Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б07-мм

Интеграция Qt/QML с OCaml

Михайлов Илья Игоревич

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
ассистент кафедры системного программирования Д. С. Косарев

Санкт-Петербург
2023

Оглавление

Введение

OCaml – это функциональный язык программирования семейства ML со статической типизацией. С проектированием графического интерфейса пользователя (GUI) в данном языке все не так просто: в opam – менеджере пакетов для OCaml – множество библиотек с базовыми наборами инструментов, но проектирование полноценных динамических GUI-интерфейсов на них затруднительно. Вне opam'а существуют GUI-фреймворки специально для OCaml, биндинги к популярным графическим библиотекам, а также синтаксические расширения языка OCaml, позволяющие проектировать WEB-приложения, но не во всех таких решениях графический интерфейс можно описывать предметно-ориентированным языком разметки.

Удобно проектировать графический интерфейс с помощью встроенной прямо в язык программирования разметки. Примером такой разметки являются HTML-теги в синтаксическом расширении JavaScript под названием JSX, где с помощью них можно описывать графический интерфейс прямо в JavaScript.

Фреймворк Qt/QML позволяет проектировать кроссплатформенные приложения, используя язык разметки QML для описания интерфейса пользователя (UI), и имеет богатый набор инструментов для создания приложений любой направленности. QML компилируется[1] в C++, и как следствие, открывается доступ к множеству библиотек на C++.

Вдохновляясь примером JSX предлагается интегрировать в OCaml предметно-ориентированный язык разметки QML из фреймворка Qt/QML

[1]href=https://www.qt.io/blog/the-new-qtquick-compiler-technology (дата обращения 14.04.2023)
Дата сборки: 15 апреля 2023 г.

1. Постановка задачи

Целью работы является написание реализации, выполняющей интеграцию Qt/QML с OCaml. Для ее выполнения были поставлены следующие задачи:

- добавить возможность написания разметки графического интерфейса внутри OCaml с помощью создания синтаксического расширения;
- добавить возможность генерации кода на QML из нового синтаксиса.

2. Обзор

Целью обзора является поиск и описание существующих библиотек и фреймворков для создания UI и WEB-приложений, поиск похожих технологий, расширяющих синтаксис языка для вызова функций, а также поиск инструментов для реализации собственного решения.

2.1. Существующие аналоги

Ниже приведены найденные решения, которые могут описать спектр возможностей для проектирования графических приложений:

- LablGTK3[2];
- lablqml[3];
- Bogue[4].

LablGTK3 – проект, представляющий собой интерфейс для взаимодействия OCaml и GTK (GIMP tool kit) – кроссплатформенного фреймворка для создания GUI на основе виджетов. Является популярным инструментом для проектирования GUI, в силу общей известности библиотеки GTK. Нет встроенного языка разметки.

lablqml – представляет собой интеграцию QtQuick с OCaml. Главной особенностью является возможность описать обработчики сигналов для контролов (объектов QML) на языке OCaml, тогда как описание GUI происходит обособленно от среды OCaml на языке QML.

Bogue – библиотека для создания GUI виджетов, написанная с нуля на основе SDL2, главной особенностью которой является легковесность, отсутствие интеграций в другие библиотеки для UI, ускорение работы графического процессора (GPU acceleration), благодаря SDL2. В данном проекте не используется язык разметки графичесеского интерфейса.

[2]https://garrigue.github.io/lablgtk/ (дата обращения 14.04.2023)
[3]http://kakadu.github.io/lablqml/ (дата обращения 14.04.2023)
[4]http://sanette.github.io/bogue/Principles.html (дата обращения 14.04.2023)

Также был осуществлен поиск решений, которые являются синтаксическими расширениями для OCaml и позволяют проектировать WEB-приложения:

- ReasonML[5] и Melange[6];
- ReScript[7];
- ReveryUI[8] и Brisk[9].

ReasonML является языком программирования, который представляет собой синтаксическое расширение языка OCaml и напоминает C и JavaScript. Благодаря интеграции с JS (компилятор Js_of_ocaml) и встроенной поддержкой JSX, становятся доступны многие популярные библиотеки для JS (например, ReactJS). ReScript также является расширением синтаксиса языка OCaml, отличающимся от Reason, и собственным компилятором в JavaScript.

2.2. JSX

JSX – синтаксическое расширения для языка JavaScript, позволяющее описывать UI HTML-тегами непосредственно в коде JS, которые раскрываются в компоненты библиотеки ReactJS. Преобразование кода выполняет компилятор Babel.js. Предлагается использовать данную идею для проектирования Qt/QML приложений прямо внутри OCaml, т.е. реализовать язык разметки, который можно будет перевести в вид, понятный Qt/QML.

2.3. Инструменты

2.3.1. Формальная грамматика и LL(1)-анализатор

Формальная грамматика [1] – это набор из четырёх компонентов:

[5]https://reasonml.github.io/docs/en/what-and-why (дата обращения 14.04.2023)
[6]https://jchavarri.github.io/melange-docs/ (дата обращения 14.04.2023)
[7]https://rescript-lang.org/ (дата обращения 14.04.2023)
[8]https://www.outrunlabs.com/revery/ (дата обращения 14.04.2023)
[9]https://github.com/briskml/brisk (дата обращения 14.04.2023)

1. набор терминалов;
2. набор нетерминалов;
3. набор правил вывода;
4. начальный символ - специальный нетерминал.

LL(1)-анализатор[10] – это синтаксический анализатор для некоторых контекстно-свободных грамматик, где буквы L означают, что анализ входных данных идёт слева направо и строится левосторонний вывод.

2.3.2. Camlp5

Camlp5 [3] – препроцессор для OCaml [2], который использует LL(1)-анализатор и который позволяет расширять язык новым синтаксисом.

В Camlp5 используется парсер с заглядыванием вперед на одну лексему. Это необходимо учитывать в написании решения, так как рассмотрения лишь одной лексемы из потока не всегда будет достаточно для распознания нужной синтаксической конструкции.

2.3.3. Dune

Dune – система сборки проектов для OCaml. В данной задаче используется для создания проекта, состоящего из нескольких модулей, а также для использования препроцессорной обработки, что Dune позволяет удобно делать.

2.3.4. Qt/QML

Qt/QML [4] – кроссплатформенный фреймворк для создания графических приложений. В данной работе используется фреймворк QtQuick, а также утилита qmlscene для отображения .qml файла.

[10] https://en.wikipedia.org/wiki/LL_parser

3. Реализация

В данный момент доступна поддержка:

- вложенных контролов (QML object) в контролы и в свойства;
- ключевого слова import.

Доступна возможность написания простых операций над базовыми типами в значениях свойства, например:

- операции над числами;
- конкатенация строк.

На данном этапе отложена поддержка:

- произвольного JavaScript кода;
- сигналов, слотов;
- свойств:

```
property <type> <name> : <value>
property alias <name> : <reference>
```

В листинге ниже приведен пример написания разметки графического интерфейса с помощью синтаксического расширения.

Листинг 1: Код на OCaml

```
open Ocamlext.Pr_qml

QML
  import "QtQuick 2.5"

  Rectangle {
    id: root;
    width: 640;
    height: 480;
```

```
    Text {
      x: parent.width / 2;
      y: root.height / 2;
      text: "Hello, World! X coord:" + x +
      " Y coord: " + y
    }
  }
ENDQML;;
```

Листинг 2: Результат кодогенерации

```
import QtQuick 2.5
Rectangle
{
        id: root
        width: 680
        height: 480
        Text
        {
                x: parent.width / 2
                y: parent.height / 2
                text: "Hello, World! X coord:" + x +
        " Y coord: " + y
        }
}
```

Заключение

В результате работы были выполнены следующие задачи:

- написание прототипа парсера разметки пользовательского интерфейса;
- кодогенерация QML из дерева разбора синтаксиса.

Задачи на будущее:

- добавить поддержку раздельного написания разметки пользовательского интерфейса;
- добавить возможность вызывать OCaml из сгенерированного кода на QML.

Ссылка на репозиторий GitHub: https://github.com/mikhaylovilya/ocaml-syntax-extension

Список литературы

[1] Compilers: Principles, Techniques, and Tools (2nd Edition) / Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. — Addison Wesley, 2006. — August. — ISBN: 0321486811.

[2] Minsky Yaron, Madhavapeddy Anil, Hickey Jason. Real World OCaml - Functional Programming for the Masses. — O'Reilly, 2013. — P. I–XXII, 1–483. — ISBN: 978-1-4493-2391-2.

[3] camlp5. — URL: https://camlp5.github.io (дата обращения: 14.04.2023).

[4] qml book. — URL: https://qmlbook.github.io/ (дата обращения: 14.04.2023).