

**Analýza technológií, dekompozícia a dátový
model k projektu
Smelý zajko sprievodcom**
Tvorba informačných systémov

Autori: Natália Ďurisová, Dominik Knechta,
Martin Mudroch, Miroslav Gregorec

Obsah

1	Úvod	3
2	Analýza technológií	4
2.1	Technológie na správu robota:	4
2.2	Technológie pre webové rozhranie :	5
3	Dekompozícia	6
3.1	Komponentový diagram	7
3.2	Popis komponentov	7
3.2.1	Komponent Zapnutie robota cez web	7
3.2.2	Komponent Jazdenie po pavilóne	7
3.2.3	Komponent Rozpoznanie Osoby	7
3.2.4	Komponent Navigácia	8
3.2.5	Komponent Zaslanie reportu	8
3.3	Triedny diagram	8
3.4	Popis tried	8

1 Úvod

Cieľom tohto dokumentu je analyzovať a popísať použité technológie, ktoré budú použité pri tvorbe projektu smelý zajko sprievodcom. Dané technológie budú v dokumente popísané slovne a taktiež znázornené pomocou komponentového diagramu. Keďže náš projekt je robot, ktorý nemá žiadne front-endové prostredie rozdeliteľné na komponenty, naša dekompozícia bude rozdelená podľa naprogramovaných tried (class), ktoré spravujú robota. Náš projekt taktiež **nepracuje s databázou, preto neobsahuje dátový model**. V dokumente sú okrem technológií na správu robota, popísané aj technológie použité pre webové prostredie a vzájomnú komunikáciu medzi robotom a webom.

2 Analýza technológií

Technológie môžeme rozdeliť na 2 skupiny podľa využitia:

- Technológie na správu robota
- Technológie pre webové rozhranie

2.1 Technológie na správu robota:

1. Programovací jazyk (C++)

Podstatou celého projektu je programovací jazyk, v ktorom budú naprogramované všetky metódy na riadenie robota. Keďže úlohou robota je aj rozpoznávať postavy rozhodli sme pre jazyk C++, pretože najlepšie spolupracuje s knižnicou OpenCV bližšie opísanou nižšie. C++ je viacparadigmaticý programovací jazyk vyššej úrovne na všeobecné použitie, ktorý umožňuje pracovať aj s prostriedkami nízkej úrovne. Má statickú typovú kontrolu, podporuje procedurálne programovanie, dátovú abstrakciu, objektovo orientované programovanie, ale aj generické programovanie.

2. Knižnica na rozpoznávanie postáv (OpenCV)

Počas jazdy robot sníma prostredie pomocou webkamery. Obrázok z webkamery sa odosiela na rozpoznávanie či sa na ňom nachádza nejaká postava. Takúto úlohu dokáže spracovať vybraná knižnica OpenCV. OpenCV je tzv. open-source, multiplatformová knižnica určená pre manipuláciu s obrazom. Je zameraná predovšetkým na počítačové videnie a spracovanie obrazu v reálnom čase.

3. Mapy vo formáte SVG

Nato aby robot mohol správne navigovať po informatickom pavilóne potrebuje poznať polohu miestností a taktiež kde sa práve nachádza. Na tento účel použijeme už vopred pripravené SVG mapy informatického pavilónu, v ktorých sú zaznamenané súradnice potrebných miestností. Scalable Vector Graphics (SVG) je značkový jazyk z rodiny značkových jazykov XML, ktorý je určený na opis dvojrozmernej, statickej alebo animovanej vektorovej grafiky.

4. Hlasový syntetizátor pre komunikáciu s človekom

Keďže robot komunikuje s človekom aj verbálne, potrebujeme na daný problém takzvaný TTS (text-to-speech) program. Robot komunikuje s človekom slovenským jazykom, preto bude potrebné použiť syntetizátor, ktorý podporuje daný jazyk.

2.2 Technológie pre webové rozhranie :

1. Hypertext Markup Language (HTML)

HTML je jednoduchý značkovací jazyk bežne používaný na tvorbu statických webových stránok zobrazovaných vo webovom prehliadači. Tvorí základ pre rozšírenie stránok o dynamické prvky pomocou jazykov ako napr. PHP. Jeho výhodou je jednoduchosť a bežná podpora zo strany internetových prehliadačov. Veľmi často je používaný spolu s kaskádovými štýlmi (CSS). Jazyk bude použitý pre statický návrh webového prostredia k robotovi, ktorý má informačný aj spravovací charakter.

2. Cascading Style Sheets (CSS)

Kaskádové štýly alebo CSS je všeobecné rozšírenie (X)HTML. Organizácia W3C označuje CSS ako jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov. Štýly umožňujú oddeliť štruktúru HTML alebo XHTML od vzhľadu. Pre náš projekt bude CSS použité na design webovej stránky.

3. Hypertext Preprocessor (PHP)

PHP je populárny open source skriptovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií (na strane servera) a pre vývoj dynamických webových stránok. Pre náš projekt bude PHP využitý na prijímanie údajov o stavoch robota, ich následné spracovanie a taktiež na prihlasovanie do administrátorského prostredia.

3 Dekompozícia

Projekt bude rozdelený na dve časti:

1. Program pre robota.

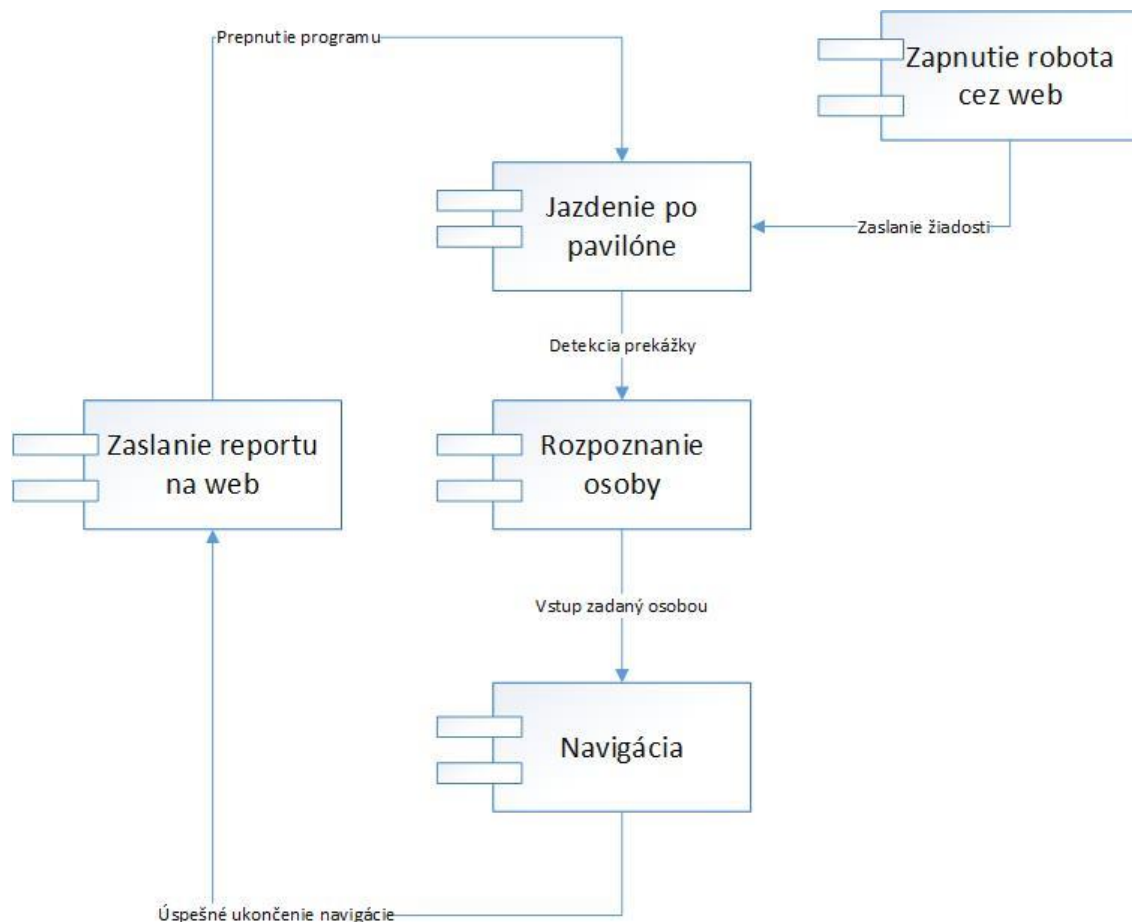
Tento program bude obsahovať triedy potrebné pre jazdu robota po pavilóne, komunikáciu s človekom, komunikáciu so serverom a vyhľadávanie miestností.

- Jazda:
Má metódy na korigovanie smeru a rýchlosti jazdy, výpočet aktuálnej pozície, kontrolu výskytu osoby alebo prekážky. Pamätá si svoju aktuálnu pozíciu a cieľ jazdy.
- Rozpoznávanie prítomnosti človeka:
Pomocou OpenCV sa spracúva výstup z kamery a reaguje na prítomnosť človeka.
- Komunikácia s človekom:
Obsahuje metódy na hlasový výstup robota, spracovanie a kontrolu vstupu a časovač.
- Komunikácia so serverom:
Metódy na odosielanie dát na server a na prijímanie dát zo serveru.

2. Webová stránka.

Na webovej stránke budú bežnému užívateľovi k dispozícii údaje o momentálnom stave robota a história jeho úloh. Admin má možnosť robota vypnúť alebo zapnúť.

3.1 Komponentový diagram



3.2 Popis komponentov

3.2.1 Komponent Zapnutie robota cez web

Komponent je potrebný pre samotný štart robota, po spustení sa zapnú senzory kamera a spustí sa aj samotný program prechádzania robota. Robot už však musí byť umiestnený v správnom pavilóne so zapnutými motormi.

3.2.2 Komponent Jazdenie po pavilóne

Po spustení robota sa zapína program ktorý zabezpečuje samotnú jazdu po pavilóne, čo zahŕňa držanie robota v strede chodby, pamätanie pozície robota, ako aj kontrolovanie prekážok pri robotovi.

3.2.3 Komponent Rozpoznanie Osoby

V tento komponent využíva predovšetkým knižnicu OpenCV, pomocou ktorej sa snaží rozpoznať objekt nachádzajúci sa pred robotom. Po zistení, že pred robotom sa nachádza osoba, komponent použije hlasový syntetizátor a osobu osloví.

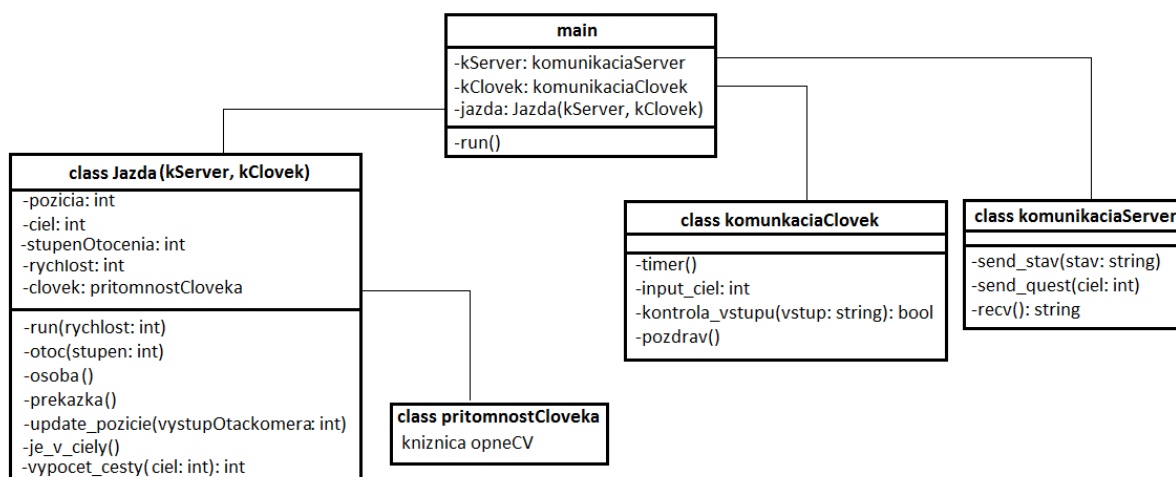
3.2.4 Komponent Navigácia

Po úspešnom zadaní vstupu od rozpoznanej osoby sa spustí modul navigácia ktorý najskôr vypočíta najkratšiu cestu k hľadanej miestnosti, vyzve osobu aby ho nasledovala a odvedie ju k hľadanému miestu.

3.2.5 Komponent Zaslania reportu

Tento komponent sa používa slúži na komunikáciu s web stránkou na ktorú po ukončení navigácie odošle informácie o navigovaní. Po odoslaní sa znova spúšťa program na jazdenie po pavilóne.

3.3 Triedny diagram



Obrázok 2 : class-diagram

3.4 Popis tried

- **Trieda Jazda**

Obsahuje metódy a premenné potrebné k ovládaniu jazdy robota a počítanie jeho polohy.

pozícia – aktuálna pozícia

ciel – aktuálny cieľ, ak sa len „prechádza“ hodnota je None

stupenOtocenia – aktuálny stupeň otočenia, na začiatku je 0

rychlost – aktuálna rýchlosť

clovek – využíva triedu pritomnostCloveka na rozpoznávanie prítomnosti osoby

run(rychlost: int) – uvedie robota do pohybu pod aktuálnym uhlom natočenia

rychlosťou ktorú dostane v argumente

otoc(stupen: int) – otočí robota o daný stupeň, tento stupeň si pamätá v premenej stupenOtocenia

osoba() – kontroluje, či sa pred robotom nachádza človek

prekazka() – udržuje robota v strede chodby a kontroluje výskyt prekážok, koriguje jazdu pomocou otoc

update_pozicie() – podľa výstupu z otáčkomera a zapamätaných hodnôt uhlu a rýchlosti zapisuje do premenej pozicia aktuálnu pozíciu

je_v_cieli() – kontroluje, či je robot v cieľi

vypocet_cesty(ciel: int):int – aktualizuje cieľ a podľa potreby robota otočí o 180° alebo o 0°

- **Trieda komunikaciaClovek**

Obsahuje metódy na komunikáciu s človekom, ako aj na zadanie vstupu.

pozdrav() – spustí sa hlasová nahrávka s pozdravom

timer() – ak do 30 sekúnd robot nedostane vstup, pokračuje v jazde

input_ciel() – čaká na zadanie cieľovej miestnosti

kontrola_vstupu(vstup) – kontroluje korektnosť zadaného cieľa

- **Trieda komunikaciaServer**

Trieda slúži na komunikáciu so serverom.

send_stav(stav: string) – odošle na server informáciu o aktuálnom stave

send_quest(quest: string) – odošle na server informáciu o aktuálnej úlohe

recv() – prijíma inštrukcie zo serveru

- **Main**

Metódou **run()** sa bude spracúvať výstup robota(údaje zo senzorov, otáčkometra...).