



Машинно-интерпретируемый формат IDS++ для спецификации и верификации требований в строительной отрасли: идея, концепция, реализация

^{1,2} В.А. Семенов, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

¹ С.В. Аришин, ORCID: 0000-0001-6128-7082 <arishin@ispras.ru>

¹ Е.М. Зорин, ORCID: 0009-0000-6614-0485 <zorin.em@ispras.ru>

^{1,2,3} С.В. Морозов, ORCID: 0000-0002-0211-2300 <serg@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

² Московский физико-технический институт,
Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., д. 9.

³ Московский государственный университет им. М.В. Ломоносова,
Россия, 119991, Москва, Ленинские горы, д. 1.

Аннотация. В последнее время в российской и зарубежной строительной практике широкое распространение получили программные инструменты спецификации и верификации требований на основе машинно-интерпретируемых форматов, обеспечивающих достоверную и прослеживаемую проверку цифровых информационных моделей (ЦИМ). Наиболее известным среди них является формат IDS, который позволяет представить требования к информационному наполнению ЦИМ, сформированной в соответствии с открытым международным стандартом IFC. Однако на сегодняшний день возможности стандарта IDS крайне ограничены. Чтобы преодолеть имеющиеся ограничения и обеспечить более полную и всестороннюю работу со сложными нормативными требованиями, выражаемыми параметрическими, геометрическими, топологическими, пространственными и иными алгебраическими условиями, разработан машинно-интерпретируемый формат IDS++. В статье описываются концепция и основные конструкции нового формата, а также обсуждаются разработанные программные приложения, предназначенные для спецификации и верификации требований в строительной отрасли.

Ключевые слова: цифровая модель BIM; стандарт IFC; спецификация IDS; формат IDS++; цифровизация требований; формальная верификация; машинно-интерпретируемые форматы.

Для цитирования: Семенов В.А., Аришин С.В., Зорин Е.М., Морозов С.В. Машинно-интерпретируемый формат IDS++ для спецификации и верификации требований в строительной отрасли: идея, концепция, реализация. Труды ИСП РАН, том 38, вып. 4, часть 1, 2026 г., стр. 201–224. DOI: 10.15514/ISPRAS-2026-38(4)-11.

Благодарности: Результаты получены с использованием услуг Центра коллективного пользования Института системного программирования им. В.П. Иванникова РАН – ЦКП ИСП РАН.

Computer-interpretable format IDS++ for specification and verification of requirements in construction: idea, concept, implementation

^{1,2} V.A. Semenov, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

¹ S.V. Arishin, ORCID: 0000-0001-6128-7082 <arishin@ispras.ru>

¹ E.M. Zorin, ORCID: 0009-0000-6614-0485 <zorin.em@ispras.ru>

^{1,2,3} S.V. Morozov, ORCID: 0000-0002-0211-2300 <serg@ispras.ru>

¹ Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

² Moscow Institute of Physics and Technology,
9, Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.

³ Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia.

Abstract. Recently, software tools for requirements specification and verification based on computer-interpretable formats, which enable reliable and traceable verification of building information models (BIM), have become widely used in Russian and international construction practice. The most well-known of these is the IDS format, which enables the presentation of requirements for the information content of a BIM formed in accordance with the open international standard IFC. However, today the capabilities of the IDS standard are extremely limited. To overcome these limitations and ensure more complete and comprehensive support for sophisticated regulatory requirements expressed through parametric, geometric, topological, spatial, and other algebraic constraints, the computer-interpretable format IDS++ has been developed. The article describes the concept and basic structures of the new format and discusses the developed software applications designed for specification and verification of requirements in the construction industry.

Keywords: BIM; IFC; IDS; IDS++; digitalization of requirements; formal verification; computer-interpretable formats.

For citation: Semenov V.A., Arishin S.V., Zorin E.M., Morozov S.V. Computer-interpretable format IDS++ for specification and verification of requirements in construction: idea, concept, implementation. *Trudy ISP RAN/Proc. ISP RAS*, vol. 38, issue 4, part 1, 2026. pp. 201–224 (in Russian). DOI: 10.15514/ISPRAS-2026-38(4)-11.

Acknowledgements: The results were obtained using the equipment of the Shared Research Center of the Ivannikov Institute for System Programming of the Russian Academy of Sciences (SRC ISP RAS).

1. Введение

Цифровые технологии получают все большее распространение в промышленности, науке, бизнесе, образовании и медицине. В их основе лежат информационные модели (объектно-ориентированные модели, реляционные схемы, онтологии) предметных областей, прикладных задач, проектов, процессов, требований, замечаний, как правило, специфицированные на одном из декларативных языков моделирования и реализованные в составе целевого прикладного программного обеспечения (ПО) с использованием модельно-ориентированных технологий программной инженерии на императивных языках программирования [1]. Репрезентативные информационные модели, принятые отраслевыми сообществами в качестве стандартных вместе с форматами обменных файлов и программными интерфейсами совместного доступа к данным, составляют конструктивную основу обеспечения interoperability ПО и формирования соответствующих цифровых экосистем [2–3]. Вместе с тем, возникают важные возможности для высокоавтоматизированной подготовки, обработки и визуализации прикладных данных, определяемых формально специфицированной моделью. В частности, проверка прикладных данных на соответствие алгебраическим условиям, накладываемым на них, или требованиям,

предъявляемым к ним, может осуществляться на основе методов формальной верификации [4-5].

Данная тенденция характерна для строительной отрасли, в которой проектно-конструкторская документация в виде цифровых информационных моделей зданий, сооружений и инфраструктуры (ЦИМ) разрабатывается в соответствующих САПР с использованием проприетарных форматов, а при обмене между участниками проектной деятельности и создании электронных архивов, как правило, преобразуются в открытые форматы в соответствии с международными и национальными стандартами по функциональной совместимости (интероперабельности) ПО. Ключевым международным стандартом по обеспечению интероперабельности ПО в строительной отрасли является Industry Foundation Classes (IFC) [6-7], специфицирующий информационную схему для представления ЦИМ объектов капитального строительства (ОКС) на декларативном языке объектно-ориентированного моделирования данных EXPRESS [8]. Информационная схема IFC, как и язык EXPRESS, приняты также в качестве соответствующих ГОСТов в РФ [9-10]. В настоящее время сформирована международная и национальная экосистемы ПО, поддерживающего IFC. Они включают более 300 программных приложений и систем.

Для автоматизации проверок ЦИМ на соблюдение нормативных требований последние должны быть формализованы и представлены в цифровом виде. Однако в настоящее время отсутствует единый и системный подход к верификации ЦИМ с соблюдением полноты, однозначности и непротиворечивости спецификации требований, корректности их интерпретации, получения достоверных и прослеживаемых результатов проверок, а также с обеспечением прозрачного обмена требованиями между участниками проектной деятельности. По используемым методам существующие инструменты спецификации и верификации требований в строительной отрасли можно условно разделить на четыре категории.

Первую категорию составляют программные инструменты, которые основаны на методах, подобных используемым для формальной верификации статических моделей в программной инженерии [11-12]. Достоинствами данных инструментов является математическая строгость спецификаций требований и однозначность их интерпретации. Однако применение инструментов общего назначения, в основном ориентированных на язык UML/OCL, к задачам верификации ЦИМ с информационными схемами на языке EXPRESS порождает проблему несоответствия метамоделей (impedance mismatch) и связанную с ней неопределенность в результатах верификации. Система EDM Express Data Checker [13], относящаяся к данной категории, позволяет специфицировать требования непосредственно на языке EXPRESS, на котором описана и сама схема IFC. Это решает проблему несоответствия метамоделей. Однако, эта система не учитывает особенности схемы IFC и усложняет спецификацию типовых требований, относящихся к таким понятиям, как свойства (*IfcProperty*, *IfcPropertySet*), классификаторы (*IfcClassification*, *IfcClassificationSystem*), объективированные взаимосвязи (*IfcRelationship*), предопределенные типы (*IfcWallTypeEnum*, *IfcDoorTypeEnum* и т.д.). К тому же ее применение предполагает знание языка EXPRESS и формальных методов, которые довольно трудны в освоении специалистами строительной отрасли, не обладающими соответствующей IT квалификацией. Ко второй категории относятся программные инструменты, имеющие в своем функционале процедуры проверки типовых требований для IFC моделей. В качестве самого популярного представителя данной категории следует указать систему Solibri Model Checker [14], которая поддерживает развитый набор параметризуемых шаблонов требований. Параметры требований задаются с помощью простого графического интерфейса в виде табличных форм, которые легко осваиваются пользователями. Принципиальным недостатком подобных инструментов является функциональная ограниченность реализованных шаблонов и необходимость привлечения штата программистов, владеющих необходимым стеком технологий, для поддержки новых требований, в том числе, появляющихся в результате

развития нормативной базы. Другим существенным недостатком является невозможность проверить или перепроверить заданные требования с помощью альтернативных инструментов. Отсутствие спецификаций требований, нейтральных по отношению к платформам, языкам и системам реализации, усложняет процедуру верификации в тех случаях, когда заказчик, проектная организация и орган государственной экспертизы используют разные программные инструменты.

Третью категорию составляют инструменты, использующие полу-структурированные описания нормативных требований на машиночитаемых диалектах языка разметки XML: RuleML, LRML, SWRL, RIF, CLIF, LDML, PMML и им подобных [15-16]. Сами требования описываются в терминах строительных тезаурусов, таксономий, классификационных систем без уточнения информационных схем, лежащих в основе проверяемых ЦИМ. По этой причине для верификации ЦИМ требуется реализация промежуточного слоя, который бы транслировал исходные описания требований в спецификации, допускающие интерпретацию непосредственно в терминах целевой информационной схемы ЦИМ. Это не только приводит к дополнительным сложностям при реализации, но и создает риски неоднозначной трактовки требований и недостоверности результатов, аналогичные тем, которые возникают из-за несоответствия метамоделей. Примерами подобных инструментов являются ACABIM Framework [17] и NSR Specification [18], предполагающие использование сторонних трансляторов требований и автономных средств верификации ЦИМ, составляющих описанную выше вторую категорию инструментов.

К четвертой категории следует отнести инструменты, предназначенные для представления требований к ЦИМ на основе открытого стандарта IDS (Information Delivery Specification) [19-20]. Важными достоинствами IDS являются его непосредственная привязка к общепринятой информационной схеме, определяемой стандартом IFC, и ориентация в большей степени на конечных пользователей, знакомых с концепцией IFC, чем на IT специалистов. Вместе с тем, IDS обеспечивает математическую строгость спецификации и непротиворечивость результатов верификации требований. В работе [5] показывается, что IDS требования представимы формальными правилами языка EXPRESS при соответствующем расширении стандартной схемы IFC, и это позволяет отнести IDS инструменты также и к категории средств формальной верификации.

Благодаря данным достоинствам IDS получил широкое распространение в строительной отрасли. Число приложений, поддерживающих стандарт, стремительно растет и составляет в настоящий момент 57 программ согласно официальному списку buildingSMART [21]. Среди свободно доступных для российского ТИМ-сообщества стоит упомянуть Sortdesk IFC Viewer [22], LastBIM IDS Editor [23], а также web-сервисы в составе Национальной ТИМ платформы [24], разработанные в ИСП РАН. Однако текущие возможности IDS ограничены проверками полноты объектного и атрибутного состава ЦИМ, корректности назначенных свойств, классификаторов, материалов, а также наличия отношений композиции («целое-часть») между объектами. К сожалению, с помощью данного стандарта невозможно специфицировать нормативные требования и своды правил, выражаемые сложными алгебраическими условиями. В силу данных причин приложения BIMcollab Zoom [25], Pilot-BIM [26], BIMIT [27], 7D Modeler [28] сочетают поддержку IDS с собственными средствами проверки ЦИМ на наличие пространственных коллизий и некоторых других конфликтов. Однако это приводит к вышеупомянутым недостаткам, свойственным инструментам второй категории.

Чтобы преодолеть имеющиеся ограничения IDS и поддержать более полную и всестороннюю работу со сложными нормативными требованиями, выражаемыми параметрическими, геометрическими, топологическими, пространственными, темпоральными и иными алгебраическими условиями, в ИСП РАН предложен и разработан машинно-интерпретируемый формат IDS++. Формат обеспечивает преемственность концепции оригинального стандарта и сохраняет все его достоинства, включая гармонизацию со схемой

IFC, ориентацию на конечного пользователя и применение формальных методов. Формат существенно расширяет возможности спецификации и верификации сложных требований, недоступные при использовании стандарта IDS, при этом гарантирует полную обратную совместимость с ним.

В настоящей статье описываются базовые принципы и ключевые возможности нового формата, а также рассматриваются разработанные программные инструменты для спецификации требований и верификации ЦИМ, применяемых в строительной отрасли. Статья структурирована следующим образом. Во втором разделе дается краткий обзор актуальной версии стандарта IDS. В третьем разделе обсуждаются базовые принципы формата IDS++, а также рассматриваются его организация и новые конструкции. Для сравнения и иллюстрации форматов приводятся их концептуальные схемы, представленные в виде диаграмм классов языка UML. В четвертом разделе подробно описываются расширенный набор фасетов и связанные с ним новые возможности IDS++. Разработанные программные инструменты для работы с IDS++ представлены в пятом разделе. В заключении определяются перспективы внедрения нового формата в практику строительных проектов и необходимые для этого усилия.

2. Организация формата IDS

IDS является открытым стандартом, предназначенным для представления требований к ЦИМ, специфицируемым в соответствии со стандартом IFC, в машинно-интерпретируемом формате для их дальнейшей верификации с использованием специализированного ПО, поддерживающего данные стандарты. Концептуальная схема формата IDS представлена на рис. 1 в виде диаграммы классов UML. Файл IDS является XML документом, включающим заголовок и набор спецификаций требований. Каждая спецификация (*Specification*) включает общую информацию, контекст применимости (*Applicability*) и условия требования (*Requirement*). В общей информации указывается название, список версий схем IFC, на которые распространяется данное требование, идентификатор, назначение, обязательный, рекомендуемый или запретительный статус требования, а также текстовые инструкции, которые могут содержать список ответственных за соблюдение данного требования, способы достижения его выполнимости и другие подобные сведения.

Условия контекста определяют множество объектов модели, которые участвуют в последующей проверке условий требования. В случае задания нескольких условий в контексте или требовании неявно предполагается, что они связаны логическим оператором AND. Для обязательного (*Required*) требования в модели должен существовать хотя бы один объект, удовлетворяющий условиям контекста. Для рекомендуемого (*Optional*) требования объекты, удовлетворяющие условиям контекста, могут отсутствовать в модели, но, если они найдены, то должны удовлетворять всем условиям требования. Для запретительного (*Prohibited*) требования объекты, удовлетворяющие условиям контекста, должны отсутствовать в модели, при этом условия самого требования, если заданы, игнорируются и не проверяются.

Для каждого из условий требования отдельно устанавливается обязательный, рекомендуемый или запретительный статус. В первом случае данное условие должно выполняться, во втором – может выполняться, а в третьем – не должно выполняться. Условия как в требовании, так и в контексте могут задаваться с использованием шести предопределенных фасетов (*Facet*). Рассмотрим назначение каждого фасета и его связь с соответствующими понятиями схемы данных IFC.

Фасет «Класс» (*Entity*) предназначен для проверки наличия объектов указанного типа в модели. Обязательный параметр «name» задает имя типа сущности (класса), определенного в схеме IFC. Необязательный параметр «predefinedType» позволяет уточнить тип объекта для

сущностей с атрибутом перечисляемого типа *PredefinedType*, предназначенным в IFC для определения новых типов без необходимости создания сущностей в схеме.

Фасет «Целое-часть» (*PartOf*) предназначен для проверки отношений композиции между объектами. Обязательный параметр «entity.name» устанавливает имя требуемого типа сущности для объекта-целого, который может быть уточнен с помощью необязательного параметра «entity.predefinedType» подобно фасету «Класс». Проверяется как прямое, так и не прямое (рекурсивное) вхождение объекта-части в объект-целое указанного типа. Необязательный параметр «relation» позволяет уточнить тип отношения композиции, который в IFC может классифицироваться как агрегация (*IfcRelAggregates*), группировка (*IfcRelAssignsToGroup*), вложение (*IfcRelNests*), принадлежность пространственной структуре (*IfcRelContainedInSpatialStructure*), высвобождение (*IfcRelVoidsElement*), заполнение (*IfcRelFillsElement*). При отсутствии значения параметра проверяются все шесть типов отношений композиции.

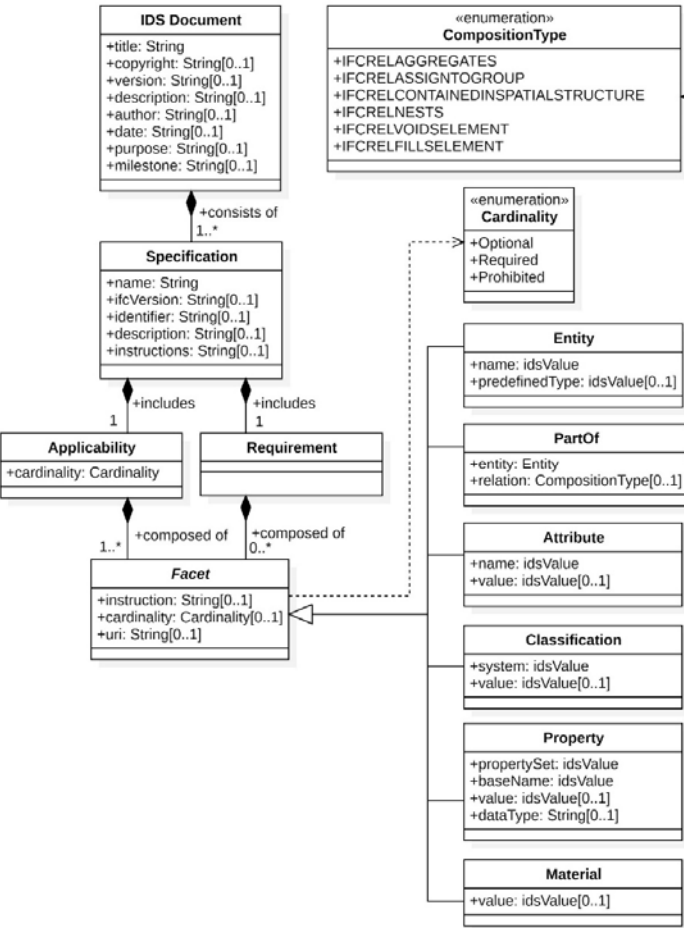


Рис. 1. Концептуальная схема формата IDS в виде диаграммы классов UML.
Fig. 1. Conceptual schema of the IDS format as a UML class diagram.

Фасет «Атрибут» (*Attribute*) предназначен для проверки корректности значения заданного атрибута сущности. Обязательный параметр «*name*» указывает имя проверяемого атрибута, а необязательный параметр «*value*» – допустимое множество значений. Если параметр не определен, то проверяется, что значение соответствующего атрибута задано (*NOT NULL*). Отметим, что большинство атрибутов сущностей в схеме IFC определены как опциональные, поэтому проверка наличия заданных значений является достаточно распространенной. Допустимое множество числовых значений может задаваться в виде одиночного значения, набора значений, интервала или полуинтервала. Для атрибутов-имен в качестве допустимых значений могут задаваться строка, множество строк, регулярное выражение, максимальная, минимальная или фиксированная длина строки. Для выбора способа задания допустимых множеств в IDS предусмотрен специальный тип данных *idsValue*.

Фасет «Классификатор» (*Classification*) предназначен для проверки наличия назначенных классификаторов из указанных таксономических систем. Последние используются в качестве дополнительного способа классификации объектов в IFC, причем индивидуальному объекту могут быть одновременно назначены классификаторы из нескольких систем. Фасет имеет два параметра: обязательный «*system*» и необязательный «*value*». Если значение второго параметра отсутствует, то проверяется факт наличия назначенного классификатора из заданной таксономической системы. В противном случае проверяется соответствие назначенных классификаторов, указанных в качестве допустимых.

Фасет «Свойство» (*Property*) предназначен для проверки задания дополнительных свойств у объектов и контроля соответствия их значений допустимым множествам. Стандарт IFC предусматривает возможность определения пользователями именованных и типизированных наборов свойств, что сближает его с онтологическим моделированием и существенно расширяет область приложений. Обязательные параметры «*propertySet*» и «*baseName*» задают имена набора свойств и свойства, находящегося в данном наборе. При отсутствии других параметров в фасете проверка ограничивается наличием у объекта назначенного свойства из заданного набора. Дополнительно можно проверить соответствие типа данных свойства указанному с помощью необязательного параметра «*dataType*», а также принадлежность значения свойства допустимому множеству, заданному необязательным параметром «*value*».

Фасет «Материал» (*Material*) предназначен для проверки материалов, назначенных ключевым объектам ЦИМ, которые играют роль архитектурных, конструктивных, инженерных и прочих элементов зданий и сооружений. Единственный параметр фасета «*value*» является необязательным. Если он не задан, то проверяется факт назначения произвольного материала. В противном случае назначенный материал сопоставляется по имени или категории с допустимыми значениями.

3. Концепция формата IDS++

Формат IDS++ следует рассматривать в качестве перспективного расширения стандарта IDS, которое может послужить конструктивной основой для национального и, возможно, международного стандарта по представлению строительных норм и правил в формализованном виде [29]. Концептуальная схема формата представлена на рис. 2 в виде диаграммы классов UML. Базовыми принципами при создании формата IDS++ послужили:

1. гармонизация с концепцией стандарта IDS и совместимость с актуальной версией 1.1;
2. простота задания типовых условий полноты ЦИМ, корректности и согласованности элементов модели с использованием функционально развитого набора фасетов;
3. универсальность и гибкость описания сложных требований, выражаемых произвольными алгебраическими условиями, в том числе, с использованием декларативных и императивных языков программирования;

4. согласованность спецификаций требований и отсутствие избыточности за счет множественного использования (*reuse*) ранее введенных определений;
5. наглядность спецификаций требований и возможность их полнофункционального редактирования с использованием специализированных графических инструментов;
6. документирование и иллюстрирование спецификаций требований с целью улучшения понимания пользователем;
7. математическая строгость и непротиворечивость спецификаций требований, их нейтральность по отношению к платформам и способам реализации средств верификации;
8. достоверность и прослеживаемость результатов верификации, полученных путем непосредственной интерпретации спецификаций требований.

3.1 Гармонизация и совместимость с IDS

IDS++, следуя концепции стандарта IDS и сохраняя структуру XML документа, поддерживает те же базовые понятия и типы данных, соответствующие документу (*IDSplusplus Document*), спецификации (*Specification*), контексту (*Applicability*), требованию (*Requirement*) и фасету (*Facet*). Условия в требовании и контексте могут задаваться с использованием шести предопределенных фасетов, рассмотренных выше. Ожидается, что подобная преемственность упростит освоение нового формата и ускорит переход ТИМ-сообщества на новые перспективные инструменты спецификации и верификации требований.

При этом сохраняется полная обратная совместимость с актуальной версией стандарта, т.е. любой файл IDS является корректным IDS++ файлом. С этой целью в структуре документа IDS++ предусмотрены два альтернативных представления для спецификаций требований: стандартное (*Standard Specification*) и расширенное (*Advanced Specification*). В стандартном представлении используется лишь один контекст применимости в отличие от расширенного, в котором может быть определено неограниченное количество контекстов. Необходимость использования множественных контекстов обусловлена нетривиальностью большинства нормативных требований и сводов правил в строительной отрасли, которые не могут быть формализованы иначе, как с использованием нескольких типизированных коллекций прикладных объектов. Наиболее распространенным примером подобных требований является отсутствие пространственных коллизий между элементами конструкций зданий и инженерных коммуникаций, относящимися к разным типам объектов и определяемыми, соответственно, разными контекстами.

Другим важным отличием расширенной спецификации является возможность применения предопределенных унарных (*Unary Facet*), бинарных (*Binary Facet*) и *N*-арных фасетов (*N-ary Facet*), соответствующих одноместным, двуместным и многоместным предикатам в составе сложных выражений с использованием логических операций и кванторов (*Logic*). Напомним, что в IDS предусмотрена лишь одна логическая операция *AND*, которая неявно применяется к унарным фасетам, перечисленным в контексте и требовании стандартной спецификации. Таким образом, формат IDS++ существенно расширяет возможности спецификации сложных требований с использованием произвольных логических условий и многоместных предикатов.

3.2 Развитая система фасетов

В IDS++ в дополнение к шести стандартным фасетам «Класс» (*Entity*), «Целое-часть» (*PartOf*), «Атрибут» (*Attribute*), «Классификатор» (*Classification*), «Свойство» (*Property*) и «Материал» (*Material*) добавлены новые предопределенные фасеты, которые упрощают задание типовых требований и расширяют возможности спецификации сложных требований.

Новыми унарными фасетами являются «Геометрия» (*Geometry*) и «Метрика» (*Metrics*). Вновь определенными бинарными фасетами являются «Коллизия» (*Collision*), «Расположение» (*Location*), «Отношение» (*Relationship*).

В состав формата также включены многоместные фасеты «Функция» (*Function*) и «Логика» (*Logic*). Более подробно новые фасеты рассматриваются в следующем разделе. Здесь лишь отметим, что в расширенной спецификации новые фасеты могут комбинироваться со стандартными в составе логических выражений, представляемых фасетом *Logic*. При этом реализация некоторых фасетов предусматривает одновременное их использование в качестве одноместных, двуместных и многоместных предикатов. В частности, фасет *Logic* одновременно наследуется от соответствующих абстракций *Unary Facet*, *N-ary Facet*.

3.3 Задание сложных алгебраических условий

Напомним, что концепция стандарта IDS предполагает использование предопределенных фасетов, которые позволяют пользователям, не имеющим надлежащей ИТ квалификации, успешно справляться с задачами подготовки типовых требований к ЦИМ. И хотя в разработанном формате набор фасетов существенно расширен и обобщен, этого недостаточно, чтобы специфицировать требования со сложными параметрическими зависимостями между элементами ЦИМ. Наиболее естественным способом решения данной проблемы является поддержка текстовых средств спецификации на одном из декларативных или императивных языков программирования.

Поскольку информационная схема IFC специфицирована на языке EXPRESS, то предпочтительна спецификация требований на этом же функционально развитом и алгебраически полном языке моделирования. Технически это могло бы быть воплощено путем до-определения схемы IFC локальными и глобальными правилами, предусматриваемыми EXPRESS. Однако, следуя концепции IDS, текстовые спецификации условий целесообразно также представлять в виде фасетов. Для этой цели в IDS++ предусмотрен фасет «Функция» (*Function*). Концепцией допускается использование альтернативных декларативных и императивных языков для описания функций.

3.4 Повторное использование определений

Одним из заметных недостатков стандарта IDS является необходимость повторного задания контекстов в тех случаях, когда нормативные требования формулируются в терминах объектов одной и той же категории. Это приводит не только к дополнительным усилиям, но и порождает риски расходящихся определений для семантически эквивалентных контекстов и их многозначной интерпретации в разных спецификациях.

Чтобы разрешить эту проблему, в формате IDS++ предусмотрена возможность определения внешних контекстов (*External Applicability*) в дополнение к стандартным внутренним контекстам (*Internal Applicability*), определяемым в спецификациях требований. Внешние контексты, как и спецификации требований, являются самостоятельными конструкциями в структуре документа IDS++. Предполагается, что корректный документ должен содержать хотя бы один внешний контекст или хотя бы одну спецификацию требования. Напомним, что корректный документ IDS содержит хотя бы одну спецификацию требования.

Внешние и внутренние контексты имеют единый набор атрибутов с тем исключением, что идентификатор внешнего контекста является обязательным атрибутом и его значение должно быть уникальным внутри документа. При определении внешних и внутренних контекстов доступны все классы унарных фасетов, предусматриваемые форматом IDS++. Используя идентификатор, можно сослаться на определение внешнего контекста из другого внешнего или любого внутреннего контекста. Более того, допустимы ссылки из контекстов, определяемых и используемых в других документах. Для того, чтобы избежать рекурсии в определении контекстов, при задании ссылок контролируется отсутствие циклов. Тем самым,

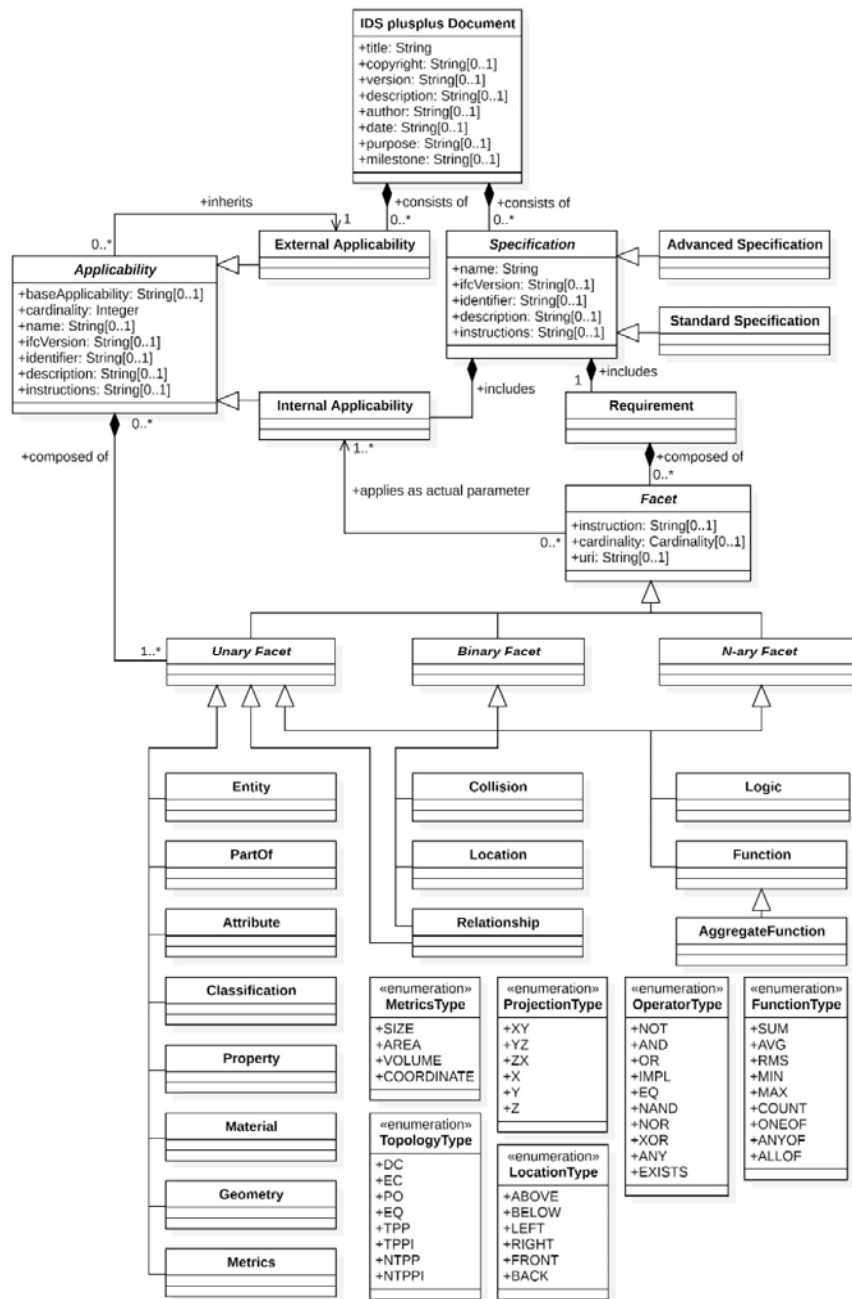


Рис. 2. Концептуальная схема формата IDS++ в виде диаграммы классов UML.
Fig. 2. Conceptual schema of the IDS format as a UML class diagram.

предусматриваются возможности организации реестров требований с определением в отдельных документах внешних контекстов как общих разделяемых ресурсов.

К связанным ссылками контекстам применяется политика обобщения и специализации, аналогичная механизму наследования классов в объектно-ориентированном программировании. Дочерний контекст, содержащий ссылку на родительский, наследует все его фасеты и связанные с ним логические условия, а также может дополнить этот набор собственными с неявным применением к ним логической операции *AND*. В этом случае дочерний контекст уточняет условия родительского контекста и, тем самым, определяет подкатегорию его объектов. Описанная политика обобщения и специализации контекстов может быть конструктивно применена для моделирования и представления строительных онтологий, таксономий и систем классификаторов непосредственно внутри документа IDS++. Таким образом, определения внешних контекстов могут быть повторно использованы при спецификации требований, снижая затраты на их подготовку и уменьшая риски ошибок.

3.5 Полнофункциональное редактирование спецификаций

Поскольку спецификации требований представляются в виде полу-структурированных XML документов, концепция стандарта IDS предусматривает наличие специализированных приложений с развитым графическим интерфейсом пользователя (GUI) для их подготовки. Число подобных приложений, называемых IDS редакторами, на сегодняшний день значительно и продолжает увеличиваться в связи с растущей популярностью стандарта и его широким применением в практике строительных проектов. IDS++ следует этому же подходу. Разработанный web-редактор в составе Национальной ТИМ платформы [24] является полнофункциональным инструментом подготовки спецификаций IDS++, который находит применение в рамках пилотных проектов и одновременно является средством апробации предложенного формата и его обоснования. Основные идеи и принципы организации оконного интерфейса редактора обсуждаются в пятом разделе статьи, а также могут быть найдены в работе [30].

3.6 Наглядность спецификаций, документирование и иллюстрирование

Одним из ключевых требований, предъявляемых как к формату представления требований, так и к поддерживающим его приложениям, является наглядность и прозрачность спецификаций для понимания пользователем. С этой целью в формате IDS++ предусмотрены возможности сопровождения спецификаций комплементарными документами с пояснениями и иллюстрациями требований, а также с описаниями примененных способов их формализации. Другой возможностью, полезной для просмотра подготовленных спецификаций, является их представление в виде текстовой нотации. Данная возможность, в частности, реализована в указанном выше web-редакторе.

3.7 Математическая строгость спецификаций

Математическая строгость описания требований в формате IDS++ определяется возможностью их представления на декларативном языке объектно-ориентированного моделирования данных EXPRESS, который обеспечивает однозначность и непротиворечивость формальных спецификаций при их интерпретации и трансляции в императивные языки программирования. В работе [5] показывается, каким образом условия стандартных фасетов IDS формализуются в виде локальных правил языка EXPRESS. Тот же подход может быть применен и к новым фасетам IDS++ с тем исключением, что фасеты, соответствующие двуместным и множественным предикатам, представимы глобальными правилами EXPRESS.

Вместе с тем, следует отметить возможную неопределенность проверки фасетов «Метрика» и «Коллизия» при использовании в ЦИМ геометрических и топологических моделей в виде сложных аналитических кривых, поверхностей, граничных представлений, конструктивных твердотельных моделей, предусматриваемых стандартом IFC. В подобных случаях вычисление метрик и определение пространственных коллизий возможны только в приближенной постановке путем тесселяции или вокселизации исходной геометрии. Поскольку применяемые вычислительные методы и их программные реализации могут различаться в сложности и точности, результаты проверки данных фасетов следует интерпретировать с учетом этих факторов, оставляя пользователю возможность вынесения окончательного вердикта о соответствии требованию.

3.8 Достоверность и прослеживаемость результатов верификации

Поскольку требования в формате IDS++ специфицируются в терминах общепринятой стандартной информационной схемы IFC, а процедура верификации реализуется путем прямой интерпретации спецификаций применительно к объектам ЦИМ, определяемым данной схемой, обеспечивается достоверность результатов верификации и их прослеживаемость на основе журналов замечаний. В подобных журналах фиксируются выявленные нарушения вместе с идентификаторами спецификаций и объектов, в которых они были обнаружены. Организация журналов в соответствии с международным стандартом BCF (BIM Collaboration Format) [31] позволяет использовать большое число существующих инструментов для их ведения [32].

Пример спецификации нормативного требования в формате IDS++ приведен на рис. 3 в виде диаграммы объектов UML.

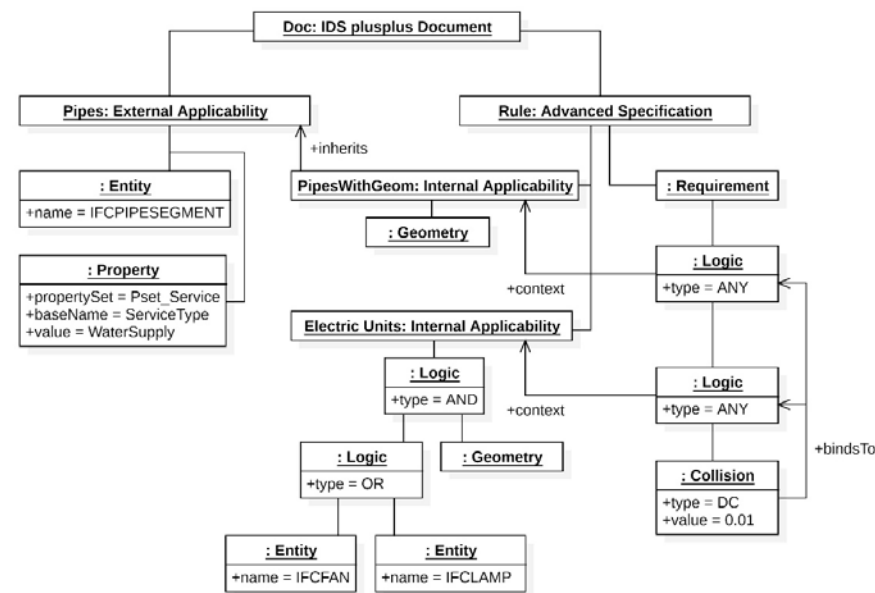


Рис. 3. Пример представления нормативного требования в формате IDS++ с помощью диаграммы объектов UML.

Fig. 3. An example of representing a regulatory requirement in IDS++ format using a UML object diagram.

Требование устанавливает правило пространственного размещения электрического оборудования, такого как вентиляторы и лампы, на требуемом удалении от систем водоснабжения. Документ *Doc* класса *IDSplusplus Document* содержит определение внешнего контекста *Pipes* класса *External Applicability* и расширенной спецификации требования *Rule* соответствующего класса *Advanced Specification*. В контексте *Pipes* с помощью стандартных фасетов «*Entity*» и «*Property*» уточняется класс объектов *IfcPipeSegment* и их свойство *WaterSupply*, которые позволяют отнести их к данному контексту.

Спецификация требования *Rule* использует два внутренних контекста *PipesWithGeom* и *Electric Units* соответствующего класса *Internal Applicability* и единственное требование класса *Requirement*. Заметим, что внутренний контекст *PipesWithGeom* является специализацией внешнего контекста *Pipes*, наследуя все его фасеты.

Для объектов контекстов с помощью фасета «*Geometry*» задаются условия наличия геометрических моделей, которые позволяют оценить или вычислить пространственное удаление объектов друг от друга. С этой целью применяется фасет «*Collision*», в качестве параметров которого указывается тип топологического отношения между объектами и минимальный зазор между ними.

Наконец, при определении внутреннего контекста *Electric Units* и требования *Requirement* применяется фасет «*Logic*». В первом случае он позволяет связать логическими операторами *AND* и *OR* условия фасетов «*Entity*», задающих классы объектов *IfcFan* и *IfcLamp*, и фасета «*Geometry*», определяющего наличие у них геометрии. Во втором случае фасет «*Logic*» позволяет представить логическую формулу с двумя кванторами *ANY* для пространственной проверки всех паросочетаний объектов контекстов. Более подробно назначение и параметры используемых в примере фасетов описываются в следующем разделе.

4. Система фасетов IDS++

Формат IDS++ обеспечивает новые возможности спецификации сложных требований в результате поддержки стандартных фасетов IDS и добавления новых, которые все вместе образуют единую классификацию и подразделяются на классы унарных, бинарных и N-арных фасетов (см. рис. 2). Рассмотрим назначение и особенности новых фасетов более подробно.

4.1 Фасет «Геометрия»

Стандартом IFC предусматривается возможность задания для элементов ЦИМ пространственно-двумерных и трехмерных моделей, объединенных общими понятиями «Представление формы», «Топологическое представление» и определяемых соответствующими классами *IfcShapeRepresentation*, *IfcTopologyRepresentation*. При этом для одного и того же элемента допустимо задание нескольких геометрических и топологических моделей, которые могут использоваться для вычисления метрических характеристик, оценки объемов работ, определения пространственных коллизий, визуализации и решения многих других прикладных задач. Однако с помощью IDS невозможно проверить ни факт наличия, ни состав назначенных моделей, что ограничивает пользователей в определении полноты ЦИМ и в понимании уровня интероперабельности применяемого ПО.

Чтобы осуществить подобные проверки, в IDS++ добавлен унарный фасет «*Геометрия*» (*Geometry*), в котором могут быть заданы следующие параметры:

- название модели, например, *BODY* для 3D форм, *FOOTPRINT* для ландшафтных 2D видов, *AXIS* для систем координат;
- название вида модели, например, *SWEPTSOLID* for 3D протяжек, *BREP* для граничного представления, *CSG* для конструктивной твердотельной геометрии,

TESSELATION для тесселяции;

- названия конкретных классов, используемых для представления геометрических и топологических моделей, например, *IfcExtrudedAreaSolid*, *IfcBlock*.

Полный список допустимых строковых значений для первых двух параметров можно найти в разделах документации стандарта IFC, относящихся к иерархии наследуемых классов *IfcRepresentation*, а названия классов – к иерархии *IfcRepresentationItem*. Фасетом предусматриваются альтернативные способы проверки моделей, назначенных элементам ЦИМ, в соответствии с приведенными параметрами. Проверки проводятся в указанном порядке перечисления параметров вплоть до вынесения первого вердикта о соответствии или несоответствии назначенных моделей. При отсутствии параметров проверяется факт наличия у элементов любых моделей. Для запретительных требований с соответствующей кардинальностью проверяется их отсутствие.

4.2 Фасет «Метрика»

Значительная часть нормативных требований связана с условиями на размеры, площади и объемы пространств в сооружаемых ОКС. Развитые BIM-приложения, как правило, имеют встроенные средства для вычисления подобных метрических характеристик и их записи в экспортируемые модели IFC в виде свойств (*IfcProperty*) и физических величин (*IfcPhysicalQuantity*) соответствующих элементов ЦИМ. В подобных случаях простые проверки могут быть реализованы с использованием стандартного фасета «*Свойство*». Однако на практике возникает необходимость прямого вычисления метрических характеристик на основе исходных геометрических данных и проверки связанных с ними условий. Причем условия могут охватывать как отдельные элементы ЦИМ, так и их коллекции с применением для них суммарных, средних, минимальных, максимальных и иных оценок.

Унарный фасет «*Метрика*» (*Metrics*) определяет следующие параметры для задания метрических условий:

- вид характеристики, устанавливаемый одним из значений *SIZE*, *AREA*, *VOLUME*, *COORDINATE*;
- опция выбора локальной системы координат элемента или глобальной системы координат ЦИМ для расчета характеристики;
- опция применения ограничивающего параллелепипеда (axis aligned box) вместо исходной геометрии для упрощенных вычислений;
- для *SIZE* может быть указана одна из осей *X*, *Y*, *Z*, для которой задается размер. В качестве альтернативы может применяться минимальное, максимальное, среднее или среднеквадратичное значение (по умолчанию, при отсутствии заданных параметров применяется среднеквадратичное значение как габарит элемента);
- для *AREA* может быть указана одна из плоскостей *XY*, *YZ*, *XZ*, на которую проецируется геометрическая модель элемента для расчета площади (при отсутствии заданной плоскости рассчитывается площадь поверхности элемента);
- для *COORDINATE* указывается одна из осей *X*, *Y*, *Z*, по которой определяется координата, а также признак *MIN*, *MAX*, *MIDDLE* или *CENTRAL*, указывающий на выбор точки с минимальной, максимальной, среднеарифметической координатой или координатой, рассчитанной в качестве центра масс;
- допустимое множество значений характеристики в виде интервала, полуинтервала или набора числовых значений.

Для задания метрических условий на коллекциях элементов фасет «*Метрика*» может быть применен в сочетании с фасетом «*Функция*», рассматриваемым далее.

4.3 Фасет «Отношение»

В стандарте IDS ограничены возможности проверки отношений между объектами, которые определяются в схеме IFC соответствующими классами, наследуемыми от *IfcRelationship*. Участие объекта в композиции или агрегации родительского объекта может быть установлено с помощью фасета «Целое-часть». Некоторые типы отношений неявно проверяются в фасетах «Свойство» (*IfcRelDefinesByProperties*), «Классификатор» (*IfcRelAssociatesClassification*) и «Материал» (*IfcRelAssociatesMaterial*).

Поскольку единый способ проверок отношений в IDS отсутствует, идея их обобщения и унификации параметров представляется естественной и логичной. Для этих целей в IDS++ добавлен новый фасет «Отношение» (*Relationship*), который может использоваться в двух вариантах. В первом варианте проверяется участие объекта (или неучастие, если требование запретительное) в отношении заданного типа в заданной роли. Во втором варианте проверяется участие пары объектов в отношении заданного типа в соответствующих ролях. Возможность многовариантного использования данного фасета отражена на рис. 2 множественным наследованием от абстракций *Unary Facet* и *Binary Facet*.

В фасете предусмотрены следующие параметры:

- название класса отношения (или классов отношений) в соответствии со схемой IFC;
- роль первого основного объекта в виде предопределенных значений *RELATING* или *RELATED*;
- кратности отношения, которые уточняют допустимое число объектов, участвующих в нем в ролях *RELATING* и *RELATED*.

В случае задания коллекций объектов в качестве фактических параметров фасета проверяется участие каждого объекта первой коллекции в заданном отношении в заданной роли, а также отсутствие связанных с ним объектов, не указанных во второй коллекции.

4.4 Фасет «Коллизия»

Фасет «Коллизия» (*Collision*) предназначен для проверки пространственных и топологических отношений между парами элементов ЦИМ. Этот вид проверок является довольно распространенным для приложений компьютерной графики и автоматизированного проектирования, с помощью которых определяются отсутствие пересечений элементов и соблюдение требуемых зазоров между ними.

В IDS++ подобные проверки рассматриваются в более общей постановке, использующей развитую систему пространственно-временных операторов [33] и включающей в себя топологические операторы RCC-8 (Region Connection Calculi) [34]. Последние определяют восемь содержательных случаев взаимного расположения пары элементов:

- не связаны – *DC (disconnected)*;
- связаны внешне – *EC (externally connected)*;
- частично перекрыты – *PO (partial overlap)*;
- равны – *EQ (equal)*;
- первая область касается второй, находясь внутри – *TPP (tangential proper part)*;
- вторая область касается первой, находясь внутри – *TPPI (tangential proper part inverse)*;
- первая область лежит внутри второй – *NTPP (non-tangential proper part)*;
- вторая область лежит внутри первой – *NTPPI (non-tangential proper part inverse)*.

Для проверки пространственно-топологических отношений и их количественной оценки фасет «Коллизия» использует следующий набор параметров:

- тип отношения в виде одного или нескольких значений из набора *DC, EC, PO, EQ, TPP, TPPI, NTPP, NTPPI*;
- для *DC, NTPP* и *NTPPI* возможно задание минимального и максимального расстояния-зазора между элементами (gap);
- для *PO* может быть дополнительно задана максимальная глубина проникновения (penetration depth);
- опция выбора системы координат для расчетов, указываемых ниже (допускается применение локальных систем первого и второго элементов или глобальной системы координат ЦИМ);
- опция применения ограничивающих параллелепипедов (axis aligned bounding box) в выбранной системе координат для упрощения вычислений;
- опция проецирования элементов на одну из главных осей *X, Y, Z* или на одну из главных плоскостей *XY, YZ, XZ* в выбранной системе координат для расчетов.

В случае задания коллекций объектов в качестве фактических параметров фасета проверяются все их паросочетания.

4.5 Фасет «Расположение»

Фасет «Расположение» (*Location*) предназначен для задания и проверки условий на взаимное расположение элементов друг относительно друга. Поскольку применение операторов ориентаций для элементов с протяженными границами крайне затруднительно, для определения их относительного расположения предполагается использовать ограничивающие параллелепипеды. Фасет использует следующий набор параметров:

- опции выбора локальной системы координат первого элемента, второго элемента или глобальной системы координат ЦИМ;
- направление расположения второго элемента относительно первого в виде одного из значений *ABOVE, BELOW, LEFT, RIGHT, FRONT, BACK*;
- допустимое смещение второго элемента относительно первого в выбранном направлении при использовании противоположных граней ограничивающих параллелепипедов (при заданном отрицательном значении параллелепипеды могут пересекаться);
- поворот второго элемента относительно первого в виде тройки углов между локальными системами координат элементов. Использование одной системы для элементов интерпретируется как отсутствие поворота.

4.6 Фасет «Функция»

Фасет «Функция» (*Function*) предназначен для спецификации условий и требований на одном из декларативных или императивных языков программирования. Фасет необходим для формального описания нетиповых требований, которые не могут быть представлены простыми логическими выражениями с предопределенными предикатами фасетов IDS или IDS++. В подобных случаях необходим более универсальный способ формализации требований с использованием развитых средств алгебраической спецификации.

Фасет имеет общую организацию, которая обеспечивает объявление логической функции и задание ее сигнатуры, определение функции в виде интерпретируемого или транслируемого кода, а также установку фактических параметров. При этом выбор языка определения функции допускается одним из параметров фасета. В текущей реализации инструментария IDS++ используется популярный в настоящее время язык Python 3, однако в будущем возможна поддержка и других языков.

При описании условия с использованием фасета «Функция» пользователь задает:

- имя функции;
- язык определения функции;
- формальные и фактические параметры функции, включая их кратность;
- тело функции с интерпретируемым или транслируемым кодом на выбранном языке.

Отметим способ задания параметров функции с помощью контекстов. Назначенный контекст неявно определяет формальный параметр объектного типа и одновременно задает объект или коллекцию объектов в качестве фактического параметра, необходимого для выполнения функции. Чтобы различать случаи задания индивидуальных объектов и их коллекций, пользователь указывает кратность, распространяемую на все параметры функции. При задании параметра указывается его имя и список методов, с помощью которых можно напрямую адресоваться к атрибутам, свойствам, классификаторам, материалам и метрическим характеристикам элементов ЦИМ непосредственно в теле функции. В зависимости от количества параметров фасет может быть отнесен к унарным, бинарным или N -арным и применен для проверок соответствующих условий.

Для экономии усилий при задании типовых условий, связанных с оценкой числовых характеристик на коллекциях объектов, предусмотрен специальный фасет «Агрегатная функция» (*Aggregate Function*), который следует рассматривать в качестве унарной специализации фасета «Функция». Фасет позволяет относительно просто задавать условия для суммарных, средних, среднеквадратичных, минимальных, максимальных и иных значений без определения тела функции и предусматривает сокращенный список параметров:

- название агрегатной функции, определяемое одним из значений *SUM (Summary)*, *AVG (Average)*, *RMS (Root Mean Square)*, *MIN (Minumum)*, *MAX (Maximum)*, *COUNT*, *ONEOF*, *ANYOF*, *ALLOF*;
- фактический параметр функции в виде заданного одиночного контекста;
- числовое поле параметра, применяемое в расчетах;
- допустимое множество значений, в том числе в виде интервала или полуинтервала.

При выборе языка Python 3 пользователю предлагается определить только тело функции, принимающей объявленные параметры и возвращающей логическое значение. Заголовок функции формируется автоматически на основании параметров фасета (имени функции и списка ее формальных параметров). Тем самым, обеспечивается согласованность сигнатуры функции с её формальным описанием в IDS++ и исключается риск синтаксических и семантических несоответствий.

4.7 Фасет «Логика»

Фасет «Логика» (*Logics*) предназначен для задания формул логики первого порядка (First-Order Logic, FOL) над конечной моделью с доменами, определяемыми объектными контекстами. Логическая формула представляется в виде ориентированного ациклического дерева, корень которого соответствует самой формуле, внутренние вершины – логическим операторам, включая кванторы и конструкторы для определения свободных и зависимых переменных, а листья – атомарным предикатам.

Сложное логическое условие в IDS++ представляется с помощью рекурсии, на каждом уровне которой определяется соответствующий оператор и вложенные фасеты любого из предусматриваемых IDS++ унарных, бинарных и N -арных типов, включая и сам фасет «Логика». В результате данного обобщения существенно расширяются возможности

стандарта IDS, допускающего композицию ограниченного набора фасетов с использованием единственного логического оператора *AND*.

Параметрами фасета являются:

- вид логического оператора, задаваемый одним из значений *NOT*, *AND*, *OR*, *NAND*, *NOR*, *XOR*, *IMPLIES*, *EQ* или значений *ANY*, *EXISTS*, применяемых для кванторов всеобщности и существования;
- определение переменной в случае задания квантора и допустимой кратности для квантора количественного существования;
- ссылки на родительский и вложенные дочерние фасеты.

Заметим, что операторы *AND*, *OR*, *NAND*, *NOR* допускают задание произвольного числа операндов-предикатов, определяемых дочерними фасетами, оператор *NOT* является унарным, а все остальные – бинарными. Отсутствие выбранного оператора по умолчанию интерпретируется как *AND* по аналогии со стандартом IDS.

При задании кванторов определяется формальная переменная, соответствующая одному из объектов контекста. Область применения переменной распространяется на все дочерние вложенные фасеты. Кроме того, допустимо определение поименованных коллекций, которые, в частности, упрощают формализацию условий с многоместными предикатами на объектах разных контекстов.

5. Программные приложения для работы с IDS++

Для контроля корректности данных в документах IDS++ и их сериализации разработана соответствующая схема XSD, которая определяет структуры, необходимые для представления всех вышеописанных конструкций и фасетов в XML файлах. В значительной степени она повторяет концептуальную схему на языке UML, поэтому в настоящей статье довольно объемная схема XSD не приводится, в том числе и по причине краткости и ясности изложения.

Разработанный формат для файлового представления требований к ЦИМ обеспечивает как их спецификацию, так и верификацию путем непосредственной интерпретации конструкций формата и исчисления предикатов, задаваемых соответствующими фасетами. Обсудим два программных приложения, разработанных для этих целей.

Первым приложением является полнофункциональный web-редактор, который реализует просмотр и редактирование спецификаций требований в формате IDS++ с поддержкой всех вышеперечисленных новых возможностей [24, 30]. Снимок главного окна web-редактора представлен на рис. 4а. В верхней части окна отображается заголовок документа, в левой нижней части – списки контекстов и спецификаций требований, справа от них – компактные текстовые нотации выбранного контекста и требования. Графический интерфейс пользователя обеспечивает как просмотр соответствующих данных документа IDS++, так и их редактирование. В частности, в главном окне предусмотрены команды для добавления нового контекста или спецификации требования, а также для открытия специальных окон для их просмотра и редактирования.

В окне редактирования контекста, показанном на рис. 4б, отображается родительский контекст, если необходим для определения текущего, а также список фасетов. В последнем выводятся родительские фасеты в порядке их наследования, а также фасеты текущего контекста. Собственные фасеты контекста допускают непосредственное редактирование в отличие от унаследованных, для изменения которых необходимо перейти к родительским контекстам. Для информирования пользователя о невозможности редактирования в текущем контексте унаследованные фасеты отображаются полупрозрачными.

В окне редактирования спецификации требования, пример снимка которого приведен на рис. 4с, отображаются внутренние контексты с меню для добавления новых контекстов, удаления

и редактирования существующих. Для редактирования требования возможен переход к соответствующему окну, подобному окну редактирования контекста с тем исключением, что при определении фасетов могут использоваться разные контексты спецификации, а при определении фасетов «Логика» – применяться кванторы и конструкторы переменных. Рис. 4d демонстрирует пример специального окна, предназначенного для редактирования фасета «Функция». В нем задается имя и сигнатура функции в виде списка параметров и методов доступа, а также определяется ее тело на выбранном языке программирования.



Рис. 4. Снимки редактора IDS++: главное окно приложения (а), окно контекста (b), окно спецификации требования (с) и окно фасета «Функция» (d).

Fig. 4. Screenshots of the IDS++ editor: the main application window (a), the context window (b), the requirement specification window (c), and the “Function” facet window (d).

Вторым разработанным приложением является web-сервис для верификации и валидации ЦИМ на соответствие требованиям в формате IDS++. В настоящее время в дополнение к стандартным конструкциям и фасетам IDS поддерживается работа с внешними контекстами на основе политики наследования, обеспечивается возможность интерпретации требований с множественными контекстами, а также реализованы проверки условий, определяемых новыми фасетами «Геометрия», «Отношение», «Коллизия» и «Метрика» с ключевыми опциями. Снимок экрана сервиса представлен на рис. 5а. Проверка ЦИМ и каталог требований загружаются в виде файлов IFC и IDS++. Выявленные ошибки публикуются в журнале и отправляются пользователю в виде документа в формате HTML или BCF для дальнейшего анализа с использованием соответствующих графических инструментов. Пример HTML отчета о выявленных ошибках также приведен внизу рис. 5а.

На рис. 5b приведен снимок экрана web-сервиса для управления замечаниями в составе Национальной ТИМ платформы [24] с загруженным BCF журналом. В дополнение к типовым функциям отображения и редактирования замечаний в виде таблиц сервис реализует развитые графические средства для просмотра замечаний в 3D окне с интерактивными возможностями визуализации ЦИМ, навигации, выбора ракурсов, фильтрации по типам элементов, отсечений по главным плоскостям, измерения расстояний, углов и площадей.

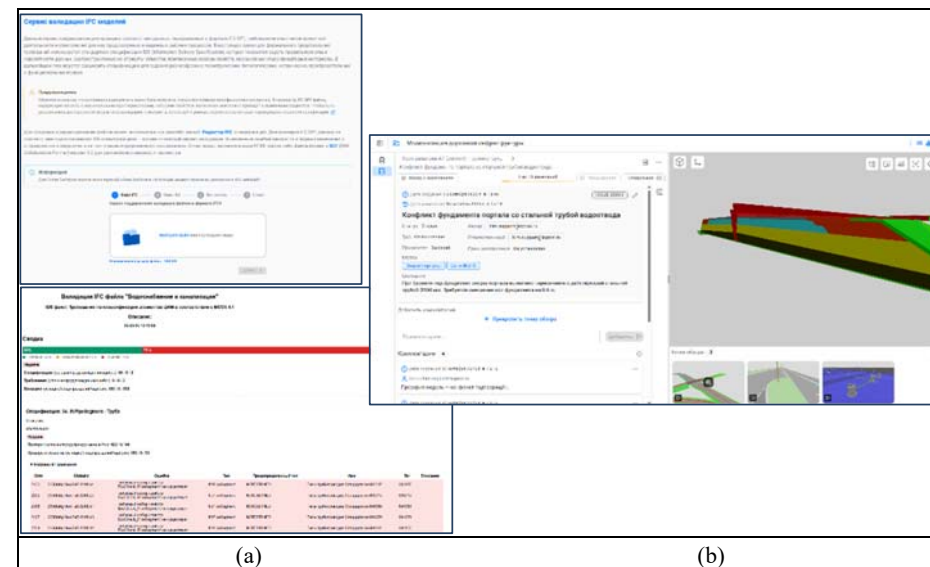


Рис. 5. Снимки web-сервисов в составе Национальной ТИМ платформы: (а) сервис верификации и валидации ЦИМ вместе с HTML отчетом об ошибках, (b) сервис управления замечаниями.

Fig. 5. Screenshots of web services within the National BIM platform: (a) the BIM verification and validation service together with an HTML error report, (b) and the issue management service.

6. Заключение

Таким образом, представлена концепция и основные конструкции нового машинно-интерпретируемого формата IDS++ для спецификации и верификации требований в строительной отрасли. Благодаря проведенным обобщениям и развитой системе фасетов формат IDS++ существенно расширяет возможности стандарта IDS при работе с нормативными требованиями, выражаемыми геометрическими, топологическими, пространственными, параметрическими и иными алгебраическими условиями. В дополнение к предопределенным фасетам, реализующим предикаты для проверки типовых условий, в IDS++ предусмотрены возможности для непосредственной спецификации сложных требований на декларативных и императивных языках. При этом важное внимание уделено повторному использованию определений, сокращающему затраты на подготовку спецификаций и снижающему риски ошибок.

Обеспечивая полную обратную совместимость с актуальной версией стандарта IDS и сохраняя преемственность с его базовыми понятиями, IDS++ допускает естественный переход с существующих, функционально ограниченных инструментов верификации ЦИМ в строительной отрасли на более общие и перспективные. В то же время важно подчеркнуть

значимость проведения пилотных проектов, направленных на подготовку примеров спецификаций сложных нормативных требований и сводов правил на основе нового формата. Нарботка базы таких практических примеров и их публикация позволили бы апробировать новый формат и способствовать его популяризации. В случае достижения консенсуса и поддержки широким индустриальным сообществом существует перспектива согласования формата и его принятия в качестве национального или международного информационного стандарта.

Список литературы / References

- [1]. Brambilla M., Cabot J., Wimmer M. Model-driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. Morgan & Claypool, 2012. 166 p.
- [2]. Arayici Y., Fernando T., Munoz V., Bassanino M. Interoperability specification development for integrated BIM use in performance based design. *Automation in Construction*. 2018, vol. 85, pp. 167-181. DOI: 10.1016/j.autcon.2017.10.018.
- [3]. Semenov V., Arishin S., Semenov G. Towards conceptual interoperability of BIM applications: Transaction management versus data exchange. *ECPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling*. 2021. CRC Press, London, pp. 51-58. DOI: 10.1201/9781003191476-7.
- [4]. Кулямин В.В. Методы верификации программного обеспечения. Конкурс обзорно-аналитических статей по направлению «Информационно-телекоммуникационные системы». 2008. 111 с. Доступно по ссылке: https://www.ispras.ru/publications/2008/methods_of_software_verification, обращение 05.03.2026 / Kulyamin V.V. Software Verification Methods. Competition of review and analytical articles on Information and Telecommunication Systems. 2008. 111 p. Available at: https://www.ispras.ru/publications/2008/methods_of_software_verification, accessed March, 05, 2026 (in Russian).
- [5]. Semenov V.A., Morozov S.V., Arishin S.V., Kuzina O.N., Rimshin V.I., Makisha E.V. Formal specification and verification of requirements in architecture and construction using the EXPRESS modeling language. *Programming and Computer Software*, 2024, vol. 50, no. 5, pp. 376-391. DOI: 10.1134/S036176882470018X.
- [6]. 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.
- [7]. 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, pp. 81-126. DOI: 10.1007/978-3-319-92862-3_5.
- [8]. 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.
- [9]. ГОСТ Р 10.0.02-2019/ISO 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. / ГОСТ R 10.0.02-2019/ISO 16739-1:2018. Building information modeling standards system. Industry core classes (IFC) for the exchange and management of construction data. Part 1. Data schema. 28 p. (in Russian).
- [10]. ГОСТ Р ИСО 10303-11-2009. Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS. 202 с. / ГОСТ R ISO 10303-11-2009. Industrial automation systems and their integration. Representation and exchange of product data. Part 11. Description methods. EXPRESS language reference guide. 202 p. (in Russian).
- [11]. González C.A., Cabot J. Formal verification of static software models in MDE: A systematic review. *Information and Software Technology*, 2014, vol. 56, no. 8, pp. 821-838. DOI: 10.1016/j.infsof.2014.03.003.
- [12]. Somogyi N., Mezei G. Generalized Formal Model-Verifier: A Formal Approach for Verifying Static Models. *Springer Nature Computer Science*, 2024, vol. 5, no. 508. DOI: 10.1007/s42979-024-02808-2.
- [13]. EXPRESS Data Manager™ – Jotne Connect. Available at: <https://jotneit.com/products/express-data-manager>, accessed 05.03.2026.

- [14]. Soliman-Junior J., Tzortzopoulos P., Baldauf J.P., Pedo B., Kagioglou M., Formoso C.T., Humphreys J. Automated compliance checking in healthcare building design. *Automation in Construction*, 2021, vol. 129. DOI: 10.1016/j.autcon.2021.103822.
- [15]. Dimyadi J., Governatori G., Amor R. Evaluating LegalDocML and LegalRuleML as a Standard for Sharing Normative Information in the AEC/FM Domain. *Proceedings of Lean & Computing in Construction Congress (LC3)*, 2017, vol. 1 (CIB W78), pp. 637-644. DOI: 10.24928/LC3-2017/0012.
- [16]. Макиша Е.В. Анализ методов перевода требований нормативно-технической документации в машиночитаемый формат для проверки информационных моделей строительных объектов. *Моделирование, оптимизация и информационные технологии*, 2019, т. 7, № 2(25), стр. 339-350. DOI: 10.26102/2310-6018/2019.25.2.028. / Makisha E.V. Analysis of methods of regulation requirements translation into machine-readable format for verification of building information models. Modeling, optimization, and information technologies, 2019, vol. 7, no. 2(25), pp. 339-350. DOI: 10.26102/2310-6018/2019.25.2.028 (in Russian).
- [17]. Li B., Schultz C., Dimyadi J., Amor R. Defeasible reasoning for automated building code compliance checking. *ECPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling*, 2021. CRC Press, London, pp. 229-236. DOI: 10.1201/9781003191476-32.
- [18]. Кутузова О. Через тернии к ... автоматической экспертизе цифровой информационной модели. *CAD master*. 2023. Т. 101. № 2. С. 104–111. / Kutuzova O. Through thorns to... automatic expertise of the building information model. *CAD master*. 2023, vol. 101, no. 2, pp. 104-111 (in Russian).
- [19]. IDS – Machine readable Information Delivery Specification. Available at: <https://github.com/buildingSMART/IDS/blob/master/Documentation>, accessed 05.03.2026.
- [20]. Tomczak A., Benghi C., van Berlo L., Hjelseth E. Requiring Circularity Data in BIM with Information Delivery Specification. *Journal of Circular Economy*, 2024, vol. 1, no. 2, pp. 1-13. DOI: 10.55845/REJY5239.
- [21]. IDS Software Implementations – buildingSMART Technical. Available at: <https://technical.buildingsmart.org/ids-software-implementations>, accessed 05.03.2026.
- [22]. Sortdesk | Free Online IFC Viewer. Available at: <https://viewer.sortdesk.com>, accessed 05.03.2026.
- [23]. IDS Editor | A quick and easy way to create IDS files. Available at: <https://ids-editor.com>, accessed 05.03.2026.
- [24]. Национальная ТИМ Платформа / National BIM Platform. Available at: <https://bim.ispras.ru/main>, accessed 05.03.2026.
- [25]. Model Checking & Clash Detection with BIMcollab Zoom. Available at: <https://www.bimcollab.com/en/products/bimcollab-zoom>, accessed 05.03.2026.
- [26]. Pilot-BIM. Available at: <https://ascon.ru/products/pilot-bim>, accessed 05.03.2026.
- [27]. BIMIT. Available at: <https://twin.bimit.ru>, accessed 05.03.2026.
- [28]. 7D Modeler. Available at: <https://www.openbimsystems.ru/7d-modeler>, accessed 05.03.2026.
- [29]. Семенов В.А., Аришин С.В., Зорин Е.М. Перспективный российский формат IDS++ для машинно-интерпретируемых требований в архитектуре и строительстве. Информационное моделирование в задачах строительства и архитектуры: материалы VIII Международной научно-практической конференции. Под общей ред. А.А. Семенова. Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, 2025. С. 158-165. DOI: 10.23968/BIMAC.2025.023. / Semenov V.A., Arishin S.V., Zorin E.M. Promising Russian IDS++ format for computer-interpretable requirements in architecture and construction. *BIM in Construction and Architecture: Proceedings of VIII International Conference (A.A. Semenov ed.)*. Saint-Petersburg, State University of Architecture and Civil Engineering, 2025, pp. 158-165 (in Russian). DOI: 10.23968/BIMAC.2025.023.
- [30]. Семенов В.А., Зорин Е.М., Шило А.В. Web-редактор машинно-интерпретируемых требований в формате IDS++. Информационное моделирование в задачах строительства и архитектуры: Материалы IX Международной научно-практической конференции. Под общей ред. А.А. Семенова. Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, 2026 (в печати). / Semenov V.A., Zorin E.M., Shilo A.V. Web-editor for computer-interpretable requirements in the IDS++ format. *BIM in Construction and Architecture: Proceedings of IX International Conference (A.A. Semenov ed.)*. Saint-Petersburg, State University of Architecture and Civil Engineering, 2026, in print (in Russian).
- [31]. BIM Collaboration Format v3.0 Technical Documentation. Available at: https://github.com/buildingSMART/BCF-XML/tree/release_3_0/Documentation, accessed 05.03.2026.

- [32]. BCF Software Implementations – buildingSMART Technical. Available at: <https://technical.buildingsmart.org/bcf-software-implementations>, accessed 05.03.2026.
- [33]. Петрищев К.С., Золотов В.А., Семенов В.А. Система операторов для пространственно-временного анализа динамических сцен. Труды Института системного программирования РАН. 2018. Т. 30. № 6. С. 237–258. DOI: 10.15514/ISPRAS-2018-30(6)-13. / Petrishchev K.S., Zolotov V.A., Semenov V.A. A system of operators for spatial-temporal analysis of dynamic scenes. Proceedings of the Institute for System Programming of the RAS, 2018, vol. 30, no. 6, pp. 237-258 (in Russian). DOI: 10.15514/ISPRAS-2018-30(6)-13.
- [34]. Randell D.A., Cui Z., Cohn A.G. A spatial logic based on regions and connection. Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning. 1992, pp. 165-176.

Информация об авторах / Information about authors

Виталий Адольфович СЕМЕНОВ – доктор физико-математических наук, профессор, заведующий отделом системной интеграции и прикладных программных комплексов Института системного программирования им. В.П. Иванникова РАН, профессор кафедры Системного программирования МФТИ. Сфера научных интересов: модельно-ориентированные технологии программной инженерии, системная интеграция, визуализация и компьютерная графика, вычислительная геометрия, информационное моделирование в области архитектуры и строительства, проектное управление и календарно-сетевое планирование.

Vitaly Adolfovich SEMENOV – Dr. Sci. (Phys.-Math.), Prof., Head of the Department of System Integration and Multi-disciplinary Applied Systems of the Ivannikov Institute for System Programming of the RAS, professor of MIPT. Research interests: model-driven methodologies and CASE toolkits, system integration, visualization and computer graphics, computational geometry, building information modeling, project management and scheduling.

Семен Васильевич АРИШИН – научный сотрудник ИСП РАН. Научные интересы: системная интеграция, интероперабельность программного обеспечения и технологии информационного моделирования в архитектуре и строительстве.

Semen Vasilyevich ARISHIN – a researcher at ISP RAS. Research interests: system integration, software interoperability and building information modeling.

Евгений Михайлович ЗОРИН – аспирант ИСП РАН. Научные интересы: программная инженерия на основе моделей, вычислительная геометрия, управление требованиями, технологии информационного моделирования в архитектуре и строительстве.

Eugene Mikhaylovich ZORIN – a postgraduate student of ISP RAS. Research interests: model-based software engineering, computational geometry, requirement management and building information modeling.

Сергей Вячеславович МОРОЗОВ – кандидат физико-математических наук, ведущий научный сотрудник ИСП РАН, доцент кафедр Системного программирования МФТИ и ВМК МГУ. Научные интересы: программная инженерия на основе моделей, управление семантически сложными данными.

Sergey Vyacheslavovich MOROZOV – Cand. Sci. (Phys.-Math.), leading researcher at ISP RAS, associate professor of MIPT and Computer Science faculty of MSU. Research interests: model-based software engineering, management of semantically complex data.