

УДК 004.43 ББК 22.18

## РЕФАЛ-СЕРВЕР

**А. А. Гусев**

ЗАО «СайнАрт», г. Санкт-Петербург

a.gusev@signart.ru

### **Аннотация**

Работа посвящена описанию проекта обновления и распространения языка программирования Рефал (далее – просто Рефал), созданного в СССР в 1960-х годах В.Ф. Турчиным. Язык изначально предназначался для различных логических преобразований, прежде всего, текстового материала и ориентирован на использование непрограммистами. На практике сфера применения оказалась шире: машинный перевод, оптимизация и компиляция программ, доказательство теорем, моделирование сложных электронных схем, решение ряда задач искусственного интеллекта. Язык сейчас имеет достаточное количество последователей, главным образом, в научных кругах.

Задачей описываемого проекта является создание продукта, позволяющего использовать Рефал в современных массовых приложениях и расширить круг его потенциальных пользователей до всего интернета. Был проведён опрос сообщества пользователей и разработчиков Рефала с целью получения представления о текущем состоянии дел, актуальных реализациях и путях развития языка. Были рассмотрены возможные средства реализации проекта. Информации о ведущихся аналогичных разработках получено не было.

**Ключевые слова:** *Рефал, сервер, обработка текстов, xml, json, искусственный интеллект, метавычисления*

### **1. ИСТОРИЯ ВОПРОСА**

Язык Рефал создавался для облегчения труда по обработке различной символьной информации. Он относится к классу сентенциальных языков, что не предполагает последовательного описывания шагов, которые должен сделать компьютер для решения задачи. Задача формулируется пользователем как

---

набор правил, а пути решения определяет уже компьютер, точнее, исполняющая среда Рефала<sup>1</sup>.

По мере развития средств исполнения Рефал-программ оказалось, что простота и, вместе с тем, большая выразительность языка позволяют решать весьма разнообразные и даже неожиданные на первый взгляд задачи, такие, например, как вывод химических формул.

В течение первых 10 лет после создания языка компиляторы с Рефала были созданы для основных массово используемых на то время в СССР компьютеров, и это были как раз серверные решения. Достаточное финансирование науки предоставляло такую возможность. Появились также различные интерпретации языка, развивающие его выразительные и вычислительные возможности.

В силу ряда обстоятельств, скорее всего, по причине массового перехода пользователей в научной сфере на персональные вычисления в 1990-х годах, все известные реализации языка в настоящее время также ориентированы на персональные вычисления.

С одной стороны, это позволило сильно удешевить развитие компиляторов, с другой — сузило круг задач, решаемых с помощью Рефала. Массовое обслуживание, в частности, в настоящее время немыслимо без серверных приложений. И только серверных — в части применяемой бизнес-логики.

Серверная реализация позволяет не только организовывать массовую удалённую обработку информации, но и эффективно описывать и исполнять бизнес-логику сложных человеко-машинных комплексов, сохраняя полный централизованный контроль над исполняемыми правилами.

## **2. ПОЧЕМУ РЕФАЛ**

Выбор конкретного языка в данном случае основан на личном опыте использования различных языков и сред программирования в течение последних 40 лет. По мере развития технологий труд программиста из занятия таинственного и романтического постепенно превращается в рутинный процесс, где ценностью объявляется исключительно конечный результат. А удовольствие, предполагаемая, должны приносить личные доходы от трудовой деятельности либо

---

<sup>1</sup> <http://refal.botik.ru/>

победы над конкурентами. При этом программы из произведений искусства, которыми они были на заре всеобщей компьютеризации, постепенно превращаются в нагромождение заплат, стилей и технологий, несмотря ни на какие усилия руководителей процесса.

Такое развитие событий вызвало у меня желание сделать что-то для языка, давно известного, снискавшего множество похвал, но почему-то за свою 50-летнюю историю не сильно расширившего круг своих адептов.

Вот его сильные стороны:

– абсолютная простота основной вычислительной концепции, в основе которой лежат принцип *конкретизации* и логика построения программ, соответствующая способу человеческого мышления (в отличие от многих распространённых языков программирования);

– относительная лёгкость вхождения в вопрос по причине как прозрачности концепции, так и достаточно проработанной методики его компиляции и выполнения, есть достаточно исследовательских работ как непосредственно по Рефалу, так и по близким по духу системам программирования;

– сильная универсальность языка: от обработки различных текстовых данных до задач имитационного моделирования и ИИ;

– решения на Рефале красивы и способствуют развитию ума, а не его усталости (сугубо личное мнение, по мнению автора, этот пункт можно было бы поставить первым).

В случае успешной реализации полученный продукт должен предоставить:

- идентичный доступ как к разработке, так и к использованию программ на Рефале с любого устройства, в т. ч. смартфонов;

- платформу для построения приложений, в том числе, коммерческих на Рефале и с использованием Рефала;

- платформу для создания Рефал-компонентов для встраивания в программы на других языках как с помощью технологий распределённых объектов (СОМ и т. п.), так и путём создания блоков кода на целевых языках, эмулирующих выполнение Рефал-программ в исполняющей среде целевого языка;

- повсеместно доступную базу для обучения начинающих использовать язык, поскольку без критической массы программистов, владеющих пред-

метом, любой язык мертв.

### **3. РЕФАЛ И HASKELL**

Первоначально, при формировании идеи разработки, вопрос в наличии «конкурирующего» языка программирования был мною упущен. Сравнение Рефала с «классическими» языками с разъяснением отличий было проведено еще В.Ф. Турчиным. Однако в 1990-х годах возник и получил развитие новый язык, имеющий с Рефалом гораздо больше пересечений — Хаскел (Haskell)<sup>2</sup>.

Речь идёт о сопоставлении с образцом — основе Рефала и одной из сторон Хаскела. С точки зрения математической, алгоритмы реализации данного элемента языка могут быть схожими, кроме того, в Хаскел введено понятие «ленивых» или «отложенных» (Lazy) вычислений, отсутствующих в каноническом Рефале и способных поднять производительность вычислений, то есть весьма полезных.

В результате возникает вопрос: почему бы не использовать более «свежий», поддерживающий аналогичный стиль программирования и уже сильно «продвинутый» и известный в мировом сообществе язык, то есть Хаскел вместо Рефала для тех же задач?

Но, думаю, первоначальный выбор не был ошибочным. У Рефала есть неоспоримое преимущество — простота концепции языка, созданного для непрограммистов, в первую очередь. Вся суть его идеи объясняется на практике за пару лекций и не требует запоминания множества правил и шаблонов. Чего категорически нельзя сказать о Хаскел.

Пользователь Рефала сам творит свою систему понятий и занимается, прежде всего, своей задачей. Поэтому для успешного применения Рефала в любой области достаточно хорошего воображения и понимания цели, которую следует сформулировать. Это даёт шанс привлечь на сторону языка новых пользователей из различных областей науки, и не только науки.

---

<sup>2</sup> <https://www.haskell.org/>

## **ТЕКУЩЕЕ ПОЛОЖЕНИЕ ДЕЛ В РЕФАЛ-СООБЩЕСТВЕ**

Завершение коммунистического периода совпало с появлением доступных персональных вычислений. Одновременно с этим начался процесс сокращения затрат на науку, её децентрализации. К счастью, сохранились преданные пользователи языка, которые смогли сохранить и развить образовавшиеся к тому времени диалекты, переведя разработку и сопровождение своих локальных систем на персональную вычислительную технику. Это позволяет упростить и удешевить разработку, но затрудняет использование полученных программных средств в удалении от разработчика. Вместо решения конкретных задач удалённому пользователю нужно выбрать один из не вполне совместимых диалектов, найти соответствующие дистрибутив, документацию. В то же время изначально язык ориентировался на неискушённых в техническом смысле пользователей, т. е. таким образом теряется значительная часть потенциальной аудитории языка.

Надо отдать должное сообществу Рефала: его члены смогли наладить устойчивый обмен информацией и не опуститься до своих узкодиалектных интересов. Также ведётся постоянное ознакомление студентов с Рефалом — теми, кто связан с высшей школой.

Но при этом степень распространения Рефала вне этого круга нельзя назвать удовлетворительной, потому что язык перешел в качество «клубного интереса». Более того, многие направления эффективного применения языка заполняются средствами, явно проигрывающими ему в технологичности и скорости разработки. Как пример можно привести создание систем машинного перевода, различных ботов (класса Алисы от Яндекс), систем обработки музыкального материала и т. п.

## **НОВАЯ ФУНКЦИОНАЛЬНОСТЬ**

Проект задуман для создания эффективного серверного окружения для многопользовательского решения задач, реализуемых на Рефале. Многопользовательский подход подразумевает возможность эффективной обработки множества одновременно выполняемых запросов от произвольного количества пользователей, безостановочную работу в режиме 7/24, одновременную поддержку различных диалектов языка.

Главная ориентация такой системы — работа в компьютерной сети, обеспечивающей доступность массового подключения к ней и должную загрузку системы. Главная проблема — обеспечение приемлемой скорости вычислений при разумной их цене.

В отличие от персонального решения, серверное требует наличия достаточных возможностей масштабирования, т. е. несложного способа увеличения производительности по мере роста загрузки системы.

Запланирована реализация ряда качеств разрабатываемой системы, которые определяют её оригинальность:

- размещение всех компонентов (функций) Рефал-программы в специальном репозитории, наделённом возможностью разграничения доступа, как следствие, наличие множества пользователей;
- поддержка версионности как на уровне кода программ, так и на уровне компилятора — это позволяет вести совершенствование системы с минимальным количеством технологических остановок;
- поддержка различных диалектов Рефала в рамках одного сервера — по мере возможности и необходимости, для начала было бы достаточно реализации Базисного Рефала;
- предоставление средств взаимодействия сервера с окружающим миром посредством обмена текстовыми данными в общепринятом открытом формате, например, XML или (и) JSON;
- предоставление единых средств оптимизации и анализа выполнения программ, а также встроенного оптимизатора с подсказками, планами выполнения, учётом времени и журналированием.

Всё вышеуказанное должно предоставить неплохую возможность для обучения и разработки с использованием сильных сторон языка Рефал.

### **ВЫБОР СРЕДСТВА РЕАЛИЗАЦИИ**

Выбор средства является интересной и крайне важной задачей, так как по мере написания кода его замена очень сильно дорожает.

Были рассмотрены варианты скриптовых языков (Javascript), собственно Java (или Kotlin), исполняющие среды SQL-серверов PostgreSQL, Oracle и Microsoft как имеющие уже встроенные механизмы оптимизации исполнения.

В результате выбор был остановлен на языке Go как ориентированным на максимальную эффективность серверной работы и обладающим достаточным инструментарием для работы с интернет-запросами и базами данных и к тому же параллельным и многоплатформенным.

Кроме того, для Go можно достаточно легко адаптировать различные идеи и методы существующих компиляторов Рефала, написанных, как правило, на языке С.

#### **4. ОСОБЕННОСТИ РЕАЛИЗАЦИИ**

Главным отличием от большинства существующих версий окружения Рефала должно стать интерпретационное исполнение программы. Однако, ввиду особенности языка, традиционная (построчная) интерпретация применительно к Рефалу предположительно годится только для достаточно простых приложений. Пока что нужно понимать, что решение задачи эффективной интерпретации может затребовать львиную долю ресурсов при разработке.

Альтернативное решение — «теневая» компиляция одним из существующих готовых компиляторов и такое же «теневое» выполнение с отправкой результата потребителю может временно закрыть вопрос, но такое решение не будет «честным» и сократит ряд преимуществ реализации в дальнейшем.

Другой важный вопрос реализации — определение входного языка интерпретатора и набора базовых библиотек. Было принято решение взять за основу синтаксис Рефала-5 в рамках возможностей Базового Рефала, т. е. для начала сделать чистое, не оптимизированное сопоставление с образцом. Что касается библиотек, то необходимы арифметика, стандартный ввод и вывод и операции с копилками и ящиками. Это даст возможность создавать самостоятельные, не интерактивные полезные программы с использованием серверной платформы.

Расширение функциональности будет необходимо вести в двух направлениях:

1. Создание специализированных встроенных функций, эффективно реализующих дополнительные возможности, например, взаимодействие с серверами баз данных; доступ к ним только от лица административных пользователей.
2. Создание библиотеки функций — оболочек к специализированным

функциям. Библиотечные функции должны быть доступны обычным пользователям.

Уже сейчас есть внутренние задачи по работе с большими массивами данных, которые желательно вынести в обработку специализированных серверных приложений.

### **ТЕКУЩЕЕ СОСТОЯНИЕ РАЗРАБОТКИ**

Нужно признать, что к моменту формирования первой публикации в августе 2019 года проект находился только в самой начальной стадии. Реальные работы начались в сентябре 2019 года. Хотя продукт ещё рано «показывать на публике», он постепенно «оживает» и приобретает новые свойства.

Сложность интерпретатора оказалось выше, чем ожидалось. Сейчас решается вопрос о жестком соответствии сопоставления по образцу эталонному (Рефал-5) и технологии отладки и получения информации для конечного пользователя.

Далее необходимо решить вопросы о соответствии производительности разумным требованиям и разделению доступа для различных пользователей.

По завершении данных работ можно будет констатировать факт выхода первой версии продукта «в свет».

### **ОЖИДАЕМЫЕ ПЕРСПЕКТИВЫ**

Будучи реализованной, описываемая система может быть загружена множеством приложений: сфера образования (обучения логике мышления на базе формулирования задач на Рефале), медицинская диагностика, помощь различным социальным службам, машинный перевод (я ещё верю в это), создание роботов-ботов для различных целей.

Отдельным важным преимуществом Рефала можно отметить его как средство создания интеллекта роботизированных систем. В отличие от популярных нынче нейронных сетей, здесь в основе алгоритма лежат целеполагание и правила, на которых базируется поведение робота. Нейронные сети же пусть здесь же занимаются свойственной им задачей — первичной обработкой поступающей информации. Ближайшее будущее — за гибридными системами.

Логично ожидать, что количество пользователей из разных потребительских секторов должно стать заметным, и популярность Рефала в своих приложе-



ниях может сравниться с популярностью Фортрана до сих пор у физиков. И тот, и другой языки имеют структуру, соответствующую образу мышления потенциального пользователя, каждый в своей сфере.

## **10. ОБЩЕСТВЕННАЯ МИССИЯ**

Для продвижения любого языка программирования «в массы» нужны следующие составляющие:

1. Способность языка решать определённый круг задач. Этот круг начинается от насущных задач самого разработчика.

2. После успешного освоения начального круга задач реализация должна заинтересовать других профессиональных разработчиков. Это требует распространения информации о продукте по различным каналам, доступным разработчику. Естественно, программный продукт должен обладать всеми атрибутами продукта, начиная с качественной документации.

3. Для привлечения внимания потенциальных пользователей, не входящих в узкопрофессиональное сообщество, следует проводить информационно-просветительскую работу среди студентов и школьников, читать лекции и проводить олимпиады по освоению технологии и решению учебных задач с помощью возможностей продукта. Для этого необходима разработка пользовательской среды, дающий доступ к возможностям сервера приложений на Рефале. Возможно, это будут разные среды, ориентированные на разные аудитории. В связи с этим вспоминается замечательная среда программирования для школьников — Лого, позволяющая детям с удовольствием рисовать мультфильмы, начиная с первых занятий.

Собственно миссия здесь — распространения технологии, приносящей радость и развитие, способной привнести в повседневную работу элемент творчества, в противовес множеству современных технологий, помимо своей главной цели просто засоряющих ум.

## СПИСОК ЛИТЕРАТУРЫ

1. В.Ф. Турчин Программирование на языке РЕФАЛ. I. Неформальное введение в программирование на языке РЕФАЛ. М.: ИПМ АН СССР, 1971. Препринт №41. 55 с.

2. М.Ш. Исламов. Декларативное программирование. Рефал в обучении [Электронный ресурс] / III Международная научно-практическая конференция "Современные информационные технологии и ИТ-образование", сб. докл. // Моск. гос.ун-т им. М.В. Ломоносова, Факультет вычислительной математики и кибернетики; под ред. В.А. Сухомлина. М.: МАКС Пресс, 2008. С. 252—256.

3. Сборник трудов по функциональному языку программирования Рефал под редакцией А.П. Немытых, том II. // Переславль-Залесский: Издательство «СБОРНИК», 2015, 156 с.

---

## REFAL ON SERVER SIDE

**A. A. Gusev**

*Company "SignArt", Saint-Petersburg*

*a.gusev@signart.ru*

### **Abstract**

This work is devoted to the description of the project of updating the technology of algorithmic language application created in the USSR in the 60s of the XX century by V. F. Turchin. The language was originally intended for various logical transformations primarily of text material. In practice, the scope was wider: machine translation, optimization and compilation of programs, proof of theorems, modeling of complex electronic circuits, solving a number of problems of artificial intelligence. The language now has a sufficient number of followers, mainly in scientific circles.

The objective of the described project is to create a product that allows the use of Refal in modern mass applications and to expand the range of its potential users to the entire Internet. A survey of the community of users and developers of Refal was conducted in order to get an idea of the current state of Affairs, current implementa-

---

tions and ways of language development. Possible means of project implementation were considered. No information was received on similar developments under way.

**Keywords:** *Refal, server-side computing, text processing, xml processing, artificial intelligence, meta-calculations*

## REFERENCES

1. *V.F. Turchin*. Программирование на языке REFAL I Неформальное введение в программирование на языке REFAL. М.: IPM AN SSSR, 1971. Preprint No 41. 55 s

2. *M.Sh. Islamov*. Декларативное программирование. Refal в обучении [Электронный ресурс] /М. Ш. Islamov III Международная научно практическая конференция “Современные информационные технологии и IT образование”: сб. Докл. Моск. Гос. Ун-т им. М.В. Ломоносова, Фак. Вычислит. математики и кибернетики под ред. В.А. Сухомина. М.: MAKS Press, 2008, S. 252—256.

3. Сbornik trudov po funktsionalnomu yazyku programmirovaniya Refal. Tom II Pod redakciej A.P. Nemytyh. Pereslavl Zalesskij: Izdatelstvo “SBORNIK”, 2015, 156 s.

## СВЕДЕНИЯ ОБ АВТОРЕ



**ГУСЕВ Александр Альфредович** – начальник отдела ЗАО «СайнАрт», специалист в области разработки банковских информационных систем.

**Alexander GUSEV** – Head of Department of CJSC SignArt, specialist in the field of development of banking information systems.

email: gusev\_aleksandr@mail.ru

*Материал поступил в редакцию 16 ноября 2019 года*