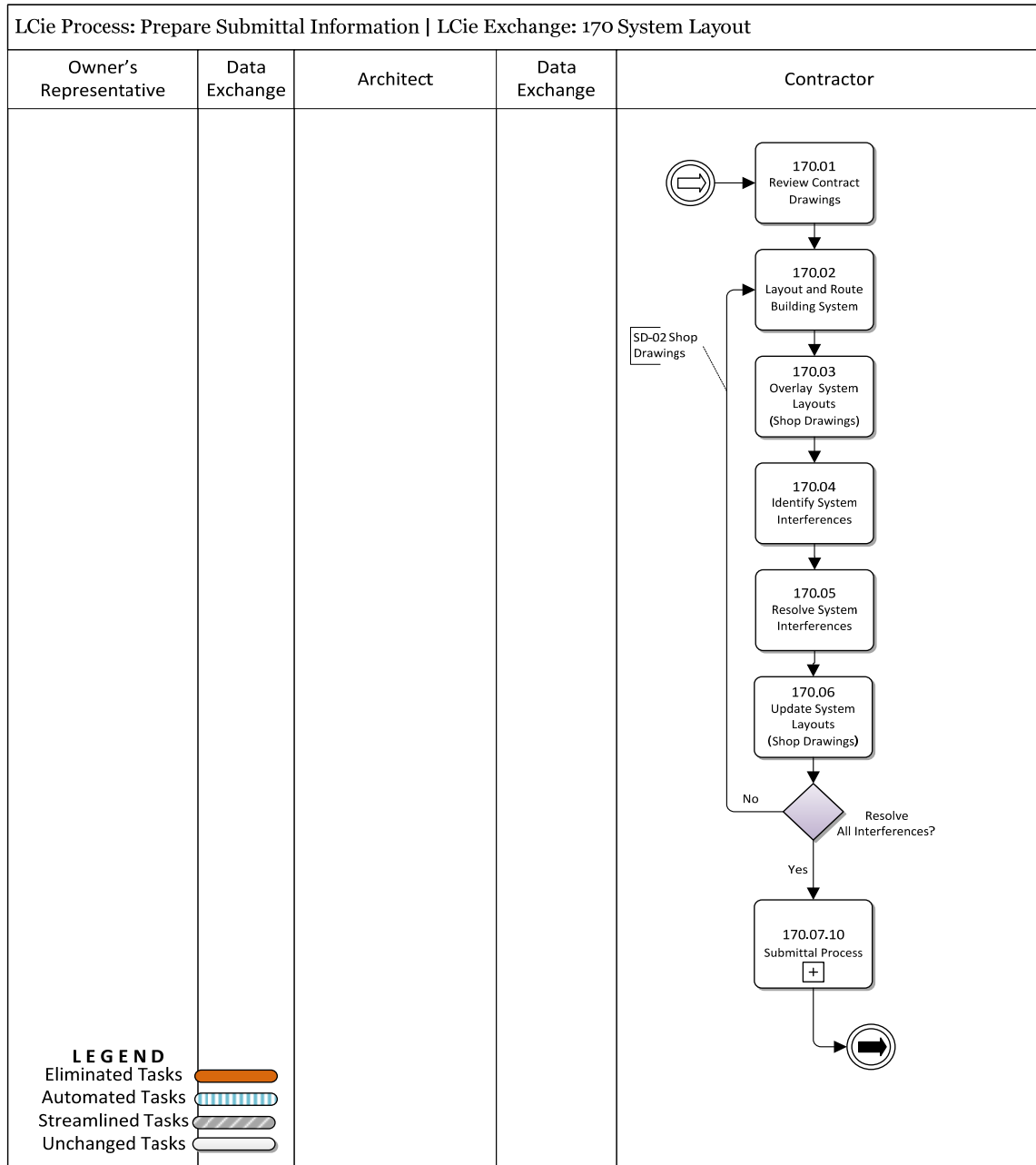
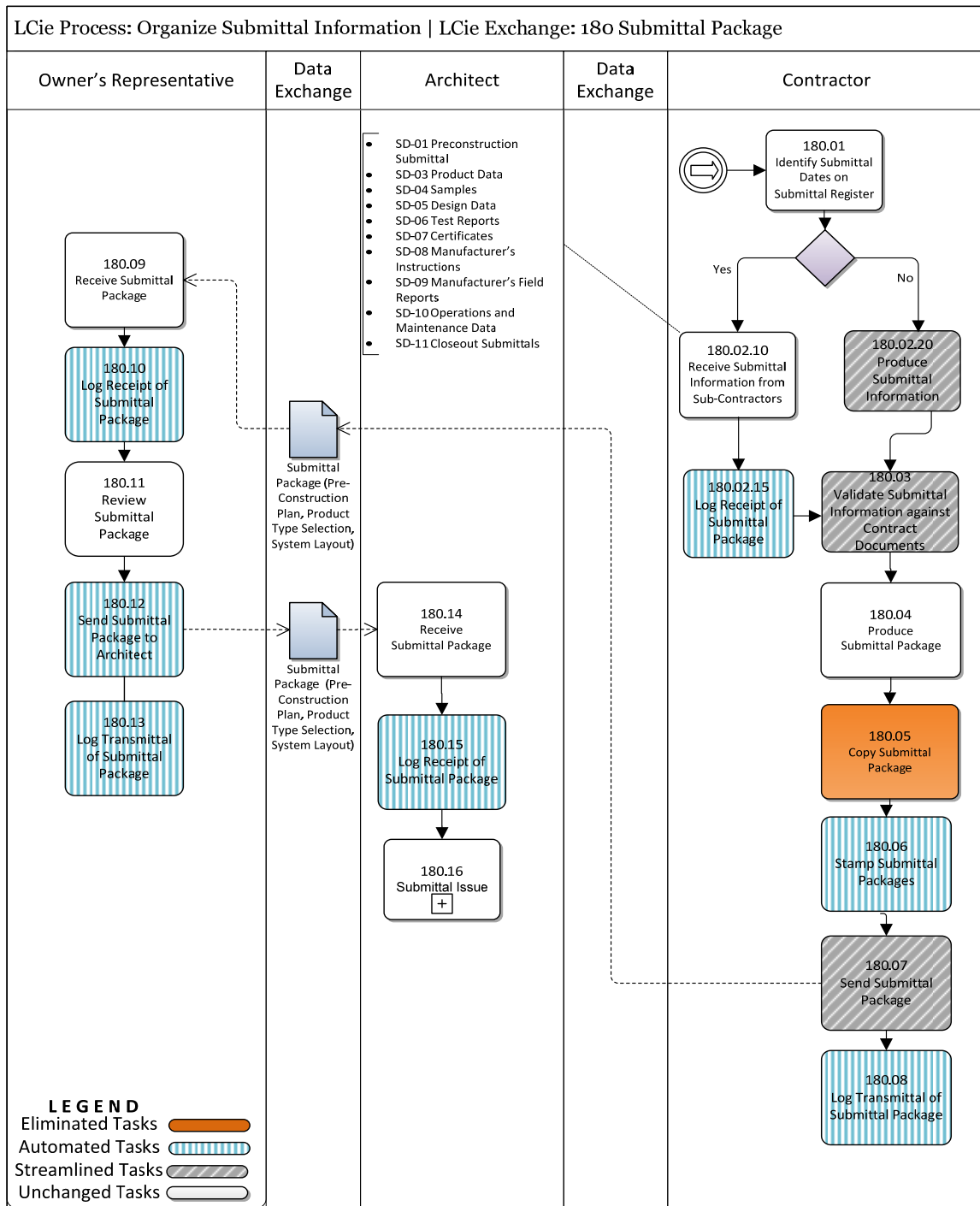


Figure 20 System Layout Process Model





**Figure 21 Organize Submittal Information Process Model**

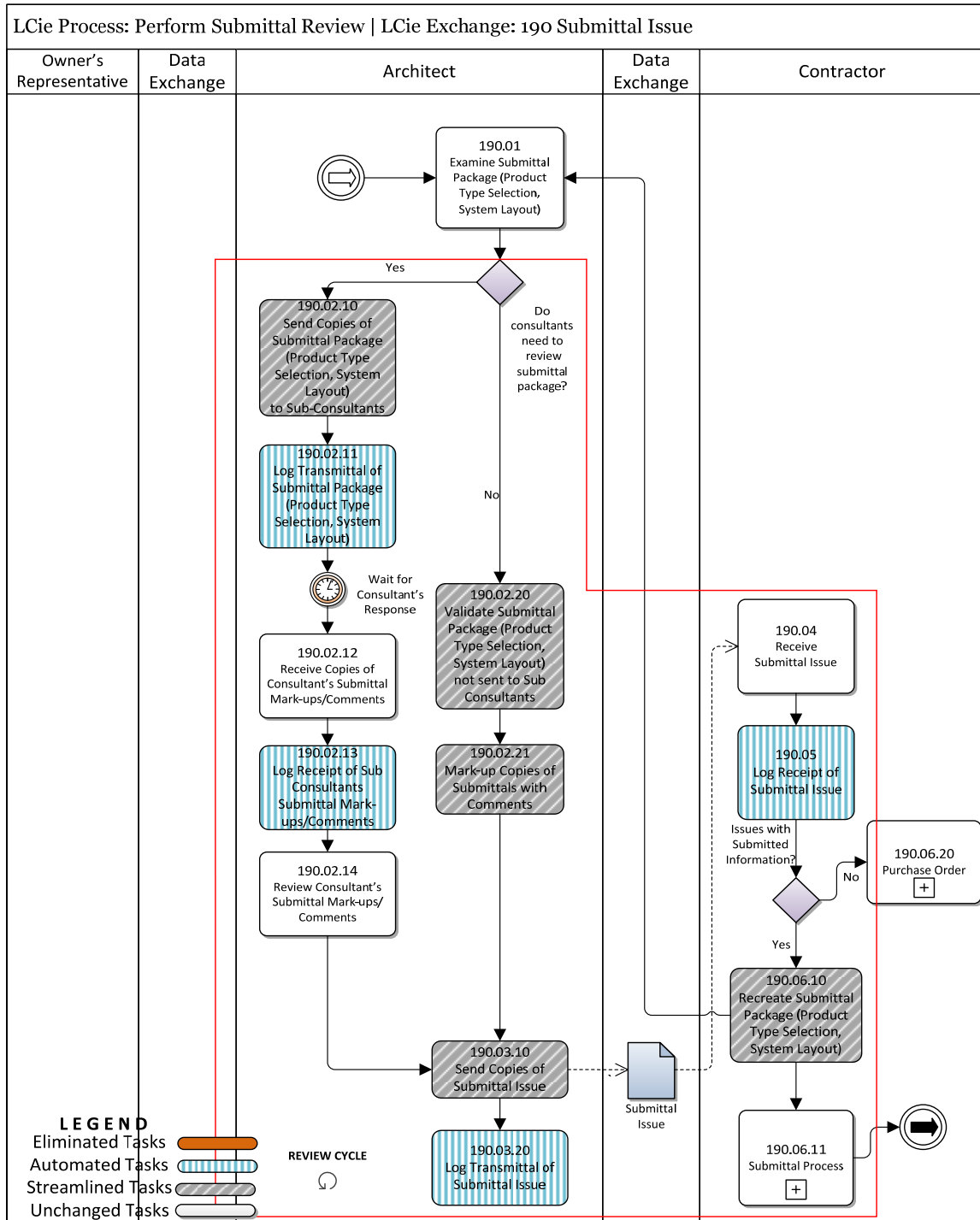


Figure 22 Perform Submittal Review - Submittal Issue Process Model

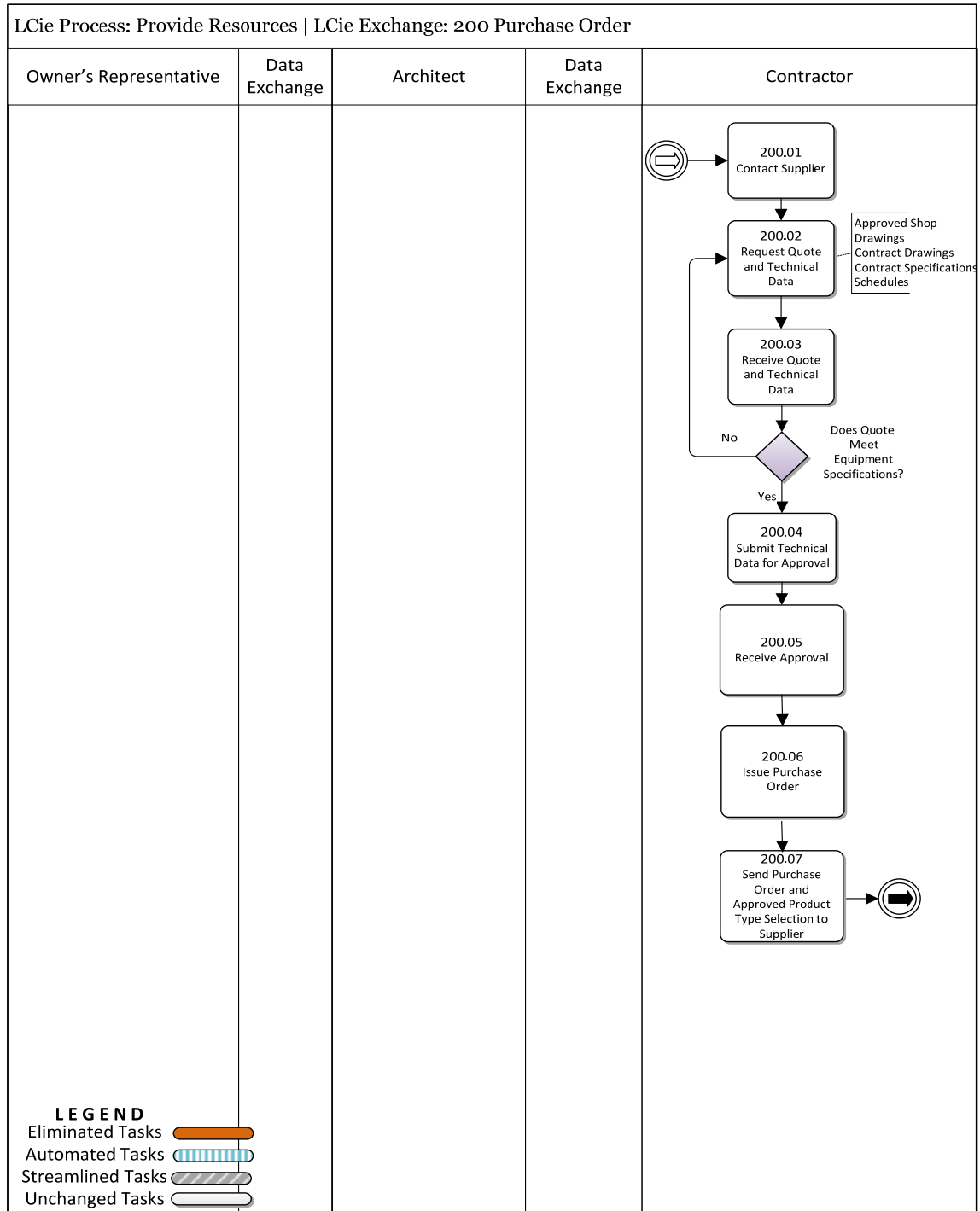


Figure 23 Provide Resources Process Model

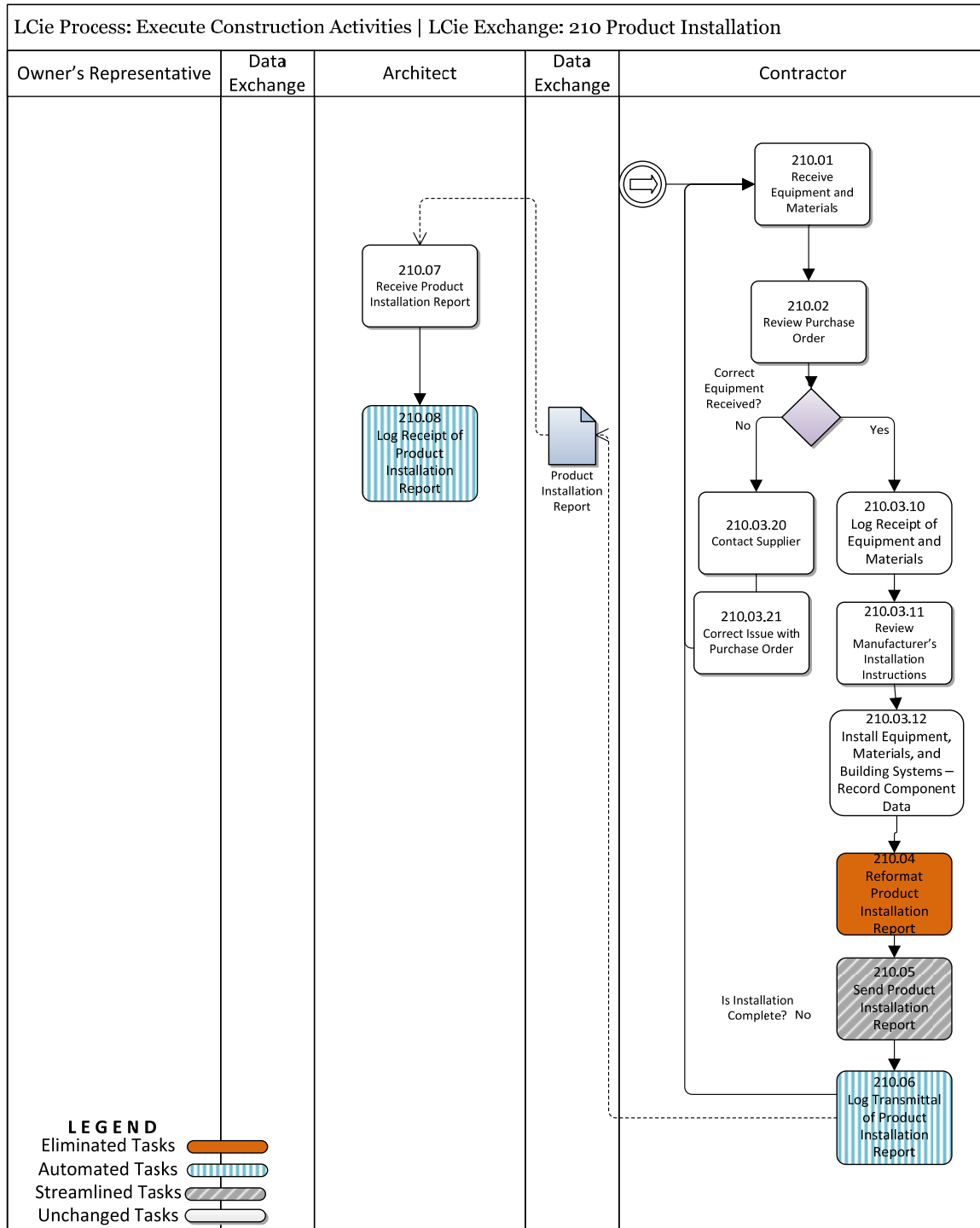


Figure 24 Execute Construction Activities Process Model

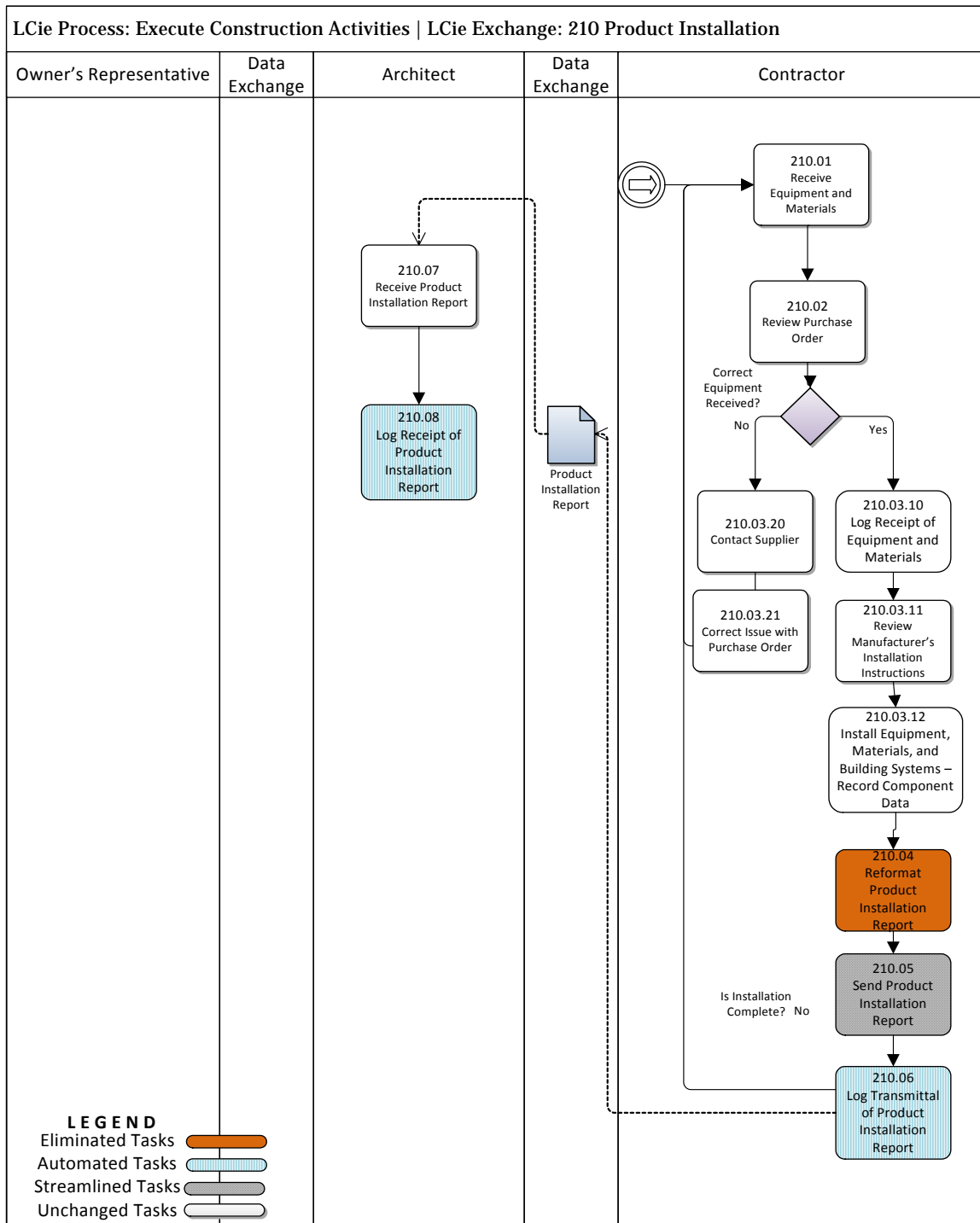


Figure 25 Product Installation Process Model

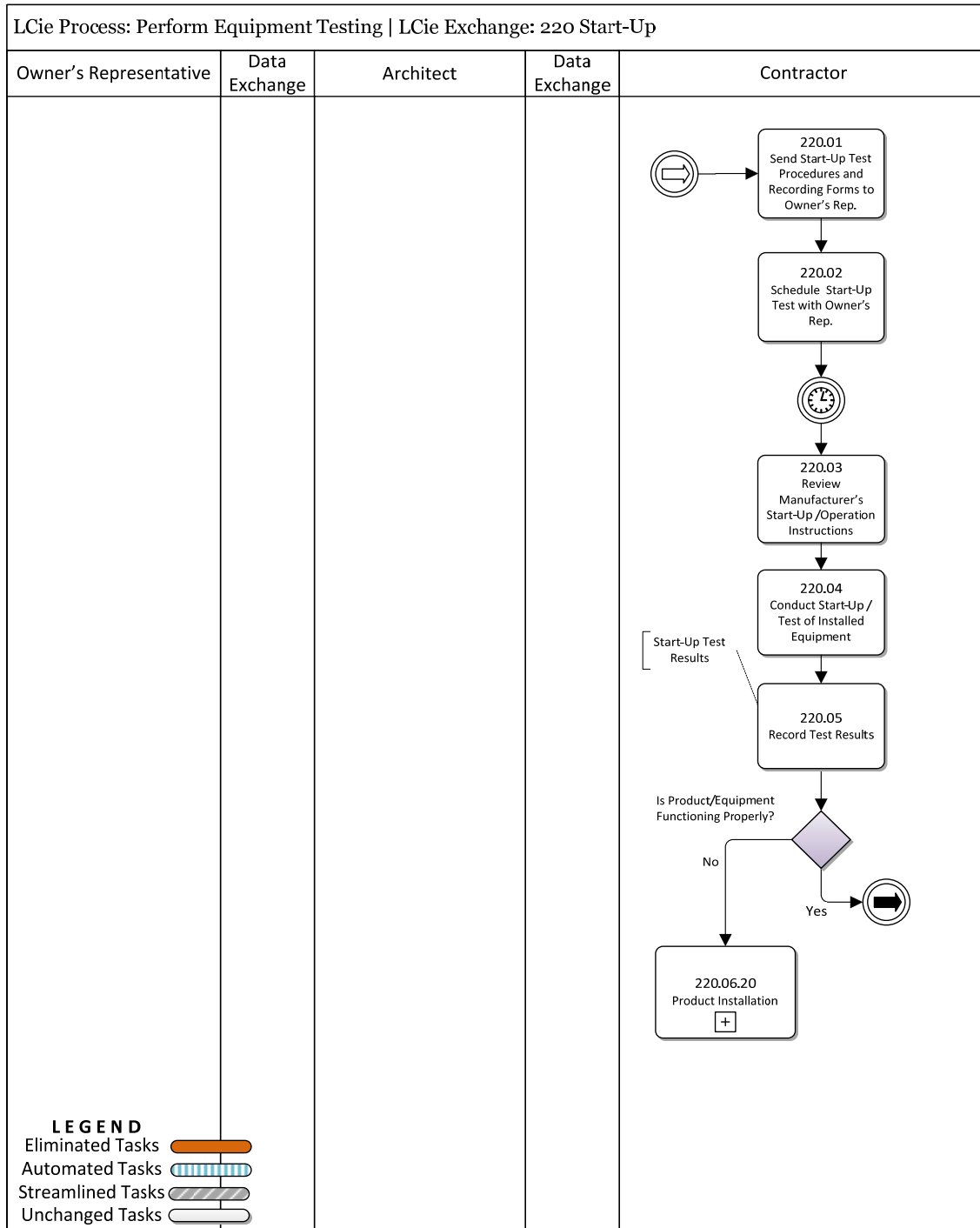


Figure 26 Perform Equipment Testing Process Model

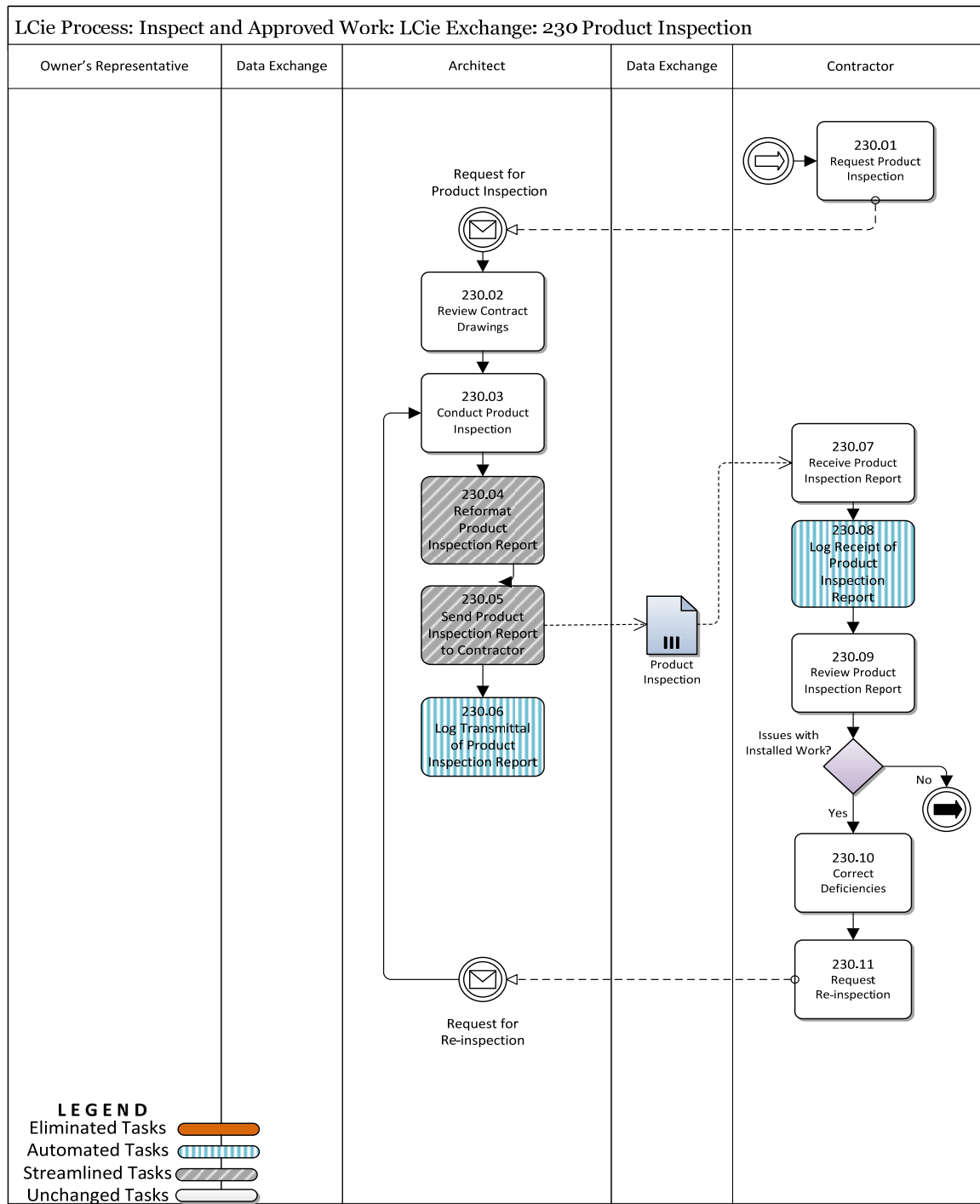


Figure 27 Inspect and Approve Work Process Model

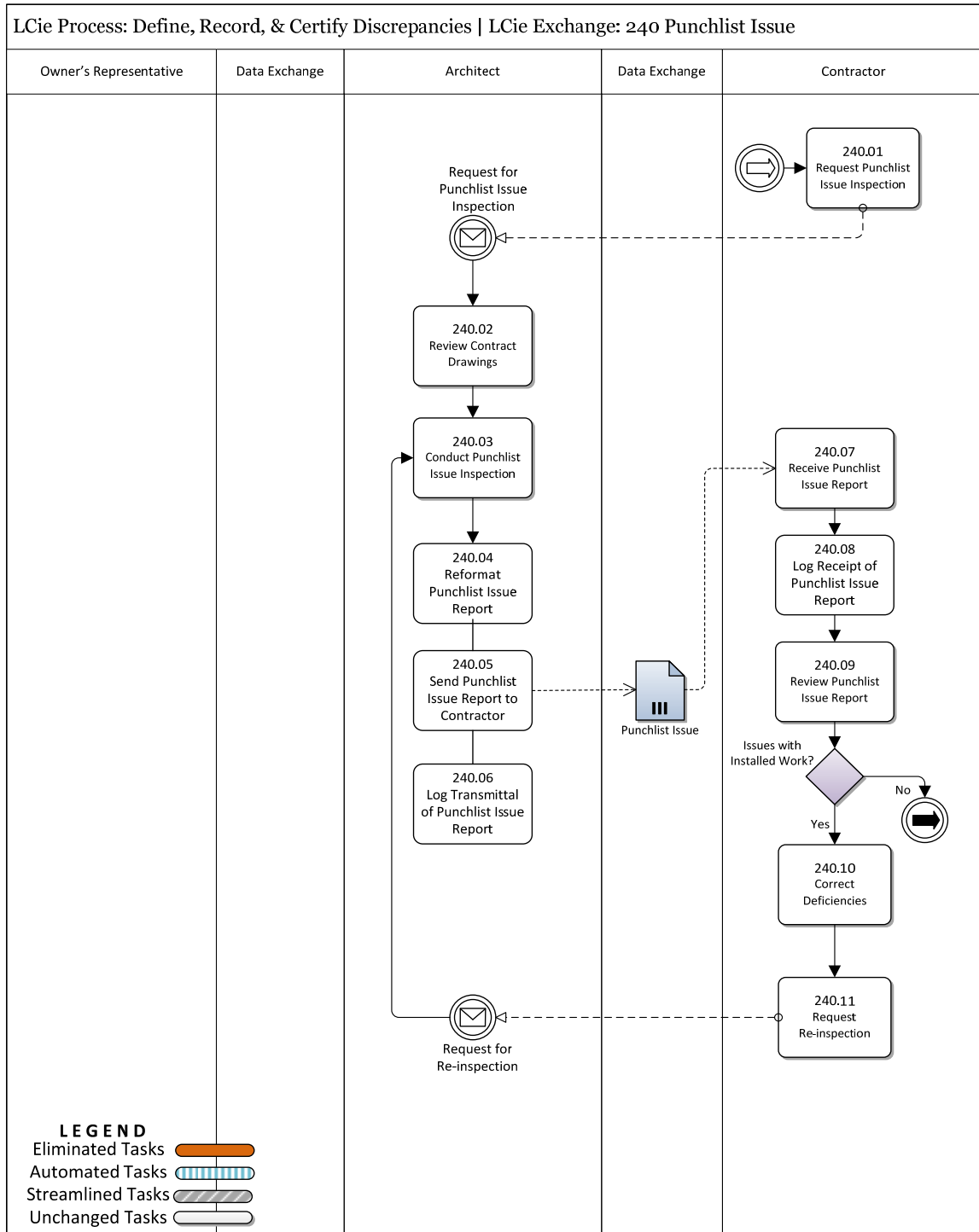


Figure 28 Define, Record, and Certify Discrepancies Process Model



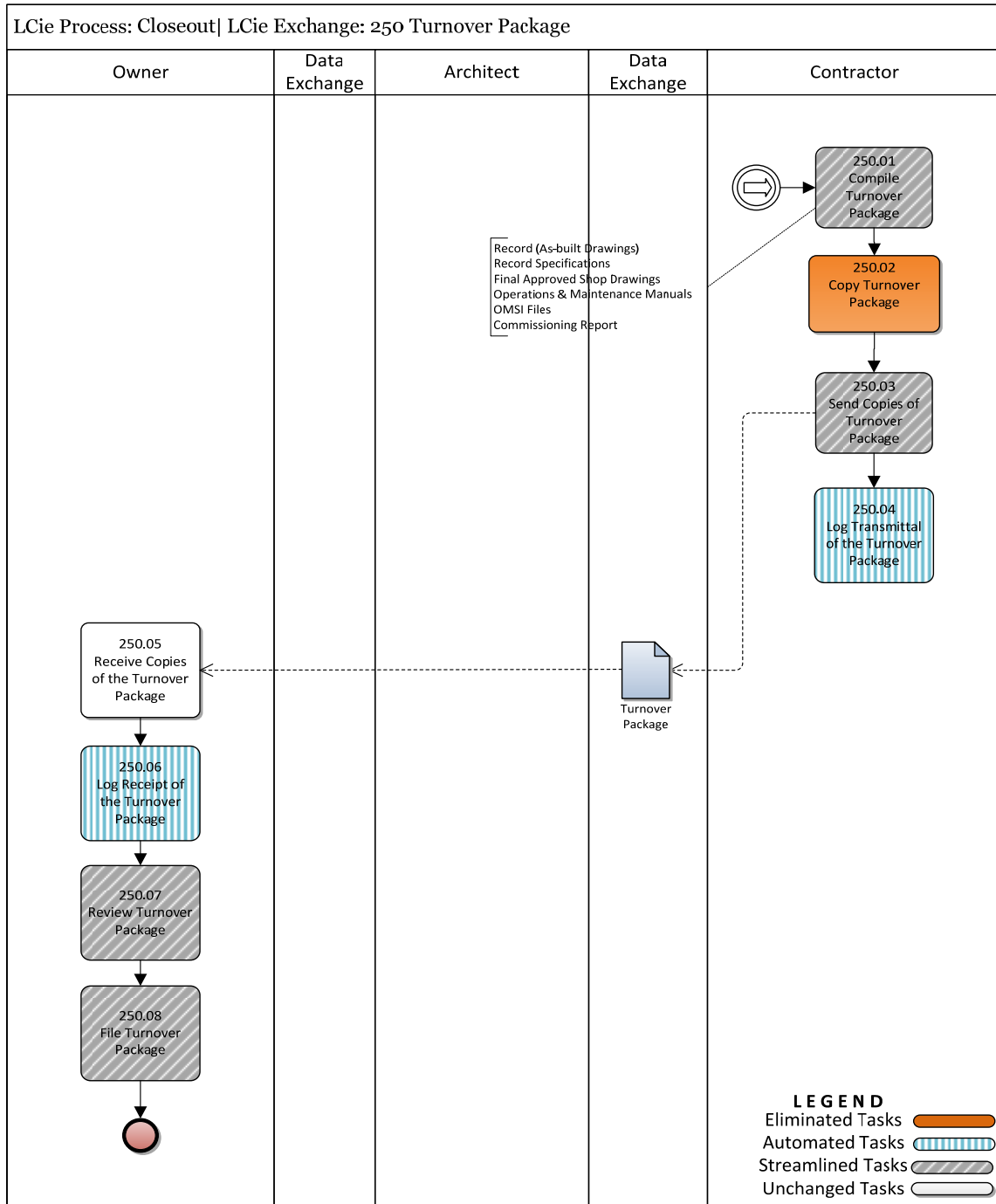


Figure 29 Closeout Process Model

## 4.2.5.2 Representative process models

### 4.2.5.2.1 Stakeholder coverage analysis

The rows in the figure below illustrate the stakeholders participating in each COBie-related business processes. The specific processes are listed in each column.

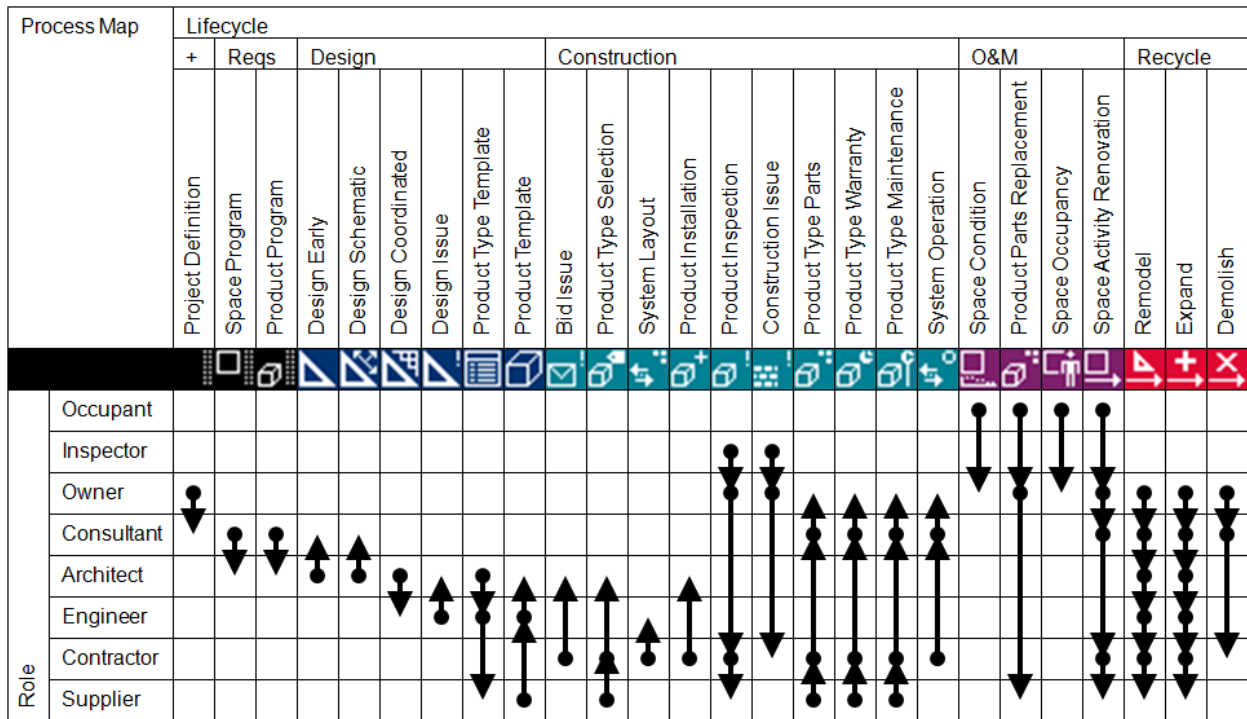


Figure 30 Stakeholder Coverage

### 4.2.5.2.2 Process coverage analysis

See Figure 30, above.

### 4.2.5.2.3 Contract documentary deliverable list

For each of the business processes identified in the COBie standard a list of the corresponding traditional contracted deliverable is provided. This information is provided as part of the NBIMS-US™ V3 standard to identify current document-based deliverables that can be replaced, in-all or in-part, by the information exchange standard defined in this document.

Table 5 Study and Define Needs Document List

Information Content:	▪ Facility Program
Contracted Exchange/Deliverable:	▪ Facility Criteria

**Table 6 Develop Design Criteria Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Type Data</li> <li>▪ Product Data</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Discipline Specification</li> </ul>

**Table 7 Study Technical Feasibility Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Feasibility Study Options</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Feasibility Study</li> </ul>

**Table 8 Communicate Results Decision Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Detailed Project Scope</li> <li>▪ Preliminary Budgetary Cost Information</li> <li>▪ Site Location &amp; Approval</li> <li>▪ Economic Analysis</li> <li>▪ Facility Planning Data</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Project Definition</li> </ul>

**Table 9 Develop Program – Space Program Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Facility Space Requirements</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Space Program</li> </ul>

**Table 10 Develop Program – Product Program Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Facility Product Requirements</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Product Program</li> </ul>

**Table 11 Prepare Invitation to Bid and Receive Proposal (Pre-Design) Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Project Definition</li> <li>▪ Space Program</li> <li>▪ Product Program</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Request for Proposal (RFP)</li> </ul>

**Table 12 Explore Concepts – Design Early Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Concept Design Drawings</li> <li>▪ Cost Estimate</li> <li>▪ Calculations</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Design Early</li> </ul>

**Table 13 Develop Design – Design Schematic Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Basis of Design Narrative</li> <li>▪ Design Schematic Drawings</li> <li>▪ Energy Analysis</li> <li>▪ Life Cycle Cost Analysis</li> <li>▪ Cost Estimate</li> <li>▪ Geotechnical Report</li> <li>▪ Calculations</li> <li>▪ Environmental Report</li> <li>▪ Outline Specifications</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Design Schematic</li> </ul>

	<ul style="list-style-type: none"> <li>Product Type Template</li> </ul>
--	---

**Table 14 Develop Design – Design Coordinated Document List**

Information Content:	<ul style="list-style-type: none"> <li>Basis of Design Narrative</li> <li>Design Coordinated Drawings</li> <li>Energy Analysis</li> <li>Life Cycle cost Analysis</li> <li>Cost Estimate</li> <li>Geotechnical Report</li> <li>Calculations</li> <li>Environmental Report</li> <li>Project Information Form</li> <li>Outline Specifications</li> <li>Submittal Register</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>Design Coordinated</li> <li>Product Type Template</li> </ul>

**Table 15 Finalize Design – Design Final Document List**

Information Content:	<ul style="list-style-type: none"> <li>Basis of Design Narrative</li> <li>Design Final Drawings</li> <li>Energy Analysis</li> <li>Life Cycle cost Analysis</li> <li>Cost Estimate</li> <li>Geotechnical Report</li> <li>Calculations</li> <li>Environmental Report</li> <li>Project Information Form</li> <li>Quality Control Data</li> <li>Color Documentation Binder</li> <li>Code Compliance Certification</li> <li>Specifications</li> <li>Submittal Register</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>Design Final</li> <li>Product Type Candidate</li> </ul>

**Table 16 Prepare Invitation to Bid and Receive Proposals (Post Design) Document List**

Information Content:	<ul style="list-style-type: none"> <li>Final Design Documents</li> <li>Specifications</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>Request for Proposal (RFP)</li> </ul>

**Table 17 Respond to Pre-Proposal Inquiries Document List**

Information Content:	<ul style="list-style-type: none"> <li>Clarification Request</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>Inquiry Issue (Clarification)</li> </ul>

**Table 18 Develop Pre-Construction Plan Document List**

Information Content:	<ul style="list-style-type: none"> <li>Equipment Lists</li> <li>Certificates of Insurance</li> <li>Surety Bonds</li> <li>List of Proposed Subcontractors</li> </ul>
----------------------	---

	<ul style="list-style-type: none"> <li>▪ List of Proposed Producers</li> <li>▪ Construction Progress Schedule</li> <li>▪ Network Analysis Schedule</li> <li>▪ Submittal Register</li> <li>▪ Schedule of Prices</li> <li>▪ Health and Safety Plans</li> <li>▪ Work Plan</li> <li>▪ Quality Control plan</li> <li>▪ Environmental Protection Plan</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Pre-Construction Plan</li> </ul>

**Table 19 Identify Discrepancies Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Request for Information</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Inquiry Issue (RFI)</li> </ul>

**Table 20 Prepare Submittal Information – Product Type Selection Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Product Data</li> <li>▪ Samples</li> <li>▪ Design Data</li> <li>▪ Test Reports</li> <li>▪ Certificates</li> <li>▪ Manufacturer's Instructions</li> <li>▪ Manufacturer's Field Reports</li> <li>▪ Operations and Maintenance Data</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Product Type Selection</li> </ul>

**Table 21 Prepare Submittal Information – System Layout Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Shop Drawings</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ System Layout</li> </ul>

**Table 22 Organize Submittal Information Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Pre-Construction Plan</li> <li>▪ Product Type Selection</li> <li>▪ System Layout</li> <li>▪ (Refer to the Information Content for each of these individual items for a complete list of included information.)</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Submittal Package</li> </ul>

**Table 23 Perform Submittal Review – Submittal Issue Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Marked-Up Submittal Package</li> <li>▪ Submittal Review Comments</li> </ul>
Contracted Exchange/Deliverable:	<ul style="list-style-type: none"> <li>▪ Submittal Issue</li> </ul>

**Table 24 Provide Resources Document List**

Information Content:	▪ Purchase Order
Contracted Exchange/Deliverable:	▪ Purchase Order

**Table 25 Execute Construction Activities Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Design Final Drawings</li> <li>▪ Product Type Candidate</li> <li>▪ Approved Shop Drawings</li> <li>▪ Manufacturer's Instructions</li> </ul>
Contracted Exchange/Deliverable:	▪ Product Installation

**Table 26 Perform Equipment Testing Document List**

Information Content:	▪ Equipment Start-Up Test Results
Contracted Exchange/Deliverable:	▪ Equipment Start-Up Report

**Table 27 Inspect and Approve Work Document List**

Information Content:	▪ Observation Field Report
Contracted Exchange/Deliverable:	▪ Product Inspection

**Table 28 Define, Record and Certify Discrepancies Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Previous Product Inspections</li> <li>▪ Final Walkthrough Report</li> <li>▪ Field Reports</li> </ul>
Contracted Exchange/Deliverable:	▪ Punch-list Issue

**Table 29 Closeout Document List**

Information Content:	<ul style="list-style-type: none"> <li>▪ Operations and Maintenance Manuals</li> <li>▪ Record of Designated Equipment</li> <li>▪ Materials Data Files</li> <li>▪ Commissioning Report</li> <li>▪ Record Specifications</li> <li>▪ Record (As-Built) Drawings</li> <li>▪ Final Approved Shop Drawings</li> </ul>
Contracted Exchange/Deliverable:	▪ Turnover Package

#### 4.2.5.2.4 Contract documentary deliverable analysis

##### 4.2.5.2.4.1 Study and define needs

Facility Criteria may be provided in COBie format as templates of specific space types. The requirements for spatial finishes and equipment may also be easily incorporated into the COBie format.

Allowing space types to be represented in a COBie format is an additional business rule that is defined in the previously noted BPie project.

#### **4.2.5.2.4.2 Develop design criteria**

Product Type information commonly found in construction specifications may be developed as a constraint model in COBie format against specific product types. The specific requirements for product types may be defined in COBie as attributes. COBie.Attributes allows the specification of properties as requirements. COBie.Attributes also allows the listing of allowed sets of values and optional values.

The definition of product specifications may be addressed in the LCie and SPie documents. Based on such definitions automated checking of product-specific information may be conducted in many types of exchanges later in the project. Note that such checks are not the same as projects related to checking building code compliance.

#### **4.2.5.2.4.3 Study technical feasibility**

Feasibility Studies that produce information for each option about facility spaces, equipment and other assets may be compared against technical criteria requirements. Such checks can identify constraint violations, but will not evaluate the quality of design from any other than a purely technical point of view.

#### **4.2.5.2.4.4 Communicate results decision**

There are many aspects of project definition that are not supported by COBie exchanges; however, information about the detailed project scope may be communicated.

#### **4.2.5.2.4.5 Develop program – space program**

The Space Program may be provided in COBie format as specific space types and instances of those types. The requirements for spatial finishes and equipment may also be easily incorporated into the COBie format.

Space types represented for COBie is defined in the previously noted BPie project. Note that having the criteria for each space type in COBie format and having the Space Program deliverables also in COBie format allows automated constraint checking to be accomplished between the planned facility and the owner's criteria.

#### **4.2.5.2.4.6 Develop program – product program**

The Product Program may be provided in COBie format as specific space types and instances of those types. The requirements for spatial finishes and equipment may also be easily incorporated into the COBie format.

Note that having the criteria for each product type in COBie format and having the Product Program deliverables also in COBie format allows automated constraint checking to be accomplished between the planned facility and the owner's program and specifications' criteria.

#### **4.2.5.2.4.7 Prepare invitation to bid and receive proposal (pre-design)**

While there are many aspects of the design RFP that cannot be represented in COBie format, clearly the Space and Product Program are able to be represented in COBie.

#### **4.2.5.2.4.8 Explore concepts – design early**

The majority of the deliverables produced in the Early Design stage are not relevant to COBie. When considering alternative layouts during the early design stage, however, checking these layouts against the owner's criteria and RFP Space/Product Program may assist the designer from making choices in violation of the owner's requirements.

The architectural drawings produced at this stage do contain scheduled facility asset information pertaining to Spaces, Doors, Windows, Plumbing Fixtures, and Lighting Fixtures. The information in these schedules is represented in COBie and submitted with the drawings.

#### **4.2.5.2.4.9 Develop design – design schematic**

Many of the deliverables produced during design have information content that is partially, or completely, found within the COBie data set. As a result designers may obtain significant benefits from the export of design deliverable reports using COBie-required data that present in their native design systems. Several specific deliverables are described in the paragraphs below.

The Basis of Design Narrative, with respect to those aspects that are required for the commissioning process, can be directly mapped to the COBie data set. The identification of building systems and zones, and the overall design factors may be documented in the COBie.Zone, COBie.System and COBie.Attributes entities. Those product types whose basis of design has been selected at this point in the design may be captured through the COBie.Type and COBie.Attribute entities.

Another format of the Basis of Design Narrative, with respect to the specifics of selecting assets and their properties is represented on design drawings as schedules. From scheduled asset information design product data sheets (product type templates) and room data sheets may also be exported. Such information may be used to check the current design against previous design stages for changes, and to evaluate later contractor submissions for appropriate finish and product selections.

Information supporting a variety of types of cost analysis such as life-cycle cost analysis and cost estimates may be supported by COBie. The use of COBie data to for counting assets would be a valuable shortcut when performing drawing take-offs. Information about the quality level of those assets, contained in COBie.Type and COBie.Attribute entities may support both conceptual and detailed cost estimating.

Information related to the requirements for various products found in COBie.Type may also be used to create outline specifications for the project from the properties identified in the COBie model. Such applications are not in widespread use; however, the demonstration of such an application in December 2010 validated this potential use of COBie data.

COBie files will not contain the following types of contractual deliverables during the design stage: drawings, schematics, reports, and calculations. Information contained in design reports and calculations is outside the scope of the COBie specification. Drawing files contain a level of geometric detail that is not suitable for inclusion on COBie. The information content of drawings, particularly that information found on drawing schedules should be included in COBie. In fact, the comparison between drawings schedule information and associated COBie deliverables is the basis for the assessment of the quality of the COBie.

#### **4.2.5.2.4.10 Develop design – design coordinated**

In addition to the discussion of design deliverable coverage noted in the previous section, the coordinated design deliverable includes deliverables such as the Project Information Form and Submittal Register. These additional contractual deliverables are described in the paragraphs below.



Since information about the overall scope of the project is available through COBie, either through attribute information, or by counting objects, areas, and volumes a significant portion of the Project Information Form data often required at this stage of design is available. Automated reports from the design product, or directly from the COBie data, can provide the first draft of this report.

A portion of the submittal register may be directly created from COBie design files through the identification of a template for each product of category. The COBie.Document entity allows the identification of “required” documents. The list of each of these documents is, in fact, the submittal register. Beyond the requirement for one COBie.Document of type “Product Data” for each COBie.Type, the COBie.Type category value may be used to create a standard set of submittals for different classes of products. Products whose category is flow control devices may require the submission of test report, while products categorized as metal doors may not require (by default) an test report.

#### **4.2.5.2.4.11 Finalize design – design final**

In addition to the discussion of design deliverable coverage noted in the previous section, the coordinated design deliverable includes deliverables such as Quality Control Data, Color Documentation Binder, Code Compliance Certification, and Construction Specifications. These additional contractual deliverables are described in the paragraphs below.

As with the automated production of a portion of the submittal register, entries from COBie.Types, their categories and attributes could be used with a template of Quality Control standards to produce a draft quality control specification and/or plan for the project. While such an application does not currently exist, the development of templates based on national product type classifications is within a realm of possible applications to automate the production of contract deliverables based on COBie data.

Information about color selections from the COBie data can be used to provide the outline for the Color Documentation Binder. Specific Color chips could be provided as attached COBie.Documents further automating the process. Photographs of fabric and texture samples should be required for project documentation and these may be included as COBie.Document entries. While the owner must be able to physically have a copy of the colors and samples, the electronic record document maybe completely generated from a simple report based on COBie.Data.

#### **4.2.5.2.4.12 Prepare invitation to bid and receive proposals (post design)**

The deliverable for this stage of the project is request for proposal package. The inclusion of a construction-documents stage COBie file, for information only, would ensure that bidders have a more even basis for their bids since individual companies would consistently count information provided in electronic versus drawing or native formats.

Concerns of liability are moot since the COBie data matches the schedule data on the drawings. If there are errors, then the normal error resolution process proceeds.

#### **4.2.5.2.4.13 Respond to pre-proposal inquiries**

The capture of requests for information or clarifications may be directly accomplished through the COBie data format. The use of this format actually enhances the value of the information and improves the speed of processing the inquiry. The value of the information is increased since such information becomes linked directly with the objects in question. The speed of inquiry may be improved since the meta-data contained in the COBie-related information package may be used to automatically route the object to the appropriate party responsible to answer the question.

#### **4.2.5.2.4.14 Develop pre-construction plan**

There are many deliverables required of the construction contractor during the pre-construction phase. The paragraphs below describe the way in which COBie data may be used to support and streamline such processes. Such streamlining is possible since the contractor producing the deliverable can use reports from COBie construction documents stage files to fill in much of the content of that deliverable.

The first set of information provided by contractors is often the lists of companies and persons needing access to the job site. This list, if provided in COBie.Contacts, allows the capture of information that may be used through the rest of the construction project. As installers capture nameplate data, or commissioning agents identify punch-list items, the names of each of these persons will be available through COBie pick-lists developed from project access control lists that are updated from COBie.Contacts as the project proceeds.

While COBie data will not directly contribute to the creation of the contractors Critical Path Method schedule, links between the building assets to be installed in a given activity and their upstream quality control and supply chain activities, may be linked to schedule activities. Such an capability, available as a result of having the COBie data, allows the contractor to integrate data systems ensuring that work on the project can be accomplished without any of the information falling “between the cracks.” For example setting up the scheduling software to produce equipment installation worksheets the equipment is to be installed, and directing them to associated subcontractor personnel would directly support the ability to evaluate the completion of specific schedule activities and the percentage of equipment installed on a given activity.

Given the zones, systems, and equipment being installed on the project some aspects of Health and Safety Plans, Work Plans, Quality Control Plans and Environmental Protection plans may be developed from COBie data. This would be accomplished by having templates for specific classifications of work. When that work is identified in a COBie file that portion of the template is used to create the project-specific plan.

The paragraphs above identify pre-construction deliverables that are likely to be supported by COBie-based data extracts are identified. There are also several pre-construction deliverables that are outside the scope of workflows associated with COBie. These include Certificates of Insurance, Bonds, Schedule of Prices, and Contractors’ equipment lists.

#### **4.2.5.2.4.15 Identify discrepancies**

Similar to coverage noted in Respond to Pre-Proposal Inquiries.

#### **4.2.5.2.4.16 Prepare submittal information – product type selection**

A variety of information related to submittals is either directly or indirectly, captured in COBie. The majority of this information can be linked to COBie data sets through the COBie.Document entity. Such information includes: Product Data, Design Data, Test Reports, Certifications, Manufacturers’ Instructions, Manufacturers’ Field Reports, and Operations and Maintenance Data.

Although it is not common practice today, several types of submittal data may provided in a data-oriented format using discrete COBie data exchanges either through SpreadsheetML or COBieLite versions directly from product manufacturers. That information includes Product Data and Operations and Maintenance Data. Work on other projects such as the SPie and LCie project will provide specific content and format specifications for such direct information exchanges.

**4.2.5.2.4.17 Prepare submittal information – system layout**

System layout drawings were originally envisioned to be able to be provided through COBie, however, additional work on the HVACie, WSie, Sparkie, and BAMie projects have indicated that the best method for the exchange of system layout information are domain-specific Model View Definitions. A COBie extract of the managed assets contained within that specific domain system, for system manuals would however be possible to produce.

**4.2.5.2.4.18 Organize submittal information**

The preparation and transmission of submittals is outside the context of COBie. While COBie-based data may be the content of such information exchanges, LCie defines an example data structure for the exchange of such information.

**4.2.5.2.4.19 Perform submittal review – submittal issue**

Similar to coverage noted in Respond to Pre-Proposal Inquiries.

**4.2.5.2.4.20 Provide resources**

The production of purchase orders as a result of submittal approvals is outside the context of the COBie. While COBie-based data may be the content of such information exchanges, AGC-XML defines an example data structure for the exchange of such information.

**4.2.5.2.4.21 Execute construction activities**

During the process of building construction a number of specific contract deliverables are required. Some of these deliverables are supported directly, or indirectly, by the use of COBie. The production of drawings, such as as-built drawings and as-built shop drawings are not supported by COBie. The scheduled asset information on these drawings does, however, directly translate to COBie.

As assets are installed on the project the installed equipment list deliverable, used to support contractor payment requests, maybe directly represented in COBie format. The specific subset of the COBie schema applicable for such an exchange are defined in the LCie for managed asset specification.

**4.2.5.2.4.22 Perform equipment testing**

Equipment start-up and performance reports may be linked to COBie data. In addition, the start-up, or energization, date for a given piece of equipment is to be documented in COBie as the COBie.Component.WarrantyStartDate. The reports themselves may be linked to the specific COBie.Component through the COBie.Document entity.

**4.2.5.2.4.23 Inspect and approve work**

Similar to coverage noted in Section 5.2.3.2.13 Respond to Pre-Proposal Inquiries. Reports developed at this stage are compiled sets of COBie.Issues.

**4.2.5.2.4.24 Define, record and certify discrepancies**

Similar to coverage noted in Section 5.2.3.2.13 Respond to Pre-Proposal Inquiries. Reports developed at this stage are compiled sets of COBie.Issues.

#### 4.2.5.2.4.25 Closeout

Given that COBie was specifically designed to capture the construction handover package all of the contents of the turnover package are either included, or referenced, within the COBie data structure. The paragraphs below describe each of these handover package contents and where such information is found within the COBie format.

Operations and Maintenance manuals produced by manufacturers may be included in COBie in one of three methods. First, an electronic copy of the manufacturer's booklet containing a product's manuals may be referenced in the COBie.Document entity and referenced back to the associated COBie.Type entity. Second, the set of activities for a given task may be directly included in the COBie.Job worksheet. Finally, if the individual activities within the entire set activities require different tools, materials, or training to accomplish, then the individual activities may be listed separately and linked to the needed COBie.Resource. It is the expectation that when product manufacturers begin to provide product data in SPie format, that these companies will provide their O&M manuals in COBie format as well.

Documentation of installed equipment also called, Record of Designated Equipment, can all be provided in COBie format. The COBie deliverable completely replaces this deliverable. Associated Material Data Files and Material Safety Data Sheets (MSDS) files may be provided in PDF format and linked to the COBie.Type through the COBie.Document tab.

Much of the information found in the Commissioning report directly relates to COBie data. Specific tests an evaluation of associated design narratives can be directly captured in COBie format. Information not conducive to capturing COBie format maybe captured in PDF files and directly linked to the associated Zone, System, or Component.

If paper copies of final record copies of the as-built drawings complete set of contract documents with all modifications, and shop drawings, are not required these may be provided in PDF format and linked through the COBie.Document tab.

#### 4.2.5.3 Process models formatting

The Business Process Modelling Notation (BPMN) template for MS Visio was used to create the process models found earlier in this section. The production of these models contained two enhancements. First, a swim-land to identified information exchanges has been added to simplify the diagram and emphasize that its primary purpose to identify information exchanges. Second, the level of detail and legend used for the NBIMS-US™ V3 COBie standard directly supports the economic evaluation of business process changes.

#### 4.2.6. Exchange requirement documentation

*Author's Editorial Comment: The NBIMS-US™ V3 COBie standard updates the information provided in NBIMS-US™ V2. In some cases new information is also provided as required by the NBMS-US™ V3 submission criteria. The following errors in NBIMS-US™ V2 were corrected in this version of the standard:*

- Zones use IfcZone rather than IfcSpatialZone to ease compatibility with IFC2x3.
- Task mappings changed to use IfcRecurrencePattern.Interval
- Product dimensions changed to use custom property set rather than IfcBoundingBox, as the nominal dimensions are not necessarily the same as the calculated geometric extents, and excluding geometry from COBie simplifies usage.

- Product material mappings changed to use IfcMaterialConstituentSet.
- File mappings now use IfcProxy (in addition to IfcRelAssociatesDocument) to capture metadata consistently and support documents without associated objects. Such documents must be associated with the overall Facility object.

*Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange submissions, the information provided in this section should be considered a moderate change.*

#### 4.2.6.1 Exchange requirements legibility

##### 4.2.6.1.1 Exchange requirements list

- Facility Criteria
- Discipline Specifications
- Project Definition
- Space Program
- Product Program
- Design Early
- Design Schematic
- Design Coordinated
- Design Issue
- Product Type Template
- Product Template
- Bid Issue
- Product Type Selection
- System Layout
- Product Installation
- Product Inspection
- Construction Issue
- Product Type Parts
- Product Type Warranty
- Product Type Maintenance
- System Operation
- Space Condition
- Product Parts Replacement
- Space Occupancy
- Space Activity Renovation
- Remodel
- Expand
- Demolish

##### 4.2.6.1.2 Exchange requirement classification list

**Table 30 Classification of Initiation Exchanges**

Project Phase	Contracting Phase	Contracted Exchanges	OmniClass Table 31–Phase	OmniClass Table 34–Actor	OmniClass Table 32–Service	buildingSMART alliance® Draft Tree
Criteria		Facility Criteria	not provided	34-21 14 00 Owner	not provided	2.44 Develop Site Criteria
		Discipline Specifications	not provided	34-21 14 00 Owner	not provided	2.43 Develop Design Criteria
Initiation	Definition	Project Definition	31-10 11 14 Description Phase	34-21 14 00 Owner	32-11 11 15 Development	2.5 Develop Project Execution Plan
Requirement		Space Program	31-10 14 21 Programming Phase	34-21 17 00 Planner	32-11 14 24 Programming	3.413 Configure Layout of Rooms and Zones

Project Phase	Contracting Phase	Contracted Exchanges	OmniClass Table 31–Phase	OmniClass Table 34–Actor	OmniClass Table 32–Service	buildingSMART alliance® Draft Tree
		Product Program	31-10 14 21 Programming Phase	34-21 17 00 Planner	32-11 14 24 Programming	3.41 Perform Systems Development and Layouts

**Table 31 Classification of Design Exchanges**

Project Phase	Contracting Phase	Contracted Exchanges	OmniClass Table 31–Phase	OmniClass Table 34–Actor	OmniClass Table 32–Service	buildingSMART alliance® Draft Tree
Design	Documents	Design Early	31-20 10 14 Conceptual Design	34-25 21 00 Architect	32-11 14 00 Designing	3.4 Develop Design
		Design Schematic	31-20 10 17 Schematic Design	34-25 21 00 Architect	32-11 14 00 Designing	3.4 Develop Design
		Design Coordinated	31-20 20 11 Detailed Design	34-25 21 00 Architect	32-11 14 00 Designing	3.4 Develop Design
		Design Issue	31-20 20 21 Engineering Analysis	34-25 21 00 Architect	32-11 14 00 Designing	3.52 Perform Design Reviews
	Specification	Product Type Template	31-20 20 24/37/31 Prod, Matl.Equip Sel	34-25 41 00 Specifier	32-11 45 00 Specifying	3.412 Select Building Materials and Equipment
		Product Template	31-20 20 24/37/31 Prod, Matl.Equip Sel	34-25 41 00 Specifier	32-11 45 00 Specifying	3.412 Select Building Materials and Equipment

**Table 32 Classification of Construction Exchanges**

Project Phase	Contracting Phase	Contracted Exchanges	OmniClass Table 31–Phase	OmniClass Table 34–Actor	OmniClass Table 32–Service	buildingSMART alliance® Draft Tree
Construction	Bidding	Bid Issue	31-30 30 21 Proposal Preparation	34-35 14 00 Contractor	32-21 21 11 Bidding	1.43 Prepare Bid and Submit
	Selection	Product Type Selection	31-40 20 27 Submittal Processing	34-35 14 00 Contractor	32-21 00 00 Execution Services	4.2 Provide Resources (Goods and Services)
		System Layout	31-40 20 27 Submittal Processing	34-35 14 00 Contractor	32-21 00 00 Execution Services	4.11 Develop Construction Plan
	Installation	Product Installation	31-40 40 11 17 Installation	34-35 14 00 Contractor	32-21 17 41 Installing	4.3 Build Building
		Product Inspection	31-40 40 91 17 Evaluation	34-35 14 00 Contractor	32-21 17 00 Constructing	4.34 Inspect and Approve Work
		Construction Issue	31-40 40 91 17 Evaluation	34-21 14 00 Owner	32-21 17 00 Constructing	4.34 Inspect and Approve Work
		Product Type Parts	31-40 50 00 Commissioning	34-35 17 00 Sub Contractor	32-21 00 00 Execution Services	4.4 Perform Commissioning
	Comission	Product Type Warranty	31-40 50 00 Commissioning	34-35 17 00 Sub Contractor	32-21 00 00 Execution Services	4.4 Perform Commissioning
		Product Type Maintenance	31-40 50 00 Commissioning	34-35 17 00 Sub Contractor	32-21 00 00 Execution Services	4.4 Perform Commissioning
		System Operation	31-40 50 00 Commissioning	34-35 17 00 Sub Contractor	32-21 00 00 Execution Services	4.4 Perform Commissioning

**Table 33 Classification of Operational Exchanges**

Project Phase	Contracting Phase	Contracted Exchanges	OmniClass Table 31–Phase	OmniClass Table 34–Actor	OmniClass Table 32–Service	buildingSMART alliance® Draft Tree
O & M		Space Condition	31-50 20 21 Facility Inspection	34-41 11 00 Facility Manager	32-41 51 11 Inspecting	5.3 Evaluate Conditions and Detect Problems
		Product Parts Re-placement	31-50 20 11 Facility Operation	34-41 21 00 Maintenance	32-41 47 11 11 Facility Repairing	5.4 Develop Solutions
		Space Occupancy	31-50 10 17 Use	34-41 11 00 Facility Manager	32-41 47 21 Space Planning	5.7 Perform Use of Facility/Building
		Space Activity Renovation	31-50 30 11 Facility Renovation	34-41 11 00 Facility Manager	32-41 47 21 Space Planning	5.8 Perform Facility/ Building Renovation
Repurpose		Remodel	31-50 30 14 Facility Remodeling	34-41 11 00 Facility Manager	32-41 47 11 17 Facility Upgrading	5.10 Perform Facility/ Building Expansion
		Expand	31-50 30 17 Facility Expansion	34-41 11 00 Facility Manager	32-41 47 11 17 Facility Upgrading	5.10 Perform Facility/ Building Expansion
		Demolish	31-60 35 00 Recycling	34-41 11 00 Facility Manager	32-41 47 11 17 Facility Upgrading	5.11 Disposal, Reuse, Recycling

#### 4.2.6.1.3 Exchange Requirement Coverage Analysis

**Table 34 Exchange Requirements Coverage**

Exchange	Process	Sender	Receiver
Facility Criteria	31-10 00 00 Inception Phase	34-10 11 00 Owner	34-10 11 11 Developer
Discipline Specifications	31-10 00 00 Inception Phase	34-10 11 17 Public Entity	34-10 11 11 Developer
Project Definition	31-20 00 00 Conceptualization Phase	34-10 11 11 Developer	34-20 21 11 Planner
Space Program	31-30 00 00 Criteria Definition Phase	34-10 11 11 Developer	34-20 11 11 Architect
Product Program	31-30 00 00 Criteria Definition Phase	34-10 11 11 Developer	34-20 11 11 Architect
Design Early	31-40 00 00 Design Phase	34-20 11 11 Architect	34-10 11 11 Developer
Design Schematic	31-40 00 00 Design Phase	34-20 11 11 Architect	34-10 11 11 Developer
Design Coordinated	31-40 00 00 Design Phase	34-20 11 11 Architect	34-20 11 21 Engineer
Design Issue	31-40 00 00 Design Phase	34-20 11 21 Engineer	34-20 11 11 Architect
Product Type Template	31-40 00 00 Design Phase	34-20 11 11 Architect	34-20 11 41 Specifier
Product Template	31-40 00 00 Design Phase	34-20 11 41 Specifier	34-20 11 11 Architect
Bid Issue	31-60 00 00 Implementation Phase	34-35 15 11 Contractor	34-20 11 11 Architect
Product Type Selection	31-60 00 00 Implementation Phase	34-35 10 21 Supplier	34-20 11 11 Architect
System Layout	31-60 00 00 Implementation Phase	34-35 15 11 Contractor	34-20 11 21 Engineer
Product Installation	31-60 00 00 Implementation Phase	34-35 15 11 Contractor	34-20 11 11 Architect
Product Inspection	31-60 00 00 Implementation Phase	34-20 51 11 Reviewer	34-35 10 21 Supplier
Construction Issue	31-60 00 00 Implementation Phase	34-20 51 17 Inspector	34-35 15 11 Contractor
Product Type Parts	31-60 00 00 Implementation Phase	34-35 10 21 Supplier	34-10 11 00 Owner

Exchange	Process	Sender	Receiver
Product Type Warranty	31-60 00 00 Implementation Phase	34-35 10 21 Supplier	34-10 11 00 Owner
Product Type Maintenance	31-60 00 00 Implementation Phase	34-35 10 21 Supplier	34-10 11 00 Owner
System Operation	31-60 00 00 Implementation Phase	34-35 15 11 Contractor	34-10 11 00 Owner
Space Condition	31-80 00 00 Operations Phase	34-41 14 11 Facility Manager	34-10 11 00 Owner
Product Parts Replacement	31-80 00 00 Operations Phase	34-41 14 11 Facility Manager	34-10 11 00 Owner
Space Occupancy	31-80 00 00 Operations Phase	34-41 14 11 Facility Manager	34-10 11 00 Owner
Space Activity Renovation	31-80 00 00 Operations Phase	34-41 14 11 Facility Manager	34-10 11 00 Owner
Remodel	31-90 00 00 Closure Phase	34-10 11 00 Owner	34-20 11 11 Architect
Expand	31-90 00 00 Closure Phase	34-10 11 00 Owner	34-20 11 11 Architect
Demolish	31-90 00 00 Closure Phase	34-10 11 00 Owner	34-35 15 11 Contractor

#### 4.2.6.2 Exchange requirements detail

##### 4.2.6.2.1 Exchange requirements definition

###### 4.2.6.2.1.1 Criteria - facility criteria

This exchange includes high-level criteria specific to the building to be constructed, but without regard for particular disciplines. Entities exported for this exchange include the following:

- IfcProject: Directory and context of data is provided.
- IfcActor: Project participants, roles, and contact information is provided.
- IfcSite: Site (General) information must be provided to identify the site upon which the facility is placed.
- IfcBuilding: Facility (General) information must be provided to identify the facilities in which floors and spaces may be found.
- IfcBuildingStory: Provide the name of each conceptual/physical vertical level of each of the facilities identified in the project wrapper, elevation value and units above local project datum, floor to Floor height and units.
- IfcSpace: Provide the name and floor for each space.
- IfcSpatialZone: Identify conceptual zones within the building that have a similar or grouped pattern of use. For example, zones may refer to mechanical heating zones, alarm systems zones, or groups of spaces to be used for specific clients.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

###### 4.2.6.2.1.2 Criteria - discipline specifications

This exchange includes discipline-specific criteria, indicating scope and function of building systems and distribution systems. Entities exported for this exchange include the following:

- IfcProject: Directory and context of data is provided.
- IfcActor: Project participants, roles, and contact information is provided.



- IfcSystem: Building systems and distribution systems are to be identified and specified.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### 4.2.6.2.1.3 Project initiation - project definition

This exchange includes initial project information to describe a project and its contents. Entities exported for this exchange include the following:

- IfcProject: Directory and context of data is provided.
- IfcActor: Project participants, roles, and contact information is provided.
- IfcSite: Site (General) information must be provided to identify the site upon which the facility is placed.
- IfcBuilding: Facility (General) information must be provided to identify the facilities in which floors and spaces may be found.

#### 4.2.6.2.1.4 Requirements - space program

This exchange includes requirements for the spatial layout of a building. Entities exported for this exchange include the following:

- IfcBuildingStory: Provide the name of each conceptual/physical vertical level of each of the facilities identified in the project wrapper, elevation value and units above local project datum, floor to floor height and units.
- IfcSpace: Provide the name and floor for each space.
- IfcSpatialZone: Identify conceptual zones within the building that have a similar or grouped pattern of use. For example, zones may refer to mechanical heating zones, alarm systems zones, or groups of spaces to be used for specific clients.
- IfcRelAssociatesClassification: Classifications may be applied to categorize spaces by function.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### 4.2.6.2.1.5 Requirements – product program

This exchange includes requirements for physical components of a building. Entities exported for this exchange include the following:

- IfcElement: Identify general requirements of products to be used within the building.
- IfcRelAssociatesClassification: Classifications may be applied to categorize products by function.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### 4.2.6.2.1.6 Design stage - design early

This exchange includes building layout information and allocation of products without regard for placement. Entities exported for this exchange include the following:

- IfcBuildingStory: Provide the name of each conceptual/physical vertical level of each of the facilities identified in the project wrapper, elevation value and units above local project datum, floor to Floor height and units.

- IfcSpace: Provide the name and floor for each space.
- IfcElement: Identify general requirements of elements within the building.
- IfcElementType: Identify specific product model types to be considered for use.
- IfcSystem: System name for each system must be identified before it can be associated with specific products and zones.
- IfcRelAssociatesClassification: Classifications may be applied to categorize spaces by function.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### 4.2.6.2.1.7 Design stage - design schematic

This exchange includes building layout information and allocation of products with placement and connectivity. Entities exported for this exchange include the following:

- IfcSpace: Provide the name and floor for each space.
- IfcElement: Identify general requirements of elements within the building.
- IfcElementType: Identify specific product model types to be considered for use.
- IfcSystem: System name for each system must be identified before it can be associated with specific products and zones.
- IfcRelConnectsElements: Indicate mounting connections between elements.
- IfcRelConnectsPorts: Indicate port connections between elements.
- IfcRelAssociatesClassification: Classifications may be applied to categorize products by function.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### 4.2.6.2.1.8 Ddesign stage - design coordinated

This exchange includes building layout information and allocation of products with placement, connectivity, and assignment to systems and project participants. In IFC4, zones may be indicated spatially with functional designation, material constituents are supported to identify multiple materials and properties. Entities exported for this exchange include the following:

- IfcSpace: Provide the name and floor for each space.
- IfcElement: Identify general requirements of elements within the building.
- IfcElementType: Identify specific product model types to be considered for use.
- IfcSystem: Design building systems, structural systems, and distribution systems.
- IfcSpatialZone: Design zones for distribution systems.
- IfcRelConnectsElements: Indicate mounting connections between elements.
- IfcRelConnectsPorts: Indicate port connections between elements.
- IfcRelAssociatesMaterial: Indicate materials, layers, and profiles of elements.
- IfcRelAssignsToActor: Indicate project participants responsible for systems and elements.
- IfcRelAssignsToGroup: Indicate partitioning of building elements to systems.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

- IfcRelAssociatesClassification: Classifications may be applied to categorize products and systems by function.
- IfcElementQuantity: Quantities may be indicates on building elements and spatial structures.

#### **4.2.6.2.1.9 Design stage - design issue**

This exchange includes final design information with formal documents, and amended with design issue requests and responses. Entities exported for this exchange include the following:

- IfcActionRequest: Indicate reported issues and resolutions.
- IfcRelAssociatesDocument: Indicate external documents.

#### **4.2.6.2.1.10 Design Stage - Product Type Template**

This exchange includes definitions of properties to be captured by product templates.

Property templates are now supported in IFC4. Project composition excluded (no placeholder building is necessary). Entities exported for this exchange include the following:

- IfcRelAssociatesClassification: Classifications may be applied to categorize properties by common definitions.
- IfcPropertySetTemplate: Property set templates may be specified to indicate information to be provided for product types.
- IfcSimplePropertyTemplate: Property templates may be specified to describe each property within a property set.

#### **4.2.6.2.1.11 Design stage - product template**

This exchange includes product type information for specific product models provided by manufacturers.

Product types may be declared within a project without having any occurrences in IFC4. Project composition excluded (no placeholder building is necessary). Project libraries added (to provide link to originating property set templates). Entities exported for this exchange include the following:

- IfcProjectLibrary: Referenced libraries of property set templates are indicated.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.
- IfcElementType: Identify specific product model types to be considered for use.
- IfcRelAssociatesClassification: Classifications may be applied to categorize product types.
- IfcRelAssociatesMaterial: Identify material constituents and material properties.

#### **4.2.6.2.1.12 Construction stage - bid issue**

This exchange includes bid submission information with formal documents, and amended with bid issue requests and responses. The IfcActionRequest entity captures all reported issues (or requests to do something), for which tasks may be assigned to carry out work to address an issue. Entities exported for this exchange include the following:

- IfcActionRequest: Indicate reported issues and resolutions.

- IfcRelAssociatesDocument: Indicate external documents.

#### **4.2.6.2.1.13 Construction stage - product type selection**

This exchange includes product type information for product models selected to be used, without regard for particular placement. Product types may be declared within a project without having any occurrences in IFC4. Project composition excluded (no placeholder building is necessary). Project libraries added (to provide link to originating property set templates). Entities exported for this exchange include the following:

- IfcProjectLibrary: Referenced libraries of product types are indicated.
- IfcElementType: Identify specific product model types to be considered for use.
- IfcRelDefinesByType: Indicate product types to be used at specific occurrences.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### **4.2.6.2.1.14 Construction Stage - System Layout**

This exchange includes detailed system connectivity information for building systems and distribution systems. Entities exported for this exchange include the following:

- IfcElement: Indicate specific element placement within the building.
- IfcSystem: Indicate system design and topology.
- IfcRelConnectsElements: Indicate general connections between elements.
- IfcRelConnectsPorts: Indicate port connections between elements.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### **4.2.6.2.1.15 Construction stage - product installation**

This exchange includes product placement information, including serial numbers at specific installations. Entities exported for this exchange include the following:

- IfcElement: Identify general requirements of elements within the building.
- IfcPropertySet: Properties may be specified to indicate requirements at particular objects.

#### **4.2.6.2.1.16 Construction stage - product inspection**

This exchange includes product inspection issues reported, which may require replacement of installed products. Entities exported for this exchange include the following:

- IfcActionRequest: Report issues related to products.

#### **4.2.6.2.1.17 Construction stage – construction issue**

This exchange includes construction issues reported, which may require additional labor. Entities exported for this exchange include the following:

- IfcActionRequest: Report issues related to construction.

#### **4.2.6.2.1.18 Construction stage – product type parts**

This exchange includes product part information, which may be used for addressing components for connectivity or replacement. IFC4 allows product types to have assigned process types and resource types, which may indicate standard processes (i.e. manufacturer-defined) for servicing or replacing parts. Entities exported for this exchange include the following:

- IfcTaskType: Indicate product part replacement processes.
- IfcConstructionProductResourceType: Indicate available resources and costs for part replacement.

#### **4.2.6.2.1.19 Construction stage – product type warranty**

This exchange includes product warranty information for parts and labor. Entities exported for this exchange include the following:

- IfcTaskType: Indicate warranty periods.
- IfcSubContractResourceType: Indicate warranty contracts.
- IfcRelAssociatesDocument: Indicate external documents for warranties.

#### **4.2.6.2.1.20 Construction stage – product type maintenance**

This exchange includes product maintenance information, including expected maintenance tasks and procedures. Entities exported for this exchange include the following:

- IfcTask: Schedule product maintenance processes.
- IfcConstructionResource: Schedule product maintenance resources.
- IfcRelAssociatesDocument: Indicate external documents for maintenance.

#### **4.2.6.2.1.21 Construction stage – system operation**

This exchange includes system operation information, including system operation procedures and events. Entities exported for this exchange include the following:

- IfcTask: Schedule system operation processes.
- IfcConstructionResource: Schedule system operation resources.
- Operations & Maintenance

#### **4.2.6.2.1.22 O&M stage – space condition**

This exchange includes reporting on the condition of spaces over time. In IFC4, all time-phased information is captured using performance-based properties on IfcPerformanceHistory. This allows for data to be recorded for multiple time periods (avoiding the single property set restriction), and to provide a

uniform way of accessing and rendering time-phased data such that software applications need not be aware of particular properties. A property set should be defined for recording space conditions over time. Entities exported for this exchange include the following:

- IfcPropertySet: Properties may be specified to indicate condition details.

#### **4.2.6.2.1.23 O&M stage – product parts replacement**

This exchange includes reporting on the replacement of product parts over time. In IFC4, the process and resource model has been formalized, such that replacement of parts may be considered as a MAINTENANCE task with assigned resources for materials, labor, equipment, products, crews, subcontracts. Entities exported for this exchange include the following:

- IfcTask: Report parts replacement over time.
- IfcConstructionResource: Report parts replacement resources.
- IfcPropertySet: Properties may be specified to indicate details.

#### **4.2.6.2.1.24 O&M stage – space occupancy**

This exchange includes scheduling occupancy of spaces over time. Entities exported for this exchange include the following:

- IfcOccupant: Indicate occupants.
- IfcTask: Indicate scheduled occupancy periods.
- IfcSubContractResource: Indicate leases for space occupancy.
- IfcRelAssignsToActor: Indicate current owner(s) and/or occupant(s) of a space.
- IfcRelAssignsToProduct: Indicate occupancy periods of spaces.
- IfcRelAssignsToProcess: Indicate resources assigned to space occupancy periods.
- IfcPropertySet: Properties may be specified to indicate details.

#### **4.2.6.2.1.25 O&M stage – space activity renovation**

This exchange includes scheduling reconfiguration of spaces over time. Entities exported for this exchange include the following:

- IfcTask: Indicate renovation tasks.
- IfcConstructionResource: Indicate renovation resources.
- IfcRelAssignsToProduct: Indicate elements to be renovated.
- IfcRelAssignsToProcess: Indicate resources assigned to tasks.
- IfcPropertySet: Properties may be specified to indicate details.

#### **4.2.6.2.1.26 Recycle stage – remodel**

This exchange includes changing the building layout of an existing structure. Entities exported for this exchange include the following:

- IfcTask: Indicate renovation tasks.
- IfcConstructionResource: Indicate renovation resources.
- IfcRelAssignsToProduct: Indicate elements to be renovated.
- IfcRelAssignsToProcess: Indicate resources assigned to tasks.
- IfcPropertySet: Properties may be specified to indicate details.

#### **4.2.6.2.1.27 (not used)**

#### **4.2.6.2.1.28 Recycle stage – expand**

this exchange includes expanding the building layout of an existing structure. entities exported for this exchange include the following:

- IfcTask: Indicate renovation tasks.
- IfcConstructionResource: Indicate renovation resources.
- IfcRelAssignsToProduct: Indicate elements to be renovated.
- IfcRelAssignsToProcess: Indicate resources assigned to tasks.
- IfcPropertySet: Properties may be specified to indicate details.

#### **4.2.6.2.1.29 Recycle stage – demolish**

This exchange includes demolishing an existing structure partially or in full. Entities exported for this exchange include the following:

- IfcTask: Indicate renovation tasks.
- IfcConstructionResource: Indicate renovation resources.
- IfcRelAssignsToProduct: Indicate elements to be renovated.
- IfcRelAssignsToProcess: Indicate resources assigned to tasks.
- IfcPropertySet: Properties may be specified to indicate details.

#### **4.2.6.2.1.30 Use of assemblies within COBie**

To simplify interoperability, part composition is excluded in this MVD. Work related to the inclusion of assemblies has, however, been mapped to support related MVD's for HVAC, water, and electrical systems. Use of Assembly information for NBIMS-US™ V3 must be specified as a supplemental implementers' agreement, only if required by contract.

#### **4.2.6.2.1.31 Use of connections within COBie**

To simplify interoperability, element and port connectivity was excluded in this MVD. Work related to the inclusion of ports has, however, been mapped to support related MVD's for HVAC, water, and electrical systems. Use of Connection information for NBIMS-US™ V3 must be specified as a supplemental implementers' agreement, only if required by contract.

#### **4.2.6.2.1.32 Use of impacts within COBie**

Proposed IFC 4 TC1 extensions may include PSet\_EnvironmentalImpact. Once that extension has been developed, and mapping back to IFC 2x3 have been accomplished, implementers' agreements may support the use of Impact information.

#### 4.2.6.2.1.33 Use of coordinates within COBie

Exchange requirements governing COBie described in this standard identify the relationship between asset information and spatial geometry simply that of "spatial containment." Spatial containment means that each individually named asset shall be identified as being "in" one space. When equipment is in a single space, as is the case with the vast majority of equipment with a facility, this requirement needs no clarification. For the special case of large equipment placed in the ceiling, spanning several spaces, the space associated with the equipment the one from which the equipment is most typically maintained. There are yet other situations, such as equipment located on a large geographic site where some geometric placement information may be helpful.

Given the very limited ability of COBie, in spreadsheet view, to represent only points, lines, and boxes, COBie is not able to provide any significant geometric representation. COBie may, in special cases, however, assist in identifying the location of equipment. To support these special cases, the following entities may, optionally, have the following Coordinate representations in COBie.

**Table 35 Use of Coordinates with COBie**

Entity	Coordinate Type	Required value
Facility	Lower-Left Corner	Typically set to 0,0,0
Floor	Lower-Left Corner	Relative to Facility
Space	Lower-Left & Upper-Right Corner	Relative to Floor
Component	Lower-Left & Upper-Right Corner	Relative to Floor
Assembly	Lower-Left & Upper-Right Corner	Relative to Floor

Use of Coordinate information for NBIMS-US™ V3 must be specified as a supplemental implementers' agreement, only if required by contract.

#### 4.2.6.2.1.34 Use of attributes within COBie – mapping between IFC and SpreadsheetML

Implementers agreements documented in buildingSMART alliance® Challenge events have separately document the updates needed for custom property sets that are provided separately from mandatory standards. These implementers agreements, found in the COBie responsibility matrix, have resolved implementation issues to ensure that the quality COBie data can be demonstrated to match the same data found on design drawing schedules and manufacturers data sheets. The standardization of specific property sets is outside the scope of the COBie effort.

#### 4.2.6.2.1.35 Use of attributes within COBie – product data

Attributes sets for standard equipment types have not been submitted for consensus review during NBIMS-US™ V3. In the absence of NBIMS-US™ standard the COBie Guide, a published commentary, states that the attributes found in a COBie file must match those product data properties found on the associated contract deliverables.

#### 4.2.6.2.2 Business rule list

The business rules verify that exchange of COBie data conform to the required data specification can be generalized into rule classes. Application of these rules results in the verification that the internal integrity of a COBie data set is correct. These rules do not validate the COBie deliverable against contract-specific requirements. The list of COBie rule classes is provided in the list below.





# National BIM Standard - United States®

an initiative of the National Institute of Building Sciences buildingSMARTalliance®

- Classification
- Conversion Units
- COBie Metadata
- COBie Contact
- Control Assignment
- External Data Constraints
- Group Assignment
- Object Typing
- Project Declaration
- Property Sets
- Quantities on Occurrences
- Sequential Connectivity
- Spatial Composition
- Spatial Containment

While the specific content of information exchanges related to Spatial and Product assets differs as a project progresses, the business rules used to verify the integrity of the information are exactly the same for all such exchanges. The following information exchanges share the business rules noted in the table below:

- Facility Criteria Exchange
- Space Program
- Product Program
- Early Design
- Design Schematic
- Design Coordinated
- System Layout

**Table 36 Business Rules – Design-Related Exchanges**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets	Quantities on Occurrences	Spatial Containment
IfcSpace	x		x	x	x					x	x	
IfcBuildingStorey	x		x		x					x		
IfcBuilding	x			x	x					x		
IfcSite	x				x							
IfcProject	x		x			x	x					
IfcZone	x		x	x				x		x		
IfcSystem	x		x					x		x		
IfcElement	x		x	x					x	x		x
IfcElementType	x		x	x						x		
IfcActor	x	x	x									

The business rules associated with Discipline Specifications exchanges differ from those listed in the previous table since these discipline-specific exchanges are concerned with the technical specifications of product oriented assets. Note that the requirement for discipline specifications to have ifcBuilding, ifcSite, ifcProject entities is an artefact of the modelling of such information using the Industry Foundation Class Model only.

**Table 37 Business Rules – Discipline Specifications**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets	Quantities on Occurrences	Spatial Containment
IfcBuilding	x			x	x					x		
IfcSite	x				x							
IfcProject	x		x			x	x					
IfcZone	x		x	x				x		x		
IfcSystem	x							x		x		
IfcElement	x		x	x					x	x		x
IfcElementType	x		x	x						x		
IfcActor	x	x	x									

Business rules associated with product definition are a sub-set of those required to define the requirements for the program, and the design that is proposed to meet those requirements. Business rules associated with COBie files submitted at the Project Definition stage are noted in the table below.

**Table 38 Business Rules – Project Definition**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets
IfcSpace	x		x	x	x					x
IfcBuildingStorey	x		x		x					x
IfcBuilding	x			x	x					x
IfcSite	x				x					
IfcProject	x		x			x	x			
IfcZone	x		x	x				x		x
IfcSystem	x							x		x

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets
IfcElement	x		x	x						x
IfcElementType	x		x	x						x
IfcActor	x	x	x							

There are exchanges related to the identification and resolution of issues on the project. While the content and timing of these exchanges differ the following exchanges all share a common set of rules used to verify the internal consistency of the information exchange:

- Design Issue
- Bid Issue
- Product Inspection
- Construction Issue

**Table 39 Business Rules – Issue Type Exchanges**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets	Control Assignment	Quantities on Occurrences	Spatial Containment
IfcSpace	x		x	x	x					x		x	
IfcBuildingStorey	x		x		x								
IfcBuilding	x			x	x								
IfcSite	x				x								
IfcProject	x		x			x	x						
IfcZone	x		x	x				x		x			
IfcSystem	x							x		x			
IfcElement	x		x	x					x	x			x
IfcElementType	x		x	x						x			
IfcActor	x	x	x										
IfcActionRequest											x		

Many information exchanges that build to create COBie data pertain to exchanges about product types. This begins with an initial configuration of a generic product, to the identification of explicit product types, to the selection of possible candidates for that specification, and then through construction during the submittal process. The following product type exchanges share the business rules found in the table below:

- Product Type Template
- Product Type Selection
- Product Type Warranty
- Product Type Parts

IfcActor applies to all exchanges, except the initial exchange Product Type Template. IfcConstructionResourceType only applies to the Product Type Parts, exchange.

**Table 40 Business Rules – Product Type Exchanges**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Property Sets
IfcBuilding*	x			x	x			
IfcSite*	x				x			
IfcProject*	x		x			x	x	
IfcElementType	x	x	x	x				x
IfcActor	x	x	x					
IfcConstructionResourceType	x	x	X					

\* NOTE: The inclusion of this entity is required for complete specification of IFC-based file exchange; however, such information is not directly required by the exchange itself if the information is provided through a transactional or web-service exchange. The transactional specification for Product Type Exchanges may be found in the NBIMS-US™ V3 LCie standard.

The specifics of individual product instances, as defined by IfcElement entities are added for:

- Product Template
- Product Installation

**Table 41 Business Rules – Product Installation**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Object Typing	Property Sets	Spatial Containment
IfcSpace*	x		x	x	x					
IfcBuildingStorey*	x		x		x					
IfcBuilding*	x			x	x					
IfcSite*	x				x					
IfcProject*	x		x			x	x			
IfcSystem*	x									
IfcElement	x		x	x				x	x	x
ifcElementType*	x		x	x						
IfcActor	x	x	x							

\* NOTE: The inclusion of this entity is required for complete specification of IFC-based file exchange; however, such information is not directly required by the exchange itself if the information is provided through a transactional or web-service exchange, versus a COBie file-based exchange. The transactional specification for Product Type Exchanges may be found in the NBIMS-US™ V3 LCie standard.

Information about operations and maintenance activities introduces the entity of an ifcTask to the exchange requirement. Business rules pertaining to those tasks ensure appropriate connection of the tasks to their associated type objects and allow sequencing of operations within tasks. The following exchanges share the business rules listed in the table below:

- Product Type Maintenance
- System Operations

**Table 42 Business Rules – O&M Information Type Exchanges**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets	Sequential Connectivity	Quantities on Occurrences	Spatial Containment
IfcSpace	x		x	x	x							x	

IfcBuildingStorey	x		x		x								
IfcBuilding*	x			x	x								
IfcSite*	x				x								
IfcProject*	x		x			x	x						
IfcZone	x		x	x				x					
IfcSystem	x							x					
IfcElement	x		x	x					x	x			x
ifcElementType	x		x	x									
IfcTask				x							x		
IfcActor	x	x	x										
IfcConstruction Resource	x		X								x		

\* NOTE: The inclusion of this entity is required for complete specification of IFC-based file exchange; however, such information is not directly required by the exchange itself if the information is provided through a transactional or web-service exchange, versus a COBie file-based exchange. The transactional specification for Product Type Exchanges may be found in the NBIMS-US™ V3 LCie standard. Upon commencement of the operation of the facility information about service orders used to repair or replace individual parts of fixed assets can be captured in the following exchange. While the information in the underlying COBie data is not changed, information about the filters, lubricants, or replacement parts are exchanged in the ifcConstructionResource entity.

**Table 43 Business Rules – Product Parts Replacement**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets	Quantities on Occurrences	Spatial Containment
IfcSpace*	x		x	x	x						x	
IfcBuildingStorey*	x		x		x							
IfcBuilding*	x			x	x							
IfcSite*	x				x							
IfcProject*	x		x			x	x					
IfcZone*	x		x	x				x				
IfcSystem*	x							x				
IfcElement	x		x	x					x	x		x
ifcElementType*	x		x	x								
IfcActor	x	x	x									
IfcConstruction Resource	x		X							x		

\* NOTE: The inclusion of this entity is required for complete specification of IFC-based file exchange; however, such information is not directly required by the exchange itself if the information is provided through a transactional or web-service exchange, versus a COBie file-based exchange. The transactional specification for Product Type Exchanges may be found in the NBIMS-US™ V3 LCie standard.

Surveys of space condition and occupancy share a common data set. The exchanges using the following post-occupancy spatial exchanges are defined in the table below:

- Space Condition
- Space Occupancy

**Table 44 Business Rules - Post-occupancy Space Exchanges**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Property Sets	Quantities on Occurrences
IfcSpace	X		x	x	x				x	x
IfcBuildingStorey	X		x		x				x	
IfcBuilding	X			x	x				x	
IfcSite	X				x				x	
IfcProject	X		x			x	x		x	
IfcZone	X		x	x				x	x	
IfcActor	X	x	x							

Note that this definition of a spatial occupancy survey may be too narrow for some facility managers. If a given facility manager also collects information about product-based assets during their surveys, then the “Post-Occupancy Type Exchanges” table may be used.

Post-Occupancy exchanges allow a delivered facilities information to be kept up to date. Information exchanges related to specific work orders should be maintained within CMMS/CAFM systems as part of work order processing and are outside the larger scope of the project life-cycle identified in this standard. COBie defines several exchanges in which information about a part of a facility, or an entire facility, may be updated. The table below identifies the business rules associated with the exchange of the following post-occupancy information exchanges:

- Product Parts Replacement
- Space Activity Renovation
- Remodel

- Expand
- Demolish

**Table 45 Business Rules – Post-occupancy Space & Product Exchanges**

Entity	COBie Metadata	COBie Contact	External Data Constraints	Classification	Spatial Composition	Conversion Units	Project Declaration	Group Assignment	Object Typing	Property Sets	Quantities on Occurrences	Spatial Containment
IfcSpace	x		x	x	x					x	x	
IfcBuildingStorey	x		x		x							
IfcBuilding	x			x	x							
IfcSite	x				x							
IfcProject	x		x			x	x					
IfcZone	x		x	x				x				
IfcSystem	x							x		x		
IfcElement	x		x	x					x	x		x
ifcElementType	x		x	x						x		
IfcActor	x	x	x									
IfcConstruction Resource	x		X									

In addition to these business rules the following expected set of categorization is expected to be provided on COBie data files. For COBie data extracted from IFC models COBie transformation rules, defined in Annexes to these documents provide the default settings. While some classifications have been developed based on IFC type enumerations, these classifications represent the general practice within the United States. As such these are provided for information only as specific regions, owners, and contracts may require alternative values for such information. The listing below is presented in a format that corresponds to the order in which these fields are encountered within the COBie Spreadsheet version.

**Contact.Category** - If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 34.

**Facility.Category** - If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 11.

**Floor.Category** - If allowable values are not specified by contract, the default values are: “Floor”, “Roof”, and “Site”.

**Space.Category** - If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 13.



**Zone.Category** - If allowable values are not specified by contract, the default values are: "Circulation Zone", "Lighting Zone", "Fire Alarm Zone", "Historical Preservation Zone", "Occupancy Zone", and "Ventilation Zone".

**Type.Category** - If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 23.

**Type.AssetType** - If allowable values are not specified by contract, the default values are: "Fixed" and "Movable".

**Type.WarrantyDurationUnit** - If allowable values are not specified by contract, the default values are: "month" and "year".

**Type.DurationUnit** - If allowable values are not specified by contract, the default values are: "month" and "year".

**System.Category** - If allowable values are not specified by contract, the default value for this information is the current OmniClass Table 21.

**Assembly.AssemblyType** - If allowable values are not specified by contract, the default values are: "Fixed", "Optional", "Included", "Excluded", "Layer", "Patch", and "Mix".

**Connection.ConnectionType** - If allowable values are not specified by contract, the default values are: "Control", "Flow", "Return", "Supply", and "Structural".

**Spare.Category** - If allowable values are not specified by contract, the default values are: "Part", "PartSet", "Lubricant", "Other", "Spare", and "SpareSet".

**Resource.Category** - If allowable values are not specified by contract, the default values are: "Labor", "Material", "Tools", and "Training".

**Job.Category** - If allowable values are not specified by contract, the default values are: "Adjustment", "Calibration", "Emergency", "Inspection", "Operation", "PM", "Safety", "ShutDown", "StartUp", "Testing", and "Trouble".

**Job.Status** - If allowable values are not specified by contract, the default values are: "Not Yet Started", "Started", and "Completed".

**Impact.ImpactType** - If allowable values are not specified by contract, the default values are: "Cost", "ClimateChange", and "PrimaryEnergyConsumption".

**Impact.ImpactStage** - If allowable values are not specified by contract, the default values are: "Production", "Installation", "Maintenance", "Replacement", "Use", and "Reuse".

**Impact.ImpactStage** - If allowable values are not specified by contract, the default values are: "Production", "Installation", "Maintenance", "Replacement", "Use", and "Reuse".

**Impact.ImpactUnit** - If allowable values are not specified by contract, the default values are: "currency", "kgCO2e", "MJ", "other", or "n/a".

**Document.Category** - If allowable values are not specified by contract, the default values are: "Preconstruction Submittals", "Shop Drawings", "Product Data", "Samples", "Design Data", "Test Reports", "Certificates", "Manufacturer Instructions", "Manufacturer Field Reports", "Operation and Maintenance", "Closeout Submittals", "Contract Drawings", "Design Review Comment", "Specifications", "Request for Information", "Client Requirements", "Contract Specifications", "Contract Drawings", "Requests for Information", "Contract Modifications", and "Punch List Items".

**Document.ApprovalBy** - If allowable values are not specified by contract, the default values are: "Owner Approval", "Contractor Certified", and "Information Only".

**Document.Stage** - If allowable values are not specified by contract, the default values are: "As Built", "Submitted", "Approved", "Exact Requirement", "Maximum Requirement", "Minimum Requirement", and "Requirement".

**Attribute.Category** - If allowable values are not specified by contract, the default values are: "As Built", "Submitted", "Approved", "Exact Requirement", "Maximum Requirement", "Minimum Requirement", and "Requirement".

**Coordinate.Category** - If allowable values are not specified by contract, the default values are: "point", "line-end-one", "line-end-two", "box-lowerleft", and "box-upperright".

**Issue.Type** - If allowable values are not specified by contract, the default values are: "Change", "Claim", "Coordination", "Environmental", "Function", "IndoorAirQuality", "Installation", "RFI", "Safety", and "Specification".

**Issue.Risk** - If allowable values are not specified by contract, the default values are: "Very High", "High", "Moderate", "Low", and "Unknown".

**Issue.Chance** - If allowable values are not specified by contract, the default values are: "Has Occurred", "High", "Moderate", "Low", and "Unknown".

**Issue.Impact** - If allowable values are not specified by contract, the default values are: "Very High", "High", "Moderate", "Low", and "Unknown".

#### 4.2.6.2.3 Business Rule Definition

**Classification** - Verify presence of a classification system composed of a number part and a name part are present.

**Conversion Units** - Verify that common units for COBie required object properties have been identified.

**COBie Metadata** - Verify unique object naming according to COBie business rules.

**COBie Contact** - Verify identification of authorship or responsible party.

**Control Assignment** - Verify that the party responsible for resolution of specific issues is a valid COBie Contact.

**External Data Constraints** - Verify relationships between objects. Verify compound keys required based on COBie business rules. Verify form of any related Documents.

**Group Assignment** - Verify that group assignments are internally consistent.

**Object Typing** - Verify that product assets have a valid Type associated with them.

**Project Declaration** - Verify that values are present in required project information fields.

**Property Sets** - Verify the proper structure of any property sets associated with any object. Of particular interest is that all numeric properties have explicitly defined units.

**Quantities on Occurrences** - Verify that values are identified for properties have been provided either through specific object properties, or property sets.

**Sequential Connectivity** - Verify that identified tasks are appropriately linked to related entities and that individual operations, if specified, are provided sequentially.

**Spatial Composition** - Verify the relations between spatial objects exist according to COBie business rules. For Components, verify that all components are assigned to a specific Space.

**Spatial Containment** - Verify that each managed asset is associated with a Space.

#### 4.2.6.3 Exchange requirements reusability

##### 4.2.6.3.1 Related business process list

Business processes that deliver information about a wide variety of assets can be found in the NBIMS-US™ V3 BPie and LCie standards. Assets specific to individual systems, represented by other NBIMS-US™ V3 standards (HVACie, Sparkie, WSie, and BAMie) require the exchange of information about a limited subset of the entire list of building assets. COBie processes serve as the template for the exchange of facility asset information (Spaces and Equipment) in each of these standards.

The table below identifies the general categories of COBie by phase of project, using OmniClass Table 31, and compares those processes to the processes identified in related NBIMS-US™ V3 information exchange submissions.

**Table 46 Related Business Process List**

Phase of Project	Building Programming (BPie)	COBie	Mechanical System Design (HVACie)	Electrical System Design (Sparkie)	Plumbing System Design (WSie)
31-10 00 00 Inception Phase		X			
31-20 00 00 Conceptualization Phase	X	X			
31-30 00 00 Criteria Definition Phase	X	X		X	X

31-40 00 00 Design Phase		X	X	X	X
31-60 00 00 Implementation Phase		X			
31-80 00 00 Operations Phase		X			
31-90 00 00 Closure Phase		X			

#### 4.2.6.3.2 Related exchange requirements list

Given that COBie defines asset-related the processes defined generically (in BPie and LCie), or within the context of a specific design and construction discipline (HVACie, Sparkie, WSie, BAMie) as noted in the previous section, it is the case that COBie also defines the generic requirements for exchange of information about these same spatial and equipment assets. The following lists the COBie exchange requirements that may be relevant to other projects.

- Facility Criteria
- Discipline Specifications
- Project Definition
- Space Program
- Product Program
- Design Early
- Design Schematic
- Design Coordinated
- Design Issue
- Product Type Template
- Product Template
- Bid Issue
- Product Type Selection
- System Layout
- Product Installation
- Product Inspection
- Construction Issue
- Product Type Parts
- Product Type Warranty
- Product Type Maintenance
- System Operation
- Space Condition
- Product Parts Replacement
- Space Occupancy
- Space Activity Renovation
- Remodel
- Expand
- Demolish

#### 4.2.6.3.3 Related exchange requirement reuse analysis

A detailed reuse analysis was conducted between the COBie, a life-cycle specification and the following design-side projects and proposed standards:

- BPie
- BAMie
- HVACie
- SPARKie
- WSie

The analysis conducted recognized that the projects identified above version 1.0 specification that focused not on the life-cycle nature of the exchanges but focused on the definition of the design-side

exchanges. This focus was necessary in order to establish that information that should be provided by designers in each context. Future extensions to these projects can further complete the table below since updated during construction and operations would also share many of the same exchange requirements. The table below demonstrates potential re-use of information contained in COBie, with other buildingSMART alliance® information exchange projects.

**Table 47 Exchange Requirement Reuse Analysis**

Exchange	COBie	BPie	BAMie	HVACie	SPARKie	WSie
Facility Criteria	X	X			X	X
Discipline Specifications	X				X	X
Project Definition	X				X	X
Space Program	X	X		X	X	X
Product Program	X	X		X	X	X
Design Early	X				X	X
Design Schematic	X				X	X
Design Coordinated	X				X	X
Design Issue	X				X	X
ProductTypeTemplate	X		X	X	X	X
Product Template	X		X	X	X	X
Bid Issue	X					
ProductTypeSelection	X					
System Layout	X		X	X	X	X
Product Installation	X					
Product Inspection	X					
Construction Issue	X					
Product Type Parts	X					
ProductTypeWarranty	X					
ProductTypeMaintenance	X					
System Operation	X		X	X		
Space Condition	X					
ProductPartsReplacement	X					
Space Occupancy	X		X			
Space ActivityRenovation	X					
Remodel	X					
Expand	X					
Demolish	x					

#### 4.2.7 Model view definition documentation

*Author's Editorial Comment: The NBIMS-US™ V2 standard did not include the content of the associated MVD. NBIMS-US™ V2 referenced the buildingSMART international Facility Management Handover Model View Definition (MVD) web site as the source for the COBie MVD. This NBIMS-US™ V3 standard replaces the originally referenced MVD with the MVD documentation found in this section. The MVD documentation provided in this chapter reproduces information needed for the COBie MVD from the IFC*

reference standard. A web-based set of information using the buildingSMART international ifcDocs XML schema will be published to allow users the familiar IFC schema browsing capability. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange submissions, the information provided in this section should be considered a moderate change.

#### 4.2.7.1 Data definitions list

The list of IFC entities contained in this MVD is shown below:

- IfcSpace
- IfcBuildingStory
- IfcBuilding
- IfcSite
- IfcProject
- IfcZone
- IfcSystem
- IfcElement
- IfcElementType
- IfcTask
- IfcActor
- IfcActionRequest
- IfcProcess
- IfcGroup
- IfcPropertySet
- IfcAnnotation
- IfcRelConnectsPorts
- IfcWorkSchedule
- IfcRelAssociatesDocument
- IfcRelConnectsWithRealizingElements
- IfcConstructionProductResource
- IfcConstructionProductResourceType

#### 4.2.7.2 Data definitions

Each data definition is listed by name along with description and specifications in EXPRESS schema definition (ISO-10303-11) and XML Schema Definition (XSD). The XSD representation is directly consumable by software tools for editing XML data exchange, while the EXPRESS representation provides rich metadata suitable for translation into programming languages and database schemas.

Data definitions are adapted from Industry Foundation Classes, where any attributes that are unused are stripped or omitted; in EXPRESS, such stripped attributes use the type “IfcStrippedOptional”, preserving the order of attributes for file format compatibility; in XSD, such stripped attributes are simply omitted as XML formats use named tags without any order dependency.

##### 4.2.7.2.1 IfcSpace definition

A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.

A space is associated to a building story (or in case of exterior spaces to a site). A space may span over several connected spaces. Therefore a space group provides for a collection of spaces included in a story. A space can also be decomposed in parts, where each part defines a partial space. This is defined by the CompositionType attribute of the supertype IfcSpatialStructureElement which is interpreted as follow:

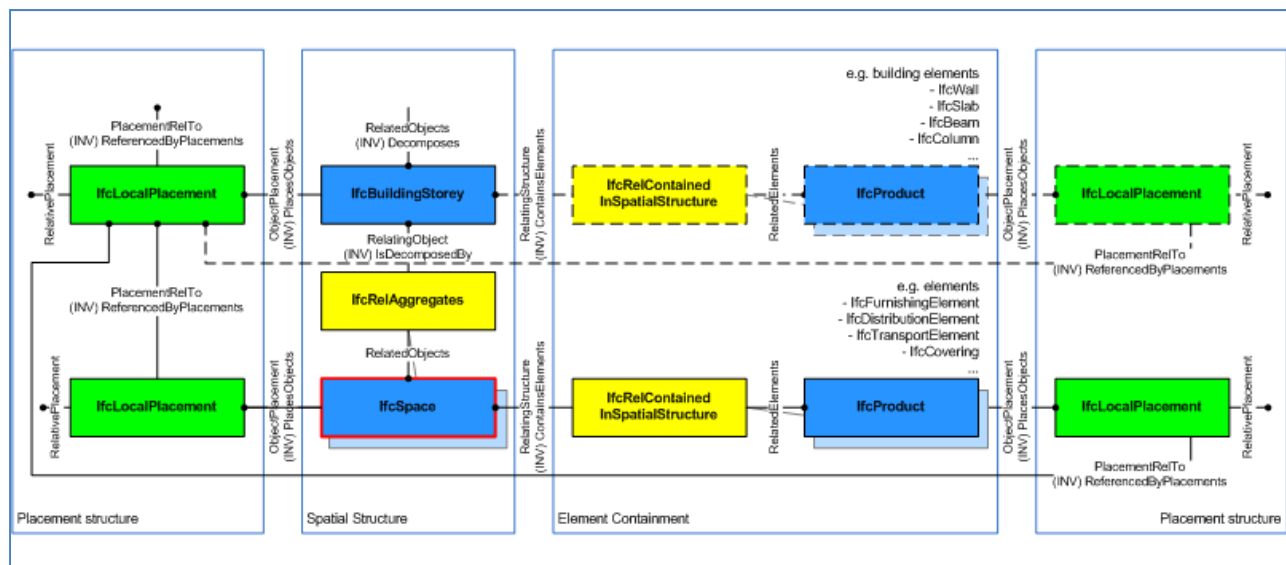
- COMPLEX = space group
- ELEMENT = space

- PARTIAL = partial space

NOTE: View definitions and implementation agreements may restrict spaces with CompositionType=ELEMENT to be non-overlapping.

The IfcSpace is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship IfcRelAggregates. The figure below shows the IfcSpace as part of the spatial structure. It also serves as the spatial container for space related elements.

NOTE: Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

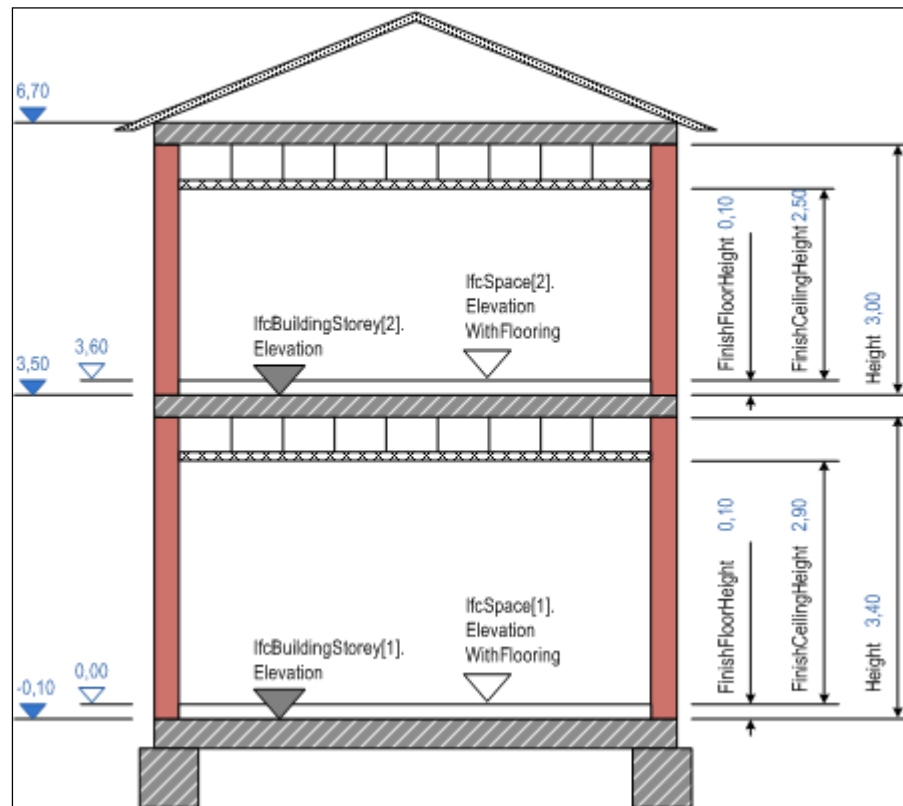


**Figure 31 ifcSpace - Spatial Structure**

The following guidelines should apply for using the Name, Description, LongName and ObjectType attributes.

- Name holds the unique name (or space number) from the plan.
- Description holds any additional information field the user may have specified, there are no further recommendations.
- LongName holds the full name of the space, it is often used in addition to the Name, if a number is assigned to the room, then the descriptive name is exchanged as LongName.
- ObjectType holds the space type, i.e. usually the functional category of the space .

NOTE: In cases of inconsistency between the geometric representation of the IfcSpace and the combined geometric representations of the surrounding IfcRelSpaceBoundary, the geometric representation of the space should take priority over the geometric representation of the surrounding space boundaries. The figure below describes the heights and elevations of the IfcSpace.



**Figure 32 ifcSpace - Heights**

HISTORY New entity in IFC1.0

#### Attribute Use Definition

- elevation of the space (top of construction slab) equals elevation of story: provided by IfcBuildingStory.Elevation relative to IfcBuilding.ElevationOfRefHeight
- elevation of the space flooring (top of flooring on top of slab): provided by IfcSpace.ElevationWithFlooring relative to IfcBuilding.ElevationOfRefHeight
- height of space (top of slab below to bottom of slab above): provided by BaseQuantity with Name="Height"
- floor height of space (top of slab below to top of flooring): provided by BaseQuantity with Name="FinishFloorHeight"
- net height of space (top of flooring to bottom of suspended ceiling): provided by BaseQuantity with Name="FinishCeilingHeight"

EXPRESS Specification:

**ENTITY** IfcBuildingStorey

**SUBTYPE OF** (IfcSpatialStructureElement);



Elevation : **OPTIONAL** IfcStrippedOptional;  
**END\_ENTITY;**

Attribute Definitions:

**Elevation** : Elevation of the base of this storey, relative to the 0,00 internal reference height of the building. The 0.00 level is given by the absolute above sea level height by the ElevationOfRefHeight attribute given at IfcBuilding.

Inheritance Graph:

**ENTITY** IfcBuildingStorey

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;  
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
 Name : **OPTIONAL** IfcLabel;  
 Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;  
 Representation : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

**ENTITY** IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

**ENTITY** IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

**ENTITY** IfcBuildingStorey

Elevation : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

```
<xs:element name="IfcBuildingStorey" type="ifc:IfcBuildingStorey" substitutionGroup="ifc:IfcSpatialStructureElement"
nillable="true"/>
<xs:complexType name="IfcBuildingStorey">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement"/>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.2 IfcBuildingStory Definition

The building story has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound.

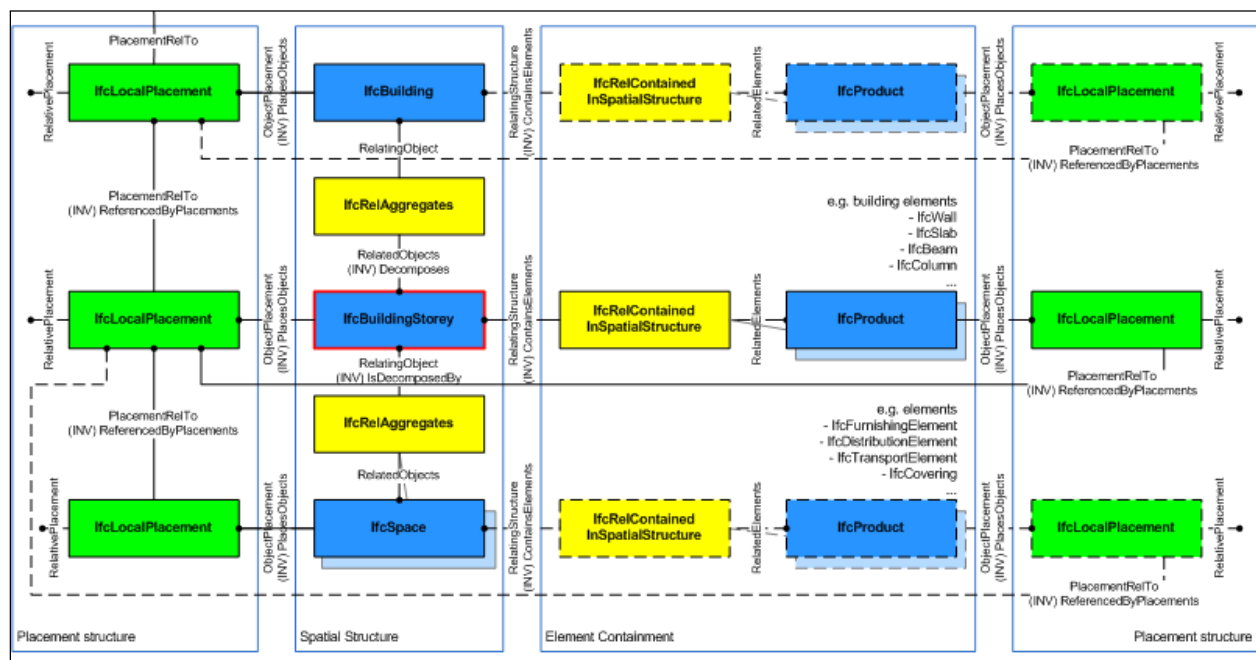
A story is (if specified) associated to a building. A story may span over several connected stories. Therefore story complex provides for a collection of stories included in a building. A story can also be decomposed in (horizontal) parts, where each part defines a partial story. This is defined by the composition type attribute of the supertype IfcSpatialStructureElements which is interpreted as follow:

- **COMPLEX:** building story complex
- **ELEMENT:** building story
- **PARTIAL:** partial building story

EXAMPLE In split level houses, a story is split into two or more partial stories, each with a different elevation. It can be handled by defining a story, which includes two or more partial stories with the individual elevations.

The IfcBuildingStory is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship IfcRelAggregates. The figure below shows the IfcBuildingStory as part of the spatial structure. It also serves as the spatial container for building and other elements.

NOTE: Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.



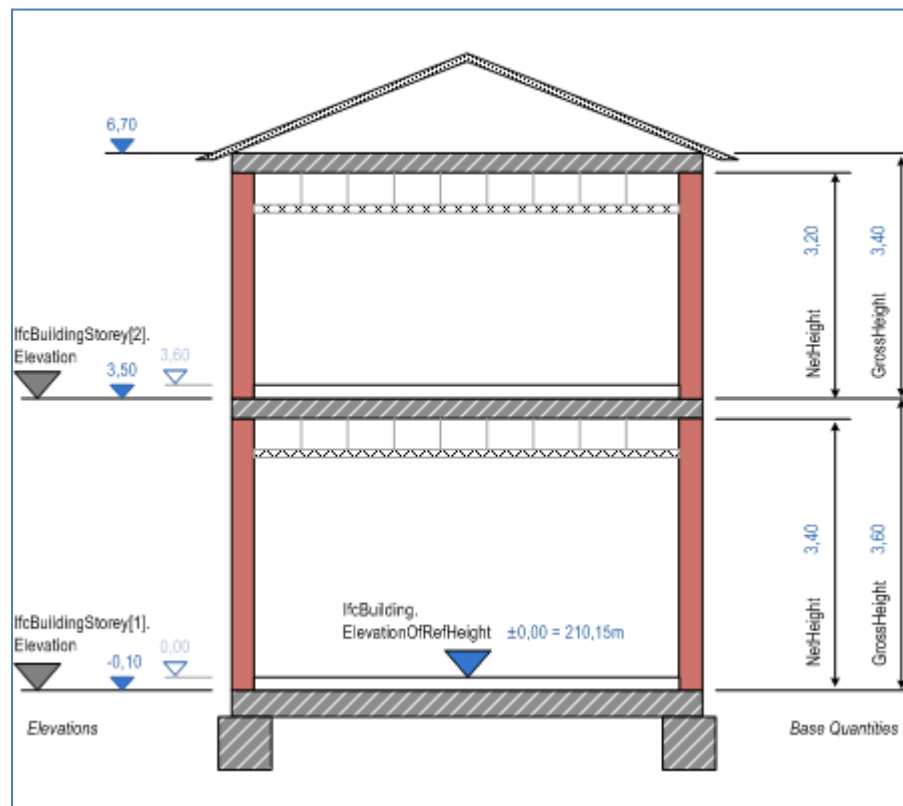
**Figure 33 ifcBuildingStory - Spatial Structure**

HISTORY New entity in IFC1.0

### Attribute Use Definition

The figure below describes the heights and elevations of the IfcBuildingStory.

- elevation of story provided by: IfcBuildingStory.Elevation as a local height value relative to IfcBuilding.ElevationOfRefHeight, it is usually the top of construction slab
- net height of story, also referred to as total height or system height (top of construction slab to top of construction slab above): provided by BaseQuantity with Name="GrossHeight"
- net height of story (top of construction slab to bottom of construction slab above): provided by BaseQuantity with Name="NetHeight"

**Figure 34 ifcBuildingStory - Heights**

EXPRESS Specification:

**ENTITY** IfcBuildingStorey

**SUBTYPE OF** (IfcSpatialStructureElement);

Elevation : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

## Attribute Definitions:

**Elevation** : Elevation of the base of this storey, relative to the 0,00 internal reference height of the building. The 0.00 level is given by the absolute above sea level height by the ElevationOfRefHeight attribute given at IfcBuilding.

## Inheritance Graph:

**ENTITY** IfcBuildingStorey**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;  
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
 Name : **OPTIONAL** IfcLabel;  
 Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;  
 Representation : **OPTIONAL** IfcStrippedOptional;

**INVERSE****ENTITY** IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

**ENTITY** IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

**ENTITY** IfcBuildingStorey

Elevation : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

```
<xs:element name="IfcBuildingStorey" type="ifc:IfcBuildingStorey" substitutionGroup="ifc:IfcSpatialStructureElement"
nillable="true"/>
<xs:complexType name="IfcBuildingStorey">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement"/>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.3 IfcBuilding definition

A building represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, story, and space).

NOTE Definition from ISO 6707-1:

Construction work that has the provision of shelter for its occupants or contents as one of its main purpose and is normally designed to stand permanently in one place.

A building is (if specified) associated to a site. A building may span over several connected or disconnected buildings. Therefore building complex provides for a collection of buildings included in a site. A building can also be decomposed in (vertical) parts, where each part defines a building section. This is defined by the composition type attribute of the supertype IfcSpatialStructureElements which is interpreted as follow:

- **COMPLEX:** building complex
- **ELEMENT:** building
- **PARTIAL:** building section

The IfcBuilding is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship IfcRelAggregates. The figure below shows the IfcBuilding as part of the spatial structure. It also serves as the spatial container for building and other elements.

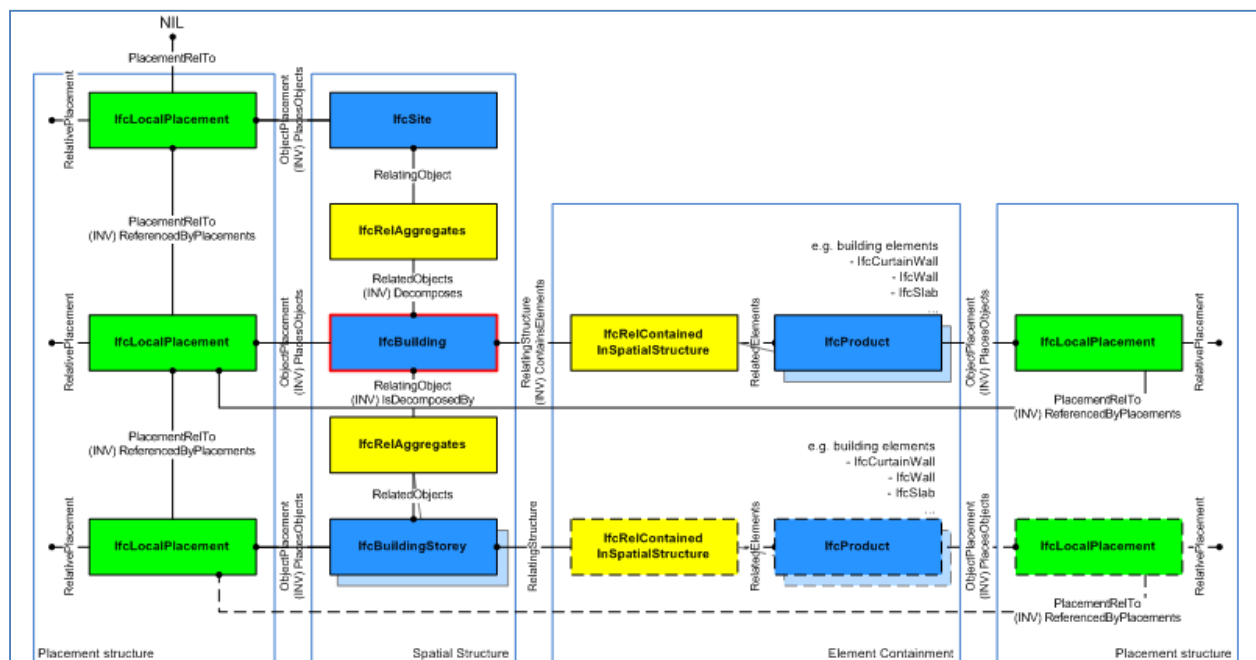


Figure 35 ifcBuilding - Spatial Structure

HISTORY New entity in IFC1.0.

NOTE: detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

Systems, such as building service or electrical distribution systems, zonal systems, or structural analysis systems, relate to IfcBuilding by using the objectified relationship IfcRelServicesBuildings.

### Attribute Use Definition

The figure below describes the heights and elevations of the IfcBuilding. It is used to provide the height above sea level of the project height datum for this building, that is, the internal height 0.00. The height 0.00 is often used as a building internal reference height and equal to the floor finish level of the ground floor.

- base elevation of building provided by: IfcBuilding.ElevationOfRefHeight, it is usually the top of construction slab
- base elevation of terrain at the perimeter of the building provided by: IfcBuilding.ElevationOfTerrain, it is usually the minimum elevation is sloped terrain
- total height of building, also referred to as ridge height (top of roof structure, e.g the ridge against terrain): provided by BaseQuantity with Name="TotalHeight"
- eaves height of building (base of roof structure, e.g the eaves against terrain): provided by BaseQuantity with Name="EavesHeight"

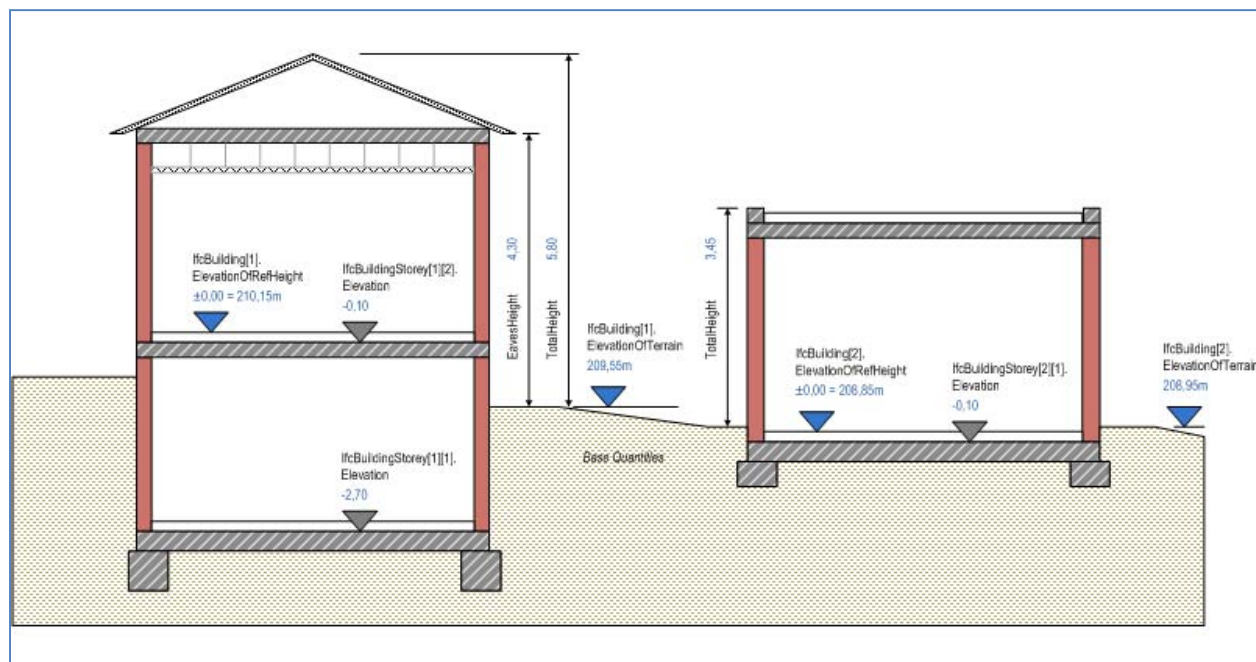


Figure 36 IfcBuilding - Heights

## EXPRESS Specification:

**ENTITY IfcBuilding****SUBTYPE OF** (IfcSpatialStructureElement);ElevationOfRefHeight : **OPTIONAL** IfcStrippedOptional;ElevationOfTerrain : **OPTIONAL** IfcStrippedOptional;BuildingAddress : **OPTIONAL** IfcStrippedOptional;**END\_ENTITY;**

## Attribute Definitions:

**ElevationOfRefHeight** : Elevation above sea level of the reference height used for all storey elevation measures, equals to height 0.0. It is usually the ground floor level.**ElevationOfTerrain** : Elevation above the minimal terrain level around the foot print of the building, given in elevation above sea level.**BuildingAddress** : Address given to the building for postal purposes.

## Inheritance Graph:

**ENTITY IfcBuilding****ENTITY IfcRoot**

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;Name : **OPTIONAL** IfcLabel;Description : **OPTIONAL** IfcText;**ENTITY IfcObjectDefinition****INVERSE**HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;**ENTITY IfcObject**ObjectType : **OPTIONAL** IfcStrippedOptional;**INVERSE**IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;**ENTITY IfcProduct**ObjectPlacement : **OPTIONAL** IfcStrippedOptional;Representation : **OPTIONAL** IfcStrippedOptional;**INVERSE****ENTITY IfcSpatialElement**LongName : **OPTIONAL** IfcStrippedOptional;**INVERSE**ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;**ENTITY IfcSpatialStructureElement**CompositionType : **OPTIONAL** IfcStrippedOptional;**ENTITY IfcBuilding**ElevationOfRefHeight : **OPTIONAL** IfcStrippedOptional;ElevationOfTerrain : **OPTIONAL** IfcStrippedOptional;BuildingAddress : **OPTIONAL** IfcStrippedOptional;**END\_ENTITY;**

```

<xs:element name="IfcBuilding" type="ifc:IfcBuilding" substitutionGroup="ifc:IfcSpatialStructureElement"
nillable="true"/>
<xs:complexType name="IfcBuilding">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement"/>
  </xs:complexContent>
</xs:complexType>

```

#### 4.2.7.2.4 IfcSite Definition

A site is a defined area of land, possibly covered with water, on which the project construction is to be completed. A site may be used to erect, retrofit or turn down building(s), or for other construction related developments.

NOTE Term according to ISO6707-1 vocabulary "area of land or water where construction work or other development is undertaken".

A site may include a definition of the single geographic reference point for this site (global position using WGS84 with Longitude, Latitude and Elevation). The precision is provided up to millionth of a second and it provides an absolute placement in relation to the real world as used in exchange with geospatial information systems. If asserted, the Longitude, Latitude and Elevation establish the point in WGS84 where the point 0.,0.,0. of the LocalPlacement of IfcSite is situated.

The geometrical placement of the site, defined by the IfcLocalPlacement, shall be always relative to the spatial structure element, in which this site is included, or absolute, i.e. to the world coordinate system, as established by the geometric representation context of the project. The world coordinate system, established at the IfcProject.RepresentationContexts, may include a definition of the true north within the XY plane of the world coordinate system, if provided, it can be obtained at IfcGeometricRepresentationContext.TrueNorth.

A project may span over several connected or disconnected sites. Therefore site complex provides for a collection of sites included in a project. A site can also be decomposed in parts, where each part defines a site section. This is defined by the composition type attribute of the supertype IfcSpatialStructureElements which is interpreted as follow:

- COMPLEX = site complex
- ELEMENT = site
- PARTIAL = site section

The IfcSite is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The figure below shows the IfcSite as part of the spatial structure. In addition to the logical spatial structure, also the placement hierarchy is shown. In this example the spatial structure hierarchy and the placement hierarchy are identical.

NOTE: Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

HISTORY New entity in IFC1.0.



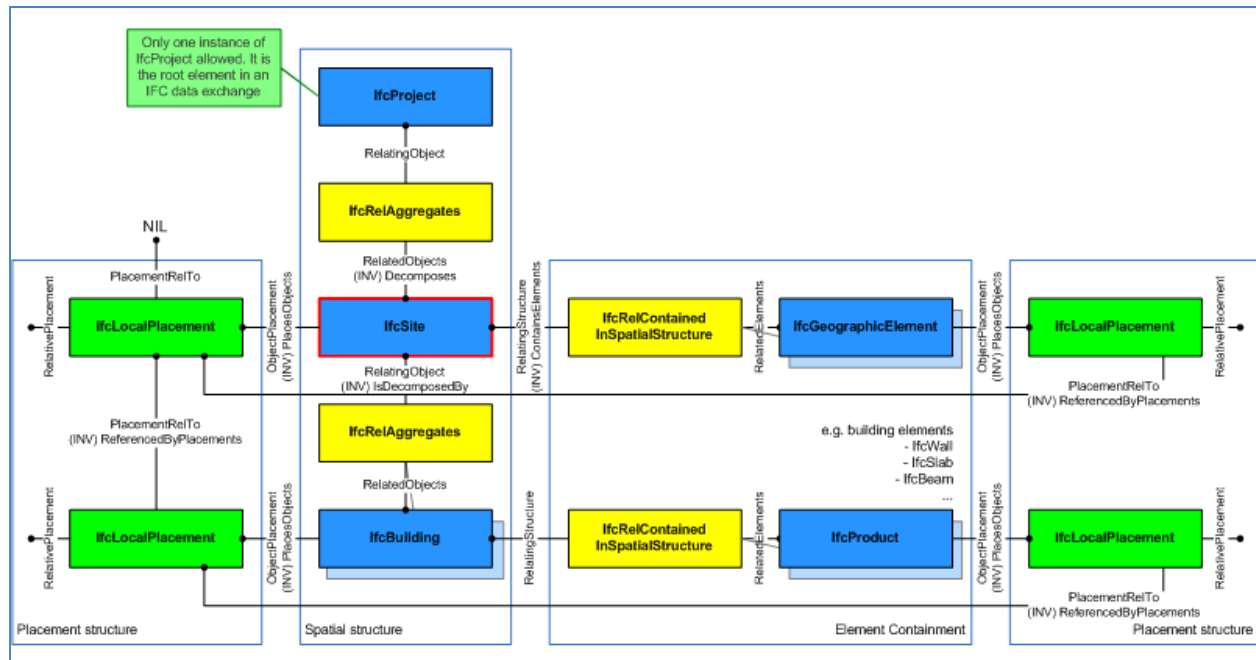


Figure 37 ifcSite - Spatial Structure

### Attribute Use Definition

The figure below describes the heights and elevations of the IfcSite. It is used to provide the geographic longitude, latitude, and height above sea level for the origin of the site. The origin of the site is the local placement.

The provision of longitude, latitude, height at the IfcSite for georeferencing is provided for upward compatibility reasons. It requires a single instance of IfcSite and WGS84 as coordinate reference system.

For exact georeferencing (or referencing to any other geographic coordinate system other than WSG84) the entities IfcCoordinateReferenceSystem and IfcMapConversion have to be used to define an exact mapping of the project engineering coordinate system to the geographic (or map) coordinate system.

- reference height of site is provided by: IfcSite.RefElevation, it is given according to the height datum used at this location.
- the reference height of each building situated at the site is given against the same height datum used at this location.
- the elevations of each story belonging to each building are given as local height relative to the reference height of the building.

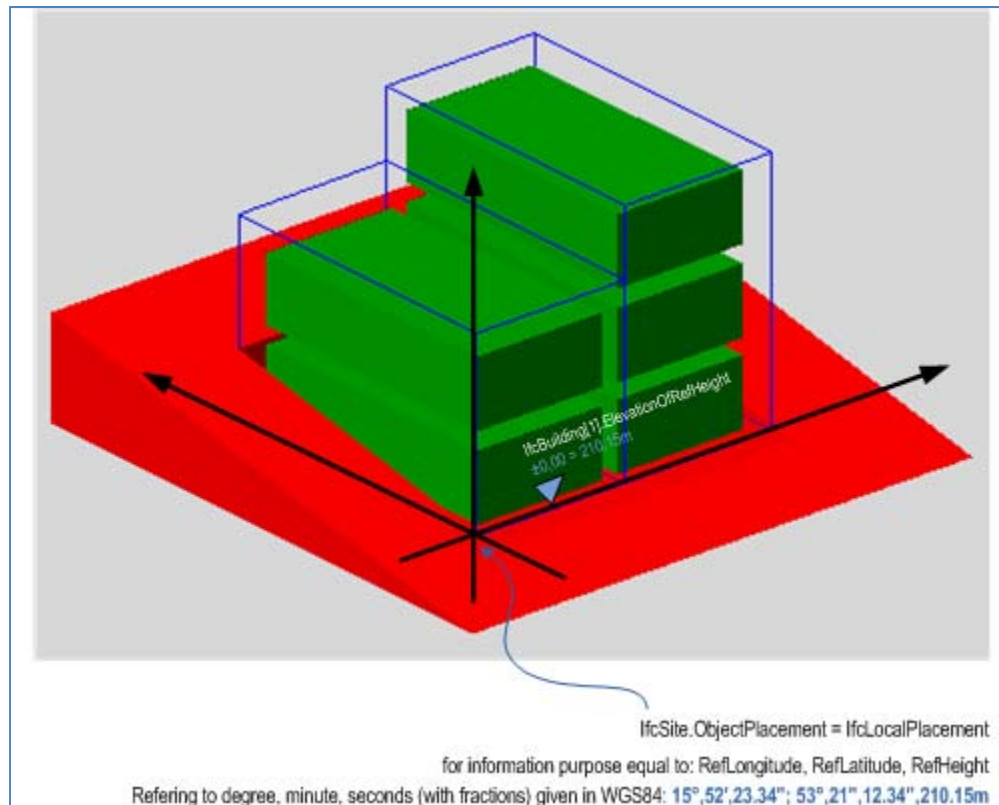


Figure 38 ifcBuilding - Heights

EXPRESS Specification:

**ENTITY** IfcSite

**SUBTYPE OF** (IfcSpatialStructureElement);

RefLatitude : **OPTIONAL** IfcStrippedOptional;  
 RefLongitude : **OPTIONAL** IfcStrippedOptional;  
 RefElevation : **OPTIONAL** IfcStrippedOptional;  
 LandTitleNumber : **OPTIONAL** IfcStrippedOptional;  
 SiteAddress : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

Attribute Definitions:

**RefElevation** : Datum elevation relative to sea level.  
**LandTitleNumber** : The land title number (designation of the site within a regional system).  
**SiteAddress** : Address given to the site for postal purposes.

Inheritance Graph:

**ENTITY** IfcSite

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;  
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
 Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;

Representation : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

**ENTITY** IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

**ENTITY** IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

**ENTITY** IfcSite

RefLatitude : **OPTIONAL** IfcStrippedOptional;

RefLongitude : **OPTIONAL** IfcStrippedOptional;

RefElevation : **OPTIONAL** IfcStrippedOptional;

LandTitleNumber : **OPTIONAL** IfcStrippedOptional;

SiteAddress : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

```
<xs:element name="IfcSite" type="ifc:IfcSite" substitutionGroup="ifc:IfcSpatialStructureElement" nillable="true"/>
<xs:complexType name="IfcSite">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement"/>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.5 IfcProject Definition

IfcProject indicates the undertaking of some design, engineering, construction, or maintenance activities leading towards a product. The project establishes the context for information to be exchanged or shared, and it may represent a construction project but does not have to. The IfcProject's main purpose in an exchange structure is to provide the root instance and the context for all other information items included.

The context provided by the IfcProject includes:

- the default units
- the geometric representation context for exchange structures including shape representations
- the world coordinate system
- the coordinate space dimension

- the precision used within the geometric representations, and
- optionally the indication of the true north relative to the world coordinate system

HISTORY New entity in IFC1.0

IFC4 CHANGE The attributes *RepresentationContexts* and *UnitsInContext* are made optional and are promoted to supertype *IfcContext*.

Informal Propositions:

There shall only be one project within the exchange context. This is enforced by the global rule *IfcSingleProjectInstance*.

EXPRESS Specification:

**ENTITY** *IfcProject*

**SUBTYPE OF** (*IfcContext*);

**WHERE**

*HasName* : EXISTS(*SELF*\*IfcRoot.Name*);

*CorrectContext* : NOT(EXISTS(*SELF*\*IfcContext.RepresentationContexts*)) OR (SIZEOF(QUERY(Temp <\*  
*SELF*\*IfcContext.RepresentationContexts* |  
'IFCREPRESENTATIONRESOURCE.IFCGEOMETRICREPRESENTATIONSUBCONTEXT' IN  
TYPEOF(Temp) )) = 0);

*NoDecomposition* : SIZEOF(*SELF*\*IfcObjectDefinition.Decomposes*) = 0;

*HasOwnerHistory* : EXISTS(*SELF*\*IfcRoot.OwnerHistory*);

**END\_ENTITY;**

Formal Propositions:

**HasName** : The *Name* attribute has to be provided for *IfcProject*. It is the short name for the project.

**CorrectContext** : If a *RepresentationContexts* relation is provided then there shall be no instance of *IfcGeometricRepresentationSubContext* directly included in the set of *RepresentationContexts*.

**NoDecomposition** : The *IfcProject* represents the root of the any decomposition tree. It shall therefore not be used to decompose any other object definition.

**HasOwnerHistory** : The *OwnerHistory* attribute has to be provided for *IfcProject*. It provides the minimum of owner information for the project data set and the last change action, that applied to the whole data set.

NOTE Each individual data item, that derives from *IfcRoot* may have an individual *OwnerHistory*. It then overrides the common ownership and change action information provided at the single *IfcProject* instance in an IFC data set.

IFC4 CHANGE New where rule.

Inheritance Graph:

**ENTITY** *IfcProject*

**ENTITY** *IfcRoot*

*GlobalId* : *IfcGloballyUniqueId*;

*OwnerHistory* : **OPTIONAL** *IfcOwnerHistory*;

*Name* : **OPTIONAL** *IfcLabel*;

Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcContext

ObjectType : **OPTIONAL** IfcStrippedOptional;

LongName : **OPTIONAL** IfcStrippedOptional;

Phase : **OPTIONAL** IfcStrippedOptional;

RepresentationContexts: **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

UnitsInContext : **OPTIONAL** IfcUnitAssignment;

**INVERSE**

IsDefinedBy : **SET** [0:?] **OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

Declares : **SET OF** IfcRelDeclares **FOR** RelatingContext;

**ENTITY** IfcProject

**END\_ENTITY;**

```
<xs:element name="IfcProject" type="ifc:IfcProject" substitutionGroup="ifc:IfcContext" nillable="true"/>
<xs:complexType name="IfcProject">
  <xs:complexContent>
    <xs:extension base="ifc:IfcContext"/>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.6 IfcZone Definition

A zone is a group of spaces, partial spaces or other zones. Zone structures may not be hierarchical (in contrary to the spatial structure of a project - see IfcSpatialStructureElement), i.e. one individual IfcSpace may be associated with zero, one, or several IfcZone's. IfcSpace's are grouped into an IfcZone by using the objectified relationship IfcRelAssignsToGroup as specified at the supertype IfcGroup.

NOTE Certain use cases may restrict the freedom of non hierarchical relationships. In some building service use cases the zone denotes a view based delimited volume for the purpose of analysis and calculation. This type of zone cannot overlap with respect to that analysis, but may overlap otherwise.

HISTORY New entity in IFC1.0

IFC4 CHANGE the entity is now subtyped from IfcSystem (not its supertype IfcGroup) with upward compatibility for file based exchange.

EXPRESS Specification:

**ENTITY** IfcZone

**SUBTYPE OF** (IfcSystem);

LongName : **OPTIONAL** IfcStrippedOptional;

**WHERE**

WR1 : (SIZEOF(SELF\IfcGroup.IsGroupedBy) = 0) OR (SIZEOF (QUERY (temp <\*  
SELF\IfcGroup.IsGroupedBy[1].RelatedObjects | NOT(('IFCPRODUCTEXTENSION.IFCZONE' IN  
TYPEOF(temp)) OR ('IFCPRODUCTEXTENSION.IFCSPACE' IN TYPEOF(temp)) OR  
('IFCPRODUCTEXTENSION.IFCSPATIALZONE' IN TYPEOF(temp)) ))) = 0);

**END\_ENTITY;**

#### Attribute Definitions:

**LongName** : Long name for a zone, used for informal purposes. It should be used, if available, in conjunction with the inherited *Name* attribute.  
NOTE In many scenarios the *Name* attribute refers to the short name or number of a zone, and the *LongName* refers to the full name.  
IFC4 CHANGE The attribute has been added at the end of the entity definition.

#### Formal Propositions:

**WR1** : An IfcZone is grouped by the objectified relationship IfcRelAssignsToGroup. Only objects of type IfcSpace, IfcZone and *IfcSpatialZone* are allowed as *RelatedObjects*.

#### Inheritance Graph:

**ENTITY** IfcZone

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;  
OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
Name : **OPTIONAL** IfcLabel;  
Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcGroup

**INVERSE**

IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

**ENTITY** IfcSystem

**INVERSE**

**ENTITY** IfcZone

LongName : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

```
<xs:element name="IfcZone" type="ifc:IfcZone" substitutionGroup="ifc:IfcSystem" nillable="true"/>
<xs:complexType name="IfcZone">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSystem"/>
  </xs:complexContent>
</xs:complexType>
```

```
</xs:complexContent>
```

```
</xs:complexType>
```

#### 4.2.7.2.7 IfcSystem Definition

A system is an organized combination of related parts within an AEC product, composed for a common purpose or function or to provide a service. A system is essentially a functionally related aggregation of products. The grouping relationship to one or several instances of IfcProduct (the system members) is handled by IfcRelAssignsToGroup.

**NOTE** The use of IfcSystem often applies to the representation of building services related systems, such as the piping system, cold water system, etc. Members within such a system may or may not be connected using the connectivity related entities (through IfcDistributionPort).

**HISTORY** New entity in IFC1.0

**ENTITY** IfcSystem

**SUPERTYPE OF**(IfcZone)

**SUBTYPE OF** (IfcGroup);

**INVERSE**

**END\_ENTITY;**

Attribute Definitions:

Inheritance Graph:

**ENTITY** IfcSystem

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcGroup

**INVERSE**

IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

**ENTITY** IfcSystem

**INVERSE**

**END\_ENTITY;**

```

<xs:element name="IfcSystem" type="ifc:IfcSystem" substitutionGroup="ifc:IfcGroup" nillable="true"/>
<xs:complexType name="IfcSystem">
  <xs:complexContent>
    <xs:extension base="ifc:IfcGroup"/>
  </xs:complexContent>
</xs:complexType>

```

#### 4.2.7.2.8 IfcElement definition

An element is a generalization of all components that make up an AEC product. Those elements can be logically contained by a spatial structure element that constitutes a certain level within a project structure hierarchy (site, building, story or space). This is done by using the `IfcRelContainedInSpatialStructure` relationship.

Elements are physically existent objects, although they might be void elements, such as holes. Elements either remain permanently in the AEC product, or only temporarily, as formwork does. Elements can be either assembled on site or pre-manufactured and built in on site.

**EXAMPLE** Examples of elements in a building construction context are walls, floors, windows and recesses.

An element can have material and quantity information assigned through the `IfcRelAssociatesMaterial` and `IfcRelDefinesByProperties` relationship.

In addition an element can be declared to be a specific occurrence of an element type (and thereby be defined by the element type properties) using the `IfcRelDefinesByType` relationship.

An element can also be defined as an element assembly that is a group of semantically and topologically related elements that form a higher level part of the AEC product. Those element assemblies are defined by virtue of the `IfcRelAggregates` relationship.

**EXAMPLE** Examples for element assembly are complete Roof Structures, made by several Roof Areas, or a Stair, composed by Flights and Landings.

Elements that performs the same function may be grouped by an "Element Group By Function". It is realized by an instance of `IfcGroup` with the `ObjectType = 'ElementGroupByFunction'`.

**HISTORY** New entity in IFC1.0

#### Quantity Use Definition

The quantities relating to the `IfcElement` are defined by the `IfcElementQuantity` and attached by the `IfcRelDefinesByProperties`. A detailed specification for individual quantities is introduced at the level of subtypes of `IfcElement`.

#### Geometry Use Definitions

The geometric representation of any `IfcElement` is given by the `IfcProductDefinitionShape` and `IfcLocalPlacement` allowing multiple geometric representations. A detailed specification for the local placement and shape representation is introduced at the level of subtypes of `IfcElement`.



## EXPRESS Specification:

**ENTITY** IfcElement**SUBTYPE OF** (IfcProduct);Tag : **OPTIONAL** IfcStrippedOptional;**INVERSE**ConnectedTo : **SET OF** IfcRelConnectsElements **FOR** RelatingElement;IsConnectionRealization: **SET OF** IfcRelConnectsWithRealizingElements **FOR** RealizingElements;ConnectedFrom : **SET OF** IfcRelConnectsElements **FOR** RelatedElement;ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;**END\_ENTITY;**

## Attribute Definitions:

- Tag** : The tag (or label) identifier at the particular instance of a product, e.g. the serial number, or the position number. It is the identifier at the occurrence level.
- ConnectedTo** : Reference to the element connection relationship. The relationship then refers to the other element to which this element is connected to.
- IsConnectionRealization** : Reference to the connection relationship with realizing element. The relationship, if provided, assigns this element as the realizing element to the connection, which provides the physical manifestation of the connection relationship.
- ConnectedFrom** : Reference to the element connection relationship. The relationship then refers to the other element that is connected to this element.
- ContainedInStructure** : Containment relationship to the spatial structure element, to which the element is primarily associated. This containment relationship has to be hierarchical, i.e. an element may only be assigned directly to zero or one spatial structure.

## Inheritance Graph:

**ENTITY** IfcElement**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;Name : **OPTIONAL** IfcLabel;Description : **OPTIONAL** IfcText;**ENTITY** IfcObjectDefinition**INVERSE**HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;**ENTITY** IfcObjectObjectType : **OPTIONAL** IfcStrippedOptional;**INVERSE**IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;**ENTITY** IfcProductObjectPlacement : **OPTIONAL** IfcStrippedOptional;Representation : **OPTIONAL** IfcStrippedOptional;**INVERSE****ENTITY** IfcElement

```

    Tag : OPTIONAL IfcStrippedOptional;
INVERSE
    ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
    IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
    ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
    ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
END_ENTITY;
<xs:element name="IfcElement" type="ifc:IfcElement" abstract="true" substitutionGroup="ifc:IfcProduct"
nillable="true"/>
<xs:complexType name="IfcElement" abstract="true">
  <xs:complexContent>
    <xs:extension base="ifc:IfcProduct"/>
  </xs:complexContent>
</xs:complexType>

```

#### 4.2.7.2.9 IfcElementType definition

IfcElementType defines a list of commonly shared property set definitions of an element and an optional set of product representations. It is used to define an element specification (i.e. the specific product information, that is common to all occurrences of that product type).

NOTE The product representations are defined as representation maps (at the level of the supertype IfcTypeProduct, which gets assigned by an element instance through the IfcShapeRepresentation.Item[1] being an IfcMappedItem.

An element type is used to define the common properties of a certain type or style of an element that may be applied to instances of that element type to assign a specific style. Element types (the instantiable subtypes) may be exchanged without being already assigned to occurrences.

HISTORY New entity in IFC2x2

EXPRESS Specification:

```

ENTITY IfcElementType
  SUBTYPE OF (IfcTypeProduct);
  ElementType : OPTIONAL IfcStrippedOptional;
END_ENTITY;

```

Attribute Definitions:

**ElementType** : The type denotes a particular type that indicates the object further. The use has to be established at the level of instantiable subtypes. In particular it holds the user defined type, if the enumeration of the attribute 'PredefinedType' is set to USERDEFINED.

Inheritance Graph:

```

ENTITY IfcElementType
  ENTITY IfcRoot
    GlobalId : IfcGloballyUniqueId;
    OwnerHistory : OPTIONAL IfcOwnerHistory;
    Name : OPTIONAL IfcLabel;

```

Description : **OPTIONAL** IfcText;  
**ENTITY** IfcObjectDefinition  
**INVERSE**  
 HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;  
**ENTITY** IfcTypeObject  
 ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;  
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;  
**INVERSE**  
 Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;  
**ENTITY** IfcTypeProduct  
 RepresentationMaps : **OPTIONAL LIST** [1:?] OF **UNIQUE** IfcStrippedOptional;  
 Tag : **OPTIONAL** IfcStrippedOptional;  
**INVERSE**  
**ENTITY** IfcElementType  
 ElementType : **OPTIONAL** IfcStrippedOptional;  
**END\_ENTITY;**

```
<xs:element name="IfcElementType" type="ifc:IfcElementType" abstract="true" substitutionGroup="ifc:IfcTypeProduct"
nillable="true"/>
<xs:complexType name="IfcElementType" abstract="true">
  <xs:complexContent>
    <xs:extension base="ifc:IfcTypeProduct"/>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.10 IfcTask definition

An IfcTask is an identifiable unit of work to be carried out in a construction project.

A task is typically used to describe an activity for the construction or installation of products, but is not limited to these types. For example it might be used to describe design processes, move operations and other design, construction and operation related activities as well.

Quantities of resources consumed by the task are dealt with by defining the IfcElementQuantity for the resource and not at the instance of IfcTask.

**HISTORY** New entity in IFC1.0. Renamed from IfcWorkTask in IFC2x.

**IFC4 CHANGE** Attributes TaskTime and PredefinedType added. IfcMove and IfcOrderRequest has been removed in IFC4 and are now represented by IfcTask. IfcRelAssignsTasks relationship has been removed as well.

#### Attribute use definition

Each occurrence of IfcTask is given a name that is indicative of its content (IfcRoot.Name). A textual description of the task may be provided and this may be further elaborated by a narrative long description

(IfcProcess.LongDescription). A work method may be declared for the method of work used in carrying out a task. A task is identified as being either a milestone task or not. A milestone task is defined by the marker IsMilestone. and has no duration. A status and priority for each task may also be set.

### Time and duration use definition

Compared to previous IFC releases, basic task time information (scheduled start time, scheduled finish time, duration) is now directly attached to IfcTask through the TaskTime attribute. Regular tasks are defined through IfcTaskTime. Recurring tasks are defined through IfcTaskTimeRecurring. In case a regular task is derived from a recurring task both tasks should be linked together through a IfcRelNests relationship, where IfcRelNests.IsNestedBy points to the recurring task and IfcRelNests.Nests points to all regular tasks that have been derived from the recurring task.

### Representation of other activities

The use definitions for IfcTask have been generalised to represent other activities as well, including activities that had been defined by own entities in previous IFC releases. This includes

- Order actions
- Move operations

IfcTask represents an order that might be carried out by a Helpdesk acting the role of interface for the organization between the facility user and the functional requirement of fulfilling their needs. The actual task represented by the IfcTask entity is turning a request into an order and initiating the action that will enable the order to be completed. The IfcProjectOrder or one of its subtypes including maintenance work order, is related to the IfcTask using IfcRelAssignsToControl.

IfcTask can also be used to describe an activity that moves people, groups within an organization or complete organizations together with their associated furniture and equipment from one place to another. It thus replaces the previous IFC entity IfcMove. The functionality is represented in IfcTask as follows:

**Move from:** The place from which actors and their associated equipment are moving. Use IfcRelAssignsToProcess where RelatingProcess points to the task and RelatedObjects holds the location(s) from which to move.

**Move to:** The place to which actors and their associated equipment are moving. Use IfcRelAssignsToProduct where RelatedObjects points to the task(s) and RelatingProduct points to the location to which to move.

**Punch list:** A list of points concerning a move that require attention. Use LongDescription or else identify sub-tasks to track punch list items individually via IfcRelNests.

EXPRESS Specification:

#### ENTITY IfcTask

**SUBTYPE OF** (IfcProcess);

Status	: <b>OPTIONAL</b> IfcStrippedOptional;
WorkMethod	: <b>OPTIONAL</b> IfcStrippedOptional;
IsMilestone	: IfcStrippedOptional;
Priority	: <b>OPTIONAL</b> IfcStrippedOptional;
TaskTime	: <b>OPTIONAL</b> IfcTaskTime;

PredefinedType : **OPTIONAL** IfcStrippedOptional;

**WHERE**

HasName : EXISTS(SELF\IfcRoot.Name);

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcTaskTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTaskTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcObject.ObjectType));

**END\_ENTITY;**

Attribute Definitions:

**Status** : Current status of the task.  
NOTE Particular values for status are not specified, these should be determined and agreed by local usage. Examples of possible status values include 'Not Yet Started', 'Started', 'Completed'.

**WorkMethod** : The method of work used in carrying out a task.  
NOTE This attribute should not be used if the work method is specified for the *IfcTaskType*

**IsMilestone** : Identifies whether a task is a milestone task (=TRUE) or not (= FALSE).  
NOTE In small project planning applications, a milestone task may be understood to be a task having no duration. As such, it represents a singular point in time.

**Priority** : A value that indicates the relative priority of the task (in comparison to the priorities of other tasks).

**TaskTime** : Time related information for the task.  
IFC4 CHANGE Attribute added

Formal Propositions:

**HasName** : The Name attribute should be inserted to describe the task name.

**CorrectPredefinedType** : The attribute ObjectType must be asserted when the value of PredefinedType is set to USERDEFINED.

Inheritance Graph:

**ENTITY** IfcTask

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;  
OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
Name : **OPTIONAL** IfcLabel;  
Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcProcess

Identification : **OPTIONAL** IfcStrippedOptional;  
LongDescription : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsPredecessorTo : **SET OF** IfcRelSequence **FOR** RelatingProcess;  
 IsSuccessorFrom : **SET OF** IfcRelSequence **FOR** RelatedProcess;  
**ENTITY** IfcTask  
 Status : **OPTIONAL** IfcStrippedOptional;  
 WorkMethod : **OPTIONAL** IfcStrippedOptional;  
 IsMilestone : IfcStrippedOptional;  
 Priority : **OPTIONAL** IfcStrippedOptional;  
 TaskTime : **OPTIONAL** IfcTaskTime;  
 PredefinedType : **OPTIONAL** IfcStrippedOptional;  
**END\_ENTITY;**

```
<xs:element name="IfcTask" type="ifc:IfcTask" substitutionGroup="ifc:IfcProcess" nillable="true"/>
<xs:complexType name="IfcTask">
  <xs:complexContent>
    <xs:extension base="ifc:IfcProcess">
      <xs:sequence>
        <xs:element name="TaskTime" type="ifc:IfcTaskTime" nillable="true" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.11 IfcActor Definition

The IfcActor defines all actors or human agents involved in a project during its full life cycle. It facilitates the use of person and organization definitions in the resource part of the IFC object model. This includes name, address, telecommunication addresses, and roles.

HISTORY New entity in IFC2.0

EXPRESS Specification:

**ENTITY** IfcActor  
**SUBTYPE OF** (IfcObject);  
 TheActor : IfcActorSelect;  
**INVERSE**  
**END\_ENTITY;**

Attribute Definitions:

**TheActor** : Information about the actor.

Inheritance Graph:

**ENTITY** IfcActor  
**ENTITY** IfcRoot  
 GlobalId : IfcGloballyUniqueId;  
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
 Name : **OPTIONAL** IfcLabel;  
 Description : **OPTIONAL** IfcText;  
**ENTITY** IfcObjectDefinition  
**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;  
**ENTITY** IfcObject  
 ObjectType : **OPTIONAL** IfcStrippedOptional;  
**INVERSE**  
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;  
**ENTITY** IfcActor  
 TheActor : IfcActorSelect;  
**INVERSE**

**END\_ENTITY;**

```

<xs:element name="IfcActor" type="ifc:IfcActor" substitutionGroup="ifc:IfcObject" nillable="true"/>
<xs:complexType name="IfcActor">
  <xs:complexContent>
    <xs:extension base="ifc:IfcObject">
      <xs:sequence>
        <xs:element name="TheActor">
          <xs:complexType>
            <xs:group ref="ifc:IfcActorSelect"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

**4.2.7.2.12 IfcActionRequest Definition**

A request is the act or instance of asking for something, such as a request for information, bid submission, or performance of work.

Requests may take many forms depending on the need including fault reports for maintenance, requests for small works, and purchase requests (where these are to be made through a help desk or buying function).

**HISTORY** New entity in IFC2x2.

**IFC4 CHANGE** Attribute *RequestID* renamed to *Identification* and promoted to supertype *IfcControl*, attributes *PredefinedType*, *Status*, and *LongDescription* added.

**EXPRESS** Specification:

**ENTITY** IfcActionRequest  
**SUBTYPE OF** (IfcControl);  
 PredefinedType : **OPTIONAL** IfcStrippedOptional;  
 Status : **OPTIONAL** IfcStrippedOptional;  
 LongDescription : **OPTIONAL** IfcStrippedOptional;  
**END\_ENTITY;**

## Attribute Definitions:

<b>Status</b>	: The status currently assigned to the request. Possible values include: Hold: wait to see if further requests are received before deciding on action NoAction: no action is required on this request Schedule: plan action to take place as part of maintenance or other task planning/scheduling Urgent: take action immediately IFC4 CHANGE The attribute has been added.
<b>LongDescription</b>	: Detailed description of the permit. IFC4 CHANGE The attribute has been added.

## Inheritance Graph:

**ENTITY** IfcActionRequest**ENTITY** IfcRoot

GlobalId	: IfcGloballyUniqueId;
OwnerHistory	: <b>OPTIONAL</b> IfcOwnerHistory;
Name	: <b>OPTIONAL</b> IfcLabel;
Description	: <b>OPTIONAL</b> IfcText;

**ENTITY** IfcObjectDefinition**INVERSE**

HasAssignments	: <b>SET OF</b> IfcRelAssigns <b>FOR</b> RelatedObjects;
HasContext	: <b>SET</b> [0:1] OF IfcRelDeclares <b>FOR</b> RelatedDefinitions;
IsDecomposedBy	: <b>SET OF</b> IfcRelAggregates <b>FOR</b> RelatingObject;
Decomposes	: <b>SET</b> [0:1] OF IfcRelAggregates <b>FOR</b> RelatedObjects;
HasAssociations	: <b>SET OF</b> IfcRelAssociates <b>FOR</b> RelatedObjects;

**ENTITY** IfcObject

ObjectType	: <b>OPTIONAL</b> IfcStrippedOptional;
------------	--

**INVERSE**

IsTypedBy	: <b>SET</b> [0:1] OF IfcRelDefinesByType <b>FOR</b> RelatedObjects;
IsDefinedBy	: <b>SET OF</b> IfcRelDefinesByProperties <b>FOR</b> RelatedObjects;

**ENTITY** IfcControl

Identification	: <b>OPTIONAL</b> IfcStrippedOptional;
----------------	--

**INVERSE**

Controls	: <b>SET OF</b> IfcRelAssignsToControl <b>FOR</b> RelatingControl;
----------	--

**ENTITY** IfcActionRequest

PredefinedType	: <b>OPTIONAL</b> IfcStrippedOptional;
Status	: <b>OPTIONAL</b> IfcStrippedOptional;
LongDescription	: <b>OPTIONAL</b> IfcStrippedOptional;

**END\_ENTITY;**

```

<xs:element name="IfcActionRequest" type="ifc:IfcActionRequest" substitutionGroup="ifc:IfcControl" nillable="true"/>
<xs:complexType name="IfcActionRequest">
  <xs:complexContent>
    <xs:extension base="ifc:IfcControl"/>
  </xs:complexContent>
</xs:complexType>

```

**4.2.7.2.13 IfcProcess Definition**



IfcProcess is defined as one individual activity or event, that is ordered in time, that has sequence relationships with other processes, which transforms input in output, and may connect to other other processes through input output relationships. An IfcProcess can be an activity (or task), or an event. It takes usually place in building construction with the intent of designing, costing, acquiring, constructing, or maintaining products or other and similar tasks or procedures. The figure below illustrates process relationships.

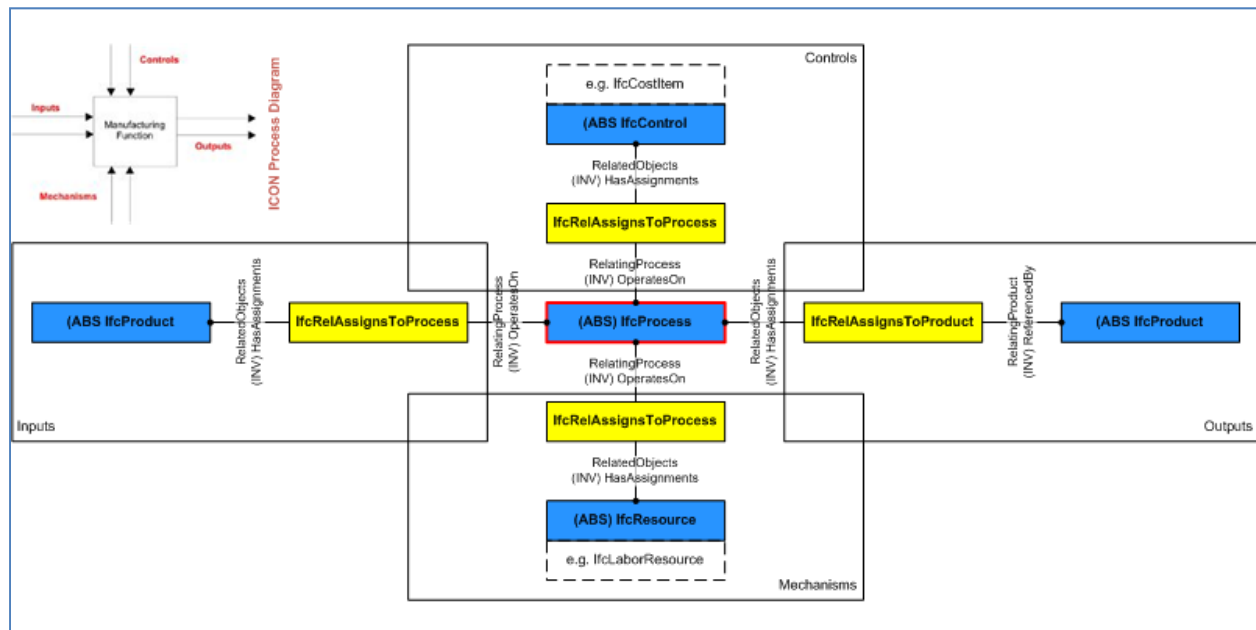


Figure 39 ifcProcess

NOTE Definition according to ISO9000: A process is a set of activities that are interrelated or that interact with one another. Processes use resources to transform inputs into outputs.

Processes are interconnected because the output from one process becomes the input for another process. In effect, processes are "glued" together by means of such input output relationships.

HISTORY New entity in IFC1.0.

IFC2x CHANGE The attribute Productivity has been removed.

IFC4 CHANGE The attribute Identification has been promoted from subtypes IfcTask and others.

### Relationship use definition

Process information relates to other objects by establishing the following relationships:

- Nesting of processes: IfcRelNests - A process can contain sub processes and thereby be nested.
- Sequencing of processes: IfcRelSequence - Processes can be placed in sequence (including overlapping for parallel tasks), and have predecessors and successors.

- Assigning process to schedules: IfcRelAssignsToControl - Activities such as tasks, and predominately summary tasks, are assigned to a work schedule.
- Having a product assigned to the process as input: IfcRelAssignsToProcess - Products can be assigned as input to a process, such as for construction process planning.
- Having a product assigned to the process as output: IfcRelAssignsToProduct - Products can be assigned as output to a process, such as for construction process planning.
- Having a control assigned to the process as process control: IfcRelAssignsToProcess - Items that act as a control onto the process can be assigned to a process, such as for cost management (a cost item assigned to a work task).
- Having a resource assigned to the process as consumed by the process: IfcRelAssignsToProcess - Items that act as a mechanism to a process, such as labor, material and equipment in cost calculations.

EXPRESS Specification:

**ENTITY** IfcProcess

**ABSTRACT SUPERTYPE OF**(IfcTask)

**SUBTYPE OF** (IfcObject);

Identification : **OPTIONAL** IfcStrippedOptional;

LongDescription : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsPredecessorTo : **SET OF** IfcRelSequence **FOR** RelatingProcess;

IsSuccessorFrom : **SET OF** IfcRelSequence **FOR** RelatedProcess;

**END\_ENTITY;**

Attribute Definitions:

**Identification** : An identifying designation given to a process or activity. It is the identifier at the occurrence level.

IFC4 CHANGE Attribute promoted from subtypes.

**LongDescription** : An extended description or narrative that may be provided.

IFC4 CHANGE New attribute.

**IsPredecessorTo** : Dependency between two activities, it refers to the subsequent activity for which this activity is the predecessor. The link between two activities can include a link type and a lag time.

**IsSuccessorFrom** : Dependency between two activities, it refers to the previous activity for which this activity is the successor. The link between two activities can include a link type and a lag time.

Inheritance Graph:

**ENTITY** IfcProcess

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;  
**ENTITY** IfcObject  
 ObjectType : **OPTIONAL** IfcStrippedOptional;  
**INVERSE**  
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;  
**ENTITY** IfcProcess

Identification : **OPTIONAL** IfcStrippedOptional;  
 LongDescription : **OPTIONAL** IfcStrippedOptional;  
**INVERSE**  
 IsPredecessorTo : **SET OF** IfcRelSequence **FOR** RelatingProcess;  
 IsSuccessorFrom : **SET OF** IfcRelSequence **FOR** RelatedProcess;  
**END\_ENTITY;**

```

<xs:element name="IfcProcess" type="ifc:IfcProcess" abstract="true" substitutionGroup="ifc:IfcObject" nillable="true"/>
<xs:complexType name="IfcProcess" abstract="true">
  <xs:complexContent>
    <xs:extension base="ifc:IfcObject"/>
  </xs:complexContent>
</xs:complexType>

```

#### 4.2.7.2.14 ifcGroup Definition

IfcGroup is a generalization of any arbitrary group. A group is a logical collection of objects. It does not have its own position, nor can it hold its own shape representation. Therefore a group is an aggregation under some non-geometrical / topological grouping aspects.

**NOTE** Use IfcRelDecomposes together with the appropriate subtypes of IfcProduct to define an aggregation of products that may have its own position and shape representation.

**EXAMPLE** An example for a group is a system, since it groups elements under the aspect of their role, regardless of their position in a building.

A group can hold any collection of objects (being products, processes, controls, resources, actors or other groups). Thus groups can be nested. An object can be part of zero, one, or many groups. Grouping relationships are not required to be hierarchical nor do they imply a dependency.

**NOTE** Use IfcRelDecomposes together with the appropriate subtypes of IfcProduct to define an hierarchical aggregation of products.

A group can be exchanged without having already objects within the group collection.

**HISTORY** New entity in IFC1.0.

**IFC4 CHANGE** The inverse IsGroupedBy relationship is set to 0..n

#### Relationship use definition

Groups are assigned to other objects (such as a process or a resource) by the relationship object that refers to the corresponding object:

- Process: assigned using IfcRelAssignsToProcess
- Resource: assigned using IfcRelAssignsToResource
- Groups can be subjected to a control. The control information is then assigned:
- Controls: affecting the group using IfcRelAssignsToControl

EXPRESS Specification:

**ENTITY IfcGroup**

**ENTITY IfcRoot**

GlobalId : IfcGloballyUniqueId;  
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;  
 Name : **OPTIONAL** IfcLabel;  
 Description : **OPTIONAL** IfcText;

**ENTITY IfcObjectDefinition**

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY IfcObject**

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY IfcGroup**

**INVERSE**

IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

**END\_ENTITY;**

```
<xs:element name="IfcGroup" type="ifc:IfcGroup" substitutionGroup="ifc:IfcObject" nillable="true"/>
<xs:complexType name="IfcGroup">
  <xs:complexContent>
    <xs:extension base="ifc:IfcObject"/>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.15 IfcPropertySet Definition

IfcPropertySet defines all dynamically extensible properties. The property set is a container class that holds properties within a property tree. These properties are interpreted according to their name attribute.

The same IfcPropertySet can be assigned to multiple object occurrences; it should then be assigned by a single instance of IfcRelDefinedByProperties to a set of related objects. Those property sets are referred to as shared property sets. It can also be assigned to an object type.

An `IfcPropertySetTemplate` may define the underlying structure, i.e. the required name, the applicable object or object types to which the property set can be attached, and the individual properties that maybe included.

**NOTE** See `IfcRelDefinesByType` for how to override property sets assigned to an object type within the object occurrence.

**HISTORY** New entity in IFC1.0

**IFC4 CHANGE** All statically defined property set entities are no longer subtypes of `IfcPropertySet`.

### Relationship use definition

Property sets are related to other objects by using the relationship object that refers to the corresponding object:

- **Occurrence Object:** `IfcRelDefinesByProperties` using the inverse attribute `DefinesOccurrence`.
- **Type Object:** using a direct link by inverse attribute `DefinesType`.
- **Underlying template:** `IfcRelDefinesByTemplate` using the inverse attribute `IsDefinedBy`.
- External reference: subtypes of `IfcRelAssociates` are used to provide a link to a classification system, or external library providing further reference to the property set. Accessible by inverse attribute `HasAssociations`.

### Attribute use definition

Instances of `IfcPropertySet` are used to assign named sets of individual properties (complex or single properties). Each individual property has a significant name string. Some property sets are included in the IFC specification and have a predefined set of properties indicated by assigning a significant name. These property sets are listed under "property sets" main menu item within this specification and from the object documentation sheet for those object to which they are applicable. The naming convention "Pset\_Xxx" applies to all those property sets that are defined as part of the IFC specification and it shall be used as the value of the `Name` attribute.

In addition any user defined property set can be captured. Property sets that are not declared as part of the IFC specification shall have a `Name` value not including the "Pset\_" prefix.

**EXPRESS Specification:**

```

ENTITY IfcPropertySet
  SUBTYPE OF (IfcPropertySetDefinition);
  HasProperties          : SET [1:?] OF IfcProperty;
  WHERE
    ExistsName          : EXISTS(SELF\IfcRoot.Name);
    UniquePropertyNames : IfcUniquePropertyName(HasProperties);
END_ENTITY;

```

**Attribute Definitions:**

**HasProperties** : Contained set of properties. For property sets defined as part of the IFC Object model, the property objects within a property set are defined as part of the standard. If a property is not

contained within the set of predefined properties, its value has not been set at this time.

Formal Propositions:

- ExistsName** : The *Name* attribute has to be provided. The attribute is used to specify the signifier of the property set. The properties that are allowed to be attached to a particular property set may be given within the property set definition part of the IFC specification. Those property set definitions are references in the semantic definition section of the individual subtypes of *IfcObjectDefinition*.
- UniquePropertyNames** : Every individual subtype of *IfcProperty* within the property set shall have a unique *Name* attribute value.

Inheritance Graph:

**ENTITY** *IfcPropertySet*

**ENTITY** *IfcRoot*

- GlobalId* : *IfcGloballyUniqueId*;  
*OwnerHistory* : **OPTIONAL** *IfcOwnerHistory*;  
*Name* : **OPTIONAL** *IfcLabel*;  
*Description* : **OPTIONAL** *IfcText*;

**ENTITY** *IfcPropertyDefinition*

**INVERSE**

- HasContext* : **SET** [0:1] OF *IfcRelDeclares* **FOR** *RelatedDefinitions*;  
*HasAssociations* : **SET** OF *IfcRelAssociates* **FOR** *RelatedObjects*;

**ENTITY** *IfcPropertySetDefinition*

**INVERSE**

- DefinesType* : **SET OF** *IfcTypeObject* **FOR** *HasPropertySets*;  
*DefinesOccurrence* : **SET** [0:1] OF *IfcRelDefinesByProperties* **FOR** *RelatingPropertyDefinition*;

**ENTITY** *IfcPropertySet*

- HasProperties* : **SET** [1:?] OF *IfcProperty*;

**END\_ENTITY**;

```
<xs:element name="IfcPropertySet" type="ifc:IfcPropertySet" substitutionGroup="ifc:IfcPropertySetDefinition"
nillable="true"/>
<xs:complexType name="IfcPropertySet">
  <xs:complexContent>
    <xs:extension base="ifc:IfcPropertySetDefinition">
      <xs:sequence>
        <xs:element name="HasProperties">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="ifc:IfcProperty" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute ref="ifc:itemType" fixed="ifc:IfcProperty"/>
            <xs:attribute ref="ifc:cType" fixed="set"/>
            <xs:attribute ref="ifc:arraySize" use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

#### 4.2.7.2.16 IfcAnnotation Definition

An annotation is a graphical representation within the geometric (and spatial) context of a project, that adds a note or meaning to the objects which constitutes the project model. Annotations include additional points, curves, text, dimensioning, hatching and other forms of graphical notes. It also include symbolic representations of additional model components, not representing products or spatial structures, such as survey points, contour lines or similar.

NOTE Additional presentation information (often 2D) such as tag number or hatching, that is directly related to a particular product representation is included within the IfcProductDefinitionShape having various IfcShapeRepresentation's of the IfcElement (and its subtypes). Only those presentation information, that cannot be directly related to a single product, have to be wrapped within the IfcAnnotation.

If available, the annotation should be related to the spatial context of the project, by containing the annotation within the appropriate level of the building structure (site, building, story, or space). This is handled by the IfcRelContainedInSpatialStructure relationship.

HISTORY New entity in IFC2x2.

#### Attribute use definition

The IfcAnnotation can provide specific 0D, 1D, and 2D geometric items as representation of the annotation, offering annotation point, curves, and surfaces. The following values shall be used for the ObjectType.

- **'Annotation point'** is an annotation provided by a point that has additional semantic. The inherited attribute ObjectType should be used to capture the type of point annotation, some predefined values are:
- **'SurveyPoint'**: A single survey point represented by a Cartesian point. A property set may add the conditions (method, accuracy, etc. to the survey point).
- **'SurveyArea'**: A set of survey points represented by Cartesian point. These coordinates are determined relative to the coordinates of a reference point, which acts as the datum for the survey. Properties attached apply equally to all points. The difference in elevation of the survey points enables terrain to be determined.
- **'Annotation curve'** is an annotation provided by a curve that has additional semantic. The inherited attribute ObjectType should be used to capture the type of curve annotation, some predefined values are:
- **'ContourLine'**: A line of constant elevation typically used on geographic maps where the spacing of lines at constant intervals of elevation may be used as an indication of slope.
- **'IsoBar'**: A line of constant pressure typically used on weather maps or to show pressure gradient in spaces, chambers or externally.
- **'IsoLux'**: A line of constant illumination typically used to show the distribution of illumination levels and/or daylighting in a space or externally.
- **'IsoTherm'**: A line of constant temperature typically used to show the distribution and effect of heating or cooling within a space or to show temperature distribution on a geographic map.
- **'Annotation surface'** is an annotation provided by a surface that has additional semantic. The inherited attribute ObjectType should be used to capture the type of surface annotation, some predefined values are:

- **'SurveyArea'**: A surface patch based on survey points.

EXPRESS Specification:

**ENTITY** IfcAnnotation

**SUBTYPE OF** (IfcProduct);

**INVERSE**

ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

**END\_ENTITY**;

Attribute Definitions:

**ContainedInStructure** : Relationship to a spatial structure element, to which the associate is primarily associated.

Inheritance Graph:

**ENTITY** IfcAnnotation

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;

Representation : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

**ENTITY** IfcAnnotation

**INVERSE**

ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

**END\_ENTITY**;

```
<xs:element name="IfcAnnotation" type="ifc:IfcAnnotation" substitutionGroup="ifc:IfcProduct" nillable="true"/>
<xs:complexType name="IfcAnnotation">
  <xs:complexContent>
    <xs:extension base="ifc:IfcProduct"/>
  </xs:complexContent>
</xs:complexType>
```



#### 4.2.7.2.17 IfcRelConnectsPorts Definition

An IfcRelConnectsPorts relationship defines the relationship that is made between two ports at their point of connection. It may include the connection geometry between two ports.

The objectified relationship IfcRelConnectsPorts is required for defining how two instances of IfcPort connect together. Each of the ports is logically contained within the IfcDistributionElement by using the ordered collection IfcRelNests.

HISTORY New entity in IFC2.0, modified in IFC2x.

IFC4 CHANGE Previously the containment of the IfcPort within the IfcDistributionElement had been realized using the IfcRelConnectsPortToElement relationship.

EXPRESS Specification:

```

ENTITY IfcRelConnectsPorts
  SUBTYPE OF (IfcRelConnects);
    RelatingPort      : IfcPort;
    RelatedPort       : IfcPort;
    RealizingElement  : OPTIONAL IfcStrippedOptional;
  WHERE
    NoSelfReference   : RelatingPort :<> RelatedPort;
END_ENTITY;

```

Attribute Definitions:

**RelatingPort** : Reference to the first port that is connected by the objectified relationship.  
**RelatedPort** : Reference to the second port that is connected by the objectified relationship.  
**RealizingElement** : Defines the element that realizes a port connection relationship.

Formal Propositions:

**NoSelfReference** : The instance of the *RelatingPort* shall not be the same instance as the *RelatedPort*.

Inheritance Graph:

```

ENTITY IfcRelConnectsPorts
  ENTITY IfcRoot
    GlobalId      : IfcGloballyUniqueId;
    OwnerHistory  : OPTIONAL IfcOwnerHistory;
    Name          : OPTIONAL IfcLabel;
    Description    : OPTIONAL IfcText;
  ENTITY IfcRelationship
  ENTITY IfcRelConnects
  ENTITY IfcRelConnectsPorts
    RelatingPort  : IfcPort;
    RelatedPort   : IfcPort;
    RealizingElement : OPTIONAL IfcStrippedOptional;
END_ENTITY;

```

<xs:element name="IfcRelConnectsPorts" type="ifc:IfcRelConnectsPorts" substitutionGroup="ifc:IfcRelConnects"

```

nillable="true"/>
<xs:complexType name="IfcRelConnectsPorts">
  <xs:complexContent>
    <xs:extension base="ifc:IfcRelConnects">
      <xs:sequence>
        <xs:element name="RelatingPort" type="ifc:IfcPort" nillable="true"/>
        <xs:element name="RelatedPort" type="ifc:IfcPort" nillable="true"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

#### 4.2.7.2.18 IfcWorkSchedule Definition

An IfcWorkSchedule represents a task schedule of a work plan, which in turn can contain a set of schedules for different purposes. The figure below shows the backbone structure of a work schedule that defines (1) a context through IfcRelDeclares (not necessarily the project) and (2) controls tasks (typically the schedule summary task) and resources. Please note that a work calendar shall be assigned to the summary task and not the work schedule.

HISTORY New entity in IFC2.0.

#### Declaration Use Definition

IfcWorkSchedule can reference a project (the single IfcProject instance) via IfcRelDeclares.

EXPRESS Specification:

**ENTITY** IfcWorkSchedule

**SUBTYPE OF** (IfcWorkControl);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

**WHERE**

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>  
IfcWorkScheduleTypeEnum.USERDEFINED) OR ((PredefinedType =  
IfcWorkScheduleTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcObject.ObjectType));

**END\_ENTITY;**

Attribute Definitions:

Formal Propositions:

**CorrectPredefinedType** : The attribute ObjectType must be asserted when the value of the IfcWorkScheduleTypeEnum is set to USERDEFINED.

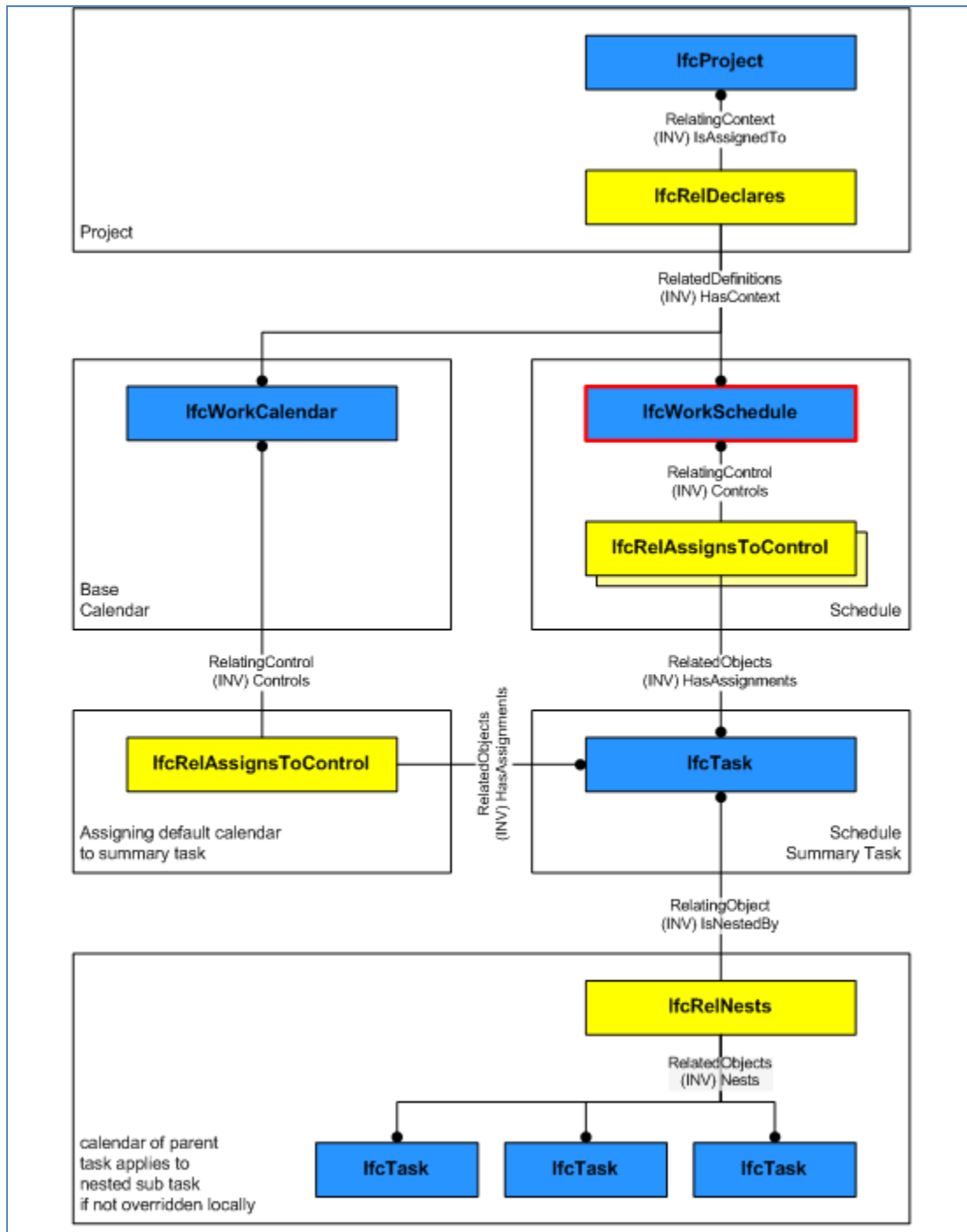


Figure 40 IfcWorkSchedule

Inheritance Graph:

**ENTITY** IfcWorkSchedule

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

**ENTITY** IfcObjectDefinition

**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY** IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY** IfcControl

Identification : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

Controls : **SET OF** IfcRelAssignsToControl **FOR** RelatingControl;

**ENTITY** IfcWorkControl

CreationDate : IfcDateTime;

Creators : **OPTIONAL SET** [1:?] OF IfcStrippedOptional;

Purpose : **OPTIONAL** IfcStrippedOptional;

Duration : **OPTIONAL** IfcStrippedOptional;

TotalFloat : **OPTIONAL** IfcStrippedOptional;

StartTime : IfcDateTime;

FinishTime : **OPTIONAL** IfcStrippedOptional;

**ENTITY** IfcWorkSchedule

PredefinedType : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

```
<xs:element name="IfcWorkSchedule" type="ifc:IfcWorkSchedule" substitutionGroup="ifc:IfcWorkControl"
nillable="true"/>
```

```
<xs:complexType name="IfcWorkSchedule">
```

```
<xs:complexContent>
```

```
<xs:extension base="ifc:IfcWorkControl"/>
```

```
</xs:complexContent>
```

```
</xs:complexType>
```

#### 4.2.7.2.19 IfcRelAssociatesDocument Definition

The objectified relationship (IfcRelAssociatesDocument) handles the assignment of a document information (items of the select IfcDocumentSelect) to objects occurrences (subtypes of IfcObject) or object types (subtypes of IfcTypeObject).

The relationship is used to assign a document reference or a more detailed document information to objects. A single document reference can be applied to multiple objects.

The inherited attribute `RelatedObjects` define the objects to which the document association is applied. The attribute `RelatingDocument` is the reference to a document reference, applied to the object(s).

**HISTORY** New entity in IFC2x.

**EXPRESS** Specification:

**ENTITY** `IfcRelAssociatesDocument`

**SUBTYPE OF** (`IfcRelAssociates`);

`RelatingDocument` : `IfcDocumentSelect`;

**END\_ENTITY**;

Attribute Definitions:

**RelatingDocument** : Document information or reference which is applied to the objects.

Inheritance Graph:

**ENTITY** `IfcRelAssociatesDocument`

**ENTITY** `IfcRoot`

`GlobalId` : `IfcGloballyUniqueId`;

`OwnerHistory` : **OPTIONAL** `IfcOwnerHistory`;

`Name` : **OPTIONAL** `IfcLabel`;

`Description` : **OPTIONAL** `IfcText`;

**ENTITY** `IfcRelationship`

**ENTITY** `IfcRelAssociates`

`RelatedObjects` : **SET** [1:?] **OF** `IfcDefinitionSelect`;

**ENTITY** `IfcRelAssociatesDocument`

`RelatingDocument` : `IfcDocumentSelect`;

**END\_ENTITY**;

```
<xs:element name="IfcRelAssociatesDocument" type="ifc:IfcRelAssociatesDocument"
substitutionGroup="ifc:IfcRelAssociates" nillable="true"/>
```

```
<xs:complexType name="IfcRelAssociatesDocument">
```

```
<xs:complexContent>
```

```
<xs:extension base="ifc:IfcRelAssociates">
```

```
<xs:sequence>
```

```
<xs:element name="RelatingDocument">
```

```
<xs:complexType>
```

```
<xs:group ref="ifc:IfcDocumentSelect"/>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
</xs:sequence>
```

```
</xs:extension>
```

```
</xs:complexContent>
```

```
</xs:complexType>
```

#### 4.2.7.2.20 `IfcRelConnectsWithRealizingElements` Definition

`IfcRelConnectsWithRealizingElements` defines a generic relationship that is made between two elements that require the realization of that relationship by means of further realizing elements.

An `IfcRelConnectsWithRealizingElements` is a specialization of `IfcRelConnectsElement` where the connecting operation has the additional attribute of (one or many) realizing elements that may be used to realize or further qualify the relationship. It is defined as a ternary relationship.

**EXAMPLE** It may be used to describe the attachment of one element to another where the attachment is realized by a 'fixing' element such as a bracket. It may also be used to describe the mounting of one element onto another such as the requirement for the mounting major plant items onto builders work bases and/or anti-vibration isolators.

**HISTORY** New entity in IFC2x2.

**EXPRESS** Specification:

**ENTITY** `IfcRelConnectsWithRealizingElements`

**SUBTYPE OF** (`IfcRelConnectsElements`);

`RealizingElements` : **SET** [1:?] **OF** `IfcElement`;

`ConnectionType` : **OPTIONAL** `IfcStrippedOptional`;

**END\_ENTITY**;

Attribute Definitions:

**RealizingElements** : Defines the elements that realize a connection relationship.

**ConnectionType** : The type of the connection given for informal purposes, it may include labels, like 'joint', 'rigid joint', 'flexible joint', etc.

Inheritance Graph:

**ENTITY** `IfcRelConnectsWithRealizingElements`

**ENTITY** `IfcRoot`

`GlobalId` : `IfcGloballyUniqueId`;

`OwnerHistory` : **OPTIONAL** `IfcOwnerHistory`;

`Name` : **OPTIONAL** `IfcLabel`;

`Description` : **OPTIONAL** `IfcText`;

**ENTITY** `IfcRelationship`

**ENTITY** `IfcRelConnects`

**ENTITY** `IfcRelConnectsElements`

`ConnectionGeometry` : **OPTIONAL** `IfcStrippedOptional`;

`RelatingElement` : `IfcElement`;

`RelatedElement` : `IfcElement`;

**ENTITY** `IfcRelConnectsWithRealizingElements`

`RealizingElements` : **SET** [1:?] **OF** `IfcElement`;

`ConnectionType` : **OPTIONAL** `IfcStrippedOptional`;

**END\_ENTITY**;

```
<xs:element name="IfcRelConnectsWithRealizingElements" type="ifc:IfcRelConnectsWithRealizingElements"
substitutionGroup="ifc:IfcRelConnectsElements" nillable="true"/>
<xs:complexType name="IfcRelConnectsWithRealizingElements">
<xs:complexContent>
<xs:extension base="ifc:IfcRelConnectsElements">
<xs:sequence>
<xs:element name="RealizingElements">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element ref="ifc:IfcElement" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="ifc:itemType" fixed="ifc:IfcElement"/>
  <xs:attribute ref="ifc:cType" fixed="set"/>
  <xs:attribute ref="ifc:arraySize" use="optional"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

#### 4.2.7.2.21 IfcConstructionProductResource Definition

IfcConstructionProductResource defines the role of a product that is consumed (wholly or partially), or occupied in the performance of construction.

HISTORY New entity in IFC2.0. Renamed from IfcProductResource in IFC2x.

Occurrences of IfcConstructionProductResource are usage of products to assist the process of construction. More specifically, they are usage of products that result from some construction processes and that are then used as resources to facilitate further construction. For instance, formworks can be instantiated as products resulting from the process 'constructing formwork'. However, they are used as resources in the process 'pouring concrete' in a later stage of the project.

EXPRESS Specification:

**ENTITY** IfcConstructionProductResource

**SUBTYPE OF** (IfcConstructionResource);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

**WHERE**

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>  
IfcConstructionProductResourceTypeEnum.USERDEFINED) OR ((PredefinedType =  
IfcConstructionProductResourceTypeEnum.USERDEFINED) AND EXISTS  
(SELF\IfcObject.ObjectType));

**END\_ENTITY;**

Attribute Definitions:

**PredefinedType** : Defines types of construction product resources.  
IFC4 New attribute.

Inheritance Graph:

**ENTITY** IfcConstructionProductResource

**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

**ENTITY IfcObjectDefinition****INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;  
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;  
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;  
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;  
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

**ENTITY IfcObject**

ObjectType : **OPTIONAL** IfcStrippedOptional;

**INVERSE**

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;  
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

**ENTITY IfcResource**

Identification : **OPTIONAL** IfcStrippedOptional;  
 LongDescription : **OPTIONAL** IfcStrippedOptional;

**INVERSE****ENTITY IfcConstructionResource**

Usage : **OPTIONAL** IfcStrippedOptional;  
 BaseCosts : **OPTIONAL LIST** [1:?] OF IfcStrippedOptional;  
 BaseQuantity : **OPTIONAL** IfcStrippedOptional;

**ENTITY IfcConstructionProductResource**

PredefinedType : **OPTIONAL** IfcStrippedOptional;

**END\_ENTITY;**

```
<xs:element name="IfcConstructionProductResource" type="ifc:IfcConstructionProductResource"
substitutionGroup="ifc:IfcConstructionResource" nillable="true"/>
<xs:complexType name="IfcConstructionProductResource">
  <xs:complexContent>
    <xs:extension base="ifc:IfcConstructionResource"/>
  </xs:complexContent>
</xs:complexType>
```

**4.2.7.2.22 IfcConstructionProductResourceType Definition**

The resource type IfcConstructionProductType defines commonly shared information for occurrences of construction product resources. The set of shared information may include:

- common productivities
- common cost rates
- common properties within shared property sets

It is used to define a construction product resource specification (i.e. the specific resource information that is common to all occurrences of that resource). Resource types may be exchanged without being already assigned to occurrences.

Occurrences of the IfcConstructionProductResourceType are represented by instances of IfcConstructionProductResource.

**HISTORY** New entity in IFC4.



## Assignment use definition

In addition to assignments specified at the base class `IfcConstructionResourceType`, a construction product resource type may have assignments of its own using `IfcRelAssignsToResource` where `RelatingResource` refers to the `IfcConstructionProductResourceType` and `RelatedObjects` contains one or more `IfcTypeProduct` subtypes. Such relationship indicates the type of product to be used as input, which is instantiated as an occurrence assigned for each resource occurrence. There may be multiple chains of production where such product type may have its own task and resource types assigned.

EXPRESS Specification:

**ENTITY** `IfcConstructionProductResourceType`

**SUBTYPE OF** (`IfcConstructionResourceType`);

`PredefinedType` : `IfcConstructionProductResourceTypeEnum`;

**WHERE**

`CorrectPredefinedType` : (`PredefinedType` <> `IfcConstructionProductResourceTypeEnum.USERDEFINED`) OR  
 ((`PredefinedType` = `IfcConstructionProductResourceTypeEnum.USERDEFINED`) AND  
 EXISTS(`SELF`\`IfcTypeResource.ResourceType`));

**END\_ENTITY**;

Attribute Definitions:

**PredefinedType** : Defines types of construction product resources.

Inheritance Graph:

**ENTITY** `IfcConstructionProductResourceType`

**ENTITY** `IfcRoot`

`GlobalId` : `IfcGloballyUniqueId`;

`OwnerHistory` : **OPTIONAL** `IfcOwnerHistory`;

`Name` : **OPTIONAL** `IfcLabel`;

`Description` : **OPTIONAL** `IfcText`;

**ENTITY** `IfcObjectDefinition`

**INVERSE**

`HasAssignments` : **SET OF** `IfcRelAssigns` **FOR** `RelatedObjects`;

`HasContext` : **SET** [0:1] **OF** `IfcRelDeclares` **FOR** `RelatedDefinitions`;

`IsDecomposedBy` : **SET OF** `IfcRelAggregates` **FOR** `RelatingObject`;

`Decomposes` : **SET** [0:1] **OF** `IfcRelAggregates` **FOR** `RelatedObjects`;

`HasAssociations` : **SET OF** `IfcRelAssociates` **FOR** `RelatedObjects`;

**ENTITY** `IfcTypeObject`

`ApplicableOccurrence` : **OPTIONAL** `IfcStrippedOptional`;

`HasPropertySets` : **OPTIONAL SET** [1:?] **OF** `IfcPropertySetDefinition`;

**INVERSE**

`Types` : **SET** [0:1] **OF** `IfcRelDefinesByType` **FOR** `RelatingType`;

**ENTITY** `IfcTypeResource`

`Identification` : **OPTIONAL** `IfcStrippedOptional`;

`LongDescription` : **OPTIONAL** `IfcStrippedOptional`;

`ResourceType` : **OPTIONAL** `IfcStrippedOptional`;

**INVERSE**

**ENTITY IfcConstructionResourceType**BaseCosts : **OPTIONAL LIST** [1:?] **OF** IfcStrippedOptional;BaseQuantity : **OPTIONAL** IfcStrippedOptional;**ENTITY IfcConstructionProductResourceType**

PredefinedType : IfcConstructionProductResourceTypeEnum;

**END\_ENTITY;**

```

<xs:element name="IfcConstructionProductResourceType" type="ifc:IfcConstructionProductResourceType"
substitutionGroup="ifc:IfcConstructionResourceType" nillable="true"/>
<xs:complexType name="IfcConstructionProductResourceType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcConstructionResourceType">
      <xs:attribute name="PredefinedType" type="ifc:IfcConstructionProductResourceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

**4.2.7.3 Data definition reference schema list**

ISO 16739:2013, “Industry Foundation Classes (IFC) for data sharing in the construction and facilities management industries,” commonly called IFC4, is the reference standard schema for data definitions in this MVD. Use of IFC4 in this MVD is optional. Backward compatibility to IFC 2x3 is maintained for all use of COBie and all testing described in this standard.

**4.2.7.4 Concept list**

The concepts used in this MVD are as follows:

- Identity
- Revision Control
- External Data Constraints
- Classification
- Spatial Composition
- Quantities on Occurrences
- Conversion Units
- Project Declaration
- Group Assignment
- Object Typing
- Spatial Containment
- Property Sets for Occurrences
- Property Sets for Types
- Task Scheduling
- Sequential Connectivity
- Contact
- Control Assignment

**4.2.7.5 Concepts definition**

Each entity is listed with corresponding concepts and a graph of attributes and objects representing the combination of all concepts applied to instances of the entity. Each block in the diagram represents an entity, where the entity name is shown at the top of the block with background in black. Each attribute within the entity is shown in order, where black is used to indicate a direct attribute and grey is used to indicate an inverse attribute. Notation to the right of each attribute indicates aggregation, where S indicates a SET (unordered unique objects) and L indicates a LIST (ordered objects), the first number in brackets indicates the minimum count, and the second number in brackets indicates the maximum count





#### 4.2.7.5.2 ifcBuildingStory concepts definition

A building story has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound. The diagram below illustrates the related entities, their expected cardinality, and properties.

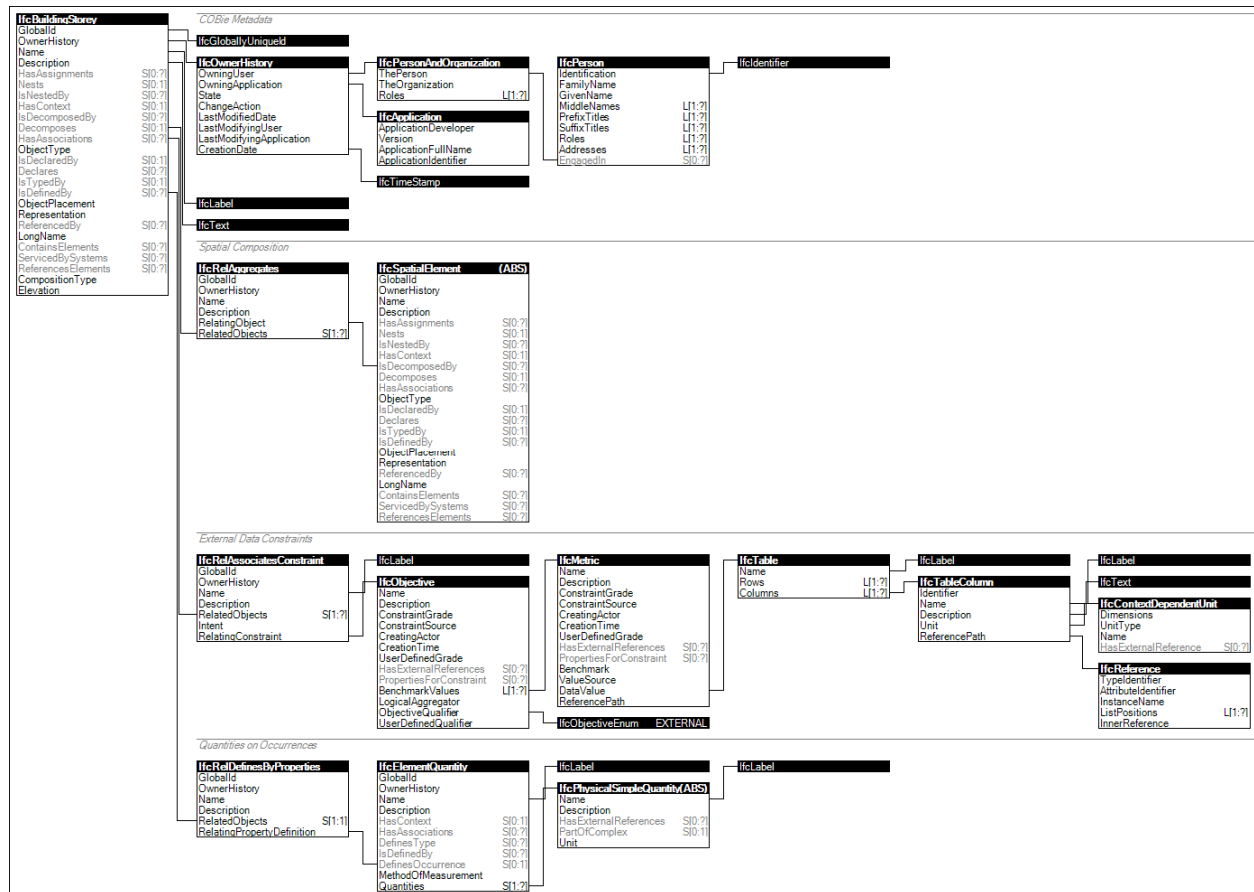


Figure 42 ifcBuildingStory Concepts

#### 4.2.7.5.3 ifcBuilding concepts definition

A building represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, story, and space). The diagram below illustrates the related entities, their expected cardinality, and properties.

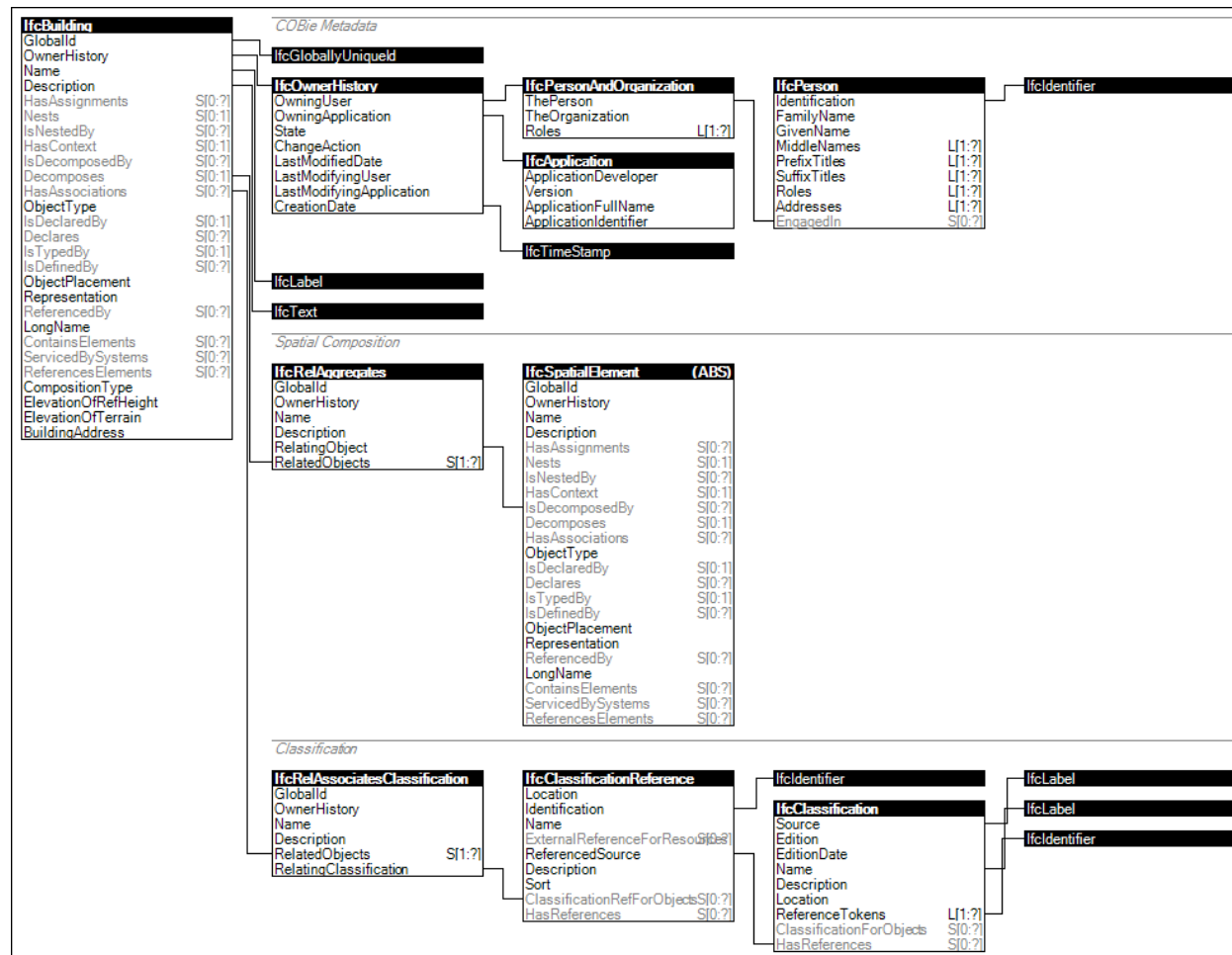


Figure 43 ifcBuilding Concepts

#### 4.2 7.5.4 ifcSite concepts definition

A site is a defined area of land, possibly covered with water, on which the project construction is to be completed. A site may be used to erect building(s) or other AEC products. The diagram below illustrates the related entities, their expected cardinality, and properties.

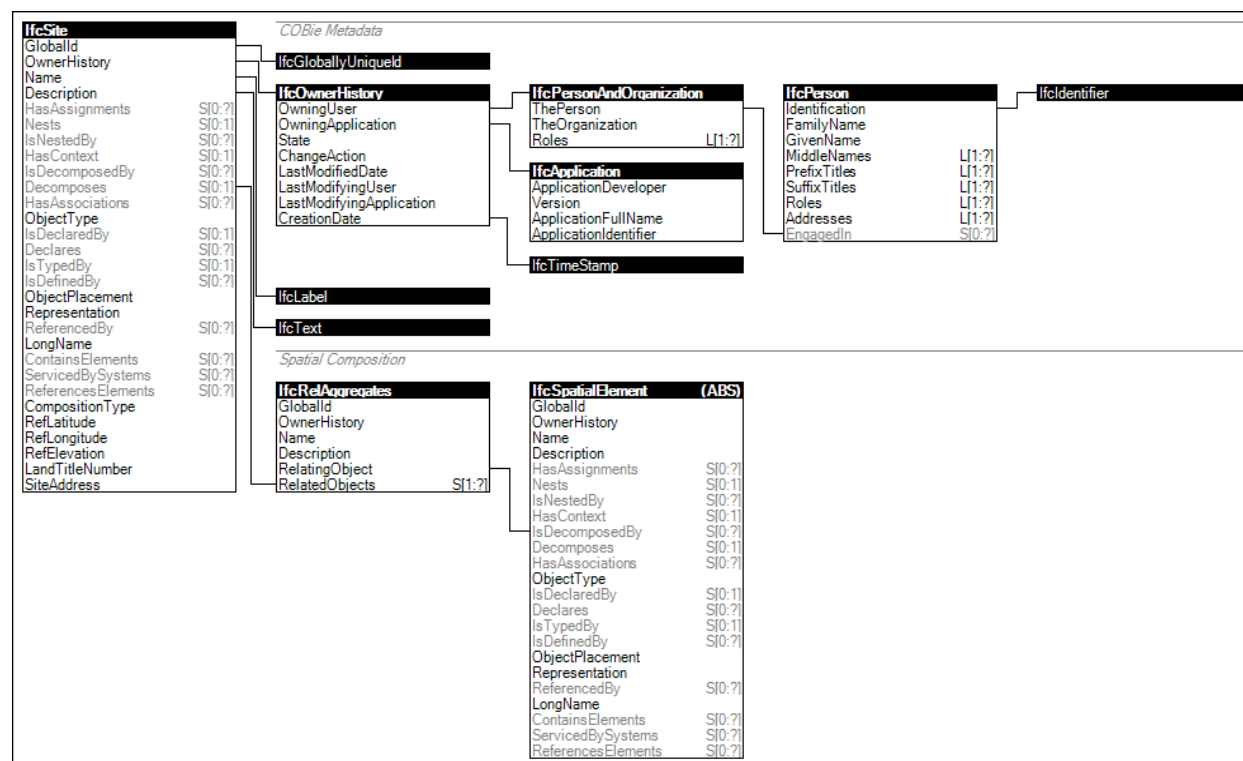


Figure 44 ifcSite Concepts

#### 4.2.7.5.5 ifcProject concepts definition

All IFC files must contain a single IfcProject instance that provides global context information including default units, and provides a directory of objects organized within the project.

Conversion Units:

Project Units define the units used in the data exchange. Units can be translated during export and import; that is, the originating model, data exchange and resulting model may all have different units.

The following units must be supported in all exchange scenarios:

- Length
- Area
- Volume
- Angle
- Currency

Additional units must be supported if they are needed in the context of an exchange scenario. Each measure value of the same type in the data exchange uses the same unit. When an application imports data, the following behavior relating to units is allowed:

Importing units as they are, providing the units match the units used by the importing application.

Translating to units of the same unit system, e.g. millimeters to meters.

Translating between unit systems, e.g. from millimeter to inch. The user must be prompted to accept the translation or cancel the import.

Informing the user about unsupported units and cancelling the import.

The following behavior is not allowed:

Importing units of a supported unit system without translation, if the units in the receiving application are not set to match the units used in the exchange, e.g. importing millimeters as meters.

Importing units of an unsupported unit system without translation, e.g. importing millimeters as inches without translation.

Mixing units of different unit systems is not allowed. Importing applications are required to detect invalid unit configurations, notify the user and cancel the import.

Additional restrictions have been imposed on units, with the requirement that angles must be measured in degrees (as opposed to radians).

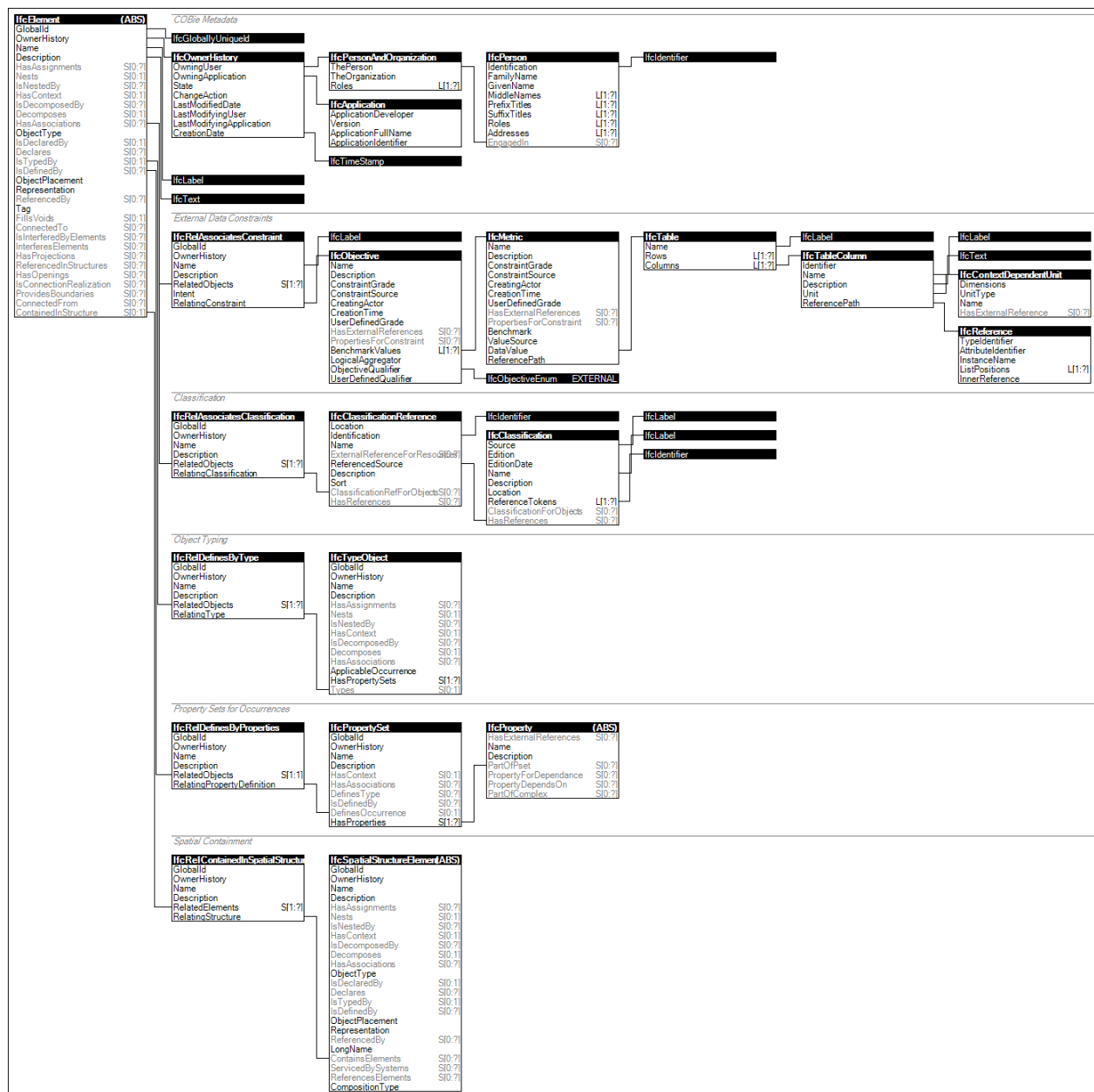






#### 2.7.5.8 ifcElement concepts definition

An element represents a physical building component. All subtypes of `lfcElement` are within scope of this MVD (for example, `lfcBoiler`), however instantiable subtypes are not incorporated within this documentation, for brevity.



### Figure 47 ifcElement Concepts

#### 4.2.7.5.9 ifcElementType concepts definition

The diagram below illustrates the related entities, their expected cardinality, and properties.

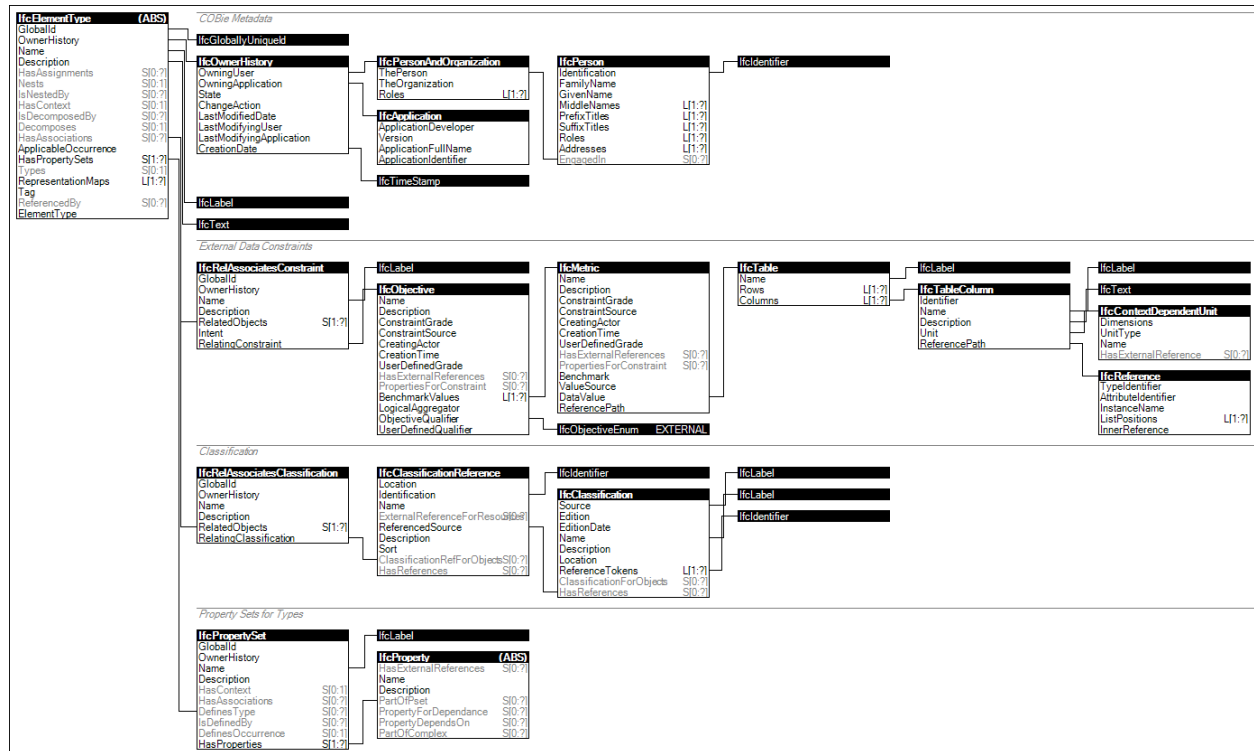
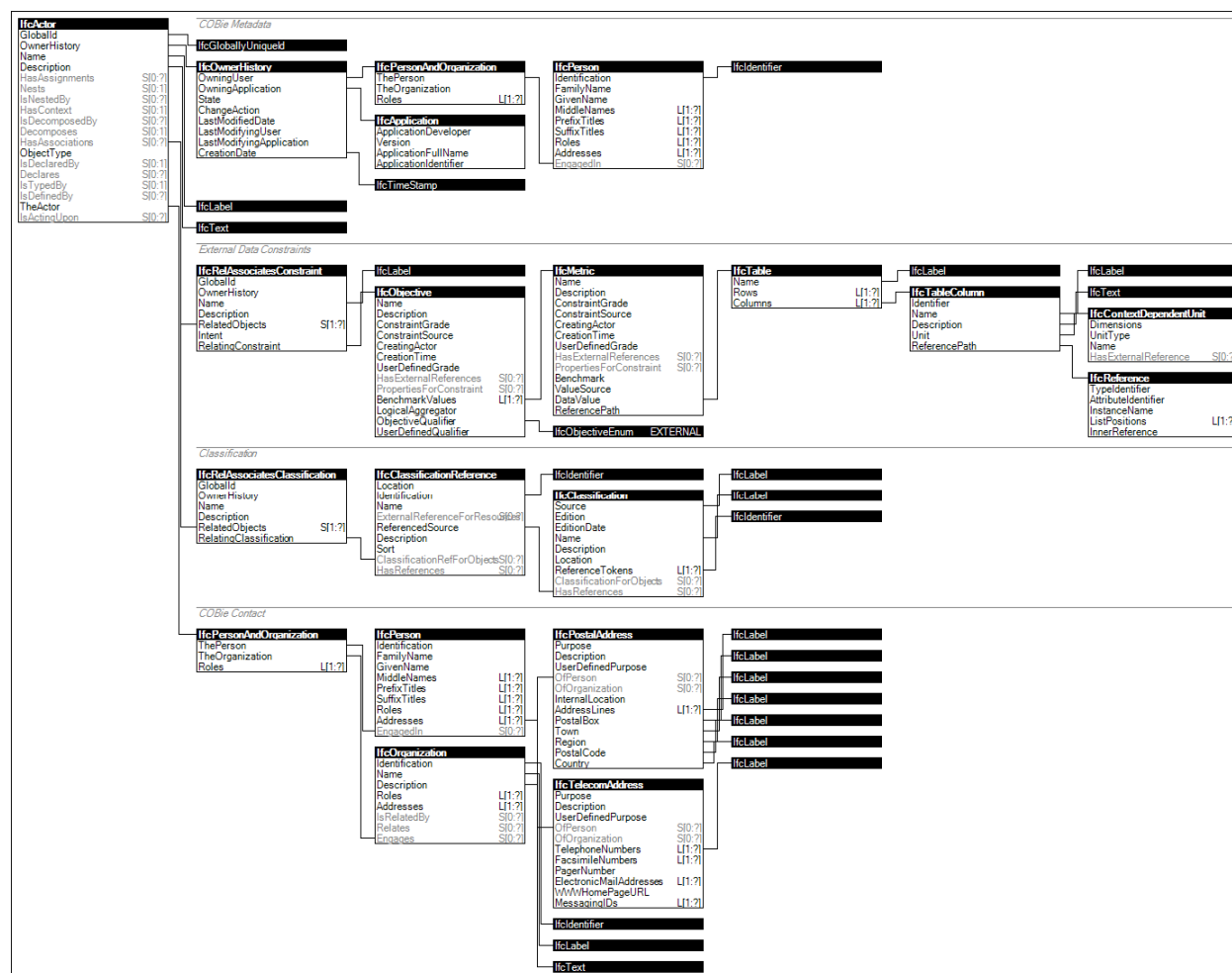


Figure 48 ifcElementType Concepts



#### 4.2.7.5.11 ifcActor concepts definition

For COBie, this entity is used to hold contact information. While COBie does not elaborate specific relationships between people and other objects (such as a person occupying a space), the user who currently owns an object is indicated using `lfcOwnerHistory`, where such contact information may be located by finding the corresponding `lfcActor`. The diagram below illustrates the related entities, their expected cardinality, and properties.



### Figure 50 ifcActor Concepts

#### 4.2.7.5.12 ifcActionRequest concepts definition

The diagram below illustrates the related entities, their expected cardinality, and properties.

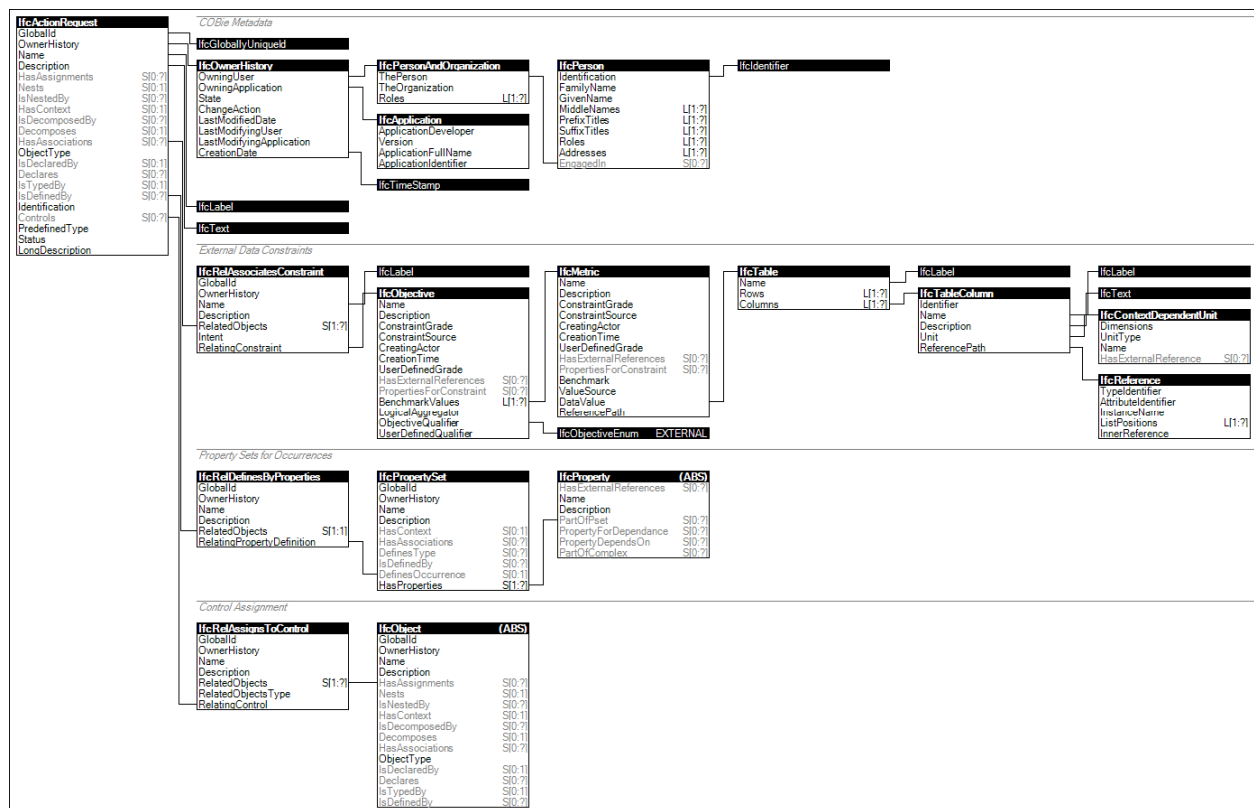


Figure 51 ifcActionRequest Concepts

#### 4.2.7.5.13 ifcProcess concepts definition

A process is a set of activities that are interrelated or that interact with one another. Processes use resources to transform inputs into outputs. Processes are interconnected because the output from one process becomes the input for another process. In effect, processes are "glued" together by means of such input-output relationships.

The diagram below illustrates the related entity.

<b>IfcProcess (ABS)</b>	
GlobalId	
OwnerHistory	
Name	
Description	
HasAssignments	S[0:?]
Nests	S[0:1]
IsNestedBy	S[0:?]
HasContext	S[0:1]
IsDecomposedBy	S[0:?]
Decomposes	S[0:1]
HasAssociations	S[0:?]
<b>ObjectType</b>	
IsDeclaredBy	S[0:1]
Declares	S[0:?]
IsTypedBy	S[0:1]
IsDefinedBy	S[0:?]
<b>Identification</b>	
LongDescription	
IsPredecessorTo	S[0:?]
IsSuccessorFrom	S[0:?]
OperatesOn	S[0:?]

**Figure 52 ifcProcess Concepts**

#### 4.2.7.5.14

#### ifcGroup concepts definition

A group is an organization of related items for a functional purpose. Specific subtypes are used for particular scenarios.

The diagram below illustrates the related entity.



IfcGroup	
GlobalId	
OwnerHistory	
Name	
Description	
HasAssignments	S[0:??]
Nests	S[0:1]
IsNestedBy	S[0:??]
HasContext	S[0:1]
IsDecomposedBy	S[0:??]
Decomposes	S[0:1]
HasAssociations	S[0:??]
ObjectType	
IsDeclaredBy	S[0:1]
Declares	S[0:??]
IsTypedBy	S[0:1]
IsDefinedBy	S[0:??]
IsGroupedBy	S[0:??]

Figure 53 ifcGroup Concepts

#### 4.2.7.5.15 ifcPropertySet concepts definition

The diagram below illustrates the related entities, their expected cardinality, and properties.

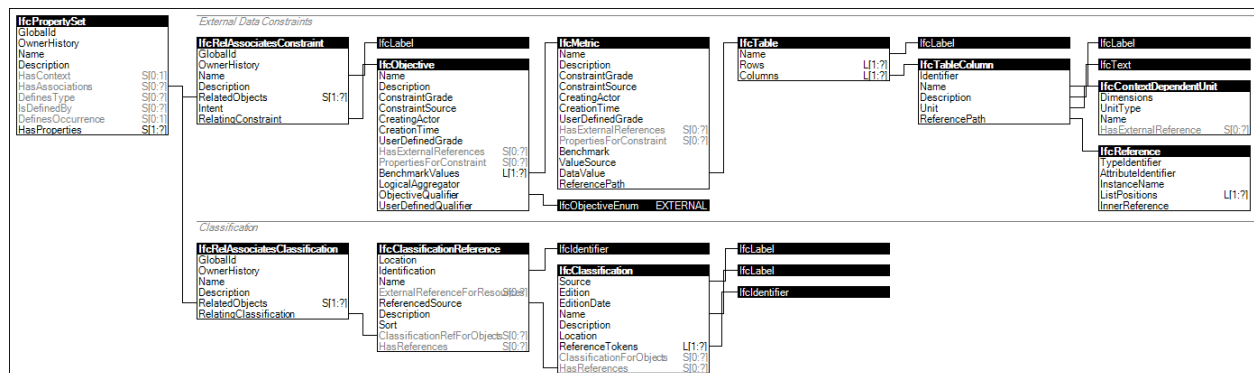


Figure 54 ifcPropertySetConcepts

#### 4.2.7.5.16 ifcAnnotation concepts definition

The diagram below illustrates the related entities, their expected cardinality, and properties.

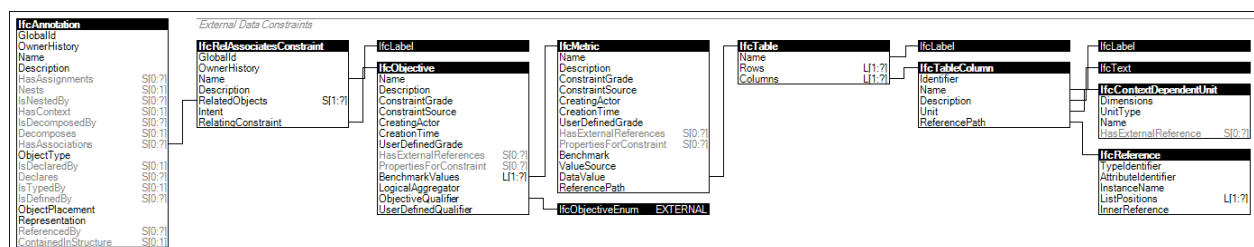


Figure 55 ifcAnnotationConcepts

#### 4.2.7.5.17 ifcRelConnectsPorts concepts definition

The diagram below illustrates the related entity.

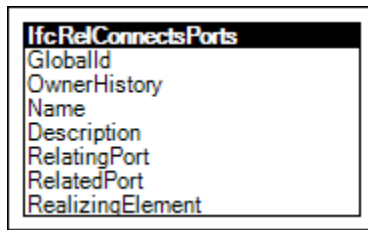


Figure 56 ifcRelConnectsPort Concept

#### 4.2.7.5.18 ifcWorkSchedule concepts definition

The diagram below illustrates the related entities, their expected cardinality, and properties.

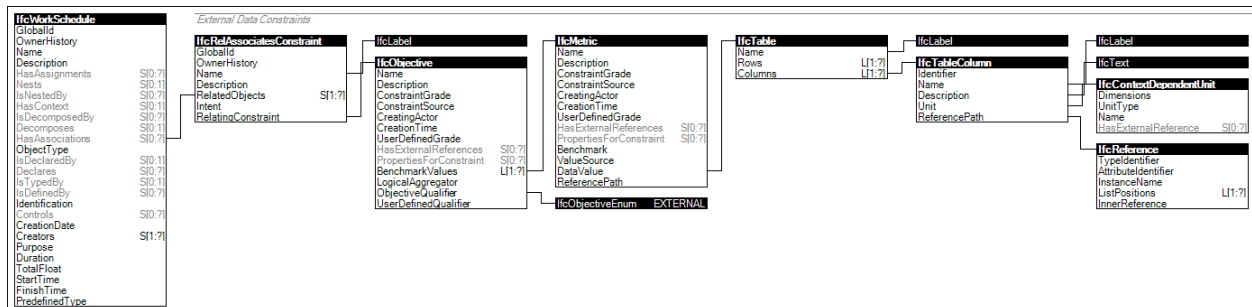


Figure 57 ifcWorkScheduleConcepts

#### 4.2.7.5.19 ifcRelAssociatesDocument concepts definition

The diagram below illustrates the related entity.

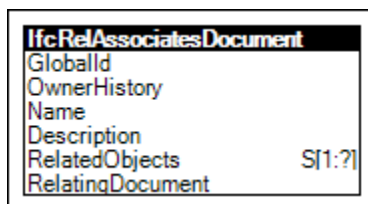


Figure 58 ifcRelAssociates Document

#### 4.2.7.5.20 ifcRelConnectsWithRealizingElement concepts definition

The diagram below illustrates the related entity.

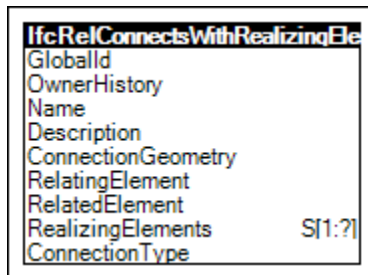


Figure 59 ifcRelConnectsWithRealizingElement Concepts

#### 4.2.7.5.21 ifcConstructionProductResource concepts definition

The diagram below illustrates the related entities, their expected cardinality, and properties.

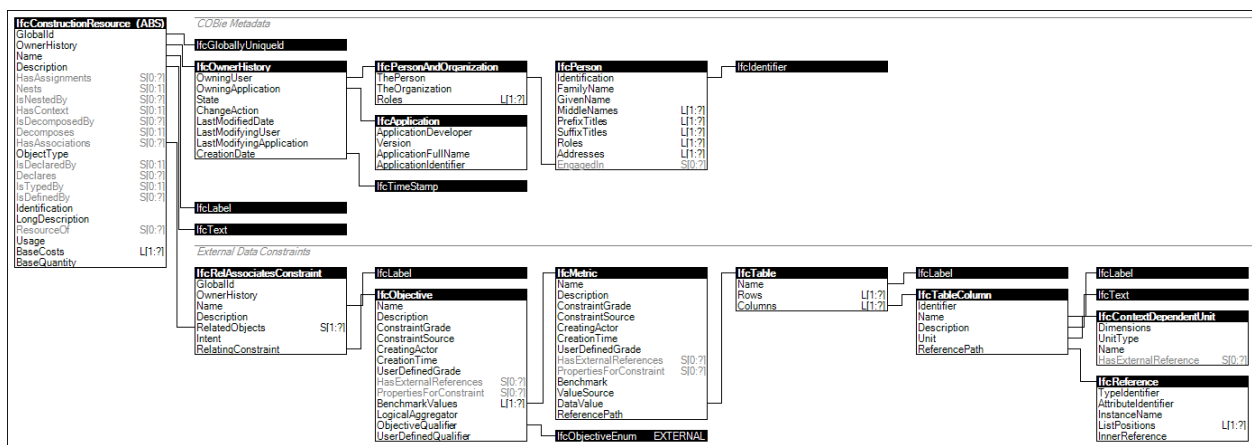


Figure 60 ifcConstructionProductResource Concept

#### 4.2.7.5.22 ifcConstructionProductResourceType Concepts Definition

The diagram below illustrates the related entity.

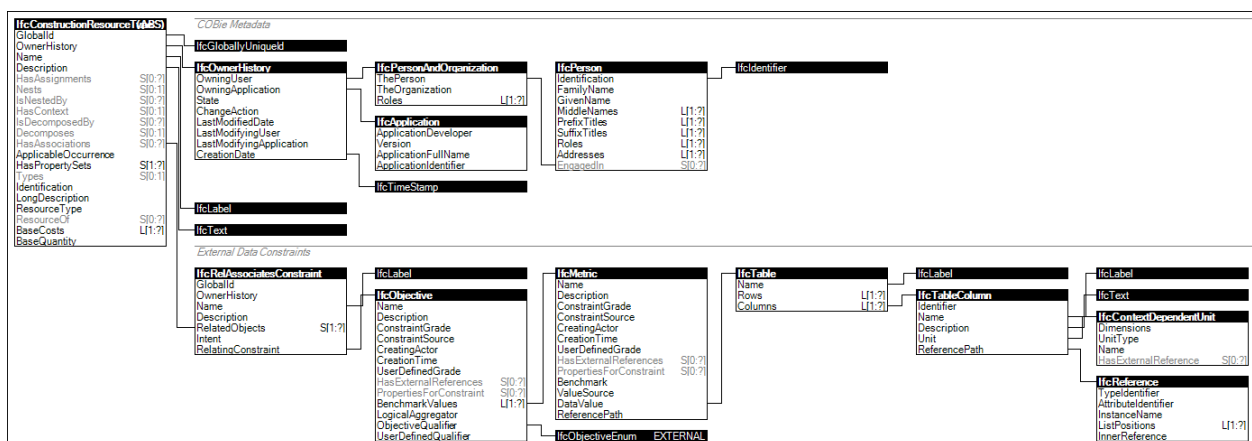


Figure 61 ifcConstructionproductResourceType Concept

#### 4.2.7.6 Concept attributes list

Concepts are also required to have parameters, properties, or attributes that specify applicable values. For example, plumbing objects may make use of ports to enable connectivity to other objects for distribution of water, and a specific entity such as a hot water heater may have specific ports such as “ColdWaterIn” and “HotWaterOut”. Defining attributes at concepts enables re-use of concepts where the data structures are the same, but applicable values may differ.

##### 4.2.7.6.1 External data constraints

To apply COBie in contexts outside those where STEP files might be used, this standard provides optional transformations between the underlying standard IFC specifications to schemas developed using SpreadsheetML (COBie Spreadsheet) and NIEM (COBieLite). The specifics of these alternative, optional schemas are found in Appendix A.

Transformation software, called the COBie ToolKit, is provided as open-source software. Testing using that COBie ToolKit has verified loss-less transformation between STEP, COBieSpreadsheet, and COBieLite formats (for in-scope COBie entities).

##### 4.2.7.6.2 Classification constraints

The following classifications are provided as default values in COBie. Specific regional or client classifications may replace these classifications.

**Table 48 Classification Constraints**

Entity	Source	Name	Tokens
IfcSpace	Construction Specifications Institute	Omniclass Table 13	13-00 00 00
IfcBuilding	Construction Specifications Institute	OmniClass Table 11	11-00 00 00
IfcZone	Construction Specifications Institute	OmniClass Table 14	14-00 00 00
IfcSystem	Construction Specifications Institute	OmniClass Table 21	21-00 00 00
IfcElement	Construction Specifications Institute	OmniClass Table 22	22-00 00 00
IfcElementType	Construction Specifications Institute	OmniClass Table 23	23-00 00 00
IfcTask	Construction Specifications Institute	OmniClass Table 31	31-00 00 00
IfcActor	Construction Specifications Institute	OmniClass Table 34	34-00 00 00
IfcPropertySet	Construction Specifications Institute	OmniClass Table 49	49-00 00 00

##### 4.2.7.6.3 Spatial composition constraints

The following entities are required to be present to comply with COBie business rules regarding spatial constraints:

- IfcSpace

- IfcBuildingStory
- IfcBuilding
- IfcSite

#### **4.2.7.6.4 Quantities on occurrence constraints**

The following IFC quantities are required to be present for any instances of ifcSpace:

- IfcQuantityLength
- IfcQuantityArea (Net)
- IfcQuantityArea (Gross)

The following IFC quantity is required to be present for any instances of ifcSpace:

- IfcQuantityLength

#### **4.2.7.6.5 Conversion unit constraints**

The following IFC conversion units are required to be associated with the ifcProject entity:

- LENGTHUNIT
- AREAUNIT
- VOLUMEUNIT
- CURRENCYUNIT

#### **4.2.7.6.6 Project declaration constraints**

The following IFC entities are required for Project Declarations:

- IfcActor
- IfcActionRequest
- IfcSystem
- IfcZone
- IfcTask
- IfcConstructionResource
- IfcTypeProduct
- IfcConstructionResourceType

#### **4.2.7.6.7 Group assignment constraints**

The following group assignments are allowed in COBie:

- ifcZone – comprised of classes of groups of ifcSpace entities
- ifcSystem – comprised of classes of groups of ifcElement entities

#### 4.2.7.6.8 Object typing constraints

The following object typing is required in COBie:

- ifcElements are required to be typed according to ifcElementType objects

#### 4.2.7.6.9 Spatial containment constraints

Use of geometry is limited within COBie to reflect the requirement to locate specific managed assets within spaces. As a result the following spatial containment is required in COBie:

- ifcElements are required to be located within one ifcSpace objects

#### 4.2.7.6.10 Property sets for occurrences constraints

The following property sets shall be used for all property sets on occurrences:

**Table 49 Property Sets for Occurrences Constraints**

Entity	Name
IfcElement	Pset_ManufacturerOccurrence
IfcTask	COBie_Task
IfcActionRequest	Pset_ActionRequest
	COBie_ActionRequest
IfcConstructionProductResource	COBie_ConstructionProductResource

#### 4.2.7.6.11 Property Sets for Type Constraints

The following property sets shall be used for all property sets on types:

**Table 50 Property Sets for Types Constraints**

Entity	Name
IfcElementType	COBie_ElementType
	Pset_ManufacturerTypeInfo
IfcConstructionProductResourceType	COBie_ConstructionProductResourceType

#### 4.2.7.6.12 Control Assignment Constraints

The following control assignment is required in COBie:

- ifcActionRequest objects must be assigned to ifcElement objects

#### 4.2.7.7 Concept Relationship Description

Concepts may inherit from other concepts such that more generic rules may be defined at a higher level and more specific rules at a lower level. For example, geometry may be defined for a distribution segment (e.g. ducts, pipes, cables) that indicate permitted use of an extruded area solid (IfcExtrudedAreaSolid) which defines a 2D cross section extruded along a 3D linear segment. Such rule may be further refined for ducts to indicate that the cross-sections are further restricted to shapes such as hollow rectangles (IfcRectangleHollowProfileDef) or hollow circles (IfcCircleHollowProfileDef).

- |                                 |                             |
|---------------------------------|-----------------------------|
| • Project                       | • Constraint                |
| • Project Declaration           | • External Data Constraints |
| • Project Units                 | • Assignment                |
| • Conversion Units              | • Control Assignment        |
| • Roots                         | • Group Assignment          |
| • Identity                      | • Composition               |
| • Revision Control              | • Aggregation               |
| • Definition                    | • Spatial Composition       |
| • Object Typing                 | • Connectivity              |
| • Property Sets                 | • Spatial Structure         |
| • Property Sets for Occurrences | • Spatial Containment       |
| • Property Sets for Types       | • Sequential Connectivity   |
| • Quantity Sets                 | • Actor                     |
| • Quantities on Occurrences     | • Contact                   |
| • Association                   | • Process                   |
| • Classification                | • Task Scheduling           |
| • Classification expected       |                             |

#### 4.2.7.8 Concept requirements applicability

The listing below allows software developers to more clearly understand the model concepts that are required to be applied, for a given entity, at each stage in the COBie deliverable process. In these tables the letter “R” indicates that the concept is required for a given exchange. The letter “O” means that the concept is required if (1) the information is required by contract or (2) the information is present in the COBie data model.

**Table 51 IfcSpace - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
External Data Constraints	O	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Classification	O	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Spatial Composition	O	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Quantities on Occurrences	O	O	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 52 IfcBuildingStory - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
External Data Constraints	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Spatial Composition	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Quantities on Occurrences	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	



**Table 53 IfcBuilding - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Classification	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Spatial Composition	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 54 IfcSite - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Spatial Composition	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

**Table 55 IfcProject - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
External Data Constraints	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Conversion Units	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Project Declaration	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 56 IfcZone - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
External Data Constraints	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Classification	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Group Assignment	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	

**Table 57 IfcSystem - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
External Data Constraints	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Group Assignment	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Classification	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

**Table 58 IfcElement - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Revision Control					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
External Data Constraints					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Classification					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Object Typing					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Spatial Containment					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Property Sets for Occurrences					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

**Table 59 IfcElementType - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity					O	O	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Revision Control					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
External Data Constraints					O	O	O	O	O	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Classification					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Property Sets for Types					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

**Table 60 IfcTask - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity												R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Revision Control												O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
External Data Constraints												R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Classification												O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Property Sets for Occurrences												O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Task Scheduling												O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Sequential Connectivity												O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	

**Table 61 IfcActor - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
External Data Constraints	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Contact	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Classification	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

**Table 62 IfcActionRequest - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity									R			R				R	R											
Revision Control									O			O				O	O											
External Data Constraints									R			R				R	R											
Property Sets for Occurrences									O			O				O	O											
Control Assignment									O			O				O	O											

**Table 63 IfcPropertySet - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
External Data Constraints	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Classification	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

**Table 64 IfcAnnotation - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity						O	O	O	O																			
Revision Control						O	O	O	O																			
External Data Constraints						O	O	O	O																			

**Table 65 IfcRelConnectsPorts - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity														O														
Revision Control														O														
External Data Constraints														O														

**Table 66 IfcWorkSchedule - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity																	O											
Revision Control																	O											
External Data Constraints																	O											

**Table 67 IfcRelAssociatesDocument - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	
External Data Constraints	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	

**Table 68 IfcRelConnectsWithRealizingElements - Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity								O							O													
Revision Control								O							O													
External Data Constraints								O							O													

**Table 69 ifcConstructionProductResource – Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity																				O	O	O	O	O	O	O	O	O
Revision Control																				O	O	O	O	O	O	O	O	O
External Data Constraints																				O	O	O	O	O	O	O	O	O
Property Setsfor Occurrences																				O	O	O	O	O	O	O	O	O

**Table 70 ifcConstructionProductResourceType – Concept Applicability**

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	Design Schematic	Design Coordinated	Design Issue	ProductType Template	Product Template	Bid Issue	ProductTypeSelection	System Layout	Product Installation	Product Inspection	Construction Issue	Product Type Parts	ProductTypeWarranty	ProductTypeMaintenance	System Operation	Space Condition	ProductPartsReplacement	Space Occupancy	SpaceActivityRenovation	Remodel	Expand	Demolish
Identity																		R										
Revision Control																		O										
External Data Constraints																		R										
Property Setsfor Types																		R										

#### 4.2.7.9 Related existing concept list

The table below demonstrates the re-use of COBie MVD concepts in related NBIMS-US™ V3 standards, and other buildingSMART alliance® projects. In the table, each row corresponds to a concept used within this model view, each column corresponds to another model view, and each cell indicates usage of the concept within the corresponding model view.

**Table 71 Exchange Requirement Reuse Analysis**

Exchange	COBie	BPie	BAMie	HVACie	SPARKie	WSie
Identity	X	X	X		X	X
Revision Control	X		X		X	X
External Data Constraints	X		X			
Classification	X	X	X		X	X
Spatial Composition		X	X			
Quantities on Occurrences			X			
Conversion Units	X		X		X	X
Project Declaration	X	X	X		X	X
Group Assignment	X		X		X	X
Object Typing	X	X	X	X	X	X
Spatial Containment	X		X		X	X
Property Sets for Occurrences	X	X	X		X	X
Property Sets for Types	X	X	X		X	X
Task Scheduling	X		X			
Sequential Connectivity			X			
Contact			X			
Control Assignment	X	X	X		X	X

#### 4.2.7.10 Concept business rule list

The tables below identify the application of each concept business rule to the specific IFC entity or parameters to which that rule apply. The values for applicable parameters are either “As-defined” meaning that the rules contained in the underlying IFC model apply or as noted in the table.

**Table 72 Identity Concept Business Rule List**

Reference	Parameter
\IfcRoot.GlobalId	As-specified
\IfcRoot.GlobalId\IfcGloballyUniqueId	As-specified
\IfcRoot.Name	As-specified
\IfcRoot.Name\IfcLabel	As-specified
\IfcRoot.Description	As-specified
\IfcRoot.Description\IfcText	As-specified



**Table 73 Revision Control Concept Business Rule List**

Reference	Parameter
\lfcRoot.OwnerHistory	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningUser	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningUser\lfcPersonAndOrganization	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningUser\lfcPersonAndOrganization.ThePerson	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningUser\lfcPersonAndOrganization.ThePerson\lfcPerson	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningUser\lfcPersonAndOrganization.TheOrganization	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningUser\lfcPersonAndOrganization.TheOrganization\lfcOrganization	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningApplication	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningApplication\lfcApplication	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningApplication\lfcApplication.ApplicationDeveloper	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.OwningApplication\lfcApplication.ApplicationDeveloper\lfcOrganization	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.State	As-specified

**Table 74 Revision Control Concept Business Rule List (cont.)**

Reference	Parameter
\lfcRoot.OwnerHistory\lfcOwnerHistory.State\lfcStateEnum	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.ChangeAction	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.ChangeAction\lfcChangeActionEnum	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.CreationDate	As-specified
\lfcRoot.OwnerHistory\lfcOwnerHistory.CreationDate\lfcTimeStamp	As-specified

**Table 75 External Data Constraints Concept Business Rule List**

Reference	Parameter
\lfcObjectDefinition.HasAssociations	As-specified
\lfcObjectDefinition.HasAssociations\lfcRelAssociatesConstraint	As-specified
\lfcObjectDefinition.HasAssociations\lfcRelAssociatesConstraint.RelatingConstraint	As-specified

Reference	Parameter
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue\IfcTable	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue\IfcTable.Name	Table
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue\IfcTable.Name\IfcLabel	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue\IfcTable.Columns	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues \IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Name	Name

**Table 76 External Data Constraints Concept Business Rule List (cont.)**

Reference	Parameter
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Name\IfcLabel	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.ReferencePath	Reference
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.ReferencePath\IfcReference	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Unit	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Unit\IfcContextDependentUnit	As-specified

\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Unit\IfcContextDependentUnit.Name	Format
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Description	Color
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.BenchmarkValues\IfcMetric.DataValue\IfcTable.Columns\IfcTableColumn.Description\IfcText	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.ObjectiveQualifier	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.ObjectiveQualifier\IfcObjectiveEnum	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.RelatingConstraint\IfcObjective.ObjectiveQualifier\IfcObjectiveEnum.Value=EXTERNAL	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.Name	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesConstraint.Name\IfcLabel	As-specified

**Table 77 Classification Concept Business Rule List**

Reference	Parameter
\IfcObjectDefinition.HasAssociations	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.Identification	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.Identification\IfcIdentifier	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.Source	Source
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.Source\IfcLabel	As-specified

\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferenceSource\IfcClassification.Name	Name
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferenceSource\IfcClassification.Name\IfcLabel	As-specified
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferenceSource\IfcClassification.ReferenceTokens	Tokens
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferenceSource\IfcClassification.ReferenceTokens\IfcIdentifier	As-specified

**Table 78 Spatial Composition Concept Business Rule List**

Reference	Parameter
\IfcSpatialElement.Decomposes	As-specified
\IfcSpatialElement.Decomposes\IfcRelAggregates	As-specified
\IfcSpatialElement.Decomposes\IfcRelAggregates.RelatingObject	Spatial Composite
\IfcSpatialElement.Decomposes\IfcRelAggregates.RelatingObject\IfcSpatialElement	As-specified

**Table 74 Quantities on Occurrences Concept Business Rule List**

Reference	Parameter
\IfcObject.IsDefinedBy	As-specified
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties	As-specified
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition	As-specified
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity	As-specified
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity.Name	Qset
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity.Name\IfcLabel	As-specified
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity.Quantities	As-specified
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity.Quantities\IfcPhysicalSimpleQuantity	Type
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity.Quantities\IfcPhysicalSimpleQuantity.Name	Quantity
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcElementQuantity.Quantities\IfcPhysicalSimpleQuantity.Name\IfcLabel	As-specified

**Table 75 Conversion Units Concept Business Rule List**

Reference	Parameter
\lfcContext.UnitsInContext	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.UnitType	UnitType
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.UnitType\lfcUnitEnum	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.Name	Name
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.Name\lfcLabel	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.ValueComponent	ConversionType
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.ValueComponent\lfcReal	ConversionFactor
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.UnitComponent	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.UnitComponent\lfcSIUnit	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.UnitComponent\lfcSIUnit.Prefix	BaseUnitPrefix
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.UnitComponent\lfcSIUnit.Prefix\lfcSIPrefix	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.UnitComponent\lfcSIUnit.Name	BaseUnitName
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionFactor\lfcMeasureWithUnit.UnitComponent\lfcSIUnit.Name\lfcSIUnitName	As-specified
\lfcContext.UnitsInContext\lfcUnitAssignment.Units\lfcConversionBasedUnitWithOffset.ConversionOffset	ConversionOffset

**Table 81 Project Declaration Concept Business Rule List**

Reference	Parameter
\lfcContext.Declares	As-specified
\lfcContext.Declares\lfcRelDeclares	As-specified
\lfcContext.Declares\lfcRelDeclares.RelatedDefinitions	Type
\lfcContext.Declares\lfcRelDeclares.RelatedDefinitions\lfcObjectDefinition	As-specified

**Table 762 Group Assignment Concept Business Rule List**

Reference	Parameter
\lfcGroup.IsGroupedBy	As-specified
\lfcGroup.IsGroupedBy\lfcRelAssignsToGroup	As-specified
\lfcGroup.IsGroupedBy\lfcRelAssignsToGroup.RelatedObjects	Type
\lfcGroup.IsGroupedBy\lfcRelAssignsToGroup.RelatedObjects\lfcProduct	As-specified

**Table 77 Object Typing Concept Business Rule List**

Reference	Parameter
\lfcObject.IsTypedBy	As-specified
\lfcObject.IsTypedBy\lfcRelDefinesByType	As-specified
\lfcObject.IsTypedBy\lfcRelDefinesByType.RelatingType	Type
\lfcObject.IsTypedBy\lfcRelDefinesByType.RelatingType\lfcTypeObject	As-specified

**Table 784 Spatial Containment Concept Business Rule List**

Reference	Parameter
\lfcElement.ContainedInStructure	As-specified
\lfcElement.ContainedInStructure\lfcRelContainedInSpatialStructure	As-specified
\lfcElement.ContainedInStructure\lfcRelContainedInSpatialStructure.RelatingStructure	Structure
\lfcElement.ContainedInStructure\lfcRelContainedInSpatialStructure.RelatingStructure\lfcSpatialStructureElement	As-specified

**Table 85 Property Sets for Occurrences Concept Business Rule List**

Reference	Parameter
\lfcObject.PredefinedType	PredefinedType
\lfcObject.IsDefinedBy	As-specified
\lfcObject.IsDefinedBy\lfcRelDefinesByProperties	As-specified
\lfcObject.IsDefinedBy\lfcRelDefinesByProperties.RelatingPropertyDefinition	As-specified
\lfcObject.IsDefinedBy\lfcRelDefinesByProperties.RelatingPropertyDefinition\lfcPropertySet	As-specified
\lfcObject.IsDefinedBy\lfcRelDefinesByProperties.RelatingPropertyDefinition\lfcPropertySet.Name	Name
\lfcObject.IsDefinedBy\lfcRelDefinesByProperties.RelatingPropertyDefinition\lfcPropertySet.HasProperties	As-specified
\lfcObject.IsDefinedBy\lfcRelDefinesByProperties.RelatingPropertyDefinition\lfcPropertySet.HasProperties\lfcProperty	As-specified

**Table 86 Property Sets for Types Concept Business Rule List**

Reference	Parameter
\lfcTypeObject.HasPropertySets	As-specified
\lfcTypeObject.HasPropertySets\lfcPropertySet	As-specified
\lfcTypeObject.HasPropertySets\lfcPropertySet.Name	Name
\lfcTypeObject.HasPropertySets\lfcPropertySet.Name\lfcLabel	As-specified
\lfcTypeObject.HasPropertySets\lfcPropertySet.HasProperties	As-specified
\lfcTypeObject.HasPropertySets\lfcPropertySet.HasProperties\lfcProperty	As-specified

**Table 797 Task Scheduling Concept Business Rule List**

Reference	Parameter
\lfcTask.TaskTime	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleStart	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleStart\lfcDateTime	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleFinish	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleFinish\lfcDateTime	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleDuration	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleDuration	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.ScheduleDuration\lfcDuration	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.RecurranceType	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.RecurranceType\lfcRecurrenceType Enum	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.DayComponent	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.DayComponent\lfcDayInMonthNumber	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.WeekdayComponent	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.WeekdayComponent\lfcDayInWeekNumber	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.MonthComponent	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.MonthComponent\lfcMonthInYearNumber	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.Position	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.Position\lfcInteger	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.Interval	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrance\lfcRecurrencePattern.Interval\lfcInteger	As-specified



**Table 88 Task Scheduling Concept Business Rule List (cont.)**

Reference	Parameter
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.Occurrences	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.Occurrences\lfcInteger	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.TimePeriods	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.TimePeriods\lfcTimePeriod	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.TimePeriods\lfcTimePeriod.StartTime	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.TimePeriods\lfcTimePeriod.StartTime\lfcTime	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.TimePeriods\lfcTimePeriod.EndTime	As-specified
\lfcTask.TaskTime\lfcTaskTimeRecurring.Recurrence\lfcRecurrencePattern.TimePeriods\lfcTimePeriod.EndTime\lfcTime	As-specified

**Table 89 Sequential Connectivity Concept Business Rule List**

Reference	Parameter
\lfcProcess.IsPredecessorTo	As-specified
\lfcProcess.IsPredecessorTo\lfcRelSequence	As-specified
\lfcProcess.IsPredecessorTo\lfcRelSequence.RelatedProcess	As-specified
\lfcProcess.IsPredecessorTo\lfcRelSequence.RelatedProcess\lfcProcess	As-specified
\lfcProcess.IsSuccessorFrom	As-specified
\lfcProcess.IsSuccessorFrom\lfcRelSequence	As-specified
\lfcProcess.IsSuccessorFrom\lfcRelSequence.RelatingProcess	As-specified
\lfcProcess.IsSuccessorFrom\lfcRelSequence.RelatingProcess\lfcProcess	As-specified

**Table 90 Contact Concept Business Rule List**

Reference	Parameter
\lfcActor.TheActor	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.AddressLines	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.AddressLines\lfcLabel	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.PostalBox	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.PostalBox\lfcLabel	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.Town	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.Town\lfcLabel	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.Region	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.Region\lfcLabel	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.PostalCode	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.PostalCode\lfcLabel	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.Country	As-specified
\lfcActor.TheActor\lfcPersonAndOrganization.ThePerson\lfcPerson.Addresses\lfcPostalAddress.Country\lfcLabel	As-specified

**Table 91 Contact Concept Business Rule List (cont.)**

Reference	Parameter
\IfcActor.TheActor\IfcPersonAndOrganization.ThePerson\IfcPerson.Addresses\IfcTelecomAddress	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.ThePerson\IfcPerson.Addresses\IfcTelecomAddress.TelephoneNumbers	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.ThePerson\IfcPerson.Addresses\IfcTelecomAddress.TelephoneNumbers\IfcLabel	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.TheOrganization	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.TheOrganization\IfcOrganization	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.TheOrganization\IfcOrganization.Name	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.TheOrganization\IfcOrganization.Name\IfcLabel	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.TheOrganization\IfcOrganization.Description	As-specified
\IfcActor.TheActor\IfcPersonAndOrganization.TheOrganization\IfcOrganization.Description\IfcText	As-specified

**Table 80 Control Assignment Concept Business Rule List**

Reference	Parameter
\IfcControl.Controls	As-specified
\IfcControl.Controls\IfcRelAssignsToControl	As-specified
\IfcControl.Controls\IfcRelAssignsToControl.RelatedObjects	Type
\IfcControl.Controls\IfcRelAssignsToControl.RelatedObjects\IfcObject	As-specified

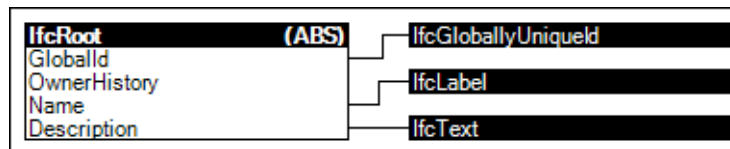
#### 4.2.7.11 Concept Business Rule Definition

##### 4.2.7.11.1 Identity Concept Business Rule

An object needs to be identifiable for accurate processing by both human and automated processes. Identification may be through several attributes such as Identification, Name, or GUID. The GUID is compressed for the purpose of being exchanged within an IFC data set - the compressed GUID is referred to as "IFC-GUID". While the IFC-GUID is normally generated automatically and has to be persistent, the Identification may relate to other informal registers but should be unique within the set of objects of the same type. The Name and Description should allow any object to be identified in the context of the project or facility being modelled.

Various objects may have additional identifications that may be human-readable and/or may be structured through classification association.

Various file formats may use additional identifications of instances for serialization purposes; however there is no requirement or guarantee for such identifications to remain the same between revisions or across applications. For example, the IFC-SPF file format lists each instance with a 64-bit integer that is unique within the particular file.



**Figure 62 Identity Concept - Business Rule**

For specific entities information the entity name **MUST** be unique. Other entities must be uniquely identified by a combination of Name and other required information. Constraints related to compound keys are not described in this paragraph.

#### **4.2.7.11.2 Revision Control Concept Business Rule**

While objects may reflect a final state, they may also be continually revised over the course of a project lifecycle and reflect transient state. For scenarios of multiple users making updates to the same information, there is a concept of local copies of information based upon a shared repository supporting multiple users. Such shared repository is often referred to as a model server. A model server is similar in concept to a document revision server, but is able to identify changes declared on a per-object basis rather than inferring changes from differences in text. A model server has a concept of revisions on a per-project basis, where each revision consists of a set of changes to contained objects by a particular user at a particular time.

To support a model server scenario, each object may be marked with a change action indicating the object was added, modified, deleted, or has no change since the project was retrieved from the server at a particular revision sequence. Given an object's identity (IFC-GUID) and change action, the state of the object may be merged when submitted to a model server. An object is considered modified when any of its direct attributes change, attributes on a referenced resource definition (any entity not deriving from IfcRoot) change, items are added or removed from sets, or items are added, removed, or reordered within lists.

For cases when multiple users make conflicting changes to the same objects, users may choose to keep their own changes, accept changes from others, merge both changes, and a combination thereof upon submitting to a server. Alternatively, to avoid such merge scenario and coordinate work, objects may be locked such that a particular user has exclusive access to read and/or write a particular object at the current time.

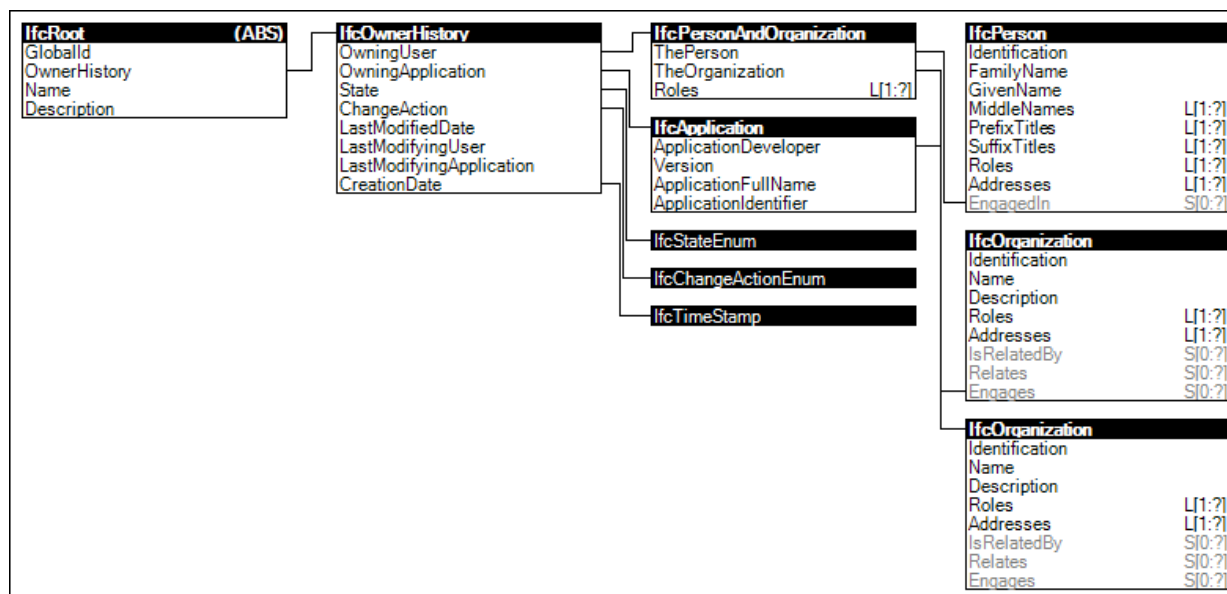


Figure 63 Revision Control Concept - Business Rule

Project libraries may also be retrieved from model servers having particular revision, and potentially different server URI than the referencing Project. As a project may include multiple revisions of the same project library (a common scenario when multiple users are involved using libraries revised by others), the IfcRoot.ObjectIdentifier IFC-GUID is only valid within the scope of the referencing project, and a separate library reference identifies a project library based object within its originating model server.

Finally, objects may also carry informational attributes indicating when an object was created, who, when, and what application was used to last modify an object, and who currently owns the object, potentially having exclusive use according to its lock state.

#### 4.2.7.11.3 External data constraints concept business rule

The constraint model may be used to indicate mappings between data in the IFC model and external documents. This concept template may also be used by software applications to translate data to/from spreadsheets without necessarily instantiating constraint relationships within an IFC data set.

To indicate an explicit mapping to a particular file or database, IFC classes may be mapped to tabular data formats using IfcResourceConstraintRelationship attached to IfcDocumentInformation.

Default mappings may be indicated using the IfcRelAssociatesConstraint relationship, with RelatingConstraint pointing to IfcObjective.

An IfcObjective of type EXTERNAL has ConstraintSource set to the name of the IFC entity (e.g. 'IfcSpace') with Benchmarks containing a single IfcMetric with DataValue set to an IfcTable. On IfcTable, the Name is set to the name of the external database table or worksheet, and the Columns attribute indicates the external table columns in order as IfcTableColumn. For each IfcTableColumn, the Name indicates the field name or column header, and the ReferencePath identifies the corresponding object attribute, for which standard mappings are indicated.

Within this document, attribute paths (as used for IfcReference) are encoded using syntax in the form 'IfcSpace.OwnerHistory\IfcOwnerHistory.CreationDate' with the following conventions:

- The period character dereferences an attribute from an entity.
- The backslash character casts an entity into a subtype, where a backslash without a subtype indicates mapping to the type itself.
- A bracket sequence with an asterisk ("[\*]") dereferences a collection into each member.
- A bracket sequence with an encoded string (e.g. "[SerialNumber]") dereferences a collection into a specific member by name.

Upon import from a spreadsheet, tables shall be identified by worksheet name regardless of order, and columns shall be identified by header name regardless of order. For export to a new spreadsheet, worksheets shall be provided with identifying name for each entity and sequenced in the order specified, and within each worksheet a header row shall be provided with each column having an identifying name and sequenced in the order specified.

Attributes are mapped to spreadsheet cells, where either NULL, an empty string, or the reserved value 'n/a' indicates a null value. Specific base types are mapped as follows:

- **STRING:** String types are represented as strings.
- **REAL:** Real number types are represented as real numbers.
- **ENTITY** Entity references are represented as strings identifying by name (IfcRoot.Name).
- **SET** Set-based collections are represented as strings identifying each object by name and separated by a comma.
- Specific mappings are defined at corresponding entities, where the following standard column names are used:
- **Name:** Indicates the name of the object, corresponding to IfcRoot.Name.
- **ExtObject:** Indicates the IFC type of the object by identifying the IFC entity by name (e.g. 'IfcBoiler'). If omitted upon import, the IFC entity shall be the base type where the concept is indicated (if non-abstract); otherwise no entity shall be imported.
- **ExtSystem:** Indicates the application that currently owns the object, mapping to IfcApplication.ApplicationIdentifier.
- **ExtIdentifier:** Indicates the GUID of the object encoded in the same format described at IfcGloballyUniqueId, mapping to IfcRoot.GlobalId. If omitted upon import, such GUID shall be constructed dynamically. If merging changes upon import, such identifier shall be used to identify an existing object (i.e. this maintains identity if an object is renamed (the Name attribute changing)).

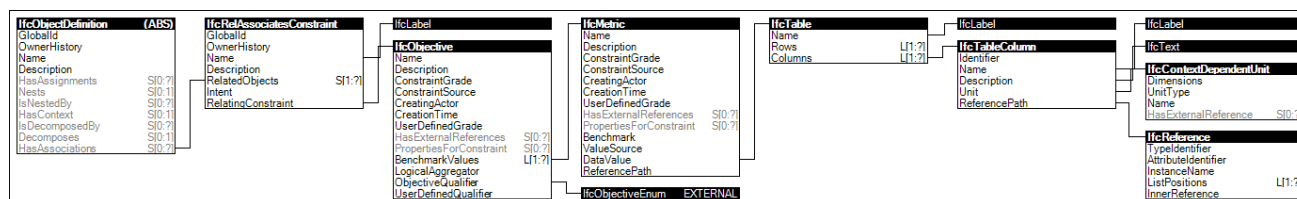


Figure 64 External Data Constraints Concept - Business Rule

#### 4.2.7.11.4 Classification concept business rule

Objects, type objects, properties, and some resource schema entities can be further described by associating references to external sources of information. The source of information can be:

- a classification system;
- a dictionary server;
- any external catalogue that classifies the object further;
- a service that combine the above features.

An individual item within the external source of information can be selected. It then applies the inherent meaning of the item to the object or property.

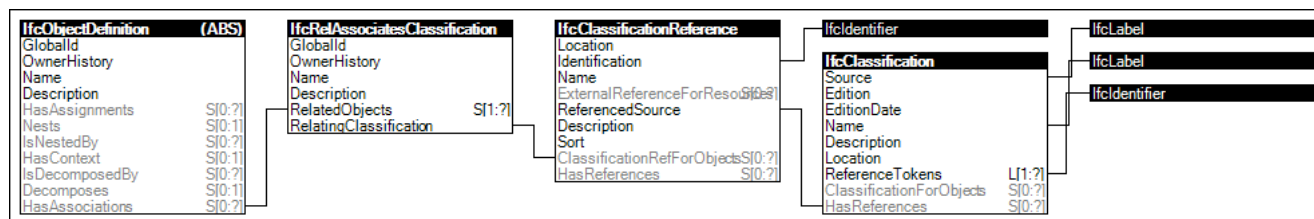


Figure 65 Classification Concept - Business Rule

#### 4.2.7.11.5 Spatial composition concept business rule

The spatial structure is a hierarchical tree of spatial structure elements (site, building, story, space) ultimately assigned to the project. Composition refers to the relationship to a higher level element (e.g. this story is part of a building). The order of spatial structure elements being included in the concept are from high to low level: IfcProject, IfcSite, IfcBuilding, IfcBuildingStory, IfcSpace. Therefore an spatial structure element can only be part of an element at the same or higher level.

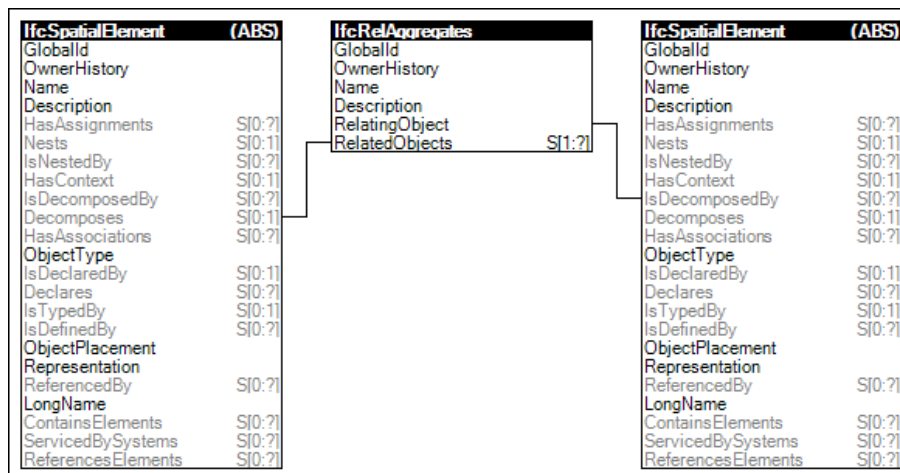


Figure 66 Spatial Composition Concept - Business Rule

#### 4.2.7.11.6 Quantities on occurrences concept business rule

Any specialization of object can be related to multiple quantity set occurrences. A quantity set contains multiple quantity occurrences. The data type of quantity occurrence are count, length, area, volume, weight, time, and combination of quantity occurrences.

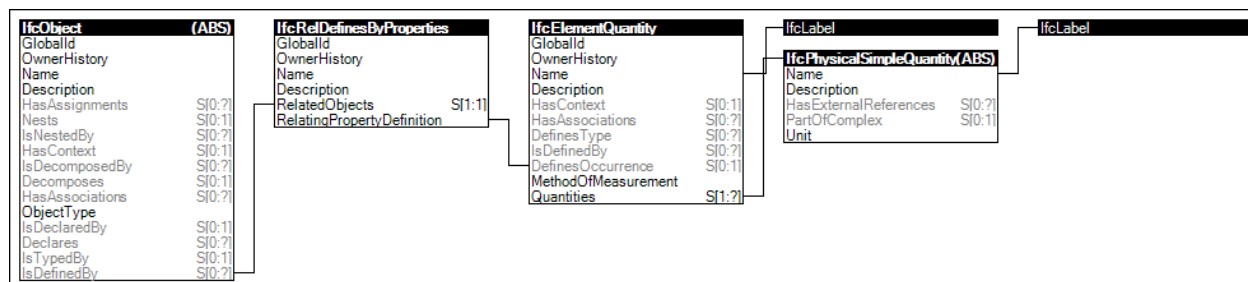


Figure 67 Quantities on Occurrences Concept - Business Rule

#### 4.2.7.11.7 Conversion units concept business rule

Conversion units are defined according to a conversion factor (and conversion offset for temperature) relative to a specified base SI unit.

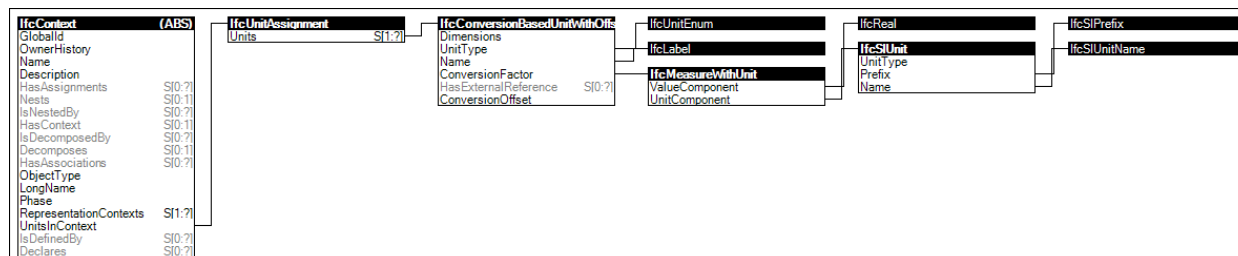


Figure 68 Conversion Units Concept - Business Rule

#### 4.2.7.11.8 Project declaration concept business rule

The project provides a directory of objects contained within using declaration relationships.

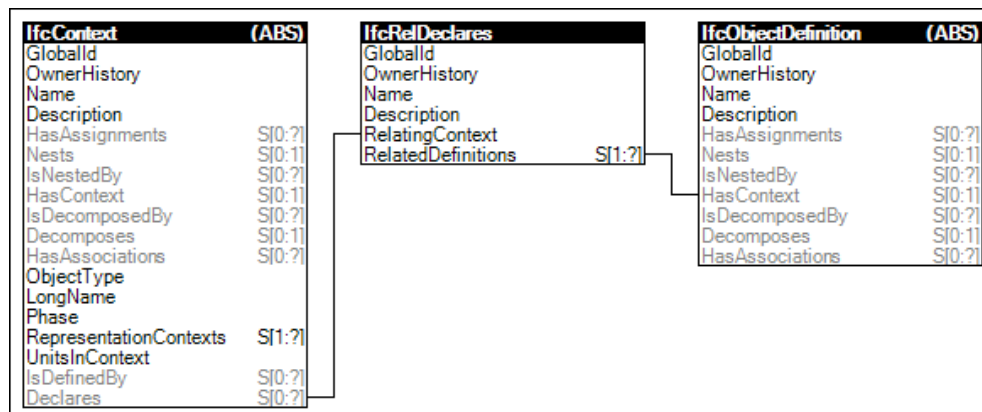


Figure 69 Project Declaration Concept - Business Rule



#### 4.2.7.11.9 Group assignment concept business rule

Groups may have assignments indicating products that are members of the group. An example of such assignment is an air handler belonging to an air conditioning system.

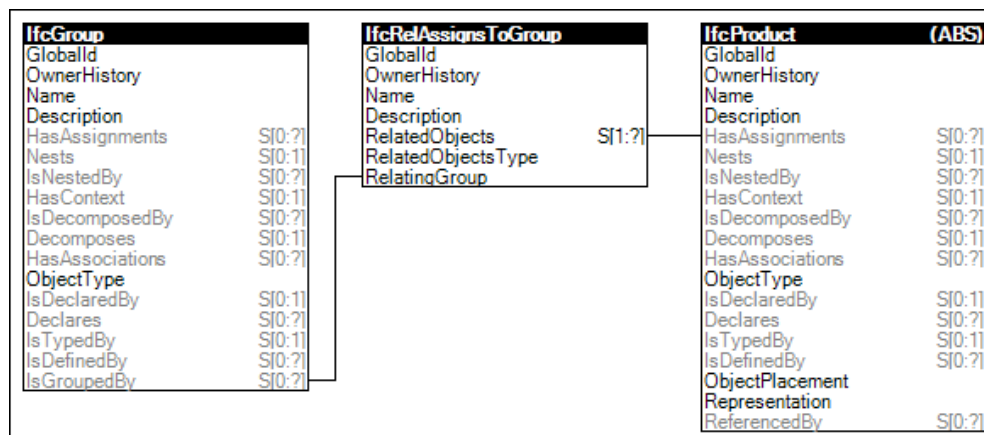


Figure 70 Group Assignment Concept - Business Rule

#### 4.2.7.11.10 Object typing concept business rule

Object Occurrences may be defined by a particular Object Type, where such type describes common characteristics. Such characteristics include common properties, shapes, materials, composition, and other concepts described at particular entities. An object occurrence may have similar state as its object type, overridden state for particular characteristics, or have no defined type object.

A pair of entities are defined for various object occurrences and object types, where such object occurrence entity may only be defined using a particular object type entity. For example, the IfcTank occurrence object entity has a corresponding IfcTankType type object entity.

Many object occurrence and object type entities have an attribute named PredefinedType consisting of a specific enumeration. Such predefined type essentially provides another level of inheritance to further differentiate objects without the need for additional entities. Predefined types are not just informational; various rules apply such as applicable property sets, part composition, and distribution ports.

For scenarios of object types having part compositions, such parts may be reflected at object occurrences having separate state. For example, a wall type may define a particular arrangement of studs, a wall occurrence may reflect the same arrangement of studs, and studs within the wall occurrence may participate in specific relationships that do not exist at the type such as being connected to an electrical junction box.

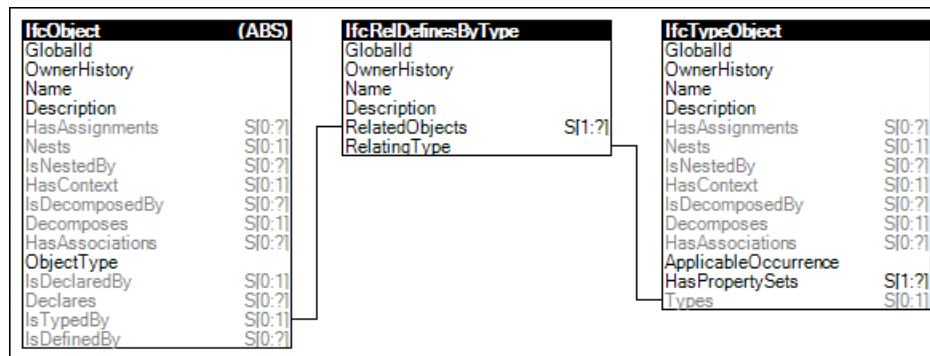


Figure 71 Object Typing Concept - Business Rule

#### 4.2.7.11.11 Spatial containment concept business rule

Spatial structures may contain physical elements, including building elements, distribution elements, and furnishing elements.

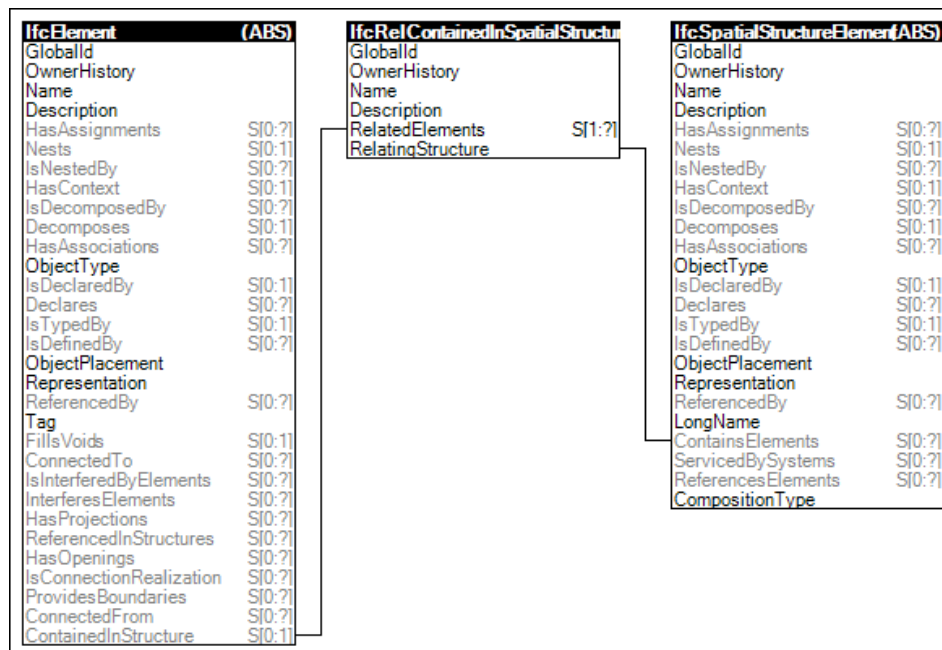


Figure 72 Spatial Containment Concept- Business Rule

#### 4.2.7.11.12 property sets for occurrences concept business rule

Any specialization of object can be related to multiple property set occurrences. A property set contains multiple property occurrences. The data types of property occurrences are single value, enumerated value, bounded value, table value, reference value, list value, and combination of property occurrences.

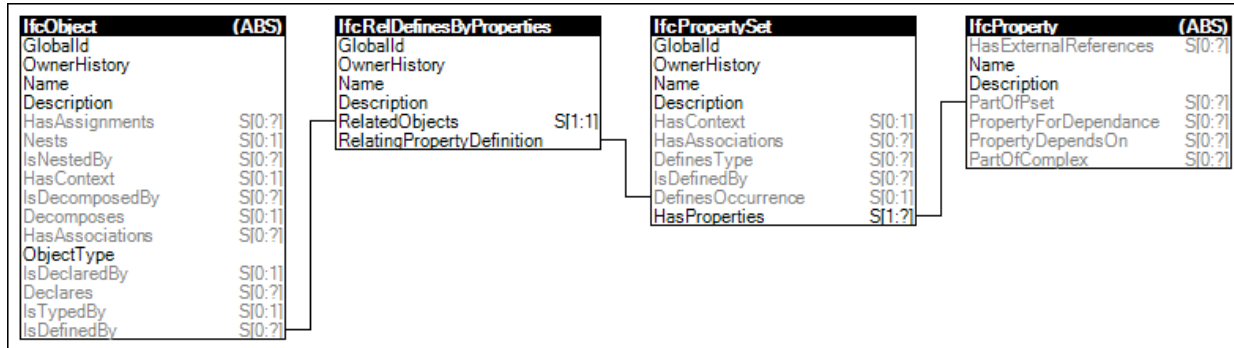


Figure 73 Property Sets for Occurrences Concept- Business Rule

#### 4.2.7.11.13 Property sets for types concept business rule

For object types, property sets are defined directly.

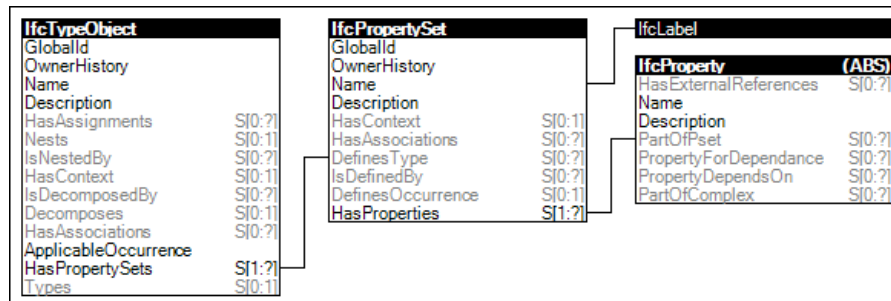


Figure 74 Property Sets for Types Concept - Business Rule

#### 4.2.7.11.14 Task scheduling concept business rule

Tasks may be scheduled to run continuously, at a single period in time, or multiple recurring periods in time.

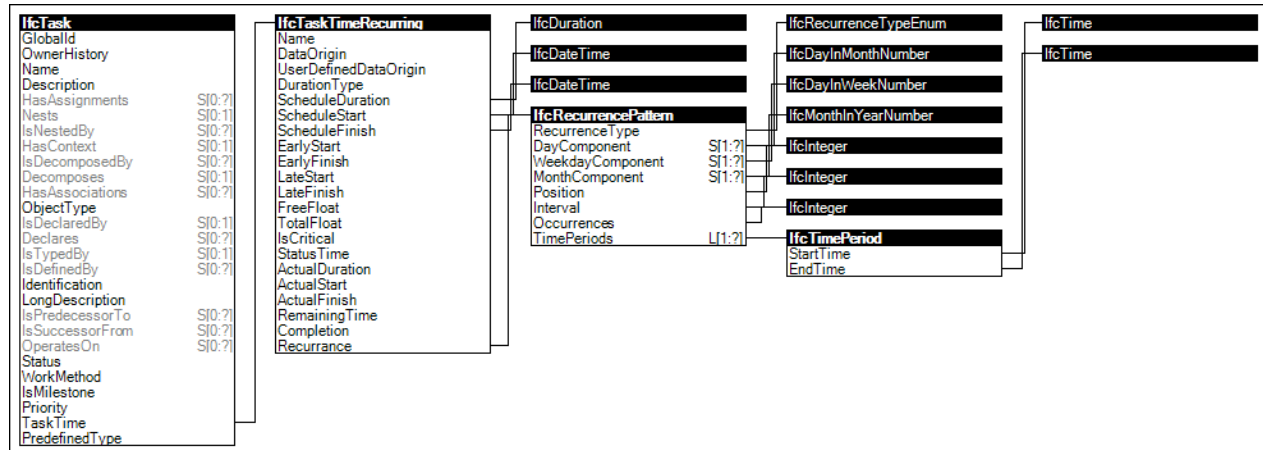


Figure 75 Task Scheduling Concept - Business Rule

#### 4.2.7.11.15 Sequential connectivity concept business rule

Processes that occur in time use this relationship to indicate the order of occurrence, such as for tasks, procedures, and events.

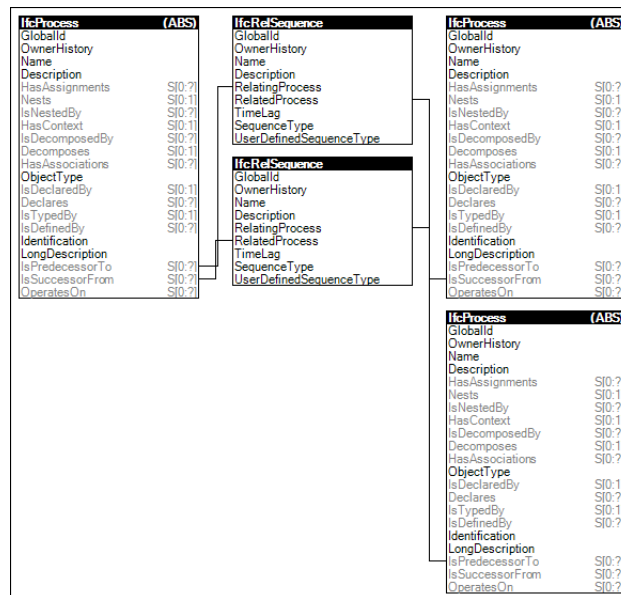


Figure 76 Sequential Connectivity Concept - Business Rule



- COBie.mvdxml - MVDXML schema transform
- COBie.ifc - IFC dynamic schema definition

An MVDXML file defines the referenced entities and rules for this model view. This file may be used to validate instance data (in IFC-SPF or IFC-XML files), filter instance data to include entities and attributes within scope of this model view, or generate sub-schemas (including the EXP and XSD representations).

An EXP file represents the schema in EXPRESS format (ISO 10303-11) which adapts the referenced Industry Foundation Classes schema (ISO 16739) by including a subset of data definitions and a subset of attributes within each data definition. The EXP file may be used by software development tools for generating programming languages schemas (e.g. C++, C#, Java), database definitions (e.g. SQL DDL), and data transport schema definitions (e.g. XSD).

An XSD file represents the schema in XML Data Definition Language (XSD) which adapts the referenced subset of data definitions. The XSD file may be used by software development tools (e.g. Eclipse, Microsoft Visual Studio) to validate XML files and generate language-specific classes.

An IFC file represents the dynamic portions of the schema in the form of property sets within an SPF (ISO 10303-21) instance file.

The rationale for publishing multiple representations is to provide the richest level of integration for different implementations; while XSD is often used in defining web standards replacing document-based exchanges (e.g. invoices), it lacks data model information needed for type safety, data integrity, indexing, and optimization; all of which may be derived from the EXPRESS representation.

#### 4.2.7.13 MVD format description

Each supported format is listed by name, with Extension indicating the default file extension to use on applicable platforms (e.g. Windows), MIME type for indicating the HTTP header when transmitting over the Internet, and Reference standard indicating the presentation layer encoding format. This standard contains all information necessary for transformation between these formats. These transformations have been demonstrated to fully maintain all information that is within the scope of this MVD

**Table 81 Allowable COBie MVD Formats**

Format	Extension	MIME	Reference
SpreadsheetML	.xml	application/xmlspreadsheet	Microsoft Office 2003
COBieLite	.xml	application/xml	U.S. Department of Homeland Security
IFC-SPF	.ifc	application/step	ISO 10303-21
IFC-XML	.ifcxml	application/xml	ISO 10303-28

SpreadsheetML is an XML format that presents COBie data in an easy to understand and digest format for members of the COBie user community who are not computer programmers.

COBieLite is a United States National Information Exchange Model (NIEM) compliant-XML schema. The use of this format is expected for system-to-system exchanges of COBie data in domains other than architectural and engineering design. This is because the exchanges needed by non-architectural and engineering design software need not carry the overhead and file size associated with STEP (or ifcXML) data formats.

IFC-SPF (ISO 10303-21) is a text format optimized to carry data with complex relationships, supporting human readability yet more compact representation (typically around 10% of size of equivalent XML).

IFC-HDF (ISO 10303-26) is a binary file format encapsulating data in a compact, indexable encoding optimized for quick retrieval and minimal memory usage. NOTE As this file type is not yet widely implemented, it is not officially part of this model view, however implementations may prefer such format for internal use.

IFC-XML (ISO 10303-28) is a hierarchical markup format with wide support from software development tools and platforms, supporting greater human readability at the expense of larger representation. NOTE As typical buildings contain millions of elements with graphs of relationships resulting in gigabytes of data, XML is not yet suitable for representing complete buildings from a pragmatic standpoint of data size, transmission cost, and loading time. However, using derived formats along with MVDXML to filter data sets may enable more efficient exchanges to take place.

IFC-ZIP (ISO 21320-1) is a compressed file format encapsulating one of the above formats to minimize data size. NOTE As this model view is primarily intended for web-based exchange, zip compression may be selected by other means according to the client and server; therefore, the IFC-ZIP format is not officially part of this model view.

#### 4.2.7.14 MVD dynamic schema analysis

Portions of data definitions are defined dynamically, to allow software applications to support extensible definitions while minimizing implementation overhead. Each property set is shown within a subsection as follows, with rows corresponding to properties. See IfcPropertySet for usage information.

**Table 82 COBie\_Task Dynamic Schema Definition**

Property	Property Type	Data Type
Category	P_ENUMERATEDV ALUE	/PEnum_TaskCategory:Adjustment,Calibration,Emergency,Inspection,Operation,PM,Safety,ShutDown,StartUp,Testing,Trouble
Status	P_ENUMERATEDV ALUE	/PEnum_TaskStatus:NotYet Started,Started,Completed

**Table 83 COBie\_ElementType Dynamic Schema Definition**

Property	Property Type	Data Type	Description
AssetType	P_ENUMERATEDVALUE	/PEnum_AssetType:Fixed,Moveable	A classification of the asset contained within one row. If allowable values are not specified by contract, the default values are: “Fixed” and “Movable”.
ReplacementCost	P_SINGLEVALUE	IfcMonetaryMeasure	During construction and handover phase: the manufacturer's suggested retail price for this type. During planning and design phase: not applicable.
NominalLength	P_SINGLEVALUE	IfcPositiveLengthMeasure	an approximate measure of the bounding box surrounding the product type.
NominalWidth	P_SINGLEVALUE	IfcPositiveLengthMeasure	an approximate measure of the bounding box surrounding the product type.
NominalHeight	P_SINGLEVALUE	IfcPositiveLengthMeasure	an approximate measure of the bounding box surrounding the product type.
Shape	P_SINGLEVALUE		During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
Size	P_SINGLEVALUE		During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
Color	P_SINGLEVALUE		During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
Finish	P_SINGLEVALUE		During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
Features	P_SINGLEVALUE	IfcText	During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
AccessibilityPerformance	P_SINGLEVALUE	IfcLabel	During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
CodePerformance	P_SINGLEVALUE	IfcLabel	During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
SustainabilityPerformance	P_SINGLEVALUE	IfcLabel	During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
Grade	P_SINGLEVALUE	IfcLabel	During planning and design phase: specific constraints that should be/have been considered during the specification process. During construction and handover phase: the text description of associated information found on a product manufacturers product data sheet.
Material	P_SINGLEVALUE		
Constituents	P_SINGLEVALUE		
	P_SINGLEVALUE	IfcText	



**Table 84 Pset\_ManufacturerOccurrence Dynamic Schema Definition**

Property	Property Type	Data Type	Description
AcquisitionDate	P_SINGLEVALUE	IfcDate	The date that the manufactured item was purchased.
BarCode	P_SINGLEVALUE	IfcIdentifier	The identity of the bar code given to an occurrence of the product.
SerialNumber	P_SINGLEVALUE	IfcIdentifier	The serial number assigned to an occurrence of a product.
BatchReference	P_SINGLEVALUE	IfcIdentifier	The identity of the batch reference from which an occurrence of a product is taken.
AssemblyPlace	P_ENUMERATEDVALUE	IfcLabel/PEnum_AssemblyPlace:FACTORY,OFFSITE,SITE,OTHER,NOTKNOWN,UNSET	Enumeration defining where the assembly is intended to take place, either in a factory, other offsite location or on the building site.

**Table 85 Pset\_ManufacturerTypeInfo Dynamic Schema Definition**

Property	Property Type	Data Type	Description
GlobalTradeItemNumber	P_SINGLEVALUE	IfcIdentifier	The Global Trade Item Number (GTIN) is an identifier for trade items developed by GS1 ( <a href="http://www.gs1.org">www.gs1.org</a> ).
ArticleNumber	P_SINGLEVALUE	IfcIdentifier	Article number or reference that is applied to a configured product according to a standard scheme for article number definition as defined by the manufacturer. It is often used as the purchasing number.
ModelReference	P_SINGLEVALUE	IfcLabel	The model number or designator of the product model (or product line) as assigned by the manufacturer of the manufactured item.
ModelLabel	P_SINGLEVALUE	IfcLabel	The descriptive model name of the product model (or product line) as assigned by the manufacturer of the manufactured item.
Manufacturer	P_SINGLEVALUE	IfcLabel	The organization that manufactured and/or assembled the item.
ProductionYear	P_SINGLEVALUE	IfcLabel	The year of production of the manufactured item.
AssemblyPlace	P_ENUMERATEDVALUE	IfcLabel/PEnum_AssemblyPlace:FACTORY,OFFSITE,SITE,OTHER,NOTKNOWN,UNSET	Enumeration defining where the assembly is intended to take place, either in a factory or on the building site.

**Table 86 Pset\_ActionRequest Dynamic Schema Definition**

Property	Property Type	Data Type	Description
RequestSourceLabel	P_SINGLEVALUE	IfcLabel	A specific name or label that further qualifies the identity of a request source. In the event of an email, this may be the email address.
RequestSourceName	P_REFERENCEVALUE	IfcPerson	The person making the request, where known.
RequestComments	P_SINGLEVALUE	IfcText	Comments that may be made on the request.

**Table 87 COBie\_ActionRequest Dynamic Schema Definition**

Property	Property Type	Data Type
Type	P_ENUMERATEDVALUE	/PEnum_ActionRequestType:Change,Claim,Coordination,Environmental,Function,IndoorAirQuality,Installation,RFI,Safety,Specification
Risk	P_ENUMERATEDVALUE	/PEnum_ActionRequestRisk:Very High,High,Moderate,Low,Unknown
Chance	P_ENUMERATEDVALUE	/PEnum_ActionRequestChance:Has Occurred,High,Moderate,Low,Unknown
Impact	P_ENUMERATEDVALUE	/PEnum_ActionRequestImpact:Very High,High,Moderate,Low,Unknown

**Table 100 COBie\_ConstructionProductResourceType Dynamic Schema Definition**

Property	Property Type	Data Type
Category	P_ENUMERATEDVALUE	/PEnum_ConstructionProductResourceTypeCategory:Part,PartSet,Lubricant,Other,Spare,SpareSet
SetNumber	P_SINGLEVALUE	
PartNumber	P_SINGLEVALUE	

**Table 88 COBie\_ConstructionProductResource Dynamic Schema Definition**

Property	Property Type	Data Type
Category	P_ENUMERATEDVALUE	/PEnum_ConstructionProductResourceCategory:Labor,Material,Tools,Training

#### 4.2.7.15 Entity exclusion set

As the number of established MVD's grows, software companies will be required to implement exports to each of these MVD's. Furthermore, the information content of such MVD's within a specific contractual context will also require software companies to filter the information exported within that MVD. To date, software companies who have provided COBie data in IFC format have done so by extending the Coordination MVD specification. As a result the information content, and size, of such files is significantly greater than that required for COBie.

To assist software companies to respond to the NBIMS-US™ COBie MVD, the tables below list IFC entities to be filtered when producing the generic COBie data sets for use in the United States. The exclusion lists are expressed as entities excluded from the COBie spreadsheet implementation format.

The Appendix to this standard provides an alternative view of the Entity Exclusion lists, showing the IFC inheritance graphs. These graphs may also be useful to fully understand the context of these exclusion lists.

Within the United States, specific owners may also extend the specification of COBie to non-managed facility assets. As a result software vendors should consider implementation of flexible exclusion lists that support both the generic standard and contract-specific exclusion lists. In addition, as COBie is internationally adopted standard software vendors may also wish to implement exclusion lists based on regional requirements that are outside of the scope of NBIMS-US™.

**Table 101 Entities Excluded from Type and Component**

IfcAnnotation	IfcJunctionBox	IfcFlowSegment
IfcBeam	IfcJunctionBoxType	IfcFlowSegmentType
IfcBeamStandardCase	IfcMechanicalFastener	IfcSpatialZone
IfcBeamType	IfcMember	IfcStair
IfcBuilding	IfcMemberStandardCase	IfcStairFlight
IfcBuildingElementPart	IfcMemberType	IfcStairFlightType
IfcBuildingStory	IfcOpeningElement	IfcStairType
IfcCableCarrierFitting	IfcOpeningStandardCase	IfcStructuralAction
IfcCableCarrierFittingType	IfcPile	IfcStructuralActivity
IfcCableCarrierSegment	IfcPipeFitting	IfcStructuralConnection
IfcCableCarrierSegmentType	IfcPipeFittingType	IfcStructuralCurveAction
IfcCableFitting	IfcPipeSegment	IfcStructuralCurveConnection
IfcCableFittingType	IfcPipeSegmentType	IfcStructuralCurveMember
IfcCableSegment	IfcPlate	IfcStructuralCurveMemberVarying
IfcCableSegmentType	IfcPlateStandardCase	IfcStructuralCurveReaction
IfcColumn	IfcPlateType	IfcStructuralItem
IfcColumnStandardCase	IfcProcedureType	IfcStructuralLinearAction
IfcColumnType	IfcProjectionElement	IfcStructuralMember
IfcCurtainWall	IfcRailing	IfcStructuralPlanarAction
IfcCurtainWallType	IfcRailingType	IfcStructuralPointAction
IfcDistributionPort	IfcRamp	IfcStructuralPointConnection
IfcDuctFitting	IfcRampFlight	IfcStructuralPointReaction
IfcDuctFittingType	IfcRampFlightType	IfcStructuralReaction
IfcDuctSegment	IfcRampType	IfcStructuralSurfaceAction
IfcDuctSegmentType	IfcReinforcingBar	IfcStructuralSurfaceConnection
IfcElementAssembly	IfcReinforcingMesh	IfcStructuralSurfaceMember
IfcElementType	IfcRoof	IfcStructuralSurfaceMemberVarying
IfcEventType	IfcSite	IfcStructuralSurfaceReaction
IfcExternalSpatialElement	IfcSlab	IfcSurfaceFeature
IfcExternalSpatialStructureElement	IfcSlabElementedCase	IfcTaskType
IfcFastener	IfcSlabStandardCase	IfcTypeObject
IfcFastenerType	IfcSlabType	IfcTypeProcess
IfcFeatureElement	IfcSpace	IfcTypeProduct
IfcFeatureElementAddition	IfcSpaceType	IfcVirtualElement
IfcFeatureElementSubtraction	IfcSpatialElement	IfcVoidingFeature
IfcFlowFitting	IfcSpatialElementType	IfcWall
IfcFlowFittingType	IfcSpatialStructureElement	IfcWallElementedCase
IfcFooting	IfcSpatialStructureElementType	IfcWallStandardCase
		IfcWallType

**Table 102 Entities Excluded From All Sheets**

IfcAnnotation	IfcJunctionBox	IfcStair
IfcBeam	IfcJunctionBoxType	IfcStairFlight
IfcBeamStandardCase	IfcMechanicalFastener	IfcStairFlightType
IfcBeamType	IfcMember	IfcStairType
IfcBuildingElementPart	IfcMemberStandardCase	IfcStructuralAction
IfcCableCarrierFitting	IfcMemberType	IfcStructuralActivity
IfcCableCarrierFittingType	IfcOpeningElement	IfcStructuralConnection
IfcCableCarrierSegment	IfcOpeningStandardCase	IfcStructuralCurveAction
IfcCableCarrierSegmentType	IfcPile	IfcStructuralCurveConnection
IfcCableFitting	IfcPipeFitting	IfcStructuralCurveMember
IfcCableFittingType	IfcPipeFittingType	IfcStructuralCurveMemberVarying
IfcCableSegment	IfcPipeSegment	IfcStructuralCurveReaction
IfcCableSegmentType	IfcPipeSegmentType	IfcStructuralItem
IfcColumn	IfcPlate	IfcStructuralLinearAction
IfcColumnStandardCase	IfcPlateStandardCase	IfcStructuralMember
IfcColumnType	IfcPlateType	IfcStructuralPlanarAction
IfcCurtainWall	IfcProcedureType	IfcStructuralPointAction
IfcCurtainWallType	IfcProjectionElement	IfcStructuralPointConnection
IfcDistributionPort	IfcRailing	IfcStructuralPointReaction
IfcDuctFitting	IfcRailingType	IfcStructuralReaction
IfcDuctFittingType	IfcRamp	IfcStructuralSurfaceAction
IfcDuctSegment	IfcRampFlight	IfcStructuralSurfaceConnection
IfcDuctSegmentType	IfcRampFlightType	IfcStructuralSurfaceMember
IfcElementAssembly	IfcRampType	IfcStructuralSurfaceMemberVarying
IfcElementType	IfcReinforcingBar	IfcStructuralSurfaceReaction
IfcEventType	IfcReinforcingMesh	IfcSurfaceFeature
IfcExternalSpatialElement	IfcRoof	IfcTaskType
IfcExternalSpatialStructureElement	IfcSite	IfcTypeObject
IfcFastener	IfcSlab	IfcTypeProcess
IfcFastenerType	IfcSlabElementedCase	IfcTypeProduct
IfcFeatureElement	IfcSlabStandardCase	IfcVirtualElement
IfcFeatureElementAddition	IfcSlabType	IfcVoidingFeature
IfcFeatureElementSubtraction	IfcSpaceType	IfcWall
IfcFlowFitting	IfcSpatialElement	IfcWallElementedCase
IfcFlowFittingType	IfcSpatialElementType	IfcWallStandardCase
IfcFlowSegment	IfcSpatialStructureElement	IfcWallType
IfcFlowSegmentType	IfcSpatialStructureElementType	
IfcFooting	IfcSpatialZone	

## 4.2.8 Conformance testing procedures

*Author's Editorial Comment: The NBIMS-US™ V2 COBie standard Part D provided general information pertaining to COBie conformance testing conducted in 2007-2008. This standard updates that information based on the experience of over a dozen public events including thirty software vendors. The conclusion reached during these sessions is that for NBIMS-US™ standard to be used in contracts, testing must evaluate the format, content, and associated workflows. As a result, buildingSMART alliance® testing is conducted as a “Challenge” and not a “Certification” event. Two types of testing have evolved as part of the buildingSMART alliance® Challenge. The “Quality Control Test” is used on COBie files produced at specific phases of the project life-cycle. The “Quality Assurance Test” is applied for software consuming COBie data files.*

buildingSMART alliance® Quality Control Testing begins with “verification” testing. Automated verification testing demonstrates that the data file provided is technically correct. Invalid files are not permitted to participate in the second stage testing. The second stage testing is a “validation” test to see if the information provided in the file could be used, ultimately, to replace the equivalent document-based contract deliverable. The criteria used to validate the information provided is to determine if the information provided would be acceptable in-lieu of the corresponding contract deliverable. An objective measure of time to manually correct data files is the objective metric that can be reported based on Quality Control Testing.

buildingSMART alliance® Quality Assurance Testing begins with the verification of the import of COBie data into a given software system. Validation of that information is evaluated, in the context of the workflows of a given software system, to determine if the information is consistently applied in all appropriate locations within that specific software. The output of Quality Assurance Testing is the identification of the information that was correctly imported into the tested software, and what information was not imported correctly.

The buildingSMART alliance® cannot predict the ultimate impact of its standards across a United States construction industry that comprises a notable percent of the Gross Domestic Product of the country.

The objective of delivering accurate and timely COBie data at the least cost on the widest number of projects is the goal of the COBie project. While buildingSMART alliance standards must be rooted in IFC to ensure that all MVD's have a common ancestry and unambiguous mapping, the ultimate implementation format for COBie data, across all use cases, may not be IFC. To this end any approved COBie mapping: IFC, ifcXML, SpreadsheetML, and COBieLite are allowed to be submitted as part of COBie testing. The selection of the implementation format is left to the software vendor.

To insure impartiality in the testing test models and expected output are publically available (East 2012a). The Medical Clinic Model at the Architectural Design and Construction Documents stage has been used as the basis for the Challenge. Specific spot checks of model content are identified as part of the checking process and then verified during the Challenge process.

Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange standards, the information provided in this section should be considered a Moderate Change.]

### 4.2.8.1 Format and content requirements

#### 4.2.8.1.1 Quality control test rule list

For software that produces COBie files, an automated Quality Control test procedure has been adopted and used in each of the Challenge events to evaluate the format and content of that COBie file. The rules are used to evaluate the integrity of the format and cardinality of a COBie test file.



- NotNull
- NotEmpty
- Unique
- CrossReference
- OneAndOnlyOneFacilityFound
- ValidNumber
- ValidNumberOrNA
- AtLeastOneRowPresent
- ZeroOrGreaterOrNA

In accordance with the COBie Guide publication, the criteria for evaluation of the content of a COBie deliverable is that the information provided in a COBie file match the information that would be found in the equivalent drawing or document set.

#### 4.2.8.1.2 Quality control test rule definition

The following general rules are applied for Quality Control testing of submitted COBie files:

**NotNull** - Must have a text value that is not n/a or empty

**NotEmpty** - Must have a text value (n/a is acceptable)

**Unique** - Must be unique within the scope of the worksheet

**CrossReference** - Must match a referenced Name column in another worksheet

**OneAndOnlyOneFacilityFound** - Only one facility is allowed

**ValidNumber** - Must represent a valid number - n/a is not acceptable

**ValidNumberOrNA** - If text is provided it must represent valid number or 'n/a'

**AtLeastOneRowPresent** - Worksheet must have at least one row

**ZeroOrGreaterOrNA** - If text is provided it must represent a valid number greater than zero or n/a

The following sections describe the full application of these rules to a given COBie file that is tested. This description is organized in a specific format to assist the reader of this standard to understand and evaluate these rules. First the rules are based on the COBie SpreadsheetML representation of COBie data. The first of the rules in each set applies to number of rows in a given COBie entity. Following rules in each set apply to specific fields within that worksheet. These tests are applied regardless of the format of the COBie test file tested (IFC, ifcXML, SpreadsheetML, COBieLite).

#### 4.2.8.1.3 Quality control test rule application

Rules applied to COBie.Contacts:

- Contact.AtLeastOneRowPresent
- Contact.CreatedBy.CrossReference
- Contact.Email.Unique
- Contact.CreatedBy.NotNull
- Contact.Email.NotNull
- Contact.CreatedOn.NotNull
- Contact.Email.Format
- Contact.ExternalSystem.NotEmpty

- Contact.ExternalObject.NotEmpty
- Contact.ExternalIdentifier.NotEmpty
- Contact.Category.NotNull
- Contact.Company.NotNull
- Contact.Phone.NotNull
- Contact.Department.NotEmpty
- Contact.OrganizationCode.NotEmpty
- Contact.GivenName.NotEmpty
- Contact.FamilyName.NotEmpty
- Contact.Street.NotEmpty
- Contact.PostalBox.NotEmpty
- Contact.Town.NotEmpty
- Contact.StateRegion.NotEmpty
- Contact.PostalCode.NotEmpty
- Contact.Country.NotEmpty

Rules applied to COBie.Facility:

- Facility.OneAndOnlyOneFacilityFound
- Facility.Name.NotNull
- Facility.Name.Unique
- Facility.CreatedBy.CrossReference (ToContact)
- Facility.CreatedBy.NotNull
- Facility.CreatedOn.NotNull
- Facility.CreatedOn.Valid (Valid Email Address)
- Facility.Category.NotNull
- Facility.Description.NotEmpty
- Facility.ProjectName.NotNull
- Facility.SiteName.NotNull
- Facility.LinearUnits.NotNull
- Facility.AreaUnits.NotNull
- Facility.VolumeUnits.NotNull
- Facility.Currency.NotNull
- Facility.AreaMeasurement.NotNull
- Facility.ExternalSystem.NotEmpty
- Facility.ExternalProjectObject.NotEmpty
- Facility.ExternalProjectIdentifier.NotEmpty
- Facility.ExternalSiteObject.NotEmpty
- Facility.ExternalSiteIdentifier.NotEmpty
- Facility.ExternalFacilityObject.NotEmpty
- Facility.ExternalFacilityIdentifier.NotEmpty

Rules applied to COBie.Floor:

- Floor.AtLeastOneRowPresent
- Floor.Name.NotNull
- Floor.Name.Unique
- Floor.CreatedBy.CrossReference (ToContact)
- Floor.CreatedBy.NotNull
- Floor.CreatedOn.Valid (Valid Email Address)
- Floor.CreatedOn.NotNull
- Floor.ExternalSystem.NotEmpty
- Floor.ExternalObject.NotEmpty
- Floor.ExtIdentifier.NotEmpty
- Floor.Category.NotNull

- Floor.Elevation.ValidNumberOrNA

Floor.Height.ZeroOrGreaterOrNARules applied to COBie.Space:

- Space.AtLeastOneRowPresent
- Space.CreatedBy.CrossReference (ToContact)
- Space.CreatedBy.NotNull
- Space.CreatedOn.NotNull
- Space.CreatedOn.Valid (Valid Email Address)
- Space.ExternalSystem.NotEmpty
- Space.ExternalObject.NotEmpty
- Space.ExtIdentifier.NotEmpty
- Space.Category.NotNull
- Space.Name.NotNull
- Space.PrimaryKey.Unique.Warning (Name)
- Space.PrimaryKey.Unique.Error (Name, FloorName)
- Space.FloorName.NotNull, FloorName.CrossReference
- Space.Description.NotNull
- Space.RoomTag.NotEmpty
- Space.UsableHeight.ZeroOrGreaterOrNA
- Space.GrossArea.ZeroOrGreaterOrNA

Space.NetArea.ZeroOrGreaterOrNARules applied to COBie.Zone:

- Zone.CreatedBy.CrossReference (ToContact)
- Zone.CreatedBy.NotNull
- Zone.CreatedOn.NotNull
- Zone.CreatedOn.Valid (Valid Email Address)
- Zone.Category.NotNull
- Zone.Description.NotEmpty
- Zone.ExternalSystem.NotEmpty
- Zone.ExternalObject.NotEmpty
- Zone.ExtIdentifier.NotEmpty
- Zone.Name.NotNull
- Zone.PrimaryKey.Unique (Name, Category, SpaceNames)
- Zone.SpaceNames.NotNull, SpaceNames.CrossReference



## Rules applied to COBie.Type:

- Type.AtLeastOneRowPresent
- Type.Name.NotNull
- Type.Name.Unique
- Type.CreatedBy.CrossReference (ToContact)
- Type.CreatedBy.NotNull
- Type.CreatedOn.NotNull
- Type.CreatedOn.Valid (Valid Email Address)
- Type.Category.NotNull
- Type.ExternalSystem.NotEmpty
- Type.ExternalObject.NotEmpty
- Type.ExtIdentifier.NotEmpty
- Type.Type.Component.AComponentForEachType
- Type.AssetType.NotNull
- Type.Manufacturer.NotNull
- Type.Manufacturer.CrossReference (Contact Sheet)
- Type.ModelNumber.NotNull
- Type.WarrantyGuarantorParts.NotNull
- Type.WarrantyGuarantorParts.CrossReference (Contact Sheet)
- Type.WarrantyDurationParts.validNumberZeroOrGreaterOrNA
- Type.WarrantyGuarantorLabor.NotNull
- Type.WarrantyGuarantorLabor.CrossReference (Contact Sheet)
- Type.WarrantyDurationLabor.ZeroOrGreaterOrNA
- Type.WarrantyDurationUnit.NotNull
- Type.ReplacementCost.ZeroOrGreaterOrNA
- Type.ExpectedLife.ZeroOrGreaterOrNA
- Type.DurationUnit.NotNull
- Type.WarrantyDescription.NotEmpty
- Type.NominalLength.ZeroOrGreaterOrNA
- Type.NominalWidth.ZeroOrGreaterOrNA
- Type.NominalHeight.ZeroOrGreater
- Type.ModelReference.NotEmpty
- Type.Shape.NotEmpty
- Type.Size.NotEmpty
- Type.Color.NotEmpty
- Type.Finish.NotEmpty
- Type.Grade.NotEmpty
- Type.Material.NotEmpty
- Type.Constituents.NotEmpty
- Type.Features.NotEmpty
- Type.AccessibilityPerformance.NotEmpty
- Type.CodePerformance.NotEmpty
- Type.SustainabilityPerformance.NotEmpty

## Rules applied to COBie.Component:

- Component.AtLeastOneRowPresent
- Component.CreatedBy.CrossReference (ToContact)
- Component.CreatedBy.NotNull
- Component.CreatedOn.NotNull
- Component.CreatedOn.Valid (Valid Email Address)
- Component.ExternalSystem.NotEmpty
- Component.ExternalObject.NotEmpty
- Component.ExtIdentifier.NotEmpty
- Component.Name.NotNull
- Component.PrimaryKey.Unique.Warning (Name)
- Component.PrimaryKey.Unique.Error (Name, Space)
- Component.TypeName.NotNull

- Component.TypeName.CrossReference (Type Worksheet)
- Component.Space.NotNull
- Component.Space.CrossReference (Component Worksheet)
- Component.Description.NotNull
- Component.AssetIdIdentifier.NotEmpty
- Component.SerialNumber.NotNull
- Component.InstallationDate.NotNull
- Component.WarrantyStartDate.NotNull
- Component.TagNumber.NotEmpty
- Component.BarCode.NotEmpty

Rules applied to COBie.System:

- System.CreatedBy.CrossReference (ToContact)
- System.CreatedBy.NotNull
- System.CreatedOn.NotNull
- System.CreatedOn.Valid (Valid Email Address)
- System.Category.NotNull
- System.ExternalSystem.NotEmpty
- System.ExternalObject.NotEmpty
- System.ExtIdentifier.NotEmpty
- System.Description.NotEmpty
- System.PrimaryKey.Unique (Name, Category, ComponentNames)
- System.Name.NotNull
- System.ComponentNames.NotNull
- System.ComponentNames.CrossReference

Rules available for COBie.Assembly, if information is present:

- Assembly.CreatedBy.CrossReference (ToContact)
- Assembly.CreatedBy.NotNull
- Assembly.CreatedOn.NotNull
- Assembly.CreatedOn.Valid (Valid Email Address)
- Assembly.ExternalSystem.NotEmpty
- Assembly.ExternalObject.NotEmpty
- Assembly.ExtIdentifier.NotEmpty
- Assembly.Description.NotEmpty
- Assembly.PrimaryKey.Unique
- Assembly.Name.NotNull
- Assembly.SheetName.NotNull
- Assembly.SheetName.CrossReference
- Assembly.ParentName.NotNull
- Assembly.ParentName.Reference
- Assembly.ChildNames.NotNull
- Assembly.ChildNames.CrossReference
- Assembly.AssemblyType.NotNull

Rules available for COBie.Connection, if information is present:



# National BIM Standard - United States®

an initiative of the National Institute of Building Sciences buildingSMARTalliance®

- Connection.CreatedBy.CrossReference (ToContact)
- Connection.CreatedBy.NotNull
- Connection.CreatedOn.NotNull
- Connection.CreatedOn.Valid (Valid Email Address)
- Connection.ExternalSystem.NotEmpty
- Connection.ExternalObject.NotEmpty
- Connection.ExtIdentifier.NotEmpty
- Connection.Description.NotEmpty
- Connection.PrimaryKey.Unique
- Connection.Name.NotNull
- Connection.ConnectionType.NotNull
- Connection.SheetName.NotNull
- Connection.SheetName.CrossReference
- Connection.RowName1.NotNull
- Connection.RowName1.CrossReference
- Connection.RowName2.NotNull
- Connection.RowName2.CrossReference
- Connection.RealizingElement.NotNull
- Connection.PortName1.NotNull
- Connection.PortName2.NotNull

Rules applied to COBie.Spare:

- Spare.CreatedBy.CrossReference (ToContact)
- Spare.CreatedBy.NotNull
- Spare.CreatedOn.NotNull
- Spare.CreatedOn.Valid (Valid Email Address)
- Spare.ExternalSystem.NotEmpty
- Spare.ExternalObject.NotEmpty
- Spare.ExtIdentifier.NotEmpty
- Spare.Description.NotEmpty
- Spare.Name.NotNull
- Spare.Name.Unique
- Spare.Category.NotNull
- Spare.TypeName.NotNull
- Spare.TypeName.CrossReference
- Spare.Suppliers.NotNull
- Spare.Suppliers.CrossReference (Contact Sheet)
- Spare.SetNumber.NotEmpty
- Spare.PartNumber.NotEmpty

Rules applied to COBie.Resource:

- Resource.Name.NotNull
- Resource.Name.Unique
- Resource.CreatedBy.CrossReference (ToContact)
- Resource.CreatedBy.NotNull
- Resource.CreatedOn.NotNull
- Resource.CreatedOn.Valid (Valid Email Address)
- Resource.ExternalSystem.NotEmpty
- Resource.ExternalObject.NotEmpty
- Resource.ExtIdentifier.NotEmpty
- Resource.Description.NotEmpty

## Resource.Category.NotNullRules applied to COBie.Job:

- Job.CreatedBy.CrossReference (ToContact)
- Job.CreatedBy.NotNull
- Job.CreatedOn.NotNull
- Job.CreatedOn.Valid (Valid Email Address)
- Job.ExternalSystem.NotEmpty
- Job.ExternalObject.NotEmpty
- Job.ExtIdentifier.NotEmpty
- Job.Description.NotEmpty
- Job.Category.NotNull
- Job.Name.NotNull
- Job.PrimaryKey.Unique (Name, TypeName, TaskNumber)
- Job.Status.NotNull
- Job.TypeName.NotNull
- Job.TypeName.CrossReference
- Job.Duration.NotEmpty
- Job.DurationUnit.NotEmpty
- Job.Start.NotEmpty
- Job.TaskStartUnit.NotEmpty
- Job.Frequency.NotEmpty
- Job.FrequencyUnit.NotEmpty
- Job.TaskNumber.NotEmpty
- Job.Priors.NotEmpty
- Job.ResourceNames.NotEmpty

## Rules applied to COBie.Document:

- Document.CreatedBy.CrossReference (ToContact)
- Document.CreatedBy.NotNull
- Document.CreatedOn.NotNull
- Document.CreatedOn.Valid (Valid Email Address)
- Document.ExternalSystem.NotEmpty
- Document.ExternalObject.NotEmpty
- Document.ExtIdentifier.NotEmpty
- Document.Description.NotEmpty
- Document.Category.NotNull
- Document.PrimaryKey.Unique (Name, Stage, SheetName, RowName)
- Document.Name.NotNull
- Document.ApprovalBy.NotEmpty
- Document.Stage.NotNull
- Document.SheetName.NotNull
- Document.SheetNameRowName.CrossReference
- Document.RowName.NotNull
- Document.Directory.NotNull
- Document.File.NotNull Reference.NotEmpty

## Rules applied to COBie.Attribute:

- Attribute.CreatedBy.NotNull
- Attribute.CreatedOn.NotNull
- Attribute.CreatedOn.Valid (Valid Email Address)
- Attribute.ExternalSystem.NotEmpty
- Attribute.ExternalObject.NotEmpty
- Attribute.ExtIdentifier.NotEmpty
- Attribute.Description.NotEmpty
- Attribute.Category.NotNull
- Attribute.PrimaryKey.Unique (Name, SheetName, RowName)
- Attribute.Name.NotNull
- Attribute.SheetNameRowName.CrossReference
- Attribute.Value.NotEmpty
- Attribute.Unit.NotEmpty
- Attribute.AllowedValues.NotEmpty

#### 4.2.8.1.4 Test rule formatting

COBie Quality Control testing rules, described above, have been implemented in a variety of formats by different organizations and companies. The checking tool of record in previous COBie Challenge events, the COBie Tool Kit has implemented these rules in an ISO rule checking format called Schematron. Widespread use of these rules has been made possible through the simplicity of the description of the rules and the distribution of these rules as an open-source project. The COBie Tool Kit is an open source product optimized to support both the verification of the content of COBie data and the validation of that information against the expected contract deliverable.

#### 4.2.8.1.5 Deliverable testing coverage

The COBie Tool Kit implements testing reports to evaluate the COBie exchanges at two points during the project: design and construction. The table below identifies the information required for design-phase COBie file submissions. The information is provided based on the COBie SpreadsheetML format to simplify understanding of the requirements, although the application is consistent regardless of the file format tested.

Under the buildingSMART alliance® Challenge events conducted to date the COBie.Assembly, COBie.Connection, COBie.Coordinate, and COBie.Issue worksheets are not required since these worksheets contain information that is currently outside the scope of the capture and exchange of information required for facility and asset management within the United States. Use of these worksheets for other purposes is possible, but is outside the scope of this standard. As a result, testing of these worksheets is only accomplished if information is provided as part of the test.

The COBie Tool Kit implements testing reports to evaluate the COBie exchanges at two points during the project: design and construction. The table below identifies the information required for handover-phase COBie file submissions. As before, the information is provided based on the COBie SpreadsheetML format to simplify understanding of the requirements, although the application is consistent regardless of the file format tested.

**Table 89 QC Testing - Design Rule Coverage**

Information Content	Required	Optional	N/A
COBie.Contacts	X		
COBie.Facility	X		
COBie.Floor	X		
COBie.Space	X		
COBie.Zone	X		
COBie.Type	X		
COBie.Component	X		
COBie.System	X		
COBie.Spare			X
COBie.Resource			X
COBie.Job			X
COBie.Document		X	
COBie.Attribute	X		

**Table 90 QC Testing - Construction Handover Coverage**

Information Content	Required	Optional	N/A
COBie.Contacts	X		
COBie.Facility	X		
COBie.Floor	X		
COBie.Space	X		
COBie.Zone	X		
COBie.Type	X		
COBie.Component	X		
COBie.System	X		
COBie.Spare	X		
COBie.Resource	X		
COBie.Job	X		
COBie.Document	X		
COBie.Attribute	X		

#### 4.2.8.2 Examples & mappings

##### 4.2.8.2.1 Example file list

The following examples are provided at the buildingSMART alliance® Common BIM File repository (East 2012a).

##### Duplex Apartment

- Planning – Space Programming
- Design - Coordinated Design
- Construction – Handover

##### Office Building

- Design - Coordinated Design

##### Medical Clinic

- Planning – Space Programming
- Design - Coordinated Design
- Construction – Handover

In the summer of 2013, one additional model entitled “Barracks 101” was also been added to the set of Common BIM Files.

#### 4.2.8.2.2 Example file description

**Duplex Apartment.** The duplex apartment model was originally created by a student who developed this building as part of a design competition. This model was used at the Dec 2009 COBie Challenge event.

**Office Building.** The two story office building model was developed based on the published sample floor plans for a specific type of mid-size office building built in the United States.

**Medical Clinic.** The Clinic Model project is based on a medical and dental clinic building at a location in the South-West United States. The model also comes with a set of redacted construction documents stage design drawings and operations and maintenance manuals. This model was used in buildingSMART alliance® Challenge events starting in 2011.

**Barracks 101.** The Barracks 101 project is based on a standard barrack building. Standard drawings in PDF format and associated design-stage COBie files have been provided. CADD files are currently being redacted for public release as of the time of preparation of this standard.

#### 4.2.8.2.3 Common BIM file reuse

The buildingSMART alliance Common BIM Files are used for all COBie testing.

The files used for the testing at buildingSMART alliance® Challenge events since 2011 have been based on the Medical Clinic project. This project was selected for several reasons: (1) it is a real project that has been designed and built, (2) a full set of redacted construction drawings and operations and maintenance manuals have been provided, and (3) it is of significant size and scope to address the majority of practical implementation issues likely to appear in practice.

#### 4.2.8.2.4 Implementers' agreements

As part of COBie Challenge events implementers' agreements have been developed and documented in a supplemental reference document called the "COBie Responsibility Matrix". This document is provided for public use and describes all implementers' agreements required and includes specific vendors' agreements. These agreements are required to support vendors with legal alternative implementations of IFC SPFF. These agreements are aimed at having most use of COBie with IFC data provided by vendors. The link to the COBie Responsibility Matrix may be found through the COBie Means and Methods website.

Implementers' agreements are not been required for submissions of SpreadsheetML and COBieLite formats since such formats are specified so that they do not allow multiple valid alternative implementations.

#### 4.2.8.2.5 SpreadsheetML transform

Detailed technical information required to transform COBie data between IFC and SpreadsheetML formats are provided in Annex A.

The use of SpreadsheetML format for COBie data is optional in this specification. It is the requirement of the implementing software companies to determine the most appropriate use of their software and resources within the context of their specific markets.

#### 4.2.8.2.6 COBieLite transform

Detailed technical information required to transform COBie data from SpreadsheetML and COBieLite schema are provided in Annex A.

The use of COBieLite is optional in this specification. It is the requirement of the implementing software companies to determine the most appropriate use of their software and resources within the context of their specific markets.

#### **4.2.8.3 Testing tools and procedures**

##### **4.2.8.3.1 Testing Tool List**

The following tools are available for testing COBie files. Information about the use of these tools and their availability of these tools is found in a later sub-section of this standard.

- COBie ToolKit
- bimServer.org
- bimServices
- EcoDomus PM
- Onuma System
- Google Docs

COBie data submitted in COBieLite format may also be validated by any number of free and commercial XML schema checking programs. The fact that COBieLite may be validated by a wide range of programmer-oriented tools reflects the overall goal of COBieLite to open use of COBie data outside the traditional buildingSMART alliance® community. These XML checking programs include, but are not limited to, the following:

- Component Assembly Mechanism (CAM) Checker
- XMLSpy
- Saxson XML Parser
- NetBeans Java Code
- MS Visual Studio

##### **4.2.8.3.2 Testing algorithm**

The testing of information exchanges must include both the testing of the export of COBie files, at the identified major project milestones, and the importation of COBie files by those who use COBie data in their products. Quality Control Tests describe the testing of exported COBie files. Quality Assurance Testing describes the process of manually checking that imported COBie data is accurate and has been used correctly within the context of the importing system. The algorithms for Quality Control and Quality Assurance testing are provided in this section.

###### **4.2.8.3.2.1 Quality control testing algorithm**

Quality control testing begins with the specification of the specific Common BIM Files needed for to test a given exchange. Recent testing has focused on the Medical Clinic model files. During a series of monthly team meetings software firms are able to ask questions pertaining to the specified files and the mapping of information between the test files and the specific data structures and workflows within their products. After meetings each software company produces the file based on the software that is



available for production use as of the date of the Challenge. Testing of beta software is allowed, by exception only.

The following components comprise the COBie Tool Kit algorithm for testing COBie 2.4 formatted files:

- Transformations necessary to normalize IFC, SpreadsheetML, and COBieLite formatted files
- A rule-set for testing COBie design phase files
- A rule-set for testing COBie construction/handover phase files

Transformation rules are described in Annex A. The rule sets themselves have been developed based on the Schematron ISO-based reference standard: ISO/IEC 19757-3:2006 “Information technology -- Document Schema Definition Language (DSDL) -- Part 3: Rule-based validation – Schematron”. The full implementation of the COBie rules, described earlier in this standard, into Schematron and associated implementation code (as needed) is available for inspection since the COBie ToolKit is an open-source project.

Automated testing of COBie files verifies that required fields are provided, business rules met, and proper data cardinality is found. Such a test does not, however, validate that the data is of any value to the user. For validation testing the information provided in the COBie file must match that provided on the equivalent drawing or submittal. The use of the Medical Clinic building is important in that the actual, redacted design and construction handover information on the real project is available for this purpose.

To validate the content quality of an exported COBie file buildingSMART alliance® testing publishes the overall statistics for the submitted files expected for design and construction phases. The software companies provide that information to their designers or data-entry personnel and develop the overall models. In addition to the overall model object counts, a limited number of specific spaces within the facility is to be modelled in detail. The collation of information found across all disciplines is provided to the companies to provide the objective set of detailed objects and associated attributes in specific spaces.

While the objective of buildingSMART alliance® Challenges is not to have the software companies re-design the test facility, it is important that some aspects of a fully designed model be tested. Such testing ensures that the software company producing the design drawings addresses individual software workflow issues in appropriate documentation.

When the number of deviations between the overall object counts and the specific space modelling requirements have been tallied the QC testing computes the amount of time required for a COBie-knowledgeable user to manually update a COBie Spreadsheet to correct these deviations. All models, testing, software instructions, and related materials is made public to allow users of tested software to reproduce the resulting tests at any time.

#### **4.2.8.3.2.2 Quality assurance testing**

Unlike testing exported COBie files, the testing of the import of COBie files is highly dependent upon the specific workflows and markets in which the importing software is to use. For example, software that assists a facility maintenance manager will need to import the preventative maintenance schedules from the COBie file to have a complete the set of information. A different type of software, for example one that only supports the management of tenants within a completed facility will not need the COBie data related to maintenance management.

Another difference between testing of exported and imported files, is that the ability to ensure that the relationships contained with the COBie, as well as the needed data, are properly represented. An example of such a test is to ensure that imported COBie components are placed within spaces. Given

that some maintenance management software has a data structure that explicitly represents space, and others do not, careful consideration of the use of the COBie data within the context of each individual software product is needed.

The conduct of Quality Assurance testing begins with the publication of the sample files by the organizer of the buildingSMART alliance® Challenge event. Typically, the files used will be based on the Medical Clinic Common BIM File example file at design and construction handover. Next, the software vendor identifies those aspects of the COBie data model that are relevant to their specific product. When ready the software vendor demonstrates the import of that COBie data set. The person conducting the QA testing may ask that specific information in the file be changed to verify that this new information is imported. All relevant software screens are reviewed for the presence and quantity of data found in the imported COBie data set. Any deviations are identified by the amount of time required to manually re-enter the data found in the COBie data set.

#### **4.2.8.3.2.3 Support of software development cycles**

The buildingSMART alliance® is supportive of software vendors using it standards within vendor's software development cycles. As a result, extensive efforts have been spent to work with vendors to provide staged approaches for partial implementation of COBie that allow the software company to fit this work within their other priorities. This means that different vendors in the same type of business may not be exporting or importing the same set of COBie data. Regardless of the stage of implementation of COBie, all testing results are made publically.

#### **4.2.8.3.2.4 Testing result format**

To reflect the variability in outcomes of various vendors' tests and the variation in level of implementation from each vendor the results of a buildingSMART alliance® Challenge are published through the buildingSMART alliance® Means and Methods Page. Each vendor has the same set of information to provide and is evaluated the same as any other vendor, within that class of software. Software products are not allowed to compete in sectors in which they are not typically used. For example, software for architectural and coordinated design must also be able to produce the contact deliverables that document the completion of those project phases.

Within a given class of software, results page lists those aspects of COBie relevant to that sector and also those which the vendor has, or has not, implemented as part of a given challenge event. The first buildingSMART alliance® Means and Methods page provides a vendor's most recent results. Past tests may be found by clicking through to see the detailed test results. The information on each of the vendors' detailed results pages is formatted to allow buildingSMART alliance® members to distinguish between those vendors with complete, or less complete, implementations of COBie.

Given that a vendor has identified a specific set of COBie data relevant to the users of their product, the number of individual differences between the expected COBie data file (or import data) and that which is provided in the vendor's export file or software are identified. These differences represent transcription or modeling errors by the software programming in the vendor's system. If the COBie data is to be corrected on the job site, manual data entry will be required. This manual data entry, for the purpose of a COBie Challenge is assessed at one minute per row of COBie data file that contains an error. If there are multiple errors in a given COBie data row, that is still counted as a single error. The quality score is reported at the number of minutes, or hours, required for a skilled person to correct the COBie file manually.

#### **4.2.8.3.3 Test files**

The buildingSMART alliance® Common Building Information Model files website (East 2012a) is the repository for all model files for COBie Challenge testing. COBie sample files for buildings (a duplex housing unit, a two story office building, and a medical/dental clinic) are available for testing. These

sample files are provided free of charge by the buildingSMART alliance® under a creative common licence. The form of that licence, which is not impacted by the reference in this document, allows the unlimited use of these for any purpose. Use of the files must be cited back to the buildingSMART alliance®. Users may change any of these files; however, if the files are changed they may no longer be referred to as one of the set of Common BIM Files.

#### 4.2.8.3.4 Testing tool software availability

The table below provides the list of tools and their associated websites. buildingSMART alliance® Challenge events used in the United States to test conformance of COBie files to NBIMS-US™ standards is the COBie Tool Kit. bimServer.org is provided in the list below since COBie ToolKit code is available either through the buildingSMART alliance® or through the open-source bimServer.org project. The authoritative source for the COBie ToolKit is the link provided through the buildingSMART alliance® provided later in this standard.

**Table 105 NBIMS-US™ COBie Testing Tools**

Product	Type	Description / Location
COBie ToolKit	Open Source	Model transformation, checking, and reporting a custom build of the bimserver.org product used in COBie Challenge events. <a href="http://www.projects.buildingsmartalliance.org/files/?artifact_id=5466">http://www.projects.buildingsmartalliance.org/files/?artifact_id=5466</a>
bimServer.org	Open Source	An IFC-based model server for life-cycle BIM application <a href="http://www.bimserver.org/">http://www.bimserver.org/</a>

Additional tools are also listed in table below. These tools may be applicable in regions other than the United States and also by users of specific company's products. For example, bimServices may be required for use in other regions, such as the United Kingdom. As such, a discussion bimServices is outside the scope of this standard.

Eventually COBie verification tools will be directly embedded within commercial software. These tools would allow those creating or consuming COBie data to identify and fix any errors that may be present in their data before submitting a given file. Two tools (EcoDomusPM, Onuma System) are provided by software companies to pre-check COBie deliverables. Another example of such a tool is one which directly integrates the collection of correct COBie data using the cloud-based Google-docs framework. Since product-based tools are not used as the official source of buildingSMART alliance® Challenge testing, such tools are outside the scope of this standard.

**Table 91 Regional and Vendor COBie Tools**

Product	Type	Description / Location
bimServices	Commercial – free version available	Model transformation, checking, and reporting. <a href="http://www.aec3.com/en/6/6_04.htm">http://www.aec3.com/en/6/6_04.htm</a>
EcoDomus PM	Commercial – free	Verifies design COBie to deliver construction handover information <a href="http://www.ecodomus.com/index.php/ecodomus-pm/">http://www.ecodomus.com/index.php/ecodomus-pm/</a>
Onuma System	Commercial – free	COBie file checker <a href="http://www.onuma.com/products/OpsAndCOBieValidate.php">http://www.onuma.com/products/OpsAndCOBieValidate.php</a>

Google Docs	Commercial - free	Web-based spreadsheet for updating COBie data  <a href="http://www.projects.buildingsmartalliance.org/files/?artifact_id=5445">http://www.projects.buildingsmartalliance.org/files/?artifact_id=5445</a>
-------------	-------------------	--

## 4.2.9 Implementation resources

*Author's Editorial Comment: Criteria for Implementation Resources were not included in NBIMS-US™ V2 criteria. The information provided here is an update to what would have been submitted had this requirement been part of the previous standard. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange standards, the information provided in this section should be considered a moderate change.*

### 4.2.9.1 Implementation resources list

#### 4.2.9.1.1 COBie guide

The COBie Guide (East 2012c) is a framework for project owners and teams to develop a practical implementation strategy toward COBie. Its function is similar to that of a “commentary” that accompanies a technical standard. Instructions provided in the Guide lead owners to identify their specific requirements and include such requirements directly in design and construction specifications.

Four COBie deliverables identified in the COBie Guide. Two are required during design and two during construction. Design deliverables at the 35% and Construction Documents stage of design are required. Construction Deliverables at the Beneficial Occupancy and Fiscal Completion stage are required.

COBie design deliverables must reflect data about scheduled assets identified in the associated design deliverables. Since scheduled assets appear on drawings in design schedules, the COBie guide provides a suggested minimum requirement for schedule table headers. Owners may adopt these table headers or leave the definition of product properties to the design team at their discretion. The only non-negotiable COBie data quality standard for design COBie files is that the data on the drawings must match the data in the COBie file. COBie construction deliverables must reflect updates to the designed assets and include construction and commissioning information consistent with the submittals that the Contractor is already required to produce in document-centric formats.

Feedback received from many teams has been that owners cannot articulate COBie requirements so the information that is provided to these owners is incomplete. COBie is designed for customization and is fully extensible through the use of owner-specific classifications, commonly required property sets, and specific requirements for space and product properties. The COBie Guide Appendix A tells the owner what should be customized in COBie and allows them to document their requirements.

#### 4.2.9.1.2 COBie Responsibility Matrix

The COBie Responsibility Matrix (East 2013) provides several sets of information to assist project team to define:

- Which team members are required to produce COBie data?
- What information should be found in a COBie file?
- How should that information be formatted?

Specifically, the sections of the COBie Responsibility Matrix include:

**Team Responsibility.** May be used by project teams to assign responsibility for different elements of the COBie data set to different stakeholders. It would be expected that such a matrix would be useful for those teams preparing a COBie implementation plan.

**Deliverable Requirements.** Provides a very detailed, life-cycle view, of what COBie information should be delivered when. This list may be used by project teams and owners to determine the expected content of COBie files at different project stages. As such, it is a synopsis of much of the information about exchange requirements identified previously in this standard.

**Spreadsheet Schema.** Provides the schema for the spreadsheet and IFC 2x3 versions of the same COBie data allowing the mapping between these formats. This list is a synopsis of the COBie Spreadsheet information and IFC mapping requirements found elsewhere in this standard.

**Type Assets.** Identifies those IFC "type" entities that, for the purpose of COBie are not considered to be "managed assets". This allows the filtering of IFC files when software companies fail to provide their own filters. This list is a synopsis of the MVD Exclusion lists identified in prior sections of this standard.

**Component Assets.** Identifies those IFC "component" entities that, for the purpose of COBie are not considered to be "managed assets". This allows the filtering of IFC files when software companies fail to provide their own filters. This list is a synopsis of the MVD Exclusion lists identified in prior sections of this standard.

**Property Sets.** Identifies IFC property sets that should be provided as part of the COBie data set. This allows filtering of IFC files when software companies fail to provide their own filters. This list is a synopsis of the MVD Exclusion lists identified in prior sections of this standard.

#### 4.2.9.1.3 Model merging guide

During design stages that require the coordination of information from multiple disciplines, the decisions regarding merging BIM model data are more complex than simply ensuring that drawings sheets are in the correct order. This is because BIM model files may have similar or overlapping data in many different files. Naively merging all data from all models into one consolidated COBie data set may be problematic since duplicate room numbers, light fixtures, plumbing fixtures and other information could occur.

At the request of software developers participating in the January 2013 buildingSMART alliance® Challenge a complete set of merge rules for all objects for both IFC 2x3 and IFC 4 were developed (East 2012d). This model merging guide was reportedly used by several software vendors who participated in the January 2013 buildingSMART alliance® Challenge.

The philosophy behind the merge rules follows the standard contract interpretation rule of "the more detailed information is likely to be more correct." A common example of the practical impact of such a rule is that the plumbing fixture schedule in an architect's model, developed before engaging a plumbing engineer, may not match the more detailed design provided by that plumbing consultant. As a result, the plumbing model data trumps (for specific types of objects) data found in the architect's model.

#### 4.2.9.1.4 Publication of challenge results

Delivering and using buildingSMART alliance® information exchange standards may be accomplished with existing software used for planning, design, construction, and facility management and operations activities. Software included in the COBie Means and Methods website (East 2012a) have worked directly with the buildingSMART alliance®, through a public process to test and demonstrate their capabilities.

The provided through that page provide the user with the most recent assessment of each products performance. Configuration guides and user manuals are also provided, based on the software at the

time of the testing. The results pages of the software systems may also be directly compared by viewing in separate browser tabs.

#### 4.2.9.1.5 COBie training

A variety of free training videos are currently provided for potential COBie users. This training is available from the main COBie website under “COBie Videos.” Additional training is provided directly by software vendors.

Training may also be available through the buildingSMART alliance®.

#### 4.2.9.1.6 COBie transformation tools

Since COBie data may be found in a variety of different formats, transformation tools may be employed to transform information in a given format to information in a format that is needed. The table below identifies two transformation tools. The first tool, the COBie ToolKit, is authorized for use in the context of NBIMS-US™. The COBie ToolKit has been tested to fully reproduce all information required in the COBie MVD regardless of format.

Another tool kit, bimServices, has been used in the United Kingdom to support its national COBie implementation. Since this tool kit has not been user or tested as part of buildingSMART alliance Challenge events, discussion of this tool is outside the scope of this standard.

**Table 92 COBie Transformation Tools**

Product	Type	Description / Location
COBie ToolKit	Open Source	Model transformation, checking, and reporting a custom build of the bimserver.org product used in COBie Challenge events.  <a href="http://www.projects.buildingsmartalliance.org/files/?artifact_id=5466">http://www.projects.buildingsmartalliance.org/files/?artifact_id=5466</a>
bimServices	Commercial – free version available	Model transformation, checking, and reporting.  <a href="http://www.aec3.com/en/6/6_04.htm">http://www.aec3.com/en/6/6_04.htm</a>

#### 4.2.9.1.7 COBie sample contract specifications

The term “information exchange” as the term used for projects such as COBie originated from the phrase “contracted information exchange” within the IFC community. In order to ensure that two parties would exchange the correct information there has to be some type of contract, either formal or informal, which sets expectations and indicates who is responsible in case something needs to be fixed. To support implementation of COBie a contract specification was created and has been made available through a link found on the primary COBie website: <http://www.wbdg.org/resources/COBie.php>.

At the time of submission of this standard the specification provided, Unified Facility Guide Specification 01 79 00, “Construction-Operations Building information exchange,” is expected to be implemented by the U.S. Army, Corps of Engineers in contracts beginning 01-Oct-14.

#### 4.2.9.1.8 LinkedIn group

Social media is being engaged as one of the first lines of support for buildingSMART alliance® information exchange formats. The first place to go for technical feedback on COBie is the Linked-In COBie Group

(<http://www.linkedin.com/groups/COBie-2638637/about>). This strongly moderated group may be able to help answer some shorter questions. Members of several companies have contributed time and support of buildingSMART alliance® information exchange projects, and related testing activities. These firms, listed below, may also be able to directly assist in your efforts.

#### **4.2.9.1.9 National Information Exchange Model**

The National Information Exchange Model (NIEM) is a collaborative network of United States Government entities at federal, state, tribal, and local levels. NIEM's particular point of view is to support government and public to share information about the world to plan, evaluate, and respond to man-made and natural disasters. Adding a NIEM-compliant information exchange schema to the set of COBie implementation formats provides the foundation to allow those in the disaster planning, mitigation, and response business to access COBie data critical to their enterprise.

#### **4.2.9.2 Business process coverage analysis**

Business processes related to the implementation of COBie covered in this standard pertain to the exchange of complete submissions of COBie files at the conclusion of major project milestones. Other workflows not addressed in this standard submission are contained in the section Workflow Coverage Analysis.

##### **4.2.9.2.1 Architectural programming coverage**

The requirements for the use of COBie for Architectural Programming have been so accepted that a more specific application of COBie, called the Building Programming information exchange (BPie) project, was begun mid-2012 (East 2012b). The BPie project subsumes many previous proprietary and non-proprietary approaches to capture and exchange architectural programming information.

##### **4.2.9.2.2 Architectural design coverage**

Those interested producing COBie data, of the early stage of a design that includes the Architect only, in a format that contains both the scheduled assets and associated geometry have used extensions to the Industry Foundation Class Coordination Model View Definition as well as COBie SpreadsheetML or COBieLite formats. Those interested in producing the COBie asset information only, may use any of the approved COBie data exchange formats.

The buildingSMART alliance® has provided Common BIM Files showing examples of the content of COBie files expected at this stage of a project (East 2012a).

While the COBie format is fixed for all exchanges, the complete set of information needed to be produced at any given deliverable must be specified for that deliverable. The list of objects expected to be included in the Architectural Design deliverable, and the quality criteria upon which such a deliverable is to be judged, is provided in the COBie Responsibility Matrix and COBie Guides. As these documents are considered commentary on the use of COBie, and not directly part of the specification of the COBie format itself, these documents are not included in this standard.

##### **4.2.9.2.3 Coordinated design coverage**

Those interested in producing COBie data, during any design stage that includes both architects and consulting engineers, in a format that contains both the scheduled assets and associated geometry may use extensions to the Industry Foundation Class Coordination Model View Definition as well as COBie SpreadsheetML or COBieLite formats. Those interested in producing the COBie asset information only, may use any of the approved COBie data exchange formats.

The buildingSMART alliance® has provided Common BIM Files showing examples of the content of COBie files expected, at the next stage of the project - the Construction Documents stage (East 2012a). These example files are technically equivalent to the COBie data files produced at this design stage.

While the COBie format is fixed for all exchanges, the complete set of information needed to be produced at any given deliverable must be specified for that deliverable. The list of objects expected to be included in the Architectural Design deliverable, and the quality criteria upon which such a deliverable is to be judged, is provided in the COBie Responsibility Matrix and COBie Guides. As these documents are considered commentary on the use of COBie, and not directly part of the specification of the COBie format itself, these documents are not included in this standard.

Another important guide produced to support software vendors in the January 2013 buildingSMART alliance® Challenge Event was a document that identified the order of precedence of objects in various designer and consultant models. This guide was produced at the direct request of software vendors. As this document should be considered a commentary on the use of COBie, within model servers or model merging software, and not directly part of the specification of the COBie format itself, this document was not included in this standard.

Information provided by the software vendors themselves, for the use of their own customers, has begun to address the variety of workflows that exist at design and consultancies. In addition to the “merge problem” noted a previous section, the problem of equipment schedule development is being addressed directly by the software vendors. This problem occurs when product and equipment schedules found on contract drawings were not created from data on the BIM objects, but pasted-in from another source. As such documents may contain proprietary information that is of interest only to the users of specific software, and not directly part of the specification of the COBie format itself, this document was not included in this standard.

#### **4.2.9.2.4 Construction Documents Coverage**

Those interested in producing COBie data, during any design stage that includes both architects and consulting engineers, in a format that contains both the scheduled assets and associated geometry may use extensions to the Industry Foundation Class Coordination Model View Definition as well as COBie SpreadsheetML or COBieLite formats. Those interested in producing the COBie asset information only, may use any of the approved COBie data exchange formats.

The deliverable of the construction documents- stage COBie data follows the same procedure used for any Coordinated Design Deliverable.

The buildingSMART alliance® has provided Common BIM Files showing examples of the content of COBie files at this stage of a project (East 2012a).

#### **4.2.9.2.5 Construction commissioning coverage**

Those producing COBie data, during the commissioning stage have, imported COBie Construction Documents stage data, in the SpreadsheetML format, and then exported updated COBie data, again in SpreadsheetML or COBieLite format. Since this is an interim, and likely an internal contractor deliverable, the specific quality criteria for such software is developed based on interviews with commissioning software vendors operating in this niche market.

The buildingSMART alliance® has provided Common BIM Files at the Construction Documents stage that are used as the sample import file (East 2012a).

#### **4.2.9.2.6 Construction handover coverage**



Those producing COBie data, at the handover stage have, imported COBie Construction Documents stage data, in the SpreadsheetML or COBieLite format, and then exported updated COBie data, again in SpreadsheetML or COBieLite format. Workflows for the construction processes included in these proprietary software systems are not evaluated during the buildingSMART alliance® Challenge process.

The buildingSMART alliance® has provided Common BIM Files at the Construction Documents stage that are used as the sample import file (East 2012a). In addition, example Construction Handover sample files are provided to assist in the evaluation of the completeness and quality of the construction handover data set.

#### **4.2.9.2.7 Facility start-up coverage**

Those consuming COBie data for purposes of maintenance- or asset-management has been accomplished using COBie data in SpreadsheetML, COBieLite, and IFC formats. Due to the differences in specific facility maintenance- and asset-management software systems, the specific quality criteria for such software are developed based on interviews with software system vendors operating in this market.

The buildingSMART alliance® has provided Common BIM Files at the Construction Handover stage that are used as the sample import file (East 2012a).

#### **4.2.9.2.8 Capture of as-built information coverage**

The creation of As-built building information using COBie as a data entry template is also a workflow that has been demonstrated. In this workflow a manual take-off of spaces and major equipment schedules is accomplished prior to the on-site visit. During an on-site visit specific name-plate and required spatial data can be directly collected in spreadsheet version of COBie. Rather than reliance on enterprise-wide data systems merging information collected can be accomplished simply by a cut and pasted of the information collected from each survey using the COBie SpreadsheetML format.

#### **4.2.9.3 Related workflow coverage analysis**

Workflows supported by the exchange of entire COBie files are only one part of the full set of workflows that pertain to facility asset information. The full set of workflows can be found in the Business Process Model section of this standard submission. The workflows that have been included and tested with regard to COBie 2.4 are the minimum essential workflows to include as contract deliverables in design and construction contracts. This is because these workflows represent specific paid contract milestone deliverables. Thus, the use of COBie for these milestone deliverables mirrors the current practice, and contractual framework, found in design and construction contracts.

The paragraphs below describe workflows that use all or part of the COBie data for different purposes other than the capture of building asset information at major project milestones. While these workflows share significant sets of the same information as COBie, they are not supported solely by the COBie 2.4 format due for two primary reasons. The first reason is that there are differences between COBie information exchanges and those noted in these processes. Some of these workflows require additional specification of COBie attribute data, others of these workflows use COBie data in contexts other than design, construction, and facility operations. The second reason is that many of these processes require persistent COBie data that would be a function of a model server or database. As COBie is an information exchange format, and not a software system or a database, these workflows may be supported by, but are not included as requirements for this standard submission.

Some of these workflows have their own buildingSMART alliance® projects, some do not. Eventually, it would be expected that each of these projects has an information exchange specification of its own.

#### 4.2.9.3.1 Product properties

Many workflows require specific information about product and equipment properties. While COBie has a place for such information, COBie.Type, COBie.Component, and COBie.Attribute, The COBie specification, itself, does not mandate specific properties for different product types. The specification of product properties was created as a separate project called the Specifiers' Properties information exchange (SPie, pronounced Spy) project. The reason that COBie and SPie were split into two projects is that the SPie effort ultimately requires the participation of many companies, associations, and manufacturers who are not currently members of the buildingSMART alliance®. When those firms are ready with the SPie data sets, their information may be safely exchanged using the COBie data format that already exists.

#### 4.2.9.3.2 Transaction data

Efforts on COBie prior to this standard submission have concerned the exchanges of entire sets of facility information in a complete COBie formatted file. There are many types of exchanges, however, where exchanging the entire COBie file exchange would not be helpful. An example of such an exchange is a designer's identification of product type properties based on a product used as a "basis of design." An example of such an exchange, during construction, is the capture of equipment installation information using a mobile device. In both of these examples it would be inefficient to exchange the entire COBie data set, when only a piece of the COBie data should be enough. A major motivation in the creation of the COBieLite schema was to support the discrete exchange of facility transactions using a variety of different types of devices and using a variety of different transmission protocols.

#### 4.2.9.3.3 Existing facility inventory surveys

Often facility managers are required to survey their facilities to identify some a specific class of asset or identify the properties associated with facility assets. COBie may be used as a consistent format to capture such information. Existing assets and associated properties could be exported for those conducting the survey. Personnel conducting the survey would add, update, or delete the information needed to make the data set current. This updated data set could then be merged from all surveyors and also merged back into the central asset management repository.

#### 4.2.9.3.4 Operations and maintenance history

An interest expressed by software companies providing maintenance and asset management tools is the potential use of COBie for facility operations and maintenance history. COBie assists in facility start-up, by providing a correct snap-shot of the as-built facility information. COBie does not address the workflows needed for facility- and asset-managers to exchange historic information after years of facility operations. It is possible that extensions to COBie's Job worksheet, and equivalent data structures, could be created to support these extensions.

#### 4.2.9.3.5 Maintaining as-operated COBie data

The question of how to update COBie data during the operations and maintenance phase of a project is a question that often arises in discussion of COBie. The objective in the development of COBie was to create an efficient replacement for paper construction handover deliverables. The business cases associated with maintaining the information once delivered to the facility maintenance, operations, and asset management offices was considered to be in the hands of the commercial software systems used for these purposes. If a buildingSMART alliance® project were to consider the future development of an open standard to exchange such information then there are three workflows where COBie data should be read, updated, or changed during the operational phase of a facility. Each of these cases is described below.

The first case is often referred to as a service call. These calls do not change the information at all, but only use the information provided. An example of such a call would be to change a filter. Since the most important managed equipment must be touched at least once a year, the capture of COBie data as part of that service call could result, within a year or eighteen months, of the capture of all major equipment at a facility.

The second case is often referred to as a work order. These jobs replace one piece of equipment with another. The location of the equipment does not change only the attributes such as manufacturer, model, serial number, and attribute data change. In this case the capture of COBie data could be a requirement needed prior to the completion of the work order.

The third case is often referred to as a facility renovation. In this situation, someone has to make a drawing in which case COBie data that is being removed from an underlying database can be flagged, and the new COBie data provided.

#### **4.2.9.3.6 Energy management**

An emerging area of interest in COBie data by several organizations and people outside the buildingSMART alliance® relates to the area of energy management. Energy management related work flows were not included in COBie. The ability of COBie to provide shared-structured information that may be used by energy managers without recollecting the information has been the subject of many conversations with those working in that space. The development that appears to have engaged many in that community is the development of the COBieLite schema. COBieLite allows programmers interested in determining information about buildings not to need to engage a detailed geometric representation but treat COBie data as just another type of data needed by software programmers.

#### **4.2.10 Revision plans**

*Author's Editorial Comment: Criteria for Revision Planning were not included in NBIMS-US™ V2 criteria. The information provided here is an update to what would have been submitted had this requirement been part of the previous standard. Based on the rules NBIMS-US™ V3 Technical Subcommittee Evaluation Criteria for Revised Information Exchange standards, the information provided in this section should be considered a moderate change.*

##### **4.2.10.1 Revision plans list**

###### **4.2.10.1.1 Revision management process**

Since its beginning, COBie has been developed using “spiral development model.” The spiral development model is characterized by a process of rapid prototyping and feedback. During each development cycle, improvements are introduced and implemented in software. Since the first version of COBie v1.0 in 2007, this spiral development model is now at the version of COBie, 2.4, that is represented in this standard.

Recent iterations of COBie began with discussion of technical developments during meetings with buildingSMART alliance® Challenge participants. During this meeting changes requested by the user community, through direct request to buildingSMART alliance® or through LinkedIn, are considered. In addition, changes requested by the software developers themselves are discussed. All changes are documented within Challenge meeting minutes and used to propose updates for future COBie versions. During the Challenge events software vendors may, at their discretion, test the agreed upon changes. If a consensus of software developers agree to these changes, then the next version of the standard is updated and documented for the next round of NBIMS-US™ revisions.

The “hands-on” revision management process described above is expected to be needed at the start of virtually all NBIMS-US™ information exchange standards. Once the standard becomes well established, however, the

level of intensity of work required for such a “hands-on” revision management process is not expected to be required. To that end, COBie Project Committees can be formed to systematize COBie revision management.

The COBie CMMS/CAFM Project Group was formed in August 2013 to assist CMMS/CAFM vendors further extend and develop the COBie model for areas beyond design and construction. The importation of COBie construction handover data sets into CMMS/CAFM software is well established.

The COBie Construction Project Group will be formed the first quarter of 2014 to assist Construction and Commissioning (C&C) stage vendors further extend and develop the COBie model for more detailed construction workflows. C&C software is used to import COBie design data, update that data with information gathered during construction, and export that data into COBie construction handover data sets. This process of importation, updating, and exporting COBie data for C&C is well established.

The COBie Project Groups are being formed to support the following functions:

- Identification of Venues for COBie Presentations
- Coordination with Venue Organizers and Speakers
- Development of Consensus-based COBie Standard Updates
- Coordination of COBie verification and validation testing, e.g. COBie Challenges

The charters of each of the COBie project Groups will cover the specifics of revision management for their community, and others who provide or delivery COBie data during the life of the project.

#### **4.2.10.1.2 Revision management notification**

To date, public change notifications have been posted on the LinkedIn COBie Group and buildingSMART alliance® COBie revision notification page. An example of the revision notifications between the NBIMS-US™ V2 COBie version 2.26 and the NBIMS-US™ V3 COBie version 2.4 may be found the COBie Version 2.4 update page (Nisbet 2012).

#### **4.2.10.2 Proposed revision deployment methods**

##### **4.2.10.2.1 Revision deployment process**

Deployment of revisions to COBie occurs through several mechanisms. First, through the software that implements the revisions. This software is then tested as part of an official buildingSMART alliance® Challenge processes. The results of software performance against the baseline any new revisions are documented through the buildingSMART alliance® Means and Methods Page (East 2013).

Parallel with software implementation is the documentation of the exact changes to the standard that have been deployed. The baseline changes to the standard are documented and distributed for all software systems participating in the buildingSMART alliance® Challenge. Since such changes may have slight, but important, variations in individual software systems additional documentation, called the COBie Responsibility Matrix, is also published as part of the Challenge. Once the challenge has been complete, the updated COBie Responsibility Matrix is released through the COBie website.

Parallel with the activities above COBie checking software, called the COBie Tool Kit, is updated to allow the results of the COBie Challenge to be tested. Once the challenge has been complete, the updated COBie Tool Kit is released through the COBie website.

To support the testing of the new revisions testing files, based on the buildingSMART alliance® Common BIM Files, are created. These files demonstrate the changes using the project used as the basis for testing during the Challenge event.

#### 4.2.10.2.2 Revision deployment notification

To date, all COBie updates have been presented at the buildingSMART alliance® Challenge event. A presentation that includes the discussion of these changes is included in the conference proceedings. The presentation is available following the meeting as part of the proceedings. Software guidance on any impacts the change may have on the use of specific commercial software is the responsibility of the software vendor. Information regarding vendor implementation of COBie is catalogued on the COBie Means and Methods page (East 2013).

### Bibliography

CSS-1, Cascading Style Sheets, level 1 — W3C Recommendation

East, E (2007) "Construction Operations Building information exchange (COBie): Requirements Definition and Pilot Implementation Standard," U.S. Army, Engineer Research and Development Center, Construction Engineering Research Laboratory, Technical Report TR-07-30.  
[http://www.wbdg.org/pdfs/erdc\\_cerl\\_tr0730.pdf](http://www.wbdg.org/pdfs/erdc_cerl_tr0730.pdf) (cited 17-Jun-13)

East., E, Love, D., Nisbet, N. (2010) "Life-Cycle information exchange," Proceedings of the 27th International Conference of WG 078, CIB. [http://projects.buildingsmartalliance.org/files/?artifact\\_id=3448](http://projects.buildingsmartalliance.org/files/?artifact_id=3448) (cited 17-Jun-13)

East, E., Love, D. (2011) "Value-Added Analysis of the Construction Submittal Process," Automation in Construction, Elsevier, 20(2011), pp. 1070-1078.  
<http://www.sciencedirect.com/science/article/pii/S0926580511000574> (cited 17-Jun-13)

East, E. (2012) (2012a) "Common Building Information Model Files and Tools," National Institute of Building Sciences, buildingSMART alliance®, [http://www.nibs.org/?page=bsa\\_commonbimfiles](http://www.nibs.org/?page=bsa_commonbimfiles) (cited 17-Jun-13)

East, E. (2012b) "Building Programming information exchange," National Institute of Building Sciences, buildingSMART alliance®, [http://www.nibs.org/?page=bsa\\_bpie](http://www.nibs.org/?page=bsa_bpie) (cited 17-Jun-13)

East, E (2012c) "The COBie Guide: A Commentary to the NBIMS-US™ COBie Standard," [http://projects.buildingsmartalliance.org/files/?artifact\\_id=5879](http://projects.buildingsmartalliance.org/files/?artifact_id=5879) (cited 18-Jun-13)

East (2012d) "COBie Model Merging Rules"  
[http://projects.buildingsmartalliance.org/files/?artifact\\_id=4705](http://projects.buildingsmartalliance.org/files/?artifact_id=4705) (cited 23-Sep-13)

East, E. (2013) "buildingSMART alliance® information exchanges: Means and Methods," [http://www.nibs.org/?page=bsa\\_COBiemm](http://www.nibs.org/?page=bsa_COBiemm) (cited 18-Jun-13)

East, E., et al. (2013b) "COBie Responsibility Matrix",  
[http://projects.buildingsmartalliance.org/files/?artifact\\_id=4093](http://projects.buildingsmartalliance.org/files/?artifact_id=4093) (cited 23-Sep-13)

Fallon, K., Fadojutimi, O., Williams, G., Crawford, N., Gran, D., (2013) "Assessment of Life-Cycle Information Exchanges (LCie): Understanding the Value-Added Benefits of a COBie Process," U.S. Army, Engineer Research and Development Center, Construction Engineering Research Laboratory, Draft Technical Report, ERDC/CERL TR-11-DRAFT.

ISO 639-1, Codes for the representation of names of languages — Part 1: Alpha-2 code

ISO 639-2, Codes for the representation of names of languages — Part 2: Alpha-3 code

ISO 639-3, Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages

ISO 6707-1, Building and civil engineering — Vocabulary — Part 1: General terms

ISO 8601, Data elements and interchange formats — Information Exchange — Representation of dates and times.

ISO 10303-11:1994, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles

ISO 10303-11, Industrial automation systems and integration — Product data representation and exchange — Part 11: description methods: The EXPRESS Language Reference Manual

ISO 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure

ISO 10303-28, Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas

ISO 10303-41, Product data representation and exchange — Integrated generic resource — Fundamentals of product description and support

ISO 10303-42, Product data representation and exchange — Integrated generic resource — Geometric and topological representation

ISO 10303-43, Product data representation and exchange — Integrated generic resource — Representation structures

ISO 10303-46, Product data representation and exchange — Integrated generic resource — Visual presentation

ISO 10303-514, Product data representation and exchange — Application interpreted construct — Advanced boundary representation

ISO 12006-3, Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information

ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation.

ISO/IEC 14772-1, Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding

ISO/IEC 19775-1, Information technology — Computer graphics and image processing — Extensible 3D (X3D) — Part 1: Architecture and base components

ISO/IEC 81346-12, Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 12: Buildings and building services

ISO/IEC 19757-3:2006 Information technology -- Document Schema Definition Language (DSDL) -  
- Part 3: Rule-based validation -- Schematron

Nisbet., N., East, E. (2012) "Construction Operations Building information exchange (COBie): Version 2.40 Update," buildingSMART alliance®, [http://www.nibs.org/?page=bsa\\_COBiev24](http://www.nibs.org/?page=bsa_COBiev24) (cited 18-June-13)

XML Schema Part 2, XML Schema Part 2: Datatypes — W3C Recommendation

RFC 3986, Uniform Resource Identifier (URI): Generic Syntax — Network Working Group NWG Standard

RFC 5646, Tags for Identifying Languages — Internet Engineering Task Force IETF Best Current Practice 47

## Table of Figures

Figure 1 Study/Define Needs Process Model.....	53
Figure 2 Develop Design Criteria Process Model.....	54
Figure 3 Study Technical Feasibility Process Model.....	55
Figure 4 Communicate Results Decisions Process Model.....	56
Figure 5 Develop Program - Space Program Process Model.....	57
Figure 6 Develop Program - Product Program Process Model.....	58
Figure 7 Prepare Invitation to Bid and Receive Proposals (Pre-Design) Process Model .....	59
Figure 8 Explore Concepts - Design Early Process Model .....	60
Figure 9 Develop Design - Design Schematic Process Model.....	61
Figure 10 Develop Design - Design Coordinated Process Model.....	62
Figure 11 Product Type Template Process Model .....	63
Figure 12 Develop Design – Product Type Candidate Process Model .....	64
Figure 13 Finalize Design - Design Final Process Model.....	65
Figure 14 Finalize Design – Product Type Candidate Process Model.....	66
Figure 15 Prepare Invitation to Bid and Receive Proposals (Post-Design) Process Model.....	67
Figure 16 Respond to Pre-Proposal Inquiries Process Model .....	68
Figure 17 Develop Pre-Construction Plans Process Model .....	69
Figure 18 Identify Discrepancies Process Model .....	70
Figure 19 Prepare Submittal Information Product Type Selection Process Model .....	71
Figure 20 System Layout Process Model.....	72
Figure 21 Organize Submittal Information Process Model.....	73
Figure 22 Perform Submittal Review - Submittal Issue Process Model.....	74
Figure 23 Provide Resources Process Model.....	75
Figure 24 Execute Construction Activities Process Model .....	76
Figure 25 Product Installation Process Model.....	77
Figure 26 Perform Equipment Testing Process Model.....	78
Figure 27 Inspect and Approve Work Process Model .....	79
Figure 28 Define, Record, and Certify Discrepancies Process Model .....	80
Figure 29 Closeout Process Model .....	81
Figure 30 Stakeholder Coverage.....	82
Figure 31 ifcSpace - Spatial Structure.....	119
Figure 32 ifcSpace - Heights .....	120
Figure 33 ifcBuildingStory - Spatial Structure.....	123
Figure 34 ifcBuildingStory - Heights .....	123
Figure 35 ifcBuilding - Spatial Structure .....	125
Figure 36 ifcBuilding - Heights.....	126
Figure 37 ifcSite - Spatial Structure .....	129
Figure 38 ifcBuilding - Heights.....	130
Figure 39 ifcProcess.....	145
Figure 40 IfcWorkSchedule .....	155
Figure 41 ifcSpace Concepts.....	163
Figure 42 ifcBuildingStory Concepts .....	165
Figure 43 ifcBuilding Concepts .....	166
Figure 44 ifcSite Concepts.....	167
Figure 45 ifcProject Concepts.....	169
Figure 46 ifcSystemConcepts.....	170
Figure 47 ifcElement Concepts.....	171
Figure 48 ifcElementType Concepts .....	172
Figure 49 ifcTask Concepts .....	173
Figure 50 ifcActor Concepts .....	174
Figure 51 ifcActionRequest Concepts .....	175
Figure 52 ifcProcess Concepts.....	176
Figure 53 ifcGroup Concepts.....	177
Figure 54 ifcPropertySetConcepts.....	177



Figure 55 ifcAnnotationConcepts .....	177
Figure 56 ifcRelConnectsPort Concept .....	178
Figure 57 ifcWorkScheduleConcepts .....	178
Figure 58 ifcRelAssociates Document.....	178
Figure 59 ifcRelConnectsWithRealizingElement Concepts.....	179
Figure 60 ifcConstructionProductResource Concept .....	179
Figure 61 ifcConstructionproductResourceType Concept.....	179
Figure 62 Identity Concept - Business Rule .....	204
Figure 63 Revision Control Concept - Business Rule .....	205
Figure 64 External Data Constraints Concept - Business Rule.....	206
Figure 65 Classification Concept - Business Rule.....	207
Figure 66 Spatial Composition Concept - Business Rule.....	207
Figure 67 Quantities on Occurrences Concept - Business Rule .....	208
Figure 68 Conversion Units Concept - Business Rule .....	208
Figure 69 Project Declaration Concept - Business Rule .....	208
Figure 70 Group Assignment Concept - Business Rule.....	209
Figure 71 Object Typing Concept - Business Rule .....	210
Figure 72 Spatial Containment Concept- Business Rule .....	210
Figure 73 Property Sets for Occurrences Concept- Business Rule .....	211
Figure 74 Property Sets for Types Concept - Business Rule.....	211
Figure 75 Task Scheduling Concept - Business Rule .....	212
Figure 76 Sequential Connectivity Concept - Business Rule .....	212
Figure 77 Contact Concept - Business Rule .....	213
Figure 78 Control Assignment Concept - Business Rule .....	213

## Table of Tables

Table 1 Business Case - Office Project.....	13
Table 2 Business Case - Medical Clinic .....	13
Table 3 - Business Case Transit Authority .....	14
Table 4 Stakeholder Coverage Analysis .....	16
Table 5 Study and Define Needs Document List.....	82
Table 6 Develop Design Criteria Document List.....	83
Table 7 Study Technical Feasibility Document List.....	83
Table 8 Communicate Results Decision Document List .....	83
Table 9 Develop Program – Space Program Document List .....	83
Table 10 Develop Program – Product Program Document List .....	83
Table 11 Prepare Invitation to Bid and Receive Proposal (Pre-Design) Document List .....	83
Table 12 Explore Concepts – Design Early Document List.....	83
Table 13 Develop Design – Design Schematic Document List.....	83
Table 14 Develop Design – Design Coordinated Document List .....	84
Table 15 Finalize Design – Design Final Document List.....	84
Table 16 Prepare Invitation to Bid and Receive Proposals (Post Design) Document List.....	84
Table 17 Respond to Pre-Proposal Inquiries Document List .....	84
Table 18 Develop Pre-Construction Plan Document List .....	84
Table 19 Identify Discrepancies Document List .....	85
Table 20 Prepare Submittal Information – Product Type Selection Document List .....	85
Table 21 Prepare Submittal Information – System Layout Document List.....	85
Table 22 Organize Submittal Information Document List.....	85
Table 23 Perform Submittal Review – Submittal Issue Document List .....	85
Table 24 Provide Resources Document List .....	86
Table 25 Execute Construction Activities Document List .....	86
Table 26 Perform Equipment Testing Document List.....	86
Table 27 Inspect and Approve Work Document List .....	86
Table 28 Define, Record and Certify Discrepancies Document List .....	86
Table 29 Closeout Document List .....	86

Table 30 Classification of Initiation Exchanges .....	93
Table 31 Classification of Design Exchanges .....	94
Table 32 Classification of Construction Exchanges .....	94
Table 33 Classification of Operational Exchanges .....	95
Table 34 Exchange Requirements Coverage.....	95
Table 35 Use of Coordinates with COBie .....	104
Table 36 Business Rules – Design-Related Exchanges .....	105
Table 37 Business Rules – Discipline Specifications .....	106
Table 38 Business Rules – Project Definition.....	106
Table 39 Business Rules – Issue Type Exchanges .....	107
Table 40 Business Rules – Product Type Exchanges .....	108
Table 41 Business Rules – Product Installation .....	109
Table 42 Business Rules – O&M Information Type Exchanges.....	109
Table 43 Business Rules – Product Parts Replacement.....	110
Table 44 Business Rules - Post-occupancy Space Exchanges.....	111
Table 45 Business Rules – Post-occupancy Space & Product Exchanges .....	112
Table 46 Related Business Process List .....	115
Table 47 Exchange Requirement Reuse Analysis .....	117
Table 48 Classification Constraints .....	180
Table 49 Property Sets for Occurrences Constraints.....	182
Table 50 Property Sets for Types Constraints.....	182
Table 51 IfcSpace - Concept Applicability .....	184
Table 52 IfcBuildingStory - Concept Applicability .....	184
Table 53 IfcBuilding - Concept Applicability .....	185
Table 54 IfcSite - Concept Applicability .....	185
Table 55 IfcProject - Concept Applicability .....	185
Table 56 IfcZone - Concept Applicability .....	186
Table 57 IfcSystem - Concept Applicability .....	186
Table 58 IfcElement - Concept Applicability .....	187
Table 59 IfcElementType - Concept Applicability .....	187
Table 60 IfcTask - Concept Applicability .....	188
Table 61 IfcActor - Concept Applicability .....	188
Table 62 IfcActionRequest - Concept Applicability.....	189
Table 63 IfcPropertySet - Concept Applicability .....	189
Table 64 IfcAnnotation - Concept Applicability .....	189
Table 65 IfcRelConnectsPorts - Concept Applicability .....	190
Table 66 IfcWorkSchedule - Concept Applicability.....	190
Table 67 IfcRelAssociatesDocument - Concept Applicability.....	190
Table 68 IfcRelConnectsWithRealizingElements - Concept Applicability .....	191
Table 69 ifcConstructionProductResource – Concept Applicability .....	191
Table 70 ifcConstructionProductResourceType – Concept Applicability .....	191
Table 71 Exchange Requirement Reuse Analysis .....	192
Table 72 Identity Concept Business Rule List.....	192
Table 73 Revision Control Concept Business Rule List.....	193
Table 79 Quantities on Occurrences Concept Business Rule List.....	196
Table 80 Conversion Units Concept Business Rule List.....	197
Table 82 Group Assignment Concept Business Rule List.....	198
Table 83 Object Typing Concept Business Rule List .....	198
Table 84 Spatial Containment Concept Business Rule List.....	198
Table 87 Task Scheduling Concept Business Rule List.....	200
Table 92 Control Assignment Concept Business Rule List.....	203
Table 93 Allowable COBie MVD Formats .....	214
Table 94 COBie_Task Dynamic Schema Definition.....	215
Table 95 COBie_ElementType Dynamic Schema Definition .....	216
Table 96 Pset_ManufacturerOccurrence Dynamic Schema Definition .....	217
Table 97 Pset_ManufacturerTypeInfo Dynamic Schema Definition.....	217

Table 98 Pset_ActionRequest Dynamic Schema Definition.....	217
Table 99 COBie_ActionRequest Dynamic Schema Definition .....	218
Table 97 COBie_ConstructionProductResource Dynamic Schema Definition .....	218
Table 103 QC Testing - Design Rule Coverage .....	229
Table 104 QC Testing - Construction Handover Coverage.....	230
Table 106 Regional and Vendor COBie Tools .....	235
Table 107 COBie Transformation Tools .....	238

[Annex A – COBie Mapping Rules](#)

[Annex B – Life Cycle information exchange \(LCie\) for Product and Product Type data exchanges, a technical Annex to COBie – Verion 2.4](#)