

Е. В. Кочуров

## Применение логики построений на графах к исполнению моделей бизнес-процессов

**Аннотация.** В статье предлагается подход к исполнению бизнес-процессов, который основан на логическом синтезе программ по модели бизнес-процесса. В подходе используется логика построений на графах GL5.

**Ключевые слова и фразы:** конструктивная логика, построения на графах, моделирование бизнес-процессов.

### Введение

В последнее время были получены результаты, позволяющие синтезировать доказуемо правильные программы, которые реализуют иерархические [1] и сетевые [2] структуры вычислений. Существует целый ряд областей, в которых подобные подходы могут быть применены.

В данной статье подход, первоначально разрабатывавшийся для суперкомпьютеров, иллюстрируется на примере его приложения в другой актуальной области – это исполнение бизнес-процессов по имеющейся модели, заданной в той или иной нотации.

Если рассматривать задачи (activity в нотации BPMN [3]) бизнес-процесса, как программы, которые должны выполняться в соответствии с правилами, заданными моделью бизнес-процесса, то такой подход позволяет привнести в область управления бизнес-процессами (Business Process Management [4]) сильные стороны доказательного программирования.

---

Поддержана Минобрнауки России по теме: «Исследования и разработка технических решений по теме Развитие инфраструктуры суперкомпьютерных центров в интересах инновационного развития государств участников СНГ» RFMEFI61314X0030.

© Е. В. Кочуров, 2015

© ИНСТИТУТ ПРОГРАММНЫХ СИСТЕМ ИМЕНИ А. К. АЙЛАМАЗЯНА РАН, 2015

© ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ, 2015

## 1. Математическая модель

Пусть дан ориентированный граф  $G = \langle V, E, \text{begin}, \text{end} \rangle$ , где

- $V$  — множество вершин,  
 $E$  — множество ребер,  
 $\text{begin} : E \rightarrow V$  — функция, которая определяет вершину–начало ребра,  
 $\text{end} : E \rightarrow V$  — функция, которая определяет вершину–конец ребра.

Поскольку в данной работе рассматриваются только ориентированные графы, в дальнейшем будем называть их просто графами.

Для удобства будем использовать несколько дополнительных терминов для определения графов с определенными свойствами:

- исходящее ребро*  $e$  вершины  $v$  — такое, что  $\text{begin}(e) = v$ ;  
*входящее ребро*  $e$  вершины  $v$  — такое, что  $\text{end}(e) = v$ ;  
*сеть* — ациклический граф;  
*связная сеть* — сеть, которая состоит из единственной компоненты связности;  
*лес* — сеть, каждая вершина которой имеет не более одного входящего ребра;  
*дерево* — лес, который состоит из единственной компоненты связности;  
*начальная вершина* — вершина сети, которая не имеет входящих ребер;  
*конечная вершина* — вершина сети, которая не имеет исходящих ребер.

Построением на графе  $G$  будем называть такую сеть

$$G_s = \langle V_s, E_s, \text{begin}_s, \text{end}_s \rangle,$$

что существует отображение  $e$ , которое отождествляет вершины и ребра сети  $G_s$  с вершинами и ребрами графа  $G$ :

- a) для любого  $x \in V_s : e(x) \in V$ ,
- b) для любого  $x \in E_s : e(x) \in E$ ,
- c) для любого  $x \in E_s : e(\text{begin}_s(x)) = \text{begin}(e(x))$ ,
- d) для любого  $x \in E_s : e(\text{end}_s(x)) = \text{end}(e(x))$ .

Содержательно, построение на графе  $G$  можно интерпретировать, как сеть из путей, допускаемых графом  $G$ . Введем отношение  $\text{split}(G, G_1, G_2)$  разбиения сети  $G = \langle V, E, \text{begin}, \text{end} \rangle$  на две компоненты  $G_1 = \langle V_1, E_1, \text{begin}_1, \text{end}_1 \rangle$  и  $G_2 = \langle V_2, E_2, \text{begin}_2, \text{end}_2 \rangle$  со следующими свойствами:

- i)  $V_1 \cap V_2 = \emptyset$ ,
- ii)  $V_1 \cup V_2 = V$ ,

iii)  $E_1 \cap E_2 = \emptyset$ ,

iv) Для всякого ребра  $e \in E$ :

$$(e \in E_1) \vee (e \in E_2) \vee (\text{begin}(e) \in V_1 \wedge \text{end}(e) \in V_2).$$

Содержательно, отношение *split* разбивает сеть на компоненты связности либо, если это невозможно – на две сети так, чтобы связывающие их ребра были направлены строго в одном направлении.

**ОПРЕДЕЛЕНИЕ 1.** Будем называть отношение *split* модульным, если дополнительно выполняется свойство: подмножество всех ребер  $e$  из  $E$ , таких, что  $\text{begin}(e) \in V_1 \ \& \ \text{end}(e) \in V_2$  либо пусто, либо соединяет каждую конечную вершину  $G_1$  с каждой начальной вершиной  $G_2$  и только их.

**ОПРЕДЕЛЕНИЕ 2.** Будем называть сеть модульной, если она атомарна (состоит из единственной вершины без ребер), либо может быть разбита модульным отношением *split* на две модульных подсети.

Рассмотрим вариант логики GL5 [5], в котором следование из неатомарной формулы будет пониматься, как наличие ребер от сети из посылки импликации к сети из заключения импликации. Для соблюдения свойства модульности такие ребра должны соединять каждую конечную вершину сети из посылки импликации с каждой начальной вершиной сети из заключения импликации. Такой вариант описывается следующими правилами вывода:

- (1) Если  $A \Rightarrow B$  и  $B \Rightarrow C$ , то:  $A \Rightarrow (B \Rightarrow C)$ .
- (2) Если  $A \Rightarrow (B \Rightarrow C)$ , то:  $A \Rightarrow B$  и  $B \Rightarrow C$ .
- (3) Если  $A \Rightarrow B$  и  $B \Rightarrow C$ , то:  $(A \Rightarrow B) \Rightarrow C$ .
- (4) Если  $(A \Rightarrow B) \Rightarrow C$ , то:  $A \Rightarrow B$  и  $B \Rightarrow C$ .
- (5) Если  $A \Rightarrow B$  и  $A \Rightarrow C$ , то:  $A \Rightarrow (B \ \& \ C)$ .
- (6) Если  $A \Rightarrow (B \ \& \ C)$ , то:  $A \Rightarrow B$  и  $A \Rightarrow C$ .
- (7) Если  $A \Rightarrow C$  и  $B \Rightarrow C$ , то:  $(A \ \& \ B) \Rightarrow C$ .
- (8) Если  $(A \ \& \ B) \Rightarrow C$ , то:  $A \Rightarrow C$  и  $B \Rightarrow C$ .

Рассмотрим следующую интерпретацию GL5. Граф  $G$  будем понимать, как модель бизнес-процесса: вершина графа интерпретируется, как задача, ребро – возможность передачи потока управления от одной задачи к другой.

Тогда теория в GL5 описывает задачи, которые могут возникнуть в ходе исполнения модели бизнес-процесса, а выводы в теории – сеть задач, которая реализует конкретный экземпляр бизнес-процесса.

Немаловажным аспектом использования логического вывода для решения практических задач является вычислительная эффективность. В большинстве случаев для этого выбирается фрагмент логики, который дает приемлемую вычислительную сложность. При разработке GL5 был применен другой подход, аналогичный примененному в работах Гуревича ([6, 7]) — система правил логического вывода изначально выбирается таким образом, чтобы обеспечить разрешимость и выводимость формул за линейное время.

Поскольку естественно ожидать, что сеть задач должна соответствовать начальным условиям и ходу исполнения экземпляра бизнес-процесса, нас будут интересовать не все возможные выводы из теории, но только те, которые ведут к достижению цели бизнес-процесса кратчайшим путем в данных условиях.

Таким образом, постановка задачи на вывод в теории включает в себя:

- множество начальных вершин (одна или несколько задач, с которых начинается исполнение бизнес-процесса);
- множество промежуточных модульных подсетей (задачи или структуры задач, которые обязательно должны быть включены в вывод);
- множество конечных вершин (одна или несколько задач, по достижении которых исполнение бизнес-процесса можно считать завершенным).

## 2. Пример бизнес-процесса

Рассмотрим следующий пример бизнес-процесса: рецензирование документа перед его публикацией. Автор отправляет документ нескольким рецензентам, каждый из которых может высказать замечания к документу либо утвердить его. Когда все рецензенты утверждают документ, автоматически происходит его публикация.

Хотя это типичный и достаточно простой для понимания пример, его реализация в существующих нотациях описания бизнес-процессов достаточно нетривиальна, т.к. подразумевает параллельное исполнение нескольких однотипных веток бизнес-процесса.

Данный пример описывается теорией со следующими аксиомами:

- (9) ПодготовитьДокумент(Автор)  $\Rightarrow$  Согласовать(Рецензент, Автор)  
 (10) Согласовать(Рецензент, Автор)  $\Rightarrow$  ИсправитьДокумент(Рецензент, Автор)  
 $\Rightarrow$  Согласовать(Рецензент, Автор)

(11)  $\text{Согласовать}(\text{Рецензент}, \text{Автор}) \Rightarrow \text{ОпубликоватьДокумент}$ .

Обращаем внимание на правило (10) – если рецензент высказал замечания к документу, задачу подготовки очередной версии документа следует адресовать тому же автору, от которого пришла предыдущая версия документа. Также следует обратить внимание на отсутствие скобок в правиле (10) – оно связано с ассоциативностью импликации в GL5.

Далее, пусть нам требуется бизнес-процесс, в котором документ должен быть согласован с двумя рецензентами  $\text{Рецензент1}$  и  $\text{Рецензент2}$ . Тогда постановка задачи на вывод выглядит следующим образом:

- **Начать с:**
  - ПД(А)
- **Пройти через:**
  - $C(P1, A) \& C(P2, A)$
- **Завершить на:**
  - ОД

Сокращения:

ПД = ПодготовитьДокумент,  
 С = Согласовать,  
 ИД = ИсправитьДокумент,  
 ОД = ОпубликоватьДокумент,  
 А = Автор,  
 P1 = Рецензент1,  
 P2 = Рецензент2.

На первом шаге вывода по правилу (9) получим:

$$\text{ПД}(A) \Rightarrow (C(P1, A) \& C(P2, A)).$$

Требования «Начать с» и «Пройти через» выполнены, осталось выполнить требование «Завершить на». После первого шага каждый из рецензентов получит входящее задание на согласование. Пусть  $\text{Рецензент1}$  с документом согласился, а  $\text{Рецензент2}$  – высказал замечания. Тогда на следующем шаге по правилам (11) для  $\text{Рецензент1}$  и (10) для  $\text{Рецензент2}$  мы получим:

$$\text{ПД}(A) \Rightarrow ((C(P1, A) \Rightarrow \text{ОД}) \& (C(P2, A) \Rightarrow \text{ИД}(P2, A))).$$

Несмотря на то, что в выводе появилась вершина  $\text{ОД}$ , процесс не завершается, т.к.  $\text{ОД}$  не является единственной конечной вершиной. Далее, появилось новое входящее задание для Автора. После его выполнения будет сделан следующий шаг вывода:

$$\text{ПД}(A) \Rightarrow ((C(P1, A) \Rightarrow \text{ОД}) \& (C(P2, A) \Rightarrow \text{ИД}(P2, A) \Rightarrow C(P2, A))).$$

Пусть теперь Рецензент2 согласен с документом:

$$\text{пд}(A) \Rightarrow ((C(P1, A) \Rightarrow \text{од}) \& (C(P2, A) \Rightarrow \text{ид}(P2, A) \Rightarrow C(P2, A) \Rightarrow \text{од})) .$$

В соответствии с правилами вывода GL5 данная формула преобразуется к виду:

$$\text{пд}(A) \Rightarrow (C(P1, A) \& (C(P2, A) \Rightarrow \text{ид}(P2, A) \Rightarrow C(P2, A))) \Rightarrow \text{од} .$$

Поскольку теперь достигнута искомая конечная вершина, бизнес-процесс завершается.

## Заключение

В качестве основного преимущества исполнения бизнес-процессов с использованием логики построений на графах, помимо доказуемости порождаемых программ, можно отметить естественную семантику исполнения параллельных задач. На рассмотренном примере хорошо видно, что порождение и слияние параллельно исполняющихся веток бизнес-процесса описывается не алгоритмически заданными гейтами, как это принято в большинстве нотаций, а правилами, близкими к тому, как они содержательно понимаются человеком.

Кроме того, описания бизнес-процессов становятся переиспользуемыми – на одном и том же множестве базовых правил (теории) могут порождаться различные схемы бизнес-процессов. Например, правила рецензирования (9) и (10) могут использоваться в нескольких различных бизнес-процессах, будь то простое согласование пресс-релиза, сразу же завершающееся его публикацией, или согласование договора, в рамках которого происходит последовательное согласование документа в нескольких службах с последующей отправкой по почте контрагенту.

## Список литературы

- [1] Е. В. Кочуров. «Конструктивный синтез пользовательских интерфейсов Web-приложений», *Программные системы: теория и приложения*, 4:4(18) (2013), с. 3–25, URL: [http://psta.psisras.ru/read/psta2013\\_4\\_45-59.pdf](http://psta.psisras.ru/read/psta2013_4_45-59.pdf) <sup>↑359</sup>
- [2] Е. В. Кочуров, Н. Н. Непейвода, И. Н. Григоревский, «Замечания о логиках построений на графах и их применении», *Вторая международная научно-практическая конференция «Технические науки: теория, методология и практика»*, Сб. науч. докл. (г. Москва, 28 ноября 2014 г.), АНО Издательский дом «Научное обозрение», М., 2014, с. 8–18. <sup>↑359</sup>

- [3] BPMN Specification — Business Process Model and Notation (дата обращения: 30.11.2015), URL: <http://www.bpmn.org/> <sup>↑359</sup>
- [4] Business process management — Wikipedia, the free encyclopedia (дата обращения: 30.11.2015), URL: [https://en.wikipedia.org/wiki/Business\\_process\\_management](https://en.wikipedia.org/wiki/Business_process_management) <sup>↑359</sup>
- [5] Е. В. Кочуров, «Об одной конструктивной логике построений на графах», *Девятые Смирновские чтения по логике*, Материалы международной научной конференции (г. Москва, 17–19 июня 2015 г.), Издательство «Современные тетради», М., 2015, с. 22–24. <sup>↑361</sup>
- [6] Yu. Gurevich, I. Neeman. “DKAL: Distributed-Knowledge Authorization Language”, 21st IEEE Computer Security Foundations Symposium CSF 2008 (June 23–25, 2008, Carnegie Mellon University, Pittsburgh, USA), pp. 149–162. <sup>↑362</sup>
- [7] Yu. Gurevich, I. Neeman. “Logic of infons: The propositional case”, *ACM Transactions on Computational Logic*, **12** (2011), pp. 1–28. <sup>↑362</sup>
- Рекомендовал к публикации д.ф.-м.н. Н.Н. Непейвода

Об авторе:



**Евгений Владимирович Кочуров**

Аналитик в Интерин Технологии, аспирант Института программных систем им. А.К. Айламазяна РАН

*e-mail:*

[eugene\\_vk@mail.ru](mailto:eugene_vk@mail.ru)

Пример ссылки на эту публикацию:

Е. В. Кочуров. «Применение логики построений на графах к исполнению моделей бизнес-процессов», *Программные системы: теория и приложения*, 2015, **6:4(27)**, с. 359–366.

URL: [http://psta.psisras.ru/read/psta2015\\_4\\_359-366.pdf](http://psta.psisras.ru/read/psta2015_4_359-366.pdf)

E. Kochurov. *Logic of constructions on graphs within application to business process execution*.

ABSTRACT. This article introduces the approach to the execution of business processes based on the logic synthesis of programs for business process models. The approach uses logic of constructions on graphs GL5. (*In Russian*).

*Key Words and Phrases:* constructive logic, constrictions on graphs, business process modeling..

### References

- [1] Ye. V. Kochurov. “Constructive synthesis of Web-based application user interfaces”, *Programmnyye sistemy: teoriya i prilozheniya*, 4:4(18) (2013), pp. 3–25 (in Russian), URL: [http://psta.psir.ru/read/psta2013\\_4\\_45-59.pdf](http://psta.psir.ru/read/psta2013_4_45-59.pdf)
- [2] Ye. V. Kochurov, N. N. Nepeyvoda, I. N. Grigorevskiy, “Remarks on logics of constructions on graphs and their applications”, *Vtoraya mezhdunarodnaya nauchno-prakticheskaya konferentsiya “Tekhnicheskkiye nauki: teoriya, metodologiya i praktika”*, Sb. nauch. dokl. (g. Moskva, 28 noyabrya 2014 g.), ANO Izdatel’skiy dom “Nauchnoye obozreniye”, M., 2014, pp. 8–18 (in Russian).
- [3] BPMN Specification — Business Process Model and Notation (data obrashcheniya: 30.11.2015), URL: <http://www.bpmn.org/>
- [4] Business process management — Wikipedia, the free encyclopedia (data obrashcheniya: 30.11.2015), URL: [https://en.wikipedia.org/wiki/Business\\_process\\_management](https://en.wikipedia.org/wiki/Business_process_management)
- [5] Ye. V. Kochurov, “About one constructive logic of graph”, *Devyatyye Smirnovskiyey chteniya po logike*, Materialy mezhdunarodnoy nauchnoy konferentsii (g. Moskva, 17–19 iyunya 2015 g.), Izdatel’stvo “Sovremennyye tetradi”, M., 2015, pp. 22–24 (in Russian).
- [6] Yu. Gurevich, I. Neeman. “DKAL: Distributed-Knowledge Authorization Language”, 21st IEEE Computer Security Foundations Symposium CSF 2008 (June 23–25, 2008, Carnegie Mellon University, Pittsburgh, USA), pp. 149–162. “Logic of infons: The propositional case”, *ACM Transactions on Computational Logic*, 12 (2011), pp. 1–28.

### Sample citation of this publication:

E. Kochurov. “Logic of constructions on graphs within application to business process execution”, *Program systems: theory and applications*, 2015, 6:4(27), pp. 359–366. (*In Russian*).

URL: [http://psta.psir.ru/read/psta2015\\_4\\_359-366.pdf](http://psta.psir.ru/read/psta2015_4_359-366.pdf)