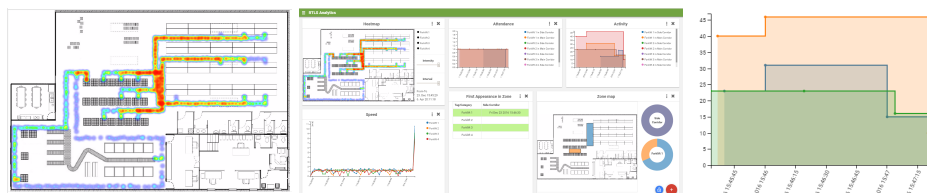


Analytický ovládací panel pro lokální systém

Lukáš Petrovič*



Abstrakt

Cieľom tejto práce je vytvoriť aplikáciu, ktorá umožňuje vizualizovať historické dáta z polohových senzorov. Dáta sú získavané z uzavrených priestorov, akými sú napríklad obchody, telocvične alebo haly. Tieto dáta sú cenné, no bez ľahko použiteľnej vizualizácie strácajú svoj potenciál. Správna vizualizácia umožňuje užívateľovi lepšie vyhodnotiť situáciu, ktorú tieto dáta reprezentujú.

Výsledkom tejto práce je nástroj, s ktorým je užívateľ schopný jednoducho analyzovať dáta, získané v určitom časovom intervale. Pre potreby analýzy poskytuje aplikácia viacero metrík, ktoré priradujú týmto dátam špecifický účel.

Dáta môžu byť reprezentované v grafoch, napríklad za účelom reprezentovať celkovú prejdenu vzdialenosť lokalizovaného objektu. Taktiež môžu byť použité v mapách, ktoré poskytujú dobrú reprezentáciu polôh vzhľadom k monitorovanej miestnosti. Všetky vytvorené vizualizačné prvky sú vložené do interaktívnych blokov. Tieto bloky tvoria takmer celý ovládací panel. S týmito blokmi je možné jednoducho manipulovať a s ich pomocou upravovať vytvorenú vizualizáciu.

Kľúčové slová: Dashboard — RTLS — Ovládací panel — Lokalizácia — Vizualizácia

Priložené materiály: N/A

*xpetro11@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

Presné lokalizačné dáta majú v dnešnej dobe veľkú hodnotu. Lokalizácia v reálnom čase, teda RTLS (z angl. Real-time locating system) firmy [Sewio Networks](#), sa zameriava na zber dát z objektov a spracovanie týchto dát. Tieto objekty môžu reprezentovať ľudí, stroje, alebo tovar. Reálne je tento systém používaný napríklad pre monitorovanie hráčov basketbalu. V situácii kedy sú k dispozícii presné a spoľahlivé polohy, je ďalším prirodzeným štádiom analýza týchto dát. Cieľom aplikácie je teda spoľahlivo reprezentovať zozbierané dáta.

Aplikácia poskytuje užívateľovi viacero spôsobov, akými môže nadobudnuté dáta reprezentovať. Dáta môžu byť reprezentované v grafoch udávajúcich celkovú prejdenu vzdialenosť alebo v mapách, ktoré priamo zo-

brazujú polohu objektu vzhľadom ku miestnosti. Tieto prvky sú oddelené do interaktívnych blokov, s ktorými je možné jednoducho manipulovať. Dôraz je kladený ale aj na priamočiare ovládanie a prehľadný dizajn.

Existujúcich analytických riešení je hneď niekoľko. Zväčša sa ale viažu na vlastný lokalizačný systém. Podobné riešenia sú často príliš komplexné. Tieto aplikácie majú často pevnú štruktúru a tým môžu obmedzovať užívateľa. Dôvod prečo nie sú použité už fungujúce generické riešenia ovládacieho panelu je aj skutočnosť, že aplikácia bude rozširovaná o ďalšie nástroje. Medzi ne patrí napríklad nástroj pre správu kategórií alebo editor zón v monitorovaných priestoroch. Niektoré riešenia sa zameriavajú na špecifickú oblasť použitia. Ako príklad je možné uviesť ovládací panel

spoločnosti Locatible¹. Ten cieľ na monitorovanie pacientov v zdravotníckych zariadeniach. Je teda špecificky zameraný na obmedzenú oblasť použitia.

Vyvíjaná aplikácia sa oproti tomu zameriava na široký rozsah použitia. Pre čo najširšie použitie je stavaná na webovej platforme. Užívateľ pri prvom kontakte s aplikáciou uvidí takmer prázdnu plochu. Tým je naznačené, že celá aplikácia sa prispôbuje užívateľovi. Následne si užívateľ vytvára jednotlivé prvky ovládacieho panelu a týmto prvkom priraduje vlastnosti. Volí si metriku zobrazenia dát a objekty, ktoré chce zobrazovať. Ovládací panel je taktiež schopný združovať jednotlivé monitorované objekty do skupín. To je možné využiť napríklad pre výpočet priemernej rýchlosti všetkých objektov v skupine. Metriky následne umožňujú súčasne zobrazovať jednotlivé objekty a skupiny. Dôležitú úlohu zohráva časový interval, ktorý ohraničuje výslednú vizualizáciu. Tento interval ale nie je vyžadovaný. To umožňuje analýzu dát, ktoré sú staré len niekoľko sekúnd.

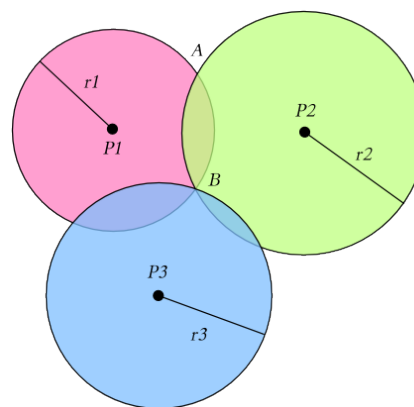
Aplikácia spoľahlivo vizualizuje polohové dáta a je ju možné jednoducho používať. Aplikáciu je možné nasadiť do prostredia intranetu a je ju možné ovládať na desktope alebo tablete bez nutnosti inštalácie špecializovaného softwaru na tieto zariadenia.

Následujúca kapitola popisuje fungovanie systému, ktorý sa stará o lokalizovanie objektov. V kapitole 3 sú objasnené dôvody a výhody výberu webovej platformy pre účely ovládacieho panelu. V kapitole 4 sú popísané možné formy vizualizácie lokalizačných dát.

2. Lokalizácia v reálnom čase

Systém lokalizácie v reálnom čase, teda RTLS sa zameriava na zber dát z objektov a spracovanie týchto dát. Spracovanie zahŕňa odstránenie nepresností v dátach, uloženie týchto dát a prezentáciu zozbieraných dát. Táto práca zabezpečuje poslednú menovanú funkciu. Využitie RTLS technológie je rozsiahle a spadajú doň mimo iné aj oblasti obchodu, zdravotníctva, armády a logistiky.

Architektúra RTLS systému sa skladá z niekoľkých prvkov. Týmto prvkami sú tagy, kotvy, jadro lokalizácie, middleware a aplikačný software. Tag je zariadenie s lokalizačnou technológiou. Využíva bezdrôtovú komunikáciu s polohovými senzormi. Na základe tejto komunikácie je vypočítavaná poloha tohto tagu a výsledné dáta sú uložené do databázy. Tieto zariadenia musia mať čo najmenšiu hmotnosť a veľkosť, a to najmä v prípade, že budú využívané pre lokalizáciu osôb. Tagy je možné integrovať do rôznych zariadení, ako sú notebooky, náramky a iné. Polohové senzory,



Obrázok 1. Obrázok² znázorňuje tri prijímače (body P1, P2, P3) a jeden vysielateľ (bod B), ktorého poloha je zisťovaná. Okolo prijímačov sú znázornené kružnice, ktorých polomer naznačuje vzdialenosť od tagu. Výsledná informácia o polohe sa vypočíta ako prienik týchto kružníc.

inak nazývané aj ako kotvy, sú využívané na lokalizáciu tagov v priestore. Lokalizácia prebieha na základe merania vzdialenosti medzi kotvou a tagom. Táto vzdialenosť je určená časom letu signálu medzi zariadeniami, teda medzi tagom a kotvou. Alternatívny názov kotva naznačuje, že toto zariadenie je statické a často býva umiestnené na stenách monitorovanej miestnosti. Pre správnu lokalizáciu je potrebné poskytnúť jadro lokalizácie dáta aspoň z troch kotiev. Jadro lokalizácie je software, ktorý komunikuje s tagmi a kotvami. Software na základe získaných informácií vypočíta polohu tagu. Vypočítané polohy sú následne ukladané do databázy. Táto databáza je teda primárny zdroj dát pre ovládací panel.

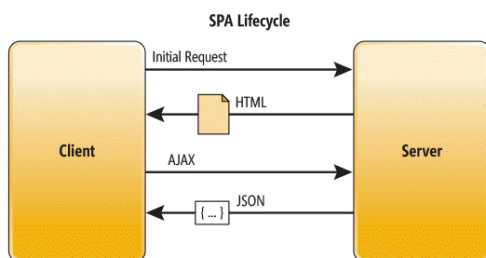
3. Web ako zobrazovacia platforma

Webová aplikácia v dnešnej dobe poskytuje rozsiahle možnosti. Pri vhodnej implementácii dokáže dokonale nahradiť aplikácie, ktoré sú priamo určené pre konkrétny systém. Taktiež sa rozširuje škála zariadení, na ktorých je možné našu aplikáciu ovládať. Výber tiež zohľadňoval existujúce aplikácie, ktoré rovnako fungujú na tejto platforme.

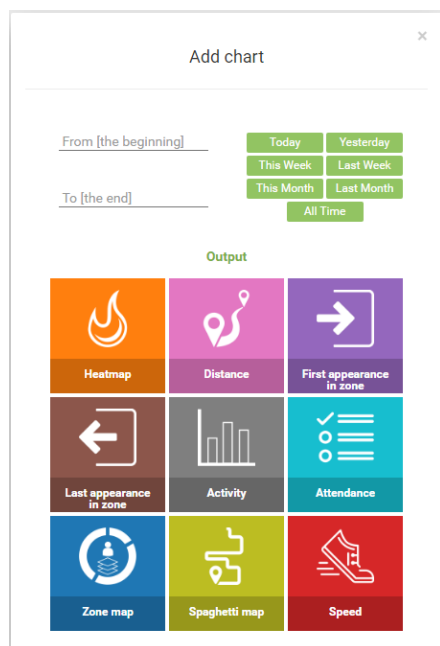
Aplikácia je založená na SPA (z angl. „single-page application“) princípe [1]. Jeho základom je, že po prvej požiadavke na aplikáciu sa klientovi pošlú všetky potrebné súbory pre vytvorenie aplikácie v prehliadači. To umožňuje aby boli jednotlivé vizualizačné prvky prepočítavané na strane klienta v reálnom čase. To poskytuje užívateľovi lepšiu kontrolu nad prvkami.

¹<http://locatible.com/>

²<https://de.wikipedia.org/wiki/Lateration#/media/File:Trilateration.png/>



Obrázok 2. Obrázok [1] popisuje životný cyklus webovej aplikácie. Po prvotnom vyžiadaní zdrojových súborov už aplikácia komunikuje so serverom len pomocou asynchrónnych požiadaviek.



Obrázok 3. Obrázok predstavuje panel, do ktorého užívateľ zadá požadované vlastnosti pre vizualizačný prvok.

Ovládací panel využíva ako zdroj dát serverové rozhranie, teda aplikáciu na strane serveru, ktorá zverejňuje prístupové body. K týmto bodom je možné pristupovať pomocou URL[2] adres. Toto rozhranie sa snaží spĺňať zásady REST[3] návrhu. Táto časť systému je taktiež súčasťou obsahu práce a zabezpečuje správne spracovanie požiadavky na dáta. Požiadavka sa skladá z metriky, časového intervalu a zo zoznamov tagov, kategórií a zón. Po spracovaní požiadavky sa požadované polohy získajú z databázy. Tieto pozície sa na základe vybranej metriky prepočítavajú. Výsledkom môže byť napríklad priemerná rýchlosť objektu vo vymedzenom čase. V tomto momente sa využívajú algoritmy, ktoré pomáhajú zredukovať objem prenášaných dát s čo najmenším dopadom na výsledné zobrazenie. Táto optimalizácia je dôležitá z dôvodu čo najmenšieho objemu prenášaných dát. Druhým dôvodom optimalizácie je lepšia odozva v prípade, keď užívateľ manipuluje s už vykresleným prvkom.

Algoritmus využíva dva parametre, s ktorými ho je možné optimálne nastaviť. Prvým parametrom je citlivosť na zmenu umiestnenia tagu v priestore. Táto citlivosť je popísaná číslom, ktoré predstavuje hraničnú vzdialenosť medzi aktuálnym bodom a bodom nasledujúcim. V prípade, že táto vzdialenosť nie je prekročená, je nasledujúci bod vynechaný z výsledku. Tým sa odfiltrujú pozície, ktoré sú príliš blízko pri sebe a nebudú mať pri vizualizácii výpovednú hodnotu. Súradnice takto vynechaných bodov sa priemerujú. Tento priemer je neskôr použitý pre zohľadnenie menších posunov v priestore. Druhým parametrom je interval v sekundách. Tento interval zaisťuje prítomnosť polohových dát vo výsledku v prípade, že tag nemení svoju pozíciu. V prípade, že tag nemení svoju pozíciu, sa po každom uplynutí tohto intervalu zapíše vyššie spomenutá priemerná pozícia do výsledku.

3.1 Interakcia s ovládacím panelom

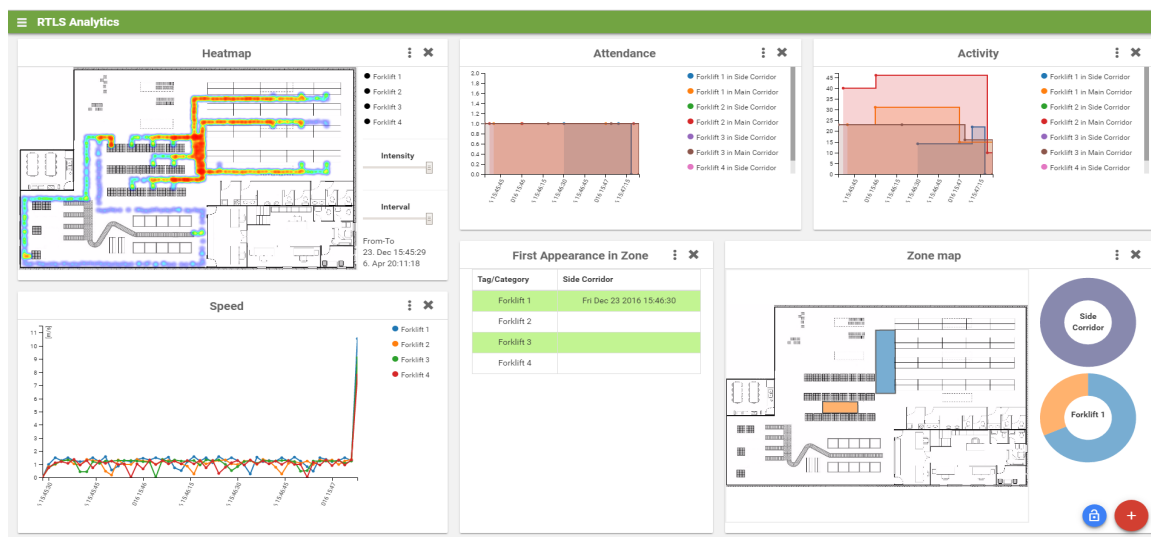
Jednou z hlavných vlastností ovládacieho panelu je schopnosť správne reagovať na užívateľove vstupy a byť flexibilný k jeho požiadavkám. V tomto prípade ide najmä o schopnosť navrhnúť vlastnú podobu ovládacieho panelu. Užívateľ zadá druh zobrazenia dát a vyberie, ktoré lokalizačné zariadenia majú byť zobrazené. Okrem týchto zariadení je možné vyberať aj zóny. Tieto zóny slúžia na špeciálne vymedzenie priestoru na mape. Aplikácia pracuje s historickými dátami, a preto je nevyhnutné, aby bola schopná tieto dáta filtrovať na základe času. O to sa stará filter, do ktorého užívateľ zadá počiatočný a koncový čas. Následne sú všetky dáta, ktoré sa zobrazia do grafu, filtrované na základe tohto obmedzenia. Obrázok 3 predstavuje rozhranie, s pomocou ktorého užívateľ zadáva svoje požiadavky aplikácii.

Na Obrázku 4 je možné vidieť vizualizačné prvky, ktoré sú vložené do interaktívnych blokov. Tieto bloky poskytujú rozhranie, ktoré rozširuje možnosti manipulácie s vizualizačným prvkom, ako napríklad duplikovanie tohto prvku.

4. Možnosti vizualizácie dát

Ako hlavnú knižnicu pre potreby zobrazenia dát bola zvolená knižnica D3.js [4]. Táto knižnica poskytuje rozhranie pre efektívnu manipuláciu s elementami na základe vstupných dát. Hlavnými oblasťami, s ktorými knižnica D3 pracuje sú HTML, CSS a SVG³. Voľba tejto knižnice priniesla výhody v podobe flexibility, čo umožňuje prispôbiť každý detail výsledného grafu. Ďalšou výhodou použitia knižnice D3 je vysoká

³https://en.wikipedia.org/wiki/Scalable_Vector_Graphics



Obrázok 4. Plne obsadený ovládací panel. Jednotlivým prvkom tohto panelu je možné meniť veľkosť a ich umiestnenie.

rýchlosť. Vďaka týmto vlastnostiam je možné s grafom manipulovať v reálnom čase a výsledné grafy sú interaktívne. Pre viac špecifickú vizualizáciu, akou je napríklad teplotná mapa, je využitá knižnica Simple-Heat.js [5].

Väčšina z nasledujúcich ukážok metrík zdieľajú rovnaký výsek dát. Tieto dáta reprezentujú polohy v prostredí skladu. Je kladený význam na fakt, že rovnaké polohy je možno reprezentovať odlišnými spôsobmi.

4.1 Tabuľková reprezentácia informácií

Tabuľka na obrázku 5 slúži na zobrazenie jednej časovej hodnoty pre jednotlivé tagy a zóny. Konkrétne je táto reprezentácia využívaná dvoma metrikami a to metrikou prvého vstupu do zóny a posledného výstupu zo zóny. Takéto zobrazenie popisuje v akom čase tag alebo skupina tagov prvýkrát vnikla do zóny. V prípade skupiny tagov je tento čas reprezentovaný prvým vniknutím niektorého z členov skupiny do tejto zóny.

Tag/Category	Main Corridor	Side Corridor
Forklift 1	Fri Dec 23 2016 15:45:40	Fri Dec 23 2016 15:46:30
Forklift 2	Fri Dec 23 2016 15:45:36	
Forklift 3	Fri Dec 23 2016 15:45:33	
Forklift 4		

Obrázok 5. Tabuľka reprezentujúca časy prvého príchodu do zóny.

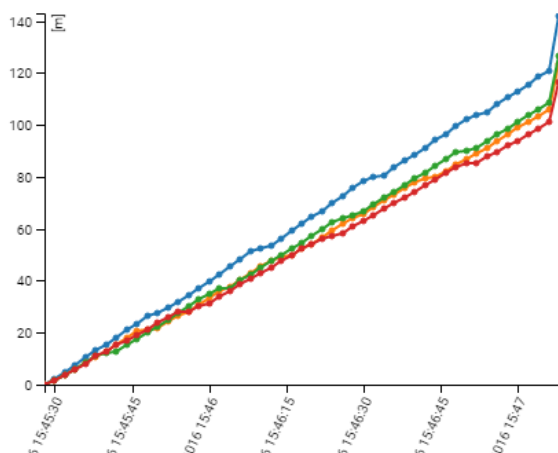
4.2 Grafové reprezentácie dát

Tieto reprezentácie sa z pravidla viažu na čas. V súčasnej dobe sú podporované štyri grafové reprezentácie dát, a to:

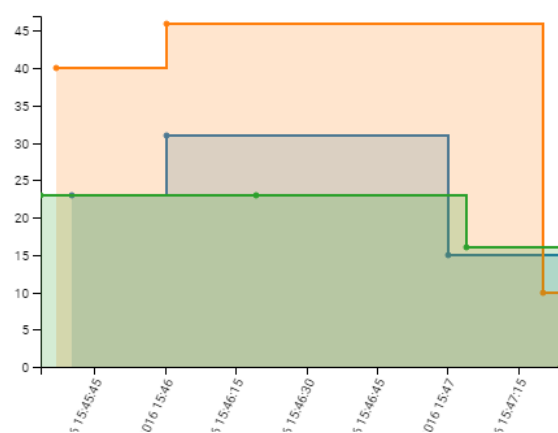
1. Výpočet celkovej prejdenej vzdialenosti
2. Zaznamenanie aktivity tagu v danej zóne
3. Zaznamenanie prítomnosti tagu v špecifickej zóne
4. Výpočet priemernej rýchlosti

V situácii, kedy potrebujeme určiť celkovú vzdialenosť, ktorú monitorovaný objekt prekonal, nastáva nasledujúci problém. Približná prejdenej vzdialenosť sa obvykle počíta ako súčet vzdialeností medzi jednotlivými bodmi z lokalizačného zariadenia. Problém ale nastáva v prípade, keď tag nenechá svoju pozíciu. V tomto prípade, získané polohy nie sú rovnaké, ale vplyvom šumu sa generujú v okolí jedného bodu. Po uplynutí niekoľkých sekúnd sú rozdiely medzi týmito chybami sčítané a môžu deklarovať prejdenu vzdialenosť. Za niekoľko sekúnd môže táto chyba dosiahnuť aj jeden meter. Existuje ale aj druhá chybná situácia, tou je skoková chyba v dátach. Takáto chyba dokáže znehodnotiť výslednú vzdialenosť. Toto môže byť spôsobené chybou merania, alebo pri strate signálu, kedy nie sú k dispozícii informácie o polohe tagu. Vtedy sa táto vzdialenosť nezapočíta do prejdenej vzdialenosti. O toto spracovanie sa stará serverová časť práce a musí na takéto situácie správne reagovať. Obrázok 6 prezentuje vizualizáciu prejdenej vzdialenosti.

Medzi ďalšie užitočné reprezentácie patrí zaznamenávanie aktivity tagu v danom čase. Obrázok 7 prezentuje túto metriku, s jej pomocou je možné odhaliť aktivitu, na rozdiel od predošlej metriky, aj v prípade,



Obrázok 6. Graf reprezentujúci rovnomerný nárast prejdenej vzdialenosti v čase.

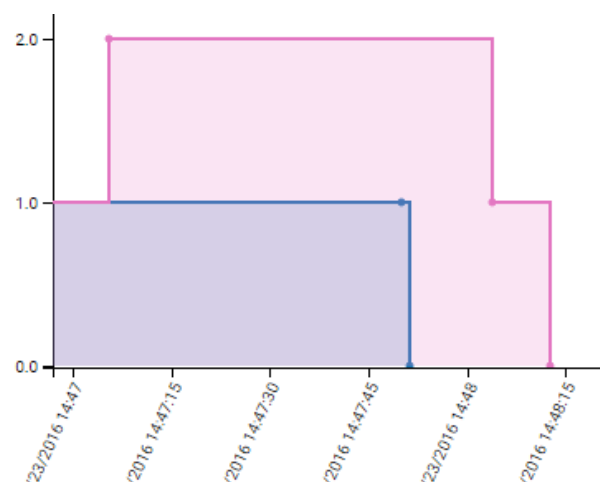


Obrázok 7. Graf reprezentujúci aktivitu tagu v závislosti na čase. Vertikálna os popisuje počet zozbieraných pozícií.

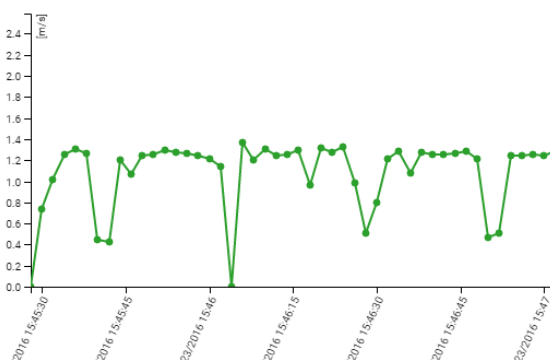
že tag nemení svoju pozíciu v priestore ale zároveň nie je statický. Takýto jemný pohyb zaznamenáva akcelerometer, ktorý je vstavaný v lokalizačnom zariadení. Takúto aktivitu je následne možné filtrovať na základe vybraných zón.

Na obrázku 8 je možné vidieť metriku prítomnosti tagov v špecifických zónach. Tá popisuje koľko zariadení je prítomných v špecifickej zóne v danom čase. Vďaka prehľadnosti grafu je veľmi jednoduché porovnávať obsadenie v jednotlivých zónach.

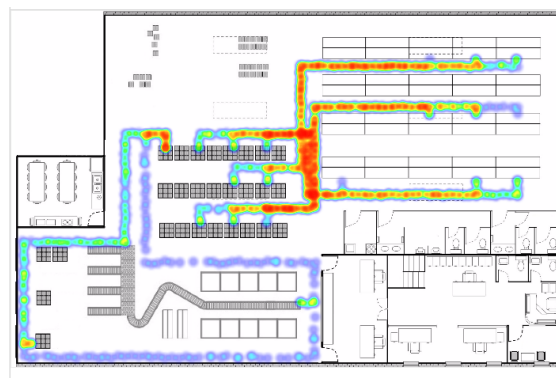
Metrika výpočtu priemernej rýchlosti, reprezentovaná obrázkom 9, rozdeľuje vybraný časový interval na konštantný počet rovnakých časových úsekov. Veľkosť tohto úseku je priamo úmerná veľkosti vybraného časového intervalu. V každom tomto úseku je vypočítaná priemerná rýchlosť. Tieto úseky sú následne vykreslené do grafu, kde užívateľ sleduje zmenu tejto priemernej rýchlosti v čase. V prípade výberu celej kategórie je výsledkom celkový priemer priemerných rýchlostí jednotlivých tagov v tejto kategórii.



Obrázok 8. Zobrazenie počtu jednotlivých monitorovaných objektov v zónach. Pomocou tohto grafu môžeme zistiť obsadenie danej zóny v konkrétnom čase.



Obrázok 9. Graf popisuje priemernú rýchlosť monitorovaného tagu.

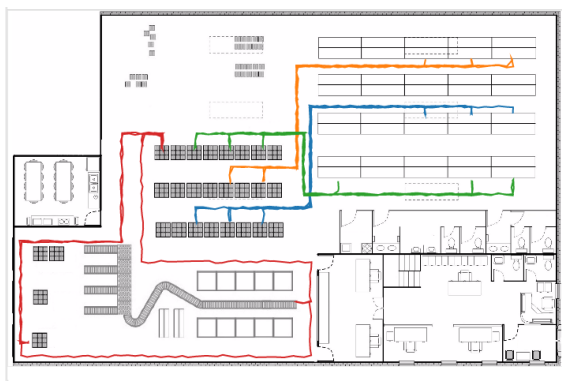


Obrázok 10. Teplotná mapa - pomocou zmeny farby, je mapa schopná vizualizovať miesta, v ktorých sa jednotlivé tagy pohybovali najčastejšie.

4.3 Mapové reprezentácie dát

Metriky používajúce mapy poskytujú dobrú reprezentáciu polôh vzhľadom k monitorovanej miestnosti. S týmito mapami je možné interaktívne manipulovať. Je možné schovávať zobrazenie jednotlivých vykreslených ciest bez potreby odkazovania sa na server.

Na obrázku 10 je možné vidieť teplotnú mapu. Používa sa pre zobrazenie polôh na dvojrozmernej



Obrázok 11. Špagety diagram (angl. spaghetti graph), vizualizuje cesty, po ktorých sa jednotlivé tagy pohybovali. Každému tagu je priradená špecifická farba cesty.

mape. Polohy sú reprezentované pomocou bodov. Čím viac bodov je umiestnených na jednom mieste, alebo v tesnej blízkosti, tým viac sa mení farba v oblasti tohto miesta. Postupný prechod farieb z modrej až k červenej znázorňuje, kde sa jednotlivý tag najčastejšie pohyboval. Priamo integrované ovládacie rozhranie prvku umožňuje určovať veľkosť oblasti, v ktorej sa jednotlivé body ovplyvňujú. Taktiež je možné upravovať už vykreslené prvky na základe času. Týmto je možné simulovať pohyb vykreslených polôh.

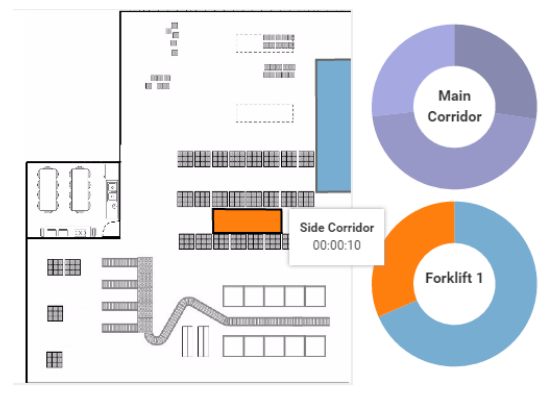
Graf ciest nazývaný aj ako špagety diagram (angl. spaghetti graph), vizualizuje cesty, po ktorých sa jednotlivé tagy pohybovali. Tagy je opäť možné združovať do skupín. Tento druh vizualizácie je zobrazený na obrázku 11.

Mapa zón slúži na porovnanie výskytu jednotlivých tagov v zónach. Užívateľ vidí jednotlivé zóny na mape. Po kliknutí na zónu sa mu zobrazí koláčový graf reprezentujúci čas všetkých tagov, ktoré sa vo vybranom časovom intervale nachádzali v tejto zóne. Popis obrázku 12 bližšie špecifikuje možnosti použitia tejto vizualizácie.

5. Záver

Lokalizácia v reálnom čase sa nezaobíde bez kvalitnej analýzy jej výsledkov. Výsledný ovládací panel má nasmerované byť univerzálnym nástrojom na analýzu dát v rôznych prostrediach. Aplikácia úzko spolupracuje s presným RTLS firmy [Sewio Networks](#). V prvotnej fáze projektu som sa s týmto systémom dobre oboznámil. Následne prebiehal výber vhodnej kombinácie knižníc, pre vytvorenie kvalitnej generickej webovej aplikácie. Implementácii predchádzalo podrobnejšie oboznámenie sa s vybranými knižnicami. Neskôr som ovládací panel v spolupráci s firmou Sewio Networks začal rozvíjať až do popisovanej podoby.

Aplikácia je schopná bez problémov vizualizovať



Obrázok 12. Interaktívna mapa zobrazujúca zóny. Po kliknutí na zónu sa zobrazí koláčový graf zachytávajúci strávený čas jednotlivých tagov v tejto zóne. Po kliknutí na výše reprezentujúcu tag, sa vytvorí druhý graf pre tento konkrétny tag. Tento druhý koláčový graf znázorňuje v akých zónach sa vybraný tag pohyboval.

dáta z desiatok lokalizačných zariadení. Výkon aplikácie ale ovplyvňuje celkový počet polôh, s ktorými je potrebné v reálnom čase manipulovať. Prekresľovanie s určitým oneskorením nastáva u mapových vizualizácií. Toto oneskorenie nastáva pri práci s viac než 100 000 polohami. Výkonnostné obmedzenie nastáva aj v aplikácii na strane servera. Pri spracovaní rozsiahlych dát sa predlžuje čakacia doba. Optimalizácia je teda nevyhnutná a do úvahy prichádza posielanie vypočítaných dát po častiach. Užívateľovi sa tak bude výsledná vizualizácia postupne rozširovať.

Aplikácia sa v budúcnosti plánuje ďalej rozširovať. V pláne je pridávanie nových metrick, vylepšenie existujúcich či vytvorenie nástroja pre komplexnejšiu editáciu zón. Žiadanou funkcionalitou je aj možnosť automatického porovnávania informácií medzi jednotlivými prvkami vizualizácie.

Osobne som si skúsil reálne testovanie lokalizačného systému. Pohyboval som sa v miestnosti s lokalizačným senzorom a zaznamenával som si svoj pohyb vzhľadom ku miestnosti. Informácie o pohybe a prejdenej vzdialenosti som následne porovnával s vizualizáciou ovládacieho panelu. Ovládací panel spĺňal všetky požiadavky, ktoré boli naň kladené.

Pod'akovanie

Chcel by som poďakovať vedúcemu mojej práce Ing. Vladimírovi Veselému, Ph.D. za nasmerovanie a kontrolu postupu pri vypracovaní tejto práce. Poďakovanie patrí aj Ing. Ľubomírovi Mrázovi a Ing. Tomášovi Kočanovi za technické vedenie projektu.

Literatúra

- [1] Mike Wasson. Asp.net - single-page applications: Build modern, responsive web apps with asp.net. blogpost, Nov 2013. <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>.
- [2] M. McCahill T. Berners-Lee, L. Masinter. Uniform resource locators. rfc, December 1994. <https://tools.ietf.org/html/rfc1738>.
- [3] Alex Rodriguez. Restful web services: The basics. blogpost, Nov 2008. <https://www.ibm.com/developerworks/webservices/library/ws-restful/>.
- [4] Mike Bostock. D3.js. library. <https://d3js.org/>.
- [5] Vladimir Agafonkin. Simpleheat.js. library. <https://github.com/mourner/simpleheat>.