# Совершенствование моделей проблемно-ориентированных пакетов

**Краткий результат главы.** Проведено совершенствование структуры знаний инструментальных средств анализа исходного кода Mothur и синтеза модулей визуального проектирования Rapidminer Studio, представляющих собой процедуры анализа данных NGS в виде визуальных функциональных блоков. Основная проблема, в решении которой произошло продвижение, - это более глубокий анализ кода с целью выявления процедур Mothur, отвечающих за структуру выходных данных. Результаты анализа позволяют сокращать количество вариантов использования визуальных блоков в модели NGS, что существенно упрощает работу исследователю. Интерфейс пользователя Rapidminer Studio настраивается в соответствии со структурой решаемой задачи, он становится адаптивным к решаемой задаче (в некоторой степени интеллектуальным). Усовершенствованы так же правила синтеза кода визуальных блоков для новых версий Mothur.

Актуальность автоматического анализа исходного кода Mothur и синтеза соответствующих структур модулей Rapidminer Studio (трансформации) определяется, во-первых, количеством модулей, которые необходимо представить в виде визуальных блоков, – 147 (на 2019 год), а, во-вторых, необходимостью поддерживать соответствие сгенерированных блоков новым версиям модулей Mothur, т.к. система находится в фазе активного развития (в 2018 году модулей было 143). Кроме того, наличие такой модели позволяет на следующих этапах проекта разрабатывать процедуры автоматизации вычислительного процесса на кластерных вычислительных системах.

Серия экспериментов с полученными на этапе 2018 моделями метод-ориентированного программного пакета Mothur показала, что сгенерированные модули Rapidminer Studio хотя и применимы для визуализации процесса исследования, но не вполне удобны исследователю, так как требуют знания внутренней структуры отображения комбинации входных параметров и параметров модулей: функции некоторых модулей масштабны, и могут выдавать десятки различных файлов на выходе. В результате анализа применимости (user experience) показано, что выявленные недостатки являются существенно мешающими внедрению разрабатываемой технологии в процессе исследований и принято решение о совершенствовании процедуры трансформации с целью выполнить следующие требования:

* перечень выходных разъемов должен формироваться динамически в зависимости от комбинации входных подключений и заданных параметров модулей;
* процедура трансформации должна поддерживать будущие версии пакета Mothur.

Для обеспечения обоих требований принято решение а) использовать максимум полезной информации из исходного кода Mothur автоматически, и б) недостающие элементы модели представлять при помощи дополнительных высказываний в стандартном для инструмента формате RDF (Resource Description Framework). Реализация принятых решений начата с совершенствования архитектуры системы трансформации интерфейсов модулей Mothur в программные модули Rapidminer Studio.

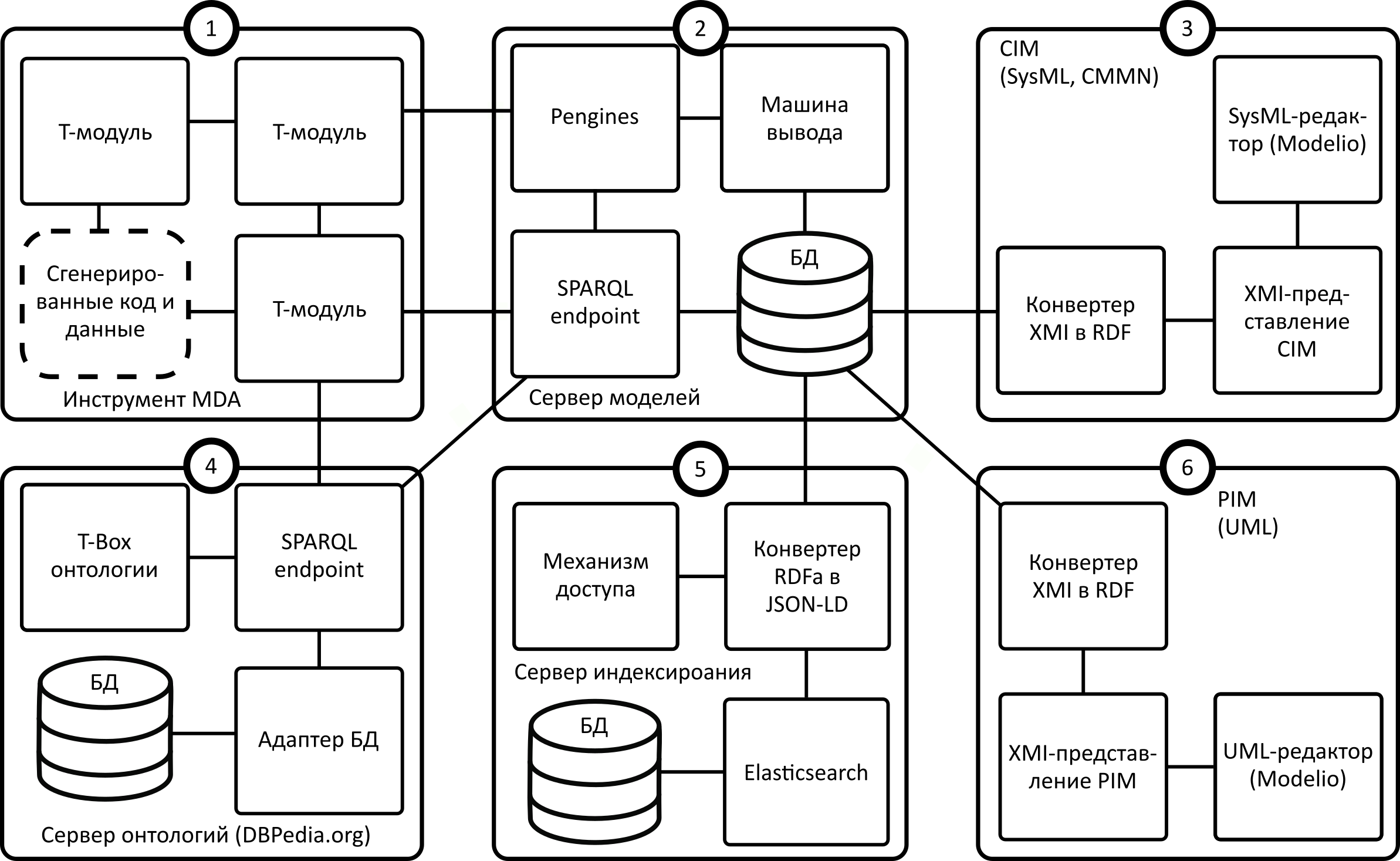


Рисунок 1 - Архитектура средств анализа интерфейса Mothur  
и синтеза модулей Rapidminer Studio

В архитектуру средств трансформации (рисунок 1) добавлены блоки 3 и 4, блок анализа исходного кода обозначен блоком 4. Теперь архитектура инструмента трансформации включает следующие основные компоненты:

* сервер исходных моделей (2), который хранит модель модуля Mothur в виде RDF-троек (технология Семантического Веба);
* механизм трансформации (1), который запрашивает данные с сервера моделей и других SPARQL-совместимых серверов (4);
* конвертеры форматов представления моделей CIM (3), PIM и исходного кода (6);
* сервис полнотекстового индекса (5).

При помощи блока 3 теперь доступны данные данных из Семантического Веба, раздела интернета, где предоставляемая информация размечена аналогично используемому формату RDF. Например, сервер Dbpedia.org представляет сущности, опубликованные в Wikipedia, т. е. практически всю существующую в естественном языке терминологию. Данные, полученные через этот блок на следующих этапах проекта будут составлять основу интегрирования сервисов публикации данных в Семантический Веб.

Блок 4 на следующих этапах проекта будет использован для импорта описания структуры хранилища генетической информации, представленной в виде структур языка JHipster. Тексты данного языка транслируются в UML, которые уже конвертерами преобразуются в RDF. В результате будет достижима интеграция Rapidminer Studio и хранилища данных ЛИН СО РАН при помощи микросервисов.

Процесс отображения модулей Mothur в виде блоков диаграмм Rapidminer Studio начинается с этапа анализа исходного кода Mothur, осуществляющийся при помощи алгоритмов, применяющих регулярные выражения для распознавания элементов структуры классов С++, задающих модули Mothur. Результаты анализа представляются в виде структур RDF. Пример представления результата анализа модуля представлен в отчете 2018 года. Данный механизм дополнен анализом программного кода методов C++, реализующих формирование перечня выходных файлов и структуры их поименования.

Вторым, заключительным, этапом является синтез программного кода модулей Rapidminer. Синтез осуществляется при помощи логической объектно-ориентированной программы на языке LogTalk. Объекты и сценарии программы анализируют множество фактов о структуре модулей Mothur и синтезируют программные конструкции соответствующих модулей Rapidminer Studio. На этапе 2018 года разработана методика построения процедуры трансформации, а также объекты, которые в общем виде реализуют указанное преобразование. На данном этапе осуществлялось совершенствование методики в направлении выполнения выработанных требований удобства использования результата.

Например, в модуле sub.sample 11 входных разъемов и 11 выходных, в модуле collect.shared — один входной и 41 выходной разъем, из которых, в большинстве случаев, используется одиннадцать. На схеме Rapidminer Studio такие блоки выглядят длинными вертикальными блоками с большим количеством разъемов, среди которых надо найти подходящий, что, очевидно, не удобно. Перечень выходных разъемов в collect.shared определяется параметром calc, который указывает, какие модули-калькуляторы следует применить к данным входного файла. Для каждого калькулятора генерируется файл с соответствующим расширением. В схеме Rapidminer Studio для каждого выходного файла collect.shared надо предусмотреть соответствующий разъем.

Информация о том, какие возможны выходные разъемы содержится в методе getOutputPattern(string type) каждого модуля Mothur, но какая конкретно комбинация будет иметь место, в общем случае, зависит от входных подключений и параметрами модуля, задаваемым пользователем. Исследование алгоритмов модулей Mothur, имеющих больше шести выходных разъемов, показало, что во всех случаях комбинация зависит от одного параметра, причем, как правило, - это параметр с возможностью выбора комбинации выходных файлов, и водном случае - выход строго один, но меняется формат выходного файла.

Результат анализа представлен в виде TTL-текста, которое дополняет описание структур модулей, полученных трансформацией из исходного кода:

mothur:collect.shared-calc-parameter ngsp:definesOutputPortSet true ;

ngsp:outputPortSet "multi" .

mothur:collect.single-calc-parameter ngsp:definesOutputPortSet true ;

ngsp:outputPortSet "multi" .

mothur:rarefaction.single-calc-parameter ngsp:definesOutputPortSet true ;

ngsp:outputPortSet "multi" .

mothur:remove.seqs-format-parameter ngsp:definesOutputPortSet true ;

ngsp:outputPortSet "single" .

Содержательно данные описания представляют новую информацию процедуре трансформации, определяющую перечень модулей с вышеуказанными особенностями, а также какой атрибут и в каком виде формирует перечень выходных разъемов визуального блока.

Таким образом, среди «особенных» оказались четыре модуля. Представление в Rapidminer Studio для этих модулей должно отличаться от остальных наличием метода, который по факту изменения пользователем параметра должен перестраивать комбинацию выходных разъемов. В Rapidminer Studio это осуществляется при помощи механизма дополнения блока новыми разъемами, состоящего из объекта, реализующего интерфейс putPortExtender управления перечнем разъемов и стратегии изменения этого перечня - виртуального метода, аргументами которого являются перечень входных разъемов и структуры самого блока.

Существующая база знаний процедуры трансформации снабжена знаниями, которые рассматривают особые случаи, и ставят ссылки на соответствующие методы, реализованные в наследуемом всеми операторами Mothur классом MothurOperator. На рисунке 2 представлен результат реализации модуля collect.shared, где определено три выходных разъема.

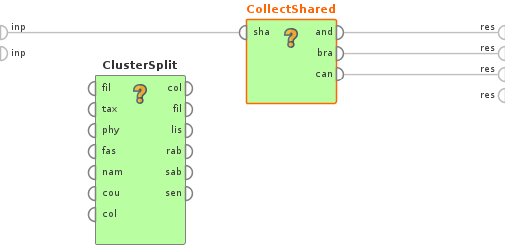


Рисунок 13 – Пример представления блока collect.shared подсчета сводных характеристик входных данных о повторяемости OTU в группах.

Другим направлением совершенствования системы знаний процедуры трансформации являются разработка знаний для преобразования данных о классификации модулей Mothur в классификации модулей Rapidminer Studio, что позволило пользователю выбирать модули из иерархического, а не линейного списка. Также усовершенствованы процедуры порождения элементов управления, используемых для установки параметров модулей.

Таким образом, в результате разработки данного направления в 2019 году усовершенствована система знаний, при помощи которой по результатам анализа исходного кода Mothur осуществляется порождение исходного кода Java-модулей для системы визуального проектирования вычислительных процессов Rapidminer Studio. Усовершенствования позволили повысить удобство использования визуального программирования для пользователя (user experience).