

Е.В. Макиша

## **АНАЛИЗ МЕТОДОВ ПЕРЕВОДА ТРЕБОВАНИЙ НОРМАТИВНО-ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ В МАШИНОЧИТАЕМЫЙ ФОРМАТ ДЛЯ ПРОВЕРКИ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ СТРОИТЕЛЬНЫХ ОБЪЕКТОВ**

*Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский Московский государственный строительный университет»  
Москва, Россия*

*В настоящее время проектная документация поступает на экспертизу в виде электронных документов, представляющих собой текстовую и графическую информацию. Стандарты, на соответствие которым документация проверяется в процессе экспертизы, публикуются в электронной форме, однако не являются машиночитаемыми. В последнее время для российских специалистов строительной отрасли все очевидней становится возможность и необходимость автоматизированной экспертизы, которая может быть доступна при машиночитаемом представлении как результатов проектирования, так и соответствующих стандартов. В данной статье представлены результаты анализа существующих подходов к переводу в машиночитаемый формат требований к информационным моделям строительных объектов как форме представления результатов проектирования для проведения автоматизированной экспертизы. Определено наиболее перспективное из них направление – язык представления знаний, основанный на правилах. Установлены и дополнены критерии к машиночитаемому представлению требований. В качестве решения предложен язык RuleML, соответствующий указанным критериям, и зафиксированный в российском стандарте по информационному моделированию. Метод получения машиночитаемой формы требований стандартов, основанный на RuleML, в дальнейшем может быть использован при переходе к автоматизированной экспертизе как в нашей стране, так и в качестве унифицированного подхода на международном уровне.*

**Ключевые слова:** строительство, информационное моделирование, строительная система, кибернетика строительных систем, верификация информационной модели, автоматизированная проверка информационной модели, машиночитаемые форматы стандартов

### **Введение**

Экспертиза проектной документации представляет собой важный этап жизненного цикла строительного объекта, позволяющий обеспечить его качество и безопасность. Согласно Градостроительному кодексу Российской Федерации, процесс экспертизы представляет собой проверку соответствия проектной документации техническим регламентам, и, в частности, нормативно-технической документации, применение которой на обязательной основе обеспечивает выполнение технических регламентов. В

настоящее время проектная документация поступает на экспертизу в виде электронных документов, содержащих текстовую и графическую информацию, представленную в виде двумерных чертежей и схем. Стандарты также публикуются в электронной форме, однако не являются машиночитаемыми. При данных форматах представления проектной документации и нормативно-технической документации, экспертиза возможна только путем анализа специалистами соответствия результатов проектирования необходимым нормам. Это делает данный процесс продолжительным по времени и допускает возможность того, что эксперт не обнаружит имеющиеся ошибки.

Приобретающая все большее распространение и внедрение технология информационного моделирования [1] позволяет получить информацию о проектируемом объекте в виде описания его элементов и их свойств (атрибутов), что в отличие от двумерных чертежей, состоящих из графических примитивов, может быть использовано для машинной проверки.

Ученые и специалисты Сингапура, Норвегии, Австралии и США уже некоторое время занимаются вопросами автоматизированной проверки информационных моделей. В результате были получены разработки, часть из которых представляет собой тиражируемые коммерческие программные продукты, выполняющие ограниченный набор проверок, такие как Autodesk NawisWorks и Solibri Model Checker. Некоторые решения были подготовлены для автоматизации проведения экспертизы в отдельных странах, например, система CORENET в Сингапуре. Ведущие российские специалисты строительной отрасли также приходят к пониманию того, что экспертизу проектной документации можно и нужно проводить в автоматизированном режиме [4]. Однако ограниченность и локальность имеющихся подходов не позволяет применить их для проведения автоматизированной экспертизы в России. При этом наибольшую сложность представляет перевод всего объема требований нормативно-технической документации в машиночитаемый формат. Одновременно с этим, все возрастающая международная интеграция в проектировании требует появления унифицированных подходов в том числе и на уровне проверок информационных моделей объектов, в связи с чем становится все более актуальной задача подбора стандартного решения по получению машиночитаемой формы требований.

В качестве цели исследования, описанного в данной статье, был определен поиск наилучшего метода перевода требований нормативно-технической документации в машиночитаемый формат, который может быть использован при переходе к автоматизированной экспертизе как в нашей стране, так и в качестве унифицированного подхода на

международном уровне.

Для решения данной цели были поставлены следующие задачи:

1. Выявление и анализ существующих подходов.
2. Поиск и формирование критериев, применимых для выбора соответствующего подхода к переводу требований в машиночитаемый формат.
3. Выбор необходимого инструмента в соответствии с результатами анализа и указанными критериями.

### **Анализ существующих подходов**

Основываясь на ранее проделанном обзоре существующих методик проверки информационных моделей [3, 4], был сделан вывод, что в настоящее время сложилось три основных подхода к переводу требований нормативных документов в машиночитаемый формат:

*1. Кодирование на языке программирования высокого уровня.*

Самым простым способом является непосредственное кодирование требований нормативов на языке программирования. В качестве языков программирования могут быть использованы как императивные, так декларативные языки.

Примером реализации данного подхода является система Jotne EDModelChecker. Требования в системе EDM закодированы на языке EXPRESS, являющемся международным стандартом моделирования данных. [5, 6]

Были выявлены следующие недостатки данного способа:

- 1) Необходимость высокого уровня владения методами программирования и конкретными языками. Это означает, что реализация данного метода скорее всего будет невозможна специалистом в области того или иного стандарта, а поэтому для кодирования требований необходимо привлечение квалифицированного программиста.
- 2) Возможность ошибочной интерпретации и кодирования программистом, в виду его недостаточных знаний предметной области.
- 3) Сложности с корректировкой, обновлением, расширением базы закодированных требований на протяжении длительного периода времени.
- 4) Зависимость закодированных требований от конкретного проприетарного программного обеспечения, и как следствие их непереносимость на другие платформы и несовместимость с другими правилами.
- 5) Отсутствие связи с исходным текстом нормативного

документа, что может привести к несвоевременному обновлению правил, а значит и к неверным результатам проверки.

### *2. Кодирование в виде параметрических Таблиц.*

Данный подход, по сути, также представляет собой такое же кодирование требований на языке программирования, однако предоставляет их пользователю в виде параметрических Таблиц. Таблицы позволяют выполнять изменение параметров требований для различных контекстов пользователями без необходимости компьютерного программирования. Параметрические Таблицы являются компромиссом между простотой использования и возможностью определить и реализовать любое требование. [5, 6]

Примером системы, реализующей данный подход, является Solibri Model Checker (SMC). Наборы требований в SMC жестко закодированы в программное обеспечение, поэтому пользователи должны комбинировать существующие условия и настраивать их параметры, либо обращаться к разработчикам Solibri для определения новых типов правил. [5, 6]

Были определены следующие недостатки данного подхода:

- 1) Высокая стоимость поддержки подобных систем в результате необходимости привлечения компании-разработчика для корректировки, обновления и расширения базы закодированных требований.
- 2) Зависимость правил от конкретного проприетарного программного обеспечения, и как следствие их непереносимость и несовместимость с другими правилами.
- 3) Отсутствие связи с исходным текстом нормативного документа, что может привести к несвоевременному обновлению правил, а значит и к неверным результатам проверки.

### *3. Использование языка представления знаний, основанного на правилах.*

Наиболее перспективным методом получения машиночитаемых требований является их описание на одном или нескольких языках представления знаний, основанных на правилах. Привлекательность данного способа по сравнению с двумя ранее описанными в первую очередь обусловлена следующими причинами:

1. Простота и наглядность для предметных специалистов, что может позволить выполнять перевод требований в машиночитаемый формат непосредственно предметным специалистом без привлечения программиста.

2. Легкость внесения дополнений и изменений в существующую базу правил. [7]

Эти причины позволяют сделать вывод о необходимости более пристального рассмотрения возможности применения данного подхода к переводу требований к информационным моделям в машиночитаемый формат.

Примером системы проверки информационных моделей, реализующей данный подход, является SMARTcodes. Однако «словарь тегов» или онтология, используемая в данном решении для генерации правил, разрабатывалась индивидуально под данный проект с учетом особенности страны и конкретного нормативного документа, и не является открытым стандартом. Опыт SMARTcodes был положен в основу ряда исследований, основанных на механизме тегирования RASE [8] и его расширении на базе механизма, правил DROOLS [9]. Другие специалисты пытались применить к маркировке SMARTcodes язык программирования запросов LINQ [10], а также использовали семантическую веб-технику OWL и SWRL в комплексе с движком правил JESS [11]. Однако, по сути, ни одно из этих начинаний не получило международного распространения, главным образом, из-за отсутствия возможности установить взаимосвязь между базой правил и исходным текстом нормативного документа. [12, 13]

### **Критерии для машиночитаемого представления требований**

При выборе того или иного способа перевода требований в машиночитаемый формат или конкретного языка, необходим набор критериев, соответствие которым является первоочередной причиной для принятия решения в пользу того или иного инструмента.

Специалисты из Новой Зеландии в своем исследовании [12] предложили критерии для машиночитаемого представления требований информационной модели строительного объекта (Таблица 1), которые в большей степени учитывают недостатки существующих подходов.

Таблица 1 - Критерии для машиночитаемого представления требований

Критерий	Описание
К1	Представление требований должно быть как человеко-, так и машино- читаемым.
К2	Должна быть тесная связь между цифровым и исходным бумажным документом, чтобы обеспечить автоматический контроль версий. Это также позволит выполнять ознакомление пользователя со структурой и буквальным содержанием исходного документа.
К3	Наличие практических инструментов для разработки правил.
К4	Представление должно быть основано на технологии открытого стандарта, которая способствует взаимодействию и поддерживает открытые стандартные языки запросов.
К5	Подход должен быть масштабируемым и иметь возможность обрабатывать любые типы документов.

В свою очередь, на основании проведенного анализа, автор считает необходимым дополнить данный перечень критериями, представленными в Таблице 2.

Таблица 2 - Критерии для машиночитаемого представления требований, предложенные автором

Критерий	Описание
К6	Представление должно быть кроссплатформенным, то есть данная реализация не должна зависеть от определенной программной или аппаратной платформы.
К7	Представление должно иметь длительный цикл поддержки (сопровождения) для обеспечения совместимости с будущими системами и правилами.
К8	Представление должно отображать неограниченный диапазон требований, включая неограниченные вложенные условия и ветвление альтернативных контекстов в пределах определенной области знаний.

### Предлагаемое решение

Как было обозначено ранее, наиболее перспективным способом получения машиночитаемых требований к информационным моделям строительных объектов является использования языка представления знаний, основанного на правилах. В число таких языков в настоящее время входит несколько, к примеру, такие как SWRL, CLIPS, JESS, Drools, и

прочие.

Вообще языки, основанные на правилах, уже долгое время применяются для решения задач во многих профессиональных сферах. В связи с этим, возникла ситуация, при которой данные языки для описания одних и тех же терминов использовали различные обозначения. Чтобы унифицировать представление правил в различных доменах потребовался стандарт, которым в настоящее время фактически принят RuleML [14]. RuleML представляет собой «мост» между различными системами правил, например между RIF, SWRL, SWSL, TPTP, Common Logic и LegalRuleML, и, скорее всего, данные взаимосвязи будут увеличиваться.

RuleML в полной мере удовлетворяет критериям, перечисленным ранее (Таблица 3). Так, RuleML основан на логике первого порядка, что позволяет получить машиночитаемое представление, в то же время понятное для человека (K1). В своем синтаксисе RuleML имеет атрибут @iri, позволяющий устанавливать связь с исходным тестом документа (K2). Для работы с RuleML реализованы практические средства разработки, такие как Lime и Rawe (K3). RuleML – фактический открытый стандарт обмена правилами, разработанный и поддерживаемый некоммерческой организацией RuleML Inc., созданной в 2000 году Гарольдом Боли, Бенджамином Грозофом и Саидом Табетом; его спецификация представлена на официальном сайте (K4). Подход является масштабируемым, например, уже имеются расширения MathML для математических правил и LegalRuleML для юридических документов (K5). Поскольку RuleML разрабатывался как обменный формат, правила, написанные на нем, не зависят от определенной программной или аппаратной платформы, имеют длительный цикл поддержки, могут взаимодействовать с другими базами правил (K6-7). Основанные на положениях логики первого и высших порядков, правила на RuleML могут отображать неограниченный диапазон требований, включая неограниченные вложенные условия и ветвление альтернативных контекстов в пределах определенной области знаний (K8).

Таблица 3 – Сравнение существующих подходов получения машиночитаемых требований с RuleML

Критерий	Кодирование на языке программирования	Кодирование в виде параметрических	Использование языка представления знаний, основанного
----------	---------------------------------------	------------------------------------	---

	высокого уровня	Таблиц	на правилах (RuleML)
K1	Нет	Частично	Да
K2	Нет	Нет	Да
K3	Да	Да	Да
K4	Нет	Нет	Да
K5	Нет	Нет	Да
K6	Нет	Нет	Да
K7	Нет	Нет	Да
K8	Нет	Нет	Да

Кроме того, применение спецификации RuleML, рекомендовано на государственном уровне в ГОСТ Р 57296-2016 «Интегрированный подход к управлению информацией жизненного цикла антропогенных объектов и сред. Описание данных для математического моделирования процессов жизненного цикла. Основные положения»:

«6.5.6 Информационная модель может иметь информационный набор нормативно-технических ограничений, описанных в соответствии со спецификацией RuleML.»

Эти причины позволяют сделать вывод, что язык моделирования правил RuleML является наиболее перспективным инструментом представления нормативных требований для верификации информационных моделей строительных объектов как в России (в соответствии с ГОСТ), так и на международном уровне (соответствие критерию K4).

RuleML охватывает широкий диапазон правил, включая декларативные и императивные правила, которые реализованы соответственно в семействах Deliberation RuleML и Reaction RuleML. На Рисунке 1 представлена структура RuleML, специализация и взаимосвязи между семействами Deliberation, Reaction и Consumer RuleML. [15]



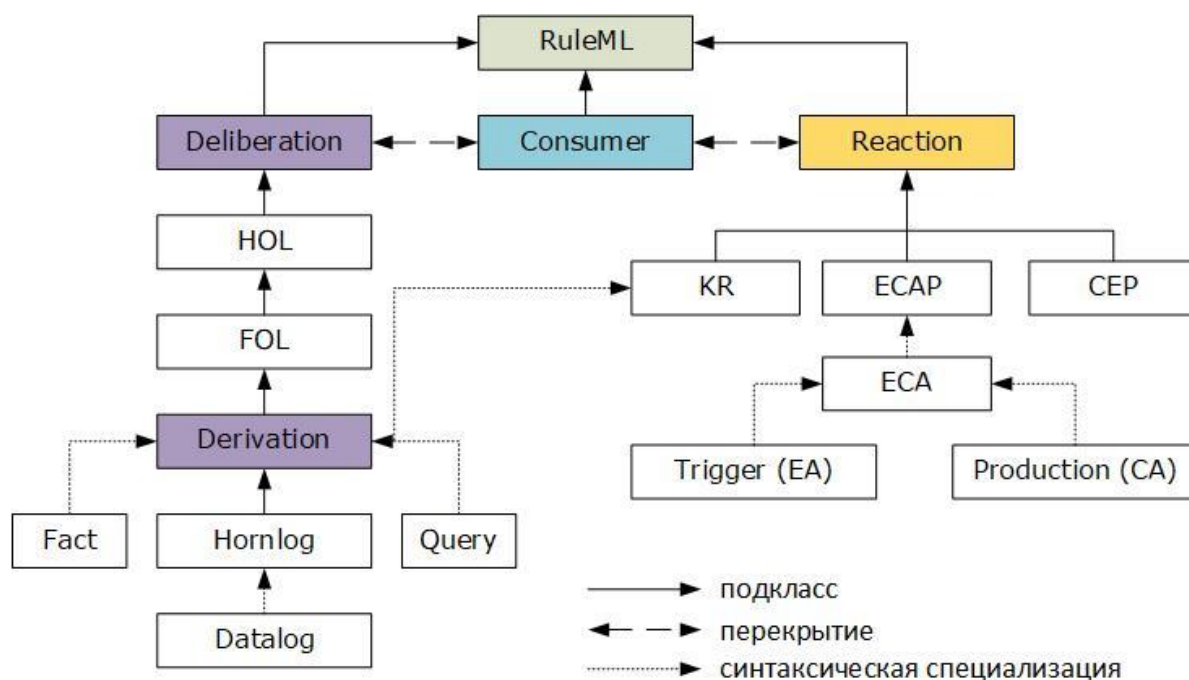


Рисунок 1 – Структура RuleML [15]

Deliberation RuleML – это одно из семейств RuleML, которое включает в себя принципы описания декларативных правил. Поскольку положения строительных нормативных актов, по сути, представляют собой описание того, каким должно быть здание или сооружение, и их можно интерпретировать как декларативные правила, то для описания нормативных требований следует использовать именно это семейство RuleML.

### Заключение

В данной статье представлены результаты анализа существующих подходов к переводу требований нормативно-технической документации к информационным моделям строительных объектов в машиночитаемый формат. Определено наиболее перспективное из них направление – язык представления знаний, основанный на правилах. В качестве такого языка предложен RuleML, поскольку он соответствует критериям, сформированным на основе недостатков существующих подходов, и закреплён в стандарте по информационному моделированию. В дальнейшем автором планируется детальное изучение принципов данного языка и разработка алгоритма формирования правил верификации информационной модели на его базе.

### ЛИТЕРАТУРА

1. Гинзбург А.В., Кулакова В.В., Куликова Е.Н., Малыха Г.Г., Шеина С.Г. Оценка уровня применения информационных технологий // Наука и бизнес: пути развития. 2018. № 9. с. 22-28.
2. Концепция внедрения системы управления жизненным циклом объектов капитального строительства с использованием технологии информационного моделирования // Информационный портал Национального объединения изыскателей и проектировщиков. URL: [nopriz.ru/upload/iblock/b6f/Kontseptsiya-BIM-pervaya-redaktsiya.pdf](http://nopriz.ru/upload/iblock/b6f/Kontseptsiya-BIM-pervaya-redaktsiya.pdf) (дата обращения: 26.04.2019)
3. Галкина Е.В. Анализ инструментов верификации проектной документации // Научно-технический вестник Поволжья. 2018. №6. с. 95-97.
4. Галкина Е.В. Перспективы использования систем проверки информационных моделей в России // Научное обозрение. 2017. № 21. с. 159-161.
5. Eastman C., Lee J.-M., Jeong Y.-S., Lee J.-K. Automatic rule-based checking of building designs // Automation in Construction. 2009. v. 18. pp. 1011–1033.
6. Lee H., Lee J.-K., Park S., Kim I. Translating building legislation into a computer-executable format for evaluating building permit requirements // Automation in Construction. 2016. v. 71. pp. 49-61.
7. Юрин А.Ю., Дородных Н.О., Коршунов С.А. Средства поддержки моделирования логических правил в нотации RVML // Программные продукты и системы. 2018. Т. 31. №4. с. 667-662.
8. Hjelseth E., Nisbet N. Capturing normative constraints by use of the semantic mark-up (RASE) methodology // CIB W78 2011 28th International Conference-Applications of IT in the AEC Industry, Sophia Antipolis, France, 2011.URL: [academia.edu/32697265/Capturing\\_Normative\\_Constraints\\_by\\_Use\\_of\\_the\\_Semantic\\_Mark-Up\\_Rase\\_Methodology](http://academia.edu/32697265/Capturing_Normative_Constraints_by_Use_of_the_Semantic_Mark-Up_Rase_Methodology) (date of access: 08.06.2018)
9. Beach T., Kasim T., Li H., Nisbet N., Rezgui Y. Towards automated compliance checking in the construction industry // Database and Expert Systems Applications, eds.: H. Decker, L. Lhotská, S. Link, J. Basl, A. Tjoa. Berlin Heidelberg: Springer, 2013. v. 8055. pp. 366–380.
10. Nawari N.O. Automating codes conformance in structural domain // ASCE International Workshop on Computing in Civil Engineering, Miami, Florida, June 19–22, 2011. URL: [ascelibrary.org/doi/abs/10.1061/41182%28416%2970](http://ascelibrary.org/doi/abs/10.1061/41182%28416%2970) (date of access: 08.06.2018)

11. Zhong B., Ding L., Luo H., Zhou Y., Hu Y., Hu H. Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking // Automation in Construction. 2012. v. 28. pp. 58–70.
12. Dimyadi J., Governatori G., Amor R. Evaluating LegalDocML and LegalRuleML as a Standard for Sharing Normative Information in the AEC/FM Domain // Proc. Lean & Computing in Construction Congress (LC3) Heraklion, Greece, July 4-7, 2017. v. 1. pp. 637-644.
13. Solihin W., Eastman C. Classification of rules for automated BIM rule checking development // Automation in Construction. 2015. v. 53. pp. 69-82.
14. RuleML. URL: [http://wiki.ruleml.org/index.php/RuleML\\_Home](http://wiki.ruleml.org/index.php/RuleML_Home) (date of access: 08.06.2018)
15. Boley H. Paschke A., Athan T., Giurca A., Bassiliades N., Governatori G., Palmirani M., Wyner A., Kozlenkov A., Zou G. Specification of RuleML 1.02. URL: [ruleml.org/1.02](http://ruleml.org/1.02) (date of access: 21.01.2019)

E.V. Makisha

**ANALYSIS OF METHODS OF REGULATION REQUIREMENTS  
TRANSLATION INTO MACHINE-READABLE FORMAT FOR  
VERIFICATION OF BUILDING INFORMATION MODELS**

*Moscow state university of civil engineering,  
Moscow, Russia*

*Currently, the project documentation is submitted for examination in the form of electronic documents that normally contain text and graphic information. Standards, which requirements the project documentation should meet during the examination, are published in electronic, but not machine-readable form. In recent years, specialists of the construction industry of Russian Federation fully realize the need of automated examination, which can be available if both sides of the process - design results and relevant standards - are presented in machine-readable form. This article reveals the results of existing approaches analysis how to transfer into the machine-readable form the regulation requirements for the building information model as a form of design results presentation for automated examination. The most promising of them is the language of knowledge representation based on the rules. There are criteria for machine readable presentation established and extended. The RuleML language is suggested as a solution, since it meets the criteria mentioned and is fixed in the regulation standards for information modeling. The RuleML based method of machine-readable requirements is acquired, which may in future be implemented within the transition to automated examination in the scope of Russia Federation and as a unified approach on the international level*

**Keywords:** construction, information modeling, construction system, construction cybernetics, verification of information modeling, automated check of information modeling, machine-readable standards

## REFERENCES

1. Ginzburg A.V., Kulakova V.V., Kulikova E.N., Malykha G.G., Sheina S.G. *Nauka i biznes: puti razvitiya*. 2018. No. 9. pp. 22-28.
2. Kontseptsiya vnedreniya sistemy upravleniya zhiznennym tsiklom obektov kapitalnogo stroitel'stva s ispolzovaniem tekhnologii informatsionnogo modelirovaniya [The concept of implementing a lifecycle management system for capital construction objects using information modeling technology]. *Informatsionnyy portal Natsionalnogo obedineniya izyskateley i proektirovshchikov*. URL: [nopriz.ru/upload/iblock/b6f/Kontseptsiya-BIM-pervaya-redaktsiya.pdf](http://nopriz.ru/upload/iblock/b6f/Kontseptsiya-BIM-pervaya-redaktsiya.pdf) (date of access: 26.04.2019)
3. Galkina E.V. *Nauchno-tekhicheskiy vestnik Povolzhya*. 2018. №6. pp. 95-97.
4. Galkina E.V. *Nauchnoe obozrenie*. 2017. no. 21. pp. 159-161.
5. Eastman C., Lee J.-M., Jeong Y.-S., Lee J.-K. *Automation in Construction*. 2009. v. 18. pp. 1011–1033.
6. Lee H., Lee J.-K., Park S., Kim I. *Automation in Construction*. 2016. v. 71. pp. 49-61.
7. Yurin A.Yu., Dorodnykh N.O., Korshunov S.A. *Programmnye produkty i sistemy*. 2018. v. 31. № 4. pp. 667-662.
8. Hjelseth E., Nisbet N. *CIB W78 2011 28th International Conference- Applications of IT in the AEC Industry*, Sophia Antipolis, France, 2011. URL: [academia.edu/32697265/Capturing\\_Normative\\_Constraints\\_by\\_Use\\_of\\_the\\_Semantic\\_Mark-Up\\_Rase\\_Methodology](http://academia.edu/32697265/Capturing_Normative_Constraints_by_Use_of_the_Semantic_Mark-Up_Rase_Methodology) (date of access: 08.06.2018)
9. Beach T., Kasim T., Li H., Nisbet N., Rezgui Y. *Database and Expert Systems Applications*, eds.: H. Decker, L. Lhotská, S. Link, J. Basl, A. Tjoa. Berlin Heidelberg: Springer, 2013. v. 8055. pp. 366–380.
10. Nawari N.O. *ASCE International Workshop on Computing in Civil Engineering*, Miami, Florida, 2011. URL: [ascelibrary.org/doi/abs/10.1061/41182%28416%2970](http://ascelibrary.org/doi/abs/10.1061/41182%28416%2970) (date of access: 08.06.2018)
11. Zhong B., Ding L., Luo H., Zhou Y., Hu Y., Hu H. *Automation in Construction*. 2012. v. 28. pp. 58–70.
12. Dimyadi J., Governatori G., Amor R. *Proc. Lean & Computing in Construction Congress (LC3) Heraklion, Greece, 2017*. v. 1. pp. 637-644.
13. Solihin W., Eastman C. *Automation in Construction*. 2015. v. 53. pp. 69-82.
14. RuleML. URL: [http://wiki.ruleml.org/index.php/RuleML\\_Home](http://wiki.ruleml.org/index.php/RuleML_Home) (date of access: 08.06.2018)
15. Boley H., Paschke A., Athan T., Giurca A., Bassiliades N., Governatori G., Palmirani M., Wyner A., Kozlenkov A., Zou G. *Specification of RuleML 1.02*. URL: [ruleml.org/1.02](http://ruleml.org/1.02) (date of access: 21.01.2019)