Изолированный запуск поставщиков типов для компилятора F#

Бережных Алексей Владимирович

группа 16.Б10-мм

Научный руководитель: к. т. н. Литвинов Ю. В.
Рецензент: к. ф.-м. н. Григорьев Д. А
Консультант: инженер-программист Аудучинок Е. П.

Сервисы компилятора для F#

Программная библиотека с открытым исходным кодом

- Сборка и компиляция проектов
- Инструменты анализа кода
- Интерпретатор для F#
- Контекстные подсказки для кода
- Используется всеми средами разработки для F#

Поставщик типов F#

Компонент для сервисов компилятора, предоставляющий типы, свойства и методы внешнего источника данных для использования в своём коде

```
open FSharp.Data.Sql

type Sql = SqlDataProvider<Common.DatabaseProviderTypes.MSSQLSERVER,
                           "Server=(localdb)\\mssqllocaldb;Database=AuthServiceDB;Trusted_Connection=True;">

SqlDataProvider<...>.dataContext
let context = Sql.GetDataContext()
System.Linq.IQueryable<obj>
let res =
    query { for user in context.Dbo.
```

```
AspNetRoles                dbo.AspNetRoles
AspNetUsers                dbo.AspNetUsers
GetHashCode                int
AspNetRoleClaims           dbo.AspNetRoleClaims
AspNetUserClaims           dbo.AspNetUserClaims
AspNetUserLogins           dbo.AspNetUserLogins
```

```
property dboSchema.AspNetUsers: dbo.AspNetUsers with get

<summary> The base table AspNetUsers belonging to schema
```

Актуальность

Поставщики типов загружаются в тот же процесс, что и сервисы компилятора

- Зависимость от среды исполнения
- Небезопасность
- Неудобство отладки при разработке поставщиков типов

Нет возможности напрямую управлять жизненным циклом поставщика типов

Мотивация

Переход кодовой базы JetBrains Rider на среду выполнения .NET Core

- Не все поставщики типов умеют исполняться на .NET Core
- Пользователи не смогут использовать несовместимые поставщики типов в своих проектах

Постановка задачи

Цель

Реализовать механизм изолированного от JetBrains Rider управления поставщиками типов

Постановка задачи

Задачи

- Изолировать порождение поставщиков типов в отдельный от работы компилятора процесс
- Разработать и реализовать протокол взаимодействия между поставщиками типов и сервисами компилятора:
  - Описать формат передаваемых по протоколу данных
  - Описать и реализовать модель взаимодействия между зонами внутри процесса Rider и изолированного процесса для передачи информации от поставщиков типов между ними
  - Обеспечить кэширование передаваемых данных
  - Обеспечить возможность запуска поставщиков типов в пользовательской среде выполнения для .NET
- Создать тестовое покрытие для функциональности изолированных поставщиков типов и произвести апробацию полученного решения

Библиотека RD

Библиотека для коммуникации между процессами

- Разработана JetBrains
- Позволяет создать процесс с выбранной средой исполнения для .NET
- Позволяет описать формат передаваемых данных и удалённо вызываемых процедур на языке Kotlin и сгенерировать клиент-серверный код на C#
- Позволяет вести непрерывную запись метрик производительности и системной информации о передаваемых данных

Предоставляемые типы

Предоставляемые поставщиками типы

- Предоставляемые классы, конструкторы, методы, поля, свойства и др.
- Являются обёртками над настоящими Runtime-типами и используются для вывода типов в сервисах компилятора

Трудности

- Высокая связность между реализацией предоставляемых типов и FCS
- Компилятору, помимо данных о предоставляемых типах, требуются данные об обёрнутых Runtime-типах
- Runtime-типы нельзя передавать по протоколу

API Изолированных поставщиков типов

Программный интерфейс для компилятора

- Порождение поставщиков типов
- Порождение предоставляемых типов
- Вывод предоставляемых типов
- Идентификация поставщиков типов

Компоненты протокола

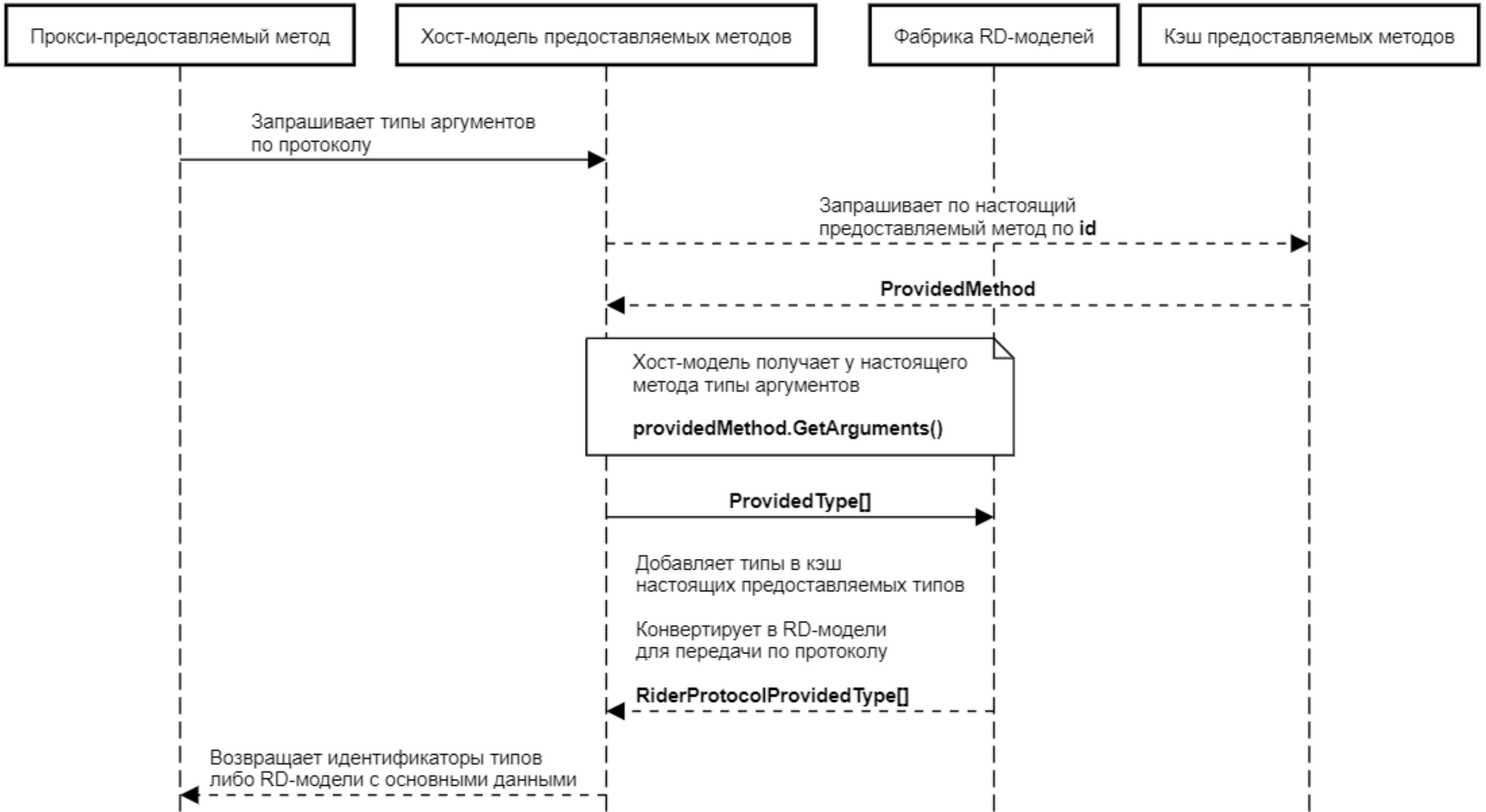
Участники на стороне Rider (Клиента)

- Прокси-поставщики типов, прокси-предоставляемые типы
- Клиенты для хостов предоставляемых типов
- Кэши

Участники на стороне изолированного процесса (Сервера)

- Предоставляемые типы
- Хосты предоставляемых типов
- Кэши

Пример создания предоставляемых типов (зона изолированного процесса)



Управление кэшированием

Кэширование на стороне Rider

- Глобальные кэши поставщиков типов
- Локальные кэши предоставляемых типов

Кэширование на стороне изолированного процесса

- Кэши уникальных предоставляемых типов
- «Легковесные» кэши предоставляемых типов

Инвалидация кэшей

Поддержка устаревших версий поставщиков типов

Проблемы

- Несовместимые версии библиотеки сервисов компилятора
- Особенности управления ссылками на обьекты в сервисах компилятора

Тестирование и апробация

Тестирование

- Написан тестовый поставщик типов
- Написаны тесты, проверяющие изолированный запуск, вывод предоставляемых типов и отображение базовых ошибок

Апробация

- Апробация на популярных поставщиках типов:
  - Поставщики типов для SQL, YAML, XML, JSON, Regex, Swagger
- Планируется провести A/B-тестирование на пользователях JetBrains Rider

Полученные результаты

- Изолировано порождение поставщиков типов в отдельный от работы компилятора процесс:
  - Предложен к публикации в компилятор и реализован в JetBrains Rider программный интерфейс, позволяющий изолировать работу поставщиков типов
- Разработан протокол взаимодействия между поставщиками типов и сервисами компилятора на основе библиотеки RD:
  - Описана и реализована модель передачи предоставляемых типов между процессами
  - Реализованы и опубликованы в компиляторе исправления инкапсуляции типов времени выполнения для предоставляемых типов
  - Реализован механизм кэширования передаваемых данных
- Создано тестовое покрытие для функциональности изолированных поставщиков типов и произведена апробация полученного решения на нескольких популярных поставщиках типов

Ссылки

API изолированных поставщиков типов в компиляторе F#

- https://github.com/dotnet/fsharp/pull/8235

Исправления инкапсуляции типов времени выполнения для предоставляемых типов

- https://github.com/dotnet/fsharp/pull/8656
- https://github.com/dotnet/fsharp/pull/9235
- https://github. com/dotnet/fsharp/pull/8809
- https://github. com/dotnet/fsharp/pull/9005

Реализация в JetBrains Rider

- https://github.com/JetBrains/fsharp-support/pull/91
- https://github.com/DedSec256/fsharp/tree/ber. a/TypeProviders