

ПРИКЛАДНЫЕ ВОПРОСЫ И ЗАДАЧИ ПРИМЕНЕНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

УДК 658.513

DOI: 10.52190/2073-2597_2023_2_42

EDN: JUNEII

ОПЫТ ПРИМЕНЕНИЯ МАШИНОЧИТАЕМЫХ ТРЕБОВАНИЙ В ПРОЕКТНОМ НАПРАВЛЕНИИ «ПРОРЫВ»

О. В. Жданова

Акционерное общество «Прорыв», Москва, Россия

О. Н. Столяров

Акционерное общество «Атомэнергопроект», Москва, Россия

Представлен опыт разработки и применения машиночитаемых требований в реализуемом Государственной корпорацией по атомной энергии «Росатом» проекте «Прорыв». Дается пояснение о том, какую эволюцию прошли машиночитаемые требования, какие информационные технологии использовались для разработки и проверки машиночитаемых требований в последние десятилетия. В статье представлен опыт применения одного из методов в проектном направлении «Прорыв», приведены примеры требований, затрагиваются аспекты, затрудняющие распространение машиночитаемых требований как специального способа формулирования ограничений и их проверки, отражена ситуация с развитием отечественной нормативной базы в части перехода на машиночитаемые форматы.

Ключевые слова: система управления требованиями, машиночитаемые требования, алгоритм проверки требований, автоматизированная проверка требований, язык записи требований.

Современный уровень развития информационных технологий существенно расширил круг решаемых задач. Одновременно с этим возникла потребность в таком хранении и представлении информации, которое позволит максимально использовать потенциал этих технологий. В частности, аналитика больших данных, искусственный интеллект, дополненная и виртуальная реальность, моделирование и симуляторы, трёхмерная печать требуют создания информационной модели объекта - очень точной многоаспектной формы его описания. Понятие «информационная модель объекта» закреплено законодательно. В 2019 г. в Градостроительный кодекс Российской Федерации введено понятие «информационная модель объекта капитального строительства». ГОСТ Р 57700.37-2021 «Компьютерные модели и моделирование. Цифровые двойники изделий. Общие положения» определяет эволюцию «информационной модели», вводя понятие «цифровой двойник». Имея настолько подробное математическое описание

предметов, позволяющее с высокой точностью имитировать их поведение, логично сделать следующий шаг развития — автоматизированную проверку выполнения нормативных требований, предъявляемых к объектам.

Методы автоматизированной проверки требований. История вопроса

Идея автоматизированной проверки требований не является новой. В 1960 г. была опубликована статья с идеями на эту тему [1], с тех пор по мере развития информационных технологий и языков программирования предпринималось большое число попыток решить вопрос разработки машиночитаемых документов и создания системы автоматизированной проверки информационных моделей, однако ранее применялась иная терминология.

В качестве ретроспективы развития мысли приведем основные методы автоматизированной

проверки требований: метод «прямого» кодирования; метод применения специализированных высокоуровневых языков программирования; метод, построенный на применении логических языков; метод применения расширяемого языка разметки; метод применения семантического подхода, основанного на технологии *Semantic Web*.

Метод «прямого» кодирования

Смысл этого метода понятен из его названия: для каждого требования разрабатывается программный код, который выполняет проверку требования. Для этого метода может быть выбран любой язык программирования. Недостатков у метода достаточно много, но наиболее важными являются большое разнообразие форматов САПР-данных и интероперабельность на уровне терминов. Названия объектов и предметов проверки должны совпадать в нормативно-технической и в электронной документации, также должна быть обеспечена наглядность того, что текст требования соответствует программному коду, который, к тому же, может быть скомпилирован. Для решения этих недостатков была предпринята попытка разработки языка более высокого уровня.

Метод применения специализированных высокоуровневых языков программирования

Одним из специализированных высокоуровневых языков программирования является *BERA* (*Building Environment Rule and Analysis*) [2]. Язык *BERA* пытается работать с информационными моделями зданий интуитивно понятным способом, чтобы создавать и анализировать правила, которые могут применяться для проверки конфигурации строительных объектов, компоновки зданий даже на ранних стадиях проектирования. Первоначальная реализация *BERA* включала общедоступную схему *IFC*. Схема *IFC*

определена в международном и российском стандартах [3], однако язык *BERA* не стал стандартом по ряду объективных причин. Развиваемая независимо от методов проверки требований схема *IFC* стала использоваться и в других альтернативных методах.

Метод, построенный на применении логических языков

В качестве примера можно привести *Semantic natural language processing-based automated compliance checking (SNACC)* [4].

По существу, использование метода сводится к трем шагам (рис. 1).

На первом шаге с помощью программного обеспечения класса *NLP* происходит автоматическое извлечение нормативных требований из строительных норм и правил, а также других типов строительных нормативных документов, и преобразование требований в логические пункты (логические правила).

На втором шаге происходит извлечение и преобразование проектной информации. С помощью языка запросов автоматически извлекается проектная информация из модели *IFC* и происходит преобразование информации в факты, высказанные на языке логики. Информация о конструкции автоматически согласуется с нормативной информацией в силу правил семантического преобразования, закодированных в логических высказываниях.

На третьем шаге выполняется автоматический ризонинг, то есть машина логического вывода проверяет логические факты на соответствие логическими правилами и генерирует отчеты о проверке соответствия.

При кажущейся простоте метода преобразовать с помощью *NLP* текст нормативных документов оказалось чрезвычайно сложной задачей. Однако идея с разметкой документов нашла свое дальнейшее развитие.

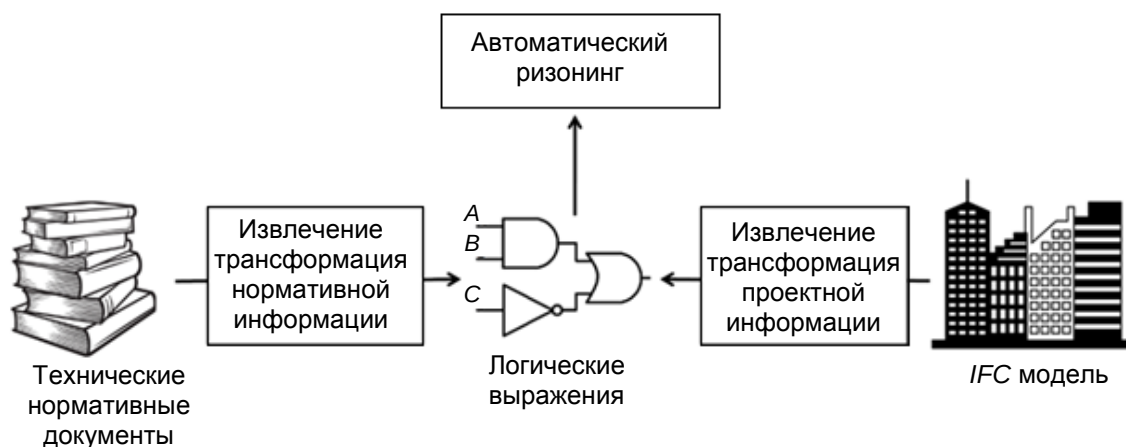


Рис. 1 Пример использования метода логических языков

Метод применения расширяемого языка разметки

Примером расширяемого языка разметки является язык *RASE*. *RASE* — это семантическая концепция преобразования нормативных документов в единое четко определенное правило, которое может быть реализовано в программном обеспечении для проверки моделей на основе *IFC*. Методология *RASE* основана на ряде правил [5]. В языке разметки *RASE* используются следующие четыре типа *RASE*-меток - «Требование» «Применяется», «Выбор» и «Исключение», которые расставляются в тексте нормативных документов в виде *XML*-тэгов. Для реализации метода осуществляются те же три шага, что и для метода, построенного на применении логических языков.

Отличие *RASE* от *SNACC* в том, что метки представляет не компьютерная программа класса *NLP*, а эксперт. Еще одной особенностью метода применения расширяемого языка разметки является то, что по итогам разметки все требования помещаются в таблицу и дополняются атрибутами.

Метод применения семантического подхода, основанного на технологии *Semantic Web*

Этот метод включает в себя идеи предыдущих методов, но его реализация основана на группе стандартов *W3C* с общим названием *Semantic Web*. Особенность технологии *Semantic Web* в том, что она использует язык дескрипционной логики и адаптированные для логической обработки способы хранения и извлечения информации, а также способы разметки *HTML* страниц и множество других вспомогательных инструментов [6].

Суть рассматриваемого метода и его главное отличие от метода, построенного на применении логических языков, в том, что первоначально формируется онтология предметной области, которая является и терминологическим словарем для разработки правил и метаданными для проектной информации. Кроме того, в отличие от *IFC*, онтология разрабатывается не ИТ-сообществом, а экспертами в предметной области - инженерами. Далее вся подлежащая проверке проектная информация извлекается из проектных баз данных и преобразовывается в высказывания на языке логики, но вместо *IFC* используется онтология. Автоматическое извлечение нормативных требований из строительных норм и правил предыдущего метода заменяется на переформулирование экспертами требований на язык логики (то, что в этом методе называется правилами). Так как вся информация хранится единым стандартным способом, то с помощью механизма логического вывода можно получать необходимые заключения о выполнении нормативных требований.

Существует несколько способов реализации технологии *Semantic Web*, некоторые представлены в работах [7, 8]. Еще один способ, примененный в проектом направлении «Прорыв», представлен ниже.

В 2020 г. аналогичные исследования в области существующих наработок, связанных с автоматизацией проверки информационной модели и в общем с подходами и методами представления требований научно-технической документации (НТД) в машиночитаемом формате, проводила организация НИЦ ЦПС. Командой НИЦ ЦПС было рассмотрено более 100 различных статей и других информационных источников. В итоге НИЦ ЦПС разработал свой способ формализации требований НТД, основанный на технологиях, которые, по его мнению, «уже сегодня можно использовать для решения конкретных практических задач» [9]. Среди них *XML*, *XQuery*, *XPath*, *SQL*, *MathML*, *SVG*. «Однако в будущем, по мнению экспертов НИЦ ЦПС, формализация требований будет двигаться в направлении семантических технологий *WEB*» [9].

Опыт применения машиночитаемых требований в проектом направлении «Прорыв»

Проанализировав представленный выше опыт применения методов автоматизированной проверки требований, в проектом направлении «Прорыв» [10] по ряду причин сделали выбор в пользу метода применения семантического подхода, основанного на технологии *Semantic Web*. Во-первых, стратегически более правильно опираться на действующие стандарты *W3C*, чем на конкретные решения. Во-вторых, этот метод самый доступный для освоения и применения. В-третьих, у метода применения семантического подхода есть доступные ИТ-решения с открытым исходным кодом (*OpenSource*). Принимая во внимание нескончаемый поток санкций в отношении России, собственное решение правильнее строить на *OpenSource* разработках. Выбор в пользу программных решений с открытым исходным кодом позволит избежать зависимости от конкретного поставщика программ и, при необходимости, осуществлять доработку собственными силами. Сегодня, когда ряд компаний с российскими «корнями» сменили российскую юрисдикцию на зарубежную, это решение представляется для проекта единственным правильным.

Работа с машиночитаемыми требованиями в проектом направлении «Прорыв» проходит в четыре этапа:

- 1) Создание онтологии предметной области.
- 2) Создание семантической базы данных *Triple Store* [11], в качестве модели данных которой выступает онтология предметной области.
- 3) Выполнение разметки документа с требованиями на отдельные требования с уникальным

кодом. Каждое отдельное требование было переформулировано на языке *SHACL* [12] с сохранением исходного текста с требования.

4) Выполнение по команде пользователя автоматизированного ризонинга для получения логических выводов о выполнении требований.

Рассмотрим подробнее, как и с помощью каких инструментов были реализованы машиночитаемые требования и их автоматизированная проверка.

Как было отмечено выше, онтология является моделью данных, соответственно, каждый класс онтологии имеет экземпляры из состава хранимых данных. Требование на языке *SHACL* называются «Формой», так как является экземпляром класса *sh:Shape*. Для каждой «Формы» задается проверяемый класс онтологии или список классов. Указание на то, какие классы онто-

логии подлежат проверке в «Форме», задает отношение *sh:targetClass*. Примеры требований к экземплярам класса «Насос» и «Клапан», изложенных на языке *SHACL*, приведены на рис. 2 и 3.

В данной работе мы не будем подробно рассматривать синтаксис языка *SHACL*. Обратим внимание на то, то машиночитаемое требование является очень коротким, лаконичным логическим утверждением и не является фрагментом исполняемого кода какого-либо языка программирования. Отношение *sh:name* содержит уникальный код требования, а отношение *sh:description* содержит исходный текст требования из документа источника. Такая форма записи требования позволяет легко проверить соответствие утверждений, сделанных на естественном и математическом языках.

```
onto:PumpShape
  a sh:NodeShape ;
  sh:targetClass onto:Pump ;
  sh:property [
    sh:name "DQR.PMP.001" ;
    sh:description "Материалы для сборочных единиц и деталей трубопроводов, входящих
                    в состав насоса, применяют в соответствии с ГОСТ 20" ;
    sh:path onto:Material ;
    sh:minCount 1 ;
    sh:nodeKind sh:Literal ;
    sh:in ("30XMA" "18X3MB" "20X2MA" "20X3MBФ" "03X17H14M3" "08X18H10T" "08X18H12T"
          "12X18H10T" "12X18H12T" "10X17H13M2T" "10X17H13M3T" "08X17H15M3T") ;
  ] ;
.
```

Рис. 2. Пример машиночитаемого требования для насосов

```
onto: ValveShape
  a sh:NodeShape ;
  sh:targetClass onto:Valve ;
  sh:property [
    sh:name "DQR.VLV.004" ;
    sh:description "Изделие должно соответствовать 3-му классу безопасности по ОПБ-88/97" ;
    sh:path onto:SafetyClass ;
    sh:minCount 1 ;
    sh:nodeKind sh:Literal ;
    sh:in ("3") ;
  ] ;
.
```

Рис. 3. Пример машиночитаемого требования для клапанов

В машиночитаемом виде можно представить любые виды ограничений, накладываемые на атрибуты (характеристики) объектов. Например, все виды условий равенства/неравенства, проверка на наличие значения из определённого списка, проверка типов данных и даже проверки зависимости одних значений от других. Можно также проверять отсутствие нужных характеристик (некоторые примеры приведены на рис. 2 и 3) и формировать ограничения на мощность отношений, например на количество тех или иных атрибутов или других связанных объектов. В качестве примера на рис. 3 приведено отношение *sh:minCount*.

Кроме того, можно проверять любые тексты на соответствие шаблону строки, ее длине или другие виды операций со строками (рис. 4).

Еще один вид проверок касается отношений между объектами. Так как все отношения между объектами выражены семантическими предикатами и отражают инженерный взгляд на предметную область, то любые отношения между объектами и их данными, предусмотренные в онтологии, могут быть использованы при создании машиночитаемого требования.

На рис. 5 приведен пример проверки наличия организации—разработчика у документа через проверку существования отношения *onto:madeBy*.

Ризонинг, выполняемый программой в качестве вывода, показывает выявленные нарушения логики, однако этот вывод можно заменить пояснительным текстом, размещённым после отношения *sh:message*, как показано на рис. 4.

Для автоматизированной проверки машиночитаемых требований необходимо иметь соответствующее программное обеспечение. Ниже представлен список инструментов с открытым исходным кодом, поддерживающих проверку логических утверждений, сделанных на языке *SHACL*:

– *dotNetRDF SHACL* — онлайн-сервис-валидатор *SHACL*, написанный на платформе *.NET Framework*;

– *pySHACL* — библиотека валидаторов *SHACL* с открытым исходным кодом для использования в командной строке, написанная на *Python*;

– *SHaclEX* — *Scala*-реализация *SHACL* и *ShEx*;

– *SHACL Playground* — это бесплатный сервис проверки *SHACL*, реализованный на *JavaScript*.

Эти инструменты можно использовать самостоятельно или включить в состав своих ИТ-систем. В частности, для обеспечения регулярной проверки выполнения машиночитаемых требований в проектом направлении «Прорыв» был разработан собственный модуль на языке *Python*, включающий библиотеку *pySHACL*.

```
onto:ReqIDDocumentShape a sh:NodeShape ;
sh:targetClass onto:Document ;
sh:property [
  sh:name "DQR.018.001" ;
  sh:description "Суффикс в коде Ссылка должен соответствовать типу фрагмента текста и иметь значение L" ;
  sh:path rdfs:label ;
  sh:pattern "^.+L$" ;
  sh:flag "i" ;
  sh:message "Код не соответствует типу фрагмента текста." ;
] ;
.
```

Рис. 4. Пример машиночитаемого требования для документов

```
onto:madeByShape a sh:NodeShape ;
sh:targetObjectsOf onto:madeBy ;
sh:property [
  sh:path rdf:type ;
  sh:name "DQR.009.001" ;
  sh:description "У каждого документа должен быть указан его разработчик из числа Организаций";
  sh:nodeKind sh:IRI ;
]
```

Рис. 5. Пример машиночитаемого требования для проверки отношения между объектами

Этот модуль является частью информационной системы управления требованиями, используемой в проекте «Прорыв» для контроля проектных требований. В модуле реализованы три экранные формы:

- форма для редактирования требований (форма доступна только пользователям с определёнными правами);
- форма для просмотра и выбора требований и запуска их автоматизированной проверки (форма доступна любому пользователю модуля);
- форма вывода результатов проверки. Результаты проверки можно сохранить в виде файла и отправить заинтересованным сторонам.

Применяемое в проекте направление «Прорыв» решение является частью информационной системы управления требованиями, введенной в промышленную эксплуатацию.

Заключение

Опыт разработки и использования в проекте направления «Прорыв» машиночитаемых требований с помощью *Semantic Web* показал доступность и развитость необходимых технологий, возможность их применения в рамках организации. Однако необходимый экономический эффект от применения машиночитаемых требований можно получить только после того, как данный подход выйдет за границы организаций и будет использован в ходе приемки работ, в процессе выполнения экспертиз и прохождения различных разрешительных процедур, лицензирования. Именно сокращение сроков проверки соответствия проектной документации нормам и правилам существенно скажется на всей экономике страны. Трудоемкость экспертиз составляет многие человеко-месяцы, что делает процедуру лицензирования очень длительной и дорогостоящей для всех задействованных сторон. Автоматизированная проверка требований не только сократит это время до машино-часов, но и сделает ее результаты более объективными.

Опыт проекта «Прорыв» по разметке нормативной документации на отдельные требования для создания лицензионной базы в информационной системе управления требованиями показал низкую пригодность большинства нормативных документов к автоматизации в силу особенностей естественного языка и применения оценочных категорий, неоднозначности формулировок, отсутствие единой согласованной терминологии и единой структуры документа.

Аспектами, влияющими на перевод требований в машиночитаемый вид, являются законодательные ограничения, предписывающие сдавать определенные документы на бумажном носителе с «живой» подписью и печатью, а также отсутствие законодательно определенных форм электронных документов, информационных моделей

и используемых для передачи и обработки данных информационных технологий (протоколы, программное обеспечение). При этом следует отметить, что опыт приемки данных от организаций и населения в цифровом виде уже достаточно большой. Например, Федеральная налоговая служба Российской Федерации принимает отчетность в цифровом виде; работа с обращениями граждан ведется через портал *gosuslugi.ru* и др.

И, наконец, еще один барьер — это необходимость участия человека при проверке определенных норм.

Однако несмотря на перечисленные аспекты, переход на машиночитаемые требования стоит в планах российских и зарубежных разработчиков стандартов и законов. В качестве примера можно привести создание «Стратегической консультативной группы по машиночитаемым стандартам» (*ISO Strategic Advisory Group on Machine Readable standards*) в Международной организации по стандартизации (ИСО), в задачи которой входит разработка единого понятийного аппарата и рекомендаций по внедрению машиночитаемых стандартов для технических комитетов по стандартизации.

В России в настоящее время идет работа по правовому оформлению возможности проведения автоматизированной проверки проектной документации на соответствие нормативным документам для строительной отрасли. Законодательное оформление этого вопроса ожидается в ближайшее время согласно Стратегии развития [13]. В соответствии с [13] предусмотрены следующие мероприятия: устранение излишних противоречивых дублирующих нормативных технических требований в строительстве, перевод нормативных технических документов в машиночитаемый и машинопонимаемый форматы.

В заключение хочется отметить, что будущее формируется уже сейчас, поэтому проведение организациями научно-исследовательских и опытно-конструкторских работ, разработка пилотных проектов, информирование широкого круга лиц о своих достижениях по этому вопросу было бы очень своевременно и позволило бы выбрать наилучшее из опробованных решений, а законодательное оформление перехода на машиночитаемые стандарты позволит зафиксировать единые подходы, сократить сроки рассмотрения проектной документации и получения лицензий на строительство объектов.

Работа выполнена при поддержке Российской Федерации, от имени которой выступает Госкорпорация «Росатом», в рамках Государственного контракта №. Н.40.24 1.19.21.1037 от 28.02.2022 с дополнительным соглашением от 20.06.2022 № 1.

Литература

1. Chasalow Mr. The Element of Predictability in Judicial Decision Making/ California State Bar Journal, Nov.-Dec. 1960.
2. Jin Kook Lee. Building Environment Rule and Analysis (BERA) Language. URL: https://smartech.gatech.edu/bitstream/handle/1853/39482/Lee_Jin-Kook_201105_PhD.pdf [Электронный ресурс] (Дата обращения: 12.05.2023).
3. 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».
4. Jiansong Zhang. A logic-based representation and tree-based visualization method for building regulatory requirements/ Visualization in Engineering. 16.03.2017.
5. Hjelseth E., Nisbet N. (2011). Capturing normative constraints by use of the semantic mark-up RASE methodology// Proceedings of CIB W78-W102 Conference, 1—10, 2011.
6. Web standards. URL: <https://www.w3.org/standards/semanticweb/> [Электронный ресурс] (Дата обращения: 12.05.2023).
7. Beach T.H., Rezgui Y.R., Li H., Kasim T. A Rule-based Semantic Approach for Automated Regulatory Compliance in the Construction Sector. URL: https://publications.aston.ac.uk/id/eprint/39414/1/Rule_based_semantic_approach.pdf [Электронный ресурс] (Дата обращения: 12.05.2023).
8. Project openLAB: Interoperate – Machine Readable Handbook V440. URL: <https://buildingpoint-scandinavia.com/blog/project-interoperate-v440> [Электронный ресурс] (Дата обращения: 10.05.2023).
9. Волкодав В. Цифровой нормативно-технический документ в строительстве. Применение в рамках автоматизации процессов проверки ИМ ОКС // Материалы онлайн-конференции «Машиночитаемые стандарты: перспективы применения в промышленности» 25.02.2021. С.41—46. URL: [MAISHSTANDARTY_отчет_буклет.pdf](https://maishstandarty.ru/maishstandarty_otchet_buklet.pdf) (rgtr.ru) [Электронный ресурс] (Дата обращения: 12.05.2023).
10. Проект «Прорыв». URL: <https://proryv2020.ru/o-proekte/> [Электронный ресурс] (Дата обращения: 12.05.2023).
11. LargeTripleStores. URL: <https://www.w3.org/wiki/LargeTripleStores> [Электронный ресурс] (Дата обращения: 12.05.2023).
12. Shapes Constraint Language (SHACL). W3C Recommendation 20 July 2017. URL: <https://www.w3.org/TR/shacl/> [Электронный ресурс] (Дата обращения: 12.05.2023).
13. Стратегия развития строительной отрасли и жилищно-коммунального хозяйства Российской Федерации на период до 2023 года с прогнозом до 2025 года. Утв. Распоряжением Правительства Российской Федерации от 31.10.2022 от № 3268-р.

THE EXPERIENCE OF MACHINE-READABLE REQUIREMENTS APPLICATION IN PRORYV PROJECT

O. V. Zhdanova

"Proryv" Joint Stock Company, Moscow Russia

O. N. Stolyarov

"Atomenergoproekt" Joint Stock Company, Moscow Russia

The article presents the experience of Proryv Project implemented by the State Atomic Energy Corporation Rosatom in developing and applying machine-readable requirements in requirement management. The authors provide an explanation of the evaluation of machine-readable requirements, what information technologies have been used to develop and verify machine-readable requirements in recent decades. The article presents in details the experience of applying one of the methods in Proryv Project and provides examples of requirements. Aspects that make difficult the further distribution of machine-readable requirements as a special way of identifying restrictions and their verification are covered in the article. Current situation with the development of Russian regulatory framework in this field is touched as well.

Keywords: requirements management system, machine-readable requirements, requirements verification algorithm, automated requirement verification, requirements recording language.

Жданова Оксана Валерьевна, начальник отдела развития информационной модели и системы управления требованиями ЗЯТЦ.
E-mail: zhdov@proryv2020.ru
Столяров Олег Николаевич, начальник управления научно-технологического обеспечения проектирования АЭС.
E-mail: Stolyarov_ON@aep.ru

Статья поступила в редакцию 6 июня 2023 г.