

УДК 004

## ОБЗОР СИСТЕМ УПРАВЛЕНИЯ БИЗНЕС-ПРАВИЛАМИ

Молдаш Г.Р., Аубакиров С.С.

**Аннотация.** Активное развитие информационных технологий требует использования эффективной системы для ведения, поддержки и исполнения бизнес-правил компании. Развитие методологии управления бизнес-правилами сегодня рассматривается как основа парадигмы процессного подхода к управлению в целом. Появление новых цифровых технологий стало тем инфраструктурным фактором, который позволил заново оценить возможности и необходимость построения системы управления бизнес-правилами предприятия. На сегодняшний день существует множество систем для управления бизнес-правилами, мы рассмотрим основные из них. В статье предложена реализация правил в программе Eclipse с использованием Drools-rules и дерева решений Rete.

**Ключевые слова:** система управления бизнес правилами, Drools-правила, дерево решений Rete, программное обеспечение Eclipse.

## OVERVIEW OF BUSINESS RULE MANAGEMENT SYSTEMS

G.R. Moldash, S.S. Aubakirov

**Annotation.** Active development of information technology requires the use of an effective system for maintaining, supporting and enforcing company business rules. The development of a business rule management methodology today is considered as the basis of the paradigm of the process approach to management as a whole. The emergence of new digital technologies has become the infrastructural factor that made it possible to re-evaluate the capabilities and the need to build an enterprise business rule management system. Today, there are many systems for

managing business rules, we will consider the main ones. The article suggests the implementation of rules in the Eclipse program using Drools-rules and the Rete decision tree.

**Keywords:** Business Rules Management Systems, Drools Rules, Rete decision tree, Eclipse software.

Управление бизнес-правилами – это технология создания, поддержания и размещения сервисов принятия решений на основе правил. Системы управления бизнес-правилами позволяют разработчикам и бизнес-пользователям быстро и легко разрабатывать правила, ориентированные на принятие решений. Например, для банка может работать правило:

**If** перевод осуществляется за пределы банка  
**then** рассчитать комиссию в размере 1,5%.

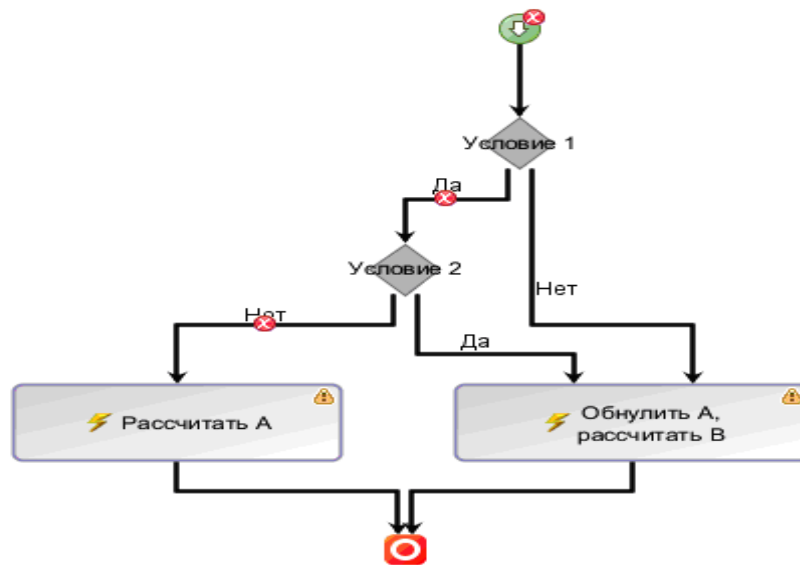
Для мобильных операторов, может быть:

**If** клиент использует тарифный план «Вечерний»  
**and** звонок после 17:00  
**then** рассчитать стоимость звонка 0.05 за минуту.

Для реализации этих правил используются BRMS (Business Rules Management Systems – системы управления бизнес-правилами).

BRMS описывают логику принятия решений в виде системы бизнес-правил. Бизнес-правила указывают на выполнение некоторых действий в случае выполнения определенных условий. BRMS позволяют менеджерам вынести решение вместе с заказчиками, которое будет удовлетворять обе стороны и соответствовать бизнес-логике предприятия. Доступность правил позволяет бизнес-пользователям быстро выполнять изменения, не вовлекая ИТ специалистов [1].

В общем виде бизнес-правило представляется в форме утверждения вида: «Если (условия), то (список действий), иначе (альтернативный список действий)» и сразу же сохраняется в виде кода, как показано на рисунке 1.



```
<?xml version="1.0" encoding="UTF-8"?>
<ilog.rules.studio.model.ruleflow:RuleFlow xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
  <name>пример</name>
  <uuid>df727624-73d7-45aa-be9d-d47c0b917c63</uuid>
  <locale>ru_RU</locale>
  <categories>any</categories>
  <rfModel>
  <Ruleflow xmlns="http://schemas.ilog.com/Rules/7.0/Ruleflow">
  <Body>
  <TaskList>
  <StartTask Identifier="task_0"/>
  <ActionTask Identifier="Обнулить А, рассчитать В">
  <Actions Language="bal"><![CDATA[Сделать А равным 0;]]&#13;<![CDATA[
  Сделать В равным a1;]]></Actions>
  </ActionTask>
  <ActionTask Identifier="Рассчитать А">
  <Actions Language="bal"><![CDATA[Сделать А равным a1 + a2;]]></Actions>
  </ActionTask>
  <StopTask Identifier="task_2"/>
  </TaskList>
  <NodeList>
  <TaskNode Identifier="node_0" Task="task_0"/>
  <BranchNode Identifier="node_1"/>
  <BranchNode Identifier="node_2"/>
  <TaskNode Identifier="node_3" Task="Обнулить А, рассчитать В"/>
  <TaskNode Identifier="node_4" Task="task_2"/>
  <TaskNode Identifier="node_5" Task="Рассчитать А"/>
  </NodeList>
  <TransitionList>
  <Transition Identifier="transition_0" Source="node_0" Target="node_1"/>
  <Transition Identifier="transition_1" Source="node_1" Target="node_3"/>
  <Transition Identifier="transition_2" Source="node_1" Target="node_2">
  <Conditions Language="bal"><![CDATA[Значение для Объект1 равен X ]]></Conditions>
  </Transition>
  <Transition Identifier="transition_3" Source="node_2" Target="node_3"/>
  <Transition Identifier="transition_4" Source="node_3" Target="node_4"/>
  <Transition Identifier="transition_5" Source="node_2" Target="node_5">
  <Conditions Language="bal"><![CDATA[Значение для Объект2 меньше Y]]></Conditions>
  </Transition>
  </TransitionList>
  </Ruleflow>
  </Body>
  </Ruleflow>
  </rfModel>
  </categories>
  </locale>
  </uuid>
  </name>
  </xmi:version>
  </xmlns:xmi>
  </RuleFlow>
```

Рис. 1. Блок-схема и код бизнес-правил

Доступность правил позволяет бизнес-пользователям быстро выполнять изменения. Результатом становится повышение гибкости бизнеса, быстрое реагирование на динамику рынка и предоставление клиентам новых возможностей [2].

Основными преимуществами использования BRMS являются:

- системы позволяет легко и гибко автоматизировать логику принятия решений в рамках бизнес-процессов компании;
- повышается прозрачность логики принятия решений в компании;
- снижается зависимость от программистов для внесения изменений в работу информационных систем;
- увеличивается контроль реализованной бизнес-логики;
- быстро реагируют на динамику рынка и предоставляет клиентам новых возможностей;
- в BRMS хранится вся история изменения бизнес-логики.

Возможности BRMS:

- управление задачами сотрудников (возможность объединять отдельные задачи в бизнес-процессы, управлять переходами от одной задачи к другой, переназначать задачи и назначить их на группы, функциональные подразделения);
- возможность оперативно отслеживать ход исполнения задач;
- возможность подключения к системе партнеров (например, коллекторских агентств, партнеров по оценке и т. д.) и полный контроль за их деятельностью со стороны сотрудников компании (сотрудники видят на какой стадии находится процесс и задачи партнеров).

К компонентам BRMS относится (рис. 2):

- Сервер исполнения бизнес-правил;

- инструмент ведения правил (репозитория), предоставляющего бизнес-пользователям удобный и функциональный интерфейс для хранения, создания и изменения бизнес логики.

Также присутствуют компоненты тестирования и анализа правил [3].

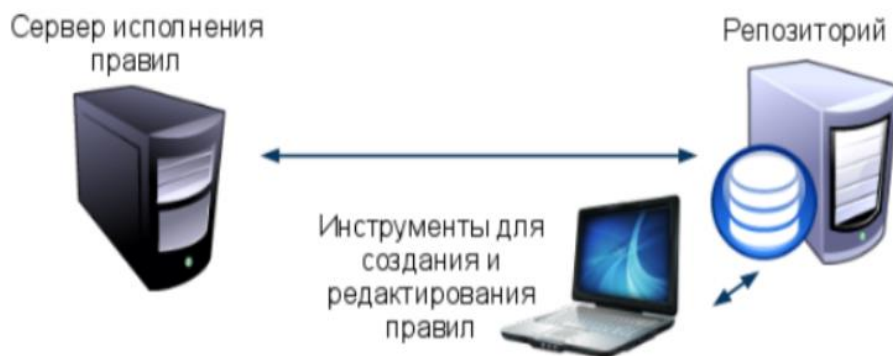


Рис. 2. Компоненты BRMS

Использование систем управления бизнес-правилами является важной частью системы управления принятием решений в организации. Топ BRMS 2019 года показано в Таблице 1.

Таблица 1. Топ BRMS системы 2019 года

№	Название	Документация/ Исходный код	Интерфейс/ Юзабилити	Стоимость
1	Drools	+	+	бесплатно
2	Red Hat News	-	+	платно
3	IBM	-	-	бесплатно
4	FICO	-	+	бесплатно
5	Agiloft	-	-	платно
6	Bosch	-	+	бесплатно
7	Sas	-	+	бесплатно

Данный список охватывает основных поставщиков BRMS-решений [4]. Исходя из анализа, мы поработали и протестировали BRMS систему Drools и запустили в Eclipse.

Drools – это процессор правил с открытым кодом, написанный на Java и выполняющий правила в соответствии с алгоритмом Rete [5].

Алгоритм Rete функционирует как сеть, которая предназначена для хранения большого объема информации и обеспечивает значительное сокращение времени отклика. Rete-алгоритм обеспечивает повышение быстродействия, что при изменении действия мгновенно знает, какие правила нужно применить [6].

Основные характеристика алгоритма Rete:

- уменьшает или устраняет определенные типы избыточности за счет использования совместного использования узлов;
- сохраняет частичные совпадения при выполнении объединений между различными типами фактов и работает очень быстро.

Единственным недостатком алгоритма Rete является большой объем памяти.

Сеть Rete состоит из альфа- и бета-подсетей. Левая сторона (альфа) формирует дискриминационную сеть, отвечающую за отбор элементов рабочей памяти (альфа-памяти), основанный на простых условиях сравнения атрибутов с фактами. Если элемент рабочей памяти успешно сопоставлен с условиями правил, он передается на следующий узел сети. Следовательно, все элементы рабочей памяти, представляющие одну сущность, пересекают данную ветвь узлов дискриминационной сети. Правая (бета) сторона графа выполняет объединение различных элементов рабочей памяти и записывает результат в бета-память [7]. Пример дерева решений показан на рисунке 3.

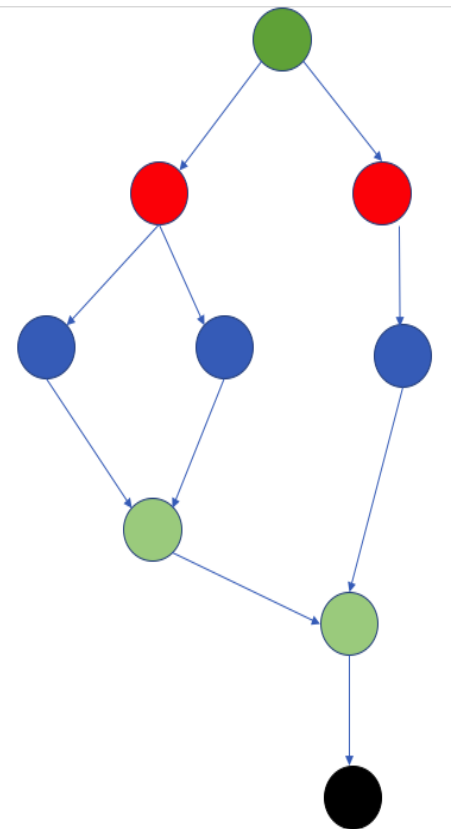
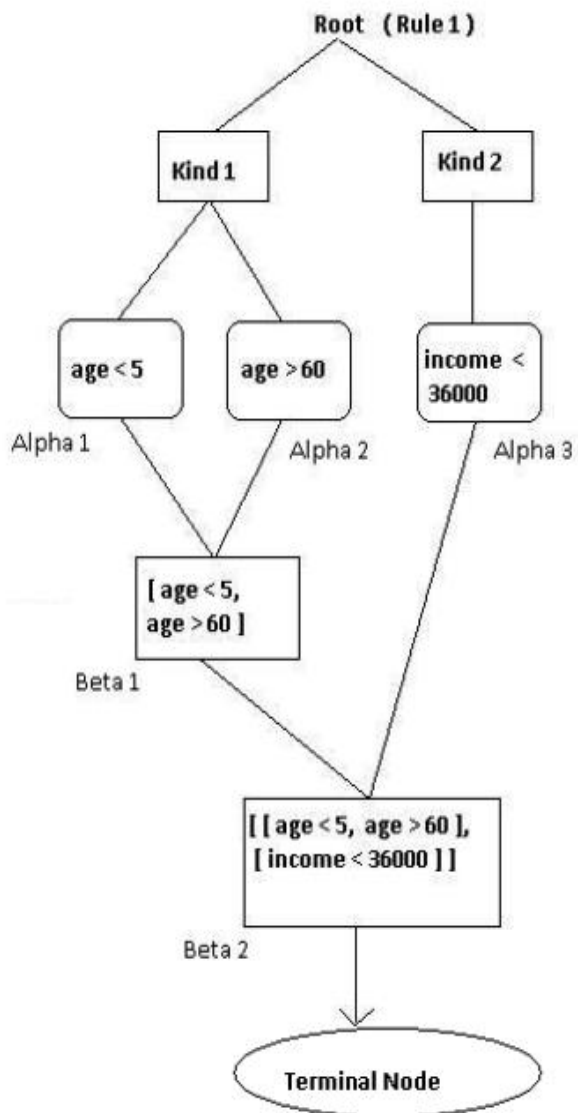


Рис. 3. Дерево решений Rete [8]

В примере дерево решений Rete:

1. Зеленый – точка входа.
2. Красный – название класса.
3. Синий – условия по отдельности (альфа узел).
4. Зеленый – бета-узел (используется для объединения нескольких условий).
5. Черный – выход (ответ).

Ниже приведены правила, которые мы использовали для тестирования в Eclipse.

Правила:

1. Если платеж равен или превышает 7 000 000 тенге, взять в контроль.
2. Если платеж 7 000 000 тенге дроблен на 2 в течение месяца, взять в контроль.
3. Если число транзакций у пользователя превышает среднее число транзакций среди пользователей, необходимо взять в контроль.
4. Если переводы от одного физического лица нескольким при отсутствии явных признаков родственных связей между отправителем и получателями, взять в контроль.
5. Если переводы от нескольких физических лиц одному при отсутствии явных признаков родственных связей между отправителями и получателем, взять в контроль.
6. Если в течение 6 часов было совершено больше 3-х транзакции на один счет, взять в контроль.

Мы написали наши правила в Drools-rules и вытащили дерево решения Rete, как показано на рисунке 4.



```
DroolsTest.java Sample.drl
8
9 rule "PaymentMore"
10 when
11     p : Payment(getPayment() >= 7000000)
12 then
13     System.out.println( "Platej raven ili prevuwaet 7 000 000 tenge" );
14 end
15
16
17 rule "SplittingUpPayment"
18 when
19     p : SplittingUpPayment(getPayment() >= 7000000)
20     aop : SplittingUpPayment(getAmountOfPayments() > 1)
21 then
22     System.out.println( "Droblenie 7 000 000 v techenie mecyasa dvuh" );
23 end
24
25
26 rule "NoRelativeConnection"
27 when
28     nrc : NoRelativeConnection(getRelativeConnection())
29     aos : NoRelativeConnection(getAmountOfSenders() == 1)
30     aor : NoRelativeConnection(getAmountOfRecipients() > 1)
31 then
32     System.out.println( "Perevodu ot odnogo phisicheskogo lica neskolkim pri otsutstv:
33 end
34
35 rule "AmountOfTransactions"
36 when
37     aotc : AmountOfTransactions(getAvgAmountOfTransactions() < getAmountOfTransactor
38 then
39     System.out.println("Chislo tranzaksii u polzovatelya prevuwaet srednee chislo tra
40 end
41
42 rule "FrequencyOfTransactions"
43 when
44     aot : FrequencyOfTransactions(amountOfTransactions > 3)
45     aor : FrequencyOfTransactions(amountOfRecipients == 1)
46     t : FrequencyOfTransactions(time == "1 chas")
47 then
48     System.out.println("6 chas bolwe 3 tranzaksii otpravlyaiut na 1 s4et" );
49 end
```

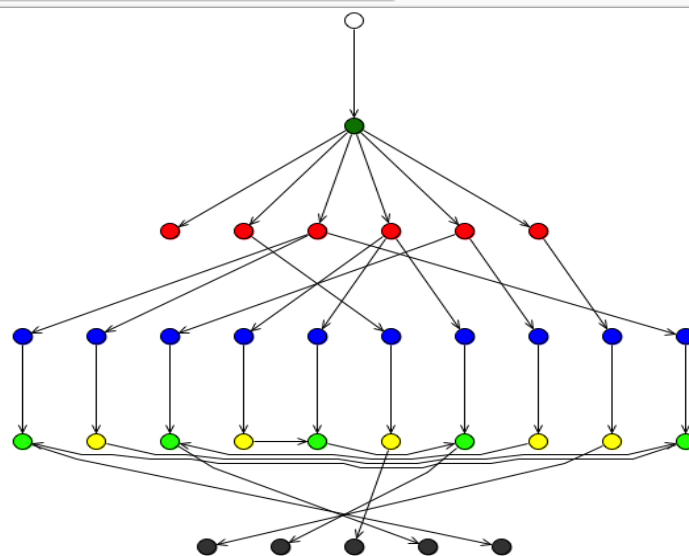


Рис. 4. Пример кода на Eclipse с использованием Drools

В нашем дереве решений Rete:

1. Зеленый – точка входа.
2. Красный – название класса.
3. Синий – условия по отдельности.
4. Желтый – альфа-узел (альфа-память).
5. Зеленый – бета-узел (используется для объединения нескольких условий).
6. Черный – выход (ответ).

BRMS позволяют легко и гибко автоматизировать логику принятия решений в рамках бизнес-процессов компании. Ценность BRMS заключается в том, что на их основе можно создавать эффективные и воспроизводимые решения, которые работают быстрее и требуют меньшего уровня технической поддержки [9].

### **Заключение.**

Системы управления бизнес-правилами, как средства моделирования, выполнения и проектирования оперативных решений, позволяют автоматизировать логику принятия решений в организациях сферы предоставления услуг. Данные системы успешно применяются в различных секторах экономики, в том числе и на рынке предоставления услуг связи населению.

### **Список литературы**

1. Романова Ю., Муzychкин П. Информационные технологии в менеджменте / Ю. Романова, П. Муzychкин. – М.: Юрайт, 2019. – 478 с.
2. Плаунов С. Репозитории правил для бизнеса и соответствующих блоков кода, которые встраиваются в продакшн, 2017 год 21 февраля. [Электронный ресурс]. – URL: <https://habr.com/ru/company/croc/blog/322274/> (дата обращения: 31.10.19).
3. Кулак А. Обзор IBM WebSphere ILOG JRules, 2011 год 16 ноября. [Электронный ресурс]. – URL:

<https://www.ibm.com/ru/events/presentations/11kz/Kulak%20ILOG%20JRules.pdf> (дата обращения: 01.10.2019).

4. Business Rules Management Systems (BRMS) 2019 год 20 августа. [Электронный ресурс]. – URL: <https://www.trustradius.com/contract-management> (дата обращения: 01.10.2019).

5. Н. Алекс Рапп, TheServerSide.com. "Реализация бизнес-логики: Введение в проект Drools" – май 2004 г. [Электронный ресурс]. – URL: <https://www.theserverside.com/news/1365158/An-Introduction-to-the-Drools-Project> (дата обращения: 01.10.2019).

6. Rete algorithm – 2009 год апрель. [Электронный ресурс]. – URL: [https://subscription.packtpub.com/book/application\\_development/9781847196064](https://subscription.packtpub.com/book/application_development/9781847196064) (дата обращения: 01.10.2019).

7. Бессмертный И.А., Платонов А.В. Интеллектуальные системы / И.А. Бессмертный, А.В. Платонов – М.: Юрайт, 2018. – 243 с.

8. Introduction to Rete algorithm, 29 July, 2013. [Электронный ресурс]. – URL: <https://webcache.googleusercontent.com/> (дата обращения: 01.10.2019).

9. Ian Graham. Business Rules Management and Service oriented architecture / Ian Graham – Wiley, 2007. – 292 p.

#### **Сведения об авторах**

Молдаш Гүлім Русланқызы – докторант 2 курса, Satpayev University; Казахстан, г. Алматы.

Аубакиров Санжар Серикович – PhD, доцент кафедры «Бизнес Информатики», Университет Международного Бизнеса; Казахстан, г. Алматы.

#### **Information about authors**

Moldash Gulim Ruslankyzy – 2nd year doctoral student, Satpayev University; Kazakhstan, Almaty.

Aubakirov Sanzhar Serikovich – PhD, Associate Professor, Department of Business Informatics, University of International Business; Kazakhstan, Almaty.