
ТЕОРИЯ ПРОГРАММИРОВАНИЯ:
ФОРМАЛЬНЫЕ МОДЕЛИ И СЕМАНТИКА

УДК 004.436

**ФОРМАЛЬНАЯ СПЕЦИФИКАЦИЯ И ВЕРИФИКАЦИЯ
ТРЕБОВАНИЙ В АРХИТЕКТУРЕ И СТРОИТЕЛЬСТВЕ
НА ОСНОВЕ ЯЗЫКА МОДЕЛИРОВАНИЯ EXPRESS**

© 2024 г. В. А. Семенов^{a,*}, С. В. Морозов^{a,**}, С. В. Аришин^{a,***},
О. Н. Кузина^{b,****}, В. И. Римшин^{b,*****}, Е. В. Макиша^{b,*****}

^aИнститут системного программирования им. В.П. Иванникова РАН

109004 Россия, Москва, ул. А. Солженицына, д. 25

^bНациональный исследовательский Московский государственный строительный университет

129337 Россия, Москва, Ярославское шоссе, д. 26

*e-mail: sem@ispras.ru

**e-mail: serg@ispras.ru

***e-mail: arishin@ispras.ru

****e-mail: kuzinaon@mgsu.ru

*****e-mail: RimshinVI@mgsu.ru

*****e-mail: makishaev@mgsu.ru

Поступила в редакцию: 23.04.2024 г.

После доработки: 03.05.2024 г.

Принята к публикации: 03.05.2024 г.

В настоящее время цифровые технологии моделирования зданий, сооружений и инфраструктуры успешно применяются в международной и российской практике реализации сложных строительных проектов и масштабных программ. Вместе с тем, реализуемый во многих странах переход к машинночитаемым стандартам с целью повышения качества проектной документации и автоматизации ее проверки сталкивается с серьезными методологическими и инструментальными проблемами. Прежде всего они связаны со сложностью самих цифровых моделей, а также с разнообразием требований, формулируемых на естественных языках и предъявляемых к моделям на государственном, региональном, ведомственном и корпоративном уровне. Предпринимаемые попытки создания реестров требований и программных инструментов для их ведения и использования, как правило, носят специализированный характер и не обеспечивают необходимую полноту, нормализацию, согласованность, взаимосвязанность, однозначность, прослеживаемость и проверяемость описаний требований. Конструктивным, в связи с этим, представляется применение формальных методов спецификации и верификации требований, зарекомендовавших себя в системной и программной инженерии. В работе проводится сравнительный анализ программных инструментов для автоматизированной проверки нормативных требований в области строительства. Отмечается возросшая популярность инструментов, ориентированных на международные стандарты IFC (Industry Foundation Classes), IDS (Information Delivery Specification) и обеспечивающих контроль полноты объектного и атрибутного состава моделей, а также уточнение допустимых областей значений. Вместе с тем, стандарт IDS не формализован и не предусматривает задание требований, выражаемых произвольными алгебраическими условиями. Перспективным для формальной спецификации и верификации требований к цифровым моделям в строительстве представляется применение языка моделирования объектно-ориентированных данных EXPRESS, на котором специфицирована и информационная схема IFC. В качестве обоснования показывается представимость IDS спецификаций логическими выражениями и функциями EXPRESS, а также возможность задания произвольных алгебраических условий в виде декларативных правил языка EXPRESS. В качестве иллюстраций предлагаемого подхода приводятся примеры формализации строительных нормативов и сводов правил РФ, предъявляемых к безопасности зданий, сооружений и процессов. Также обсуждаются возможности гармонизации предлагаемого формального подхода со стандартом IDS в результате определения новых паттернов для представления локальных, уникальных и глобальных правил языка EXPRESS.

Keywords: BIM, IFC, EXPRESS, IDS, спецификация и верификация требований, инженерия требований

DOI: 10.31857/S0132347424050052, **EDN:** OLMAMS

1. ВВЕДЕНИЕ

В настоящее время цифровые технологии моделирования зданий, сооружений и инфраструктуры успешно применяются в международной и российской практике реализации сложных строительных проектов и масштабных программ. Модели разрабатываются в соответствующих САПР, а при обмене между участниками проектной деятельности, как правило, преобразуются в файловые форматы в соответствии с международными и национальными стандартами по функциональной совместимости (интероперабельности) программного обеспечения. Ключевыми среди них являются открытые стандарты и сервисы openBIM, разработанные международным альянсом buildingSMART [1]. Прежде всего, это IFC (Industry Foundation Classes) [2], IDM (Information Delivery Manual) [3], BCF (BIM Collaboration Format) [4], bSDD (buildingSMART Data Dictionary) [5], которые в настоящее время активно используются ведущими промышленными странами. Детальная информация о стандартах, сервисах и технологиях buildingSMART содержится в монографии [6].

Вместе с тем, реализуемый во многих странах переход к машиночитаемым стандартам с целью повышения качества проектной документации, автоматизации и интеллектуализации ее проверки сталкивается с серьезными методологическими проблемами [7]. Одной из главных причин является отсутствие единого понимания, каким образом должны быть организованы реестры требований, чтобы обеспечить необходимую полноту (complete), нормализацию (normalized), детализируемость (granular), уникальность (unique set), согласованность (consistent), взаимосвязанность (linked set), модифицируемость (modifiable) описаний требований, а также каким образом должны быть представлены сами требования, чтобы обеспечить необходимый уровень абстрактности (abstract) при однозначной интерпретируемости (unambiguous), прослеживаемости (traceable) и проверяемости (validatable). Данные принципы закреплены в международном стандарте по системной инженерии [8].

Методами управления требованиями в соответствии с перечисленными принципами занимается дисциплина “инженерия требований” [9], в которой для обеспечения математически строгих гарантий корректной спецификации требований и их однозначной интерпретации применяются формальные методы [10]. Например, для контроля целостности данных и согласованно-

сти состояний в приложениях системной и программной инженерии, разработанных на основе универсального языка моделирования UML (Unified Modeling Language) [11], часто применяется метод контрактных спецификаций с инвариантами, пред- и постусловиями на языке объектных ограничений OCL (Object Constraint Language) [12].

Однако в архитектурно-строительной отрасли формальные методы ограниченно применяются из-за сложности в освоении пользователями, не являющимися IT-специалистами. Хотя стандартная информационная схема IFC, определяющая типы данных и правила целостности цифровых моделей, специфицирована на декларативном языке моделирования объектно-ориентированных данных EXPRESS [13], конечные пользователи, как правило, не способны формализовать и описать на этом языке собственные правила, выражающие те или иные требования к моделям. Другими факторами, затрудняющими переход к машиночитаемым стандартам, являются разнообразие норм и сводов правил, предъявляемых к моделям на государственном, региональном, ведомственном, корпоративном уровне; неоднозначность и нестрогость формулировок на естественных языках; необходимость в определении алгебраически полного языка, обеспечивающего формальную и конструктивную спецификацию разнообразных требований.

В качестве примера реализации формального подхода к спецификации и верификации требований следует указать систему EDM Express Data Checker [14], в которой требования оформляются в виде декларативных правил EXPRESS. Поскольку информационные схемы цифровых моделей, подлежащих верификации, также специфицируются на EXPRESS, решается проблема несоответствия метамodelей (impedance mismatch) и обеспечивается необходимая математическая строгость в представлении и интерпретации требований. Вместе с тем, данный инструмент не предусматривает функциональных расширений, которые могли бы существенно упростить формализацию и спецификацию норм и правил, например, с учетом особенностей стандарта IFC.

Важное место в практике верификации цифровых моделей на основе стандарта IFC занимают инструменты, имеющие в своем арсенале predefined, готовые к использованию шаблоны проверок. Самым популярным, безусловно, является система Solibri Model Checker [15]. К этому семейству следует также отнести 7D Modeler

[16], BIMcollab Zoom [17], BlenderBIM [18], Open IFC Viewer [19], usBIM.checker [20], которые позволяют выполнить типовые проверки объектной организации модели, ее атрибутивного состава и пространственного размещения элементов. Принципиальным недостатком подобных инструментов и императивного подхода, в целом, является функциональная ограниченность. Наличие прикладных программных интерфейсов не решает проблему, поскольку отсутствуют спецификации исходных требований, нейтральные по отношению к платформам, языкам и системам реализации, а их неформальная интерпретация может приводить к разным вердиктам.

Альтернативу рассмотренным подходам составляют инструменты, сочетающие в себе декларативные средства для описания требований, как правило, в терминах специализированных словарей или классификационных систем, и императивные средства для проверки выполнимости требований применительно к заданным цифровым моделям. Требования обычно задаются с помощью специальных редакторов и представляются на диалектах языков правил, подобных RuleML, LRML, SWRL, LDML [21]. Несмотря на кажущуюся гибкость эклектичных инструментов, они неизбежно порождают проблемы несоответствия метамodelей и недостоверности результатов из-за необходимости интерпретации правил в терминах целевых моделей и реализации промежуточного слоя между декларативными и императивными средствами.

В качестве особого семейства следует указать инструменты, ориентированные на перспективный открытый стандарт IDS (Information Delivery Specification) [22], официальный релиз 1.0 которого планируется к принятию альянсом buildingSMART во второй половине 2024 года. Стандарт предусматривает спецификацию требований в терминах информационной схемы IFC, что решает упомянутую выше проблему несоответствия. Примером полнофункциональной реализации стандарта IDS являются сервисы и редактор машиночитаемых требований ИСП РАН [23]. Сервисы обеспечивают контроль соответствия цифровой модели схеме IFC, а также выполняют проверки полноты ее объектного и атрибутивного состава, корректности областей значений, заданных в виде регулярных выражений, перечислимых множеств, числовых интервалов, а также отношений композиции и агрегации. Вместе с тем, на сегодняшний день стандарт IDS не формализован, а также не предусматривает средств для спецификации сложных требова-

ний, выражаемых параметрическими, пространственными, топологическими, метрическими, темпоральными и прочими алгебраическими условиями. Известны подходы к расширению IDS с использованием императивных средств и отмеченными выше проблемами [24, 25].

Настоящая статья адресована проблемам и возможностям применения языка моделирования EXPRESS для формальной спецификации и верификации требований в архитектуре и строительстве. В разделе 2 обсуждаются открытые международные стандарты IFC, IDS и BCF, применяемые для автоматизированной проверки цифровых моделей зданий, сооружений и инфраструктуры. Раздел 3 посвящен особенностям декларативного языка EXPRESS и его основным конструкциям для определения объектных типов данных и задания правил целостности на них. В разделе 4 проводится формализация стандарта IDS в результате представления стандартных паттернов логическими выражениями и функциями EXPRESS. Примеры формализации строительных нормативов и сводов правил РФ, предъявляемых к безопасности зданий, сооружений и процессов, приводятся в разделе 5 в качестве обоснования универсальности и гибкости языка EXPRESS для спецификации требований, выражаемых сложными алгебраическими условиями. Наконец, в разделе 6 описываются результаты апробации предлагаемого формального подхода, возможности его гармонизации со стандартом IDS и расширения последнего путем определения новых паттернов для представления локальных, уникальных и глобальных правил языка EXPRESS.

2. ОТКРЫТЫЕ СТАНДАРТЫ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ В АРХИТЕКТУРЕ И СТРОИТЕЛЬСТВЕ

Industry Foundation Classes (IFC) – это открытая информационная схема и стандартизованный формат данных, независимые от поставщиков программного обеспечения (ПО) и разработанные международным альянсом buildingSMART для обеспечения интероперабельности ПО в области архитектуры и строительства (АЕС). Будучи одобренным ISO в качестве международного стандарта [2], а также принятым в РФ в качестве соответствующего ГОСТ [26], IFC (версия 4.0.2.1) широко используется различными программными приложениями и инструментами [27] для обмена междисциплинарными данными (включая архитектуру, описание механических,

электрических систем, систем отопления, вентиляции и кондиционирования, графики работ, сметы, модели для структурного и энергетического анализа и т. д.) и их совместного использования различными участниками архитектурно-строительных проектов. В марте 2024 года ISO приняла версию 4.3.2.0 [28], ранее одобренную buildingSMART и расширяющую информационную схему IFC понятиями, используемыми в области строительства инфраструктурных объектов (дороги, мосты, станции, тоннели, водные пути, каналы, пристани, порты и т. п.).

Базируясь на методологии STEP (STandard for the Exchange of Product model data, ISO 10303) [29], модель данных IFC специфицируется на языке объектно-ориентированного моделирования EXPRESS [13] и поддерживает хранение и транспортировку данных в форматах, предоставляемых STEP, включая:

- IFC STEP Physical File (SPF) — формат кодирования STEP-данных открытым текстом в соответствии со стандартом ISO 10303–21:2016 [30] (наиболее широко используемый формат в настоящее время);
- IFC XML — формат XML для представления STEP-данных, определенный стандартом ISO 10303–28:2007 [31] (не получил широкого распространения из-за раздутого размера XML-файлов по сравнению с SPF);
- IFC ZIP — файл IFC SPF или IFC XML, сжатый в формате ZIP;
- IFC HDF — бинарный Hierarchical Data Format, определенный технической спецификацией ISO/TS10303–26:2011 [32] для представления STEP-данных (имеющий экспериментальный статус для IFC).

Заметим, что модель данных IFC сочетает в себе как объектно-ориентированный, так и онтологический [33] подходы. К последнему относятся следующие возможности:

- отдельные определения для типов объектов (наследники типа *IfcTypeObject*) и экземпляров объектов или индивидов (наследники типа *IfcObject*) для расширения семантики модели путем создания новых экземпляров типов и их ассоциации с соответствующими индивидами;
- объектификация взаимоотношений (наследники типа *IfcRelationship*), позволяющая расширить семантику языка EXPRESS путем определения иерархии различных типов взаимоотношений (включая ассоциации, композиции, назначения, определения, декларации, связи) и хранить характеристики взаимоотношений отдельно от типов взаимосвязанных объектов;

- определение дополнительных статических свойств объектов и их типов наряду со стандартными атрибутами в виде динамически расширяемых наборов качественных (*IfcPropertySet*) и количественных (*IfcQuantitySet*) характеристик;

- дополнительная категоризация объектов и их типов с использованием predefined таксономических схем или систем классификации (*IfcClassification*).

Детальную информацию об IFC, его использовании и развитии можно найти в разделе 3.2 монографии [6], а также в следующих источниках [34–37].

В настоящее время все большую популярность приобретает открытый стандарт IDS (Information Delivery Specification) [22], разрабатываемый сообществом buildingSMART и предназначенный для перевода требований к цифровой информационной модели (ЦИМ) объекта строительства, представленной в соответствии со стандартом IFC, в машиночитаемый вид и их дальнейшей верификации с использованием инструментов, поддерживающих данный стандарт [27]. Текущая версия IDS0.9.6 позволяет формализовать типовые проверки ЦИМ, в частности, полностью ее атрибутивного состава, наличия требуемых дополнительных свойств, соответствия установленным правилам классификации, именования, соответствия значения указанному образцу, его принадлежность заданному множеству или интервалу.

С технической точки зрения IDS документ представляет собой XML файл с приписанной схемой XSD, которая должна применяться для контроля соответствия данных, содержащихся в этом файле, стандарту IDS. В частности, подобный XML файл должен содержать общий заголовок документа и набор формализованных правил. Общий заголовок включает в себя название документа, краткое назначение, развернутое описание, а также версию документа, владельца авторских прав, e-mail авторов, дату создания и этапа жизненного цикла объекта капитального строительства, на котором предполагается использование данных правил.

Каждое формализованное правило состоит из следующих компонентов:

- общая информация (general), в которой указывается название спецификации, идентификатор, назначение, инструкции, версии стандарта IFC, которым соответствует данное правило, а также статус, устанавливающий обязательный или рекомендуемый характер соответствующего требования;

- контекст или применимость (applicability), задаваемый декларативным образом, используя объектный запрос или, менее формально, фильтр, результатом которого являются выбранные объекты исходной модели в IFC, подлежащие проверке выполнимости соответствующего требования;

- требование (requirement), которое задается с использованием тех же декларативных конструкций, что позволяет интерпретировать правило как формальное суждение о принадлежности каждого объекта, удовлетворяющего условиям контекста, множеству объектов, удовлетворяющих условиям требования.

Для задания контекста и требования в IDS предусмотрено шесть предопределенных паттернов (facets), каждый из которых позволяет определить условия соответствующего вида:

- *Entity*: принадлежность объектов указанным объектным типам данных IFC с возможностью их уточнения предопределенными типами при их наличии для заданных типов объектов в спецификации модели IFC (например, в наборе данных должны существовать плиты *IfcSlab* с предопределенным типом “перекрытие”

IfcSlabTypeEnum.FLOOR:

Name="IFCSLAB",

PredefinedType="FLOOR");

- *PartOf*: задание отношений композиции между объектами заданных типов с возможностью уточнения вида композиции: агрегация, группировка, принадлежность пространственной структуре, вложение, высечение, заполнение (например, некоторый объект должен быть частью навесной стены типа *IfcCurtainWall*: *Relation="IFCRELAGGREGATES"*, *Entity="IFCCURTAINWALL"*);

- *Attribute*: наличие значений у атрибутов, предопределенных схемой IFC, и принадлежность их заданным областям определения допустимых значений, при этом отсутствие области определения значений соответствует наличию определенного (*NOT NULL*) значения (например, имя объекта *Name* должно быть не длиннее 50 символов: *Name="Name"*, *Value="MaxLength=50"*);

- *Property*: наличие установленных дополнительных свойств объектов (property или quantity), а также, опционально, указание соответствия их значений заданному типу данных и области определения допустимых значений (например, в наборе основных количественных параметров плиты должен быть указан ее чистый объем, значение которого должно быть в диапазоне от 20 до 100 кубических метров:

Property Set="Qto_SlabBaseQuantities", *Name="Net-Volume"*, *DataType="IFCVOLUMEMEASURE"*, *Value="20<=Value<=100"*);

- *Classification*: наличие у объектов назначенных классификаторов и их соответствие заданным системам классификации UniFormat [38], OmniClass [39], КСИ [40] и т. п., при этом указание только системы классификации означает возможность использования произвольных классификаторов из данной системы, а задание паттерна без параметров соответствует наличию произвольных классификаторов у данного объекта (например, объекты должны быть классифицированы с использованием системы OmniClass, классификатор должен начинаться с

'23-': System="OmniClass", *Value="23-**);

- *Material*: наличие у строительных элементов IFC назначенных материалов, сопоставляемых по имени или категории, при этом пустое значение в данном паттерне соответствует наличию произвольного материала (например, элементы должны быть стальные либо бетонные: *Value=["steel"*, *"concrete"]*).

Для более детального знакомства со стандартом IDS можно обратиться к статье [41], разделу 3.7 монографии [6] и технической документации на сайте проекта [42].

В процессе верификации требований необходимо сформировать детальный отчет, включающий, в том числе, и информацию об обнаруженных нарушениях. С этой целью наиболее удобно использовать журнал замечаний в формате BCF (BIM Collaboration Format) [4], открытая спецификация которого также разрабатывается и поддерживается альянсом buildingSMART. Текущая версия BCF – 3.0. BCF представляет собой файловый архив в формате ZIP, включающий несколько XML файлов со стандартными схемами для описания используемой версии формата и ее расширения, самого журнала замечаний, включающего традиционные для отслеживания замечаний понятия (замечание, комментарий, точка обзора, снимок, документ), каталог используемых документов со ссылками на них, а также сопутствующие документы (изображения, нормативы, и т. п.) для детального разъяснения обнаруженных замечаний. Замечания BCF ссылаются на соответствующие объекты в IFC файле, используя их GUID. buildingSMART предоставляет API для работы с BCF в режимах обмена файлами между отдельными приложениями и REST API для использования BCF в веб-сервисах. Детальную информацию о BCF и его использовании можно найти в разделе 3.4 монографии [6], техническую

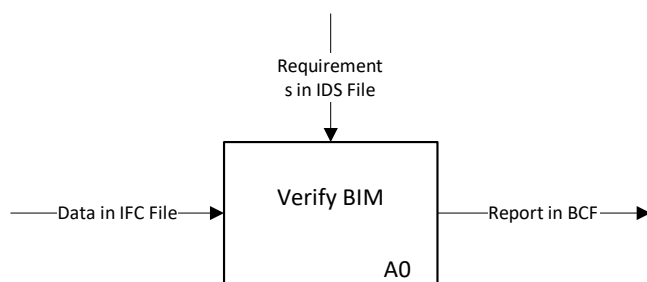


Рис. 1. Процесс верификации требований в строительстве на основе открытых стандартов.

документацию по BCF API — на сайте [43]. Список поддерживающего данный формат программного обеспечения находится в общей базе совместимого ПО buildingSMART [27].

Таким образом, международный альянс buildingSMART предоставляет развитое семейство открытых стандартов для представления ЦИМ объектов строительства, описания требований к ним и формирования журналов замечаний к ЦИМ в процессе их формирования и применения. Организация процесса верификации с использованием данных стандартов изображена на рис. 1. Приложение верификации принимает данные в формате IFC и требования к ним в IDS файле и выдает отчет в виде журнала замечаний в формате BCF.

3. ЯЗЫК МОДЕЛИРОВАНИЯ ДАННЫХ EXPRESS

EXPRESS [13] является декларативным языком информационного моделирования, предназначенным для независимого от особенностей реализации формального описания объектов реального мира, определения их статических свойств (структур данных) и ограничений целостности. Являясь частью методологии стандартов семейства STEP [29], он широко используется для описания моделей данных в различных отраслях промышленности (машиностроение, строительство и архитектура, нефте- и газодобыча, судостроение и т. д.).

EXPRESS — объектно-ориентированный язык моделирования данных и основной структурой данных в нем является объект типа *ENTITY*. Остальные типы данных используются при определении атрибутов объектных типов. К ним относятся простые типы данных с общепринятой семантикой: числовые (*NUMBER*, *INTEGER*, *REAL*), логические (*BOOLEAN*, *LOGICAL*), символьные и битовые строки (*STRING*, *BINARY*). К сложным типам, кроме объектного (*ENTITY*), относят также перечисляемые типы (*ENUMERATION*),

выборки (*SELECT*), различные коллекции: упорядоченные (*ARRAY*, *LIST*) и неупорядоченные (*BAG*, *SET*). Типы данных, включая простые, возможно переопределять с помощью конструкции *TYPE* с указанием имени нового типа и, опционально, ограничения определения области допустимых значений. В EXPRESS предусматриваются и обобщенные типы данных, которые могут использоваться при определении формальных параметров функций и процедур. Тип *GENERIC* является обобщением всех допустимых в языке типов данных. Тип данных *AGGREGATE* является обобщением для типов коллекций *ARRAY*, *LIST*, *BAG*, *SET* и их специализаций. Более подробно о типах данных языка EXPRESS можно узнать из стандарта [13], а также из следующих статей [36, 44–46].

Для настоящей статьи наиболее интересно рассмотреть возможности спецификации правил, определяющих ограничения целостности данных. В языке EXPRESS допустимы определения правил трех типов:

- локальные правила (*WHERE*) в объектных или переопределенных типах, определяющие ограничения области допустимых значений отдельного атрибута или типа данных, или же некоторые зависимости между значениями атрибутов в объектном типе;
- правила уникальности (*UNIQUE*) в объектных типах, устанавливающие условия уникальности значений одного атрибута или их комбинации на экстенсте соответствующего объектного типа данных;
- глобальные правила (*RULE*), задающие ограничения целостности для экстенстов одного или нескольких объектных типов.

Несмотря на то, что язык EXPRESS является декларативным, в определениях локальных и глобальных правил применяются императивные конструкции: выражения, а также вызовы предопределенных и определяемых пользователем функций и процедур, в теле которых допускается использование операторов присваивания, условного перехода, выбора, циклов, возврата, как и в традиционных языках программирования. Выражения языка EXPRESS отличаются богатством синтаксических конструкций. В них можно использовать следующие операции и предопределенные функции:

- арифметические (включая целочисленное и вещественное деление), сравнения, нахождение значения в интервале, тригонометрические, возведение в степень, вычисление экспоненты, логарифмов, квадратного корня, модуля, опре-

деление четности, преобразование числа в строку и строки в число для числовых типов данных;

- логические (and, or, xor, not), сравнения для логических типов данных;
- конкатенация, определение длины, выделение подстроки, символа/бита по индексу, сравнения для символьных и битовых строк, сравнение с образцом только для символьных строк;
- сравнения для перечисляемых типов данных;
- глубокое сравнение на равенство или неравенство, определение идентичности или неидентичности экземпляров, определение множества имен ролей, исполняемых экземпляром в связях с другими объектами, определение мультимножества экземпляров;
- использующих заданный объект в указанной роли, конструирование экземпляра (с возможностью конкатенации конструкторов для создания экземпляра сложного типа), обращение к атрибуту, доступ к подтипу (взятие всех атрибутов, определенных в этом подтипе) для объектов типов данных;
- теоретико-множественные, сравнение на равенство или неравенство, определение идентичности или неидентичности элементов, подмножества и надмножества, принадлежности элемента (по равенству или идентичности), наличия уникальных по значению элементов, количества элементов, нижней и верхней границы (индекса для массивов), запрос (итерирование с вычисле-

нием заданного логического выражения для каждого элемента и включением элемента в результирующую коллекцию в случае истинности этого выражения), вставка/удаление элемента (только для списков), взятие одного или нескольких элементов для типов коллекций;

- проверка существования значения, задание альтернативного значения для неопределенного состояния, определение множества имен типов данных, которым принадлежит заданное значение для любого типа данных.

Продemonстрируем пример записи локального правила для трехмерного объекта, получаемого путем экструзии: перемещения сечения в заданном направлении на фиксированное расстояние (см. рис. 2). Правило *ValidExtrusionDirection* устанавливает, что заданное направление экструзии не должно быть перпендикулярно оси Z в локальной системе координат объекта. В нем используется вызов функции *IfcDotProduct* для вычисления скалярного произведения двух векторов.

Таким образом, на языке EXPRESS можно сформулировать достаточно сложные локальные и глобальные правила, используя богатый репертуар его синтаксических конструкций.

4. ФОРМАЛИЗАЦИЯ СТАНДАРТА IDS

Рассмотрим, как можно представить паттерны IDS с помощью конструкций языка EXPRESS. Подобные конструкции (логические выражения

```
ENTITY IfcExtrudedAreaSolid
  SUPERTYPE OF
    (IfcExtrudedAreaSolidTapered)
  SUBTYPE OF (IfcSweptAreaSolid);
  ExtrudedDirection : IfcDirection;
  Depth : IfcPositiveLengthMeasure;
  WHERE
    ValidExtrusionDirection :
      IfcDotProduct(
        IfcRepresentationItem() ||
        IfcGeometricRepresentationItem() ||
        IfcDirection([0.0,0.0,1.0]),
        SELF.ExtrudedDirection) <> 0.0;
END_ENTITY;
```

```
FUNCTION IfcDotProduct
  (Arg1, Arg2 : IfcDirection) : REAL;
LOCAL
  Scalar : REAL;
  Vec1, Vec2 : IfcDirection;
  Ndim : INTEGER;
END_LOCAL;

IF NOT EXISTS (Arg1) OR NOT EXISTS (Arg2) THEN
  Scalar := ?;
ELSE
  IF (Arg1.Dim <> Arg2.Dim) THEN
    Scalar := ?;
  ELSE
    BEGIN
      Vec1 := IfcNormalise(Arg1);
      Vec2 := IfcNormalise(Arg2);
      Ndim := Arg1.Dim;
      Scalar := 0.0;
      REPEAT i := 1 TO Ndim;
        Scalar := Scalar +
          Vec1.DirectionRatios[i]*Vec2.DirectionRatios[i];
      END_REPEAT;
    END;
  END IF;
END_IF;
RETURN (Scalar);
END_FUNCTION;
```

Рис. 2. Пример спецификации объекта с локальным правилом на языке EXPRESS.

и функции) используются для указания контекста применимости требований (выделения множества объектов для проверки их выполнимости) и формулирования самих требований, выражаемых простой конъюнктивной формой в IDS. Тем самым, представимость паттернов IDS конструкциями EXPRESS служит обоснованием формализации всего стандарта IDS. Заметим, что при отображении в EXPRESS значений следует учитывать их реальные типы данных в соответствии со схемой IFC, поскольку типы данных в XML ограничиваются только строками, числами и булевыми значениями.

4.1. Представление паттерна Entity

Паттерн *Entity* можно представить в виде логического выражения, в состав которого входит функция *TYPEOF*. Так как согласно стандарта EXPRESS данная функция возвращает множество строк, содержащих названия всех допустимых типов данных для значения, передаваемого в нее в качестве параметра, при этом названия типов квалифицируются именем схемы, содержащей определение этого типа, то строку с именем типа, полученную из паттерна IDS, следует квалифицировать именем схемы стандарта IFC соответствующей версии. В IDS имена типов в данном паттерне могут задаваться как в виде простого значения (одиночной строки), так и в виде перечисления (множества строк). Поэтому для сравнения имен в первом случае следует использовать операцию *IN* для проверки вхождения названия типа во множество, возвращаемое функцией *TYPEOF*. Во втором случае следует выполнить операцию пересечения двух множеств и сравнить размер результата с нулем.

Если в паттерне используется необязательный параметр *PredefinedType*, то условие проверки принадлежности типу данных должно быть связано конъюнкцией с условием равенства значения атрибута *PredefinedType* в соответствующем типе объекта IFC заданному в IDS для случая простого значения или с условием вхождения этого значения во множество допустимых (операция *IN*) для случая перечисления значений в IDS. При этом строковые значения IDS должны быть преобразованы к соответствующим им перечисляемым литералам EXPRESS, так как в объектных типах IFC атрибут *PredefinedType*, как правило, определяется типами перечислений. Так, для приведенного выше примера паттерна *Entity* (*Name*="IFCSLAB", *PredefinedType*="FLOOR") используется следующее логическое выражение EXPRESS:

```
'IFC4.IFCSLAB' IN TYPEOF(Obj) AND
Obj.PredefinedType = IfcSlabTypeEnum.FLOOR
```

Будем считать, что существует одноместный объектный предикат *IsClassOf*, первый параметр *Obj* которого является объектом корневого типа *IfcRoot*, чей конкретный тип необходимо проверить, второй параметр *Name* является множеством строк названий допустимых типов, а третий параметр *PredefinedType* задает множество (возможно, пустое) перечисляемых констант для уточнения предопределенного типа объекта. Типом данных для элементов множества *PredefinedType* должен быть селективный тип, объединяющий все перечислимые типы схемы данных IFC, используемые для определения предопределенных типов объектов.

```
FUNCTION IsAggregatedIn(
    PartObj : IfcObjectDefinition;
    ParentTypes : SET [1:?] OF STRING) : LOGICAL;
LOCAL
    ParentObj : IfcObjectDefinition;
END_LOCAL;
IF SIZEOF(PartObj.Decomposes) = 0 THEN
    RETURN (FALSE);
END_IF;
ParentObj :=
    PartObj.Decomposes[1].RelatingObject;
IF (SIZEOF(TYPEOF(ParentObj) * ParentTypes) >
    0) THEN
    RETURN (TRUE);
ELSE
    RETURN (IsAggregatedIn(ParentObj,
        ParentTypes));
END_IF;
END_FUNCTION;
```

Рис. 3. Функция на языке EXPRESS для определения наличия отношений агрегации.

4.2. Представление паттерна *PartOf*

Рассмотрим представление паттерна *PartOf* для отношений агрегации. Так как поиск родительского объекта, представляющего целое в данном отношении, должен осуществляться рекурсивно согласно стандарту IDS, для этих целей разработана рекурсивная функция *IsAggregatedIn* (см. рис. 3). Допустимые типы родительского объекта передаются как множество строк и при сопоставлении используется операция пересечения данного множества с результатом функции *TYPEOF*, примененной к родительскому объекту.

Для отношений другого вида можно написать аналогичные функции на языке EXPRESS, которые в статье не приводятся для краткости изложения. Также существует функция *PartOf*, которая в зависимости от типа передаваемого объекта и списка типов отношений вызывает упомянутые вспомогательные функции для проверки его вхождения в те отношения, которые задаются входным параметром и являются допустимыми для его типа согласно спецификации IFC.

4.3. Представление паттерна *Attribute*

Паттерн *Attribute* можно представить в виде логических выражений согласно таблице 1 в зависимости от способа задания области определения допустимых значений в соответствующем паттерне IDS. В качестве примеров в таблице используются следующие требования к представлению данных о работах по проекту (тип данных *IfcTask*): для работ следует задавать описание; их длительность должна определяться как рабочее время (по календарю); статус работ указывается только тремя величинами: *NOTSTARTED*, *STARTED*, *COMPLETED*; значение идентификатора работ должно соответствовать следующему шаблону: первые три символа 'DT-', за которыми следуют два произвольных символа, далее через тире может быть записана произвольная строка; приоритет работы должен являться целым числом от 1 до 100; длина имени работы не должна превышать 50 символов.

Заметим, что синтаксис регулярных выражений в XML не соответствует применяемому в операции *LIKE* языка EXPRESS. Детально проблема отображения регулярных выражений не исследовалась, но, несомненно, простые шаблоны в образцах IDS могут быть представлены в EXPRESS с применением небольших и несложных преобразований. Если атрибут является необязательным согласно схеме IFC, то логическое выражение EXPRESS является конъюнкцией вызова функции *EXISTS* для проверки наличия

определенного значения и соответствующей конструкции из таблицы 1 для проверки соблюдения области допустимых значений.

4.4. Представление паттерна *Property*

Рассмотрим возможное представление паттерна *Property* на языке EXPRESS. При этом для краткости ограничимся вариантом, где указываются только имена набора свойств и свойства для проверки их наличия в объекте без указания допустимого типа значений и области их определения. Соответствующая универсальная функция *HasProperty* применяется к объектам-наследникам типа *IfcObjectDefinition* (см. рис. 4). Поскольку в модели данных IFC наборы свойств назначаются по-разному различным потомкам типа *IfcObjectDefinition*: контекстам *IfcContext*, объектам *IfcObject* и их типам *IfcTypeObject* (в первых двух случаях — через отношение *IfcRelDefinesByProperties*, в последнем — напрямую), в ней сначала формируется вспомогательная коллекция назначенных наборов свойств *PropSets*, а затем осуществляется поиск нужного набора и свойства в нем по заданным именам. Функция для поиска свойства в наборе по имени *FindPropertyInSet* является вспомогательной.

Заметим, что в более сложном случае при указании допустимого значения и его типа для проверки области значений дополнительно вызывается функция *GetPropertyValue* (ее спецификация не приводится для краткости изложения), которая возвращает значение найденного свойства, а затем осуществляется проверка типа данных этого значения путем вызова функции *TYPEOF*, а также проверка области определения с помощью соответствующей конструкции согласно табл. 1 в зависимости от вида ограничения области определения значений в паттерне IDS.

4.5. Представление паттерна *Classification*

Паттерн *Classification* выражается на языке EXPRESS с помощью следующих логических выражений. При отсутствии в нем названий классификационной системы и классификатора данное выражение имеет вид:

```
SIZEOF(QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations |
'IFC4.IFCRELASSOCIATESCLASSIFICATION' IN
TYPEOF(temp))) > 0;
```

где *SELF* соответствует объекту одного из типов, унаследованных от *IfcObjectDefinition*, куда классификаторы могут быть назначены через отношение *IfcRelAssociatesClassification* согласно модели данных IFC.

<pre> FUNCTION FindPropertyInSet (PropSet : IfcPropertySetDefinition; Name : STRING) : LOGICAL; LOCAL Properties : SET [1:?] OF IfcProperty; Quantities : SET [1:?] OF IfcPhysicalQuantity; END_LOCAL; IF 'IFC4.IFCPROPERTYSET' IN TYPEOF(PropSet) THEN Properties := PropSet\IfcPropertySet.HasProperties; REPEAT i:=1 TO HIINDEX(Properties); IF Properties[i].Name = Name THEN RETURN (TRUE); END_IF; END_REPEAT; ELSE IF 'IFC4.IFCELEMENTQUANTITY' IN TYPEOF(PropSet) THEN Quantities := PropSet\IfcElementQuantity.Quantities; REPEAT i:=1 TO HIINDEX(Quantities); IF Quantities[i].Name = Name THEN RETURN (TRUE); END_IF; END_REPEAT; END_IF; END_IF; RETURN (FALSE); END_FUNCTION; FUNCTION HasProperty (Obj : IfcObjectDefinition; PropSetName : STRING; PropName : STRING) : LOGICAL; LOCAL PropSets : SET [1:?] OF IfcPropertySetDefinition := []; Relations : SET [0:?] OF IfcRelDefinesByProperties := []; Definition : IfcPropertySetDefinitionSelect; DefinitionSet : IfcPropertySetDefinitionSet; END_LOCAL; </pre>	<pre> IF 'IFC4.IFCTYPEOBJECT' IN TYPEOF(Obj) THEN IF EXISTS(Obj\IfcTypeObject.HasPropertySets) THEN PropSets := Obj\IfcTypeObject.HasPropertySets; ELSE RETURN (FALSE); END_IF; ELSE IF 'IFC4.IFCOBJECT' IN TYPEOF(Obj) THEN Relations := Obj\IfcObject.IsDefinedBy; ELSE Relations := Obj\IfcContext.IsDefinedBy; END_IF; IF SIZEOF(Relations) = 0 THEN RETURN (FALSE); END_IF; REPEAT i:=1 TO HIINDEX(Relations); Definition := Relations[i].RelatingPropertyDefinition; IF 'IFC4.IFCPROPERTYSETDEFINITION' IN TYPEOF(Definition) THEN PropSets := PropSets + Definition; ELSE IF 'IFC4.IFCPROPERTYSETDEFINITIONSET' IN TYPEOF(Definition) THEN DefinitionSet := Definition; REPEAT j:= 1 TO HIINDEX(DefinitionSet); PropSets := PropSets + DefinitionSet[j]; END_REPEAT; END_IF; END_REPEAT; END_IF; REPEAT i:=1 TO HIINDEX(PropSets); IF PropSets[i]\IfcRoot.Name = PropSetName THEN RETURN (FindPropertyInSet(PropSets[i], PropName)); END_IF; END_REPEAT; END_REPEAT; RETURN (FALSE); END_FUNCTION; </pre>
---	--

Рис. 4. Функции на языке EXPRESS для поиска назначенных свойств.

Таблица 1. Отображение паттерна *Attribute* в язык EXPRESS

Параметр <i>Value</i> паттерна <i>Attribute</i> в IDS	Конструкция языка EXPRESS
Отсутствие <i>Name="Description"</i>	Вызов функции <i>EXISTS</i> <i>EXISTS(Description)</i>
Простое значение <i>Name="DurationType", Value="WORKTIME"</i>	Операция сравнения = <i>TaskTime.DurationType = IfcTaskDurationEnum.WORKTIME</i>
Перечисление <i>Name="Status", Value=["NOTSTARTED", "STARTED", "COMPLETED"]</i>	Операция принадлежности множеству <i>IN</i> <i>Status IN ['NOTSTARTED', 'STARTED', 'COMPLETED']</i>
Образец <i>Name="Identification", Value="DT-..-?"</i>	Операция подобия <i>LIKE</i> <i>Identification LIKE 'DT-??-?'</i>
Интервал <i>Name="Priority", Value="0<Value<=100"</i>	Операции сравнения <i>0 < Priority <= 100</i>
Длина <i>Name="Name", Value="MaxLength=50"</i>	Вызов функции <i>LENGTH</i> и операции сравнения <i>LENGTH(Name) <= 50</i>

Если же в паттерне IDS указываются требуемые классификационные системы, например, OmniClass, то выражение имеет следующий вид:

```
SIZEOF(QUERY(temp < *
SELF\IfcObjectDefinition.HasAssociations |
('IFC4.IFCRECLASSOCIATESCLASSIFICATION' IN
TYPEOF(temp)) AND
HasClassificationSystems(temp\IfcRelAssociat
esClassification.RelatingClassification,
['OmniClass'], FALSE))) > 0;
```

Функция *HasClassificationSystems* (см. рис. 5) находит классификационную систему по назначенному классификатору с помощью вспомогательной рекурсивной функции *GetClassificationSystem* и осуществляет сопоставление ее имени с параметром, который может быть задан в паттерне в виде простой строки, перечисления или образца. Перечисление передается в виде множества строк *Systems*. В случае простой строки или образца данное множество состоит из одного элемента. В случае образца параметр *IsPattern* равен *TRUE*.

Если в паттерне указывается допустимое множество классификаторов, которое также может быть задано в виде простой строки, перечисления или образца, то в вышеприведенном логическом

выражении после вызова функции *HasClassificationSystems* через конъюнкцию добавляется вызов функции *HasClassifiers*, которая формирует полное имя назначенного классификатора с учетом иерархии в классификационной системе и осуществляет его сопоставление с соответствующим параметром паттерна. Для краткости данная функция в статье не приводится.

4.6. Представление паттерна Material

Наличие произвольных материалов, назначенных на строительный элемент (если в паттерне IDS не указываются названия материалов), можно проверить с помощью следующего логического выражения на языке EXPRESS:

```
SIZEOF(QUERY(temp < *
SELF\IfcObjectDefinition.HasAssociations
| 'IFC4.IFCRECLASSOCIATESMATERIAL' IN
TYPEOF(temp))) > 0;
```

где *SELF* соответствует объекту одного из типов строительных элементов.

Если в паттерне IDS задаются имена материалов (в виде строки, перечисления или образца), то логическое выражение будет иметь следующий вид:

```
FUNCTION GetClassificationSystem(Ref : IfcClassificationReference) :
IfcClassification;
LOCAL
    Result : IfcClassification;
END_LOCAL;
IF 'IFC4.IFCCLASSIFICATION' IN TYPEOF(Ref.ReferencedSource) THEN
    Result := Ref.ReferencedSource;
ELSE
    Result := GetClassificationSystem(Ref.ReferencedSource);
END_IF;
RETURN(Result);
END_FUNCTION;

FUNCTION HasClassificationSystems(Classifier : IfcClassificationSelect;
    Systems : SET [1:?] OF STRING; IsPattern : BOOLEAN) : LOGICAL;
LOCAL
    Classification : IfcClassification;
END_LOCAL;
IF 'IFC4.IFCCLASSIFICATIONREFERENCE' IN TYPEOF(Classifier) THEN
    Classification := GetClassificationSystem(Classifier);
ELSE
    Classification := Classifier;
END_IF;
IF IsPattern THEN
    IF Classification.Name LIKE Systems[1] THEN
        RETURN (TRUE);
    END_IF;
ELSE
    IF Classification.Name IN Systems THEN
        RETURN (TRUE);
    END_IF;
END_IF;
RETURN (FALSE);
END_FUNCTION;
```

Рис. 5. Функция на языке EXPRESS для сопоставления используемых систем классификации.

```
SIZEOF(QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations
| ('IFC4.IFCRELAASSOCIATESMATERIAL' IN
TYPEOF(temp)) AND
(HasMaterials(temp\IfcRelAssociatesMaterial.RelatingMaterial, ['steel', 'concrete'],
FALSE)))) > 0;
```

Здесь используется приведенный выше пример наличия стальных или бетонных элементов. Функция *HasMaterials* сопоставляет назначенные материалы по имени или категории согласно требованиям стандарта IDS. На рис. 6 приведен небольшой ее фрагмент для двух типов материалов в IFC: простого материала и списка материалов. Остальные 8 типов сложных материалов IFC обрабатываются аналогично: сначала сопоставление идет по собственному имени или категории сложного материала, а затем сопоставляются простые материалы, входящие в состав сложного. Вспомогательная функция *MatchSimpleMaterial* работает только с типом простого материала *IfcMaterial*. Имена для сопоставления передаются как множество строк, что соответствует определению перечисления в IDS. В случае простого

строкового значения это множество состоит из одного элемента. В случае образца во множество включается также один элемент, при этом параметр *IsPattern* равен *TRUE*. Во всех остальных случаях он должен быть равен *FALSE*.

Таким образом, все паттерны, определенные в текущей версии стандарта IDS, могут быть представлены с помощью конструкций языка EXPRESS и определены формально как одноместные объектные предикаты.

5. ФОРМАЛЬНАЯ СПЕЦИФИКАЦИЯ НОРМАТИВОВ И СВОДОВ ПРАВИЛ

В ходе формализации стандарта IDS была разработана библиотека функций на языке EXPRESS, которые проверяют факт выполнения условий, устанавливаемых паттернами, для объектов, типы которых определяются схемой IFC. С математической точки зрения данные функции представляют собой одноместные объектные предикаты, где первым аргументом является некоторый IFC объект, а остальные аргументы являются параметрами соответствующего паттерна. Эти функции можно рассматривать

```
FUNCTION MatchSimpleMaterial(Material : IfcMaterial;
Names : SET [1:?] OF STRING; IsPattern : BOOLEAN) : LOGICAL;
IF IsPattern THEN
  IF Material.Name LIKE Names[1] THEN
    RETURN (TRUE);
  ELSE
    IF (EXISTS(Material.Category)) AND (Material.Category LIKE Names[1]) THEN
      RETURN (TRUE);
    END_IF;
  END_IF;
ELSE
  IF Material.Name IN Names THEN
    RETURN (TRUE);
  ELSE
    IF (EXISTS(Material.Category)) AND (Material.Category IN Names) THEN
      RETURN (TRUE);
    END_IF;
  END_IF;
END_IF;
RETURN (FALSE);
END_FUNCTION;

FUNCTION HasMaterials(Material : IfcMaterialSelect;
Names : SET [1:?] OF STRING; IsPattern : BOOLEAN) : LOGICAL;
IF 'IFC4.IFCMATERIAL' IN TYPEOF(Material) THEN
  RETURN (MatchSimpleMaterial(Material, Names, IsPattern));
ELSE
  IF 'IFC4.IFCMATERIALLIST' IN TYPEOF(Material) THEN
    RETURN (SIZEOF(QUERY(temp <* Material\IfcMaterialList.Materials |
MatchSimpleMaterial(temp, Names, IsPattern))) > 0);
  ELSE
    ...
  END_IF;
RETURN (FALSE);
END_FUNCTION;
```

Рис. 6. Функция на языке EXPRESS для сопоставления материалов по имени или категории.

как расширения стандартной схемы IFC, также специфицированной на языке EXPRESS. Тогда требования могут быть формализованы как правила языка EXPRESS (локальные, глобальные, уникальности), формулируемые с использованием определенного набора предикатов. В таком случае для верификации требований необходим транслятор языка EXPRESS в исполняемый код на одном из императивных языков для генерации части кода приложения верификации либо данное приложение должно включать интерпретатор EXPRESS для непосредственного вычисления соответствующих выражений в процессе работы.

Заметим, что для выражения пространственных условий (допустимого расстояния между объектами или их взаимного расположения) одноместных предикатов недостаточно. Будем предполагать, что для типов объектов, имеющих в IFC геометрическое представление (например, строительные элементы), могут быть реализованы двуместные предикаты, аргументами которых являются два объекта соответствующих типов для проверки выполнимости наложенных пространственных ограничений.

Проиллюстрируем предложенный подход на нескольких примерах из строительных нормативов и сводов правил РФ, предъявляемых к безопасности зданий, сооружений и процессов.

Согласно СП 60.13330.2020 [47] соотношение сторон для воздуховодов прямоугольного сечения не должно превышать 1 к 4. Поскольку данное требование определяет условие, которое должно выполняться для любого воздуховода прямоугольного сечения, то его можно сформулировать

в виде локального правила *CorrectRectangularDuct* для объектного типа *IfcDuctSegment*, определяющего элемент воздуховода. В модели данных IFC стандартные параметры воздуховодов (форма сечения *Shape*, размеры *NominalDiameterOrWidth*, *NominalHeight*) задаются как свойства внутри стандартного набора *PSet_DuctSegmentTypeCommon*, назначаемого объектам типа *IfcDuctSegment*. Тогда в логическом выражении данного правила можно задействовать определенные ранее функции *HasProperty* и *GetPropertyValue*, как показано на рис. 7.

Согласно СП 1.13130.2020 [48] число эвакуационных выходов из здания должно быть не менее числа эвакуационных выходов с любого этажа здания. Чтобы проверить выполнение данного требования, необходимо считать количество объектов в модели (выходов из здания типа *IfcDoor*), удовлетворяющих соответствующему условию. Поэтому в данном случае требование формализуется как глобальное правило *EvacuationRule*, определенное на экстенде объектного типа *IfcDoor* (см. рис. 8).

При проверке условия используются три свойства у объектов-дверей, значения которых извлекаются с помощью определенной ранее функции *GetPropertyValue*:

- 1) идентификатор (как правило, номер) этажа, на котором находится дверь (свойство “*Этаж*” в наборе “*Местоположение*”);
- 2) функциональное назначение двери в соответствии с пунктом 4.2 ГОСТ 475-2016 [49] (свойство “*Назначение*” в наборе “*Идентификация*”, для наружных выходов из здания равно “*H*”);

```
ENTITY IfcDuctSegment SUBTYPE OF (IfcFlowSegment);
  PredefinedType : OPTIONAL IfcDuctSegmentTypeEnum;
WHERE
  CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR
    (PredefinedType <> IfcDuctSegmentTypeEnum.USERDEFINED) OR
    ((PredefinedType = IfcDuctSegmentTypeEnum.USERDEFINED) AND
      EXISTS(SELf\IfcObject.ObjectType));
  CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR
    ('IFCHVACDOMAIN.IfcdDuctSegmentType' IN
      TYPEOF(SELf\IfcObject.IsTypedBy[1].RelatingType));
  CorrectRectangularDuct : HasProperty(SELf, 'Pset_DuctSegmentTypeCommon',
    'Shape') AND
    HasProperty(SELf, 'Pset_DuctSegmentTypeCommon', 'NominalDiameterOrWidth')
AND
  HasProperty(SELf, 'Pset_DuctSegmentTypeCommon', 'NominalHeight') AND
    ((GetPropertyValue(SELf, 'Pset_DuctSegmentTypeCommon', 'Shape') <>
      PEnum_DuctSegmentShape.RECTANGULAR) OR
    (0.25 <= (GetPropertyValue(SELf, 'Pset_DuctSegmentTypeCommon',
      'NominalDiameterOrWidth') / GetPropertyValue(SELf,
      'Pset_DuctSegmentTypeCommon', 'NominalHeight') <= 4.0)));
END_ENTITY;
```

Рис. 7. Формализация требования для допустимого соотношения сторон воздуховодов в виде локального правила на языке EXPRESS.

```

RULE EvacuationRule FOR IfcDoor;
LOCAL
  Storeys : SET OF IfcText := [];
  BuildingExit : INTEGER := 0;
  Result : LOGICAL := TRUE;
  Purpose : IfcText := '';
  Emergency : BOOLEAN := FALSE;
  Storey : IfcText := '';
END_LOCAL;
REPEAT i := 1 TO HIINDEX(IfcDoor);
  Purpose := GetPropertyValue(IfcDoor[i], 'Идентификация', 'Назначение');
  Emergency := GetPropertyValue(IfcDoor[i], 'Пожарные параметры',
'Эвакуационный');
  IF ((Purpose = 'H') AND (Emergency = TRUE)) THEN
    BuildingExit := BuildingExit + 1;
  END_IF;
  Storey := GetPropertyValue(IfcDoor[i], 'Местоположение', 'Этаж');
  IF EXISTS(Storey) THEN
    Storeys := Storeys + Storey;
  END_IF;
END_REPEAT;
REPEAT i := 1 TO HIINDEX(Storeys);
  IF (SIZEOF(QUERY(temp <* IfcDoor |
    ((Storeys[i] = GetPropertyValue(temp, 'Местоположение', 'Этаж')) AND
    (GetPropertyValue(temp, 'Пожарные параметры', 'Эвакуационный') =
TRUE)))) > BuildingExit) THEN
    Result := FALSE;
    ESCAPE;
  END_IF;
END_REPEAT;
WHERE
  WR1 : Result = TRUE;
END_RULE;

```

Рис. 8. Формализация требования для количества эвакуационных выходов из здания в виде глобального правила на языке EXPRESS.

- 3) булевское значение, указывающее, что данная дверь является (или не является) эвакуационным выходом (свойство “Эвакуационный” в наборе “Пожарные параметры”).

В первом цикле подсчитывается количество эвакуационных выходов из здания, а также формируется множество идентификаторов этажей, на которых располагаются двери. Во втором цикле по этажам здания с помощью операции запроса, применяемой к экстену типа *IfcDoor*, формируется множество эвакуационных выходов на каждом этаже, подсчитывается число элементов в нем и сравнивается с подсчитанным ранее количеством эвакуационных выходов из здания.

Согласно требованиям ЦГЭ.ЦИМ-3.0 [50] помещения (тип *IfcSpace*) в модели должны иметь уникальный номер, задаваемый свойством “Номер” в наборе “Идентификация”. Его можно представить в виде правила уникальности в объектном типе *IfcSpace*, как показано на рис. 9, переопределив данное свойство как производный атрибут *SpaceNumber* в данном объектном типе, поскольку в EXPRESS уникальность может устанавливаться только для атрибутов. Для извлечения значения свойства используется определенная ранее функция *GetPropertyValue*.

Рассмотрим, как можно представить требования об отсутствии геометрических коллизий между строительными элементами, например, приведенные в ЦГЭ.ЦИМ-3.0 для стен. Пусть реализована функция для проверки пересечения двух объектов, имеющих геометрическое представление (в IFC это объекты, чей тип является производным от *IfcProduct*) со следующей сигнатурой:

```

FUNCTION LieCloserThan (Obj1: IfcProduct; Obj2
: IfcProduct; Distance: REAL): LOGICAL;

```

Тогда данное требование формулируется как глобальное правило на экстене объектного типа *IfcWall*, что продемонстрировано на рис. 10.

Таким образом, использование языка EXPRESS в сочетании со вспомогательными одноместными и двуместными предикатами позволяет формализовать сложные требования, что было невозможно в рамках стандарта IDS.

6. АПРОБАЦИЯ ПРЕДЛОЖЕННОГО ПОДХОДА

Благодаря проведенной формализации стандарта IDS перспективной видится идея его гармонизации с предложенным формальным

```

ENTITY IfcSpace SUBTYPE OF (IfcSpatialStructureElement);
  PredefinedType : OPTIONAL IfcSpaceTypeEnum;
  ElevationWithFlooring : OPTIONAL IfcLengthMeasure;
  INVERSE
  HasCoverings : SET [0:?] OF IfcRelCoversSpaces FOR RelatingSpace;
  BoundedBy : SET [0:?] OF IfcRelSpaceBoundary FOR RelatingSpace;
  DERIVE
  SpaceNumber : IfcText := GetPropertyValue(SELF, 'Идентификация', 'Номер');
  UNIQUE
  UR1 : SpaceNumber;
  WHERE
  CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR
    (PredefinedType <> IfcSpaceTypeEnum.USERDEFINED) OR
    ((PredefinedType = IfcSpaceTypeEnum.USERDEFINED) AND
      EXISTS(SELF\IfcObject.ObjectType));
  CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR
    ('IFCPRODUCTEXTENSION.IfSpaceType' IN
      TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));
END ENTITY;

```

Рис. 9. Формализация требования уникальности номеров помещений в виде правила уникальности на языке EXPRESS.

```

RULE CollisionMissingRule FOR IfcWall;
LOCAL
  Result : LOGICAL := TRUE;
END_LOCAL;
  REPEAT i := 1 TO HIINDEX(IfcWall);
    REPEAT j := i + 1 TO HIINDEX(IfcWall);
      IF LieCloserThan(IfcWall[i], IfcWall[j],
0.02) THEN
        Result := FALSE;
        ESCAPE;
      END_IF;
    END_REPEAT;
    IF NOT(Result) THEN
      ESCAPE;
    END_IF;
  END_REPEAT;
  WHERE
    WR1 : Result = TRUE;
END_RULE;

```

Рис. 10. Формализация требования об отсутствии геометрических коллизий между стенами на языке EXPRESS.

подходом. Подобная гармонизация способна обеспечить простоту задания типовых требований, универсальность и гибкость алгебраической спецификации сложных требований, а также строгость и достоверность верификации цифровых моделей, тем самым, сочетая в себе достоинства декларативных и императивных средств.

Предлагаемый способ гармонизации состоит в развитии стандарта IDS и его расширении следующими возможностями:

- поддержка нескольких контекстов применимости требований, что необходимо, например, для определения пространственных и топологических условий;
- расширение набора паттернов IDS, включая, как минимум, паттерны, соответствующие трем видам декларативных правил языка EXPRESS (локальные, глобальные, уникальности);

- разработка библиотеки функций на языке EXPRESS, являющихся расширением стандартной схемы IFC и реализующих предикаты для типовых пространственных, топологических, метрических и прочих условий.

Предложенные идеи были успешно апробированы в ходе реализации редактора машиночитаемых требований ИСП РАН, который обеспечивает просмотр, модификацию спецификаций требований в соответствии со стандартом IDS, а также их документирование в виде файлов HTML на русском и английском языках. На рис. 11 показан снимок главного окна перспективного редактора. В верхней части снимка располагается общий заголовок документа, включающий параметры в соответствии со стандартом IDS, как описано в разделе 2. В качестве дополнительной опции указывается количество контекстов, необходимое для задания спецификации.

В таблице спецификаций, располагаемой ниже слева, выводятся название спецификации и ее статус, указывающий на соответствие стандарту и возможность ее корректного сохранения в виде XML файла. Для создания, редактирования, перемещения и удаления спецификаций предусмотрены меню команд и панель инструментов. Справа от таблицы выводится заголовок выделенной спецификации и ее текстовая нотация, дающая полное и непротиворечивое описание соответствующего требования.

Для просмотра и редактирования конкретной спецификации предназначено модальное окно с вкладками доступа к ее заголовку, контекстам и требованию (см. рис. 12). Задание условий на категории объектов, отношения, классификаторы, атрибуты, свойства и материалы осуществляется

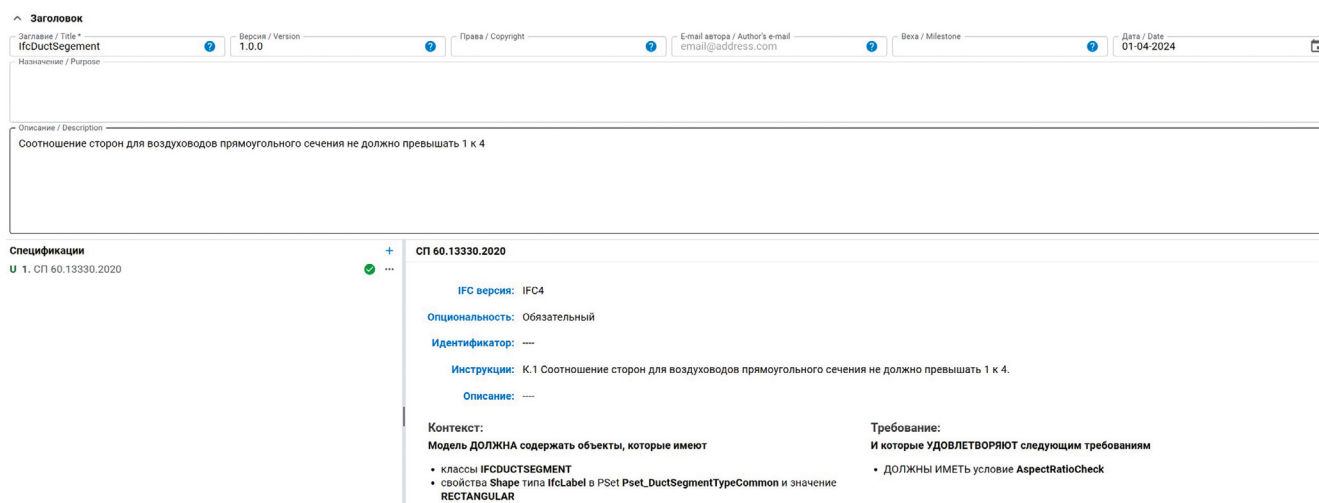


Рис. 11. Экранный снимок главного окна перспективного редактора требований.

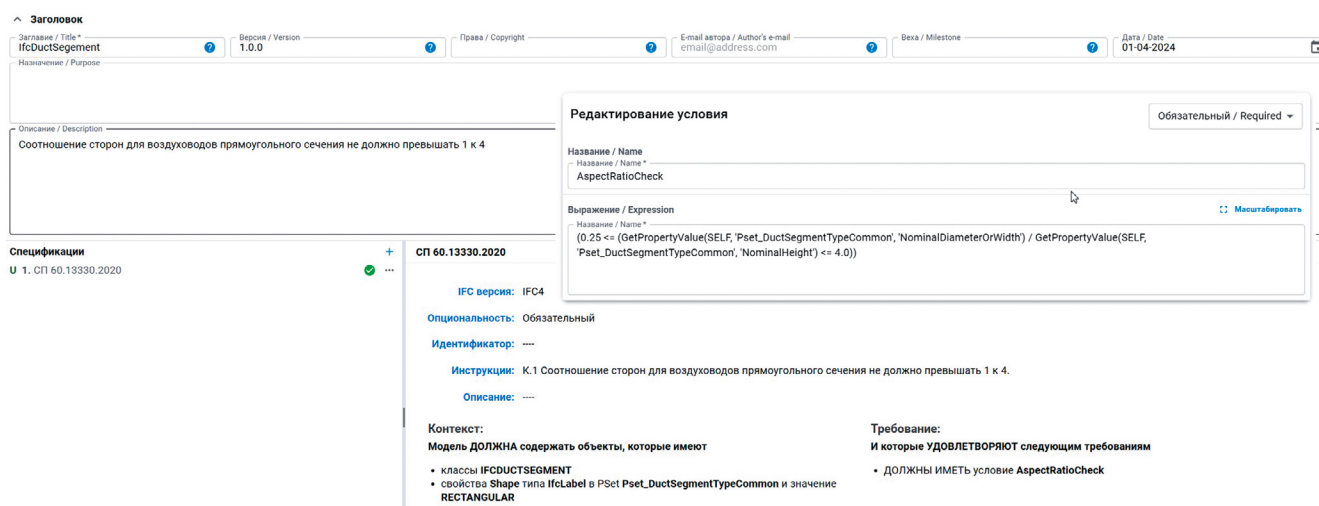


Рис. 12. Экранный снимок окна редактора спецификаций требований.

с помощью стандартных IDS паттернов, для поддержки которых предусмотрены меню в таблице условий и соответствующие элементы редактирования снизу. Для быстрого и верного ввода названий простых и объектных типов данных, а также многочисленных атрибутов, определяемых IFC схемой, в элементах редактирования реализована контекстная помощь, позволяющая выбирать допустимые названия из предлагаемого списка или иерархии.

Для реализации предлагаемого формального подхода в поддерживаемый набор стандартных IDS паттернов добавлено три новых паттерна, соответствующих декларативным правилам языка EXPRESS. При инстанцировании соответствующего правила в нижней части диалога отображаются элементы для ввода его названия и тела спецификации на языке EXPRESS. Парсер позволяет оперативно выявить допущенные пользова-

телем синтаксические ошибки, а семантический контроль осуществляется на этапе верификации цифровой модели и интерпретации спецификации требования. На приведенном скриншоте показано представление требования соотношения сторон для воздуховодов прямоугольного сечения, ранее рассмотренного в разделе 5.

7. ЗАКЛЮЧЕНИЕ

Таким образом, в статье проведен сравнительный анализ программных инструментов для автоматизированной проверки нормативных требований в области архитектуры и строительства. Отмечена возросшая популярность инструментов, ориентированных на международные открытые стандарты IFC, IDS и BCF, а также выделены их ограничения для верификации сложных требований. Исследованы возможности применения

языка моделирования EXPRESS для формальной спецификации и верификации требований в архитектуре и строительстве, а также определены перспективы и способы гармонизации предложенного формального подхода с актуальным стандартом IDS. Планируемые дальнейшие исследования будут связаны с полнофункциональной реализацией редактора машиночитаемых требований с использованием декларативных правил языка EXPRESS, а также соответствующих сервисов верификации цифровых моделей, представленных в соответствии со стандартом IFC.

СПИСОК ЛИТЕРАТУРЫ

1. About buildingSMART International. <https://www.buildingsmart.org/about>, 22.04.2024.
2. ISO 16739-1:2018. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema. Edition 1. 1474 p.
3. ISO 29481-1:2016. Building information models – Information delivery manual – Part 1: Methodology and format. Edition 2. 29 p.
4. BIM Collaboration Format (BCF) – buildingSMART Technical. <https://technical.buildingsmart.org/standards/bcf>, 22.04.2024.
5. buildingSMART Data Dictionary (bSDD) – buildingSMART Technical. <https://technical.buildingsmart.org/services/bsdd>, 22.04.2024.
6. Eichler C.C., Schranz Ch., Krischmann T., Urban H., Hopferwieser M., Fischer S. BIMcert Handbook – Basic knowledge openBIM. Edition 2024. Mironde-Verlag, Niederfrohna. 2024. 224 p. <https://doi.org/10.34726/5383>.
7. Nisbet N. Using uncertainty to link compliance and creativity. ECPPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling. 2021. CRC Press, London. P. 29–34. <https://doi.org/10.1201/9781003191476-4>.
8. ISO/IEC/IEEE29148:2018. Systems and software engineering – Life cycle processes – Requirements engineering. Edition 2. 92 p.
9. Hull E., Jackson K., Dick J. Requirements Engineering, Third Edition. Springer-Verlag, London. 2011. 207 p.
10. Kuliamin V.V. Integration of verification methods for program systems. Programming and Computer Software. 2009. V. 35. № 4. P. 212–222. <https://doi.org/10.1134/S0361768809040057>.
11. ISO/IEC19505-1:2012. Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure. Edition 1. 220 p.
12. ISO/IEC19507:2012. Information technology – Object Management Group Object Constraint Language (OMG OCL). Edition 1. 233 p.
13. ISO 10303-11:2004. Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. Edition 2. 255 p.
14. EXPRESS Data Manager™ – Jotne Connect. <https://jotneit.com/products/express-data-manager>, 22.04.2024.
15. Soliman-Junior J., Tzortzopoulos P., Baldauf J.P., Pedro B., Kagioglou M., Formoso C.T., Humphreys J. Automated compliance checking in healthcare building design. Automation in Construction. 2021. V. 129. <https://doi.org/10.1016/j.autcon.2021.103822>.
16. 7D Modeler | Open BIM Systems. <https://www.openbimsystems.ru/en/7d-modeler>, 22.04.2024.
17. BIMcollab Zoom: model validation that gets issues solved. <https://www.bimcollab.com/en/products/bimcollab-zoom>, 22.04.2024.
18. BlenderBIM Add-on – beautiful, detailed, and data-rich OpenBIM. <https://blenderbim.org>, 22.04.2024.
19. Open IFC Viewer. <https://openifcviewer.com>, 22.04.2024.
20. IFC Checker and BIM Validation | usBIM.checker | ACCA software. <https://www.accasoftware.com/en/ifc-checker>, 22.04.2024.
21. Li B., Schultz C., Dimyadi J., Amor R. Defeasible reasoning for automated building code compliance checking. ECPPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling. 2021. CRC Press, London. P. 229–236. <https://doi.org/10.1201/9781003191476-32>.
22. Information Delivery Specification IDS – buildingSMART Technical. <https://technical.buildingsmart.org/projects/information-delivery-specification-ids>, 22.04.2024.
23. Национальная ТИМ Платформа | Сервис валидации IFC моделей. URL: <https://bim.ispras.ru/validate-ifc>, 22.04.2024.
24. van Berlo L., Willems P., Pauwels P. Creating Information Delivery Specifications using linked data. Proceedings of the 36th International CIB W78 Conference. 2019. P. 647–660.
25. Kremer N., Beetz J. Extending Information Delivery Specification for linking distributed model checking services. Proceedings of the 40th International CIB W78 Conference. 2023. <https://doi.org/10.35490/EC3.2023.266>.
26. ГОСТ Р 10.0.02-2019/ИСО 16739-1:2018. Система стандартов информационного моделирования зданий и сооружений. Отраслевые базовые классы (IFC) для обмена и управления данными об объектах строительства. Часть 1. Схема данных. 28 с.
27. Software Implementations – buildingSMART Technical. <https://technical.buildingsmart.org/resources/software-implementations>, 22.04.2024.

28. ISO 16739-1:2024. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema. Edition 2. 496 p.
29. ISO 10303-1:2024. Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Edition 3. 23 p.
30. ISO 10303-21:2016. Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure. Edition 3. 3 p.
31. ISO 10303-28:2007. Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas. Edition 1. 309 p.
32. ISO/TS10303-26:2011. Industrial automation systems – Product data representation and exchange – Part 26: Implementation methods: Binary representation of EXPRESS-driven data. Edition 1. 5 p.
33. *Staab S., Studer R. (eds.) Handbook on Ontologies, Second Edition.* Springer-Verlag, Berlin. 2009. 811 p.
34. IFC Implementation Guidance – buildingSMART Technical. <https://technical.buildingsmart.org/resources/ifcimplementationguidance>, 22.04.2024.
35. *van Berlo L.A.H.M., Beetz J., Bos P., Hendriks H., van Tongeren R.C.J.* Collaborative engineering with IFC: New insights and technology. ECPPM 2012 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 9th European Conference on Product & Process Modelling. 2012. CRC Press, London. P. 811–818.
36. *Borrmann A., Beetz J., Koch C., Liebich T., Muhic S.* Industry Foundation Classes – A standardized data model for the vendor-neutral exchange of digital building models. Building Information Modeling – Technology Foundations and Industry Practice. Springer, Cham. 2018. P. 81–126. https://doi.org/10.1007/978-3-319-92862-3_5.
37. *van Berlo L., Krijnen T.F., Tauscher H., Liebich T., van Kranenburg A., Paasiala P.* Future of the Industry Foundation Classes: towards IFC5. Proceedings of the 38th International CIB W78 Conference. 2021. P. 123–137.
38. UniFormat® – Construction Specifications Institute. <https://www.csiresources.org/standards/uniformat>, 22.04.2024.
39. OmniClass® – Construction Specifications Institute. <https://www.csiresources.org/standards/omniclass>, 22.04.2024.
40. Классификатор Строительной Информации. <http://ksi.faufcc.ru>, 22.04.2024.
41. *Tomczak A., Benghi C., van Berlo L., Hjelseth E.* Requiring Circularity Data in BIM with Information Delivery Specification. Journal of Circular Economy. 2024. V. 1. N. 2. P. 1–13. <https://doi.org/10.55845/REJY5239>.
42. IDS – Machine readable Information Delivery Specification. URL: <https://github.com/buildingSMART/IDS/blob/master/Documentation>, 22.04.2024.
43. BIM Collaboration Format v3.0 Technical Documentation. https://github.com/buildingSMART/BCF-XML/tree/release_3_0/Documentation, 22.04.2024.
44. *Semenov V., Ilyin D., Morozov S., Tarlapan O.* Effective consistency management for large-scale product data // Journal of Industrial Information Integration. 2019. V. 13. P. 13–21. <https://doi.org/10.1016/j.jii.2018.11.006>.
45. *Морозов С.В., Ильин Д.В., Семенов В.А., Тарлапан О.А.* Библиотека ограничений для спецификации индустриальных моделей данных. Труды Института системного программирования РАН. 2015. Т. 27. № 4. С. 69–110. [https://doi.org/10.15514/ISPRAS-2015-27\(4\)-5](https://doi.org/10.15514/ISPRAS-2015-27(4)-5).
46. *Semenov V.A., Arishin S.V., Semenov G.V.* Formal rules to produce object notation for EXPRESS schema-driven data. Programming and Computer Software. 2022. V. 48. N. 7. P. 455–468. <https://doi.org/10.1134/S0361768822070076>.
47. Свод правил СП 60.13330.2020. Отопление, вентиляция и кондиционирование воздуха. 149 с.
48. Свод правил СП 1.13130.2020. Системы противопожарной защиты. Эвакуационные пути и выходы. 46 с.
49. ГОСТ 475–2016. Блоки дверные деревянные и комбинированные. Общие технические условия. 35 с.
50. Требования к цифровым информационным моделям объектов капитального строительства, предоставляемым для проведения экспертизы. Редакция 3.0 (ЦГЭ.ЦИМ-3.0). СПб ГАУ “Центр Государственной Экспертизы”. 2022. 203 с.

FORMAL SPECIFICATION AND VERIFICATION OF REQUIREMENTS IN ARCHITECTURE AND CONSTRUCTION USING THE EXPRESS MODELING LANGUAGE

© 2024 V. A. Semenov^{a,*}, S. V. Morozov^{a,**}, S. V. Arishin^{a,***},
O. N. Kuzina^{b,****}, V. I. Rimshin^{b,*****}, E. V. Makisha^{b,*****}

^a*Ivannikov Institute for System Programming, RAS*

25, Alexander Solzhenitsyn str., Moscow, 109004 Russia

^b*National Research University Moscow State University of Civil Engineering*

26, Yaroslavskoye shosse, Moscow, 129337 Russia

Currently, digital technologies for modeling buildings and infrastructure are successfully used in international and national practice in the implementation of complex construction projects and large-scale programmes. At the same time, the transition to machine-readable standards being implemented in many countries in order to improve the quality of design documentation and to automate its verification faces serious methodological and technical problems. First of all, they are associated with the complexity of the digital models, as well as with the variety of requirements formulated in natural languages and imposed on models at the state, regional, departmental and corporate levels. Attempts to create catalogues of requirements and software tools for their maintenance and use, usually have specialized nature and do not provide the necessary completeness, normalization, consistency, linked set, unambiguity, traceability and validity of requirement descriptions. In this regard, it seems constructive to use formal methods for specification and verification of requirements that have proven themselves in system and software engineering. The paper provides a comparative analysis of software tools for automated verification of regulatory requirements in the construction domain. There is an increased popularity of tools focused on the international standards IFC (Industry Foundation Classes), IDS (Information Delivery Specification) and providing control of the completeness of the object and attribute composition of models, as well as clarification of acceptable ranges of values. At the same time, the IDS standard is not formalized and does not provide for specifying the requirements expressed by arbitrary algebraic conditions. The use of the object-oriented data modeling language EXPRESS, in which the IFC information schema is also specified, looks promising for the formal specification and verification of requirements for digital models in construction. As a justification, it is shown that IDS specifications can be represented by logical expressions and EXPRESS functions, as well as arbitrary algebraic conditions can be specified as EXPRESS declarative rules. Examples of the formalization of some requirements from national construction standards and set of rules on the safety of buildings, structures and processes are provided in the paper as illustrations of the proposed approach. The possibilities of harmonizing the proposed formal approach with the IDS standard as a result of defining new facets for representing EXPRESS local, unique and global rules are also discussed.

Keywords: BIM, IFC, EXPRESS, IDS, specification and verification, requirements engineering

REFERENCES

1. About buildingSMART International. <https://www.buildingsmart.org/about>, 22.04.2024.
2. ISO 16739-1:2018. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema. Edition 1. 1474 p.
3. ISO 29481-1:2016. Building information models – Information delivery manual – Part 1: Methodology and format. Edition 2. 29 p.
4. BIM Collaboration Format (BCF) – buildingSMART Technical. <https://technical.buildingsmart.org/standards/bcf>, 22.04.2024.
5. buildingSMART Data Dictionary (bSDD) – buildingSMART Technical. <https://technical.buildingsmart.org/services/bsdd>, 22.04.2024.
6. Eichler C.C., Schranz Ch., Krischmann T., Urban H., Hopferwieser M., Fischer S. BIMcert Handbook – Basic knowledge openBIM. Edition 2024. Mironde-Verlag, Niederfrohna. 2024. 224 p. <https://doi.org/10.34726/5383>.
7. Nisbet N. Using uncertainty to link compliance and creativity. ECPPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling. 2021. CRC Press, London. P. 29–34. <https://doi.org/10.1201/9781003191476-4>.
8. ISO/IEC/IEEE29148:2018. Systems and software engineering – Life cycle processes – Requirements engineering. Edition 2. 92 p.
9. Hull E., Jackson K., Dick J. Requirements Engineering, Third Edition. Springer-Verlag, London. 2011. 207 p.
10. Kuliamin V.V. Integration of verification methods for program systems-. Programming and Computer Software. 2009. V. 35. № 4. P. 212–222. <https://doi.org/10.1134/S0361768809040057>.

11. ISO/IEC19505-1:2012. Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure. Edition 1. 220 p.
12. ISO/IEC19507:2012. Information technology – Object Management Group Object Constraint Language (OMG OCL). Edition 1. 233 p.
13. ISO 10303-11:2004. Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. Edition 2. 255 p.
14. EXPRESS Data Manager™ – Jotne Connect. <https://jotneit.com/products/express-data-manager>, 22.04.2024.
15. *Soliman-Junior J., Tzortzopoulos P., Baldauf J.P., Pedro B., Kagioglou M., Formoso C.T., Humphreys J.* Automated compliance checking in healthcare building design. *Automation in Construction*. 2021. V. 129. <https://doi.org/10.1016/j.autcon.2021.103822>.
16. 7D Modeler | Open BIM Systems. <https://www.openbimsystems.ru/en/7d-modeler>, 22.04.2024.
17. BIMcollab Zoom: model validation that gets issues solved. <https://www.bimcollab.com/en/products/bimcollab-zoom>, 22.04.2024.
18. BlenderBIM Add-on – beautiful, detailed, and data-rich OpenBIM. <https://blenderbim.org>, 22.04.2024.
19. Open IFC Viewer. <https://openifcviewer.com>, 22.04.2024.
20. IFC Checker and BIM Validation | usBIM.checker | ACCA software. <https://www.accasoftware.com/en/ifc-checker>, 22.04.2024.
21. *Li B., Schultz C., Dimyadi J., Amor R.* Defeasible reasoning for automated building code compliance checking. *ECPPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling*. 2021. CRC Press, London. P. 229–236. <https://doi.org/10.1201/9781003191476-32>.
22. Information Delivery Specification IDS – buildingSMART Technical. <https://technical.buildingsmart.org/projects/information-delivery-specification-ids>, 22.04.2024.
23. Natsional'naya TIM Platforma | Servis validatsii IFC modelej [National BIM Platform | IFC model validation service] (in Russian). URL: <https://bim.ispras.ru/validate-ifc>, accessed 22.04.2024.
24. *van Berlo L., Willems P., Pauwels P.* Creating Information Delivery Specifications using linked data. *Proceedings of the 36th International CIB W78 Conference*. 2019. P. 647–660.
25. *Kremer N., Beetz J.* Extending Information Delivery Specification for linking distributed model checking services. *Proceedings of the 40th International CIB W78 Conference*. 2023. <https://doi.org/10.35490/EC3.2023.266>.
26. GOST R10.0.02-2019/ISO 16739-1:2018 Sistema standartov informatsionnogo modelirovaniya zdaniy i sooruzhenij. Otrazlevye bazovye klassy (IFC) dlya obmena i upravleniya dannymi ob ob'ektah stroitel'stva. Chast' 1. Shema dannyh [GOST R 10.0.02-2019/ISO 16739-1:2018. System of standards for building information modeling. Industry Foundation Classes (IFC) for exchanging and managing construction site data. Part 1. Data schema]. 28 p. (in Russian).
27. Software Implementations – buildingSMART Technical. <https://technical.buildingsmart.org/resources/software-implementations>, accessed 22.04.2024.
28. ISO 16739-1:2024. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema. Edition 2. 496 p.
29. ISO 10303-1:2024. Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Edition 3. 23 p.
30. ISO 10303-21:2016. Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure. Edition 3. 3 p.
31. ISO 10303-28:2007. Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas. Edition 1. 309 p.
32. ISO/TS10303-26:2011. Industrial automation systems – Product data representation and exchange – Part 26: Implementation methods: Binary representation of EXPRESS-driven data. Edition 1. 5 p.
33. *Staab S., Studer R. (eds.)* Handbook on Ontologies, Second Edition. Springer-Verlag, Berlin. 2009. 811 p.
34. IFC Implementation Guidance – buildingSMART Technical. <https://technical.buildingsmart.org/resources/ifcimplementationguidance>, accessed 22.04.2024.
35. *van Berlo L.A.H.M., Beetz J., Bos P., Hendriks H., van Tongeren R.C.J.* Collaborative engineering with IFC: New insights and technology. *ECPPM 2012 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 9th European Conference on Product & Process Modelling*. 2012. CRC Press, London. P. 811–818.
36. *Borrmann A., Beetz J., Koch C., Liebich T., Muhic S.* Industry Foundation Classes – A standardized data model for the vendor-neutral exchange of digital building models. *Building Information Modeling – Technology Foundations and Industry Practice*. Springer, Cham. 2018. P. 81–126. https://doi.org/10.1007/978-3-319-92862-3_5
37. *van Berlo L., Krijnen T.F., Tauscher H., Liebich T., van Kranenburg A., Paasiala P.* Future of the Industry Foundation Classes: towards IFC5. *Proceedings of the 38th International CIB W78 Conference*. 2021. P. 123–137.
38. UniFormat® – Construction Specifications Institute. <https://www.csiresources.org/standards/uniformat>, accessed 22.04.2024.

39. OmniClass® – Construction Specifications Institute. <https://www.csiresources.org/standards/omniclass>, accessed 22.04.2024.
40. Klassifikator Stroitel'noj Informatsii [Construction Information Classifier] (in Russian). <http://ksi.faufcc.ru>, accessed 22.04.2024.
41. Tomczak A., Benghi C., van Berlo L., Hjelseth E. Requiring Circularity Data in BIM with Information Delivery Specification. *Journal of Circular Economy*. 2024. V. 1. № 2. P. 1–13. <https://doi.org/10.55845/REJY5239>
42. IDS – Machine readable Information Delivery Specification. URL: <https://github.com/buildingSMART/IDS/blob/master/Documentation>, accessed 22.04.2024.
43. BIM Collaboration Format v3.0 Technical Documentation. https://github.com/buildingSMART/BCF-XML/tree/release_3_0/Documentation, accessed 22.04.2024.
44. Semenov V., Ilyin D., Morozov S., Tarlapan O. Effective consistency management for large-scale product data // *Journal of Industrial Information Integration*. 2019. V. 13. P. 13–21. <https://doi.org/10.1016/j.jii.2018.11.006>.
45. Morozov S.V., Ilyin D.V., Semenov V.A., Tarlapan O.A. Biblioteka ogranichenij dlya spetsifikatsii industrial'nyh modelej dannyh [A constraint library for specification of industrial data models]. 2015. V. 27. N. 4. P. 69–110 (in Russian). [https://doi.org/10.15514/ISPRAS-2015-27\(4\)-5](https://doi.org/10.15514/ISPRAS-2015-27(4)-5).
46. Semenov V.A., Arishin S.V., Semenov G.V. Formal rules to produce object notation for EXPRESS schema-driven data. *Programming and Computer Software*. 2022. V. 48. № 7. P. 455–468. <https://doi.org/10.1134/S0361768822070076>.
47. Svod pravil SP 60.13330.2020. Otoplenie, ventilyatsiya i konditsionirovanie vozduha [Set of rules SP 60.13330.2020. Heating, ventilation and air conditioning]. 149 p. (in Russian).
48. Svod pravil SP 1.13130.2020. Sistemy protivopozharnoj zashchity. Evakuatsionnye puti i vyhody. [Set of rules SP 1.13130.2020. Fire protection systems. Evacuation routes and exits]. 46 p. (in Russian).
49. GOST 475-2016. Bloki dvernye derevyannye i kombinirovannye. Obshchie tehicheskie usloviya [GOST 475–2016. Wooden and combined door blocks. General technical conditions]. 35 p (in Russian).
50. Trebovaniya k tsifrovym informatsionnym modelyam ob'ektov kapital'nogo stroitel'stva, predstavlyaemym dlya provedeniya ekspertizy. Redaktsiya 3.0 (TGE.TIM-3.0) [Requirements for digital information models of capital construction projects submitted for expertise. Edition 3.0 (TGE.TIM-3.0)]. SPb GAU “Tsentr Gosudarstvennoj Ekspertizy” [SPb SAI “Center for State Expertise”]. 2022. 203 p (in Russian).