

SVK tým
(Sloup, Vavříčka, Kořán)

Vize projektu Mapového lokátoru

Verze 0.2

Martin Sloup
27.3.2009

Historie verzí

Datum	Verze	Popis	Autor
25. března 2009	0.1	Vytvoření dokumentu	Martin Sloup
27. března 2009	0.2	Přidání chybějících částí	Martin Sloup

Obsah

Historie verzí.....	2
1 Úvod	3
2 Popis navrhovaného řešení	3
2.1 Komponentový model aplikace	3
2.2 Vizuální část	3
2.3 Registr objektů	4
2.4 Lokátory	4
2.4.1 Lokátor načítající ze souboru	4
2.5 Konfigurace registru objektů	4
3 Stěžejní funkce	4
4 Plán implementace funkcí	4
5 Rozdělení prací	4
6 Omezení produktu	5
7 Požadavky na prostředí	5
8 Glosář	5

1 Úvod

Na univerzitě se pracuje v komponentovém modelu OSGi. Rádi by demonstrovali použití komponentového modelu OSGi v aplikaci umožňující uživateli sledování pohybujících se objektů na mapě. Předpokladem je možnost pohybu s mapou její přiblížení, oddálení a výběr zobrazených objektů.

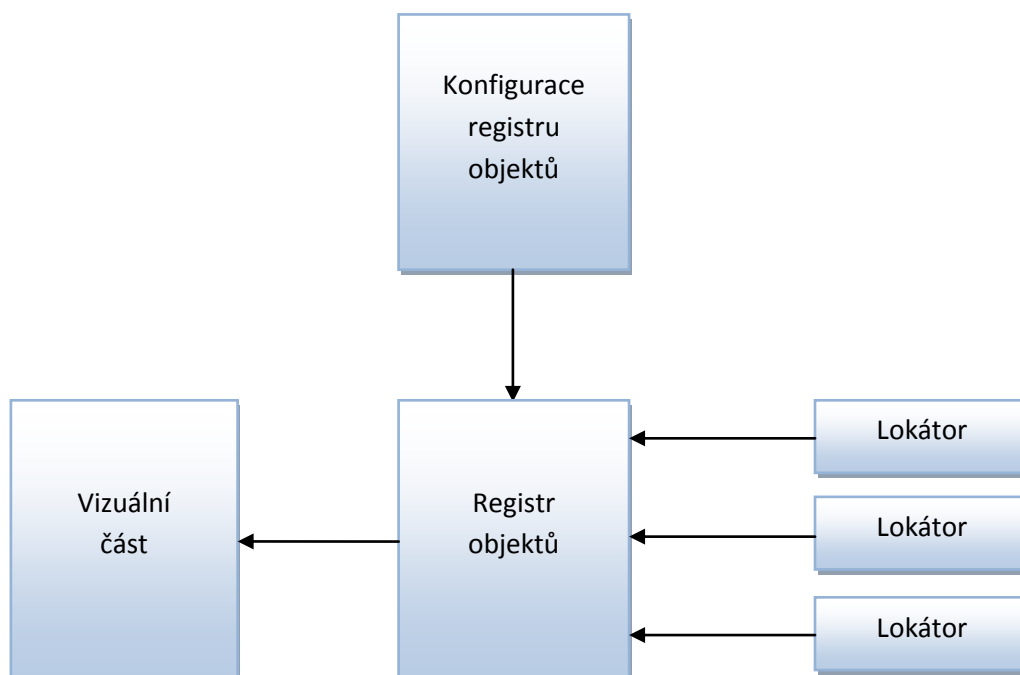
2 Popis navrhovaného řešení

Zadavatelem projektu bylo navrženo rozčlenění objektů aplikace do příslušných komponent OSGi model. Jeho rozvržení se ukázalo vhodné, a proto jsme ho s menšími úpravami přijali za model komponent.

Při návrhu bylo nutné vybrat mapové API, které bude vykreslovat, jak samotnou mapu, tak i příslušné objekty na mapě. Jako konečné řešení bylo zvoleno mapové API Google Maps z důvodu, podrobnosti map, jednoduchého ovládání, ale i jednoduché implementace v produktu. Google Maps API je tvořeno komponentou ve webové stránce, proto jsme usoudili, že vizuální část by bylo vhodné vytvořit ve formě webové aplikace. Uživatelské prostředí bude vytvořeno pomocí HTML značkovacího jazyka společně se skriptováním na straně klienta v jazyce JavaScript. Bylo nutné vymyslet způsob komunikace klientské části a serverové části tvořenou servlety. Rozhodli jsme se pro komunikaci použít REST nad protokolem HTTP v datovém formátu JSON.

2.1 Komponentový model aplikace

Následující komponentový model aplikace vychází z návrhu zadavatele:



Mapový lokátor se vnitřně bude dělit na tři části: Vizuální část, Registr objektů, Konfigurace registru objektů a samotné Lokátory. Každá z těchto čtyř částí bude vytvořena jako komponenta OSGi modelu. U Lokátoru je každá implementace vedena zvlášť jako komponenta.

2.2 Vizuální část

Vizuální část slouží jako prostředek pro komunikaci s uživatelem. Obsahuje mapu, se kterou může uživatel pohybovat. Zároveň se na mapě zobrazují objekty aktualizující svoji polohu na mapě v reálném čase. Uživatel má možnost měnit zobrazování či skrytí bodů zájmu.

2.3 Registr objektů

Registr objektů slouží jako shromaždiště typu sledovaný objekt \leftrightarrow lokátor, a zároveň funguje jako prostředník komunikace mezi lokátory a vizuální částí. Jeho hlavní úlohou je uchování aktuální polohy objektů v rejstříku. Nástroj pro konfiguraci objektů v registru objektů není potřebný, stačí tedy jednoduchý konfigurační soubor.

2.4 Lokátory

Lokátor je jakýsi adaptér zpřístupňující informace o poloze objektu. Tyto informace získává buď ze souboru, z internetu, z GPS, či jiného prostředku. Každý lokátor je implementován zvlášť jako komponenta OSGi modelu.

2.4.1 Lokátor načítající ze souboru

Jedním ze způsobu získání informací o poloze objektu na mapě je načítání ze souboru. V souboru jsou uloženy souřadnice objektu v závislosti na čase. Tento typ lokátoru musí mít možnost konfigurace. V konfiguraci je možnost určit kromě souboru s daty, také počáteční čas a koncový čas smyčky určující odkud a kam se objekt pohybuje.

2.5 Konfigurace registru objektů

Poskytuje možnosti nastavení registru objektů. Kromě seznamu aktuálně použitých objektů udržuje informace o nastavení objektu.

3 Stěžejní funkce

- FU1: Posun mapy, přiblížit, oddálit mapu
- FU2: Zobrazit/schovat určité objekty na mapě
- FU3: Aktualizovat v časových intervalech polohu bodu na mapě
- FU4: Zobrazit informace o příslušném objektu na mapě
- FU5: Načítat informace o poloze objektu z textového souboru obsahující záznam trasy v NMEA formátu
- FU6: Načítat informace o poloze objektu z webové služby

4 Plán implementace funkcí

Funkce	Iterace 1	Iterace 2	Iterace 3
FU1	Implementováno		
FU2	Není implementováno	Implementováno	
FU3	Není implementováno	Implementováno	
FU4	Není implementováno	Implementováno	
FU5	Není implementováno	Implementováno	
FU6	Není implementováno	Není implementováno	Implementováno

5 Rozdělení prací

- Martin Sloup
 - Implementace Vizuální části
 - Implementace uživatelského rozhraní
 - Zajištění komunikace mezi Vizuální částí a Registrem objektů

- Zdeněk Kořán
 - Implementace Registru objektů
 - Implementace Konfigurace registru objektů
- Tomáš Vavříčka
 - Implementace Lokátoru načítající záznam trasy ve formátu NMEA
 - Implementace Lokátoru načítající aktuální pozici objektu z webové služby

6 Omezení produktu

Výsledný produkt nebude mít následující funkce:

- Zobrazení historie trasy pohybu objektu
- Export informací o zobrazených objektech
- Požadované přihlášení jmenem a heslem pro zobrazení některých objektů
- Seskupení objektů do tematických skupin
- Hledání nebo filtrování objektů na základě klíčových slov

7 Požadavky na prostředí

Předpokladem pro práci s výsledným produktem je běžný počítač s nainstalovanou podporou Javy ve verzi 1.5 a vyšší s možností připojení na Internet a jeden z následujících plně podporovaných prohlížečů:

- Internet Explorer 6.0 a vyšší
- Mozilla Firefox 2.0 a vyšší

8 Glosář

NMEA formát

Neboli NMEA 0183 je standard pro komunikaci s GPS přijímačem vytvořenou organizací National Marine Electronics Association

OSGi

OSGi je specifikace, která definuje standard pro vývoj, nasazení a správu aplikací v řízeném prostředí. Řízeným prostředím je OSGi kontejner a aplikací je takzvaný bundle. Zjednodušeně řečeno OSGi kontejner je vlastní prostředí vystavěné nad JVM, které řídí soužití aplikací (takzvaných bundles, nebo-li balíků) v tomto prostředí - viditelností určitého API počínaje a definicí závislostí konče.

REST

Representational State Transfer je koncept pro design distribuované architektury. Distribuovaná architektura v tomto smyslu znamená, že části programu běží na různých strojích a pro svoji komunikaci využívají síť. Pod programem si můžete představit například webovou aplikaci, kde internetový prohlížeč komunikuje s webovým serverem, aplikaci pro výměnu dat mezi finančními institucemi, kde dochází k vzájemnému volání mezi servery.

JSON

JavaScript Object Notation je odlehčený formát pro výměnu dat. Je jednoduše čitelný i zapisovatelný člověkem a snadno analyzovatelný i generovatelný strojem. Je založen na pod-

množině programovacího jazyka JavaScript. JSON je textový, na jazyce zcela nezávislý formát, využívající však konvence dobře známé programátorům jazyků rodiny C (C, C++, C#, Java, JavaScript, Perl, Python a dalších). Díky tomu je JSON pro výměnu dat opravdu ideálním jazykem.