

© А.Г. ИВАШКО, А.В. ГРИГОРЬЕВ, А.А. КРОПОТИН,
Е.О. ОВСЯННИКОВА

ivashco@mail.ru, 107th@mail.ru, aakropotin@icloud.com,
ek.ovsyannikova@gmail.com

УДК 510.5; 510.22; 519-6

ПРИМЕНЕНИЕ ТАБЛИЧНОГО АЛГОРИТМА ДЛЯ ВЕРИФИКАЦИИ МОДЕЛЕЙ БИЗНЕС-ПРОЦЕССОВ*

АННОТАЦИЯ. Целью работы является разработка метода верификации моделей бизнес-процессов с применением аппарата дескрипционных логик. Задача верификации заключается в определении наличия в моделях ошибок, которые могут привести к недостижимости тех или иных действий. В статье рассмотрены 5 основных типов ошибок, возникающих при моделировании в результате неправильного отображения бизнес-процесса, модифицирован метод формализации диаграмм бизнес-процессов в нотациях BPMN и UMLActivity как наиболее популярных среди используемых. Отмечены недостатки существующих подходов для верификации, в частности, инструмента сетей Петри. Предложен новый метод верификации моделей с помощью только лишь формализма дескрипционных логик. Для тестирования метода разработано программное обеспечение, формирующее базу знаний на языке OWL. Для осуществления логического вывода и поиска ошибок в моделях использовался табличный алгоритм. Представлены примеры тестовых моделей разных классов в зависимости от их размерности, а также результаты тестирования метода.

SUMMARY. The aim of this work is to develop the verification method of a business-process model applying the description logic instrument. The task of verification is to determine the presence of inconsistencies in the models, which can lead to unreachable actions. The article considers five main types of inconsistencies that arise from a business process modeling due to incorrect business-process representation and modification of a formalization method of the business process diagrams in BPMN and UML Activity notations as most widely-used. Limitations of existing approaches for verification are given, in particular the instrument of Petri Nets. The new method of verification with the help of only description logic formalism is suggested. The special software forming the knowledge base in OWL was developed for testing the above method. The tableaux algorithm was used as the instrument for reasoning and model inconsistencies detection. Examples of test models of different classes depending on their dimension and results of method testing are presented.

КЛЮЧЕВЫЕ СЛОВА: Имитационное моделирование, дескрипционная логика, верификация, бизнес-процесс, BPMN, UMLActivity.

KEYWORDS: Simulation modeling, description logic, verification, business-process, BPMN, UMLActivity.

* Работа выполнена по проекту «Методология построения управляемой архитектуры предприятия на основе онтологического представления информационных систем» в соответствии с заданием Министерства образования и науки Российской Федерации на 2013 год.

Введение. Верификация моделей бизнес-процессов позволяет определить возникшие на стадии построения подобных моделей синтаксические ошибки или несоответствия поведения модели бизнес-процессу. Именно благодаря успешному проведению верификации модели бизнес-процесса можно приступить к этапу анализа процесса по его модели.

Актуальность работы обусловлена тем, что построение моделей бизнес-процессов является неотъемлемой частью задачи проектирования информационных систем и бизнес-реинжиниринга [1]. В работе рассматриваются наиболее популярные среди существующих нотаций и методологий моделирования бизнес-процессов, такие как BPMN [2], UMLActivity [3].

Предложенный в работе подход верификации бизнес-процессов позволяет определить ошибки проектирования, не связанные с несоблюдением синтаксических правил нотации проектирования, а ошибки, возникающие в результате неправильного отображения бизнес-процесса при построении его модели. В первом случае достаточно просто проверить соблюдение синтаксических правил нотации проектирования, но во втором случае этого может оказаться недостаточно, так как для выявления подобных ошибок проектирования необходимо произвести имитацию бизнес-процесса с целью сравнения поведения модели и самого процесса. Отметим, что выявление рассмотренных в данной работе ошибок без применения информационных технологий затрудняется по мере роста размера (количества элементов) модели бизнес-процесса, т.к. нотации и методологии, которым посвящена данная работа, содержат в себе не один десяток возможных синтаксических конструкций (имеют достаточно большую степень выразительности), которые встречаются на реальных предметно-ориентированных моделях не одну сотню раз. Из чего можно сделать вывод, что имитация построенных моделей с целью выявления неправильной проекции процесса средствами мысленного моделирования (проведения мысленного эксперимента) практически невозможна в связи с ограниченными человеческими способностями (кратковременная память, внимание и т.д.).

В последние годы активно разрабатываются методы формализации моделей бизнес-процессов с целью применения логического аппарата для верификации этих моделей [4-17], что не может не утверждать о том, что на данный момент не существует идеального и неоспоримого метода представления модели бизнес-процесса в какой-либо формальной системе и алгоритмов верификации этой модели, а также, что данная тематика является достаточно актуальной.

Можно добавить, что на данный момент существует методика выявления синтаксических ошибок проектирования с применением дескрипционной логики [13-17] и выявления ошибок, связанных с несоответствием поведенческих аспектов моделируемого бизнес-процесса своей проекции с помощью применения алгоритмов сетей Петри. Идея работы состоит в создании метода проверки диаграмм на наличие ошибок любого рода с применением только лишь формализма дескрипционных логик, без применения алгоритмов сетей Петри, тем самым будет исключена необходимость построения моделей в нескольких формальных системах, учитывая особенности каждой.

Целью данной работы является разработка альтернативного подхода верификации бизнес-процессов с применением аппарата дескрипционных логик, который позволит разработать гибкий инструмент верификации моделей.

Постановка задачи и обзор решений. Задача верификации бизнес-процессов заключается в определении наличия в моделях ошибок, которые могут привести к недостижимости тех или иных действий. W. Sadiq, и M. Orlowska выделяют 5 возможных типов ошибок в моделях потоков данных [5]:

1. incorrectusage — синхронизация объекта только с одним входящим потоком;
2. deadlock — синхронизация двух взаимно исключающих потоков;
3. livelock — заикливание, без возможности прерывания;
4. unintentional multiple execution — слияние двух параллельных потоков;
5. active termintaion — параллельные потоки ведут более чем к одной завершающей задаче.

К настоящему моменту наиболее широко используемым подходом для верификации моделей является инструмент сетей Петри [1, 18]. Данный подход применим в случаях, когда элементы построенной модели могут быть однозначно отображены в элементы сети Петри, при этом алгоритм имитирует выполнение задач перемещением маркеров в сети, соответствующей бизнес-процессу. Подход сетей Петри используется для верификации моделей, описанных с помощью наиболее популярных нотаций: EPC [18], BPMN [2], UMLActivity [3]. Однако алгоритмы, использующие сети Петри, обладают рядом недостатков, среди которых прежде всего стоит отметить их принадлежность классу NP (имеют не полиномиальную вычислительную сложность), что может значительно сократить время работы алгоритма при увеличении количества элементов в моделях; также к недостаткам можно отнести негибкость данных алгоритмов, это значит, что определение нового типа ошибок приведет к доработке (или тотальной переработке) реализованного алгоритма.

Решение. В качестве нотации для описания моделей бизнес-процессов используется нотация BPMN, так как она предоставляет широкие возможности для описания любых возможных случаев в бизнес-процессах [19], кроме того, она является одной из наиболее популярных нотаций используемых при разработке информационных систем [20]. Для определения наличия в моделях ошибок авторами используется формализм дескрипционных логик [21] и табличный алгоритм [22], который используется для осуществления логического вывода и поиска ошибок в моделях.

На рисунке 1 показана диаграмма бизнес-процесса, в которой находится цикл, проходящий через элементы t1, g1 и t3. Однако ввиду того, что g1 является исключаяющим шлюзом (exclusive gateway), то процесс может быть завершен при прохождении через элементы e1, t1, g1, t2 и e2. Поэтому процесс не считается заикленным, если в какой-либо из интерпретаций он достигает своего окончания.

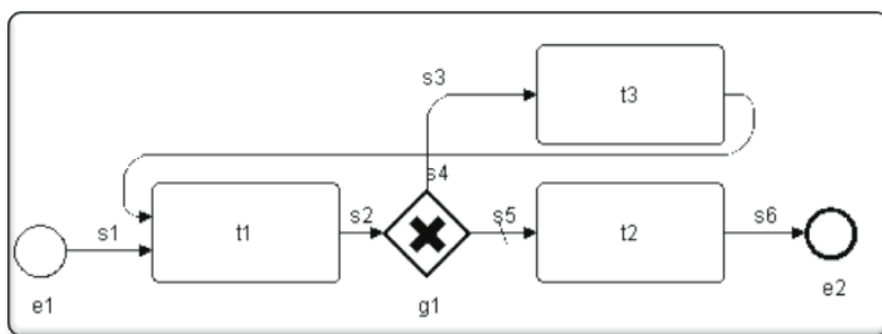


Рис. 1. Не зацикленный процесс в нотации BPMN

Для использования дескрипционных логик [22] и табличного алгоритма формализуем элементы нотации BPMN, применяя модифицированный подход R. Dijkman, M. Dumas и С. Оуанг [2], однако ограничимся лишь элементами задач, включающего и исключающего шлюзов, начальными и конечными событиями, а также потоками управления, учитывая тот факт, что большинство моделей могут быть построены без применения других элементов. Обозначим T — множество всех задач, G — множество всех шлюзов, также выделим множества исключающих (G_E) и включающих шлюзов (G_I) таких, что $G_E \cap G_I \equiv \emptyset$, $G_E \cup G_I \equiv G$, начальное событие s и конечное событие e . Множество всех объектов диаграммы обозначим $O \equiv G \cup T \cup \{s\} \cup \{e\}$. Отношением передачи управления R обозначим множество пар $(o_1, o_2) \in (O \times O)$, отражающих все стрелки на диаграмме BPMN. Для осуществления задач логического вывода используем логику *SROIQ*, которая позволяет использовать стандартные операции логики ALC [21], расширенные индивидами и обратными ролями.

При проверке моделей будет создана онтология, согласованность которой будет отражать наличие ошибок в модели. Для построения онтологии определим функцию f , отображающую множество O в элементы $AVox$ A онтологии, описанной на языке дескрипционной логики, а также отношение L , устанавливающее связи между элементами A :

1. каждый элемент $t_i \in T$ будет отображен в индивида $y_i \in A$, $f(t_i) = y_i$;
2. каждый элемент $g_i \in G_E$ будет отображен в индивида $h_i \in A$, $f(g_i) = h_i$;
3. каждый элемент $p_i \in G_I$ будет отображен в индивида $k_i \in A$, $f(p_i) = k_i$;
4. элемент $\{s\}$ будет отображен в индивида $d \in A$, $f(s) = d$;
5. элемент $\{e\}$ будет отображен в индивида $g \in A$, $f(e) = g$;
6. $L(f(s), f(o_j)) = \{CF\}$, для любого $o_j \in O$ такого, что $(s, o_j) \in R$;
7. $L(f(o_j), f(e)) = \{CF\}$, для любого $o_j \in O$ такого, что $(o_j, e) \in R$;
8. $L(f(t_i), f(o_j)) = \{CF\}$, для любого $o_j \in O$ такого, что $(t_i, o_j) \in R$;
9. $L(f(g_i), f(o_j)) = \{CF\}$, для любого $o_j \in O$, такого, что $(g_i, o_j) \in R$;
10. $L(f(o_j), f(o_j)) = \{-CF\}$, для любого $o_j \in O$;
11. Для элементов o_i определим аксиому $f(o_i) \sqsubseteq (\forall RC. \{ \neg(f(o_j)) \})$, для $o_i \in O$ при $(g_i, o_i) \notin R$;
12. Для элементов g_i определим концепт C_{g_i} , интерпретации которого они принадлежат, при этом $C_{g_i} \equiv (\sqcup \{ \exists CF. \{ f(o_j) \} \} \sqcap \forall CF. \neg C_j)$ для любого $o_j \in O$

при $(g_i, o_j) \in R$, $o_k \in O$ при $(o_k, g_i) \in R$, концепт C_j будет только тех индивидов $f(o_h)$, для которых $(g_i, o_h) \in R$ и $o_h \neq o_j$; в множество аксиом TBox T будет добавлена аксиома $\{g_i\} \sqsubseteq C_{g_i}$;

13. Для каждого из элементов g_i добавим в T аксиомы $\{g_i\} \sqcap \exists RC. \{t_j\} \sqcap \exists RC. \{tk\} \sqsubseteq \perp$, для всех различных t_j и t_k , при том, что $(t_i, g_i) \in R$ и $(t_j, g_i) \in R$ или $(g_i, t_i) \in R$ и $(g_i, t_j) \in R$;
14. Для каждого из элементов p_i , определим аксиому $\{f(p_i)\} \sqcap \exists CF. \{f(o_j)\} \sqsubseteq (\sqcap \exists RF. \{f(o_k)\})$, для любого $o_j \in O$ при $(p_i, o_j) \in R$, $o_k \in O$ при $(o_k, g_i) \in R$;
15. Для определения достижимости окончания процесса необходимо добавить аксиому $f(s) \sqsubseteq (\sqcap \exists CF. \{f(o_j)\})$, для любого $o_j \in O$;

Для проверки наличия livelock ошибок необходимо исключить шаг 11 данной формализации и добавить аксиому транзитивности роли CF. Для проверки согласованности онтологии будет использован стандартный табличный алгоритм для логики *SROIQ*, реализованный в системе HermiT [23]. Процедура проверки вернет значение *false*, если в модели нет зацикленных действий, иначе процедура вернет *true*. Формализация алгоритма представлена в листинге 1.

Листинг 1. Проверка наличия livelock ошибок.

Процедура Существует цикл (Диаграмма B)

1. OWL Онтология O = Преобразовать диаграмму (**B**);
2. Из онтологии O удалить аксиомы $f(o_i) \sqsubseteq (\forall RC. \{\neg(f(o_i))\})$;
3. Добавить в онтологию O аксиому о транзитивности роли **CF**.
4. Проверить согласованность онтологии O ;
5. Если O согласовано, то вернуть *false*, иначе вернуть *true*;

Процедура **Преобразовать диаграмму** преобразует модель в термины OWL в соответствии с представленными выше правилами.

Для определения недостижимых действий необходимо проверить согласованность онтологии, построенной по указанным правилам. Если такая база знаний будет невыполнима, то окончание процесса будет недостижимо. Определить достижимость какого-либо процесса можно также проверкой наличия элемента $f(t_i)$ в какой-либо из интерпретаций базы знаний. Процедура определения недостижимых действий представлена в листинге 2.

Листинг 2. Проверка наличия deadlock ошибок.

Процедура Недостижимые действия (Диаграмма B).

1. OWL Онтология O = Преобразовать диаграмму (**B**);
4. Проверить согласованность O ;
5. Если O несогласованно, то Вернуть *true*;
6. Вернуть *false*;

Процедуры поиска ошибок были разделены на два различных алгоритма, для того чтобы при верификации моделей можно было бы отличить тип каждой возникшей ошибки.

Не смотря на то, что изначально была поставлена задача верификации моделей бизнес-процессов, построенных в нотации BPMN, было обнаружено, что данный подход способен обеспечить поддержку верификации моделей, построенных и в нотации UMLActivity. В работе [19] представлены шаблоны для трансляции диаграмм бизнес-процессов из одной в другую, построенных в нотациях BPMN и UMLActivity. Таким образом, многие синтаксические конструкторы

ции BPMN диаграмм могут быть представлены синтаксическими конструкциями и UMLActivity диаграмм, не смотря на то, что они имеют различное графическое представление, т.е. один и тот же бизнес-процесс может быть одинаково отображен, как и в нотации BPMN, так и в нотации UMLActivity. Например, процесс, представленный на рисунке 1 в виде BPMN диаграммы, можно представить в виде UMLActivity диаграммы, отображенной на рисунке 2.

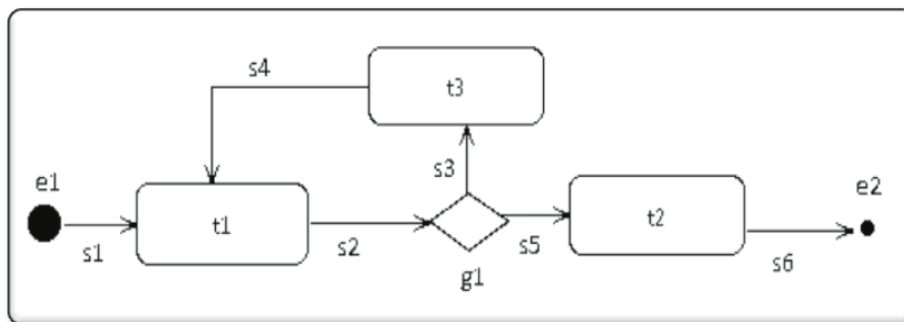


Рис. 2. Не зацикленный процесс в нотации UMLActivity

Благодаря возможности одинакового представления элементов процесса в нотациях BPMN и UMLActivity предложенный подход к верификации бизнес-процесса применим в обоих случаях.

Результаты. Для тестирования разработанного метода было разработано программное обеспечение BPMN2OWL, которое формирует базу знаний на языке OWL из XML файла, описывающего модель, которая создается в программном продукте Eclipse Modeling Tools, который использует инструменты построения BPMN2-Modeler и Papugus [20]. Для определения зацикленности процессов использовалась система логического вывода HermiT [23], предоставляемая вместе с инструментом построения онтологий Protege.

В качестве набора тестовых данных использовались модели трех классов в зависимости от их размерности (количества графических элементов на диаграммах):

- класс А — от 10 до 35 элементов;
- класс В — от 36 до 100 элементов;
- класс С — от 101 до 300 элементов.

Кроме того, на каждый из классов моделей строились диаграммы трех вариантов:

- не содержащие ошибки;
- содержащие ошибки livelock;
- содержащие ошибки, связанные с недостижимостью какого-либо действия.

Таким образом, положительной оценкой проверки нашего подхода является определение всех содержащих ошибки моделей и определение всех построенных правильно моделей бизнес-процессов табличным алгоритмом независимо от применяемой нотации (BPMN или UMLActivity).

Исходя из ограниченности пространства, на рисунках 3-6 приведены примеры тестовых моделей бизнес-процессов класса А.

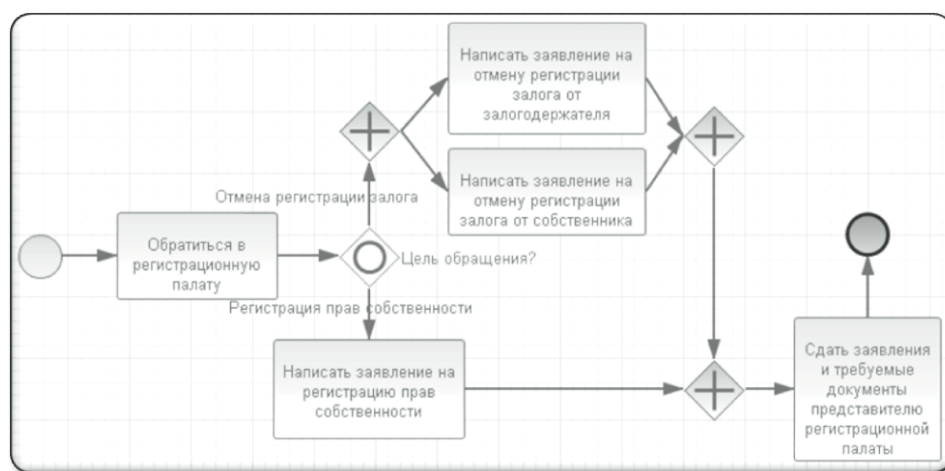


Рис. 3. Тестовый процесс класса А, не содержащий ошибок, в нотации BPMN

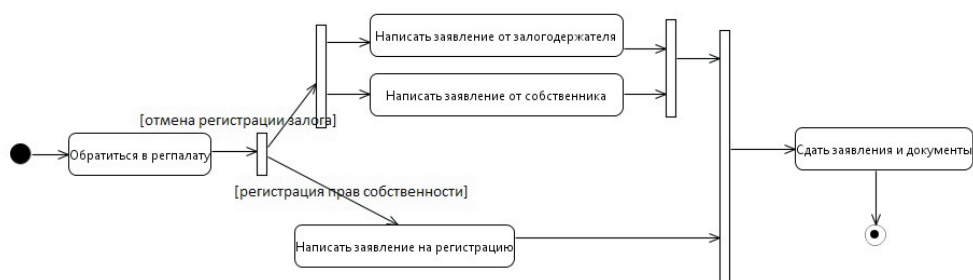


Рис. 4. Тестовый процесс класса А, не содержащий ошибок, в нотации UMLActivity

На рисунках 3-4 приведен пример тестового процесса класса А, не содержащего ошибок проектирования. На диаграмме отображен процесс регистрации прав собственности купленной в ипотеку квартиры за одно обращение в регистрационную палату. Основными задачами процесса являются подача заявлений на отмену регистрации залога и подача заявления на регистрацию прав собственности. Причем, эти задачи не могут выполняться параллельно, но и должны быть обе выполнены для достижения цели процесса, т.к. если выполнить одну из них, то будет либо отменена регистрация залога, либо получено сообщение об отказе регистрации в связи с тем, что нельзя регистрировать права собственности на имущество, находящееся в залоге.

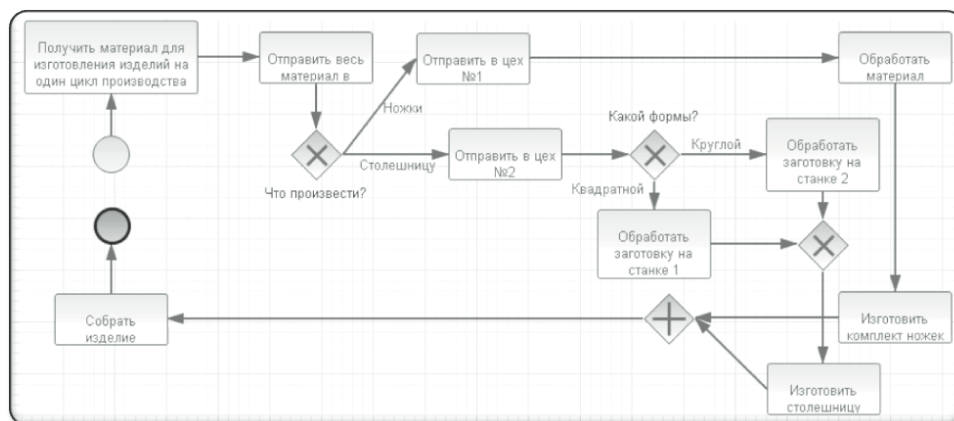


Рис. 5. Тестовый процесс класса А, содержащий ошибку недостижимости какого-либо действия, в нотации BPMN

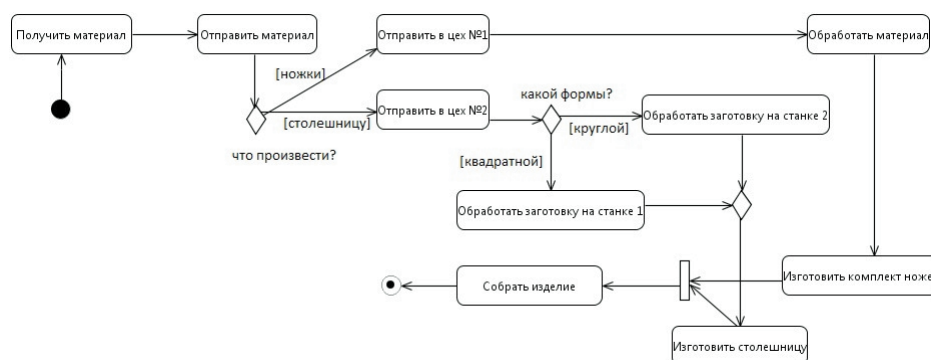


Рис. 6. Тестовый процесс класса А, содержащий ошибку недостижимости какого-либо действия, в нотации UMLActivity

На рис. 5-6 приведен пример тестового процесса класса А, содержащего ошибку недостижимости какого-либо действия. На диаграмме отображен процесс изготовления столов. В рамках данного процесса весь полученный для изготовления материал направляется на изготовление изделий в рамках одного цикла производства. Причем, все сырье отправляется либо на изготовление набора ножек будущего стола, либо на изготовление столешниц одной из двух форм, но для изготовления стола необходимо, чтобы одновременно были изготовлены и ножки, и столешница. В результате чего возникает недостижимость действия «Собрать изделие», потому что для выполнения этого действия необходимо одновременное завершение всех ветвей в логическом операторе «и» — чего никогда не может произойти.

Тестирование было проведено на компьютере, имеющем следующую конфигурацию: CPU: 2,2 GHz, RAM: 1024Mb под управлением операционной системы Windows XP Professional 64bit. Результаты тестирования моделей табличным алгоритмом приведены в таблице 1.

Таблица 1

Результаты тестирования метода

Размерность модели	Несодержащие ошибок	Содержащие ошибки livelock	Содержащие недостижимые действия
Класс А	1,76	2, 09	0,721
Класс В	2,579	1,402	1,632
Класс С	60,406	10,281	12,109

Результаты тестирования системы (табл. 1) показали, что разработанный метод определил ошибки во всех содержащих ошибки моделях и не нашел ошибок в построенных верно моделях, отображенных в виде BPMN и UMLActivity диаграмм. Кроме того, результаты показали, что на проверку наличия ошибок в моделях затрачивается время, приемлемое для верификации моделей.

Выводы. В результате работы, в области исследований были проанализированы синтаксические конструкции нотаций BPMN и UMLActivity, а также разработаны:

- методика выявления OWL онтологии из диаграммы бизнес-процесса, построенной в нотациях BPMN и UMLActivity;
- аксиоматические утверждения дескрипционной логики, позволяющие определить ошибки в моделях потоков данных, построенных в нотациях BPMN и UMLActivity;
- программное обеспечение BPMN2OWL, которое формирует базу знаний на языке OWL из XML файла построенной в нотациях BPMN и UMLActivity диаграммы.

СПИСОК ЛИТЕРАТУРЫ

1. Wynn, M.T., Verbeek, H.M.W., W.M.P. van der Aalst, A.H.M. ter Hofstede, Edmond, D. Business Process Verification — Finally a Reality! // Business Process Management Journal, Vol. 15(1), Pp.74-92.
2. Dijkman, R.M., Dumas, M., Ouyang, C. Formal Semantics and Analysis of BPMN Process Models using Petri Nets // Intelligent Information Technology Application. IITA '08. Second International Symposium. 2008. Pp. 70-74.
3. Lopez-Grao, J.P., Merseguer, J., Campos, J. From UML Activity Diagrams To Stochastic Petri Nets: Application To Software Performance Engineering* // Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium. 2008. P 70-74.
4. Object Management Group. Business Process Model and Notation (BPMN) Version 2.0 // <http://www.omg.org/spec/BPMN/2.0/PDF> (дата обращения: 18.05.2013)
5. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T. Querying business processes // In VLDB 06. 2006. Pp. 343-354.
6. Dimitrov, M., Simov, A., Stein, S., Konstantinov, M. A bpmo based semantic business process modeling environment // In Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management at the ESWC, vol. 251 of CEUR-WS, 2007.
7. Koschmider, A., Oberweis, A. Ontology based business process description // In Proceedings of the CAiSE-05 Workshops, LNCS. 2005. Pp. 321-333.
8. Markovic, I. Advanced querying and reasoning on business process models. In W. Abramowicz and D. Fensel, editors, BIS. Vol. 7 of Lecture Notes in Business Information Processing. 2008. Pp. 189-200.

9. De Nicola, A., Lezoche, M., Missikoff, M. An ontological approach to business process modeling // In Proceedings of the 3^d Indian International Conference on Artificial Intelligence (IICAI). 2007. Pp 1794-1813.
10. Thomas, O., Fellmann, M. Semantic epc: Enhancing process modeling using ontology languages // In Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM). 2007. Pp. 64-75.
11. Weber, I., Hoffmann, J., Mendling, J. Semantic business process validation // In Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM). 2008.
12. Wong, P.Y.H., Gibbons, J. A relative timed semantics for BPMN // In Proceedings of 7th International Workshop on the Foundations of Coordination Languages and Software Architectures (FOCLASA08). 2008.
13. Born, M., Dorr, F., Weber, I. User-friendly semantic annotation in business process modeling // In Hf-SDDM-07: Proceedings of the Workshop on Human-friendly Service Description, Discovery and Matchmaking WISE-07. 2007.
14. Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., P. Tonella. Reasoning on semantically annotated processes // In Proc. of the 6th Int. Conference on Service Oriented Computing (ICSOC'08), Vol. 5364 of LNCS. 2008. Pp. 132-146.
15. Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P. Semantically-aided business process modeling // In Proceedings of 8th International Semantic Web Conference (ISWC 2009). 2009. Pp. 114-129.
16. Ghidini, C., Rospocher, M., Serani, L. A formalisation of BPMN in description logics // Technical Report TR 2008-06-004, FBK, 2008.
17. Bozzato, L., Ferrari, M., Trombetta, A. Building a domain ontology from glossaries: a general methodology // In Proceedings of 5th Workshop on Semantic Web Applications and Perspectives, 2008.
18. Van Dongen, B.F. and Verbeek, H.M.W. Verification of EPCs: Using Reduction Rules and Petri Nets // Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05), Vol. 3520 of Lecture Notes in Computer Science. 2005. Pp. 372-386;
19. White, S. Process Modeling Notations and Workflow Patterns // In Workflow Handbook, 2004. Pp. 265-294.
20. MDT Papyrus Group. Papyrus // URL: <http://www.eclipse.org/papyrus/> (дата обращения: 20.05.2013).
21. Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P.F. (editors). The Description Logic Handbook: Theory, Implementation and Applications. CUP, 2003.
22. Schmidt-Schau, M. and Smolka, G. Attributive concept descriptions with complements // Artificial Intelligence, Vol. 48(1). 1991. Pp. 1-26.
23. Horrocks, I., Motik, B., Wang, Z. The HermiT OWL Reasoner // In Proc. ORE 2012. Pp. 136-141.

REFERENCES

1. Wynn, M.T., Verbeek, H.M.W., W.M.P. van der Aalst, A.H.M. ter Hofstede, Edmond, D. Business Process Verification — Finally a Reality! *Business Process Management Journal*. Vol. 15(1), Pp. 74-92.
2. Dijkman, R.M., Dumas, M., Ouyang, C. Formal Semantics and Analysis of BPMN Process Models using Petri Nets. Intelligent Information Technology Application. IITA '08. Second International Symposium. 2008. Pp. 70-74.

3. Lopez-Grao, J.P., Merseguer, J., Campos, J. From UML Activity Diagrams To Stochastic Petri Nets: Application To Software Performance Engineering*. Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium. 2008. Pp.70-74.
4. Object Management Group. Business Process Model and Notation (BPMN) Version 2.0 // <http://www.omg.org/spec/BPMN/2.0/PDF> (дата обращения: 18.05.2013)
5. Beerli, C., Eyal, A., Kamenkovich, S., Milo, T. Querying business processes // In VLDB 06. 2006. Pp. 343-354.
6. Dimitrov, M., Simov, A., Stein, S., Konstantinov, M. A bpmo based semantic business process modeling environment. In Proc. of the Workshop on Semantic Business Process and Product Lifecycle Management at the ESWC. Vol. 251 of CEUR-WS, 2007.
7. Koschmider, A., Oberweis, A. Ontology based business process description // In Proceedings of the CAiSE-05 Workshops, LNCS. 2005. Pp. 321-333.
8. Markovic, I. Advanced querying and reasoning on business process models. In W. Abramowicz and D. Fensel, editors, BIS. Vol. 7 of Lecture Notes in Business Information Processing. 2008. Pp. 189-200.
9. De Nicola, A., Lezocche, M., Missikoff, M. An ontological approach to business process modeling // In Proceedings of the 3^d Indian International Conference on Artificial Intelligence (IICAI). 2007. Pp 1794-1813.
10. Thomas, O., Fellmann, M. Semantic epc: Enhancing process modeling using ontology languages // In Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM). 2007. Pp. 64-75.
11. Weber, I., Hoffmann, J., Mendling, J. Semantic business process validation // In Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM). 2008.
12. Wong, P.Y.H., Gibbons, J. A relative timed semantics for BPMN // In Proceedings of 7th International Workshop on the Foundations of Coordination Languages and Software Architectures (FOCLASA08). 2008.
13. Born, M., Dorr, F., Weber, I. User-friendly semantic annotation in business process modeling // In Hf-SDDM-07: Proc. of the Workshop on Human-friendly Service Description, Discovery and Matchmaking WISE-07. 2007.
14. Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., P. Tonella. Reasoning on semantically annotated processes // In Proc. of the 6th Int. Conference on Service Oriented Computing (ICSOC'08), Vol. 5364 of LNCS. 2008. Pp. 132-146.
15. Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P. Semantically-aided business process modeling // In Proc. of 8th International Semantic Web Conference (ISWC 2009). 2009. Pp. 114-129.
16. Ghidini, C., Rospocher, M., Serani, L. A formalisation of BPMN in description logics // Technical Report TR 2008-06-004, FBK, 2008.
17. Bozzato, L., Ferrari, M., Trombetta, A. Building a domain ontology from glossaries: a general methodology // In Proceedings of 5th Workshop on Semantic Web Applications and Perspectives, 2008.
18. Van Dongen, B.F. and Verbeek, H.M.W. Verification of EPCs: Using Reduction Rules and Petri Nets // Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05), Vol. 3520 of Lecture Notes in Computer Science. 2005. Pp. 372-386;
19. White, S. Process Modeling Notations and Workflow Patterns // In Workflow Handbook, 2004. Pp. 265-294.
20. MDT Papyrus Group. Papyrus // URL: <http://www.eclipse.org/papyrus/> (дата обращения: 20.05.2013).

- 21. Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P.F. (editors). The Description Logic Handbook: Theory, Implementation and Applications. CUP, 2003.
- 22. Schmidt-Schau, M. and Smolka, G. Attributive concept descriptions with complements // Artificial Intelligence, Vol. 48(1). 1991. Pp. 1-26.
- 23. Horrocks, I., Motik, B., Wang, Z. The HermiT OWL Reasoner // In Proc. ORE 2012. Pp. 136-141.