

Univerza v Ljubljani  
Fakulteta za računalništvo in informatiko

Miha Hribar

**Razvoj medplatformne knjižnice za uporabo  
v mobilnih in spletnih aplikacijah**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

Dejan Lavbič  
MENTOR

Ljubljana, 2014



© 2014, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.<sup>1</sup>

---

<sup>1</sup>V dogovorju z mentorjem lahko kandidat diplomsko delo s pripadajočo izvirno kodo izda tudi pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. Creative Commons, GNU GPL.



*Namesto te strani se vstavi original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!*



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom Dejana Lavbiča,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki “Dela FRI”.

— Miha Hribar, Ljubljana, maj 2014.





Univerza v Ljubljani  
Fakulteta za računalništvo in informatiko

Miha Hribar

## Razvoj medplatformne knjižnice za uporabo v mobilnih in spletnih aplikacijah

### POVZETEK

Razvoj aplikacij za več različnih platform je težaven. Odpira veliko možnosti za napake, oteži testiranje in odpravljanje napak, ter skoraj onemogoči sočasno nadgrajevanje aplikacij. Rezultat so dolgotrajni razvojni cikli in počasno dodajanje funkcionalnosti, kar v današnjem "startup" svetu ni zaželeno.

Kljub različnosti med posameznimi platformami je ponavadi veliko kode z identično funkcionalnostjo, ki jo je potrebno razviti za vsako platformo posebej. Velikokrat je v podjetju za vsako od platform zadolžen drug razvijalec, še bolj pogosto pa razvoj na različnih platformah ne poteka sočasno. Rešitev iz te zagate je razvoj medplatformne knjižnice.

Cilj diplomske naloge je razvoj knjižnice za RRULE RFC5545<sup>2</sup> specifikacije, ki omogoča generiranje ponavljajočih se koledarskih dogodkov in jo je možno uporabiti v spletni, iOS, Android in Windows Phone aplikaciji. Našteli bomo možne pristope, navedli prednosti in slabosti, ter na koncu izbrali najbolj primerno rešitev za implementacijo knjižnice.

**Ključne besede:** medplatformna knjižnica, iOS, Android, Windows Phone, JavaScript, Emscripten, LLVM

---

<sup>2</sup><http://tools.ietf.org/html/rfc5545#section-3.3.10>



University of Ljubljana  
Faculty of Computer and Information Science

Miha Hribar

## Developing a cross platform library for use in mobile and web applications

### ABSTRACT

Developing applications for different platforms is complicated. It opens a lot of avenues for mistakes, complicates testing and bugfixing, while almost completely destroys any chance of simultaneous application upgrade. The result of this are prolonged development cycles and slow feature creep, which in today's "startup" world is not an option.

Despite the differences between different platforms, they most likely share a lot of functionality which has to be developed for each platform. Most of the time each platform is handled by a different developer and usually not simultaneously with other applications. The solution to this problem is to develop a cross-platform library.

The goal of the thesis is to develop a library for the RRULE RFC5545<sup>3</sup> specification, which enables applications to schedule and display recurring events. The library will then be used in an web, iOS, Android and Windows Phone application. We will outline different approaches to writing the shared library, list the pros and cons and in the end decide on the best approach.

**Key words:** cross platform library, iOS, Android, Windows Phone, JavaScript, Emscripten, LLVM

---

<sup>3</sup><http://tools.ietf.org/html/rfc5545#section-3.3.10>



## ZAHVALA

*Rad bi se zahvalil mentorju doc. dr. Dejanu Lavbiču za strokovno svetovanje in predvsem za potrpežljivost pri nastanku diplomskega dela. Hvala družini za vso podporo in finančno pomoč pri študiju. Hvala vsem ostalim, ki ste mi stali ob strani.*

*In seveda hvala Petri, ker mi pustiš da sem to kar sem.*

— Miha Hribar, Ljubljana, maj 2014.



## KAZALO

<b>Povzetek</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Zahvala</b>	<b>v</b>
<b>1 Uvod</b>	<b>1</b>
<b>2 Pregled metod medplaformnega razvoja</b>	<b>3</b>
2.1 Celovit . . . . .	3
2.1.1 Qt . . . . .	3
2.1.2 Xamarin . . . . .	4
2.1.3 Adobe Air . . . . .	4
2.2 Hibriden . . . . .	5
2.2.1 Apache Cordova / PhoneGap . . . . .	5
2.2.2 Appcelerator Titanium . . . . .	5
2.3 Deljen . . . . .	5
2.3.1 Lua . . . . .	5
2.3.2 Haxe . . . . .	5
2.3.3 XMLVM . . . . .	5
2.3.4 C++ in emscripten . . . . .	5
<b>3 Razvoj knjižnice</b>	<b>7</b>
3.1 C++ . . . . .	7
3.2 Emscripten . . . . .	7
3.3 Omejitve . . . . .	7

<b>4</b>	<b>Vključitev knjižnice v različne platforme</b>	<b>9</b>
4.1	iOS . . . . .	9
4.2	Android . . . . .	9
4.3	Windows Phone . . . . .	9
4.4	Spletna aplikacija . . . . .	9
<b>5</b>	<b>Ugotovitve</b>	<b>11</b>
<b>A</b>	<b>Appendix A</b>	<b>15</b>
A.1	Intro section . . . . .	15
A.2	... . . . .	15



# 1 Uvod

Dandanes uporabljamo več različnih naprav sočasno. V lasti imamo najverjetneje prenosni računalnik, pametni telefon in po možnosti še tablico. Ko najdemo aplikacijo, ki nam je všeč, od te pričakujemo brezhibno delovanje na vseh naših napravah.

To je seveda zelo težko doseči, sploh z majhno ekipo. Dodatno se stvari zakomplicirajo, če so vse te naprave na različnih operacijskih sistemih. Tako imam lahko Windows prenosnik, Android telefon in Apple tablico. Kot razvijen uporabnik pričakujem, da je izbrana aplikacija na voljo na vse naštetih platformah.

Za razvijalca smo ravnokar opisali nočno moro. Da zadovolji potrebe uporabnikov, je primoran razviti isto aplikacijo za vsako od platform. Četudi omejimo razvoj na najbolj priljubljene platforme iOS, Android in Windows Phone, smo ravnokar našeli tri povsem različne tehnologije, tri različne jezike in s tem tri priložnosti za povsem različne težave pri implementaciji naše aplikacije. Veliko truda in energije je potrebno, da so te aplikacije poenotene in da skladno sledijo razvoju novih funkcionalnosti.

Izkušen razvijalec bo pri predstavitvi problema takoj pomislil na medplatformni razvoj, ki si ga bomo ogledali v drugem poglavju. Omenili bomo tako imenovane “celotive”

metode, kot so Qt in Xamarain, “hibridne” kot sta recimo PhoneGap in Appcelerator Titanium, ter “deljene” metode npr. Lua, Haxe in C++.

V tretjem poglavju si bomo ogledali zakaj smo se odločili za razvoj knjižnice s pomočjo C++, ter jo tudi zgraditi. V četrtem poglavju jo bomo ovili v nekaj ovojev in uspešno uporabili v ObjC (iOS), Javi (Android) in C# (Windows Phone). Predstavili bomo tudi način kako C++ kodo prevesti neposredno v JavaScript s pomočjo orodja emscripten, ter na koncu knjižnico vključili tudi v spletno aplikacijo.

Pa začnimo.

## 2 Pregled metod medplaformnega razvoja

### 2.1 Celovit

Celovita metoda za razvoj uporablja eno samo okolje, s pomočjo katerega nato aplikacijo izvozimo za različne platforme - tako imenovani princip “piši enkrat, uporablaj povsod”, ki nam je znan iz programskega jezika Java. Zajema tako orodja za grafični vmesnik, kot prenosljivost programske kode.

#### 2.1.1 Qt

Qt<sup>1</sup> je ogrodje za grafično programiranje za več platform s pomočjo jezika C++ in QML<sup>2</sup>. Omogoča nam sočasni razvoj za platforme OSX, Linux, Windows, Android in iOS. Podpira tudi uporabo HTML5<sup>3</sup> namesto QML, kar pomeni, da spletni razvijalci lahko uporabijo že obstoječe znanje in učenje novega jezika ni potrebno.

Qt projekt je povsem odprtokoden in dovoljuje uporabo v skladu z licencama GPL

---

<sup>1</sup><http://qt-project.org>

<sup>2</sup>Qt Meta Language ali Qt Modeling Language, vir [Wikipedia](#)

<sup>3</sup><http://en.wikipedia.org/wiki/HTML5>

v3<sup>4</sup> in LGPL v2.1<sup>5</sup>, a če želite orodje uporabiti za razvoj mobilne aplikacije, boste morali za to odšteti 149\$ mesečno.

Če bi izpostavili dva projekta, ki uporabljata ogrodje Qt, bi omenil medijski predvajalnik VLC in program Mathematica.

Glavne slabosti Qt so neskladnost z izgledom ostalih aplikacij na mobilnih platformah, plačljiva licenca za razvoj mobilnih aplikacij ter končna velikost samih programov.

### 2.1.2 Xamarin

Xamarin<sup>6</sup> je ogrodje za sočasen razvoj aplikacij za platforme iOS, Android, Mac in Windows v jeziku C#. Izjaha iz projekta Mono<sup>7</sup>, ki omogoča uporabo ogrodja .NET na različnih platformah. Cenovno ni ravno ugoden, saj se cene začnejo pri 299\$/mesec za vsakega razvijalca in vsako platformo.

Ogrodje omogoča razvoj aplikacij, katerih izgled je skladen z ostalimi aplikacijami na izbrani platformi. Kot primer si lahko ogledamo aplikacijo za poslušanje glasbe Rdio<sup>8</sup>, ki je na voljo za iOS, Android in Windows Phone.

Glavna slabost ogrodja Xamarin je cena. Za majhno ekipo je začetna cena enostavno previsoka. Vprašljiva je hitrost dodajanja funkcionalnosti posameznih platform, ko se te nadgradijo, določen riziko predstavlja tudi muhavost posameznih platform pri omejitvah uporabe tega ogrodja.

### 2.1.3 Adobe Air

Adobe Air je brezplačno ogrodje, ki omogoča zagon iste aplikacije na platformah iOS, Android, Mac, Windows in Linux. Čeprav za razvoj namiznih aplikacij omogoča uporabo HTML in Javascript, je za razvoj mobilnih aplikacij omejen na uporabo jezika ActionScript. V času pisanja diplomske naloge ogrodje ne omogoča zagon na platformi Windows Phone.

Kot glavno slabost ogrodja Adobe Air bi navedel upadanje zanimanja za orodje Flash. Špekuliramo lahko tudi o planih podjetja Adobe, saj so pred kratkim kupili podjetje Nitobi, ki je avtor ogrodja Phone Gap (katerega si ga bomo ogledali v nadaljevanju).

---

<sup>4</sup><http://www.gnu.org/copyleft/gpl.html>

<sup>5</sup><https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>

<sup>6</sup><https://xamarin.com>

<sup>7</sup><http://www.mono-project.com>

<sup>8</sup><https://www.rdio.com>

## 2.2 Hibriden

Hibridna metoda za razvoj aplikacij uporablja spletne tehnologije v sožitju z kodo za posamezno platformo (t.i. premostitvena tehnika), ki omogoča dostop do glavnih funkcij naprav (kot so kamera, pospeškometer in podobno).

### 2.2.1 Apache Cordova / PhoneGap

Ogrodje Apache Cordova<sup>9</sup> je odprtokodni projekt, ki vsebuje vtičnike za uporabo funkcionalnosti naprav, kot so kamera in pospeškometer. V času pisanja diplomske naloge ogrodje podpira iOS, Android, Windows Phone, Blackberry, Palm WebOS, Bada in Symbian. Projekt PhoneGap<sup>10</sup> je dejansko samo ena od distribucij projekta Apache Cordova, ki poleg vseh obstoječih funkcionalnosti ponuja tudi razne storitve na katerih delajo v podjetju Adobe.

Za razvoj aplikacij razvijalci lahko uporabljajo spletne tehnologije HTML, CSS in JavaScript. S pomočjo ogrodiv jQuery Mobile in Sencha Touch je možno izdelati aplikacije, katerih izgled je zelo lep približek ostalim aplikacijam na izbrani platformi.

Glavna slabost tega pristopa tiči v performanci in odzivnosti aplikacije. Trenutno je tudi težko izdelati aplikacije, ki so grafično zahtevnejše.

### 2.2.2 Appcelerator Titanium

## 2.3 Deljen

### 2.3.1 Lua

### 2.3.2 Haxe

### 2.3.3 XMLVM

### 2.3.4 C++ in emscripten

---

<sup>9</sup><http://cordova.apache.org>

<sup>10</sup><http://phonegap.com>



## 3 Razvoj knjižnice

### 3.1 C++

### 3.2 Emscripten

### 3.3 Omejitve





## 4 Vključitev knjižnice v različne platforme

### 4.1 iOS

### 4.2 Android

### 4.3 Windows Phone

### 4.4 Spletna aplikacija



# 5 Ugotovitve





## LITERATURA



# A Appendix A



## A.1 Intro section

All material unsuitable for the main part of the work (e.g. source code listings, extensive proofs, ...) should be placed in the Appendix.

## A.2 ...