Министерство строительства
и жилищно-коммунального хозяйства
Российской Федерации

Федеральное автономное учреждение
«Федеральный центр нормирования, стандартизации
и оценки соответствия в строительстве»

Методическое пособие

ОБЕСПЕЧЕНИЕ ИНТЕРОПЕРАБЕЛЬНОСТИ ПРИ ИНФОРМАЦИОННОМ МОДЕЛИРОВАНИИ ОБЪЕКТОВ СТРОИТЕЛЬСТВА

Москва 2017

gost.e-ecolog.ru

Содержание

Введение
1 Основные положения 6
1.1 Область применения 6
1.2 Нормативные ссылки 6
1.3 Предпосылки 7
1.4 Дополнительные термины, обозначения и определения, используемые в настоящем методическом пособии 9
1.4.1 Термины и определения 9
1.4.2 Обозначения и сокращения 12
2 Обеспечение интероперабельности на организационном уровне 13
2.1 Роли в бизнес-процессах по информационному моделированию 14
2.1.1 Роль: BIM-менеджер 14
2.1.2 Роль: BIM-координатор организации 15
2.1.3 Роль: BIM-автор организации 15
2.2 Валидация информации 16
2.2.1 Процесс валидации 16
2.2.2 Контроль качества информации 16
2.2.3 Виды проверок 17
2.2.4 Проверка на наличие коллизий. 18
2.2.5 Координационные совещания по информационному моделированию 18
2.3 Информационные требования заказчика 19
2.4 Хранение и обмен данными в организации 20
2.5 Пример фрагмента регламента организации об обмене данными 21
3 Обеспечение интероперабельности на семантическом уровне 23
3.1 Структура области знаний информационной модели 24
3.1.1 Тезаурусы 26
3.1.2 Классификаторы 26
3.1.3 Кодирование позиций 30

3.2 Международные системы классификации 31
3.2.1 Требования международных стандартов 32
3.2.2 Примеры международных классификаторов 36
3.3 Отечественные системы классификации 37
3.3.1 Требования к отечественной структуре классификации 37
3.3.2 Российский классификатор национального уровня 38
3.4 Синтаксическая и семантическая проверка информационных моделей 40
3.5 Семантическое описание информационных моделей 40
3.6 Выводы 45
4 Обеспечение интероперабельности на программно-техническом уровне 46
4.1 Методические положения по экспорту и импорту данных информационной модели 46
4.2 Виды технической интероперабельности 47
4.3 Особенности применения IFC 50
4.4 Рекомендации по использованию возможностей формата IFC при экспорте данных в программные комплексы 61
Приложение 1 Форматы обмена данными 63
Приложение 2 Некоммерческая международная организация buildingSMART 155
Список использованных источников 157

Введение

В настоящем методическом пособии приведены основные положения для достижения интероперабельности в области технологии информационного моделирования объектов строительства. Пособие разработано для применения широким кругом специалистов, чья деятельность связана с проектированием и строительством зданий (сооружений):

- проектных, строительных и эксплуатационных организаций;
- государственных и иных органов экспертизы и согласования;
- органов лицензирования и сертификации.

Внедрение технологий информационного моделирования объектов промышленного и гражданского строительства требует решения проблемы эффективного обмена информацией в гетерогенной среде информационных систем, функционирующих в проектных, строительных, эксплуатационных организациях, а также у заказчика (инвестора).

Благодаря технологии информационного моделирования в гетерогенной среде возникает дополнительный системообразующий фактор – информационная модель, эволюционирующая в течение жизненного цикла здания (сооружения).

Настоящее методическое пособие способствует решению вопросов четкой организации передачи информации, увязки ее смыслового содержания и форматов обмена данными, то есть решению проблемы интероперабельности при внедрении технологии информационного моделирования.

Настоящее методическое пособие разработано в развитие положений свода правил «Информационное моделирование в строительстве. Правила обмена между информационными моделями объектов и моделями, используемыми в программных комплексах», 2016 г. и ГОСТ Р 55062-2012 «Информационные технологии. Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения».

В пособии приведены методы достижения интероперабельности при взаимодействии информационных систем и их компонентов (программных

комплексов и программных платформ, поддерживающих технологию информационного моделирования).

Пособие разработано авторским коллективом Акционерного общества «Научно-исследовательский центр «Строительство» (АО «НИЦ «Строительство») – ЦНИИСК им. В.А. Кучеренко в составе: к.т.н. Ю.Н. Жук, А.В. Ананьев, Б.В. Волков, Ю.А. Сыромятников, к.т.н. Е.В. Позняк.

1 Основные положения

1.1 Область применения

Данное методическое пособие предназначено для специалистов и руководителей проектно-изыскательских, строительных организаций, эксплуатационных организаций, учреждений и служб заказчика (инвестора) и других заинтересованных организаций, с целью обеспечения их организационно-методическими материалами, которые позволяют применять высокоэффективные технологии информационного моделирования.

Применение методического пособия позволит в конечном счете повысить качество выполняемых проектных работ, сократить сроки и снизить стоимость строительства, снизить затраты на эксплуатацию объектов за счет использования типовых подходов к выполнению работ по обеспечению интероперабельности.

1.2 Нормативные ссылки

В настоящем пособии применены нормативные ссылки на следующие стандарты:

1) ГОСТ Р 55062-2012 Информационные технологии. Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения.
2) ГОСТ Р ИСО 12006-2–2017 Строительство. Модель организации данных о строительных работах. Часть 2. Основы классификации информации.
3) ГОСТ Р ИСО 12006-3–2017 Строительство. Модель организации данных о строительных работах. Часть 3. Основы обмена объектно-ориентированной информацией.
4) ГОСТ Р ИСО 22263–2017. Модель организации данных о строительных работах. Структура управления проектной информацией.

5) ГОСТ Р 57563-2017 Моделирование информационное в строительстве. Основные положения по разработке стандартов информационного моделирования зданий и сооружений.
6) ГОСТ Р ИСО 16739 Моделирование информационное зданий и сооружений. Отраслевые базовые классы (IFC) для обмена информацией на всех этапах жизненного цикла. Основные положения.
7) ГОСТ Р 57310-2016 Моделирование информационное в строительстве. Руководство по доставке информации. Методология и формат.
8) ГОСТ Р 57311-2016 Моделирование информационное в строительстве. Требования к эксплуатационной документации объектов завершенного строительства.
9) СП Информационное моделирование в строительстве. Правила обмена между информационными моделями объектов и моделями, используемыми в программных комплексах.
10) СП Информационное моделирование в строительстве. Правила формирования информационной модели объектов на различных стадиях жизненного цикла.
11) СП Информационное моделирование в строительстве. Правила описания компонентов информационной модели.
12) СП 301.1325800.2017 Информационное моделирование в строительстве. Правила организации работ производственно-техническими отделами.

1.3 Предпосылки

Недостаточная интероперабельность часто упоминается в качестве основного барьера, препятствующего эффективному обмену данными и информацией между участниками проектных групп, работающих в архитектурно-строительной отрасли и является предметом многих проводимых в мире в этой отрасли исследований. В других областях знаний, в том числе в области информационных технологий (ИТ), здравоохранения и военных исследований, понятие интероперабельности уже вышло за пределы технической сферы и в настоящее время приобрело более

широкий смысл. Так в отрасли ИТ сравнительно недавно был введен в действие ГОСТ Р 55062-2012 «Информационные технологии. Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения», который предназначен для заказчиков, поставщиков, разработчиков, потребителей, а также персонала, сопровождающего информационные системы и осуществляющего программное обеспечение и услуги. В 2014 году Техническим комитетом по стандартизации ТК 100 «Стратегический и инновационный менеджмент» внесен ГОСТ Р ИСО 11354-1-2012 «Усовершенствованные автоматизированные технологии и их применение. Требования к установлению интероперабельности процессов промышленных предприятий. Часть 1. Основа интероперабельности предприятий», который устанавливает основы и базовые принципы интероперабельности предприятий, определяющие размеры и направления для рассмотрения барьеров при достижении интероперабельности, возможных решений и взаимосвязей между ними.

Во введении к ГОСТ Р 55062-2012 в частности указывается, что «интероперабельность приобретает все большее значение, в первую очередь потому, что сегодня практически ни одна сфера жизни (государственное управление, здравоохранение, образование, наука, бизнес и др.) не обходится без использования информационно-коммуникационных технологий. Можно констатировать, что обеспечение интероперабельности является одной из основ формирования и развития информационного общества. Развитие информационно-коммуникационных технологий (ИКТ) и насыщение всех сфер деятельности различными средствами вычислительной техники привели к созданию гетерогенной среды, в которой разнородные информационные системы (компоненты) должны взаимодействовать друг с другом, причем уровень гетерогенности среды постоянно увеличивается. Основным способом решения проблемы интероперабельности или «прозрачности» гетерогенной среды выступает последовательное применение принципов открытых систем.

В архитектурно-строительной отрасли назрела необходимость определения способов обмена информацией и взаимодействия организаций, совместно работающих над одним инвестиционным проектом.

1.4 Дополнительные термины, обозначения и определения, используемые в настоящем методическом пособии

1.4.1 Термины и определения

1.4.1.1 **актор**: Физическое лицо, организация или ее подразделение (такое как департамент, группа и т.д.), вовлеченные в процесс строительства.

1.4.1.2 **бизнес-правило**: Утверждение, которое формально определяет или ограничивает некоторые аспекты бизнеса, правила, согласно которым функционирует организация, или решения, влияющие на бизнес-процессы.

1.4.1.3 **бизнес-требования**: Требования, которые описывают в терминах для в области-гражданско-правовых отношений к тому, что необходимо предоставить или реализовать.

1.4.1.4 **дескриптор**: Термин естественного языка (слово или словосочетание), используемый при описании документов или показателей, который имеет самостоятельный смысл и неделим без изменения своего значения.

1.4.1.5 **жизненный цикл здания или сооружения;** ЖЦ: Период, в течение которого осуществляются инженерные изыскания, проектирование, строительство (в том числе консервация), эксплуатация (в том числе текущие ремонты), реконструкция, капитальный ремонт, снос здания или сооружения.

1.4.1.6 **BIM-проект**: Инвестиционно строительный проект, выполняемый с применением технологии информационного моделирования.

1.4.1.7 **интероперабельная система**: Система, в которой входящие в нее подсистемы работают по независимым алгоритмам, не имеют единой точки управления, все управление определяются единым набором стандартов — профилем интероперабельности.

1.4.1.8 **интероперабельность**: Способность двух или более информационных систем или компонентов к обмену информацией и использованию информации, полученной в результате обмена.

1.4.1.9 **информационная модель**: Совокупность представленных в электронном виде графических и неграфических данных по объекту строительства, технической и иной документации, размещаемая в соответствии с установленными правилами в среде общих данных, представляющей собой единый достоверный источник информации по объекту на всех или отдельных стадиях его жизненного цикла.

Примечание – При постадийной разработке информационной модели рекомендуется выделять концептуальную/эскизную, проектную, строительную, исполнительную и эксплуатационную информационные модели.

1.4.1.10 **информационное моделирование объектов строительства**: Процесс создания и использования информации по строящимся, а также завершенным объектам капитального строительства в целях координации входных данных, организации совместного производства и хранения данных, а также их использования для различных целей на всех этапах жизненного цикла.

1.4.1.11 **карта взаимодействия**: Представление ролей и транзакций, соответствующих конкретной цели, в виде карты.

1.4.1.12 **карта процесса**: Представление характеристик процесса и определенных целей.

1.4.1.13 **классификация**: логическая операция, состоящая в разделении всего изучаемого множества предметов по обнаруженным сходствам и различиям на отдельные группы путем перечисления и указания признака или нескольких признаков, присущих члену группы.

1.4.1.14 **модель требования к обмену информацией**: Техническое выражение требования к обмену информацией в виде схемы.

1.4.1.15 **организационная интероперабельность**: Способность участвующих систем достигать общих целей на уровне бизнес-процессов.

1.4.1.16 **плагин**: Программный модуль, разрабатываемый независимо от основной программы и динамически к ней подключаемый.

1.4.1.17 **программная платформа технологии информационного моделирования**: Программный продукт, обеспечивающий собственно процесс создания информационной модели и получение производной технической документации.

1.4.1.18 **профиль интероперабельности**: Согласованный набор стандартов, структурированный в терминах модели интероперабельности.

1.4.1.19 **роль**: Функция, выполняемая актором в определенный момент времени.

1.4.1.20 **семантическая интероперабельность**: Способность любых взаимодействующих в процессе коммуникации информационных систем одинаковым образом понимать смысл информации, которой они обмениваются.

1.4.1.21 **сущность**: Класс информации, определяемый схожими атрибутами и ограничениями.

Примечание – Термин аналогичен термину «класс» в общепринятых языках программирования, но описывающему исключительно структуру данных (например, методы).

1.4.1.22 **техническая интероперабельность**: Способность к обмену данными между участвующими в обмене системами.

1.4.1.23 **требования к обмену информацией**: Набор информации, необходимый для обмена в процессе поддержания конкретного бизнес-требования на определенной фазе или стадии процесса.

1.4.1.24 **проприетарное программное обеспечение**: программное обеспечение, являющееся частной собственностью авторов или правообладателей и не удовлетворяющее критериям свободного ПО (наличия открытого программного кода недостаточно). Правообладатель сохраняет за собой монополию на его использование, копирование и модификацию, полностью или в существенных моментах.

1.4.2 Обозначения и сокращения

САПР - система автоматизированного проектирования;

API - интерфейс прикладного программирования;

IFC - формат основных отраслевых классов данных с открытой спецификацией для совместного использования данных в строительстве и управлении зданиями;

MVD - определение модельного вида IFC, определяющее подмножество схемы IFC, которое необходимо для удовлетворения одного или нескольких требований по обмену информацией в строительстве;

XML - расширяемый язык разметки;

EXPRESS - формат структуры обмена, использующий кодирование данных об изделии (информационной модели) открытым текстом;

STEP - стандарт обмена данными модели изделия. Позволяет описать весь жизненный цикл изделия, включая технологию изготовления и контроль качества продукции;

CamelCase - система условных обозначений. Стиль написания составных слов, при котором несколько слов пишутся слитно без пробелов, при этом каждое слово пишется с заглавной буквы;

ZIP - формат архивации файлов и сжатия данных.

2 Обеспечение интероперабельности на организационном уровне

На организационном уровне интероперабельности согласовываются бизнес-процессы и подходы к их координации. Это необходимо делать как внутри организации, так и на межорганизационном уровне. На этом уровне интероперабельности формируются бизнес-цели и соглашения о сотрудничестве между участниками инвестиционно-строительного процесса, которые хотят обмениваться информацией.

Стандартизация на уровне организационной интероперабельности лежит в области разработки и адаптации организационно-технических, нормативно-правовых и нормативно-методических документов.

В целях обеспечения организационной интероперабельности должны быть разработаны и адаптированы нормативные документы, регламентирующие стадии жизненного цикла здания или сооружения, а также объем и состав информации, передаваемой по стадиям ЖЦ; стандарты описывающие методику разработки бизнес-процессов по обмену объектно-ориентированными данными; стандарты коллективной работы над проектом с применением технологии информационного моделирования и систем управления инженерными данными; типовые формы контрактов и соглашений и типовые технические задания на разработку проекта с применением технологии информационного моделирования.

Для корректной, эффективной и стабильной работы организации необходимо иметь профиль интероперабельности, то есть согласованный набор стандартов организации как для внутреннего взаимодействия между различными отделами и сотрудниками, так и для взаимодействия со сторонними организациями.

Все возможные бизнес-процессы, а также цепочки видов деятельности в рамках этих процессов, возникающие в работе различных отделов организации, должны быть однозначно определены в соответствующих стандартах и регламентах этой организации.

Структура и состав регламентов организации остаются на усмотрение разработчика стандарта в этой организации и согласовываются с руководством.

Такие документы рекомендуется создавать с использованием действующих на территории РФ нормативно-правовых и юридически значимых документов, описывающих ответственность каждого из участников.

Стандарты должны определять следующее:

- Роли, выполняемые ими функции и зоны ответственности в рамках работы над конкретными бизнес-процессами, а также уровень их доступа к информации.
- Карту взаимодействия, устанавливающую связи между ролями в рамках работы над конкретным бизнес-процессом.
- Шаблоны, структуру и способы обмена информацией между участниками бизнес-процесса, включая различную сопутствующую информацию(название, автора и список внесенных им изменений, уникальный идентификатор).
- Карту процесса, описывающую порядок и конфигурацию необходимых работ, выполняемых в рамках определенного бизнес-процесса. Карта показывает логическую последовательность действий участников процесса и устанавливает ограничения для передаваемой информации.
- Бизнес-правила, определяющие действия и ограничения к передаваемой информации.
- Структуру бизнес-процесса (постановка задачи, определение объемов работ, определение количества и типа ресурсов, утверждение плана процесса и др.);
- План выполнения BIM-проекта, содержащий основную информацию о проекте, информационных моделях, целях и требуемых показателях проекта, а также состав участников и др.

2.1 Роли в бизнес-процессах по информационному моделированию

2.1.1 Роль: BIM-менеджер

BIM-менеджер в организации – лицо, ответственное за процесс информационного моделирования в рамках BIM-проекта. В его обязанности входит:

- исследование и анализ лучших практик;
- взаимодействие с заказчиками и партнерами;

- разработка и поддержка BIM-стандартов и регламентов организации;
- сопровождение BIM-проекта, контроль сроков выполнения работ и их соответствие плану выполнения BIM-проекта;
- распределение ролей и зон ответственности между участниками бизнес-процессов.

Данная роль имеет ключевое значение в организации. BIM-менеджер является связующим и контролирующим звеном между всеми частями BIM-проекта. Успех BIM-проекта зависит от стратегического управляющего, который должен обладать соответствующим опытом. BIM-менеджером может быть собственный или приглашенный специалист.

2.1.2 Роль: BIM-координатор организации

В небольших организациях или при работе с небольшими BIM-проектами роли BIM-менеджера и BIM-координатора может выполнять один человек. При работе с крупными проектами количество людей, задействованных в управлении и сопровождении BIM-проекта, может увеличиваться. BIM-координатор в организации выполняет управленческие функции. Данная роль выполняется на уровне проекта.

К обязанностям BIM-координатора относят:

- разработка и сопровождение плана выполнения BIM-проекта;
- регулярное проведение проверок проектной информации, разрешение споров и возникающих текущих вопросов и проблем;
- проведение координационных совещаний;
- контроль над процессом создания и распространения информации. Контроль качества передаваемой информации.

2.1.3 Роль: BIM-автор организации

BIM-автор в организации берет на себя производственную функцию и является непосредственным разработчиком компонентов информационной модели. Роль BIM-автора выполняется на уровне проекта. В проектных организациях, как правило, обязанности BIM-автора ложатся на плечи проектировщиков по

профильным разделам проекта. Такие сотрудники должны иметь навыки и опыт работы с программным обеспечением, поддерживающим технологию информационного моделирования.

При назначении сотрудника на роль BIM-автора главным критерием является не опыт работы в области информационного моделирования, а опыт проектирования. Все сотрудники, назначенные на роль BIM-автора, должны иметь соответствующие знаниями, профессиональными навыкам и опытом работы в области проектирования.

2.2 Валидация информации

2.2.1 Процесс валидации

Валидация – это соответствие результатов работы выставленным к этой работе требованиям.

При валидации определяется соответствие передаваемой информации требованиям заказчика, стандартам и регламентам организации. Устанавливается точность, полнота и качество подготовленной информации. Если в качестве передаваемой информации служит информационная модель или ее компоненты, то при валидации проверяется, насколько точно идентифицируются элементы модели и возникают ли проблемы при извлечении информации из ее элементов, а также учитываются результаты проверки на коллизии и прочие нестыковки в модели.

2.2.2 Контроль качества информации

BIM-менеджер должен разработать и внедрить систему контроля качества обмениваемой информацией информационных моделей. Данная система базируется на основе регулярных проверок и координационных совещаний. Должны быть разработаны согласованные правила, требования и процедуры, которые обеспечат эффективную работу с информацией для всех участников бизнес-процесса. Эти правила и требования должны быть четко и ясно сформулированы и не допускать двоякого понимания.

Каждый BIM-автор несет ответственность за полноту и качество информации, передаваемой им другим участникам бизнес-процесса.

2.2.3 Виды проверок

Валидация должна проводиться по одному или нескольким из следующих основных направлений:

- проверка пространственного положения и геометрических параметров информационных моделей;
- проверка данных;
- проверка на коллизии.

Проверки могут быть выполнены двумя способами. Визуально в ручном режиме - непосредственно ответственным исполнителем или автоматизированно – с помощью соответствующих программных средств.

Проверка пространственного положения и геометрических параметров должна включать:

- проверку соответствия элементов модели требованиям уровня детализации элементов модели (LOD). Выявляются избыточный и недостаточный уровень детализации;
- проверку соответствия элементов модели исходным данным и точности их построения (анализ примыканий элементов модели);
- проверку на отсутствие повторяющихся или пересекающихся элементов.

Периодичность проверок может быть любой и зависит только от внутренней организации работ, целесообразности и задается в плане выполнения BIM-проекта. Проверка данных устанавливает, насколько они систематизированы, классифицированы, структурированы и актуальны, а также их соответствие требованиям стандартов и регламентов организации, а также плану выполнения конкретного BIM-проекта.

Перед отправкой блока передаваемой информации или его размещением в обменном хранилище, он должен быть очищен он неиспользуемых элементов и ссылок.

2.2.4 Проверка на наличие коллизий.

Проверка на коллизии производится для обнаружения и устранения всех потенциальных конфликтов и недопустимых пересечений между элементами модели на этапе проектирования и предотвращения их появления на строительной площадке.

Проверка на коллизии подразумевает:

- создание сводной модели;
- определение ряда текущих тестов модели, которые необходимо провести, и требований для их успешного прохождения;
- проведение промежуточных проверок с составлением журнала выявленных проблем и назначением ответственных за их устранение.

BIM-менеджер ответственен за проведение и координацию проверок на коллизии.

2.2.5 Координационные совещания по информационному моделированию

Для анализа конфликтов и нестыковок, найденных в ходе проверок на коллизии, а также поиска решений текущих вопросов и назначения ответственных исполнителей должны проводиться координационные совещания. В совещаниях, как правило, участвуют руководители подразделений, отделов и групп, задействованных на различных стадиях выполнения BIM-проекта, а также BIM-менеджер(ы) и руководитель проекта.

Координационные совещания проводятся по мере необходимости. Способы проведения таких совещаний могут быть как очными (в зале для совещаний на базе организации), так и дистанционными (с помощью различных электронных средств коммуникации), в связи с территориальным расположением отделов, задействованных в BIM-проекте. Средства коммуникации между участниками проекта, должны обеспечивать максимальную эффективность совместной работы всех подразделений, участвующих в BIM-проекте.

На координационных совещаниях рассматриваются различные вопросы о ходе выполнения BIM-проекта, такие как:

- вопросы, связанные с конфликтами и нестыковками, выявленные в процессе проверки на коллизии. Решение возникающих проблем и назначение ответственных за их разрешение;
- текущие вопросы и проблемы, возникающие в ходе выполнения BIM-проекта на различных этапах, в том числе технические, организационные и др.;
- вопросы, не решенные после предыдущего совещания;
- вопросы коллективного взаимодействия участников BIM-проекта;
- вопросы уточнения и пересмотра объемов работ по действующему договору;
- вопросы соответствия фактического положения дел плану выполнения BIM-проекта и многие другие.

2.3 Информационные требования заказчика

Информационные требования заказчика – это документ, который закладывают в основу для разработки плана выполнения BIM-проекта. Данный документ включают в техническое задание на проектирование для формирования требований к информации, представляемой заказчику в процессе реализации BIM-проекта и по его завершении.

Состав информационных требований согласовывается с заказчиком. В документ включают следующие разделы:

- цели и задачи использования информационного моделирования в данном проекте;
- этапы работ и точки выдачи промежуточной информации;
- требования к составу, форматам и способам выдачи результатов проекта;
- требования к минимальному количеству разделов проекта;
- требования к объемам моделирования по всем стадиям и разделам проекта;
- требования к уровням детализации элементов модели по всем стадиям и разделам проекта;
- требования к системе классификации элементов модели;
- требования к регламентам проверки информационных моделей;

- требования к процедурам согласования и внесения изменений в проект;
- требования к форматам файлов обмена информацией;
- другие разделы.

2.4 Хранение и обмен данными в организации

При обмене информацией между участниками BIM-проекта требуется соблюдать установленные в регламентах форматы, а также отслеживать версионность передаваемых файлов с данными. Все передаваемые элементы информационной модели должны строго соответствовать принятой в организации системе классификации или плану выполнения BIM-проекта. Перед передачей информации ее следует актуализировать и «очистить» от неиспользуемых ссылок и прочего «информационного мусора». Также необходимо наладить систему оповещения участников BIM-проекта о появляющихся изменениях в проекте.

Требования к месту хранения информации и уровням доступа к ней, а также процедуры хранения и передачи данных в BIM-проекте должны быть отражены в плане выполнения BIM-проекта. Это позволяет обеспечить сохранность и безопасность всех данных о проекте.

Каждая организация должна проработать структуру хранения информации о проекте. Структура и имена используемых папок и файлов для каждого BIM-проекта должны соответствовать внутренним требованиям организации к хранению и обмену информацией и используемому программному обеспечению. Наименования папок и файлов, а также их расположение в хранилищах данных должны быть строго регламентированы и не иметь альтернативы для изменения.

При работе с крупными проектами информационная модель может достигать огромных размеров и работа с ней затрудняется. Поэтому такие модели необходимо делить на части. Правила разбиения, именования частей проекта и структура файлов и папок должна быть строго регламентирована и внесена в план выполнения BIM-проекта.

Информационная модель должна представлять собой идеальную основу для совместного использования данных о проекте, поэтому необходимо также строго регламентировать правила обмена данными и указать список разрешенных форматов для обмена.

2.5 Пример фрагмента регламента организации об обмене данными

Информация, представляется в электронном виде обменными массивами данных, в виде:

- файлов баз данных;
- файлов классификаторов, справочников и др.

В процессе обмена данными в организации-источнике данных формируется обменный массив данных для передачи. Передача данных осуществляется в два этапа:

- на первом этапе данные передаются в полном объеме, также передаются файлы классификаторов, справочников и др.;
- на втором этапе представляются с установленной периодичностью только изменения данных, произошедшие с момента предыдущей передачи; при изменении классификаторов, справочников они передаются в полном объеме.

По запросу на втором этапе возможна передача данных в полном объеме. Графическая информация представляется всегда в полном объеме.

Передача может осуществляться по электронной почте, курьером (на электронном носителе) или иным способом. При передаче курьером прием-передача данных производится в организации-источнике данных. Для обеспечения приема-передачи данных в организациях, участвующих в обмене данными, приказами руководителей этих организаций назначаются должностные лица, ответственные за прием-передачу данных, определяются адреса электронной почты и закрепляются за должностными лицами, ответственными за прием-передачу данных. Выписки из приказов с указанием ответственных должностных лиц и адресов электронной почты представляются в соответствующие организации, участвующие в обмене

данными, до начала обмена. По факту приема-передачи данных составляется акт, который подписывается ответственными представителями передающей и принимающей сторон в момент приема-передачи. При передаче по электронной почте организация-получатель направляет в адрес организации-источника данных уведомление о получении данных.

Ответственность за своевременную подготовку данных в установленном формате и структуре, достоверность информации, а также выполнение мероприятий по защите передаваемой информации до подписания акта приема-передачи (получения уведомления о приеме) несет организация-источник данных.

Ответственность за прием, хранение, обеспечение целостности данных, а также обеспечение защиты информации после подписания акта приема-передачи (направления уведомления о приеме данных) несет организация-получатель данных.

В случае обнаружения в представленной информации противоречивых данных организация-получатель данных направляет запрос в организацию-источник данных и организацию, которая в соответствии с нормативными документами несет ответственность за сбор и поддержание в актуальном состоянии соответствующих данных. Ответ на запрос и откорректированные данные в электронном виде (если такая коррекция произведена) направляются в соответствующую организацию-получатель данных.

Форматы данных, структуры передаваемой информации и сроки представления определяются протоколом обмена данными в электронном виде, согласованным с участниками обмена.

На последующих этапах внедрения и развития, ввода в эксплуатацию новых систем возможно внесение изменений и дополнений в Регламент и Протокол. Изменения в Протокол вносятся путем оформления дополнений к нему, согласованных с участниками обмена.

3 Обеспечение интероперабельности на семантическом уровне

Единая междисциплинарная информационная модель объекта строительства состоит из множества отдельных компонентов и их сборок, выполняющие различные функции. Для возможности работы различных групп специалистов с большим объемом разрозненной по смыслу информации необходимо создание единого языка общения. Функцию такого единого языка общения могут выполнять тезаурусы, классификаторы исходных ресурсов, процессов деятельности, полученных результатов в том случае, если они становятся общепринятыми или одобренными в качестве стандарта.

Современные программные платформы информационного моделирования оперируют объектно-ориентированной информацией, что дает возможность прямого указания принадлежности их компонентов к какому-либо классификатору.

Существующие в российской и международной практике классификационные системы различаются в первую очередь по назначению их использования. Чтобы обеспечить поддержку полного жизненного цикла объекта строительства, необходимо использовать единый классификатор ресурсов, процессов, результатов деятельности, стадий жизненного цикла и др., либо набор из нескольких таких классификаторов. Единый классификатор должен быть доступен всем участникам инвестиционно-строительного проекта на протяжении всего жизненного цикла объекта строительства, а наполняющие его компоненты – понятными по смыслу.

Одной из решаемых с помощью классификаторов задач является оценка стоимости реализации проекта. В этом случае классификаторы содержат финансовые показатели поставки ресурсов и работ по их использованию, которые могут быть напрямую соотнесены с компонентами информационной модели. Современные программные платформы информационного моделирования позволяют разбивать сложные сборки компонентов на отдельные составляющие и представлять единую модель здания в виде подробных ресурсных спецификаций. В результате объединения возможностей программных платформ и классификаторов

оказывается возможным получение оценки стоимости объекта строительства на любой стадии его жизненного цикла, в том числе, ресурсным методом.

3.1 Структура области знаний информационной модели

Знания об информационных моделях можно разделить на несколько составляющих: модели, таксономии, классификации и словари.

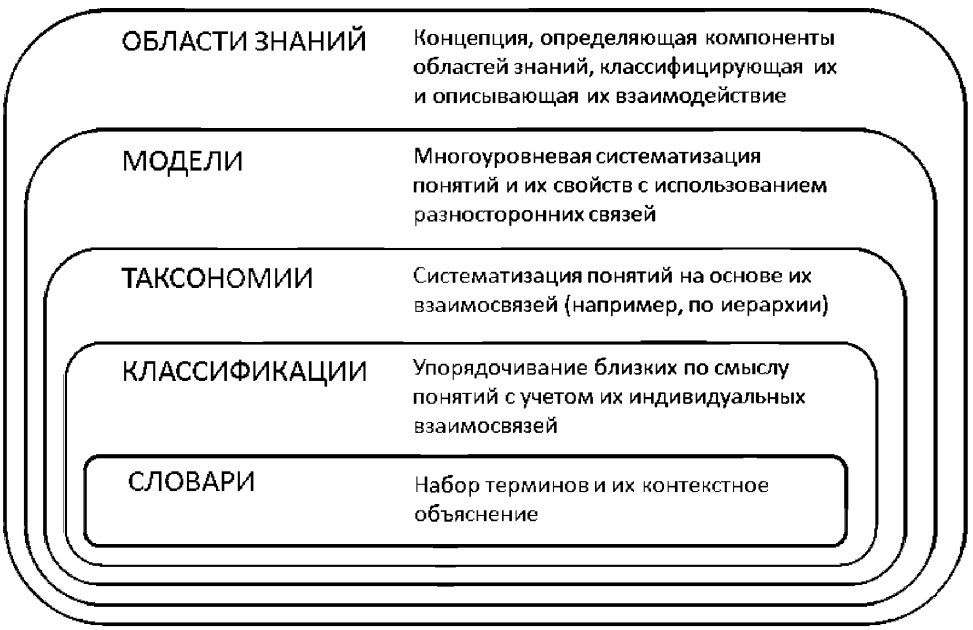


Рисунок 3.1 Структура взаимосвязанных компонентов областей знаний [4]

На рисунке 3.1 показана структура взаимосвязанных компонентов областей знаний:

- Словари представляют собой массив значений, связывающих термины друг с другом и с другими базами знаний.
- Классификация представляет собой процесс разделения множества разрозненной по смыслу информации одной предметной области на различные классы по схожим или отличным предметным признакам с целью дальнейшей систематизации.
- Таксономии представляют собой наборы категорий для систематизации сложных систем в виде иерархической структуры.

- Модели (концептуальные модели) – предметные области исследования определенных видов деятельности, включающие в себя классификации взаимосвязанных понятий, процессов, их свойств и характеристик.
- Структура области знаний показывает общую концептуальную онтологию, описывает отношения между взаимосвязанными структурами знаний информационной модели.

На рисунке 3.2 показана общая схема генерации, обобщения и применения знаний в процессе информационного моделирования.

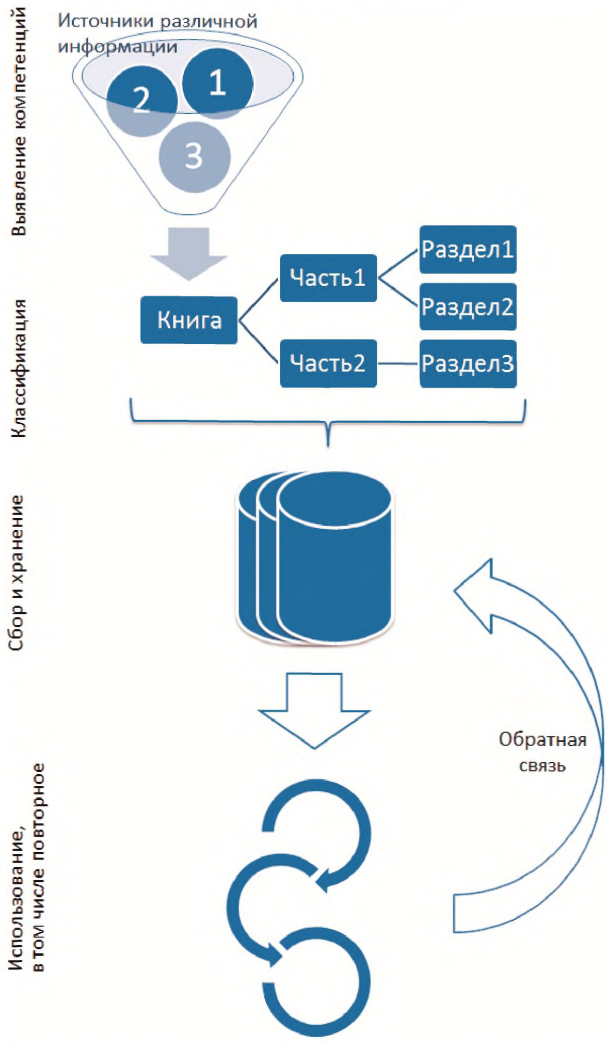


Рис 3.2 Общая схема генерации, обобщения и применения знаний в процессе информационного моделирования [4].

3.1.1 Тезаурусы

Тезаурус является словарем. Формирование тезауруса происходит за счет установления смысловых взаимосвязей между устойчивыми словами или словосочетаниями – лексическими единицами языка. Наличие семантических взаимосвязей между понятиями и их группами выгодно отличает тезаурус от толкового словаря и позволяет его успешно использовать в системах искусственного интеллекта и базах данных [5].

По смысловым внутренним связям тезаурусы подразделяются на следующие виды:

1. идеографический тезаурус – это словарь, в котором слова расположены не по алфавиту, а по их смысловой близости;
2. аналогический тезаурус – это словарь, состоящий из объединенных общим признаком или понятием групп слов – лексико-семантических групп (ЛСГ), представленных в алфавитном порядке по значению наиболее значимых по смыслу понятий.
3. ассоциативный тезаурус – это словарь, содержащий запас наиболее употребляемых слов к текущему моменту времени, представленный в форме ассоциативно-вербальной сети (АВС). В отличие от книжного издания не имеет завершенного состояния и постоянно изменяется и дополняется.

К тезаурусам, в зависимости от их назначения, могут предъявляться требования национальных или международных стандартов, например: ГОСТ 7.24-2007, ГОСТ Р 7.0.91-2015 (ИСО 25964-1:2011), ISO 25964-2:2013.

3.1.2 Классификаторы

Классификатор представляет собой структурированный перечень именованных объектов, каждому из которых присвоен уникальный идентификатор.

Для упорядочивания множества предметов с помощью классификатора сначала необходимо определиться с его назначением, затем выбрать наиболее важные признаки и выполнить группировку по одному из методов классификации.

В строительной отрасли, для классификации технико-экономической информации, целесообразно использовать иерархический и многоаспектный методы классификации. Также иерархические классификаторы называют одномерными, фасетные – двумерными.



Рисунок 3.3 Структура разновидности систем классификации

Основные цели создания систем классификации:

- выработка и использование единого системного подхода для структурирования информации;
- применение структурированной технико-экономической информации в различных отраслях;
- объединение участников инвестиционно-строительного проекта на всех стадиях жизненного цикла;
- обеспечение однозначности в понимании информации.

3.1.2.1 Иерархический метод классификации

Иерархический метод классификации предусматривает последовательное деление совокупности исследуемых предметов на ряд подмножеств. Деление предметов между каждым уровнем происходит по выбранному признаку – аспекту. В результате образуется иерархическая структура древовидного вида, с узлами в виде группировок.

Главным достоинством иерархической системы классификации является практически неограниченный объем хранимой информации и вместе с тем удобство и простота поиска как машинного, так и ручного. Емкость системы зависит от количества групп подмножеств и количества параметров, по которым конкретизируются исследуемые предметы.

К недостаткам иерархической системы классификации можно отнести невозможность выполнения выборки данных по набору нескольких произвольных признаков. Кроме того, метод имеет недостаточную гибкость структуры из-за фиксированного порядка следования признаков деления – необходимость добавления новой подчиненный группировки (на основе аспекта) приводит к переработке всего классификатора. Чтобы этого избежать, предусматриваются дополнительные резервные ёмкости.

Присвоение уникального идентификатора для позиций иерархической системы обычно выполняется с помощью последовательного метода кодирования. В этом случае в коде обозначения предмета каждого уровня хранится обозначение деления на предыдущих ступенях.

Иерархическая классификация применяется для учета и систематизации во многих отраслях, в том числе информационном моделировании объектов строительства.

3.1.2.2 Многоаспектный метод классификации

Многоаспектный метод классификации предусматривает хранение информации о предметах в виде независимых групп, каждая их которых основана на различиях классифицируемых признаков. Запросы на выборку данных выполняются по комбинациям значений признаков группировок. Метод включает в себя фасетную и дескрипторную классификации.

3.1.2.3 Фасетный метод классификации

Фасетный метод классифицирует предметы одновременно по нескольким различным признакам.



Рисунок 3.4 Иллюстрация сетки фасетной классификации

Принцип непересекаемости фасета является обязательным условием работы данного метода, когда признаки одного фасета одновременно не должны принадлежать фасетам, находящимся в рамках этого же класса.

Главное достоинство фасетного метода обусловлено возможностью добавления, удаления фасетов или внесения изменений в любой из фасетов без влияния на остальные. Такая гибкость системы позволяет создавать новые выборки информации на основе множества различных последовательностей расположения фасетов, так называемых фасетных формул, число которых ограничено лишь предельным количеством возможных сочетаний признаков.

К недостаткам фасетного метода классификации можно отнести неполное использование ёмкости структуры, сложность использования.

Присвоение уникального идентификатора для фасетной классификации обычно выполняется с помощью параллельного метода кодирования. В этом случае код обозначения предмета на каждом уровне состоит из последовательности признаков, определенных фасетной формулой.

Фасетные классификации применяются преимущественно в исследовательских целях, когда в имеющемся многообразии предметов и их свойств необходимо расставить приоритеты, выявить главное и второстепенное, определить

оптимальную структуру представления, обмена и хранения информации. Данный метод классификации хорошо реализуется с помощью программных средств [7].

3.1.2.4 Дескрипторный метод классификации

Дескрипторный (описательный) метод классификации в своей основе использует словарь устойчивых слов или словосочетаний (дескрипторов), имеющих определенные типы взаимосвязей. Благодаря взаимосвязям между синонимами ключевых слов удается выполнять расширенный предметный поиск. Данный метод используется для ведения тезаурусов, в библиотечных системах поиска информации. В результате каждый документ информационной системы получает свое описание на языке, по форме приближенном к естественному языку.

3.1.3 Кодирование позиций

Кодирование вводится с целью однозначного определения предмета, информация о котором вносится, хранится и передается через каналы связи между информационными системами и их пользователями. С помощью кодирования оптимизируется поиск информации по нужным параметрам и диапазонам их значений, увеличивается скорость передачи информации за счет сокращения ее объема, посредством обратной расшифровки создаются сводные отчеты и спецификации технико-экономической информации по произвольным пользовательским настройкам представления текстовой и графической информации, оказывается возможным выполнение анализа большого объема данных за сравнительно короткое время [8].

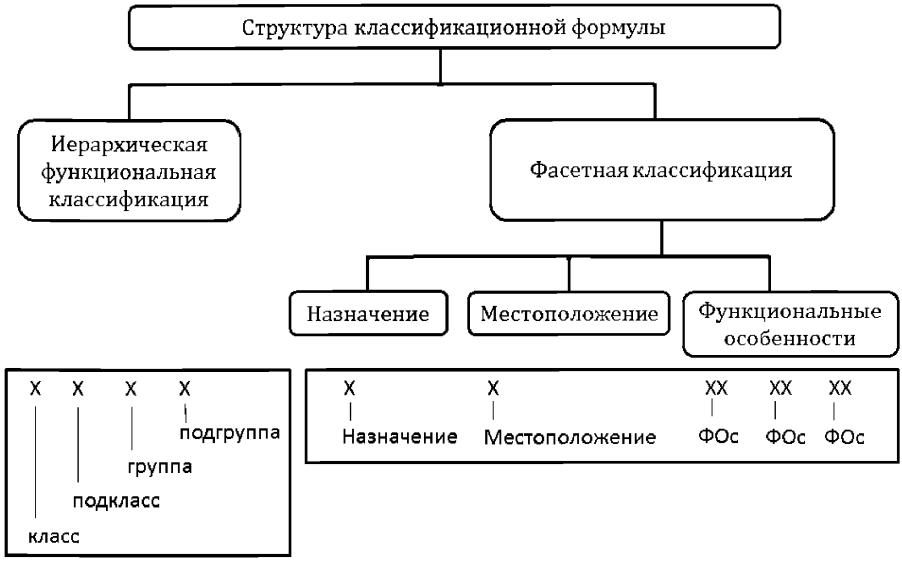


Рисунок 3.5 Пример классификационной формулы отраслевого классификатора

Коды позиций номенклатуры могут быть цифровыми, буквенными и комбинированными, дополнительно в ряде случаев используются штрихкоды и матричные коды (QR – Quick Response). Опционально классификатор может иметь функцию расчета контрольного числа, позволяющую значительно снизить вероятность ошибки при не цифровой (аналоговой, визуальной, голосовой и т.д.) передаче длинных кодов позиций, а в особых случаях даже исправить возникшую ошибку. Например, контрольным числом может быть последняя цифра суммы всех чисел кода, либо оно может быть описано с помощью метаданных.

3.2 Международные системы классификации

В настоящее время в рамках деятельности подкомитета TC59/SC13 международной организации по стандартизации ISO действуют два стандарта, регламентирующие создание систем классификации:

- ISO 12006-2:2015 «Строительство. Модель организации данных о строительных работах. Часть 2: Основы классификации информации» [9];

- ISO 12006-3:2007 «Строительство. Модель организации данных о строительных работах. Часть 3: Основы обмена объектно-ориентированной информацией» [18].

В рамках программы национальной стандартизации была выполнена разработка одноименных национальных стандартов ГОСТ Р ИСО, получивших идентичную степень соответствия названным международным стандартам.

Интересно отметить, что стандарт ISO 12006-2:2015 пришел на смену ISO 12006-2:2001 и более раннему техническому отчету ISO/TR 14177:1994 «Классификация информации в строительной отрасли», таким образом можно утверждать, что введенные впервые национальные стандарты ГОСТ Р ИСО учитывают многолетний опыт разработки, практического использования и взаимной интеграции международных классификационных систем различного назначения.

3.2.1 Требования международных стандартов

Стандарт ISO 12006-2 определяет классификационную структуру для строительной отрасли и оговаривает набор рекомендованных классификационных таблиц и их названий для различных классов строительных объектов в соответствии с их назначением. Использование стандарта нацелено на поддержку обмена информацией как внутри инвестиционно-строительного проекта (ИСП), так и между различными ИСП. Это может быть достигнуто посредством системного последовательного анализа и синтеза всего многообразия предметов строительства.

Одной из важных идей для понимания концепции стандарта ISO 12006-2 является определение термина «Строительный процесс − это процесс, который использует строительные ресурсы в целях достижения строительного результата». Таким образом «красной нитью» через весь стандарт проходит формула:

РЕСУРС (3) + ПРОЦЕСС (2) = РЕЗУЛЬТАТ (1)

Из наглядной формулы следует, что планирование инвестиционно-строительного проекта первоначально начинается с постановки целей (1) и как следствие – дальнейшего определения возможностей и способов управления

проектом (2), оценки материальных и временных ресурсов (3). В результате использования такого классификатора каждой его позиции (например, исходные ресурсы, производственные процессы, строительные результаты) присваивается уникальный идентификатор на основе одного из существующих методов кодирования. Элементы классов могут быть дополнительно классифицированы по своим характеристикам и рассматриваться как отдельный самостоятельный класс, причем классы должны быть взаимоисключающими. Необходимо, чтобы один предмет упоминался в классификационной системе только однажды. При необходимости упоминания одного предмета в разных классификациях предлагается использование фасетной классификации или ее комбинирования с иерархической.

Согласно ГОСТ Р ИСО 12006-2–2017, классификационные системы помимо распределения по специализации (классы и подклассы), могут подразделяться в соответствии с составом представленной в них информации: по отношениям вида «type-of» и «part-of». На рисунках ниже представлены примеры таких связей «род-вид», «часть-целое», «комплекс-элемент». Деление на классы выполняется от общего к частному. На рисунках 3.6 и 3.7 показана структура иерархического метода представления информации.

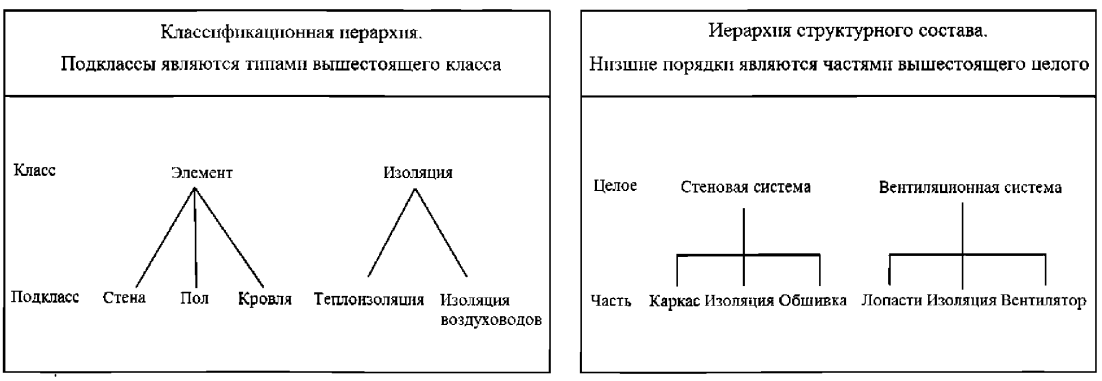


Рисунок 3.6 Иерархия классификации и иерархия структурного состава [9]

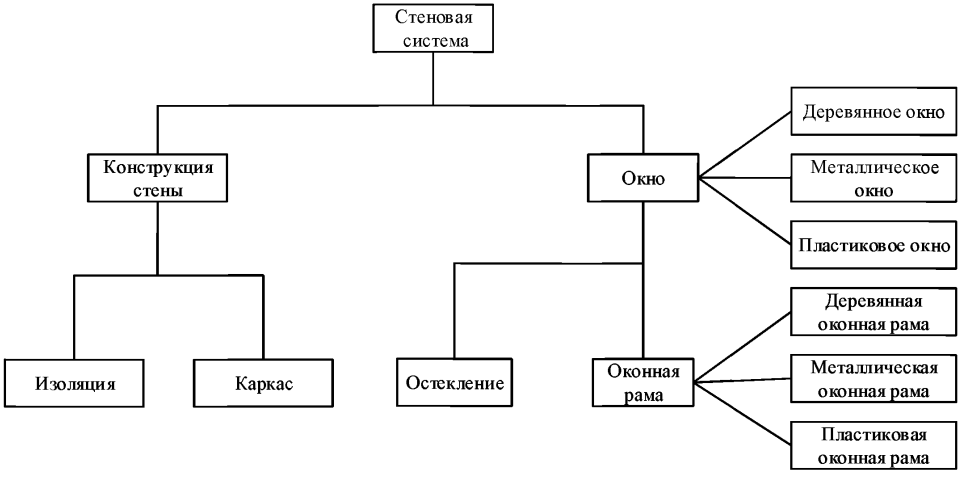


Рисунок 3.7 Иллюстрация иерархий структурного состава и классификации [9]

Примерами систем в строительстве являются: несущая система, дренажная система, кровельная система (пирог кровли), система вентиляции и кондиционирования, пожарная система, электрическая система, системы автоматики и связи, оконная система, ландшафтная система, экосистема, система менеджмента, ремонтная система, система закупок.

Одной из главных целей стандарта по классификации ISO 12006-2 является создание классификационных систем, способных обмениваться информацией между собой. Прочие классификационные структуры, основанные, например, на собственных отраслевых или корпоративных правилах и не учитывающие в полной мере положений международного стандарта, могут быть успешно гармонизированы с международными классификациями при соблюдении рекомендаций стандарта ISO 22274:2013 «Системы управления терминологией, знаниями и содержимым. Концептуальные аспекты разработки и интернационализации систем классификации». При этом необходимо разработать вспомогательный документ, в котором будут описаны соответствия содержимого систем классификаций в исходном и гармонизированном виде [3]. Представленный в табл. 3.1 вариант классификации предназначен для придания классификаторам первоначальной универсальности.

Таблица 3.1. Принципы специализации, применяемые к классам предметов [9].

Класс	Классификация
КЛАССИФИКАЦИЯ ПО РЕСУРСАМ	
Строительная информация	Содержание
Строительная продукция	Функция или форма, или материал, или любая их комбинация
Исполнитель	Дисциплина или роль, или любая их комбинация
Вспомогательный ресурс	Функция или форма, или материал, или любая их комбинация
КЛАССИФИКАЦИЯ ПО ПРОЦЕССАМ	
Менеджмент	Деятельность по управлению
Строительный процесс	Строительная деятельность, строительный процесс жизненного цикла или любая их комбинация
КЛАССИФИКАЦИЯ ПО РЕЗУЛЬТАТАМ	
Строительный комплекс	Форма, функция, активность пользователя или любая их комбинация
Строительный объект	Форма, функция, активность пользователя или любая их комбинация
Искусственно созданное пространство	Форма, функция, активность пользователя или любая их комбинация
Строительный элемент	Функция, форма, позиция или любая их комбинация
Результат работы	Трудовая деятельность и используемые ресурсы
КЛАССИФИКАЦИЯ ПО ХАРАКТЕРИСТИКАМ	
Строительная характеристика	Тип характеристики

Положения стандарта ISO 12006-2 распространяется на полный жизненный цикл инвестиционно-строительного проекта, включая составление технического задания, предпроектную подготовку, разработку документации, строительство, эксплуатацию и техническое обслуживание.

В стандарте ISO 12006-3 на базе открытой спецификации EXPRESS представлена независимая от языка информационная модель, которую можно использовать при разработке словарей для хранения и предоставления информации

о строительной деятельности. Это позволяет связать элементы структуры информационной модели с позициями систем классификации, моделями объектов и процессов.

Спецификация EXPRESS создавалась для реализации стандарта STEP (STandard for Exchange of Product model data – совокупности стандартов ISO 10303) для внедрения CALS-технологий (Continuous Acquisition and Lifecycle Support), которые в свою очередь имели цель осуществления непрерывной информационной поддержки поставок и жизненного цикла изделий на производственных предприятиях.

Модель, описанная в ISO 12006-3 [18], предлагается в качестве моста между классификационными системами, основанными на положениях ISO 12006-2 [9], и процессами информационного моделирования изделий ISO 10303-41, ISO 10303-221, ISO 15926-2, ISO 16739 [17].

3.2.2 Примеры международных классификаторов

В глобальном мировом масштабе выделяются два лидера – британская система классификации Uniclass и североамериканская Omniclass. Классификатор Uniclass1997 предлагался пользователям в бумажном виде на возмездной основе. На тот момент система Omniclass 2000 выгодно отличалась бесплатным представлением. Информацией пользовались различные категории специалистов строительной отрасли, поэтому представление Omniclass 2000 в виде набора из пятнадцати таблиц было удобным для своего времени. Вместе с тем, программные средства не могли напрямую работать с информацией из классификатора Omniclass как с единым источником информации – таблицы представлялись в двух разных форматах Excel и PDF. Данного недостатка лишился Uniclass в версии 2015 г., получив единый формат хранения данных, что очень важно для информационного моделирования в целом [3].

На рисунке 3.8 показаны взаимосвязи таблиц, входящих в классификаторы Omniclass и Uniclass.

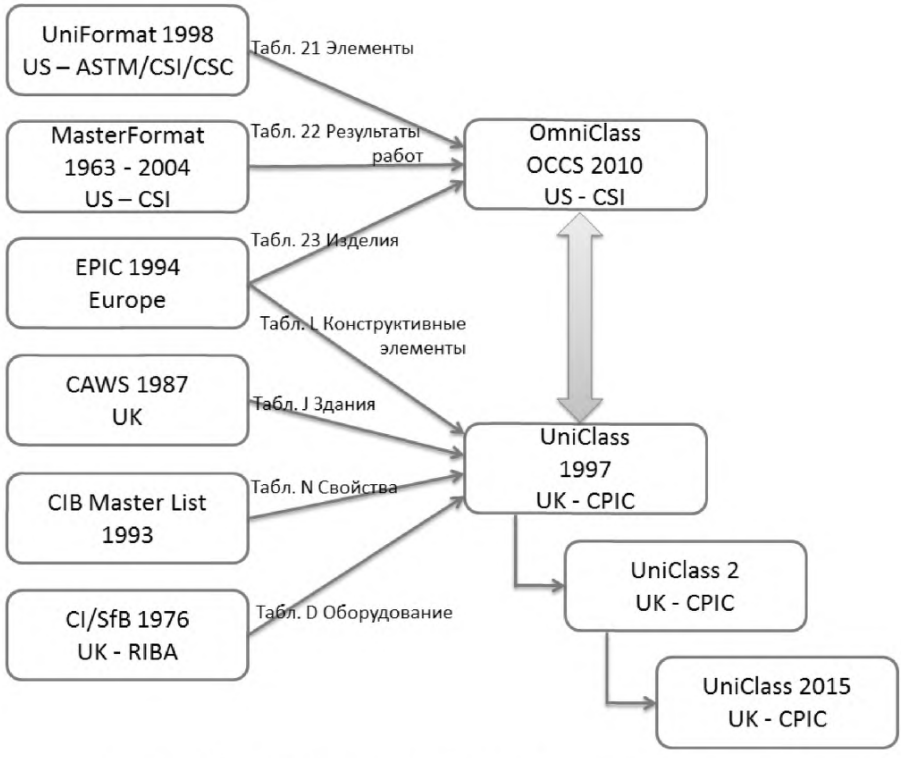


Рисунок 3.8 Таблицы международных классификаторов [10]

3.3 Отечественные системы классификации

3.3.1 Требования к отечественной структуре классификации

Исходя из вышесказанного можно выделить следующие требования к структуре отечественной классификации [3]:

- по способу представления информации система должна быть доступной пользователям в цифровом виде, удобном для ручного чтения и машинной (программной) обработки;
- по способу организационной структуры система должна:

- содержать информацию в открытом единообразном формате;
- учитывать положительный международный опыт;
- соответствовать требованиям ГОСТ Р ИСО 12006-2–2017 (ISO 12006-2:2015) [9], [19];

- по набору компетенций система должна:
  - поддерживать полный жизненный цикл объекта строительства;
  - быть многодисциплинарной;
  - поддерживать задачи в рамках строительной отрасли;
  - в определенной мере учитывать приемы работы и контекст действующих национальных классификаторов.
- по способу идентификации элементов систем предусматривать кодирование позиций одним из общепринятых методов, при необходимости использование штрих-кодов, матричных QR-кодов или радиочастотных RFID-меток (в зависимости от внешних условий).

По сферам применения классификаторы делятся на:

- международные – единая основа для взаимодействия пользователей из разных стран;
- национальные (общероссийские) – используемые для поддержания технико-экономических процессов в рамках одной страны;
- отраслевые – применяемые для решения задач определенной отрасли;
- корпоративные – используемые для частных задач отдельного предприятия.

3.3.2 Российский классификатор национального уровня

В российском классификаторе строительных ресурсов (КСР) применен последовательный метод кодирования иерархической структуры. КСР построен на основе синхронизации со Статистической классификацией продукции по видам деятельности в Европейском экономическом сообществе (КПЕС 2008) – Statistical Classification of Products by Activity in the European Economic Community, 2008

version (CPA 2008) и общероссийским классификатором продукции по видам экономической деятельности (ОКПД2) ОК 034-2014 (КПЕС 2008) путем привязки к кодам ОКПД2 (КПЕС 2008) (до девяти знаков включительно). В КСР применен иерархический принцип деления от общего к частному с использованием цифрового десятичного кода [11]. В табл. 3.2 и 3.3 приведены примеры обозначения позиции «Двутавр обычный, марка стали ВСт3кп2-1, № 14» последовательными методом кодирования:

Таблица 3.2

Ступень классификации КПЕС 2008	Классификационные группировки	Примечание
24	Металлы основные	Книга КПЕС 2008
24.10	Железо, чугун, сталь и ферросплавы	Часть КПЕС 2008
24.10.71	Профили незамкнутые горячекатаные, горячетянутые или экструдированные, без дополнительной обработки, из нелегированных сталей	Раздел КПЕС 2008

Таблица 3.3

Ступень классификации КСР 2016	Классификационные группировки	Примечание
08	Изделия металлические, металлопрокат, канаты	Книга КСР 2016
08.3	Металлопрокат	Часть КСР 2016
08.3.01	Двутавры	Раздел КСР 2016
08.3.01.01	Обычные	Группа КСР 2016
08.3.01.01-0001	№14	Позиция КСР 2016

Итоговый номер в классификаторе КСР 2016 для позиции «Двутавр обычный, марка стали ВСт3кп2-1, № 14» имеет вид: 24.10.71.08.3.01.01-0001, т.

3.4 Синтаксическая и семантическая проверка информационных моделей

На практике возникают случаи, когда проект изменяется множество раз, начиная со стадии предпроекта и заканчивая его изготовлением. Для предотвращения ошибок при многократной передаче данных через формат IFC требуется выполнять синтаксические и семантические проверки. Если правила синтаксических проверок можно описать сравнительно легко – заданием именованным переменным соответствия типов данных с проверкой их целостности, то выполнение семантических проверок Определения модельного вида (MVD) оказывается более трудоемким.

3.5 Семантическое описание информационных моделей

Примером правильного семантического описания является сущность IfcRealAssociatesMaterial (Принадлежность материалу), имеющая взаимосвязи с другими обязательными сущностями схемы IFC – IfcBuildingElement (Строительный элемент) и IfcMaterial (Материал).

Ручная проверка

В общем случае схему набора проверок файла IFC можно представить в виде изображенной на рисунке 3.9 блок-схемы. В первом блоке выполняются синтаксические проверки кода, во втором – семантические, третий блок предназначен для дополнительных пользовательских проверок по проекту, в том числе, на соответствие количественных показателей модели определенным строительным нормам и правилам.



Рисунок 3.9 Блок-схема проверки IFC-модели [2]

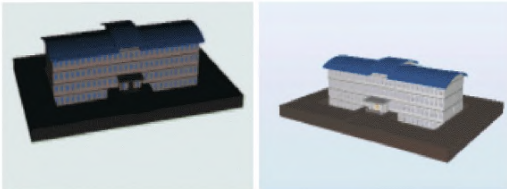
Такие проверки способствуют повышению качества передаваемых IFC-данных за счет уменьшения количества неточностей их интерпретации.

Настраиваемая проверка

Одним из сервисов для выполнения пользовательских семантических и синтаксических проверок IFC-файла является IFC WebServer (IfcSvr), доступный по адресу [13]. IfcSvr позволяет бесплатно создать личную учетную запись для загрузки IFC-файлов и использовать облачный Viewer для просмотра и навигации по модели.

Для загруженного на сервер IFC-файла могут формироваться сводные отчеты с представлением структуры сущностей проекта в виде дерева или списка.

Для анализа и исследования IFC-файлов IfcSvr дает пользователям возможности как для простой фильтрации классов по заданным условиям, так и расширенные возможности для выполнения собственных сценариев. По своему объему и сложности вычислений сценарии могут быть как простыми (например, для выполнения расчетов геометрических параметров объектов модели), так и очень сложными (например, представлять собой плагин для вывода данных в формат COBie или нахождение различий между двумя версиями IFC-файла – см. рисунок 3.10).



Walls difference

ID	Wall GUID	Wall name
1	'054SZ4_9H5tRVfRLsmTmFu'	'Wand-005'
2	'0tNy_iP2b0Ge3YC49v0CwY'	'Wand-003'
3	'19sGy1PrHF_g3O1l2OlGZg'	'Wand-033'
4	'1w7i3VLnHBxwkBecZDpTr$'	'Wand-032'
5	'1YglXaVGz84vHhbAKNJ_5N'	'Wand-035'
6	'2XOyYx4Of0QhwaNJbJRTS3'	'Wand-034'
7	'0Tam14iFj2YAFol6iPl1z$'	'Wand-037'
8	'2zm79Baqz7hvGifG6oryf9'	'Wand-031'
9	'3FVy1CHMLCJQF_C$0WmrLQ'	'Wand-036'
10	'1ZkrmZCfvDUe2ucUcWc7PF'	'Wand-030'

Zones difference

ID	Zone GUID	Zone name
1	'0fbFKlSePEw8FYtmZ$cYnZ'	'Raumtemperatur-20-Grad'
2	'37O2EFTUT9PfWFcQVB9Sqi'	'Raumtemperatur-15-Grad'
3	'2VsNrXB95DbPJiq2H60Z4C'	'Raumtemperatur-10-Grad'

Рисунок 3.10 Результаты сценария поиска различий между двумя версиями IFC-файла

Кроме анализа загруженных IFC-файлов IfcSvr позволяет выполнить интерактивный анализ и найти различия в версиях и релизах самого формата IFC на уровне сравнения сущностей и их атрибутов. Так, на рисунке 3.11 показаны результаты сравнения атрибутов сущности IfcActionRequest для версий IFC 2x3_TC1 и IFC 4 RC4.

ID	Class Name	IFC2X3_TC1 653 Class / 1817 Attribute	IFC4RC4 764 Class / 2101 Attribute
2	Ifc2DCompositeCurve	segments selfIntersect	
3	IfcActionRequest	objectType globalId ownerHistory name description requestID	objectType globalId ownerHistory name description identification predefinedType status longDescription requestID

Рисунок 3.11 Сравнение атрибутов сущности IfcActionRequest для двух версий IFC формата.

Другим сервисом для выполнения семантических проверок является программа ifcDoc с открытым исходным кодом. В приложение ifcDoc интегрируются описания различных Определений модельного вида (MVD), на соответствие которым можно проверять пользовательские IFC структуры данных, при этом используются операторы наподобие UNIQUE или WHERE. Проверки выполняются на основе концепции – набора шаблонов проверочных критериев Определения модельного вида (MVD) с учетом версии формата IFC.

Чтобы различные участники BIM-проектирования могли проверить правильность использования конкретного Определения модельного вида (MVD) для них необходимо обеспечить доступ к общей IFC структуре данных для возможности внедрения собственных различных проверочных критериев.

Интерфейс ifcDoc, показанный на рисунке 3.12, предоставляет пользователям возможность выполнять проверку структуры проекта на предмет правильности взаимосвязей и ссылок, правильности описания значений и свойств атрибутов – пользователь может задать шаблон для любой из конкретных сущностей проекта IFC с использованием различных критериев.

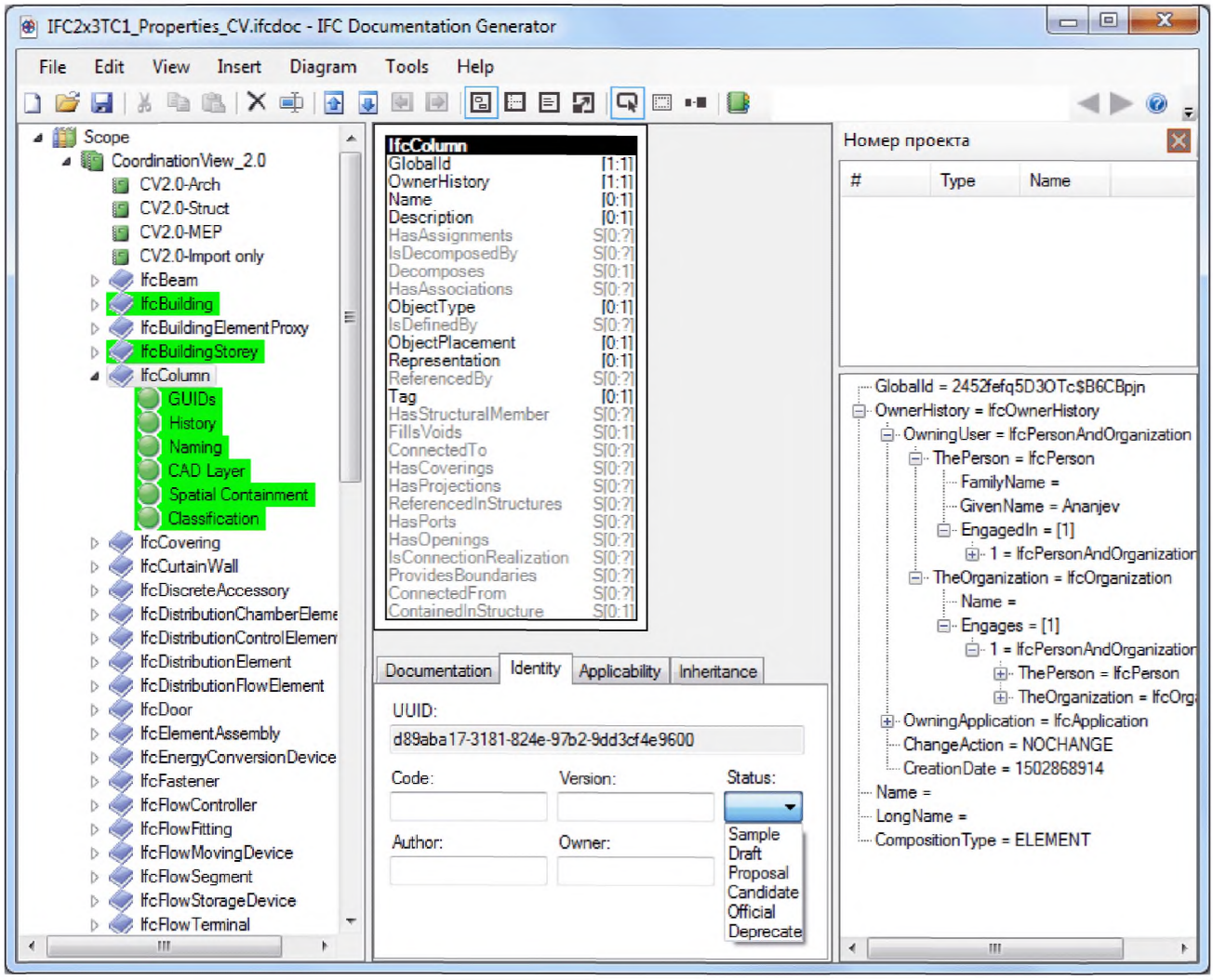


Рисунок 3.12 Пользовательский интерфейс программы IfcDoc

Программа IfcDoc разрабатывается компаниями Digital Alchemy & Digital Building Laboratory, Georgia Tech (США) и buildingSMART International. Приложение ifcDoc доступно для скачивания с сайта BuildingSmart [14].

Автоматическая проверка

Подобные проверки реализованы во многих программных продуктах, таких как Express Engine EXPRESSO, JDSAI™, Solibri Model Checker®, EXPRESS Data Manager™, IfcDoc, IFC Server ActiveX Component, однако в этом случае сценарии проверок скрыты от пользователя.

3.6 Выводы

Классификатор строительных ресурсов (КСР) содержит в себе информацию о материальных строительных ресурсах, тем самым оправдывая свое название. Однако для поддержки технологии информационного моделирования на протяжении всего жизненного цикла необходимо оперировать более широким набором информации об объекте строительства, например: компетенциями и ролями участников строительства, средствами управления строительством, видами производственной деятельности, стадиями жизненного цикла и пр.

Созданные на основе требований ГОСТ Р ИСО 12006-2–2017 [9] системы классификации смогут обеспечить обмен данными с другими классификационными системами, в том числе международными Omniclass (США, Канада), Uniclass 2015 (Великобритания) и другими. Например, в настоящее время классификатор строительных ресурсов может обмениваться информацией с двумя однотипными по смыслу таблицами классификатора Omniclass: таблицей 23 «Продукты» и таблицей 41 «Материалы», что служит поводом для дальнейшего развития КСР по части увеличения разновидностей таблиц.

4 Обеспечение интероперабельности на программно-техническом уровне

4.1 Методические положения по экспорту и импорту данных информационной модели

Техническая интероперабельность означает способность к обмену данными между участвующими в обмене системами.

Применительно к технологиям информационного моделирования технической интероперабельности лежит, главным образом, в стандартизации форматов обмена данными между различными программными комплексами.

Низкая степень технической интероперабельности, обусловленная отсутствием единых стандартов обмена и закрытостью форматов данных программного обеспечения CAD/BIM, приводит к необходимости предпринимать затратные действия для повторного ввода уже закодированных данных, а также снижает эффективность рабочих процессов по причине утраты части данных, их несовместимости или искажения при передаче.

Информационное моделирование зданий и сооружений (BIM), подразумевает интеграцию в работу над проектом всех внутренних заинтересованных лиц, а также внешних участников проекта, и унификацию их коммуникаций на основе интеллектуальной трехмерной модели как основы, служащей средством достижения данной цели. Реализация данной концепции предопределяет необходимость обеспечения взаимосвязанного функционирования различных программных продуктов и сопутствующего обмена данными.

Таким образом, можно дать определение интероперабельности как способности управлять программными продуктами и передавать проектные данные при проектировании, строительстве, эксплуатации и в бизнес-процессах как внутри организаций, так и между организациями, ведущими совместную работу.

4.2 Виды технической интероперабельности

В общем случае под обменом данными между различными программными системами понимается подготовка и упорядоченное представление информации (геометрической и атрибутивной), отвечающей требованиям к формату и степени достоверности, на одной из нескольких предварительно установленных стадий инвестиционно-строительного проекта.

Сложность обмена данными вызвана как субъективными, так и объективными причинами. Разработчики ПО, как правило, строят архитектуру собственных систем на различных принципах, поэтому перенести данные одной системы в другую является достаточно непростой задачей.

На практике при обмене данными между программными комплексами различных производителей применяются различные способы (и их комбинации) передачи данных.

На всех стадиях жизненного цикла здания или сооружения техническая интероперабельность может быть отнесена к одному из трех видов:

- интероперабельность между программными решениями одного производителя (вендора) на основе внутренних форматов и прямых API-интерфейсов (интерфейс прикладного программирования, от англ. application programming interface);
- интероперабельность между программными решениями различных производителей (вендоров) на основе проприетарных форматов и прямых API-интерфейсов;
- интероперабельность между программными решениями различных производителей (вендоров) на основе открытых стандартов и форматов данных.

Программы-конвертеры для обмена данными

Одним из способов автоматизации процесса проектирования является использование плагинов для конвертации данных в формат принимающего программного комплекса. Такие плагины разработаны для решения задач по обмену

данными на основе предусмотренного разработчиком программного продукта интерфейса прикладного программирования API.

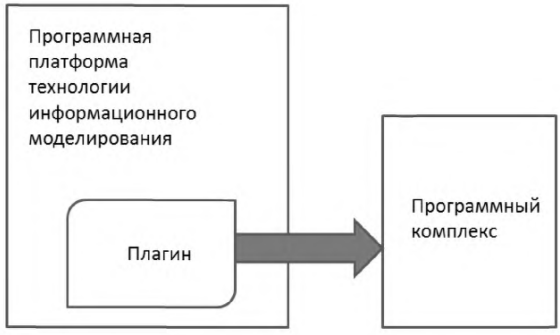


Рисунок 4.1 - Пример реализации схемы передачи данных из информационной модели в программный комплекс

Например, для расчета конструкции на прочность передаются необходимые данные: геометрические, физические и логические параметры модели. В программных платформах для информационного моделирования имеются встроенные функции API для разработки плагинов. На схеме (Рисунок 4.1) показан процесс передачи данных между программной платформой и программным комплексом. Плагин получает из формирует необходимые параметры информационной модели и записывает их в текстовый файл.

При конвертации такого файла записанная информация преобразуются в формат данных принимающего программного комплекса. Таким образом при использовании такого способа передачи данных информационной модели минимизируются ошибки, связанные с неверной интерпретацией данных. Однако программное обеспечение для конвертации данных должна быть модифицирована при выпуске новых версий принимающего и передающего программного обеспечения.

Серьезный вклад в области разработки открытых стандартов и спецификаций по обеспечению интероперабельности в строительной отрасли вносит некоммерческая международная организация buildingSMART (подробнее об организации в Приложении 4).

В Российской Федерации в качестве национального стандарта разработан ГОСТ Р «Моделирование информационное в строительстве. Отраслевые базовые классы (IFC) для обмена информацией на всех этапах жизненного цикла. Основные положения». Тем самым признается необходимость работать с использованием данного формата как одного из наиболее эффективных и перспективных форматов данных с открытой спецификацией.

Стандарт IFC предназначен для использования не проектировщиками, архитекторами, строителями или специалистами эксплуатирующих организаций, а разработчиками программных приложений и разного рода информационных систем, то есть, по сути, специалистами в области информационных технологий. Проектировщики и архитекторы применяют стандарт IFC только через интерфейс конкретного программного продукта для передачи информации и из одной автоматизированной системы (программной платформы, программного комплекса) в другую. Формат предусматривает хранение и передачу данных, а не создание информационной модели. Моделирование производится средствами программных продуктов.



Рисунок 4.2 – Схема обмена данными на основе IFC на всех стадиях жизненного цикла зданий и сооружений [12]

В то же время необходимо отметить, что в настоящее время использование стандарта IFC не гарантирует, что передача геометрической и пространственной информации будет выполнена без ошибок.

Проблема передачи данных через формат IFC также связана с качеством реализации программ-трансляторов из проприетарных форматов, которые разрабатывают производители программных средств. В связи с выходом новых версий собственного программного обеспечения, а также с выходом новых релизов формата IFC трансляторы подлежат периодической переработке. Все это в совокупности с недостатками самого формата IFC является дополнительным фактором неопределенности и неточности при передаче данных.

Ввиду перспектив развития информационного моделирования и дальнейшему стремлению международного сообщества инженеров при переходе от второго к третьему BIM-уровню (BIM Level 3) [20], поддержка отечественными разработчиками формата IFC как открытого стандарта обмена данными должна стать приоритетным направлением в развитии их программных решений. В настоящее время этот формат поддерживает довольно малое количество отечественных разработчиков. Отечественным разработчикам в области архитектурно-строительных САПР (BIM/CAD/CAE) необходимо использовать сегодняшнюю ситуацию для совершения следующего шага в развитии программного обеспечения, интегрируемого с зарубежными разработками технологии информационного моделирования. Разработчики должны реализовывать поддержку IFC для обеспечения возможности продвигать собственные решения на зарубежных рынках.

Для обеспечения интероперабельности также важно и стремление к двустороннему (многостороннему) сотрудничеству в этой области компаний-производителей программного обеспечения.

4.3 Особенности применения IFC

В данном разделе рассматриваются особенности применния формата IFC для различных целей [15].

4.3.1 IFC является обменным форматом

Это не формат конкретного программного обеспечения, как форматы с открытым исходным кодом (DOCX, JPEG, XVID и т.д.) или проприетарные форматы (DWG, PDF и т.д.). IFC разработан как формат, который используется как промежуточный при обмене данными между другими форматами программных средств. Нет большой разницы между форматом обмена и форматом программного обеспечения, оба являются просто структурированными данными. Разница состоит в назначении применения. Формат программного обеспечения структурирован в соответствии с функциональностью программного обеспечения, формат обмена структурирован для передачи данных, а не для поддержки функциональности. Например, формат IFC не поддерживает параметрическую геометрию, потому что это не требуется для целей применения формата IFC, но эта функциональность невероятно полезна для программного обеспечения при создании информационной модели BIM.

При импорте данных из файлов в формате IFC программное обеспечение преобразует данные в собственный формат. Программное обеспечение не имеет возможности фактически редактировать файл IFC. Но это означает, что есть две точки возможного создания ошибки – когда данные в формате IFC экспортируется и когда они импортируется. В обычных форматах есть только одна точка возможного создания ошибки – при импорте при использовании формата программного обеспечения или при экспорте при использовании другого программного обеспечения.

Поэтому, хотя обменный формат, такой как IFC, может быть теоретически более эффективным в отрасли, практика применения сильно зависит от качества реализации процесса обмена программными средствами. Чтобы получить конкретный результат, проще (и быстрее) написать один набор кода в одном программном обеспечении, чем писать два набора кода в двух отдельных программных средствах.

4.3.2 Функциональность формата IFC

Формат IFC не отражает всю функциональность программного обеспечения для создания информационных моделей. Например, он содержит данные о размерах, но не содержит информации, какие геометрические объекты контролируют эти размеры. Поэтому он не может передавать параметрические объекты с поддержкой динамического изменения их рабочих параметров. Эффективно импорт IFC создает статические объекты, которые больше не редактируются. Это создает проблемы, которые должны быть преодолены с развитием информационного моделирования.

Формат IFC не может содержать сложные данные, которые могут быть созданы с помощью программного обеспечения. Диапазон данных может со временем расширяться по мере улучшения формата IFC, но он всегда будет подмножеством всего, что имеется в программном обеспечении для создания информационной модели. Поэтому возникает потребность в создании прямых средств экспорта и импорта для передачи этих более сложных данных без участия формата IFC. Это не ставит под сомнение необходимость использования формата IFC, но отражает тот факт, что формат нуждается в непрерывном совершенствовании, что и происходит с появлением новых версий формата IFC.

4.3.3 Формат IFC и создание компонентов

Можно создать файл IFC здания, содержащего компоненты, но IFC не предполагает создание файла только для одного компонента. Файл IFC является фактически зданием с одним компонентом в нем. Использование открытого формата для распространения компонентов было бы очень полезно. Но формат IFC просто не имеет возможности содержать параметрические свойства, которые мы ожидаем от компонентов информационной модели (см. п. 4.3.1 выше). Существует ряд инициатив по созданию стандартов компонентов BIM, таких как Национальная библиотека BIM в Великобритании NBS (UK NBS National BIM Library). Но в NBS Великобритании пришлось использовать файлы проприетарного формата из-за отсутствия необходимой функциональности файлов IFC. Тем не менее, возможно появится способ для создания файловой схемы компонента IFC.

4.3.4 Файл в формате IFC - это архивный файл

Регулярно обновляется стандарт формата IFC, регулярно обновляются программные средства, изменяются их функциональность и формат данных. При обновлении стандарт IFC претерпевает изменения: некоторые классы удаляются, другие изменяются и добавляются новые. Поэтому нет гарантии, что все сохраненные файлы всегда будут корректно читаться программными средствами в будущем.

Формат IFC не является программным форматом какого-либо конкретного программного обеспечения. Это формат обмена, который требует от других программных средств реализации возможностей по его созданию и чтению. Чтобы поддерживать возможность чтения более старых версий IFC, необходимо полагаться на сертифицированное проприетарное программное обеспечение, поддерживающее преемственность своих версий форматов. Если версии IFC будет обновляться редко, то проприетарное программное обеспечение не должно будет поддерживать очень много версий IFC-конвертеров, а если версии IFC будет обновляться часто, то получится ситуация, когда проприетарное программное обеспечение будет не успевать за изменениями стандарта IFC. Все это надо учитывать при использовании формата IFC для архивного хранения цифровых данных.

4.3.5 Использование IFC для завершенных проектов

IFC неэффективен для использования в качестве ресурса для будущих изменений в здании. Созданный с целью обмена формат IFC неэффективно редактировать напрямую. Значительная часть данных, которая используется программным обеспечением при создании и редактировании информационной модели при обмене в формате IFC будет потеряна. Таким образом, формат IFC в лучшем случае может использоваться только как статический фон (подоснова) в программном обеспечении для доработки частей модели путем их редактирования и ограниченным удалением частей. При импорте статической информационной модели IFC возникает проблема достоверности передачи информации, когда файл IFC импортируется в программное обеспечение для разработки и редактирования информационной модели. Без тщательной проверки каждого элемента нельзя быть

уверенным в полной репликации всего объема данных из модели IFC. Затем, когда были внесены изменения в информационную модель с помощью программного обеспечения (например, программной платформы), экспорт обратно в IFC также проблематичен. Если была изменена только часть здания, остается вопрос ее интеграции в исходную модель IFC. Если все здание будет экспортировано обратно, заменив оригинальную модель IFC, то появится задержка во времени, которая означает различия в содержании данных, или проблемы, возникающие, если модель IFC должна использоваться и в других программных комплексах в процессе проектирования.

4.3.6 Сложности в понимании стандарта IFC

Описание спецификации IFC имеет очень большой объем и в первую очередь предназначено для программистов. IFC является стандартом программного обеспечения, а не стандартом для описания правил действий человека. Теоретически, пользователи должны быть защищены от знания внутренней работы IFC программным обеспечением, которым они пользуются. Но в действительности пользователи вынуждены вникать в IFC, потому что программное обеспечение автоматически не обеспечивает всего, что требуется для выполнения их работы.

4.3.7 Формат IFC широко используется

Альтернативой использованию формата IFC для обмена данными является использование специальных инструментов программного обеспечения для обмена данными:

- программные решения одного производителя (вендора) на основе внутренних форматов и прямых API-интерфейсов (интерфейс прикладного программирования);
- программные решения различных производителей (вендоров) на основе проприетарных форматов и прямых API-интерфейсов.

Но этот путь имеет свои недостатки. Поэтому несмотря на ограничения формата IFC он широко используется во всем мире.

4.3.8 Расчеты

Как правило, информационная модель, используемая для определенных целей, не требует всей модели. Конструктивный расчет требует только конструктивных компонентов. Программное обеспечение для расчета может использовать только часть информационной модели здания. Поскольку файлы IFC являются файлами открытого формата, они могут быть разбиты на части без применения программного обеспечения, которым они были созданы. Это можно сделать с помощью другого программного обеспечения для редактирования.

Поскольку IFC обычно проще, чем форматы проприетарного программного обеспечения для разработки и редактирования информационных моделей, то программному обеспечению для расчета требуется меньший объем данных для получения требуемой информации, вследствие чего упрощается процесс обработки этих данных перед импортом. Эта проблема становится все более важной, поскольку для анализа используются облачные вычисления.

Статической модели IFC оказывается достаточно для выполнения анализа с помощью аналитического программного обеспечения. Однако если вы хотите изменить модель на основе результатов анализа, то IFC не позволит этого сделать.

4.3.9 Эксплуатация готовых объектов строительства

Система управления объектами не нуждается в возможности легко изменять геометрию здания. Геометрия в основном используется в качестве навигационного инструмента для данных, хранящихся в системе. На самом деле лучше, когда нельзя случайно перемещать фиксированные элементы (что может произойти при использовании соответствующего программного обеспечения). Не требуется применять излишне сложный формат файла, который позволяет сделать больше, чем требуется в этом случае. Поэтому использование IFC при эксплуатации готовых объектов строительства имеет смысл. Это простой формат, в котором геометрия объекта не может быть легко изменена. Поэтому в этом случае применяемое программное обеспечение может использовать IFC, поскольку есть возможность импортировать данные из нескольких источников. Однако на практике следует учитывать качество реализации создания данных в формате IFC различными

источниками. Конечно, если нужно будет внести изменения в геометрию здания, например, добавить дверь или сменить стену, то нужно будет сделать это с помощью другого программного обеспечения и попытаться интегрировать это изменение обратно в модель IFC.

Формат IFC не является форматом для создания модели с помощью программного обеспечения и в его нынешней форме имеет ограничения по отношению к функциям внесения изменений в существующее здание. Эффективность его использования зависит от тщательности разработки модели для экспорта в формат IFC. Например, можно разбить модель на связанные подмодели. Если предполагается использовать формат IFC, то процесс обновления геометрии здания следует сделать до передачи данных в систему управления объектами.

4.3.10 Обнаружение коллизий

Специализированное программное обеспечение для обнаружения коллизий работает путем импорта статических моделей, при этом отсутствует необходимость в наличии функций по изменению импортированной геометрии. Поэтому этот тип программного обеспечения широко использует данные в формате IFC.

Идеальным было бы иметь возможность изменять геометрию модели и исправлять обнаруженные коллизии. Но это зависит от применяемого программного обеспечения и его возможностей. Поэтому в настоящее время для обнаружения коллизий формат IFC подходит оптимально.

4.3.11 Приемы минимизации потерь и искажения данных при использовании формата IFC для передачи данных из платформы в платформу и из платформы в программный комплекс

Если для передачи данных применяется Определение модельного вида «Скоординированный Вид» (Model View definition «Coordination View») (геометрия наружной поверхности), который задает контурное представление, то каждый элемент экспортируется с геометрией BREP (Boundary REPresentation – контурное представление). При такой упрощенной контурной передаче элемента в принимающем ПК могут быть потеряны параметры элемента, а элементы BREP из

импортированного файла IFC могут преобразовываться в не редактируемые элементы.

Этот факт отмечен многими разработчиками BIM-приложений. Так, на сайте gost.e-ecolog.ru отмечается, что при импорте IFC-модели некоторые элементы, определенные как IfcWall, IfcSlab и т.д., импортируются как объекты, а не как стены, перекрытия и т.д. Причина заключается в их упрощенном геометрическом BREP-представлении, задающем «наружную поверхность» или «контурное представление». Это означает, что кроме геометрической формы элемент не включает никакой другой информации, например, нет его привязки к осевым линиям. В этом случае программная платформа информационного моделирования интерпретирует такие элементы как объекты (с подтипом Wall, Slab и т.д.). Отметим, что в интерфейсе установки параметров IFC в программной платформе или в программе просмотра IFC эти элементы видны как стены или перекрытия, хотя на самом деле они являются объектами и требуют дополнительного определения свойств.

Таким образом, при проектировании конструкций со сложной геометрией в случае обмена данными по стандарту IFC приходится сталкиваться с потерей информации.

4.3.12 Систематизация ошибок и несоответствий при BIM-проектировании

Можно выделить три основные группы системных ошибок в процессе проектирования применением информационного моделирования:

1. Ошибки при обмене данными. При экспорте данных теряются иерархия элементов и их взаимосвязь, настроенная в специализированном программном комплексе (например, установка привязок элементов или параметрические связи). При импорте из информационной модели не все данные могут быть распознаны принимающим приложением (например, экспорт BREP-геометрии, или экспорт нестандартных элементов информационой модели, таких, например, как эскалаторы), рисунок 4.6.

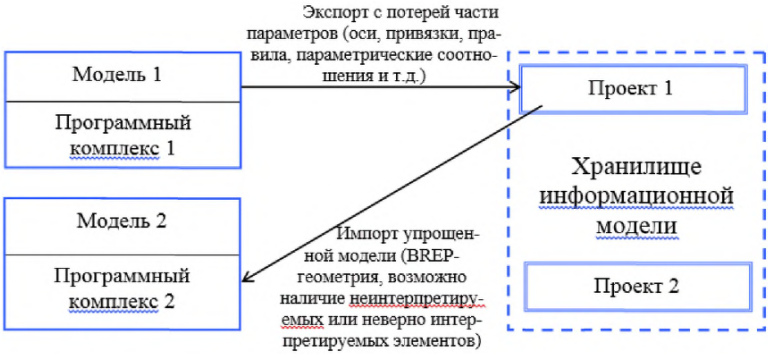


Рисунок 4.6 Экспорт и импорт модели раздела информационной модели

2. Коллизии между смежными разделами при совместном проектировании (например, наложение элементов водоснабжения или отопления на оконные проемы). Для просмотра и визуальной проверки сводной цифровой модели необходимо приложение-просмотровщик. При обнаружении коллизии участники проекта обмениваются аннотациями и исправляют проекты по своим разделам, в том числе с использованием открытого формата BIM-взаимодействия BCF (BIM Collaboration Format). Однако в сложных проектах часть конфликтных ситуаций может быть не распознана визуально. Автоматическая проверка на коллизии проводится специализированными приложениями, рисунок 4.7.

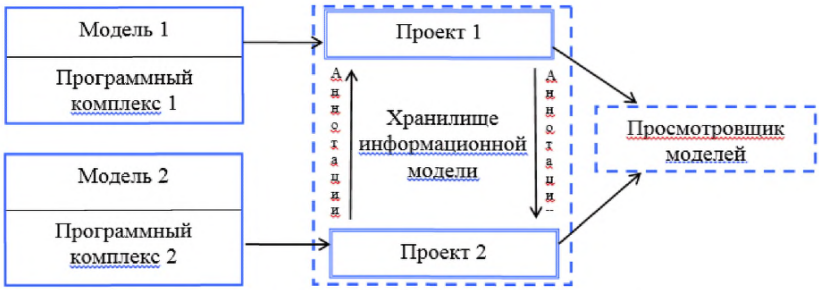


Рисунок 4.7 Решение проблемы коллизий

3. Несогласованность изменений в рамках модели одного раздела проекта. Все объекты модели взаимосвязаны, поэтому изменение одного объекта может привести к конфликту связанных с ним объектов по другим разделам проекта.

4.3.13 Создание и проверка данных при использовании формата IFC

Для управления IFC-данными (просмотра, экспорта, импорта, редактирования, изменения места в иерархии и т.д.) в программных платформах информационного моделирования и в программных комплексах должны быть предусмотрены специальные возможности.

После того, как модель здания экспортирована, следует проверить файлы до того, как они будут переданы, и убедиться, что все объекты были переданы корректно (см. п. 3.5. по тексту выше).

Всякий раз, когда при обмене данными или при создании и импорте IFC файлов, данные преобразуются в другой формат. Это всегда изменяет данные, хотя и незначительно. Из-за различных программных спецификаций и описаний элементов, вряд ли можно конвертировать данные точь-в-точь в другой формат, независимо от целевого формата или исходного формата.

Можно сделать вывод, что реимпорт созданных файлов или импорт получаемых файлов не дает надежного контроля, потому что файлы не содержат оригинальных данных.

Если нужно проверить файл IFC в первоначальном формате перед его преобразованием или передачей, потребуется программа, которая может напрямую читать этот формат (в котором модель была создана).

Сделать это можно двумя различными способами:

- можно использовать средство просмотра файлов IFC, которое позволит открыть и проверить файлы в их исходном формате;
- можно выгрузить данные на другой сервер данных. С помощью соответствующего приложения можно совместно показать большое число различных форматов и частично даже интерактивно проанализировать их.

Некоторые производители BIM-платформ разрабатывают свои серверные приложения для совместной работы. В этом случае, модель в проприетарном формате загружается напрямую без ее конвертации в промежуточные форматы и становится доступной для всех участников коллективной работы для координации проекта.

Независимо от используемых функций, всегда следует проверять созданные или полученные при проектировании IFC файлы, по меньшей мере, в отношении двух аспектов:

- необходимо проверять, что геометрия здания корректна, и что компоненты корректно расположены; необходимо убедиться, что модель завершена (имеется запись конца в тексте IFC-файла);
- необходимо проверять, что наборы атрибутов объектов включают все необходимые параметры, атрибуты и информацию.

В зависимости от того, какая программа используется для контроля, доступность функций и порядок выполнения действий могут отличаться. Работа со средством просмотра IFC для различных программ также может отличаться.

Управление данными IFC-модели заключается не только в создании, сохранении, просмотре и редактировании данных, но и включает вспомогательные и проверочные функции, такие как:

- создание собственных IFC-данных, не входящих в Схему IFC;
- сопоставление данных разных версий одной IFC-модели и обнаружение изменений и коллизий;
- обновление модели;
- фильтрация определенных свойств IFC-модели для сравнения общих данных для смежных проектов (например, выбрать колонны первого этажа или выбрать только элементы, участвующие в коллизиях);
- проверка на заданные ограничения; публикация данных IFC
- и т.д.

Для решения возможных коллизий и управления изменениями необходима специальная функция обнаружения изменений. Эта функция находит геометрические различия между двумя версиями экспортированной модели IFC, то есть сравнивает два файла IFC, являющимися двумя версиями одного и того же проекта. Элементы, которые были подвергнуты изменениям, объединяются во временный проект.

4.4 Рекомендации по использованию возможностей формата IFC при экспорте данных в программные комплексы

1. Универсального решения проблемы корректного обмена данными на данной стадии развития проектирования с использованием информационного моделирования не существует. Получение полного набора данных, необходимых для проектирования геометрически сложных объектов, обеспечивается либо при «ручной настройке» выгрузки данных IFC в принимающем приложении, либо с помощью специальной программы-конвертера.
2. При передаче IFC-данных из одного приложения в другое необходимо иметь в виду таблицу соответствия элементов программного комплекса и IFC (Приложение 1). Это позволит избежать появления неинтерпретируемых элементов и точно указать типы продуктов IFC.
3. IFC предусматривает ситуацию, когда необходимо передать нестандартные данные. Использование Свойств IFC, Атрибутов и Ссылок на Классификации дополняет свойства элементов информационной модели специфическими данными из смежных проектов и обеспечивает взаимодействие со специалистами других разделов проектирования.
4. В программных платформах информационного моделирования необходимо использовать функции или приложения (Менеджеры) для управления данными информационной модели, включающие создание, сохранение, просмотр, редактирование данных, а также вспомогательные приложения для автоматического обновления модели, обнаружения коллизий, исправления ошибок, проверки на заданные ограничения и т.д.
5. Различные организации и группы разработчиков самостоятельно могут разрабатывать спецификации для передачи своих данных, расширяя принятые в стандартах IFC. Критерием здесь служит удобство формата для разрабатываемого ПО и собственных проектов.

6. Согласованность изменений в рамках одного проекта достигается специальными приемами, например, параметрическое моделирование (создание семейств элементов с взаимосвязанными свойствами, чаще всего геометрическими).

Приложение 1 Форматы обмена данными

1.1 Типы форматов обмена данными

Форматы обмена можно разделить на следующие типы:

Форматы, используемые при обмене BIM-данными

- IFC - Industry Foundation Classes File
- IFCXML - Industry Foundation Classes XML
- IFCZIP - Industry Foundation Classes ZIP
- BCF – BIM Collaboration Format
- COBie - Construction Operations Building Information Exchange

CAD-форматы (2D/3D)

- DWG - Drawing Database File
- DXF - Drawing Exchange Format
- DGN - MicroStation Design File

Векторные графические форматы файлов

- DWF(X) - Drawing Web Format
- PDF - Portable Document Format file

Форматы файлов трехмерной графики

- SKP - SketchUp Document
- STL - STereoLithography
- OBJ - WaveFront File
- 3DS - 3D Studio Scene
- FBX - Filmbox File

Прочие, применяемые форматы

- CSV - Comma-Separated Values
- gbXML - Green Building XML
- DSTV – формат для станков ЧПУ

1.2 Открытый стандарт обмена данными – IFC (Industry Foundation Classes)

1.2.1 Общие положения

Формат IFC был разработан для упрощения взаимодействия в строительной индустрии. Используется как нейтральный формат для информационной модели здания (Building Information Model) и содержит соответствующие классы объектов, закрывающие различные потребности жизненного цикла зданий и сооружений.

Сейчас формат IFC используется практически всеми BIM-платформами, как универсальный формат обмена данными.

Язык IFC интерпретирует описания 3D-элементов модели здания (таких как перекрытия, стены, колонны, балки и т.д.) между различными программами в процессе проектирования. В отличие, например, от DXF – формата графических элементов, файлы IFC содержат полные спецификации архитектурных элементов, такие как их геометрические 3D-характеристики и атрибутивные данные и их взаимосвязь с другими объектами.

Модель взаимоотношений различных сущностей IFC описывается на языке моделирования данных EXPRESS (являющемся частью формата STEP для обмена САПР-данными).

Определение Модельного вида

Industry Foundation Classes (IFC, buildingSMART International) - это открытый формат для обмена данными OpenBIM. Чтобы осуществить обмен данными, необходимо задать *Определение Вида IFC* (IFC View Definition) или *Определение Модельного Вида* (Model View Definition - MVD). Фактически это подмножество структуры IFC Schema для задания всех понятий, необходимых для описания объекта (классы, атрибуты, связи, наборы свойств, количественные определения и т.д.), другими словами - спецификация требований для реализации интерфейса IFC.

Для экспорта и импорта геометрических данных через интерфейс IFC многие ПК с поддержкой BIM поддерживают самое раннее и самое распространенное

определение модельного вида *Скоординированный Вид* (Coordination View), разработанное buildingSMART International. Основная задача Coordination View - дать возможность архитекторам, инженерам-проектировщикам и службам эксплуатации совместно использовать геометрические данные (например, объемно-планировочное решение) из информационной модели здания. В Coordination View содержатся определения пространственных конструкций и технических систем здания.

Скоординированный Вид (Геометрия Поверхности) (Coordination View (Surface Geometry)) - упрощенный формат публикации (другими словами, подмножество) Скоординированного Вида. Этот формат подходит для просмотра (поддерживается всеми программами просмотра IFC), координации проектирования, предотвращения коллизий в процессе проектирования и проверки наличия коллизий. Скоординированный Вид (Геометрия Поверхности) задает контурное представление каждого элемента, то есть каждый элемент экспортируется с геометрией BREP (Boundary REPresentation - контурное представление). Это представление используется для воспроизведения реальных форм элемента. Однако при применении такого формата возможна потеря части информации, например, привязки и ориентация локальных осей (примеры таких ситуаций приводятся ниже).

Расширенная версия Скоординированного Вида - Основной Вид Передачи FM (с передачей информации об эксплуатации здания и управлении его инфраструктурой (FM)), в котором передаются геометрические характеристики и спецификация оборудования для пространственных инженерно-технических систем здания. Основной Вид Передачи FM предоставляет возможности, выходящие за рамки Скоординированного Вида: назначение компонентам мебели и оборудования пространствам (Содержимое Пространства, Система IFC), назначение зон (IFC Zone), классификация пространств и компонентов, назначение компонентам основных свойств производителя (стандартные и специальные Свойства IFC), назначение дверей и окон пространствам (Пространственные Границы), назначение компонентам информации о типе (Тип продукта IFC), экспорт основных показателей всех компонентов и пространственных конструкций проекта. В

качестве примера можно назвать COBie (Construction-Operations Building information exchange/обмен информацией о Строительных Операциях Зданий) - спецификации, используемой при передаче FM-информации в формате электронной таблицы для передачи информации о модели здания, а не о геометрии его модели. Документация COBie формируется из данных IFC с помощью бесплатных или коммерческих программ-конвертеров; такую возможность обеспечивает, например, ArchiCAD [1].

Имеются и другие определения модельного вида (расширенные варианты скоординированного вида), которые специфицируются организациями или группами разработчиков, не входящих в buildingSMART International. Например, определение модельного вида Concept Design BIM 2010, которое поддерживается General Services Administration (US GSA), Statsbygg (Norway) и Senate Properties (Finland). Дополнительные определения модельного вида требуют программ для поддержки данных IFC, которые выходят за рамки стандарта Скоординированного вида. К таким данным относятся Ссылка на классификацию (Classification Reference), Владелец пространства (Space Occupant), Действующее лицо (Actor), Назначение Системы (System Assignment), Назначение Каталога временного ряда (Time Series Schedule Assignment), а также специфические наборы свойств и параметры. Разработчики ПК обычно предусматривают возможность передачи таких типов данных, например, интерфейс IFC ArchiCAD предоставляет возможность определять, экспортировать и импортировать эти типы данных. В результате пользователь может выполнить, например, специфическую GSA-классификацию согласно разнообразным классам OmniClass, Statsbygg и Senate Properties нормативам.

Все Определения Модельных Видов могут быть расширены, если необходимо вывести дополнительную информацию. Так, в ArchiCAD доступны следующие полезные расширения: расчет потребности в материальных и трудовых ресурсах (преобразование базисных количественных данных всех пространственных и конструктивных элементов в стоимостные или ресурсные показатели), учет пространственных границ (добавление к строительным элементам

пространственных взаимосвязей для теплового и энергетического анализа зданий), практически во всех приложениях доступны 2D-аннотации.

В Revit Autodesk дополнительная информация необходима для визуализации и выпуска проектной документации по результатам расчетов смежных разделов, проведенных в облаке на основе аналитической модели, например, после выполнения конструктивных расчетов - отображение и детализация армирования, узлов металлоконструкций, бетонных и железобетонных элементов и т.д.

Файловые форматы данных IFC

Основные файловые форматы данных IFC, обеспечивающие экспорт и импорт информации:

Формат IFC представлен в следующих спецификациях:

- **ifc**: формат обмена IFC по умолчанию, использующий физическую структуру файла STEP (международный стандарт ISO10303-21);
- **ifcZIP**: файл данных IFC, использующий алгоритм сжатия ZIP (сжатый вариант .ifc или.ifcXML);
- **ifcXML**: файл данных IFC, использующий структуру документа XML, который используется в том случае, когда сторонние программы не читают исходный формат ifc, но могут работать с базами данных xml (например, сметные расчеты, энергетический анализ и т.д.). Этот формат может содержать ту же информацию о модели, что обычный формат ifc, но в нем элементы и их свойства хранятся в более информативных структурах данных (файл ifcXML обычно в 3-4 раза больше файла ifc).

Обмен данными IFC может быть улучшен благодаря взаимодействию с использованием BCF (BIM Collaboration Format). Этот формат позволяет добавлять комментарии для элементов IFC-модели. Благодаря этому остальные участники проекта могут легко находить нужные элементы IFC-модели. Обмен данными на основе BCF хорошо подходит для маркировки элементов - источников коллизий (например, возникающих между архитектурными и конструктивными элементами).

Сущность IFC

В модели IFC проектная информация (сгенерированная, например, архитектурным ПО) представляется в виде набора Сущностей IFC - это элементы, покрытия и взаимосвязь между ними. В описание сущности (например, IfcWall) входит фиксированное множество атрибутов IFC и произвольное количество дополнительных свойств IFC. IFC включает несколько сотен сущностей, среди которых строительных сущностей IfcBuildingElement (элементов строительной конструкции, таких как стена IfcWall, колонна IfcColumn, балка IfcBeam) всего 25 (рисунок 1.1).



Рисунок 1.1 Перечень строительных элементов с сайта http://helpcenter.graphisoft.ru/

Некоторые сущности IFC представляют характеристики других сущностей (например, профиль IfcProfileDef, строительный материал IfcMaterial, многослойный материал IfcMaterialLayerSet (объемная геометрия) или IfcMaterialList (геометрия BREP), слой IfcPresentationLayerAssignment, поверхность IfcSurfaceStyleRendering).

Контейнер IFC

Контейнер IFC (связаный с IfcRelAggregates) - это сущность IFC, не имеющая собственной геометрии. При этом компоненты этой сущности (связанные объекты IfcRelAggregates) содержат все геометрические характеристики и данные об их конструкции. К примеру, в контейнер IfcStair (Лестница) входят геометрически определенные компоненты IfcStairFlight (ЛестничныйМарш), IfcSlab (Перекрытие с типом Площадка), IfcRailings (Ограждение), или другой тип сущности, такой как IfcBuildingElementProxy.

Тип продукта IFC

Тип продукта (Type Product) IFC - это сущность IFC, которая определяет конкретный стиль/тип других сущностей посредством их связывания с обычными атрибутами и свойствами IFC. Например, элемент (сущность) Балка (IfcBeam) соответствует типу продукта IfcBeamType с атрибутами Профиль/Имя Строительного Материала + размер профиля, а элемент Перекрытие IfcSlab - типу продукта IfcSlabType с атрибутами Строительный Материал/имя Многослойной Конструкции + толщина (табл. 1.1).

Таблица 1.1 (из [1])

Элементы (Сущности), Типы и Атрибуты типов

Элемент (Сущность IFC)	**Тип (тип продукта IFC)**	**Атрибуты Типа продукта**
Колонна (IfcColumn)	IfcColumnType	Профиль/Имя Строительного Материала + размер профиля
Балка (IfcBeam)	IfcBeamType	Профиль/Имя Строительного Материала + размер профиля
Стена (IfcWall)	IfcWallType	Строительный Материал/имя Многослойной Конструкции + толщина
Перекрытие (IfcSlab)	IfcSlabType	Строительный Материал/имя Многослойной Конструкции + толщина
Навесная стена (IfcCurtainWall)	IfcCurtainWallType	“Curtain Wall Type” - фиксированное имя

Элемент (Сущность IFC)	Тип (тип продукта IFC)	Атрибуты Типа продукта
Панель стены (IfcPlate)	IfcPlateType	Тип панели (“Основная” или “Дополнительная”) + размеры панели
Рама навесной стены (IfcMember)	IfcMemberType	Тип рамы (“Обвязка”, “Основной профиль” или “Дополнительный профиль”) + размеры профиля
Дверь (IfcDoor)	IfcDoorStyle	имя библиотечного элемента
Окно (IfcWindow)	IfcWindowStyle	имя библиотечного элемента
Объекты GDL	-	имя библиотечного элемента, напр., IfcFurnitureType

Element Classification	IFC Type (PREDEFINED TYPE)	Default IFC Type Product	Alternative IFC Type Product (can be set in IFC Manager)
Duct Flow Terminal	IfcFlowTerminal	IfcAirTerminalType	IfcElectricApplianceType, IfcElectricHeaterType, IfcFireSuppressionTerminal Type, IfcGasTerminalType, IfcLampType, IfcLightFixtureType, IfcOutletType, IfcSanitaryTerminalType, IfcStackTerminalType, IfcWasteTerminalType
Duct Segment	IfcFlowSegment	IfcDuctSegmentType	IfcCableCarrierSegment Type, IfcCableSegmentType, IfcPipeSegmentType
Energy Conversion Device	IfcEnergyConversion Device	IfcTransformerType	IfcAirToAirHeatRecovery Type, IfcBoilerType, IfcChillerType, IfcCoilType, IfcCondenserType, IfcCooledBeamType, IfcCoolingTowerType, and more

Рисунок 1.2 Классификация элементов ArchiCAD, типы IFC, типы продукта IFC по умолчанию и альтернативные варианты типов продукта (из [1])

При установлении типов элементов IFC необходимо следить за правильным выбором типа продукта (рисунок 1.2), так как возможно несколько альтернативных вариантов.

Атрибуты IFC

Атрибуты (Attributes) IFC - это основные идентификаторы сущностей IFC. Имена атрибутов IFC фиксированы и являются стандартными. Большинство атрибутов извлекаются из проекта. Например, из ArchiCAD можно получить следующие атрибуты элемента здания IfcWall:

GlobalId: Глобальный Уникальный Идентификатор IfcWall в модели IFC (недоступен для редактирования);

OwnerHistory: информация о текущем владении IfcWall;

Name: имя элемента, например, производное от ID IfcWall;

Description: текстовое описание (не обязательный атрибут);

ObjectType: текст, описывающий подтип элемента или несущий дополнительную информацию о типе (не обязательный атрибут);

Tag: уникальный ID в проекте конкретного ПК (не совпадает с IFC GlobalId).

Свойства IFC

IFC-свойства являются дополнительными (в большинстве случаев относящимися к Типу IFC) параметрами, назначаемыми Сущностям IFC. IFC-свойства могут быть либо стандартными (данные Схемы IFC), либо они могут быть специальными. Специальные свойства создаются экспортирующими приложениями. Имена таких свойств хранятся в наборе свойств (имена наборов свойств обычно содержат имя экспортирующей программы).

Свойства IFC полезны для введения в проект таких специальных данных, которые не предусмотрены в специализированном программном комплексе. Например, степень огнестойкости и акустической защиты может быть назначена для элементов модели, не являющихся Объектами, при помощи Свойств IFC. Любые подобные данные можно увидеть при помощи стандартных средств программного комплекса, например, Выносных Надписей, диалога Найти и Выбрать и т.д.

Таким образом, использование Свойств IFC, Атрибутов и Ссылок на Классификации обеспечивает не только взаимодействие со специалистами других разделов проектирования, дополняя данные о свойствах элементов модели.

Иерархия модели IFC

Модель IFC - это совокупность иерархически упорядоченных элементов (сущностей IFC). Как показано на рисунке 1.3, самый верхний уровень - IfcProject (Проект), его подуровень - IfcSite (Местность). Элементы могут связываться напрямую с IfcSite (Местность). Так, соседние здания и элементы окружающей среды (деревья, ограды, дороги, и т.п.) могут принадлежать местности, а не проектируемому зданию и его этажам. Следующий подуровень - IfcBuilding (Здание), затем идут этаж (IfcBuildingStorey) и элементы этого этажа IfcBuildingElement (например, IfcWall).

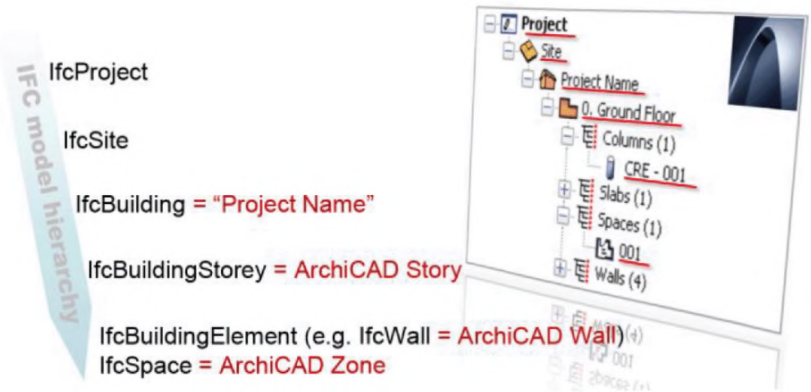


Рисунок 1.3 Иерархия элементов IFC (из [1])

Например, в ArchiCAD все объекты и элементы привязываются к собственным этажам и представляются в иерархии модели IFC в виде списка IfcBuildingStorey с собственными именами этажей (рисунок 1.3).

IfcSite является географическим местом IfcProject. Она не является обязательной, но может иметь геометрию. В ArchiCAD геометрия местности представляется элементами 30-сетки и объектами типа местность и элементами на местности. Другие приложения могут экспортировать здание вместе с описанием местности IfcSites.

Назначения IFC

Назначения IFC (IfcRelAssigns) определяют взаимные связи между различными элементами проекта. Каждый тип назначения может иметь свой собственный атрибут IFC (Object Type) и стандартные или специальные свойства IFC.

Основные назначения IFC:

- Группа IFC (IfcGroup): можно сгруппировать в единое целое любые элементы проекта (элемент здания, этаж и т.д.). Например, колонна и балка являются сгруппированными компонентами рамной конструкции. Для назначения IfcGroup доступно группирование с использованием многоуровневой иерархии. Например, можно сгруппировать рамные группы в группу «Система рам».
- Зона IFC (IfcZone): предназначена для объединения в одну группу элементов IfcSpace. Например, зоны ArchiCAD, имеющие идентичную функцию, группируются в IfcZone под именем Зона безопасности. Также для назначения IfcZone доступно группирование с использованием многоуровневой иерархии. Например, можно сгруппировать несколько Зон ArchiCAD (IfcSpaces) в IfcZone (Зоны Безопасности), являющуюся частью группы IfcZone более высокого уровня (Правительственные Зоны).
- Система IFC (IfcSystem): предназначена для ручного группирования любых элементов проекта (особенно элементов инженерных сетей) в иерархические системы, например, элементов систем, созданных с помощью Конструктора MEP или экспортированных из различных инженерных приложений. Доступна установка иерархии подсистем (родительских и дочерних систем). Например, сгруппированные лифты могут составлять систему вертикальных связей, являющуюся дочерней подсистемой системы оборудования. Другой пример - элементы, сгруппированные в систему холодного водоснабжения, могут являться подсистемой водоснабжения. Кроме того, можно назначать системы по пространственным структурным связям (IfcSite, IfcBuilding, IfcStorey или IfcSpace), например, добавлять все Зоны (IfcSpace) к системе проходящих через них воздуховодов.

- Действующее лицо (IfcActor): для приписывания действующего лица (человека и/или организации) и его роли (например, владелец, архитектор, заказчик) любому элементу проекта (проект, этаж, элемент здания и т.д.). Пример: присвоение действующих лиц «Архитектор проекта» и «Заказчик/владелец проекта» (GSA- требования) Ifc-проекту.
- Владелец пространства (Space Occupant) (IfcOccupant): для определения взаимосвязи «Владеет» между действующим лицом (человеком и/или организацией) и одним или более IfcSpaces или их группами (IfcZone).
- Каталог Временного Ряда (IfcTimeSeriesSchedule): Используется для настройки расчетного числа людей, освещения и оборудования таких элементов проектов как Зоны (в ArchiCAD). Например, значения Временного Ряда могут быть приписаны для IfcSpaces с указанием дат начала и окончания, времени эксплуатации в течении года, ежемесячно, еженедельно, ежедневно или на протяжении повторяющихся периодов и т.д.

Все назначения данных создаются при импорте модели IFC (например, IfcSystems, включенные инженерную (MEP-типа) модель IFC или IfcZones, определенные в FM- приложении).

Ссылка на классификацию IFC

Ссылка на классификацию (IfcClassificationReference) используется для упорядочения элементов IFC в категории в соответствии с заранее определенной классификационной таблицей (пользовательские данные). Всем элементам проекта (таким, например, как здание, этажи, конструктивные элементы, зоны) можно присвоить ссылочный идентификатор элемента, реквизит классификационного имени и некоторые другие параметры. Например, классифицирование элементов здания в соответствии с таблицей OmniClass является требованием US GSA, а классификация по системе Uniclass требуется в Великобритании.

Таким образом, для упорядочения работы с информационной моделью желательно разработать универсальную классификационную таблицу компонентов BIM-модели, единую для всех участников, задействованных в проекте:

архитекторов, проектировщиков, конструкторов, технологов и строителей. В ней будет установлена иерархия элементов, в том числе нестандартных.

Типы элементов IFC

Каждый элемент архитектурной модели, элемент строительной конструкции и элемент аннотации имеет свой аналог в структуре IFC (тип элемента IFC).

В таблице 2 приводится соответствие между элементами ArchiCAD и типами элементов IFC. Подобная таблица соответствия элементов ПК и IFC необходима при передаче данных из одного программного в другой.

ArchiCAD			Подтип	
Wall (стена)	IfcWall или IfcWallStandard Case (в зависимости от геометрии)		Объект > стена	IfcWall
Дверь	IfcDoor		Объект > Дверь	IfcDoor
Окно	IfcWindow		Объект > Окно	IfcWindow
Световой люк	IfcWindow		Объект > Проем	IfcOpeningElement
Крыша	IfcSlab (предварительно определенный тип Roof)		Объект > Крыша	IfcSlab (предварительно определенный тип Roof)
Оболочка	IfcSlab (предварительно определенный тип Roof)		Объект > Балка	IfcBeam
Beam (балка)	IfcBeam		Объект > Колонна	IfcColumn
Column (колонна)	IfcColumn		Объект > Свая	IfcPile
Перекрытие	IfcSlab (предварительно определенный тип Floor)		Объект > Перекрытие	IfcSlab (предварительно определенный тип Floor)
Лестница	IfcStair		Объект > Сборная железобетонная плита	IfcSlab
Пандус (Stair Maker)	IfcStair		Объект > Плита	IfcPlate
SD-сетка	IfcBuildingElementProxy (как геометрия IfcSite)		Объект > Элемент	IfcMember
Навесная стена	IfcCurtainWall		Объект > Предварительно напряжённая арматура	IfcTendon
Зона	IfcSpace		Объект > Лестница	IfcStair

ArchiCAD			Подтип	
Морф	IfcBuildingElementProxy		Объект > Лестничный марш	IfcStairFlight
Угловое окно	IfcWindow		Объект > Пандус	IfcRamp
Элемент сетки	IfcGridAxis		Объект > Марш пандуса	IfcRampFlight
Система сетки	IfcGrid		Объект > Ограждение	IfcRailing
Источник света	IfcFlowT erminal		Объект > Навесная стена	IfcCurtainWall
Размер	IfcAnnotation		Объект > Мебель	IfcFurnishingElement
Отметка уровня	IfcAnnotation		Объект > Деревянная ферма	IfcBeam
Text (текст)	IfcAnnotation		Объект > Фундамент	IfcFooting
Выносная надпись	IfcAnnotation		Объект > Основание	IfcFooting
Штриховка	IfcAnnotation		Объект > Пространство	IfcSpace
Линия	IfcAnnotation		Объект > Покрытие	IfcCovering
Дуга/окружность	IfcAnnotation		Объект > Арматурный стержень	IfcReinforcingBar
Полилиния	IfcAnnotation		Объект > Арматурная сетка	IfcReinforcingMesh
Угловой размер	IfcAnnotation		Объект > Электрический элемент	IfcFlowTerminal
Сплайн-кривая	IfcAnnotation		Объект > Фитинг потока	IfcFlowFitting

1.2.2 Примеры описания сущностей схем основных данных

Схемы основных данных устанавливают объединяющий слой внутри архитектуры схемы IFC. Сущности, определенные в данном слое, могут быть обозначены и конкретизированы всеми сущностями, находящимися визуально выше по иерархии. Основной слой образовывает базовую структуру, фундаментальные связи и общие концепции для всех дальнейших конкретизаций в аспекте указанных моделей.

Все сущности, определенные в данном слое и выше, происходят от IfcRoot (Корневой каталог), имеющего уникальное обозначение, имя, описание и изменение информационных данных).

Схемы основных данных:

- ядро (IfcKernel).
- расширение контроля (IfcControlExtension);
- расширение процессов (IfcProcessExtension);
- расширение продуктов (IfcProductExtension).

Пример описания: IfcGroup (Группа)

IfcGroup является обобщением любой произвольной группы. Группа является логическим набором объектов. Она не имеет своего собственного положения, и при этом не может содержать свое собственное представление формы. Поэтому группа является агрегацией в соответствие с некоторыми негеометрическими/топологическими аспектами группировки.

Примечание – следует ипользовать IfcRelDecomposes вместе с надлежащими подтипами IfcProduct для определения агрегации продуктов, которые могут иметь ее собственное положение и представление формы.

Пример: примером для группы является система, которая группирует элементы в соответствие с таким аспектом как их роли, независимо от их положения в здании.

Группа может содержать любой набор объектов (являющийся продуктами, процессами, средствами управления, ресурсами, участниками или другими группами). Таким образом группы могут быть созданы. Объект может быть частью одной или нескольких групп, может не принадлежать ни одной группе. Отношения группирования не требуются, чтобы быть иерархическими, и при этом они не подразумевают зависимость.

Примечание – следует использовать IfcRelDecomposes вместе с надлежащими подтипами IfcProduct для определения иерархической агрегации продуктов. Группа может быть заменена без изменения уже имеющихся объектах в наборе группы.

ИСТОРИЯ: новая сущность в версии в IFC1.0.

ИЗМЕНЕНИЕ в версии IFC4: обратное отношение *IsGroupedBy* установлено в 0.. n

Определение использования отношения

Группы назначаются другим объектам (таким как процесс или ресурс) отношением объекта, который относится к соответствующему объекту:

- Процесс: для назначения используется IfcRelAssignsToProcess
- Ресурс: для назначения используется IfcRelAssignsToResource

Группы могут быть подвергнуты средствам управления. Тогда назначается информация о средствах управления:

- Средства управления: для воздействия на группу используется IfcRelAssignsToControl

Определения общего использования

Следующие понятия унаследованы в супертипах:

- IfcRoot: Id**ENTITY**, Revision Control

Назначение группы

Понятие Назначение группы к этой сущности.

XSD Specification:

```
<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>
```

EXPRESS Specification:

```
ENTITY IfcGroup
 SUPERTYPE OF(ONEOF(IfcAsset, IfcInventory, IfcStructuralLoadGroup,
 IfcStructuralResultGroup, IfcSystem))
 SUBTYPE OF IfcObject;
 ENTITY
```

IsGroupedBy :**SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

END_ENTITY;

EXPRESS-G диаграмма

Attribute Definitions:

IsGroupedBy :Ссылка на отношение IfcRelAssignsToGroup, которое назначает нескольким членам группы объект IfcGroup.
ИЗМЕНЕНИЕ в версии IFC4: количество элементов было изменено от 1..1 на 0..? для разрешения обмена понятием группы без того чтобы назначить уже имеющиеся члены группы. Это теперь также позволяет использовать многих случаяхв IfcRelAssignsToGroup чтобы назначать членов группы. Изменение было сделано с восходящей совместимостью для файла определяющего обмен.

Inheritance Graph:

ENTITY IfcGroup

ENTITY IfcRoot

GlobalId :IfcGloballyUniqueId;

OwnerHistory :**OPTIONAL** IfcOwnerHistory;

Name :**OPTIONAL** IfcLabel;

Description :**OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

ENTITY

HasAssignments :**SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests :**SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy :**SET OF** IfcRelNests **FOR** RelatingObject;

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** IfcLabel;

ENTITY

IsDeclaredBy :**SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares :**SET OF** IfcRelDefinesByObject **FOR** RelatingObject;

IsTypedBy :**SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy :**SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcGroup

ENTITY

IsGroupedBy :**SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

END_ENTITY;

Пример описания: IfcObject (Объект)

IfcObject является обобщением любой семантически рассматриваемой вещи или процесса. Объекты являются вещами, поскольку они появляются - т.е. возникают.

Примечание – примеры IfcObject включают физически материальные позиции, такие как стена, балка или покрытие, физически существующие позиции, такие как пространство или концептуальные позиции, такие как сетки или виртуальные границы. Также используется для обозначения процессов, таких как задачи работы, для средств управления, таких как позиции стоимости, для участников, таких как заинтересованные лица в процессе проектирования.

Объекты можно именовать с помощью унаследованного атрибута *Name*, который должен быть распознаваемым маркером для пользователя при возникновении объекта. Дальнейшие объяснения к объекту могут быть даны с помощью унаследованного признака *Description*. Атрибут *ObjectType* используется:

- для сохранення определенных пользователем значений для всех подтипов IfcObject, где атрибут *PredefinedType* использван, и его значение установлено в USERDEFINED;
- для предоставления информации типа (может быть замечен как очень легкий классификатор) подтипа IfcObject, если не дан атрибут *PredefinedType*. Это

часто имеет место, если никакой всесторонний список предопределенных типов не доступен.

Объекты являются независимыми данными, которые могутсодержать или ссылаться на другие данные. Существует несколько отношений, в которые могут быть вовлечены объекты:

- **Ассоциация к внешней/внутренней информации о ресурсе** - отношение ассоциации, которое относится к внешним/внутренним источникам информации. Следует смотреть супертип IfcObjectDefinition для получения дополнительной информации.
- **Назначение других объектов** - отношение назначения, которое относится к другим типам объектов. Следует смотреть супертип IfcObjectDefinition для получения дополнительной информации.
- **Агрегация других объектов** - отношение агрегации, которое устанавливает отношение целое/часть. Объекты могут или быть целым, или частью или обоими. Следует смотреть супертип IfcObjectDefinition для получения дополнительной информации.
- **Назначение типа**: *IsTypedBy* - отношение определения IfcRelDefinesByType, которое использует определение типа для определения общих характеристик этого случаи, потенциально включая общее представление формы и общие свойства для всех случаев объекта, назначенных этому типу. Это отношение характерный-возникший с подразумеваемыми зависимостями (поскольку свойства возникновения зависят от свойств типа, но могут отвергнуть их).

Примечание – следует смотреть IfcRelDefinesByType для объяснения. Также следует смотреть, как переопределить свойства типа свойствами возникновения.

- **Назначение частичного типа**: *IsDeclaredBy*, *Объявляет* - отношение определения IfcRelDefinesByObject, которые используют компонент определения типа (часть типа, названного «частью объявления») для определения компонента возникновения (часть возникновения, названного «отраженной частью»). Это также называется как «глубокая копия». Общие характеристики всех частей в возникновении определяются частями в типе. Это отношение характерный-

возникший с подразумеваемыми зависимостями (поскольку свойства возникновения зависят от свойств типа, но могут отвергнуть их).

Примечание – следует смотреть IfcRelDefinesByObject для объяснения.

- **Назначение проявлений свойства**: *IsDefinedBy* - отношение определения IfcRelDefinesByProperties, который назначает определения проявлений свойств возникающему объекту.

ИСТОРИЯ: новая сущность в IFC1.0

ИЗМЕНЕНИЕ в версии IFC4: обратные связи *Declares*, *IsDeclaredBy*, и *IsTypedBy* были добавлены, типы более не включенные в отношение *IsDefinesBy*. IfcProject был продвинут, чтобы быть подтипом IfcObjectDefinition-> IfcContext.

Неофициальные предложения:

1. Частичное назначение типа, т.е. обратный признак *IsDeclaredBy* или *Declares*, должен использоваться, если объект будет частью декомпозиции, т.е. если или *IsDecomposedBy* или *Decomposes*, проявляется.

Спецификация XSD:

```
<xs:element name="IfcObject" type="ifc:IfcObject" abstract="true"
substitutionGroup="ifc:IfcObjectDefinition" nillable="true"/>
 <xs:complexType name="IfcObject" abstract="true">
  <xs:complexContent>
   <xs:extension base="ifc:IfcObjectDefinition">
    <xs:attribute name="ObjectType" type="ifc:IfcLabel" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS спецификация:

```
ENTITY IfcObject
 ABSTRACT SUPERTYPE OF(ONEOF(IfcActor, IfcControl, IfcGroup, IfcProcess,
 IfcProduct, IfcResource))
 SUBTYPE OF IfcObjectDefinition;
```

ObjectType :**OPTIONAL** IfcLabel;

ENTITY

IsDeclaredBy :**SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares :**SET OF** IfcRelDefinesByObject **FOR** RelatingObject;

IsTypedBy :**SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy :**SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

END_ENTITY;

Пример описания: IfcProject (Проект)

IfcProject указывает совершение некоторого процесса проектирования, разработки, строительства или технического обслуживания, приводящего к продукту. Проект устанавливает контекст для получения информации, которая будет обменена или будет в общем использовании, это может представлять проект строительства. Главная цель IfcProject в обменной структуре состоит в том, чтобы обеспечить начало и контекст для всей другой информации.

Контекст, обеспеченный IfcProject, включает:

- единицы по умолчанию
- геометрический контекст представления для обменных структур включая представления формы
- мировая система координат
- измерение координат пространства
- точность, используемая в пределах геометрических представлений, и произвольно указание относительно истинного севера в мировой системе координат

ИСТОРИЯ: новая сущность в IFC1.0

ИЗМЕНЕНИЕ в версии IFC4: атрибут, *RepresentationContexts* и *UnitsInContext* сделаны дополнительными и поддерживают супертип IfcContext.

Неофициальные предложения:

1. В пределах обменного контекста должен быть только один проект. Это необходимо из-за глобального правила IfcSingleProjectInstance.

Определения общего использования

Следующие понятия унаследованы в супертипах:

- IfcRoot: Id**ENTITY**, Revision Control

Проектная декларация

Понятие Проектная декларация относится к этой сущности как показано в таблице.

Type (Тип)	**Описание**
IfcProjectLibrary	Проекты, на которые ссылаются, соединились в этот проект.
IfcPropertySetTemplate	Шаблоны назначения типа, определенные этим проектом.
IfcTypeObject	Определения типа, определенные этим проектом.
IfcActor	Участники, участвующие в этом проекте.
IfcControl	Средства управления определенные в рамках этого проекта.
IfcGroup	Группы определенные этим проектом.
IfcProcess	Процессы определенные этим проектом.
IfcResource	Ресурсы определенные этим проектом.

Таблица — декларация проекта IfcProject

IfcProject является также контекстом для другой информации о проекте строительства, таком как план работ. Структуры не-продуктов назначены их первым уровнем объектов IfcProject с помощью отношения IfcRelDeclares.
IfcProject обеспечивает контекст для пространственных элементов и связанных продуктов, и для планов работ (или других основных не-продуктов) описания проекта строительства. Это обработано двумя различными объектами отношения как показано в рисунке.

Примечание – пространственная структура и структура графика может анализироваться. Например, IfcBuilding может анализироваться в IfcBuildingStorey и IfcWorkPlan может анализироваться в IfcWorkSchedule.

Примечание – продукты и задачи могут анализироваться и далее. Например, IfcCurtainWall может анализироваться в IfcMember и IfcPlate, IfcTask может анализироваться в другой IfcTask.

Примечание – продукты и задачи могут иметь прямые отношения соединения. Например, IfcCurtainWall можно назначить на IfcTask как исходные данные или производительность для графика строительства.

Примечание – аномалия для использования составной структуры через IfcRelAggregates для назначения верхнего пространственного контейнера к IfcProject вследствие восходящей совместимости с более ранними выпусками этого стандарта. Рисунок 1.4 иллюстрирует контекст для библиотек проекта, которые в ответ предоставляют контекст позициям библиотеки, назначенным на него. Типы продукта являются примером для позиций, которые могут быть включены в библиотеку проекта.

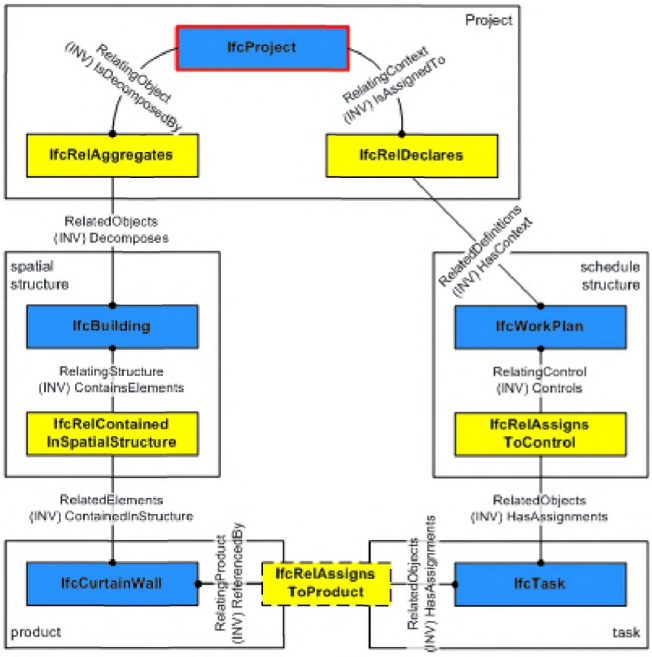


Рисунок 1.4 — Пространственный проект и структура плана работ

Рисунок иллюстрирует проектные отношения с пространственными структурами, продуктами и библиотеками типа.

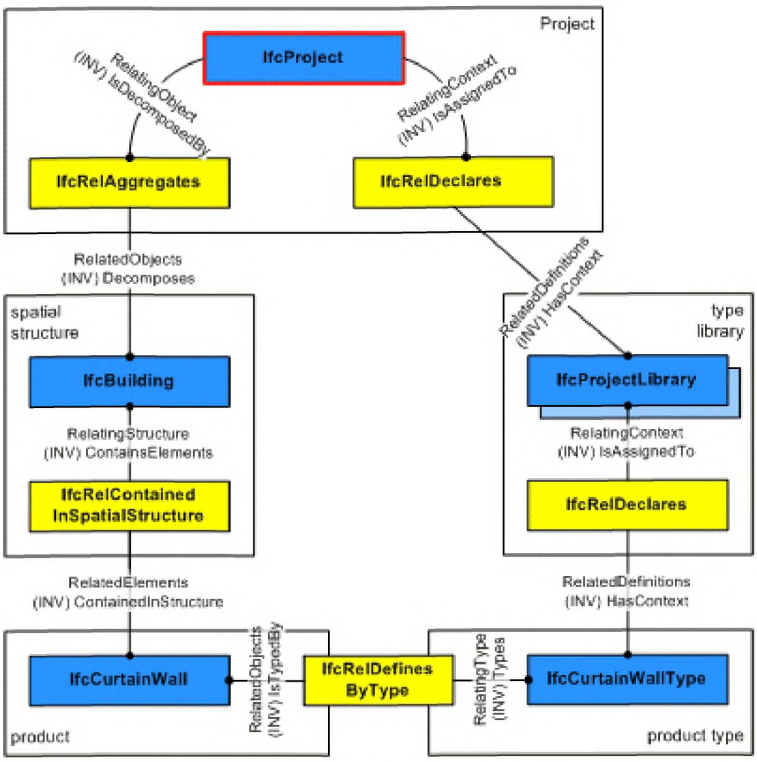


Рисунок 1.5—Пространственная структура проекта, продукты и библиотека типов продукта

Пространственное декомпозиция

Понятие Пространственная декомпозиция применяется к этой сущности как показано в Таблице.

Spatial Parts (Пространственные части)	**Описание**
IfcSite	Если ссылается, место является корнем пространственной структуры.
IfcBuilding	Если ссылается, здание является корнем пространственной структуры (никакая явная информация об участке не включена).
IfcSpatialZone	Если ссылается, пространственная зона является корнем пространственной структуры (чтобы использоваться в качестве стержня для не-строительных связанных проектов).

Таблица — IfcProject пространственная декомпозиция

IfcProject используется для ссылки на корень пространственной структуры здания или другого проекта строительства (который служит основной проектной разбивкой и должен быть иерархическим). Пространственные элементы структуры соединены и к IfcProject при помощи воплощенного отношения IfcRelAggregates.

Проектные единицы

Понятие Проектные единицы относится к этой сущности как показано в Таблице.

NamedUnitType	DerivedUnitType	Описание
LENGTHUNIT		Единица длины (метр, миллиметр или дюйм).
PLANEANGLEUNIT		Единица измерения угла (градус или радиан).

Таблица — Проектные единицы IfcProject

Проектный контекст

Понятие проектный контекст относится к этой сущности как показано в Таблице.

ContextIdentifier	ContextType	Описание
Model	3D	Контекст для всей 3D геометрии.
Plan	2D	Контекст для всех 2D аннотаций.

Таблица 7 — Проектный контекст IfcProject

Информация о проектной классификации

Понятие Информация о проектной классификации относится к этой сущности.

Информация о проектном документе

Понятие Информация о проектном документе относится к этой сущности.

Информация о библиотеке проекта

Понятие Информация о библиотеке проекта относится к этой сущности.

Спецификация XSD:

<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>
```

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.IFCGEOMETRICREP
                     RESENTATIONSUBCONTEXT' IN TYPEOF(Temp) )) = 0);
NoDecomposition : SIZEOF(SELF\IfcObjectDefinition.Decomposes) = 0;
HasOwnerHistory : EXISTS(SELF\IfcRoot.OwnerHistory);
END_ENTITY;
```

EXPRESS-G диаграмма

Formal Propositions:

HasName :Атрибут *Name* должен быть предусмотрен для IfcProject. Это - краткое название для проекта.

CorrectContext :Если отношение *RepresentationContexts* будет обеспечено, то тогда не должно быть случая IfcGeometricRepresentationSubContext непосредственно включенного в набор *RepresentationContexts*.

NoDecomposition :IfcProject представляет корень любого дерева декомпозиции. Поэтому это не должно использоваться для декомпозиии любого другого определения объекта.

HasOwnerHistory:Атрибут *OwnerHistory* должен быть предусмотрен для IfcProject.

Это обеспечивает минимум информации о владельце для набора данных по проекту и последние изменения, которые относяся к целому набору данных.

Прмечание: Каждый отдельный элемент данных, который происходит из IfcRoot, может иметь отдельный *OwnerHistory*. Это не допускает совместное владение и случайные действия с информацией, предоставленной в единственном случае IfcProject в наборе данных IFC.

ИЗМЕНЕНИЕ в версии IFC4: новое правило.

Inheritance Graph:

ENTITY IfcProject

ENTITY IfcRoot

GlobalId :IfcGloballyUniqueId;

OwnerHistory :**OPTIONAL** IfcOwnerHistory;

Name :**OPTIONAL** IfcLabel;

Description :**OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

ENTITY

HasAssignments :**SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests :**SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy :**SET OF** IfcRelNests **FOR** RelatingObject;

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** IfcLabel;

LongName :**OPTIONAL** IfcLabel;

Phase :**OPTIONAL** IfcLabel;

RepresentationContexts: **OPTIONAL SET** [1:?] **OF** IfcRepresentationContext;

UnitsInContext :**OPTIONAL** IfcUnitAssignment;

ENTITY

IsDefinedBy :**SET** [0:?] **OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

Declares :**SET OF** IfcRelDeclares **FOR** RelatingContext;

ENTITY IfcProject

END_ENTITY;

Пример описания: IfcRoot

IfcRoot является самым абстрактным и класс корня для всех определений сущности IFC, который внедряется в ядро или в последующие слои модели объекта IFC. Это - общий супертип всех сущностей IFC определенных в схеме ресурса IFC. Все предприятия, которые являются подтипами IfcRoot, могут использоваться независимо, тогда как сущности схемы ресурса, которые не являются подтипами IfcRoot, как предполагается, не являются независимыми сущностями.

Примечание – определения вида, и соглашение о внедрении могут наложить дополнительные ограничения на использование *OwnerHistory* для обработки объектного управления версиями.

ИСТОРИЯ: новая сущность в IFC1.0

ИЗМЕНЕНИЕ в версии IFC4: атрибут *OwnerHistory* сделан OPTIONAL.

Определения общего использования

Идентичность

Понятие Идентичность относится к этой сущности.

IfcRoot назначает глобально уникальный идентификатор. Кроме того, может быть предусмотрено имя и описание о понятии.

Контроль за изменениями

Понятие контроль за изменениями относится к этой сущности.

Владение, история и состояние слияния озватываются с помощью IfcOwnerHistory.

Спецификация XSD:

```
<xs:element name="IfcRoot" type="ifc:IfcRoot" abstract="true"
substitutionGroup="ifc:ENTITY" nillable="true"/>
<xs:complexType name="IfcRoot" abstract="true">
 <xs:complexContent>
  <xs:extension base="ifc:ENTITY">
   <xs:sequence>
    <xs:element name="OwnerHistory" type="ifc:IfcOwnerHistory" nillable="true"
minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="GlobalId" type="ifc:IfcGloballyUniqueId" use="optional"/>
   <xs:attribute name="Name" type="ifc:IfcLabel" use="optional"/>
   <xs:attribute name="Description" type="ifc:IfcText" use="optional"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

EXPRESS Specification:

```
ENTITY IfcRoot
 ABSTRACT SUPERTYPE OF(ONEOF(IfcObjectDefinition, IfcPropertyDefinition,
 IfcRelationship));
  GlobalId          :IfcGloballyUniqueId;
  OwnerHistory      :OPTIONAL IfcOwnerHistory;
  Name              :OPTIONAL IfcLabel;
  Description       :OPTIONAL IfcText;
 UNIQUE
  UR1               :GlobalId;
END_ENTITY;
```

EXPRESS-G диаграмма

Attribute Definitions:

GlobalId :Назначение глобально уникального идентификатора в пределах

всего программного обеспечения.

OwnerHistory : Назначение информации о текущем владении объектом, включая владение участником, применением, местной идентификацией и информацией о недавних изменениях объекта.
Примечание – только последние изменения сохраняются - дополнение, удаление или изменение.
ИЗМЕНЕНИЕ в версии IFC4: атрибут был изменен, чтобы быть OPTIONAL.

Name : Дополнительное наименование для использования участвующими системами программного обеспечения или пользователями. Для некоторых подтипов IfcRoot может требоваться вставка атрибута Name. Это должно выполняться как правило.

Description : Дополнительное описание, предусматривающеел обмен информативными комментариями.

Inheritance Graph:

ENTITY IfcRoot

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

END_ENTITY;

1.2.1 Примеры описания сущностей схем данных общих элементов

Схема данных общих элементов содержит описание вспомогательных сущностей – они выделены по тексту ниже синим цветом. Сущности, определяющие этот слой, могут иметь взаимосвязи и уточнять все сущности, находящие выше по иерархии. Слой общих элементов содержит более специализированные объекты, относящиеся к различным доменным областям.

1.2.3 Примеры описания сущностей схем взаимодействия основных данных

Схемы взаимодействия основных данных содержат промежуточные специализации сущностей. Сущности, определенные в этом слое, могут ссылаться на все сущности, стоящие выше в иерархии. Общий (совместный) слой элементов обеспечивает отношения между несколькими доменами и качественно описывает их

Схемы взаимодействия основных данных:

- общие строительные элементы (IfcSharedBldgElements);
- общие элементы инженерного обеспечения (IfcSharedBldgServiceElements);
- общие составные элементы (IfcSharedComponentElements);
- общие элементы управления оборудованеим (IfcSharedFacilitiesElements).
- общие элементы управления (IfcSharedMgmtElements).

Пример описания: IfcWall (Стена)

Стена представляет собой вертикальную конструкцию, ограничивающая или разделяющая пространство. Стены обычно являются вертикальными или около вертикальными плоскими элементами, чаще всего воспринимающими полезные нагрузки. Однако стены не обязательно должны быть несущими.

Примечание – в соответствии с ISO 6707-1: вертикальные конструкции, обычно изготавливаемые из каменной кладки или железобетона, выполняют функции несущих или ограждающих конструкций.

Примечание – рассматриваемые здесь стены для использования в конструктивной модели являются подтипом IfcStructuralMember, в свою очередь являющегося частью IfcStructuralAnalysisModel.

Примечание – произвольный плоский элемент, который по смыслу не может быть описан как стена, (не является преимущественно вертикальным), должен быть задан как IfcPlate.

Стены могут иметь дверные или оконные проемы, ниши и углубления. Они относятся к IfcOpeningElement, относящимся к стене с использованием взаимосвязи *HasOpenings*, указывающей на IfcRelVoidsElement.

Стены могут быть описаны тремя сущностями:

1. IfcWallStandardCase используется для всех экземпляров стен, имеющих постоянную толщину на всем своем протяжении, где значение толщины задано в параметрах материала. Эти стены всегда геометрически представляются в виде «Осевых» (плоских) и «SweptSolid» (или усеченные на основе «SweptSolid») если они заданы в виде трехмерной геометрии. Параметры профиля материала и контрольные точки геометрии стены могут быть описаны с помощью IfcMaterialProfileSetUsage.

2. IfcWallElementedCase используется для экземпляров стен, набранных из составных элементов по определенным правилам, описанным с помощью IfcRelAggregates.

3. IfcWall используется для всех экземпляров стен в остальных случаях, в частности со ступенчатым или неравномерным изменением толщины, или имеющих только контурное представление «BREP» или геометрию в виде сетки «SurfaceModel».

ИСТОРИЯ: новая сущность в IFC1.0

Общие положения для использования

Частные понятия и их подчинение супертипам:

- IfcRoot: Id**ENTITY**, Revision Control
- IfcElement: Product Placement, Box Geometry, FootPrint Geometry, Body SurfaceOrSolidModel Geometry, Body SurfaceModel Geometry, Body Tessellation Geometry, Body Brep Geometry, Body AdvancedBrep Geometry, Body CSG Geometry, Mapped Geometry
- IfcBuildingElement: Surface 3D Geometry

Типизация объектов (Object Typing)

По тексту ниже описаны общие сведения о присвоении типа рассматриваемым объектам IfcWall.

Экземпляру объекта может быть присвоен определенный тип. Пары вида «Объект – Тип_объекта» определены для большинства семантических сущностей – экземпляру объекта соответствует тип сущности объекта.

НАПРИМЕР: сущность IfcTank представляет собой объект экземпляра сущности, которому соответствует объект типа сущности IfcTankType.

В качестве образца, экземпляр может быть:

- аналогичной по отношению к его типу, в том числе по отношению ко всем описанным в сущности типа характеристикам;
- переопределенной по отношению к некоторым характеристикам;
- не имеющей своего определенного типа.

Параметры, определяемые на уровне типа сущности, могут содержать в себе:
- общее наименование и предопределенный тип (predefined type);
- общие параметры из состава набора типовых свойств (property set);
- общее представление о геометрии, соотнесенное с каждой первичной сущностью;
- общую принадлежность к материалу (за исключением набора свойств для его использования);
- общее представление о структуре состава.

Многие экземпляры и предметы типов сущностей имеют атрибут с именем PredefinedType, имеющий определенную нумерацию. Такое предопределение по сути обеспечивает еще один вид (иерархического) разделения объектов без необходимости создания дополнительных сущностей.

ПРИМЕР: для типов объектов, состоящих из отдельных частей, их части могут отражать объекты экземпляров из которых они состоят. Например, сущность типа «Стена» определяет конкретное расположение стержней армирования; экземпляр стены может также отражать конкретное расположение стержней армирования, при этом стержни армирования в ней имеют определенные взаимосвязи, например, учитывают расположение электрических распаечных коробок, которые они не имеют в определении сущности типа стены.

ПРИМЕЧАНИЕ: если тип объекта имеет объединенные элементы, то такие объекты отражаются объектами экземпляров с использованием взаимоотношений IfcRelDefinesByObject.

Type
IfcWallType

Таблица 183 — Типизация объектов (Object Typing)

Наборы параметров объектов (Property Sets)

Понятие шаблона набора параметров объектов описывает, каким образом экземпляр объекта взаимосвязан с одним или несколькими наборами параметров. Набор параметров содержит одно или несколько свойств. По типу данных индивидуальные параметры могут быть односимвольными, исчисляемыми, ограниченными (имеющими предельные значения), в виде таблицы, ссылок или списка, а также в виде комбинации параметров экземпляра.

Наборы параметров также могут быть взаимосвязаны с типами объектов. Они определяют общие параметры для всех экземпляров одного и того же типа. Если параметр совпадают (по имени) с параметром из набора параметров, то параметры экземпляра объекта, переопределяют свойства, присвоенные типу объекта.

Наборы параметров объектов применительно к рассматриваемой сущности имеют отношения, показанные в табл. 184.

PredefinedType	Имя
	Pset_ReinforcementBarPitchOfWall
	Pset_WallCommon
	Pset_ConcreteElementGeneral
	Pset_PrecastConcreteElementFabrication
	Pset_PrecastConcreteElementGeneral
	Pset_Condition
	Pset_EnvironmentalImpactIndicators
	Pset_EnvironmentalImpactValues
	Pset_ManufacturerOccurrence
	Pset_ManufacturerTypeInformation
	Pset_PackingInstructions
	Pset_ServiceLife
	Pset_Warranty

Таблица 184 — наборы свойств IfcWall для объектов

Наборы количественных параметров (Quantity Sets)

Объекты любой специализации могут иметь отношение к множеству наборов количественных параметров экземпляров. Наборы количественных параметров содержат множество разновидностей количественных величин экземпляров – ими могут быть количество, длина, площадь, объем, вес, время или комбинации параметров. Для каждой величины определено имя, значение, возможно описание и формула.

Наборы количественных параметров выражаются экземплярами IfcElementQuantity, где атрибут имени является указателем набора. Эта спецификация содержит ряд предопределенных наборов параметров, каждому из которых задан свой шаблон. Название шаблона должно использоваться как значение, сохраненного в атрибуте Имя. Атрибут MethodOfMeasurement определяет метод, по которому вычисляются значения отдельных величин.

Для шаблонов наборов количественных параметров, включенных в эту спецификацию, значение атрибута MethodOfMeasurement должно быть базовой величиной ("BaseQuantities").

Наборы количественных параметров применительно к рассматриваемой сущности имеют параметры, показанные в табл. 185.

Qto_WallBase Quantities	Имя	Описание
	Length	Суммарная номинальная длина стены по линии, принадлежащей срединной плоскости стены (даже если она отличается от направления стены).
	Width	Суммарная номинальная ширина (или толщина) стены, измеренная по нормали (перпендикулярно) к направлению стены. Параметр должен быть определен только в том случае, если его величина постоянна на протяжении всей длины стены.
	Height	Суммарная номинальная высота стены. Параметр должен быть определен только в том случае, если его величина постоянна на протяжении всей длины стены.
	GrossFootprintArea	Площадь стены брутто, без внимания к учету каких-либо изменений (например, углублений и проемов). Также называется как «Foot Print» – «Площадь основания» стены.
	NetFootprintArea	Площадь стены нетто, за вычетом каких-либо изменений (например, углублений и проемов). Также называется как «Foot Print» – «Площадь основания» стены.
	GrossSideArea	Площадь стены брутто, которая получается из продольного сечения по центральной оси, без вычета проемов.
	NetSideArea	Площадь стены нетто, которая получается из продольного сечения по центральной оси, с вычетом внутренних проемов.
	GrossVolume	Объем стены брутто без внимания к учету объема отверстий и объемов геометрии узлов

Qto_WallBase Quantities	Имя	Описание
		соединений.
	NetVolume	Объем стены нетто за вычетом отверстий и объемов геометрии узлов соединений.
	GrossWeight	Общий вес стены брутто без внимания к учету дополнительных элементов, отверстий и углублений.
	NetWeight	Общий вес нетто стены за вычетом дополнительных элементов, отверстий и углублений.

Таблица 185 — IfcWall, наборы количественных параметров (Quantity Sets)

Набор слоев материала (Material Layer Set)

Наборы слоев материала применительно к рассматриваемой сущности, описанные в этом подразделе.

Сущность материала IfcMaterial определяется напрямую IfcMaterialLayerSet или как резервный с помощью IfcMaterial или непосредственно с помощью IfcWallType.

Примечание – неправильно назначать IfcMaterialLayerSetUsage для IfcWall. Только подтип IfcWallStandardCase поддерживает эту концепцию.

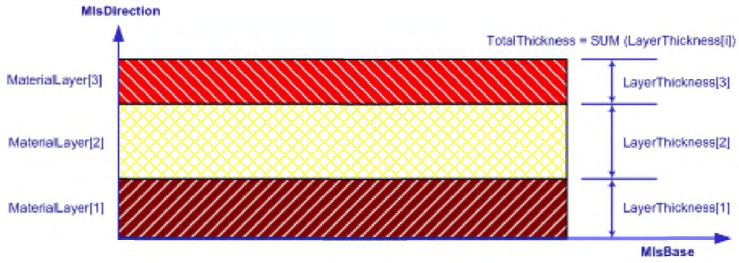


Рис. Набор слоев материала

Набор слоев материалов связан с продуктами или типами продуктов и содержит параметрические сведения о слоях, имеющих данные о точных толщинах заполнения контуров продуктов или экземпляров типов продуктов. Примерами таких продуктов или типов продуктов являются панели, плиты и стены.

Траектория соединения (Path Connectivity)

Элементы, представляющиеся как осевые (такие как стены, балки и колонны), обладают относительной траекторией направления, отображающей параметры привязок и показывающей, какая грань элемента является базовой для набора слоев материала.

Понятие траектория соединения относится к этой сущности как показано в Таблице 186.

RelatedElement (связанный элемент)	Описание
IfcWall	Стены с равным или более низким приоритетом связаны в RelatedElement.

Таблица 186 — возможность соединения пути IfcWall

Пространственное включение (Spatial Containment)

Понятие пространственного включения задает отношение физических элементов, таких как конструктивные элементы здания, протяженные элементы или элементы отделки с их представлением как пространственных конструктивных элементов.

Любой подтип сущности IfcElement может иметь два вида взаимосвязей: во-первых – позицию в общей иерархии (применяемую в большинстве случаев) и во-вторых – собственную внутри сборочного элемента (опционально).

Понятие пространственное включение относится к этой сущности как показано в Таблице 187.

Structure (Конструкция)	Описание
IfcBuildingStorey	Пространственный контейнер по умолчанию
IfcBuilding	Пространственный контейнер для элемента в случае, если он не может быть назначен строительному этажу
IfcSite	Пространственный контейнер для элемента в случае, если он помещен на площадке (за пределами здания)

Таблица 187 — IfcWall пространственное сдерживание

IfcWall, как любой подтип IfcBuildingElement, может участвовать альтернативно в одном из двух различных отношений включения:

- Пространственное включение (определенно здесь), или
- композиция элементов.

Ось 2D геометрии (Axis 2D Geometry)

Осевое 'Axis' представление с использованием представления 'Curve2D' преимущественно используется для выравнивания набора слоев материала относительно геометрического представления стандартных стен

Понятие Ось 2D геометрии относится к этой сущности как показано в Таблице 188.

Идентификатор	Тип	Пункты	Описание
Ось	Curve2D	IfcBoundedCurve	Ось стены.

Таблица 188 — ось 2D геометрии IfcWall

Примечание – 'Ось' не используется, чтобы определить местонахождение набора слоя материала, только подтип IfcWallStandardCase, который обеспечивается этой способностью.

Геометрия поверхностей (Surface Geometry)

Элементы могут иметь вид Поверхности 'Surface', описывающей внешнюю поверхность объекта.

Понятие геометрия поверхностей относится к этой сущности.

Примечание – 'Поверхность' может использоваться, чтобы определить модель здания с помощью геометрии поверхностей (например, в аналитических целях, или при снижении уровня детализации представления).

Твердотельная геометрия (Body SweptSolid Geometry)

Геометрические тела вида SweptSolid представляют собой твердотельные трехмерные объекты, создаваемые за счет выдавливания (Extrude) или вращения плоских площадей.

Понятие геометрии тел SweptSolid относится к этой сущности.

Следующие дополнительные ограничения относятся к представлению 'SweptSolid':

- Solid: IfcExtrudedAreaSolid требуется,
- Profile: IfcArbitraryClosedProfileDef требуется.
- Extrusion: Все направления вытеснения должны быть поддержаны.

Примечание – если можно описать тело стены вертикальным вытеснением многоугольного следа с постоянной толщиной вдоль оси (где вертикальный = в направление глобальной Оси Z), то подтип IfcWallStandardCase должен использоваться. Если вытеснение не равно глобальному Z, то IfcWall должен использоваться.

Геометрия отсечения тела

Понятие геометрии отсечения тела, относится к этой сущности.

Освобождающий элемент

Понятие Освобождающий элемент относится к этой сущности как показано в Таблице 189.

Назначение продукта

Понятие Назначение продукта относится к этой сущности как показано в Таблице 190.

Тип	Описание
IfcStructuralSurfaceMember	Идеализированный участник конструкции, соответствующий стене.
IfcTask	Задача для операции по отношению к стене.

Таблица 190 — назначение продукта IfcWall

Спецификация XSD:

```
<xs:element name="IfcWall" type="ifc:IfcWall"
substitutionGroup="ifc:IfcBuildingElement" nillable="true"/>
 <xs:complexType name="IfcWall">
  <xs:complexContent>
   <xs:extension base="ifc:IfcBuildingElement">
    <xs:attribute name="PredefinedType" type="ifc:IfcWallTypeEnum" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

EXPRESS Спецификация:

```
ENTITY IfcWall
 SUPERTYPE OF(ONEOF(IfcWallElementedCase, IfcWallStandardCase))
 SUBTYPE OF IfcBuildingElement;
  PredefinedType   :OPTIONAL IfcWallTypeEnum;
 WHERE
  CorrectPredefinedType: NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
                         IfcWallTypeEnum.USERDEFINED) OR ((PredefinedType =
                         IfcWallTypeEnum.USERDEFINED) AND EXISTS
                         (SELF\IfcObject.ObjectType));
  CorrectTypeAssigned  :(SIZEOF(IsTypedBy) = 0) OR
                         ('IFCSHAREDBLDGELEMENTS.IFCWALLTYPE' IN
                         TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

Диаграмма EXPRESS-G

Определения свойств:

PredefinedType Предопределенный универсальный тип для стены, которая определена в перечислении. Может быть набор свойств, данный определенно для предопределенных типов.

Примечание – *PredefinedType* должен использоваться, если никакому IfcWallType не назначен, обеспечивая его собственный *IfcWallType. PredefinedType*.

ИЗМЕНЕНИЕ в IFC4: свойство было добавлено в конец определения сущности.

Формальные суждения:

CorrectPredefinedType Если ни одно свойство *PredefinedType* не назначено (например, потому что IfcWallType связан), или унаследованное свойство, который нужно предоставить *ObjectType*, если *PredefinedType* будет установлен в USERDEFINED.

CorrectTypeAssigned Если ни один объекта типа ситена не связанан, т.е. обратная связь *IsTypedBy* не обеспечена, или связанный объект типа должен иметь тип IfcWallType.

Граф наследования:

ENTITY IfcWall
ENTITY IfcRoot
GlobalId :IfcGloballyUniqueId;
OwnerHistory :OPTIONAL IfcOwnerHistory;
Name :OPTIONAL IfcLabel;
Description :OPTIONAL IfcText;
ENTITY IfcObjectDefinition
ENTITY
HasAssignments :SET OF IfcRelAssigns FOR RelatedObjects;
Nests :SET [0:1] OF IfcRelNests FOR RelatedObjects;
IsNestedBy :SET OF IfcRelNests FOR RelatingObject;
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 IfcLabel;
ENTITY
IsDeclaredBy :SET [0:1] OF IfcRelDefinesByObject FOR RelatedObjects;
Declares :SET OF IfcRelDefinesByObject FOR RelatingObject;
IsTypedBy :SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;
IsDefinedBy :SET OF IfcRelDefinesByProperties FOR RelatedObjects;
ENTITY IfcProduct
ObjectPlacement :OPTIONAL IfcObjectPlacement;
Representation :OPTIONAL IfcProductRepresentation;
ENTITY
ReferencedBy :SET OF IfcRelAssignsToProduct FOR RelatingProduct;

ENTITY IfcElement
Tag :OPTIONAL IfcIdentifier;
ENTITY
FillsVoids :SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo :SET OF IfcRelConnectsElements FOR RelatingElement;
IsInterferedByElements :SET OF IfcRelInterferesElements FOR RelatedElement;
InterferesElements :SET OF IfcRelInterferesElements FOR RelatingElement;
HasProjections :SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures :SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasOpenings :SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization: SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries :SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom :SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure :SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElement
ENTITY
HasCoverings :SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
ENTITY IfcWall
PredefinedType :OPTIONAL IfcWallTypeEnum;
END_ENTITY;

Пример описания: IfcWindow (Окно)

Окно - строительный элемент, который преимущественно используется, чтобы обеспечить естественный свет и свежий воздух. Возможно вертикальное открытие и горизонтальное открытие, такое как окна в крыше или легкие купола. Возможно решение с покачиванием, поворотом, скольжением, или автоматически вращающимися створками и зафиксированными створками. Окно состоит из наличника и одной или нескольких створок.

Примечание - Определение по данным ISO 6707-1 Конструкция для закрытия вертикального или вертикального проема в стене или крыше, которая пропускает свет и может пропускать свежий воздух.

IfcWindow определяет подробные сведения для окна, вставленного в пространственный контекст проекта. Окно может:

- Быть вставленным в IfcOpeningElement, используя отношения IfcRelFillsElement, при условии что у IfcWindow есть обратный признак *FillsVoids*,
- быть частью сборки элемента, часто IfcCurtainWall, использует отношение IfcRelAggregates, при условии что есть обратный признак *Decomposes (Разложение)*.
- или быть окном «со свободным положением», при условии что есть у IfcWindow нет обратный признака *FillsVoids* или *Decomposes (Разложение)*.

Примечание - определения вида, или соглашения лица, осуществляющего внедрение могут ограничить отношения, разрешив включать одно окно (или дверь) в один проем.

Есть две сущности для определения окна:

- IfcWindowStandardCase используется для всех случаев окон, у которых есть представление формы 'Profile' ('Профиль'), определяюшее набор параметров формы наличника и рамы. Дополнительно требуется предоставление IfcWindowType, которое ссылается на один IfcWindowLiningProperties и на многие IfcWindowPanelProperties.
- IfcWindow, используется для всех других случаев окон, особенно для окон, имеющих только 'Brep' или геометрию 'SurfaceModel', не применяя параметры формы.

Фактический параметр окна и/или его формы определен в IfcWindow как в случае определения (или случай проекта), или IfcWindowType как возникновении определения (или в случае проекта). Следующие параметры даны:

- в IfcWindow или IfcWindowStandardCase для возникновения определенных параметров. IfcWindow определяет:
- ширину и высоту окна
- направление открытия окна (осью Y *ObjectPlacement*)
- в IfcWindowType, с которым IfcWindow связан обратной связью *IsDefinedBy*, указывающий на IfcRelDefinesByType для параметров типа, характерных для всех случаев того же самого типа.

- тип разделения (глухое, двустворчатое, трехстворчатое, и др.),
- тип работы (колебание, наклон и поворот, центр вращается, зафиксированный случай и т.д.),
- положение петель створки окна (при помощи двух различных стилей для правых и левых вводных окон)
- тип строительного материала
- определенные свойства для наличников IfcWindowLiningProperties
- определенные свойства для створок IfcWindowPanelProperties

ИСТОРИЯ: новая сущность в IFC1.0.

ИЗМЕНЕНИЕ в IFC4: свойства *PredefinedType* и *OperationType* добавлены, тип объекта для прменения был изменено на IfcDoorType.

Представление Parameteric, используя параметры в IfcWindowType

Параметры, которые определяют форму IfcWindow, даны в IfcWindowType и наборах свойств, которые включены в IfcWindowType. IfcWindow только определяет локальное размещение. Полный размер IfcWindow, который будет использоваться, чтобы применить свойства наличника или створки, обеспеченные IfcWindowType, определеные IfcShapeRepresentation с RepresentationIdentifier = 'Profile' ('Профиль'). Только в случае IfcWindow, введенного в IfcOpeningElement, используя отношение IfcRelFillsElement, имея горизонтальное вытеснение (вдоль оси Y IfcDoor), полный размер определен профилем вытеснения IfcOpeningElement.

Рисунок 1.6 иллюстрирует вставку окна в IfcOpeningElement, создавая случай IfcWindow с *PartitioningType = DoublePanelHorizontal*. Параметры *OverallHeight* и *OverallWidth* показывают степень окна в положительном Z и Оси X местного размещения окна. Наличник и фрамуга созданы данными параметрами.

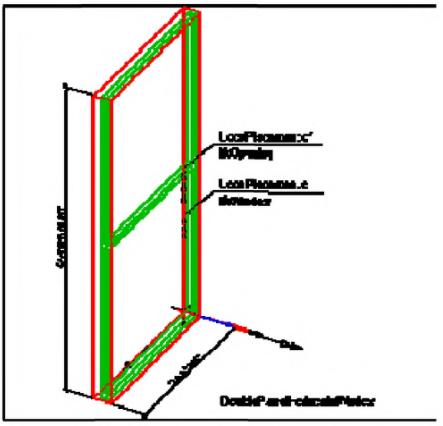


Рисунок 1.6— размещение Окна

Рисунок 1.7 иллюстрирует заключительное окно (DoublePanelHorizontal) первым групповым наличием *PanelPosition = TOP (ВЕРШИНА)*, *OperationType = BOTTOMHUNG* и наличие втрой створки *PanelPosition = BOTTOM (ОСНОВАНИЕ)* и *OperationType = TILTANDTURNLEFTHAND*.

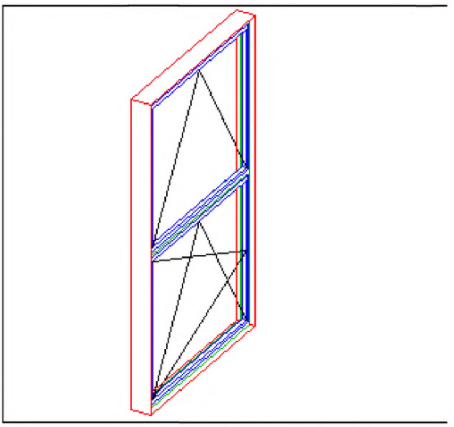


Рисунок 1.7 — плоскости Окна

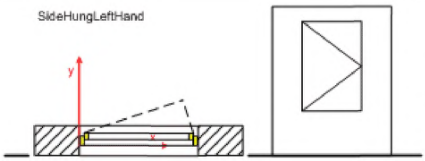
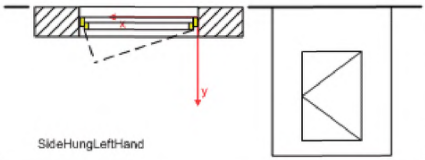
Воздействие типа окна на оконный проем

Параметры, который определяет форму IfcWindow, даны в IfcWindowType и наборах свойств, которые включены в IfcWindowType. IfcWindow только определяет

локальное размещение, которое определяет направление открытия окна. Полное расположение IfcWindow определено его *IfcWindowType. PartitioningType*. У каждой створки окна есть свой собственный тип работы, обеспеченный *IfcWindowPanelProperties. OperationType*. Все створки окна, как предполагается, открываются в том же самом направление (при необходимости для конкретного типа створки окна). Обратная сторона (открывается ли окно налево или вправо) определена IfcWindowPanelProperties. *OperationType*.

Примечание - Существуют различные соглашения в разных странах о том, как показать символическое представление работы створок окна («треугольники»). Или, как показано с внешней стороны, или с внутренней стороны. Следующие данные показывают символику с внешней стороны (соглашение, которое используется преимущественно в Европе).

Рисунок 1.8 иллюстрирует типы работы створок окна.

	Створка окна (для стороны подвески окна) всегда открывается в направление положительной Оси Y локального размещения. Определение того, открывается ли окно налево или вправо сделано в *IfcWindowPanelProperties. OperationType*. Здесь это - левая сторона открывания окна, данное*OperationType* = SideHungLeftHand.
	Если окно должно открыться с другой стороны, то локальное размещение должно быть изменено.

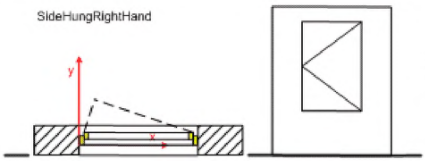
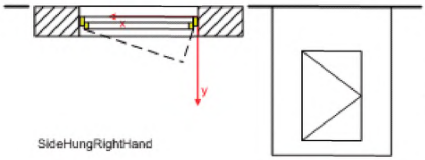
	Это - все еще левое повешенное окно, данное *IfcWindowPanelProperties.OperationType* = SideHungLeftHand.
	Если створка окна (для стороны подвески окна) открывается вправо, стиль створки конкретного окна должен использоваться (здесь *IfcWindowPanelProperties.OperationType* = SideHungRightHand), и это всегда открывается в направление положительной Оси Y локального размещения.
	Если окно должно открыться с другой стороны, то локальное размещение должно быть изменено. Это - все еще повешенное окно права, данное *IfcWindowPanelProperties.OperationType* = SideHungRightHand.

Рисунок 1.8 — Работа окна

Определения общего использования

Следующие понятия унаследованы в супертипах:

- IfcRoot: Id**ENTITY**, Revision Control
- IfcElement: Box Geometry, FootPrint Geometry, Body SurfaceOrSolidModel Geometry, Body SurfaceModel Geometry, Body Tessellation Geometry, Body Brep Geometry, Body AdvancedBrep Geometry, Body CSG Geometry, Mapped Geometry

- IfcBuildingElement: Product Assignment, Surface 3D Geometry

Пример диаграммы

Присвоение типа объекту

Понятие Object Typing относится к этой сущности как показано в Таблице 195.

Тип
IfcWindowType

Таблица— присвоение типа объекту IfcWindow

Наборы свойств для объектов

Понятие Наборы свойств для объектов относятся к этой сущности как показано в Таблице.

PredefinedType (предодределенный тип)	Имя
	Pset_WindowCommon
	Pset_ConcreteElementGeneral
	Pset_PrecastConcreteElementFabrication
	Pset_PrecastConcreteElementGeneral
	Pset_Condition
	Pset_EnvironmentalImpactIndicators
	Pset_EnvironmentalImpactValues
	Pset_ManufacturerOccurrence
	Pset_ManufacturerTypeInformation
	Pset_PackingInstructions
	Pset_ServiceLife
	Pset_Warranty

Таблица — наборы свойств для объектов IfcWindow

Наборы количеств

Понятие Наборов количеств относится к этой сущности как показано в Таблице.

Имя
Qto_WindowBaseQuantities

Таблица— наборы количеств IfcWindow

Применяемые материалы

Понятие Применяемые материалы относится к этой сущности как показано в Таблице.

Имя	Описание
Наличник	Указывает, что существенный элемент относится к наличнику окна.
Рама	Указывает, что существенный элемент относится к створкам окна; при условии если информация о материале наличника относится также и к створкам.
Остекление	Указывает, что существенный элемент относится к части остекления.

Таблица— Применяемые материалы IfcWindow

Материал IfcWindow определен IfcMaterialConstituentSet или как отступает IfcMaterial и присоединяется IfcRelAssociatesMaterial. *RelatingMaterial*. Это доступно обратными отношениями *HasAssociations*.

Если ссылаются на единственный IfcMaterial, это относится к наличнику и раме окна.

Пространственное включение

Понятие Пространственного включения относится к этой сущности как показано в Таблице.

Конструкция
IfcBuildingStorey
IfcBuilding
IfcSite

Таблица — пространственное включение IfcWindow

IfcWindow, как любой подтип IfcBuildingElement, может участвовать альтернативно в одном из двух различных отношений включения:

- Пространственное включение (определено здесь), или
- Композиция элемента.

IfcWindow может также быть связан с IfcOpeningElement, в который он помещен как наполнитель. В этом случае отношения пространственного включения должны быть обеспечены, как показано на рисунке 1.9.

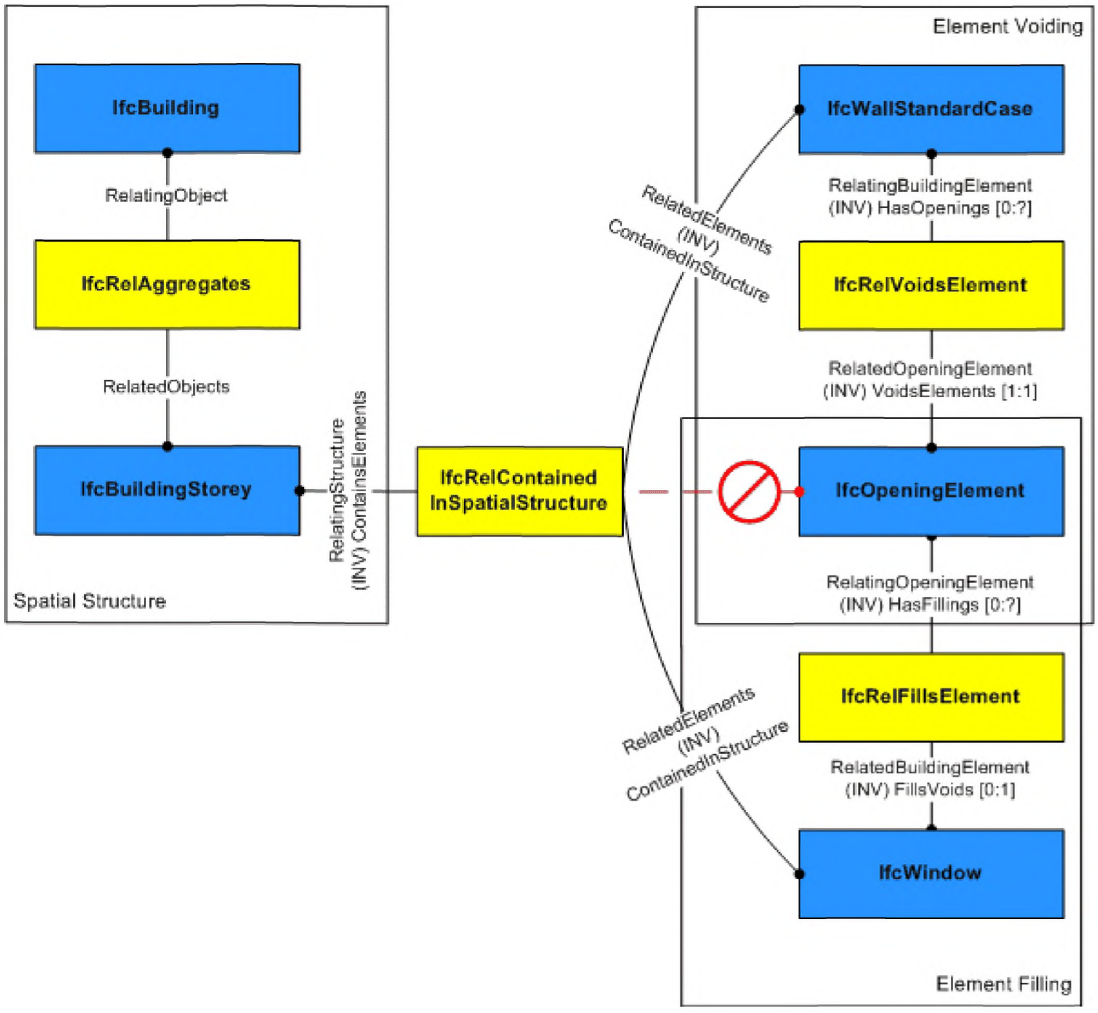


Рисунок 1.9 —пространственное включение окна

Примечание - Включение должно быть определено независимо от заполняющихся отношений, то есть, даже если IfcWindow будет заполнением проема, установленного IfcRelFillsElement, это также содержится в пространственной конструкции IfcRelContainedInSpatialStructure.

Размещение продукта

Понятие размещение продукта относится к этой сущности как показано в Таблице.

Тип	Родственник	Описание
IfcLocalPlacement	IfcLocalPlacement	Относительное размещение согласно положению и вращение относительно контейнера.
IfcLocalPlacement		Абсолютное размещение согласно положению и вращение мировой системы координат.
IfcGridPlacement		Размещение согласно пересечению сетки.

Таблица — размещение продукта IfcWindow

Следующее ограничение введено:

1. Отношения *PlacementRelTo* по отношению к IfcLocalPlacement должны указать на локальное размещение того же самого элемента (если дано), в котором IfcWindow используется в качестве заполнения (обычно IfcOpeningElement), в соответствии с отношениями IfcRelFillsElement.
2. Если IfcWindow не будет введен в IfcOpeningElement, то отношения *PlacementRelTo* по отношению к IfcLocalPlacement должны указать (если дано) на локальное размещение того же самого IfcSpatialStructureElement, который используется в признаке обратного свойтва *ContainedInStructure* или к пространственному элементу конструкции, на который ссылаются, в более высоком уровне.
3. Если относительное размещение не используется, абсолютное размещение определено в пределах мировой системы координат.

Примечание - Размещение продукта используется, чтобы определить направление открытия окна.

Геометрия профиля

Понятие Геометрия профиля относится к этой сущгости как показано в Таблице.

Идентификатор	Тип	Пункты

Таблица— геометрия профиля IfcWindow

Профиль окна представлен трехмерной закрытой кривой в пределах особого представления формы. Профиль используется, чтобы применить параметр параметрического представления окна. Должны использоваться следующие значения свойств для IfcShapeRepresentation, поддерживающего это геометрическое представление:

- *RepresentationIdentifier*: 'Profile' ('Профиль')
- *RepresentationType*: 'Curve3D', только единственная закрытая кривая должна содержаться в наборе *IfcShapeRepresentation.Items*.

Представление 'Profile' ('Профиль') должно быть обеспечено если:

- параметрическое представление должно быть применено к окну И
- окно - 'свободное положение', или
- проем, в которое вставлено окно, не вытеснено горизонтально (т.е. где профиль проема не соответствует профилю окна),

Геометрия 3D профиля

Понятие Геометрия 3D профиля относится к этой сущности.

Спецификация XSD:

```
      <xs:element name="IfcWindow" type="ifc:IfcWindow"
substitutionGroup="ifc:IfcBuildingElement" nillable="true"/>
 <xs:complexType name="IfcWindow">
  <xs:complexContent>
   <xs:extension base="ifc:IfcBuildingElement">
    <xs:attribute name="OverallHeight"
type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="OverallWidth"
type="ifc:IfcPositiveLengthMeasure" use="optional"/>
    <xs:attribute name="PredefinedType"
type="ifc:IfcWindowTypeEnum" use="optional"/>
    <xs:attribute name="PartitioningType"
type="ifc:IfcWindowTypePartitioningEnum" use="optional"/>
    <xs:attribute name="UserDefinedPartitioningType"
type="ifc:IfcLabel" use="optional"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
```

Спецификация EXPRESS:

```
ENTITY IfcWindow
 SUPERTYPE OF(IfcWindowStandardCase)
 SUBTYPE OF IfcBuildingElement;
  OverallHeight          OPTIONAL IfcPositiveLengthMeasure;
  OverallWidth           OPTIONAL IfcPositiveLengthMeasure;
  PredefinedType         OPTIONAL IfcWindowTypeEnum;
  PartitioningType       OPTIONAL IfcWindowTypePartitioningEnum;
```

```
    UserDefinedPartitionin      OPTIONAL IfcLabel;
gType
    WHERE
    CorrectStyleAs      (SIZEOF(IsTypedBy) = 0) OR
signed                  ('IFCSHAREDBLDGELEMENTS.IFCWINDOWTYPE' IN
                        TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));
    END_ENTITY;
```

Диаграмма EXPRESS-G

Определения свойств:

OverallHeight Полная мера высоты, это отражает измерение Z ограничивающего прямоугольника, включая открытие окна. Если опущено, *OverallHeight* должен быть взят из геометрического представления *IfcOpening*, в который вставлено окно.

Примечание - тело окна могло бы быть более высоким тогда открытие окна (например, в случаях, где выравнивание окна включает кожух). В этих случаях *OverallHeight* нужно давать как высоту проема окна, и не как полную высоту выравнивания окна.

OverallWidth Полная мера ширины, это отражает X измерений ограничивающего прямоугольника, включая проем окна. Если опущено, *OverallWidth* должен быть взят из геометрического представления *IfcOpening*, в который вставлено окно.

Примечание - тело окна могло бы быть более широким тогда открытие окна (например, в случаях, где наличник окна включает корпус). В этих случаях

OverallWidth нужно давать как ширину проема окна, а не как полная ширина наличников окна.

PredefinedType Предопределенный универсальный тип для окна, которое определено в перечислении. Может быть набор свойств, специфичный для предопределенных типов.

Примечание – только *PredefinedType* должен использоваться, если никакому IfcWindowType не назначат, обеспечивая его собственный *IfcWindowType. PredefinedType*.

ИЗМЕНЕНИЕ в IFC4: свойство было добавлено в конце определения сущности.

PartitioningType Определение типа общего расположения окна с точки зрения разделения створок.

Примечание – только *PartitioningType* должен использоваться, если никакой объект типа IfcWindowType не будет назначен, обеспечивая его собственный *IfcWindowType. PartitioningType*.

ИЗМЕНЕНИЕ в IFC4: свойство было добавлено в конце определения сущности.

UserDefinedPartitioningType Указатель определенного пользователем типа разделения при условии, если он должен быть обеспечен, если значение *PartitioningType* будет установлено в USERDEFINED.

Формальные суждения:

CorrectStyleAssigned Или нет никакого дверного объекта типа, связанного, т.е. обратная связь *IsTypedBy* не обеспечена, или связанный объект типа должен иметь тип IfcWindowStyle.

ЗАМЕЧАНИЕ - осуждаемый тип IfcWindowStyle все еще включен по причинам обратной совместимости.

Граф наследования:

ENTITY IfcWindow

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

ENTITY

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

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** IfcLabel;

ENTITY

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;

Representation : **OPTIONAL** IfcProductRepresentation;

ENTITY

ReferencedBy **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcIdentifier;

ENTITY

FillsVoids **SET** [0:1] OF IfcRelFillsElement **FOR** RelatedBuildingElement;

ConnectedTo **SET OF** IfcRelConnectsElements **FOR** RelatingElement;

IsInterferedByElements **SET OF** IfcRelInterferesElements **FOR** RelatedElement;

InterferesElements **SET OF** IfcRelInterferesElements **FOR** RelatingElement;

HasProjections **SET OF** IfcRelProjectsElement **FOR** RelatingElement;

ReferencedInStructures **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

HasOpenings **SET OF** IfcRelVoidsElement **FOR** RelatingBuildingElement;

IsConnectionRealization **SET OF** IfcRelConnectsWithRealizingElements **FOR** RealizingElements;

ProvidesBoundaries **SET OF** IfcRelSpaceBoundary **FOR** RelatedBuildingElement;

ConnectedFrom **SET OF** IfcRelConnectsElements **FOR** RelatedElement;

ContainedInStructure **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcBuildingElement

ENTITY

HasCoverings **SET OF** IfcRelCoversBldgElements **FOR** RelatingBuildingElement;

ENTITY IfcWindow

OverallHeight : **OPTIONAL** IfcPositiveLengthMeasure;

OverallWidth : **OPTIONAL** IfcPositiveLengthMeasure;

PredefinedType : **OPTIONAL** IfcWindowTypeEnum;

PartitioningType : **OPTIONAL** IfcWindowTypePartitioningEnum;

UserDefinedPartitioningType : **OPTIONAL** IfcLabel;

END_ENTITY;

1.2.4 Примеры описания сущностей схем данных доменов

Схемы данных доменов содержат окончательные специализации отраслевых сущностей. Сущности, определенные в этом слое являются автономными и не могут ссылаться на любой другой слой. Каждый домен организует определения сущностей в соответствии с отраслевыми дисциплинами:

- домен архитектурный (IfcArchitectureDomain);
- домен контроля над ходом строительства (IfcBuildingControlsDomain);
- домен электрооборудования (IfcElectricalDomain);
- домен систем отопления, вентиляции и кондиционирования (IfcHvacDomain);
- домен сетей противопожарной защиты (IfcPlumbingFireProtectionDomain);
- домен анализа конструктивных элементов (IfcStructuralAnalysisDomain);

- домен конструктивных элементов (IfcStructuralElementsDomain);
- домен управления строительством (IfcConstructionMgmtDomain).

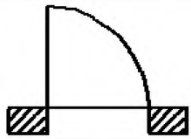
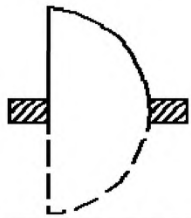
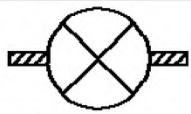
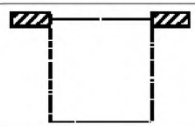
Пример описания: IfcDoorPanelOperationEnum (Типы дверных створок)

Это перечисление определяет основные пути, как работают отдельные дверные панели (показано далее на рисунке).

ИСТОРИЯ: новое перечисление в IFC2.0.

ИЗМЕНЕНИЕ в IFC4: добавлен перечислитель FIXEDPANEL.

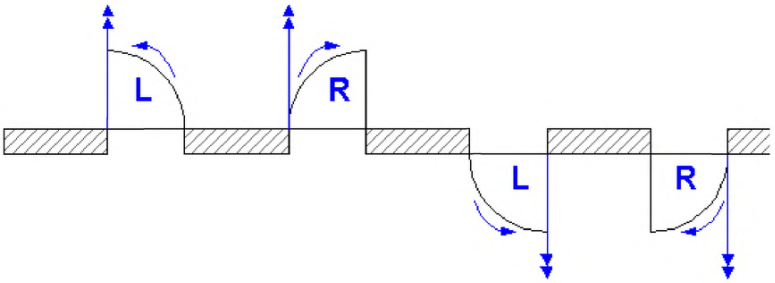
Перечислимые определения изделия:

Перечислитель	Иллюстрация
Покачивание	
Двойное действие	
Скольжение	
Сворачивание	
Вращение	
Свертывание	

Фиксированная створка	 Примечание – Перечислитель добавлен в версии IFC4.
Определенно пользователем	
Не определено	

Рисунок – работа двери

Направление открытия дверных створок задается локальным размещением IfcDoor. Положительная ось Y определяет направление как показано в рисунке.



Рисунок— направление открытия дверной створки

Примечание - иллюстрации (символическое представление) зависят от национальных строительных норм и правил. Эти числа показаны только как иллюстрации.

Спецификация XSD:

```
<xs:simpleType name="IfcDoorPanelOperationEnum">
  <xs:restriction base="xs:string">
   <xs:enumeration value="swinging"/>
   <xs:enumeration value="double_acting"/>
   <xs:enumeration value="sliding"/>
```

<xs:enumeration value="folding"/>
<xs:enumeration value="revolving"/>
<xs:enumeration value="rollingup"/>
<xs:enumeration value="fixedpanel"/>
<xs:enumeration value="userdefined"/>
<xs:enumeration value="notdefined"/>
</xs:restriction>
</xs:simpleType>

Спецификация EXPRESS:

TYPE IfcDoorPanelOperationEnum = **ENUMERATION OF** (
SWINGING,
DOUBLE_ACTING,
SLIDING,
FOLDING,
REVOLVING,
ROLLINGUP,
FIXEDPANEL,
USERDEFINED,
NOTDEFINED);
END_TYPE;

Пример описания: IfcDoorPanelPositionEnum (Типы положений дверных створок)

Это перечисление определяет основные способы описать расположение дверных створок.

ИСТОРИЯ: новое перечисление в IFC2x

На рисунке показано обозначение дверной створки с *PanelPosition* = LEFT и дверной створки с *PanelPosition* = RIGHT в стиле двери с *OperationType* = DOUBLE_DOOR_SINGLE_SWING. Позиция дана как показано в плоскости XZ локального размещения, при взгляде в направлении положительной Оси Y.

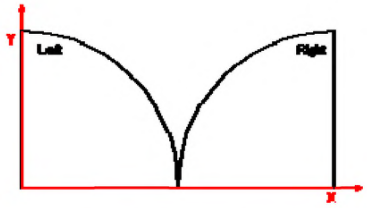


Рисунок — типы положений дверных створок

Спецификация XSD:

```
<xs:simpleType name="IfcDoorPanelPositionEnum">
 <xs:restriction base="xs:string">
  <xs:enumeration value="left"/>
  <xs:enumeration value="middle"/>
  <xs:enumeration value="right"/>
  <xs:enumeration value="notdefined"/>
 </xs:restriction>
</xs:simpleType>
```

Спецификация EXPRESS:

```
TYPE IfcDoorPanelPositionEnum = ENUMERATION OF (
  LEFT,
  MIDDLE,
  RIGHT,
  NOTDEFINED);
END_TYPE;
```

Пример описания: IfcDoorLiningProperties (Свойства дверного наличника)

Дверной наличник является рамой, позволяющей створке двери быть фиксированной в позиции. Дверной наличник используется для навешивания створки двери. Параметры наличника двери (IfcDoorLiningProperties), определяют геометрически соответствующий параметр наличника.

Примечание – IfcDoorLiningProperties должен быть применен только получением параметрического определения 3D формы двери, если признак *IfcDoorType. ParameterTakesPrecedence* является набором TRUE.

IfcDoorLiningProperties включены в список свойств *IfcDoorType. HasPropertySets*. Больше информации о дверном наличнике может быть включено в тот же список *IfcDoorTyype* с помощью другого IfcPropertySet для динамических расширений.

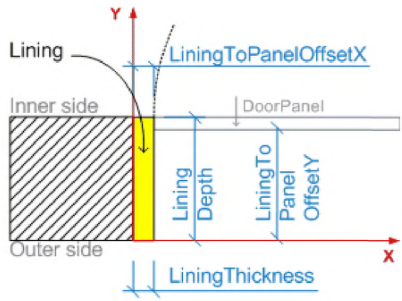
ИСТОРИЯ: новая сущность в IFC2.0. Был переименован из *IfcDoorLining* в IFC2x.

ИЗМЕНЕНИЕ в IFC4: следующие признаки были добавлены: *LiningToPanelOffsetX*, *LiningToPanelOffsetY*. Признак *ShapeAspectStyle* отменяется и больше не должен использоваться. Супертип измененный на новый: IfcPreDefinedPropertySet.

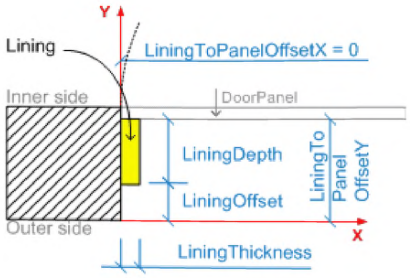
Определения использования геометрии

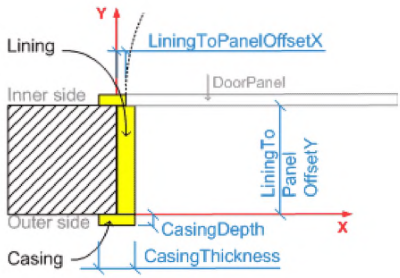
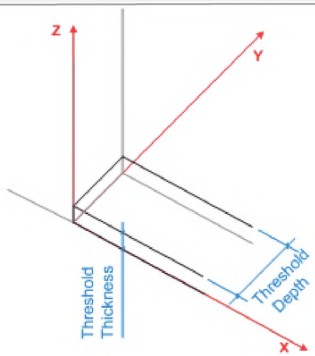
IfcDoorLiningProperties не имеет своего собственного геометрического представления. Однако, определяет параметры, которые могут использоваться для создания формы дверного стиля (который вставляется IfcDoor в пространственный контекст проекта), как показано в рисунке.

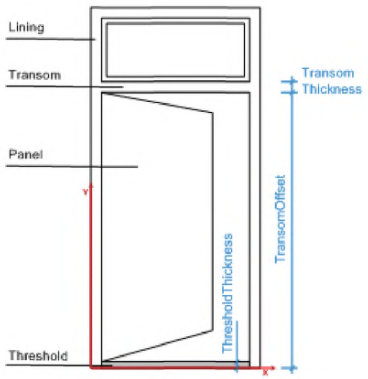
Параметры IfcDoorLiningProperties определяют стандартный дверной наличник, включая (если есть) порог и фрамугу. Внешняя граница наличника определяется параметром возникновения, назначенным в IfcDoor, вставленным в IfcDoorStyle.



Наличник применен слева, правая и верхняя сторона открытия показаны. Параметры:
LiningDepth, если опущено, равный толщине стены - это вступает в силу, если дано значение для *LiningThickness*. Если оба параметра не даны, то нет никакого наличника.

	LiningThickness *LiningToPanelOffsetX* *LiningToPanelOffsetY* Примечание – параметры *LiningToPanelOffsetX* и *LiningToPanelOffsetY* добавлены в версии IFC4.
	Наличник может покрыть только часть проема. *LiningOffset*: задан, если край наличника имеет смещение к оси X локального размещения. Примечание – в дополнение к *LiningOffset*, локальное размещение IfcDoor может уже иметь смещение к краю стены и таким образом переместить наличник вдоль оси Y. Фактическая позиция наличника вычисляется от происхождения локального размещения вдоль положительной оси Y с расстоянием, данным *LiningOffset*.

	Наличник может включать часть, покрывающую стенную поверхность вокруг проема. Может быть закрыта левая, правая и верхняя сторона наличника с обеих сторон стены. Параметры: *CasingDepth* *CasingThickness*
	Наличник может включать порог, покрывающий нижнюю сторону проема. Параметры: *ThresholdDepth* — если опущено, равный толщине стены - это вступает в силу только, если значение для *ThresholdThickness* дано. Если оба параметра не даны, то нет никакого порога. *ThresholdThickness* *ThresholdOffset* (не показанный): задан, если край порога имеет смещение к оси X локального размещения.



Наличник может иметь фрамугу, отделяющую дверную створку от оконной створки. Фрамуга, если задана, определяется:

TransomOffset: параллельный край к оси X локального размещения

TransomThickness

Глубина фрамуги идентична глубине наличника и не задана как отдельный параметр.

Рисунок — свойства наличника двери

Примечание – LiningDepth описывает длину наличника вдоль раскрытия дверного проема. Это может быть дано абсолютной величиной, если дверной наличник имеет определенную глубину в зависимости от дверного стиля. Однако, часто это равно толщине стены. Если тот же дверной стиль используется (например, в случае типа единственной вращающейся двери), но вставляется в различные стены с различными толщинами, необходимо создать специальный дверной стиль для каждой толщины стены. Поэтому разные системы САПР позволяют устанавливать значение в «автоматически выровненный» к толщине стены. Это должно быть изменено путем применения дополнительного признака LiningDepth со значением не назначенный. То же соглашение применяется к *ThresholdDepth*.

Спецификация XSD:

```
<xs:element name="IfcDoorLiningProperties" type="ifc:IfcDoorLiningProperties"
substitutionGroup="ifc:IfcPreDefinedPropertySet" nillable="true"/>
 <xs:complexType name="IfcDoorLiningProperties">
  <xs:complexContent>
   <xs:extension base="ifc:IfcPreDefinedPropertySet">
    <xs:sequence>
     <xs:element name="ShapeAspectStyle" type="ifc:IfcShapeAspect" nillable="true"
```

```
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LiningDepth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
  <xs:attribute name="LiningThickness" type="ifc:IfcNonNegativeLengthMeasure"
use="optional"/>
  <xs:attribute name="ThresholdDepth" type="ifc:IfcPositiveLengthMeasure"
use="optional"/>
  <xs:attribute name="ThresholdThickness" type="ifc:IfcNonNegativeLengthMeasure"
use="optional"/>
  <xs:attribute name="TransomThickness" type="ifc:IfcNonNegativeLengthMeasure"
use="optional"/>
  <xs:attribute name="TransomOffset" type="ifc:IfcLengthMeasure" use="optional"/>
  <xs:attribute name="LiningOffset" type="ifc:IfcLengthMeasure" use="optional"/>
  <xs:attribute name="ThresholdOffset" type="ifc:IfcLengthMeasure" use="optional"/>
  <xs:attribute name="CasingThickness" type="ifc:IfcPositiveLengthMeasure"
use="optional"/>
  <xs:attribute name="CasingDepth" type="ifc:IfcPositiveLengthMeasure" use="optional"/>
  <xs:attribute name="LiningToPanelOffsetX" type="ifc:IfcLengthMeasure"
use="optional"/>
  <xs:attribute name="LiningToPanelOffsetY" type="ifc:IfcLengthMeasure"
use="optional"/>
 </xs:extension>
</xs:complexContent>
</xs:complexType>
```

Спецификация EXPRESS:

Схема EXPRESS-G

Определения свойств:

LiningDepth Глубина дверного наличника, измеренного перпендикулярно к плоскости дверного наличника. Если не задан (и с данным значением к толщине наличника) это указывает корректируемую глубину (т.е. глубина, корректирующаяся к толщине стены, в которую возникновение этого дверного стиля вставляется).

LiningThickness Толщина наличника двери, как описано выше. Если значение *LiningThickness* 0 это обозначает дверь без наличника (все другие параметры наличника должны быть установлены равными NIL в этом случае). Если *LiningThickness* имеет значение NIL, он обозначает, что значение не доступно.

ИЗМЕНЕНИЕ в IFC4: тип данных изменен, чтобы быть IfcNonNegativeLengthMeasure.

ThresholdDepth Глубина (измерение в одной плоскости перпендикуляр к створке двери) дверного порога. Задан только, если дверной наличник включает порог. Если не задан (и с данным значением к толщине порога) это указывает корректируемую глубину (т.е. глубина, корректирующаяся к толщине стены, в которую возникновение этого дверного стиля вставляется).

ThresholdThickness Толщина дверного порога, как объяснено выше. Если значение *ThresholdThickness* равно 0 это обозначает дверь без порога (*ThresholdDepth* должен иметь значение NIL в этом случае). Если *ThresholdThickness* имеет значение NIL, он обозначает, что информация о пороге не доступна.

ИЗМЕНЕНИЕ в IFC4: тип данных изменен, чтобы быть IfcNonNegativeLengthMeasure.

TransomThickness Толщина (ширина в одной плоскости параллельны к створке двери) фрамуги (если обеспечено - т.е. если признак *TransomOffset* установлен), который делит створку двери от глазурования (или окно) выше. Если *TransomThickness* обнуляется (и набор *TransomOffset* к положительной длине), то дверь разделена вертикально в лист и область окна фрамуги без физической рамы.

ИЗМЕНЕНИЕ в IFC4: тип данных изменен на IfcNonNegativeLengthMeasure.

TransomOffset Смещение фрамуги (если задано), которое делит створку двери от глазурования (или окно) выше. Смещение дано от нижней части дверного проема.

LiningOffset Смещение (измерение в одной плоскости перпендикуляр к створке двери) дверного наличника. Смещение дано как расстояние до оси X локального размещения.

ThresholdOffset Смещение (измерение в одной плоскости перпендикуляр к створке двери) дверного порога. Смещение дано как расстояние до оси X локального размещения. Задан только, если дверной наличник включает порог и параметр известен.

CasingThickness Толщина корпуса (измерение в одной плоскости створки двери). Если задан, то это применено одинаково ко всем четырем сторонам смежной стены.

CasingDepth Глубина корпуса (измерение в одной плоскости перпендикуляр к створке двери). Если задан. то это применено одинаково ко всем четырем сторонам смежной стены.

ShapeAspectStyle Указатель на аспект формы, если задан. Аспект формы отражает часть дверной формы, представляющей дверной наличник.

ИЗМЕНЕНИЕ в IFC4: свойство отменяется и больше не

должно использоваться, т.е. значением должен быть NIL.

LiningToPanelOffsetX Смещение между наличником и оконной створкой, измеренной вдоль оси X локального размещения.

ИЗМЕНЕНИЕ в IFC4: новое свойство добавлено в конце определения сущности.

LiningToPanelOffsetY Смещение между наличником и дверной створкой, измеренной вдоль оси Y локального размещения.

ИЗМЕНЕНИЕ в IFC4: новое свойство добавлено в конце определения сущности.

Формальные суждения:

WR31 Любой из двух параметров *LiningDepth* и *LiningThickness* даны, или только *LiningThickness*, тогда *LiningDepth* является переменным. Не действительно, чтобы утверждать только *LiningDepth*.

Примечание – значение NIL в *LiningDepth* указывает дверной стиль с наличником равным толщине стены.

ИЗМЕНЕНИЕ в IFC4: правило исправлено.

WR32 Любой из двух параметров *ThresholdDepth* и *ThresholdThickness* даны, или только *ThresholdThickness*, тогда *ThresholdDepth* является переменным. Это не действительно, чтобы утверждать только *ThresholdDepth*.

Примечание – значение NIL в *ThresholdDepth* указывает дверной стиль с наличником равным толщине стены.

ИЗМЕНЕНИЕ в IFC4: правило исправлено.

WR33 Любой из двух параметров *TransomDepth* и *TransomThickness* даны, или ни один из них.

WR34 Любой из двух параметров *CasingDepth* и *CasingThickness*, даны, или ни один из них.

WR35 IfcDoorLiningProperties должен использоваться только в контексте IfcDoorType.

Примечание – сущность IfcDoorStyle также применима.

Граф наследования:

ENTITY IfcDoorLiningProperties

ENTITY IfcRoot

GlobalId	:	IfcGloballyUniqueId;
OwnerHistory	:	**OPTIONAL** IfcOwnerHistory;
Name	:	**OPTIONAL** IfcLabel;
Description	:	**OPTIONAL** IfcText;

ENTITY IfcPropertyDefinition

ENTITY

HasContext	:	**SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
HasAssociations	:	**SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcPropertySetDefinition

ENTITY

DefinesType	:	**SET OF** IfcTypeObject **FOR** HasPropertySets;
IsDefinedBy	:	**SET OF** IfcRelDefinesByTemplate **FOR** RelatedPropertySets;
DefinesOccurrence	:	**SET** [0:1] OF IfcRelDefinesByProperties **FOR** RelatingPropertyDefinition;

ENTITY IfcPreDefinedPropertySet

ENTITY IfcDoorLiningProperties

LiningDepth	:	**OPTIONAL** IfcPositiveLengthMeasure;
LiningThickness	:	**OPTIONAL** IfcNonNegativeLengthMeasure;
ThresholdDepth	:	**OPTIONAL** IfcPositiveLengthMeasure;
ThresholdThickness	:	**OPTIONAL** IfcNonNegativeLengthMeasure;
TransomThickness	:	**OPTIONAL** IfcNonNegativeLengthMeasure;
TransomOffset	:	**OPTIONAL** IfcLengthMeasure;
LiningOffset	:	**OPTIONAL** IfcLengthMeasure;
ThresholdOffset	:	**OPTIONAL** IfcLengthMeasure;
CasingThickness	:	**OPTIONAL** IfcPositiveLengthMeasure;
CasingDepth	:	**OPTIONAL** IfcPositiveLengthMeasure;

ShapeAspectStyle : **OPTIONAL** IfcShapeAspect;
LiningToPanelOffsetX : **OPTIONAL** IfcLengthMeasure;
LiningToPanelOffsetY : **OPTIONAL** IfcLengthMeasure;
END_ENTITY;

1.2.5 Примеры описания сущностей схем данных ресурсов

Задача слоя схемы данных, определяющего ресурсы, состоит в поддержке всей структуры данных. Сущности и типы, определенные в этом слое, могут ссылаться на все сущности других слоев. В отличие от сущностей в других слоях структура данных для определения ресурсов не может существовать независимо, но может существовать только тогда, когда имеется ссылка (прямая или косвенная) на одну или более сущность IfcRoot (Корневой каталог). Поскольку определение сущности ресурса не имеет понятия идентичности (например, GUID), несколько объектов, ссылающихся на один и тот же экземпляр сущности ресурса, не подразумевают создание отношения.

Например, две ломаных линии (IfcPolyline), ссылающихся на тот же экземпляр сущности ресурса точки (IfcCartesianPoint), и две ломаных линии использующих различные экземпляры идентичных точек (например, имеющих координаты 0,0,0) в рамках концепции являются семантически эквивалентными.

Примечание – слой определения ресурсов не следует путать с сущностями ресурсов строительства (IfcResource подтипы). В то время как термины схожи, понятия не связаны.

Схемы данных ресурсов:

- ресурс участников (IfcActorResource);
- ресурс подтверждений (IfcApprovalResource);
- ресурс ограничений (IfcConstraintResource);
- ресурс затрат (IfcCostResource);
- ресурс времени (IfcDateTimeResource);
- ресурс внешних ссылок (IfcExternalReferenceResource);

- ресурс геометрических ограничений (IfcGeometricConstraintResource);
- ресурс геометрических моделей (IfcGeometricModelResource);
- ресурс геометрических представлений (IfcGeometryResource);
- ресурс материала (IfcMaterialResource);
- ресурс единиц измерения (IfcMeasureResource);
- ресурс форм представлений (IfcPresentationAppearanceResource);
- ресурс определения представлений (IfcPresentationDefinitionResource);
- ресурс структур представлений (IfcPresentationOrganizationResource);
- ресурс профилей (IfcProfileResource);
- ресурс свойств (IfcPropertyResource);
- ресурс количеств (IfcQuantityResource);
- ресурс вида представления (IfcRepresentationResource);
- ресурс нагрузок (IfcStructuralLoadResource);
- ресурс топологий (IfcTopologyResource);
- ресурс утилит (IfcUtilityResource).

Пример описания: IfcApprovalResource (ресурс подтверждений)

Схема *IfcApprovalResource* в слое ресурса определяет фундаментальные понятия для подтверждений, которые будут назначены любому объекту или типу объекта или определению свойства в модели IFC, полученной из IfcRoot. Назначение делается, используя IfcRelAssociatesApproval в схеме *IfcControlExtension*. Кроме того, подтверждения могут быть назначены определенным объектам уровня ресурса (документы, свойства, люди/организации или материалы), используя IfcResourceApprovalRelationship.

ИСТОРИЯ: новая схема в IFC2x.

Пример описания: IfcApproval (подтверждение)

IfcApproval представляет информацию о процессах подтверждения плана, дизайна, предложения, или заявки на изменение в проекта управления

оборудованием или управлении строительством. На IfcApproval ссылается IfcRelAssociatesApproval в схеме *IfcControlExtension*, и таким образом можно связать со всеми подтипами IfcRoot. Подтверждение может также быть дано при использовании объектов ресурса IfcResourceApprovalRelationship

ИСТОРИЯ: новая сущность в IFC2.0

ИЗМЕНЕНИЕ в IFC4: Иидентификатор свойств и Имя сделали дополнительными, поэтому правило добавлено, чтобы требовать по крайней мере одно из подтверждений. Добавлены обратные свойства ApprovedObjects, ApprovedResources и HasExternalReferences. Удалены обратные свойства (более общие отношения через обратный ApprovedResources, который будет использоваться вместо этого).

Спецификация XSD:

```
<xs:element name="IfcApproval" type="ifc:IfcApproval"
substitutionGroup="ifc:ENTITY" nillable="true"/>
<xs:complexType name="IfcApproval">
<xs:complexContent>
<xs:extension base="ifc:ENTITY">
<xs:sequence>
<xs:element name="RequestingApproval" nillable="true" minOccurs="0">
<xs:complexType>
<xs:group ref="ifc:IfcActorSelect"/>
</xs:complexType>
</xs:element>
<xs:element name="GivingApproval" nillable="true" minOccurs="0">
<xs:complexType>
<xs:group ref="ifc:IfcActorSelect"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="Identifier" type="ifc:IfcIdentifier" use="optional"/>
```

```
  <xs:attribute name="Name" type="ifc:IfcLabel" use="optional"/>
  <xs:attribute name="Description" type="ifc:IfcText" use="optional"/>
  <xs:attribute name="TimeOfApproval" type="ifc:IfcDateTime" use="optional"/>
  <xs:attribute name="Status" type="ifc:IfcLabel" use="optional"/>
  <xs:attribute name="Level" type="ifc:IfcLabel" use="optional"/>
  <xs:attribute name="Qualifier" type="ifc:IfcText" use="optional"/>
 </xs:extension>
</xs:complexContent>
</xs:complexType>
```

Спецификация EXPRESS:

ENTITY IfcApproval**;**

Identifier	:	**OPTIONAL** IfcIdentifier;
Name	:	**OPTIONAL** IfcLabel;
Description	:	**OPTIONAL** IfcText;
TimeOfApproval	:	**OPTIONAL** IfcDateTime;
Status	:	**OPTIONAL** IfcLabel;
Level	:	**OPTIONAL** IfcLabel;
Qualifier	:	**OPTIONAL** IfcText;
RequestingApproval	:	**OPTIONAL** IfcActorSelect;
GivingApproval	:	**OPTIONAL** IfcActorSelect;

ENTITY

HasExternalReferences	:	**SET OF** IfcExternalReferenceRelationship **FOR** RelatedResourceObjects;
ApprovedObjects	:	**SET OF** IfcRelAssociatesApproval **FOR** RelatingApproval;
ApprovedResources	:	**SET OF** IfcResourceApprovalRelationship **FOR** RelatingApproval;
IsRelatedWith	:	**SET OF** IfcApprovalRelationship **FOR** RelatedApprovals;
Relates	:	**SET OF** IfcApprovalRelationship **FOR** RelatingApproval;

WHERE

HasIdentifierOrName : EXISTS (Identifier) OR EXISTS (Name);

END_ENTITY;

Диаграмма EXPRESS-G

Определения свойств:

Identifier : Поддающийся использованию компьютером идентификатор, который известен как подтвержденный.

Name : Имя, данное подтверждению.

Description : Общее текстовое описание дизайна, задачи работы, плана, и т.д. которые подтверждаются.

TimeOfApproval : Дата и время подтверждения.
ИЗМЕНЕНИЕ в IFC4: тип данных свойства изменился на IfcDateTime в соотвесвие с ISO 8601, переименован из ApprovalDateTime и сделан OPTIONAL.

Status : Результат или текущее состояние подтверждения, например, Требуемый, Обработанный, Подтвержденный, Не Подтвержденный.

Level : Уровень подтверждения, например, Черновой или Завершенный проект.

Qualifier : Текстовое описание специальных ограничений или условий для подтверждения.

RequestingApproval : Актер, который действует в роли, определенной в IfcOrganization или индивидуально в IfcPerson, и просит подтверждение.

ИЗМЕНЕНИЕ в IFC4: новое свойство для замены запроса подтверждения IfcApprovalActorRelationship (удаляемый).

GivingApproval : Актер, который действует в роли, определенной в IfcOrganization или индивидуально в IfcPerson, и дает подтверждение.

ИЗМЕНЕНИЕ в IFC4: новое свойство для замены предоставления подтверждения IfcApprovalActorRelationship (удаляемый).

HasExternalReferences : Ссылка на внешние ссылки, например, библиотека, классификация или информация о документе, которые связаны с подтверждением.

ИЗМЕНЕНИЕ в IFC4: новое обратное свойство.

ApprovedObjects : Ссылка на случаи IfcRelAssociatesApproval, связывающие это подтверждение с объектам (подтипы IfcRoot)

ApprovedResources : Набор отношений, которыми объекты ресурса подтверждены этим подтверждением.

IsRelatedWith : Набор отношений, которыми это подтверждение связано с другими.

Relates : Набор отношений, которыми другие подтверждения связаны с этим.

Формальные суждения:

HasIdentifierOrName : Или Идентификатор или Имя (или оба), которыми определяется подтверждение должны быть заданы.

Граф наследования:

ENTITY IfcApproval

ENTITY IfcApproval

Identifier : **OPTIONAL** IfcIdentifier;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

TimeOfApproval : **OPTIONAL** IfcDateTime;

Status : **OPTIONAL** IfcLabel;

Level : **OPTIONAL** IfcLabel;

Qualifier : **OPTIONAL** IfcText;

RequestingApproval : **OPTIONAL** IfcActorSelect;

GivingApproval : **OPTIONAL** IfcActorSelect;

ENTITY

HasExternalReferences : **SET OF** IfcExternalReferenceRelationship **FOR** RelatedResourceObjects;

ApprovedObjects : **SET OF** IfcRelAssociatesApproval **FOR** RelatingApproval;

ApprovedResources : **SET OF** IfcResourceApprovalRelationship **FOR** RelatingApproval;

IsRelatedWith : **SET OF** IfcApprovalRelationship **FOR** RelatedApprovals;

Relates : **SET OF** IfcApprovalRelationship **FOR** RelatingApproval;

END_ENTITY;

Пример описания: IfcApprovalRelationship (отношение подтверждения)

IfcApprovalRelationship связывает подтверждение (одно связанное подтверждение или одно или несколько связанных подтверждений), каждое имеющий различный статус или уровень как, процесс подтверждения или подтверждения объектов в развивитии.

ИСТОРИЯ: новая сущность в IFC2x2.

ИЗМЕНЕНИЕ в IFC4: подтип IfcResourceLevelRelationship, последовательность свойств изменена.

Спецификация XSD:

<xs:element name="IfcApprovalRelationship" type="ifc:IfcApprovalRelationship" substitutionGroup="ifc:IfcResourceLevelRelationship" nillable="true"/>
<xs:complexType name="IfcApprovalRelationship">
<xs:complexContent>
<xs:extension base="ifc:IfcResourceLevelRelationship">
<xs:sequence>
<xs:element name="RelatingApproval" type="ifc:IfcApproval" nillable="true"/>
<xs:element name="RelatedApprovals">
<xs:complexType>
<xs:sequence>
<xs:element ref="ifc:IfcApproval" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute ref="ifc:itemType" fixed="ifc:IfcApproval"/>
<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>

Спецификация EXPRESS:

ENTITY IfcApprovalRelationship
SUBTYPE OF IfcResourceLevelRelationship**;**
RelatingApproval : IfcApproval;
RelatedApprovals : **SET** [1:?] **OF** IfcApproval;
END_ENTITY;

Диаграмма EXPRESS-G

Определения свойств:

RelatingApproval : Подтвердение, с которым связано другое подтвердение.

RelatedApprovals : Подтвердения, которые связаны с другими (связь) подтвердениями.
ИЗМЕНЕНИЕ в IFC4: количество элементов этого свойства было изменено на SET.

Граф наследования:

ENTITY IfcApprovalRelationship

ENTITY IfcResourceLevelRelationship

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcApprovalRelationship

RelatingApproval : IfcApproval;

RelatedApprovals : **SET** [1:?] **OF** IfcApproval;

END_ENTITY;

Пример описания: IfcResourceApprovalRelationship

IfcResourceApprovalRelationship используется для соединения подтверждения и свойств объекта. Единственное подтверждение может быть дано одному или нескольким пунктам через IfcResourceObjectSelect.

ИСТОРИЯ: новая сущность в IFC4.

Спецификация XSD:

<xs:element name="IfcResourceApprovalRelationship"
type="ifc:IfcResourceApprovalRelationship"
substitutionGroup="ifc:IfcResourceLevelRelationship" nillable="true"/>
<xs:complexType name="IfcResourceApprovalRelationship">

```
<xs:complexContent>
<xs:extension base="ifc:IfcResourceLevelRelationship">
<xs:sequence>
<xs:element name="RelatedResourceObjects">
<xs:complexType>
<xs:group ref="ifc:IfcResourceObjectSelect" maxOccurs="unbounded"/>
<xs:attribute ref="ifc:itemType" fixed="ifc:IfcResourceObjectSelect"/>
<xs:attribute ref="ifc:cType" fixed="set"/>
<xs:attribute ref="ifc:arraySize" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="RelatingApproval" type="ifc:IfcApproval" nillable="true"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

Спецификация EXPRESS:

```
ENTITY IfcResourceApprovalRelationship
 SUBTYPE OF IfcResourceLevelRelationship;
  RelatedResourceObjects      : SET [1:?] OF IfcResourceObjectSelect;
  RelatingApproval            : IfcApproval;
END_ENTITY;
```

Диаграмма EXPRESS-G

```
ENTITY IfcResourceApprovalRelationship
 SUBTYPE OF IfcResourceLevelRelationship;
  RelatedResourceObjects      : SET [1:?] OF IfcResourceObjectSelect;
  RelatingApproval            : IfcApproval;
END_ENTITY;
```

Определения свойств:

RelatedResourceObjects : Объекты ресурса, которые подтверждены.

RelatingApproval : Подтверждение для объектов ресурса отобрано.

Граф наследования:

ENTITY IfcResourceApprovalRelationship

ENTITY IfcResourceLevelRelationship

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcResourceApprovalRelationship

RelatedResourceObjects : **SET** [1:?] **OF** IfcResourceObjectSelect;

RelatingApproval : IfcApproval;

END_ENTITY;

1.3 Формат BCF

BCF: формат файла для обмена замечаниями/комментариями по проекту, позволяющий к комментариям добавлять соответствующие скриншоты.

Качество обмена IFC-данными можно повысить путем использования BIM Collaboration Format (BCF). BCF – это открытый формат, официально одобренный buildingSMART, и используемый во многих приложениях конструктивного и инженерного проектирования и в программах проверки моделей. Этот формат позволяет добавлять снимки экрана, расположение камер и плоскости 3D-сечений в IFC-модели. С его помощью можно создавать тематические комментарии для отдельных элементов модели. Это позволяет другим пользователям легко находить связанные элементы IFC-модели.

1.4 Формат COBie

Construction Operations Building Information Exchange (COBie) – формат данных для обмена на основе электронной таблицы, содержащий негеометрическую информацию о здании. С помощью COBie может осуществляться передача данных, полученных на основе информационной модели здания (BIM), необходимых для эксплуатации здания после завершения его строительства.

COBie помогает собирать и записывать важные данные по проекту, такие как перечень используемого оборудования, информация о гарантийном обслуживании, списки запчастей к используемому оборудованию, таблицы данных по эксплуатации объекта строительства.

Эти данные являются необходимыми для обслуживания, технического обеспечения и управления построенными зданиями.

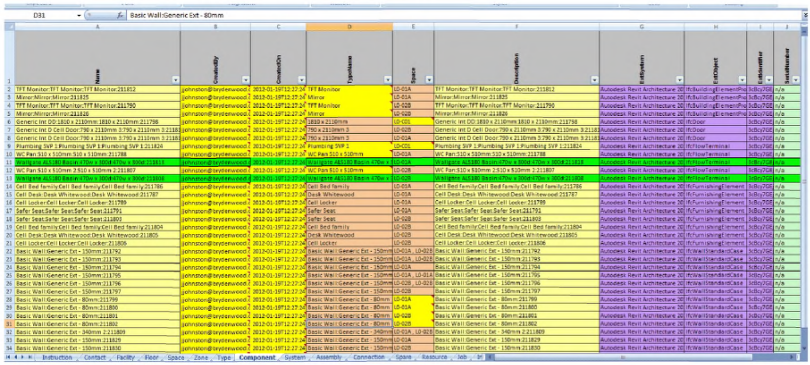


Рисунок 1.4 – Пример фрагмента таблиц COBie

Формат данных COBie поддерживается различными компьютерными приложениями для планирования, проектирования, строительства, ввода в эксплуатацию, обслуживанию и управлению зданиями. COBie может быть представлен в различном виде:

- Электронные таблицы (*.xls).
- STEP-Part 21 (так же называемый IFC).
- ifcXMLВ начале 2013 года организация BuildingSMART работала над созданием облегченной версией XML формата для COBie, который был назван COBieLite

(облегчённая версия). Результат разработки стал доступен для использования в апреле 2013 года.

Формат COBie адаптирован и стандартизован в Великобритании в виде национального стандарта BS 1192-4:2014 «Collaborative production of information Part 4: Fulfilling employer's information exchange requirements using COBie – Code of practice».

1.5 CAD-форматы (2D/3D)

- DWG - Drawing Database File
- DXF - Drawing Exchange Format
- DGN - MicroStation Design File

CAD-форматы (2D и 3D) сейчас поддерживают очень многие САПР, в частности файлы DWG широко применяется в проектной практике виде Xref-ссылок («подложек» 2D и реже 3D «опорных» моделей) при обмене геометрическими данными из различных САПР (BIM/CAD).

DWG: бинарный формат файла, используемый для хранения двухмерных (2D) и трехмерных (3D) проектных данных и метаданных. Является стандартом де-факто для многих систем автоматизированного проектирования. Формат DWG поддерживается многими САПР-приложениями косвенно: то есть данные из одного формата данных перемещаются в другой через функции импорт-экспорт. Форматы dws («drawing standards» — стандарты чертежа), dwt («drawing template» — шаблон чертежа) также являются форматом DWG. Современные программы, работающие в dwg, часто создают на диске временные и резервные копии документов в формате dwg в файлах с расширениями. sv$ («temporary automatic save» — временное автоматическое сохранение) и. bak («backup» — резервная копия).

DXF: формат DXF изначально разрабатывался как внешний формат программы AutoCAD, для обмена чертежами с другими системами. Но из-за популярности AutoCADa формат DXF стал фактически стандартом обмена файлами CAD-чертежей почти для всех САПР. На самом деле, почти в каждой из

появляющихся новых САПР имеется транслятор в формат DXF и обратно. DXF-файл – это текстовый файл в формате ASCII, состоящий из пяти разделов: Header (Заголовок), Table (Таблица), Block (Блок), **ENTITY** (Элемент) и Terminate (Конец). В разделе Header описывается среда AutoCAD, в которой был создан DXF-файл. В разделе Table содержится информация о типах линий, слоях, стилях текста и видах, которые могут быть определены в чертеже. В разделе Block содержится список графических элементов, определенных как группа. Конкретные данные по каждому элементу хранятся в разделе **ENTITY**. Раздел **ENTITY** – это главный раздел DXF-файла, в котором описываются все элементы, присутствующие на чертеже.

DGN: расширение DGN используется в САПР программном обеспечении MicroStation, разработанной и продаваемой компанией Bentley Systems. DGN файл содержит чертеж, который создается и используется в программе. Это один из основных форматов MicroStation, который основан на стандарте или Intergraph Standard File Format (ISFF), или более новом стандарте V8 DGN standard. Формат DGN также используется в некоторых других программах. DGN – это сокращение от английского слова «design». Файлы .DGN можно конвертировать в более универсальный формат DWG с помощью программы AutoCAD.

1.6 Основные векторные графические форматы файлов

DWF: открытый формат файлов, разработанный компанией Autodesk для обмена проектными данными, их просмотра, печати и рецензирования. Открывается при помощи бесплатного программного обеспечения Autodesk Design Review, а также служб облачного сервиса Autodesk 360 в интернет-браузере и мобильных устройствах. Информация, содержащаяся в файле формата DWF, также может быть использована в Revit, AutoCAD и других программах.

Данные в DWF-файлах сжимаются, поэтому обмен проектными данными осуществляется гораздо быстрее. Сжатие сокращает издержки, связанные с пересылкой рисунков, созданных в САПР (а также с управлением внешними ссылками и зависимостями). DWF позволяет отобрать определенные проектные

данные для просмотра заинтересованными лицами и выполнить публикацию многолистовых наборов рисунков из нескольких DWG-файлов в один DWF-файл.

DWF-файлы не заменяют CAD-форматы файлов (DWG): редактирование и обновление проектных данных по-прежнему выполняется в исходных программах.

Формат DWF позволяет просматривать проекты на компьютерах, где не установлены пакеты САПР (AutoCAD, 3ds Max и пр.). Файлы DWF заблокированы от изменений, возможны только пометки.

Создание файлов DWF в AutoCAD возможно, как через процедуру печати виртуальным плоттером DWF6 eplot, так другими инструментами экспорта.

PDF: (Portable Document Format) - межплатформенный формат электронных документов, разработанный фирмой Adobe Systems с использованием ряда возможностей языка PostScript. В первую очередь предназначен для представления полиграфической продукции в электронном виде. Для просмотра существует множество программ, а также официальная бесплатная программа Adobe Reader. Значительное количество современного профессионального печатного оборудования имеет аппаратную поддержку формата PDF, что позволяет производить печать документов в данном формате без использования какого-либо программного обеспечения. Традиционным способом создания PDF-документов является виртуальный принтер, то есть документ как таковой готовится в своей специализированной программе — графической программе или текстовом редакторе, САПР и т. д., а затем экспортируется в формат PDF для распространения в электронном виде, передачи в типографию и т. п.

PDF с 1 июля 2008 года является открытым стандартом ISO 32000.

Формат PDF позволяет внедрять необходимые шрифты (построчный текст), векторные и растровые изображения, формы и мультимедиа-вставки. Поддерживает RGB, CMYK, Grayscale, Lab, Duotone, Bitmap, несколько типов сжатия растровой информации. Имеет собственные технические форматы для полиграфии: PDF/X-1a, PDF/X-3. Включает механизм электронных подписей для защиты и проверки подлинности документов. В этом формате распространяется большое количество сопутствующей документации.

1.7 Основные Форматы файлов трехмерной графики

SKP: формат данных системы трехмерного моделирования SketchUp. Применяется для эскизного моделирование в архитектуре.

STL: формат файла, широко используемый для хранения трехмерных моделей объектов для использования в технологиях быстрого прототипирования, обычно, методом стереолитографии. Информация об объекте хранится как список треугольных граней, которые описывают его поверхность, и их нормалей. STL-файл может быть текстовым (ASCII) или двоичным.

OBJ: формат файлов описания геометрии, разработанный в Wavefront Technologies для их анимационного пакета Advanced Visualizer. Формат файла является открытым и был принят другими разработчиками приложений 3D графики и может быть экспортирован/импортирован в e-Frontier's Poser, Maya, XSI, Blender, MeshLab, Misfit Model 3D, 3D Studio Max и Rhinoceros 3D, Hexagon, Newtek Lightwave, Art of Illusion, milkshape 3d,Modo, Cinema 4D, Zanoza Modeller и т. д. По большей части это общепринятый формат. Формат файлов OBJ — это простой формат данных, который содержит только 3D геометрию, а именно, позицию каждой вершины, связь координат текстуры с вершиной, нормаль для каждой вершины, а также параметры, которые создают полигоны.

FBX: технология и формат файлов, которые используются для обеспечения совместимости различных программ трехмерной графики. В данном формате информационная модель Autodesk Revit экспортируется для использования в программе визуализации, например, Autodesk 3ds Max.

1.8 Прочие используемые форматы

gbXML: (Green Building XML) открытый формат, основан на XML, предназначенный для хранения и обмена геометрической информации об ограждающих конструкциях зданий и сооружений. Применяется для передачи данных из BIM-моделей в ПО для проведения теплотехнических расчетов.

CSV: текстовый формат, предназначенный для представления табличных данных. Каждая строка файла — это одна строка таблицы. Значения отдельных колонок разделяются разделительным символом (delimiter) — запятой (,). Однако, большинство программ вольно трактует стандарт CSV и допускают использование иных символов в качестве разделителя. В частности, в локалях, где десятичным разделителем является запятая, в качестве табличного разделителя, как правило, используется точка с запятой. Значения, содержащие зарезервированные символы (двойная кавычка, запятая, точка с запятой, новая строка) обрамляются двойными кавычками ("); если в значении встречаются кавычки — они представляются в файле в виде двух кавычек подряд. Строки разделяются парой символов CR LF (0x0D 0x0A) (в DOS и Windows эта пара генерируется нажатием клавиши Enter). Однако конкретные реализации могут использовать другие общепринятые разделители строк, например, LF (0x0A) в UNIX.

Несмотря на наличие RFC, на сегодняшний день под CSV, как правило, понимают набор значений, разделенных какими угодно разделителями, в какой угодно кодировке с какими угодно окончаниями строк. Это значительно затрудняет перенос данных из одних программ в другие, несмотря на всю простоту реализации поддержки CSV.

DSTV: файл для ЧПУ. Это промышленный стандартный формат, определенный Немецкой ассоциацией металлостроителей (Deutsche Stahlbau-Verband). Файл DSTV представляет собой текстовый файл формата ASCII. В большинстве случаев каждая деталь имеет свой файл DSTV. Поддерживается Tekla Structures и другими ПО.

Декларированные возможности обмена данными BIM-платформ

Таблица 1.1

Основные форматы обмена, поддерживаемые BIM-платформой Revit

Формат	Revit Architecture	Revit MEP	Revit Structure
IFC	↕	↕	↕
gbXML	↑	↕	↑

IGES	≈	≈	≈
FBX	↕	↕	↕
CIS/2			↕
DWF(x)	↕	↕	↕
DWG	↕	↕	↕
DGN	↕	↕	↕
csv/xml	↕	↕	↕
E57 (облако точек)	↓	↓	↓
STL	↑	↑	↑
COBie	↑	↑	↑
NWC	↑	↑	↑
BCF	↕	↕	↕

Экспорт	↑
Экспорт и импорт	↕
Импорт	↓
Частичная поддержка	

Таблица 1.2

Основные форматы обмена, поддерживаемые BIM-платформой ArchiCAD

Формат	Импорт	Экспорт
DWG	X	X
DXF	X	X
DGN	X	X
IFC	X	X
SKP	X	X
3DM	X	X
STL	X	X
gbXML		X
3DS	X	X
CSV/XML	X	X

Таблица 1.3

Основные форматы обмена, поддерживаемые BIM-платформой Tekla

Формат	Импорт	Экспорт
DWG	X	X
DXF	X	X
DGN	X	X
CIS/2	X	X
IFC	X	
IGES	X	X
LandXML	X	
PDMS		X
SKP	X	X
SDF, SDNF	X	X
STEP	X	
CSV/XML	X	X

Таблица 1.4

Основные форматы обмена, поддерживаемые BIM-платформой Allplan

Формат	Импорт	Экспорт
DWG	X	X
DXF	X	X
DGN	X	X
IFC	X	X
C4D	X	X
SKP	X	X
3DM	X	X
STL	X	X
VRML	X	X
DWF		X
CSV/XML	X	X

Таблица 1.5

Основные форматы обмена, поддерживаемые BIM-платформой Renga Architecture (Аскон)

Формат	Импорт	Экспорт
DXF	X	X
IFC	X	X
OBJ	X	X
SKP	X	X
3DS		X
STL		X
CSV/XML		X

Приложение 2 Некоммерческая международная организация buildingSMART

Международный альянс по интероперабельности (IAI) был создан в 1995 и объединил в своих рядах специалистов проектных, строительных организаций и разработчиков программного обеспечения из 16 стран. С 2008 года альянс носит название buildingSMART.

BuildingSMART International Limited является международной неправительственной некоммерческой ассоциацией региональных и национальных отделений buildingSMART.

Основной задачей альянса является разработка отраслевого стандарта файлового формата данных с открытой спецификацией, которая не контролируется ни одной компанией или группой компаний - Industry Foundation Classes (IFC). Формат IFC был разработан для упрощения взаимодействия в строительной индустрии. Используется как нейтральный формат для информационной модели здания/сооружения (Building Information Model) и содержит соответствующие классы объектов, закрывающие различные потребности жизненного цикла зданий и сооружений. Первый релиз IFC вышел в 1995 году.

Оригинальный стандарт IFC является электронным документом, официально опубликованным в виде html документа, поставляется на цифровых носителях и не существует в печатном виде. Документ распространяется как архив файлов, представляющих собой пригодную для просмотра копию соответствующего раздела сайта организации разработчика стандарта IFC (некоммерческая международная организация buildingSMART).

Разработчиками стандарт IFC изначально сформирован как динамически развивающаяся структура взаимосвязанных данных. С момента появления первой версии выпущено уже 15 вариантов. Новые версии были выпущены в 2015 году и в 2016 году. В настоящее время разработка отраслевых базовых классов сильно интенсифицировалась и обновлений будет все больше и выходить они будут чаще.

Разработчики программных продуктов во всех странах, в том числе в России, используют оригинальную версию IFC, опубликованную на официальном сайте

gost.e-ecolog.ru
и сертифицируют свои продукты в некоммерческой международной организации buildingSMART. Некоммерческая международная организация buildingSMART, в свою очередь, проводит добровольную сертификацию программных продуктов на соответствие выбранной разработчиками версии стандарта. Как уже было сказано выше, их более 15.

Сейчас формат IFC используется практически всеми BIM-платформами и стал стандартом представления информационных моделей во многих государственных и коммерческих организациях в целом ряде стран.

Список использованных источников

1. IFC Reference Guide for ArchiCAD 19 Copyright © 2015 by GRAPHISOFT.
2. Lee Y., Eastman C.M., Lee J. Validations for ensuring the interoperability of data exchange of a building information model. Automation in Construction 58 (2015) 176–195.
3. «Отчет о научно-исследовательской работе «Анализ методологии автоматизированной оценки и контроля стоимости строительства и эксплуатации зданий и сооружений, зарубежной и отечественной практики применения технологии информационного моделирования для расчета стоимости строительства и эксплуатации в течение жизненного цикла объектов», 2017» АО «НИЦ «Строительство», М.: 2017.
4. «27. Conceptual Hierarchy», Редакция Jul 29, 2014, «16. Competency Flow», Редакция Mar 16, 2014. (портал gost.e-ecolog.ru).
5. Федотов А.М., Идрисова И.А., Самбетбаева М.А., Федотова О.А. Использование тезауруса в научно-образовательной информационной системе // Вестник Новосибирского гос. ун-та. Серия: Информационные технологии. 2015. Т. 13, вып. 2. С. 86–102.
6. Статья «Классификатор» портала Википедия (gost.e-ecolog.ru /Классификатор).
7. Заочная школа школьного библиотекаря: Сб. лекций. — М.: Школьная библиотека, 2004.
8. Балдин К.В., Уткин В.Б. Информационные системы в экономике: Учебник – 5-е изд. – М.: Издательско-торговая корпорация «Дашков и К», 2008.
9. ГОСТ Р ИСО 12006-2–2017 "Строительство. Модель организации данных о строительных работах. Часть 2. Основы классификации информации". М.: Стандартинформ, 2017.
10. Документация о стандартах для совместной работы в openBIM. (портал gost.e-ecolog.ru).
11. Классификатор строительных ресурсов КСР-2016. Минстрой. М.:2016

12. Отчет о научно-исследовательской работе «По мониторингу и анализу нормативных технических документов в области информационного моделирования в сфере строительства», АО «НИЦ «Строительство», М.: 2015.
13. IFC WebServer (портал gost.e-ecolog.ru)
14. BuildingSmart, ifcDoc (портал gost.e-ecolog.ru specifications/ /specification-tools/ifcdoc-tool)
15. Antony McPhee. «IFC, What is it good for?», 2013 (портал gost.e-ecolog.ru)
16. ISO 29481-1:2016 Building information models. Information delivery manual. Part 1: Methodology and format (Моделирование информационное зданий и сооружений. Руководство по доставке информации. Часть 1. Методология и формат).
17. ISO 16739. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries (Отраслевые базовые классы (IFC) для обмена информацией на всех этапах жизненного цикла).
18. ISO 12006-3:2007. Building construction. Organization of information about construction works. Framework for object-oriented information (Строительство. Модель организации данных о строительных работах. Часть 3. Основы обмена объектно-ориентированной информацией).
19. ISO 12006-2:2015. Building construction – Organization of information about construction works – Part 2: Framework for classification (Строительство. Модель организации данных о строительных работах. Часть 2. Основы классификации информации).
20. Король М.Г. «Британцы сообщили миру, что такое BIM уровня 3: это — Digital Built Britain», 2015, (портал gost.e-ecolog.ru _num=17570)

gost.e-ecolog.ru