

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství
Obor: Aplikace softwarového inženýrství



Datový sklad s aplikací v železniční dopravě

Data warehouse with application in railway transportation

DIPLOMOVÁ PRÁCE

Vypracoval: Bc. Martin Švamberg
Vedoucí práce: Ing. Tomáš Liška, Ph.D.
Rok: 2022

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Akademický rok 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Martin Švamberg

Studijní program: Aplikace přírodních věd

Obor: Aplikace softwarového inženýrství

Název práce česky: Datový sklad s aplikací v železniční dopravě

Název práce anglicky: Data warehouse with application in railway transportation

Pokyny pro vypracování:

1. Seznamte se s problematikou datového skladu v souvislosti s oblastí zpracování velkých objemů dat. Popište z hlediska teorie proces ETL.
2. Analyzujte a popište oblast aplikace datového skladu a technologie s ní spojené.
3. Navrhněte architekturu datového skladu, popište procesy extrakce, transformace a načtení dat. Zhodnoťte vhodné způsoby vizualizace související s podstatou dat ze zvolené oblasti. Posuďte, zda je možné zpracovávat data v reálném čase, případně navrhněte vhodnou alternativu.

Doporučená literatura:

- [1] NOVÝ, J. *Zpracování velkých objemů dat v rámci fyzikálního experimentu COMPASS*, disertační práce, ČVUT FJFI, 2019
- [2] ŽABA, A. *Datový sklad s aplikací v letecké dopravě*, diplomová práce, ČVUT FJFI, 2017
- [3] LIŠKA, T. *The Cluster and GRID Computing on Mass Data Production Systems for High Energy Physics Experiments*, CERN, ČVUT FJFI, 2009
- [4] DEVLIN, B. *Data warehouse: from architecture to implementation*. Reading, Mass: Addison-Wesley, c1997. ISBN 0201964252.

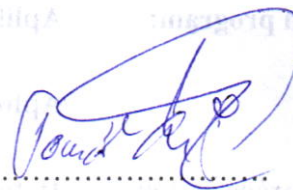
Jméno a pracoviště vedoucího práce:

Ing. Tomáš Liška, Ph.D.

Katedra softwarového inženýrství, Fakulta jaderná a fyzikálně inženýrská, ČVUT v Praze

Jméno a pracoviště konzultanta:

—



vedoucí práce

Datum zadání diplomové práce: 08. 10. 2021

Termín odevzdání diplomové práce: 2. 5. 2022

Doba platnosti zadání je dva roky od data zadání.

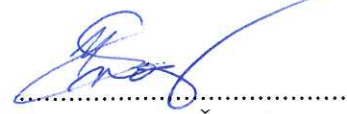

.....
garant oboru
.....
vedoucí katedry
.....
děkan

V Praze dne 08 - 10 - 2021

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 3.1.2023



Bc. Martin Švamberg

Poděkování

Děkuji vedoucímu práce Ing. Tomáši Liškovi, Ph.D. za svědomité vedení mé diplomové práce a neocenitelné rady při její tvorbě. Dále děkuji svým kolegům za skvělou spolupráci a pečlivost na projektu PP, který byl tématem praktické části této diplomové práce.

Bc. Martin Švamberg

Název práce:

Datový sklad s aplikací v železniční dopravě

Autor: Bc. Martin Švamberg

Studijní program: Aplikace přírodních věd

Obor: Aplikace softwarového inženýrství

Druh práce: Diplomová práce

Vedoucí práce: Ing. Tomáš Liška, Ph.D.

Katedra softwarového inženýrství, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

Konzultant: –

Abstrakt: V této diplomové práci se zabýváme návrhem a integrací zcela nového modulu zpracování dat prostorové průchodnosti a jejich vizualizací do již v praxi provozovaného datového skladu v oboru železniční dopravy. K novému modulu se chováme jako k samostatnému datovému skladu, přičemž ale máme stále na paměti jeho vazby k předešlým modulům, čímž zachováváme soudržnost celého robustního datového skladu. V úvodních kapitolách si popisujeme stěžejní aspekty datového skladu, včetně významného procesu ETL. Navazujeme teorií prostorové průchodnosti podstatné pro zpracování nového modulu a na závěr představujeme dokončený modul prostorové průchodnosti.

Klíčová slova: Datový sklad, ETL, železniční doprava, prostorová průchodnost, průjezdný průřez

Title:

Data warehouse with application in railway transportation

Author: Bc. Martin Švamberg

Abstract: In this master thesis we focus on designing and integration of entirely new modul of processing and visualization of spatial clearance data into a in practice operated DataWarehouse in railway transportation field. We work with new modul as it would be standalone DataWarehouse, but we keep in mind it's connections with other modules, thus keeping cohesion with the whole robust DataWarehouse. In introductory chapters we describe crucial aspects of DataWarehouse including significant process ETL. We continue with theory of spatial clearance, vital for processing of new modul and as conclusion of thesis we present finished modul of spatial clearance.

Key words: DataWarehouse, ETL, railway transportation, spatial clearance, structure gauge

Obsah

Úvod	11
1 Datový sklad	13
1.1 Úvod	13
1.2 Představení datového skladu	13
1.2.1 Výstavba datového skladu	14
1.3 Proces ETL	15
1.3.1 Extrakce dat	16
1.3.2 Transformace dat	17
1.3.3 Načtení dat do datového skladu	19
2 Teorie průjezdného profilu	21
2.1 Prostorová průchodnost	21
2.1.1 Průjezdný průřez a prostorové uspořádání	22
2.1.2 Referenční profil	23
2.1.3 Pojistný prostor	24
2.2 Měření prostorové průchodnosti	25
2.2.1 Malý měřicí systém	25
2.2.2 Fotogrametrický stroj	26
3 Návrh a implementace datové skladu	29
3.1 Úvod	29
3.2 Import dat	29
3.2.1 Automatizovaný skript	30
3.2.2 Importovaná data	31
3.2.3 Postup automatizovaného skriptu	31
3.2.4 Import úsekových měření	33
3.2.5 Import dat referenčního profilu	33
3.2.6 Import dat subprofilů	36
3.2.7 Import dat překročení	36
3.3 Návrh uživatelského rozhraní	38
3.3.1 Webová aplikace	38
3.3.2 Vyhledávání naměřených úseků	38
3.3.3 Vizualizace dat	43
3.4 Implementace zobrazení čelních fotek	44
3.5 Implementace vykreslení bodových měření a referenčního profilu na canvas	45

3.5.1	Načítání a vykreslení dat	46
3.5.2	Vyhodnocení bodů překročení - první verze	48
3.5.3	Vyhodnocení bodů překročení - druhá verze	49
3.5.4	Operace s referenčním profilem	52
3.5.5	Exporty canvasu	52
3.5.6	Ovládací prvky	55
3.5.7	Měření vzdáleností na canvasu	56
3.6	Simulace jízdy	57
3.7	Implementace grafu celého úseku	58
3.7.1	Knihovna Highcharts	58
3.7.2	Příprava dat v PHP	58
3.7.3	Zobrazení dat v javascriptu	61
4	Ukázky vizualizace	67
4.1	Čelní fotografie	67
4.2	Zobrazení bodového měření na canvas	69
4.2.1	Referenční profil	69
4.2.2	Ovládání canvasu	72
4.2.3	Měření vzdáleností na grafu	74
4.3	Simulace jízdy	74
4.4	Graf měřeného úseku	76
	Závěr	79
	Literatura	80

Úvod

V této práci se zabýváme teorií datového skladu a jeho následnou aplikací nad reálnými provozními daty z vybraného technologického odvětví. Zvoleným odvětvím praktické části je železniční průmysl, konkrétně prostorová průchodnost železničních vozidel na české železniční dráze.

Přestože nevytváříme zcela nový podnikový datový sklad a rozšiřujeme již existující, přistupujeme k rozšiřujícímu modulu jako k samostatnému datovému skladu s ohlednutím na okolní vazby pro zachování celkové integrity.

V první kapitole se věnujeme teoretickým východiskům a principům konstrukce datových skladů a jim příslušných procesů, na které později referujeme při tvorbě návrhu nového rozšíření datového skladu. Kapitola připomíná a rozšiřuje již nashromážděné poznatky o datovém skladu, jež byly sepsány ve výzkumném úkolu, na téma zpracování velkých objemů dat, předcházejícímu této práci.

Poznatky o měření dat prostorové průchodnosti pravidelně prováděné měřícími vozy jsou obsaženy v druhé kapitole této práce. Přiblížíme si v ní teorii a stěžejní pojmy, bez kterých by nebylo možné sestavit správně fungující podnikový nástroj, využívaný experty ve svém oboru.

Ve třetí kapitole se zabýváme samotným návrhem rozšíření datového skladu. Kapitola zahrnuje nejen popis zpracování dat pravidelně proudících do systému, ale také postup vývoje jednotlivých nástrojů pro práci s daty.

Práci uzavíráme ukázkami vizualizace zpracovaných dat v několika různých podobách, včetně ukázek funkčnosti sestavených nástrojů.

Kapitola 1

Datový sklad

1.1 Úvod

V návaznosti na poznatky o zpracování velkých objemů dat, obsažených ve výzkumném úkolu viz. [zdroj], v této kapitole navazujeme a rozšiřujeme jeden z hlavních pojmů souvisejících s tímto tématem - datový sklad.

Uvedeme si zde principy a procesy, které jsou jádrem každého datového skladu, bez ohledu na to, jak je obsáhlý. Hlavním procesem, který je při zmínce o datovém skladu vždy uveden, je proces ETL (Extract, Transform, Load). Tomuto procesu budeme věnovat značnou část této kapitoly.

1.2 Představení datového skladu

Zpracování velkých objemů dat vždy vyžaduje kompletní hloubkovou a detailní analýzu. Vznikly tak architektury, jejichž účelem je uskladnění strukturovaných či nestrukturovaných dat. Právě jednou takovou architekturou je datový sklad.

Obecně můžeme definovat datový sklad jako uložistiště a nástroj pro správu strukturovaných dat. Korektně zprostředkovanou správu svých podnikových dat potřebují větší i menší společnosti, jelikož je pro ně na trhu s početnou konkurencí velmi důležité mít přehled o svých datových aktivech.

Vzniká tak pojem „podnikový datový sklad“, který by měl představovat jedno unifikované spolehlivé úschovné místo všech historických podnikových dat společnosti. Využití podnikového datového skladu nicméně nekončí jen u obchodních společností, u kterých si dokážeme jednoduše představit využití ve vizualizaci statistik prodeje produktů a služeb.

Podoba a obsah podnikového datového skladu záleží čistě na oblasti působení společnosti. Využití v průmyslových odvětvích má taktéž nezměrné využití a výstavba takového podnikového skladu pak může být značně komplexní.

Hlavním důvodem výstavby podnikového datového skladu je získání co nejkvalitněj-

ších dat pro Business intelligence. Čím více různorodých podnikových zdrojů dat ve firmě figuruje, tím těžší bývá je správně sloučit a data pro podnikovou analýzu se pak stávají nekonzistentní a vzniká nestrukturovaná skupina dat, která se blíží definici datové bažiny (viz výzkumný úkol zdroj [8]).

Pro odstranění těchto problémů a získání přehledných strukturovaných dat, nad kterými lze provádět efektivní operace, je navržen právě centralizovaný datový sklad správně slučující data, nehledě na zdroji jejich původu.

1.2.1 Výstavba datového skladu

Datový sklad vznikl původně okolo databázových prostředí a tak i po jeho růstu do komplexního podnikového systému jsou jeho jádrem strukturované databáze, ve kterých jsou data uchovávána.

Co datový sklad dělí od pouhé databáze jsou nástroje, které zprostředkovávají extrakci dat z jednotlivých podnikových systémů, jejich čištění, kontrolu a případnou datovou transformaci pro zvýšení jejich kvality.

Budování datového skladu jako takového lze rozdělit do několika fází, které se po určité době s rozšířením opakují:

- Zadání projektu společností
- Návrh a modelování
- Samotná výstavba
- Testování koncovým uživatelem

Zadání projektu společností

Každý datový sklad se buduje na základě vytvoření projektu od společnosti. V této fázi se shrnují potřeby podniku a možný rozpočet pro realizaci. Z rozpočtových důvodů a technických důvodů se pak základní projekt postupně rozšiřuje.

I na nový rozšiřující modul datového skladu pak můžeme hledět jako na samotný datový sklad, jelikož se proces jeho vystavení schoduje z hlediska realizace s procesem výstavby celkového datového skladu.

Návrh a modelování

Po schválení rozpočtů a potřeb podniku přichází na řadu fáze návrhu a modelování. V této fázi je stěžejní kooperace mezi koncovým uživatelem a vývojovým týmem. Je zde zapotřebí určitého porozumnění oblasti podniku pro lepší pochopení jeho potřeb a následného zohlednění při vytváření návrhu.

Kvalitnější znalost oblasti pak vede ke správnější realizaci požadavků, které nemusí být na první pohled zcela jednoznačné a mohou způsobit jistá nedorozumění vedoucí na úpravy návrhu i v pozdějších fázích.

Samotná výstavba

Jako další následuje realizace výstavby datového skladu, která zahrnuje veškeré programovací úkony, jakými jsou například zakládání databáze, programování procesu ETL a vytváření grafického uživatelského rozhraní.

Testování koncovým uživatelem

Po dokončení implementace systém postupně zpřístupňujeme koncovému uživateli za účelem testování. I v této fázi je kooperace mezi dvěma stranami (vývojový tým a uživatel) stěžejním aspektem vývoje datového skladu.

Jedná se především o ladění a úpravy na základě zpětné vazby, které však nepřekračují původní hranice dohodnutého projektu. Některé úpravy mohou být nad rámec původní návrhu (finančně či realizačně) a je potřeba pro ně vytvořit projekt nový.

1.3 Proces ETL

ETL (Extract, Transform, Load) je proces, který nejprve extrahuje data z různých zdrojů, např. operační systémy, databáze, externí uložení dat. Následně data prochází transformací, která může zahrnovat určité kalkulace, filtrace apod. a ve finále takto zpracovaná data ukládá do systému datového skladu.

Přestože se jedná o poměrně jednoduchou a přímočarou definici, proces samotný ve své exekuci nabírá rychle na komplexnosti a je považován za nejvíce náročnou část utváření datového skladu, ať už z technického či časového hlediska.

K jeho vyhotovení je zapotřebí aktivní spolupráce klienta i vývojového týmu. Ze strany klienta je zapotřebí představit své datové zdroje a důležité znalosti k povaze, typu a objemu dat, dle kterých pak může vývojový tým vhodně navrhnout dílčí části ETL procesu.

K udržení datového skladu jako vhodného nástroje pro business intelligence je zapotřebí ho upravovat dle potřeb podniku. S tím jde ruku v ruce i upravování ETL procesu a tak přestože je důležité, aby měl proces definované pevné základy, je zároveň potřeba sestavit ho tak, aby nové požadavky nevyústili v jeho kompletní přestavbu.

1.3.1 Extrakce dat

Extrahování dat ze zdrojových systémů za účelem obdržení jejich kopie je základním stavebním kamenem celého procesu. Výstupem této části by měla být data v takové podobě, se kterou mohou další fáze procesu snadno pracovat.

Pro umožnění podnikových analýz je důležité udržovat data v aktuálním stavu, čehož se dosahuje extrakcí v pravidelných intervalech, jejichž délka závisí čistě na rychlosti a objemu přibývání dat. Některé extrakce stačí provádět např. jednou měsíčně, jiné i každou hodinu.

Zároveň se zde musí počítat kromě různého charakteru dat na operačních systémech i se závislostí, kterou mezi sebou data mohou mít, byť až v pozdější fázi procesu. Pokud ve fázi transformace používáme data z různých zdrojů pro výpočet a kompletaci do strukturovanější potřeby, musíme zajistit, že máme všechna data opravdu k dispozici.

Tím je především myšleno, že se musí dbát na správnou koordinaci datových zdrojů, do kterých mohou data přicházet v různých časových intervalech. Intervaly nemusí být ve zdrojích z technických, či jiných důvodů mezi sebou navzájem synchronizované.

Online a offline extrakce

Extrahování dat z operačních databází spadá obecně pod dva mechanismy a to online či offline. Online extrakcí se pak myslí přímé napojení do databáze zdrojového operačního systému, ze kterého pak čerpáme data.

Offline extrakce se oproti online extrakci nenapojuje přímo na zdrojový operační systém, nýbrž mimo na jiný zdrojový systém, kam jsou data pravidelně odkládána v předem definované struktuře. Může se jednat o různé typy struktur, častým příkladem je však prostý databázový soubor.

Charakteristika dat z hlediska uchování historie

Na charakter dat můžeme pohlížet z několika úhlů, jedním z nich je jejich specifikace vůči uchovávání vlastní historie, která pak vede i na různé způsoby zpracování.

- Krátkodobá data - jedná se o databázový záznam, u kterého může v nepravidelném intervalu dojít k přepsání, nebo smazání. Historie takových změn se nikde neukládá a záznam existuje vždy jen v aktuálním stavu.
- Částečně periodická data - v případě tohoto typu databázového záznamu uchováváme několik jeho posledních změn. Nejstarší záznam je pak v momentu další změny smazán, čímž udržujeme omezenou historii změn. Např. můžeme chtít držet v databázi dvě poslední adresy.

- Periodická data - v tomto případě uchováváme kompletní historii změn záznamu. Historii pak můžeme v databázi zprostředkovat např. unikátním klíčem spolu s označením verze, či využitím tzv. časového razítka (timestamp).

Charakteristika dat operačního systému a podnikového skladu

U datového skladu je podstata v něm uložených dat zpravidla periodická, jelikož chceme uchovat spojitou historii dat. Data operačního systému jsou oproti tomu z většiny krátkodobá, nebo částečně periodická. Méně častý případ nastává, když databáze operačního systému uchovává i historická data.

To musíme brát v úvahu při jejich čerpání a převodu do podnikového datového skladu, jelikož záznamy určitého typu mohou být po neurčité době přepsány či kompletně z databáze vymazány.

Zachycení změn datových záznamů pak můžeme provádět dvěma přístupy:

- Zachycení pomocí operační aplikace - tento princip spočívá v nastavení operační aplikace tak, aby kromě uchovávání dat vedla i záznam o jejich změnách (ne změny samotné). Tato funkcionality může být pro implementaci do existující operační aplikace různé náročná a je samozřejmě jednodušší ji zapracovat při zřizování zcela nové aplikace, kde se s ní může předem počítat.
- Kompletní extrakce - nazývaná také „statická extrakce“ extrahuje vždy všechna data ze zdrojového systému. Výsledkem takové extrakce je aktuální obraz zdrojové databáze, anglicky „snapshot“. Díky tomu nemusíme u dat zavádět a udržovat žádné datové atributy. Vždy máme k dispozici aktuální data, dle poslední extrakce. Kompletní extrakce se využívá zejména při iniciálním plnění datového skladu.

1.3.2 Transformace dat

Druhým krokem procesu je transformace přijatých dat v podobě, ve které byly extrahovány z různých datových zdrojů. Tuto fázi lze rozdělit na několik dílčích operací s daty, které mají za úkol data pročistit a připravit pro načtení do datového skladu.

Počet operací je libovolný a je úzce svázán s potřebami transformace dat. Může se například jednat o operace třídění, doplnění časových razítek, zavedení unikátních identifikátorů, sloučování, kontroly, různé matematické operace apod.

Čištění dat

Nekonzistentní, či chybná data se běžně vyskytují ve zdrojích, ze kterých se data extrahují. Věcnou otázkou pak je, kde a jakým způsobem by se jejich korekce, či čištění mělo provádět. Vhodně implementovaný krok transformace by měl být schopen si s většinou nekonzistentních dat poradit, nicméně měl by se tím zabývat?

Obecně se považuje úkon dodání korektních dat za něco, co by mělo být zprostředkováno již v operačním systému, jež je jejich zdrojem, pomocí vhodných modifikací a kontrol. Transformační proces by pak měl nad daty provádět úkony nesouvisející s jejich chybovostí.

Někdy může chybovost plynout už z prvotních zápisů do databáze operační systému, prostřednictvím operační aplikace. Může se jednat o prostý překlep, či zadání dat do nepřislušného sloupce, který není nijak ošetřen. Příkladem může být vepsání telefonního čísla do kolonky pro poštovní směrovací číslo.

Takto chybná data mohou narušit jinak plynulý ETL proces. Obecně se s nimi můžeme setkat v několika podobách:

- Chybějící data - jedná se o části, nebo celý záznam neobsahující hodnoty.
- Duplicity - záznam se vyskytuje ve zdroji vícekrát.
- Špatný datový typ - záznam, který by měl obsahovat např. hodnotu typu boolean, obsahující textové znaky.
- Nekonzistentní hodnoty - dva různé způsoby zápisu, které mají zastávat stejnou hodnotu. Například když jsou do sloupce geolokačních souřadnic vepsané jednou ve formátu GPS a jednou ve formátu JTSK (jednotná trigonometrická síť katastrální).

Rozdělení dat

Rozdělení dat podle určitých kritérií je nejjednodušší formou transformace. Může se jednat například o rozdělení dat do větších logických celků, které pak následně mohou v případě potřeby projít normalizací.

Normalizace dat

Normalizace dat je proces, kdy je struktura dat přeorganizována tak, aby odpovídala relačnímu modelu dat pro relační databázi. Tento proces slouží k efektivnějšímu ukládání dat, jejich prohlédávání zpracování, či třídění.

Většinou jsou zdrojová data denormalizovaná, ale existují i výjimky, kdy je zdrojová databáze již normalizovaná a její zpracování je pak mnohem jednodušší a vyžaduje méně komplexní operace.

Databáze je v praxi považovaná za normalizovanou pokud splňuje alespoň první tři normální formy, kde každá následující forma vyžaduje splnění formy předchozí. Níže si zběžně uvedeme popis jednotlivých normálových forem:

- 1NF - Každý atribut obsahuje pouze atomické hodnoty (dále nedělitelná data).
- 2NF - Každý neklíčový atribut je závislý na klíčovém atributu.

- 3NF - Neklíčové atributy jsou navzájem nezávislé.
- 4NF - Relace vyjadřuje mezi klíčem a atributy jen příčinnou souvislost.
- 5NF - Relaci není možno rozložit bezetrátově.

Konverze dat

Úzce souvisí s operací normalizace a provádí se v ní dříve zmíněné úkony upravující podstatu dat. Tyto konverze převádějící data do jiné formy jsou většinou definovány buď nějakým algoritmem, nebo „lookup tabulkou“.

Jednoduchými algoritmickými konverzemi mohou být například matematické přepočty či převod fyzikálních veličin. V případech, ve kterých konverzi není možno vyjádřit jednoduchým algoritmem se pak využívá lookup tabulka.

1.3.3 Načtení dat do datového skladu

Načtení dat do cílové databáze datového skladu je posledním krokem celého procesu ETL. Běžně se při naplňování datového skladu setkáváme s velkými objemy dat a je proto zapotřebí proces správně optimalizovat.

Zároveň v případě, kdy při naplňování dat dojde k chybě, musí být implementován mechanismus, který provede návrat do posledního funkčního stavu a neohrozí integritu stávající databáze.

V závislosti na cílové podobě dat pak můžeme rozdělit způsob načtení dat na několik typů. Tyto typy ale nemusí být využity nutně pro všechny tabulky cílové databáze a můžeme jejich kombinaci vhodně využívat. Např. pokud rozdělíme tabulky do několika kategorií, nad kterými se pak provádí určitý typ načítání dat.

Kompletní přehrazení dat

Jedná se o nejjednodušší způsob nahrání dat do databáze a využíváme ho pokud nás v cílové tabulce či tabulkách zajímají jen aktuální data a periodická pro nás nemají užitek. Stávající záznamy jsou z jedné či více tabulek smazány a kompletně přehrazeny záznamy novými.

V rámci tohoto typu zmíníme i „iniciální načtení“, které představuje prvotní naplnění všech tabulek databáze.

Přidávání dat

Jak plyne z názvu tohoto typu, jedná se i případ, kdy do tabulek cílové databáze pouze vkládáme nová data a stávající data nejsou smazána. Nevýhodou je případný vznik duplicitních dat.

Proto se tento způsob využívá v případech, kdy nová data nemohou být duplicitní a každý nový záznam je unikátní a je potřeba ho uložit. Příkladem mohou objednávky zboží, kde každá má svůj unikátní identifikátor.

Za okolností, kde nelze zajistit unikátnost dat se pak upravuje základní princip přidávání dat buď na „destruktivní režim sloučení dat“, nebo „konstruktivní režim sloučení dat“.

Destruktivní režim sloučení dat

Tento princip spočívá v přepisování stávajících záznamů novými v případě, že jsou shodné dle klíčových identifikátorů. Zbylé nové záznamy jsou pak jednoduše přidány.

Konstruktivní režim sloučení dat

Jedná se o způsob, který byl navržen pro zpracování periodických dat, zohledňující úpravy na stávajících datech. Oproti destruktivnímu režimu sloučení dat, neprobíhá přepisování existujících dat, nýbrž přidání záznamu stejně jako v principu přidávání.

Zabránění duplicit pak spočívá v přidání různých příznaků k záznamům. Nejznámějšími a nejpoužívanějšími příznaky pro odlišení dat jsou časové razítko či verze. Časová razítka mohou být již předem daná ze zdrojového systému, nebo je můžeme přidělit ve fázi transformace.

Nad časovými razítky uvažujeme jako nástrojem pro udržení historie v datech, nemusí však odpovídat realitě vzniku záznamu ve zdrojovém systému. Verze mohou vznikat stejným způsobem jako časové razítko.

Musíme si však dávat pozor při jejich používání, jelikož se jedná i o běžný datový atribut v záznamech a je hojně využíván pro označování mnoha parametrů, které nemusí mít vždy souvislost s historií dat. Bez správného označení pak může dojít ke zmatku, která verze reprezentuje data z historické povahy.

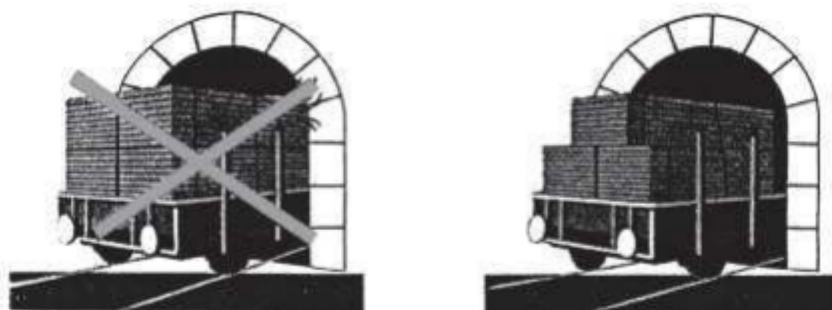
Kapitola 2

Teorie průjezdného profilu

2.1 Prostorová průchodnost

Parametr prostorové průchodnosti spadá v oboru železniční dopravy pod jednu z hlavních parametrů trati, se kterými musí být konkrétní kolejové vozidlo svými charakteristikami (délka, hmotnost, výška včetně nákladu) v souladu, aby bylo vpuštěno na trať, jinými slovy aby bylo po trati přechodné.

Pojmem prostorové průchodnosti trati chápeme jako zaručený prostor bez překážek pro volný průjezd konkrétního kolejového vozidla o jistém obrysu, ložní míře (nakládací směrnice viz. obr. 2.1) za povolené rychlosti v rychlostním pásmu tratě.



Obrázek 2.1: Ukázka rozdílů mezi dodržením a nedodržením směrodatné ložní míry. Zdroj [3], strana 20

Pravidelnou kontrolou prostorové průchodnosti trati se verifikuje zachování správných parametrů průjezdného průřezu, o kterém se dozvíme více v 2.1.1.

2.1.1 Průjezdny průřez a prostorové uspořádání

Průjezdny průřez, jehož přesnou definice nalezneme v definici 1 zastává zejména stavební význam, jelikož stanovuje polohy okolních budov a zařízení. Ať už se jedná o stavbu nových, či o přestavbu stávajících.

Za pomoci zdefinování průjezdného průřezu a prostorového uspořádání (definice 2) zajišťujeme, aby budovy a zařízení nezasahovaly do průjezdného průřezu obrazce. Obrazec v tomto případě chápeme jako kolejové vozidlo.

Znalost správného průjezdného průřezu má pak velký vliv při plánování revitalizace úseků České republiky. Nejen z hlediska rozsáhlejších oprav jako je například kompletní revitalizace nástupišť, ale také z hlediska umístování opravných zařízení na sousední (opravovanou) kolej vůči koleji aktivní.

Definice 1 *Průjezdny průřez je obrys obrazce v rovině kolmé k ose koleje, který vymezuje vzdálenosti vně ležících staveb, zařízení a předmětů od středu koleje a nad temenem kolejnic, kromě případů, kdy z funkčních důvodů musí dojít ke styku těchto zařízení s drážním vozidlem.*

Zdroj [6], strana 11, definice (1)

Definice 2 *Prostorové uspořádání označuje rozpoložení drážních i mimodrážních staveb, které mohou v určitých případech tvořit překážky prostorové průchodnosti.*

Průjezdny průřez dále můžeme rozdělit ještě do tří podkategorií dle obrysu kolejového vozidla:

- Základní průjezdny průřez - neměnný od poloměru oblouku větším jak 250m, včetně.
- Jmenovitý průjezdny průřez - za předpokladu, že nelze aplikovat základní průjezdny průřez, což může být způsobenou např. stavbami či zařízeními, které do jeho prostoru zasahují, vypočítává se dle parametrů tratě v dané části úseku průjezdny průřez jmenovitý.
- Mezní průjezdny průřez - pokud nelze použít ani jmenovitý průjezdny průřez, vypočítává se za schválení Drážního úřadu, mezní průjezdny průřez. Je stanoven dle statistického vyhodnocení pravděpodobností současných výskytů přírážek na přesah.

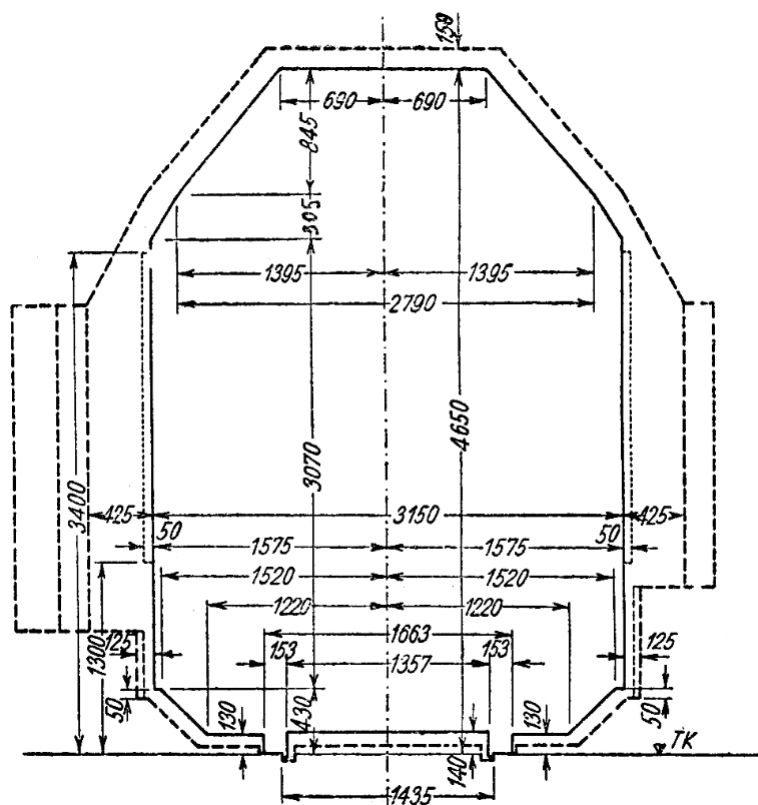
2.1.2 Referenční profil

V definicích průjezdného průřezu jsme používali sousloví *obrys obrazce*, pro definování vymežujícího prostoru, do kterého nesmí vstoupit překážky pro bezpečný průjezd kolejových vozidel.

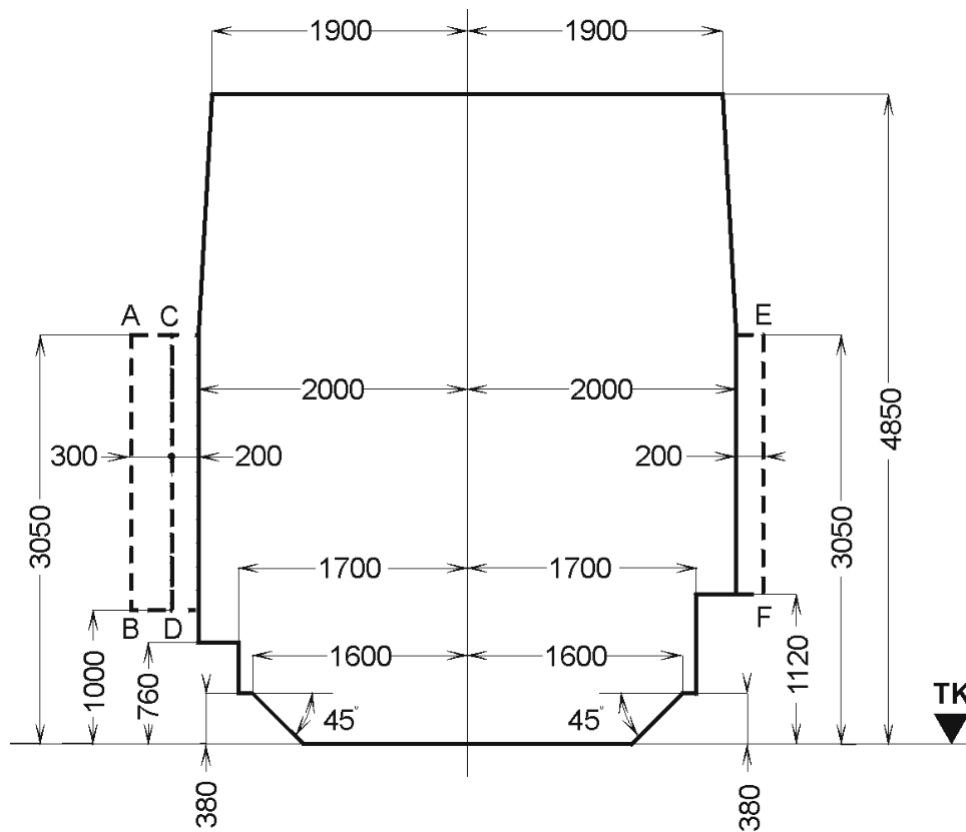
Jelikož z definice není zřejmé jak je tento vymežující prostor konstruován, rozvedeme si zde obecný princip. Průjezdný průřez je skládán ze základního obrysu kolejového vozidla, přírážek na přesah a naklonění vozidla a ve poslední řadě bezpečnostní rezervou (někdy nazývané také pojistným prostorem).

Referenční profil je pak technický název pro základní obrys kolejového vozidla. Na obrázcích 2.2 a 2.3 můžeme vidět příklad obecných referenčních profilů kolejových vozidel.

Obecných protože i přesto, že kolejové vozidlo spadá pod určitou kategorii referenčního profilu, může obsahovat speciální technologické prvky, které je při výpočtu průjezdného průřezu důležité vzít v potaz.



Obrázek 2.2: Referenční profil vozů s rozchodem 1435 mm. Zdroj [5]



Obrázek 2.3: Referenční profil Z-GC. Zdroj [5]

2.1.3 Pojistný prostor

Pojistný prostor, či bezpečnostní rezerva je dodatečný volný prostor mezi stavbami a zařízeními na trati, rozšiřující referenční profil kolejového vozidla, přidávaný z důvodů možných odchylek GPK (geometrická poloha koleje).

Jeho minimální hodnota je většinou 50mm, nicméně se může různě lišit v závislosti na vybrané části úseku.

2.2 Měření prostorové průchodnosti

Měření prostorové průchodnosti probíhá za pomoci dvou typů měřících zařízení, která si detailněji popíšeme v částech 2.2.1 a 2.2.2. Každé z nich má své výhody a nevýhody, z čehož také plynou rozhodnutí o jejich nasazení do konkrétních míst traťových úseků.

2.2.1 Malý měřicí systém

Malý měřicí systém je lehké, jednoduše manévrovatelné zařízení, vybavené rotačním laserovým skenerem pro měření prostorové průchodnosti.

Díky jeho rozměrům, jak můžeme vidět na obrázku č. 2.4, je možné ho přemístit na místo měření v opravném vozu, či v běžné dodávce.

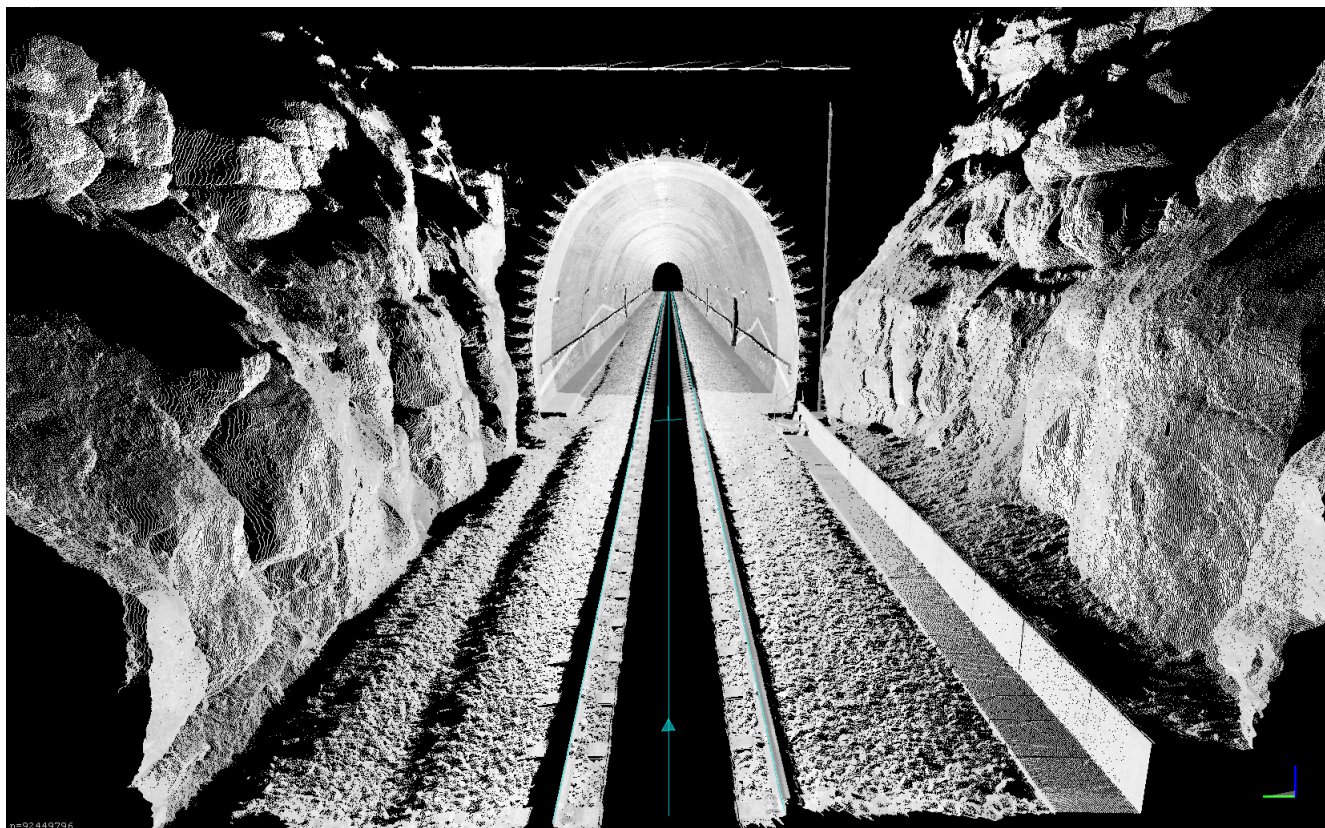
Využívá se na měření a diagnostiku kratších úseků tratě jako jsou například nástupiště, tunely, mosty apod. Často ho i můžeme vidět i při kontrole nově zrekonstruovaných úseků. Je operován pracovníkem s tabletem pro sběr dat. Skenování a měření tak probíhá rychlostí chůze.



Obrázek 2.4: Malý měřicí systém. Zdroj: <https://www.tudc.cz/wp-content/uploads/2017/05/mms2.jpg>

Rotační bodový laser skenuje prostor vertikálně, což v praxi znamená, že např. měřený tunel se musí změřit dvakrát, tj. podruhé z opačného směru. Skenování horizontálně je zprostředkováno samotnou jízdou po koleji.

Na obrázku č. 2.5 je ukázka vizualizace výsledku bodového měření malého měřicího zařízení, zpracovaného speciálním softwarem.

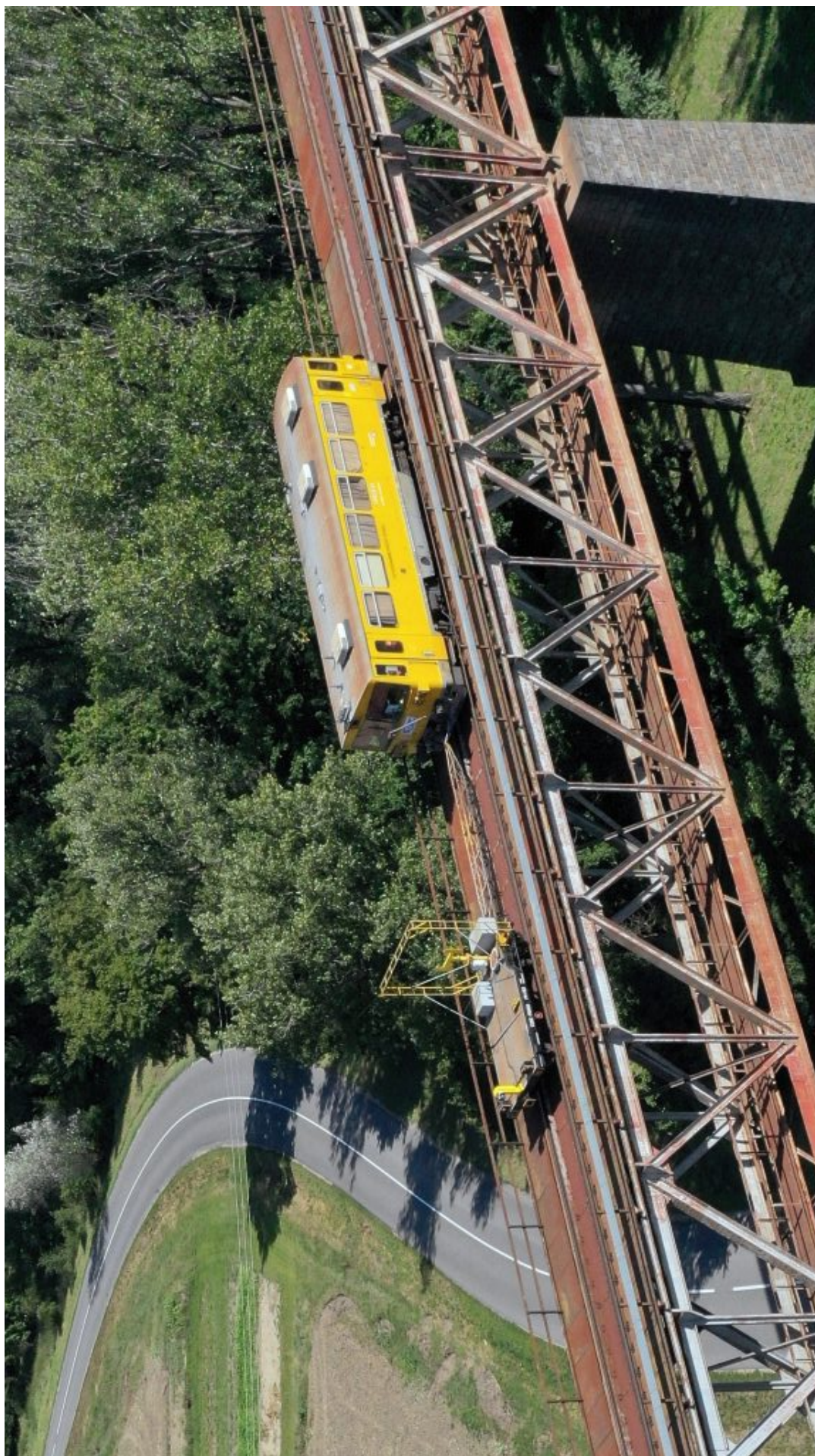


Obrázek 2.5: Výsledek měření MMS. Zdroj: <https://www.tudc.cz/wp-content/uploads/2017/05/mms6.png>

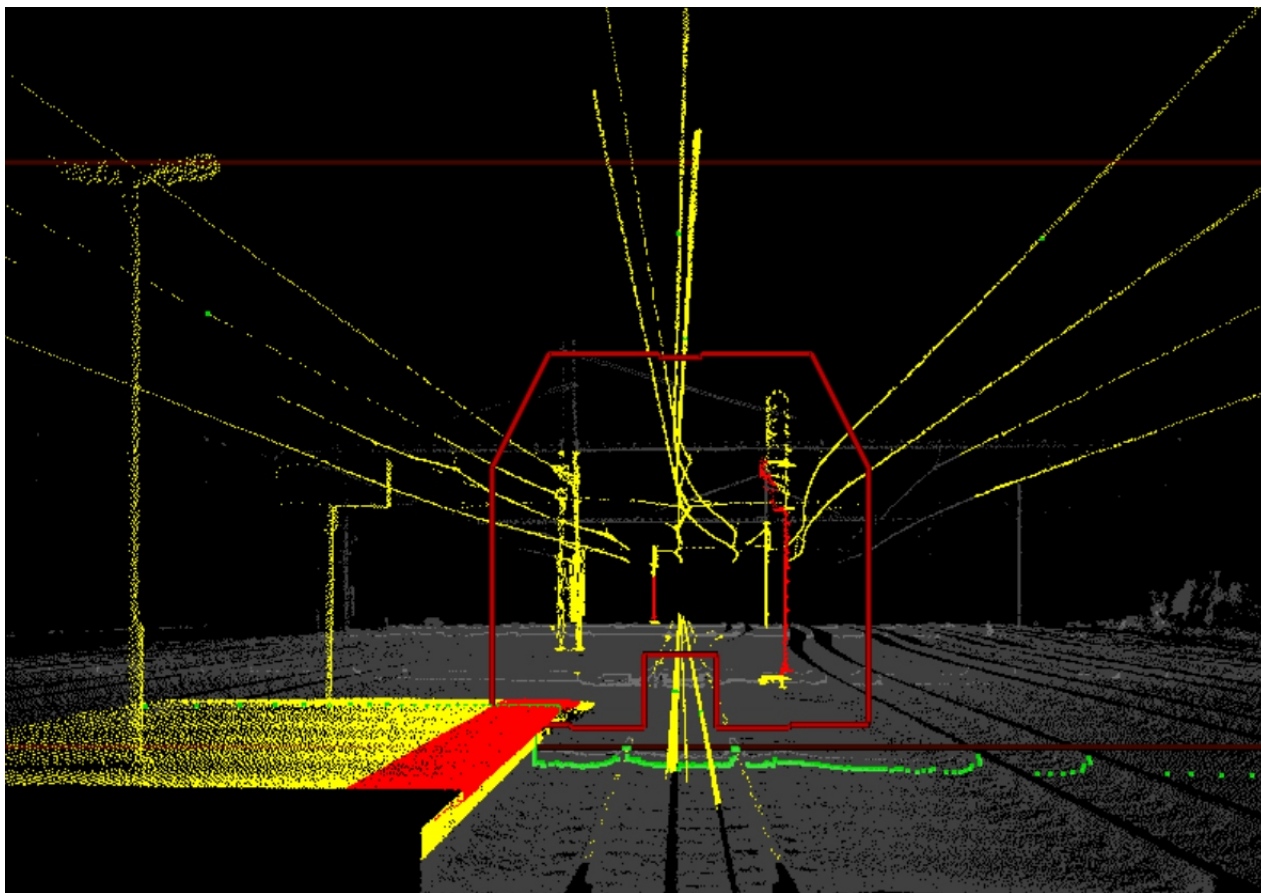
2.2.2 Fotogrametrický stroj

Fotogrametrický stroj, který můžeme vidět na obrázku č. 2.6 je upravené kolejové vozidlo řady 810, doplněné o vozík s rotačním bodovým laserem a kamerou pro pořizování snímků tratě ve směru jízdy.

Hlavní technologií je jednosnímková digitální fotogrametrie, přičemž laserové skenery se využívají k souvislému měření prostorové průchodnosti delších traťových úseků.



Obrázek 2.6: Fotogrametrický stroj. Zdroj:<https://www.tud.c.cz/index.php/diagnosticke-prostredky/fotogrammetricka-souprava/>



Obrázek 2.7: Výsledek měření Fotogrametrického stroje. Zdroj: <https://www.tudc.cz/index.php/diagnosticke-prostredky/fotogrammetricka-souprava/>

Fotogrametrický stroj sunoucí vozík s rotačním laserem ve směru jízdy se pohybuje tratí o rychlosti třiceti kilometrů v hodině, přičemž je nutné u každé překážky zastavit. Měření může probíhat pouze ve vlakových prostopích.

Ukázku výsledku zpracování bodových měření speciálním softwarem nalezneme na obrázku č. 2.7.

Kapitola 3

Návrh a implementace datové skladu

3.1 Úvod

Při navrhování jakéhokoliv podnikového datové skladu musíme vždy vycházet z praktických znalostí z dané oblasti. Tvoříme totiž nástroj, který budou používat pracovníci vyškolení ve svém oboru. To, co se nám může zprvu zdát nedůležité, nebo byt i nelogické se pak za správného kontextu z oboru ukáže jako podstatné pro jejich práci.

Samozřejmě je zapotřebí počítat i s různými změnami během vývoje, kterými mohou být například změny zdrojů, datových typů, pozměnění metodik apod. Vývoj každé aplikace provází změny, na které se přijde až po zahájení práce na projektu. Z těchto důvodů musí být náš návrh datového skladu dobře promyšlený a do určité míry i ohebný, aby zvládl případné změny.

V této kapitole si projdeme postupně návrh a implementaci rozšíření datového skladu o nový modul prostorové průchodnosti. Od postupu importu až po výsledné funkční uživatelské prostředí pro operaci s daty.

3.2 Import dat

Pro navržení importu dat je zapotřebí mít představu o objemu, rychlosti přibývání a o variaci dat. Jedná se v zásadě o srozumění s definicí 3V od Douga Laneye z teorie velkoobjemových dat. Pojdme se tedy uvést do kontextu dat, která v rámci prostorové průchodnosti budou proudit do datového skladu.

V celé české republice je přibližně devět tisíc kilometrů trati, kde pro každý úsek máme jednu či více průjezdných kolejí. Pro každou z těchto kolejí se provádí postupně měření fotogrametrických strojem, který jsme popsali v podkapitole 2.2.2.

Fotogrametrický stroj zaznamenává několik parametrů pro každý čtvrtmetr vybraného úseku. Ke každému čtvrtmetru se pak pojí několik tisíc bodových hodnot, naměřených rotačním laserem.

To znamená, že např. pro kolej č.1 úseku Benešov u Prahy - Čerčany, která je dlouhá přibližně osm a půl kilometru máme pak třicet čtyři tisíc čtvrtmetrů ke zpracování. A jedná se pouze o jedinou kolej jednoho úseku. Měření navíc probíhají průběžně několikrát do roku, pro zachování aktuálnosti dat.

Z tohoto příkladu vidíme, že se opravdu jedná o velkoobjemová data a je zapotřebí zajistit jejich pravidelné a postupné zpracování v rozumné míře. V závislosti na rychlosti přibývání dat se pak rozhodujeme, zda bude proces automatizovaný a pravidelně si bude kontrolovat existenci nových dat, nebo zda se bude spouštět povolnou osobou, která si je vědoma přírůstků dat.

Druhá možnost zprostředkována uživatelským rozhraním je možná, pokud data proudí způsobem jednoho velkého „balíku“, např. souhrná data za měsíc. V našem případě, kdy data proudí do zdrojového systému průběžně, není tento způsob adekvátní a přikláníme se k automatizovanému skriptu pravidelně se spouštějícím na serveru.

3.2.1 Automatizovaný skript

Pro spouštění pravidelných úkonů na unixových operačních systémech se používá takzvaný cron scheduler. Je to nástroj, kterým můžeme ve specifický čas opakovaně spouštět příkazy či skripty. My jej využíváme k automatizaci importů na serveru datového skladu.

Vzhledem k tomu, že se jedná o nový modul, musíme mít na paměti i import a zpracování všech historických dat týkající se dané oblasti, důležitých pro statistiky a porovnávání aktuálního stavu tratí s předešlými obdobími.

Z tohoto důvodu se spouští importní skript, který si popíšeme detailněji níže, dle nastavení v cron scheduleru každou hodinu. Nicméně import delších úseků může trvat i několik hodin a tak musíme zajistit, aby se skripty nepouštěly předtím, než je předchozí dokončen.

Pro tyto účely je vytvořena v SQL databázi na serveru datového skladu tabulka, ve které evidujeme stav v jakém se cron job (skript puštěný cron schedulerem) nachází. Tyto stavy jsou samovysvětlující: Ve zpracování, Zpracovaný a Chyba.

Tabulka pak ještě obsahuje počáteční a koncový čas běhu importu. Před spuštěním dalšího importu se vždy zkontroluje, jestli poslední záznam v tabulce je ve stavu „Zpracovaný“. Pokud ano, zjistí se zda jsou k importování další data a případně se vytvoří nový záznam se stavem „Ve zpracování“.

Průběh skriptu je zapisován do logu na server a v případě, že skončí ve stavu „Chyba“ je možné dohledat konkrétní příčinu spadnutí skriptu. V takovém případě se nad databází provede rollback, navrácení do předchozího stavu před spuštěním skriptu, čímž zabráníme uložení neúplných dat, v tomto případě neúplného měření úseku.

3.2.2 Importovaná data

Ještě před tím, než si popíšeme přesný postup automatizovaného skriptu, přiblížíme si více všechna data, která budeme importovat. Jak již bylo zmíněno, importují se jednotlivé čtvrtmetry měřeného úseku a jim příslušná bodová data.

Dále pak fotogrametrický stroj zachycuje místa s vysokým zásahem do prostorové průchodnosti kolejových vozidel. Tato data „překročení“ jsou samozřejmě patrná i z bodových měření, ale evidujeme je zvláště mimo jiné i pro potřeby zobrazení v grafu viz. 3.7.

V podkapitole 2.1.2 jsme si představili referenční profily kolejových vozidel, kterých je celá řada. I údaje o jejich tvarech je zapotřebí mít uložené v databázi právě pro porovnání s naměřenými bodovými daty.

Zde se pozastavíme nad metodikou zpracování referenčních profilů, která se změnila v průběhu tvorby tohoto modulu z důvodů rychlostních požadavků, které si dále přiblížíme v podkapitole 3.5.

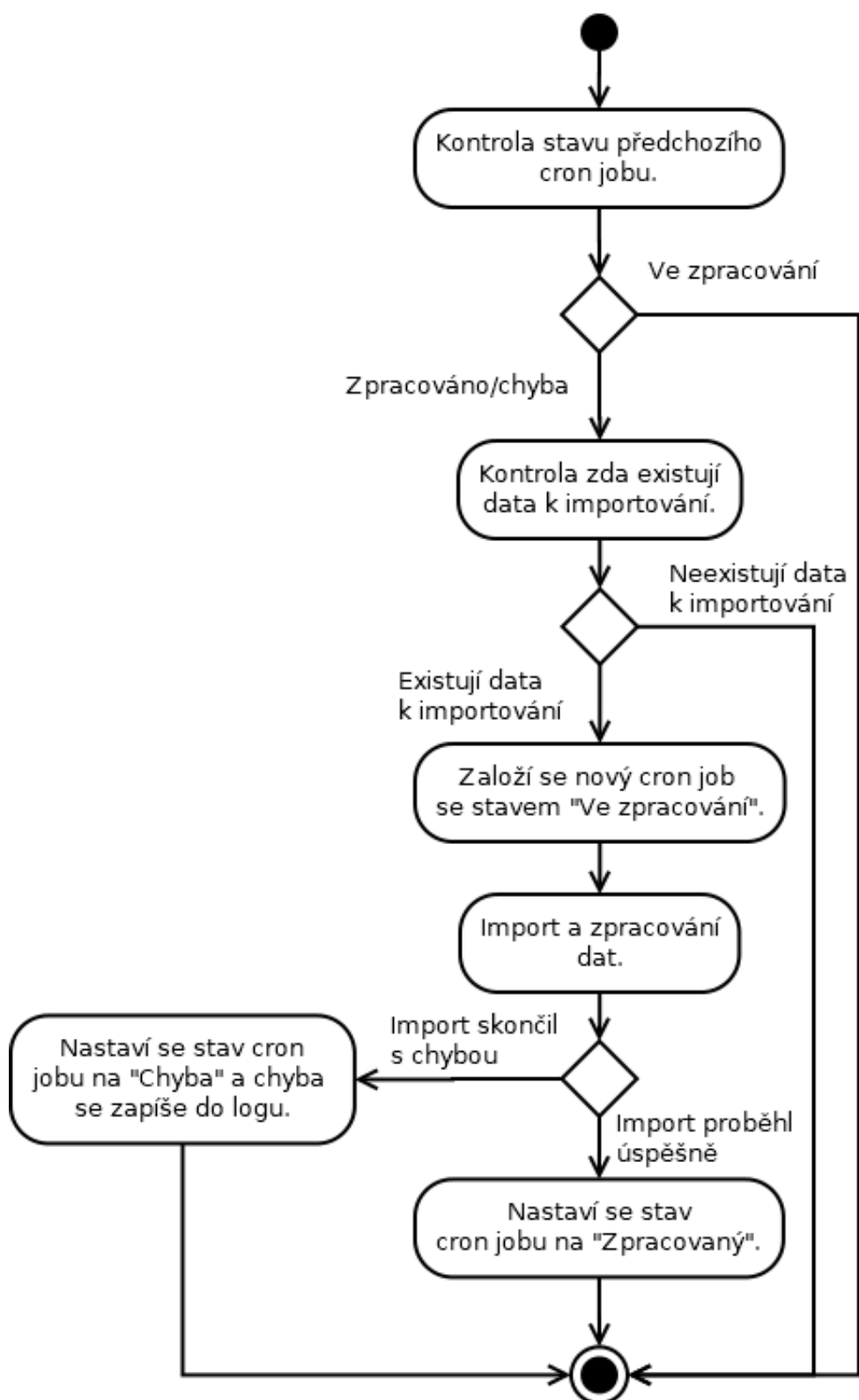
Původně se totiž na základě několika vrcholů vypočítával bodový obrys, který sestával přibližně z pěti tisíc dílčích bodů v závislosti na konkrétním referenčním profilu. Později se z důvodů změny metodiky přešlo na polygonální strukturu, tedy uchovává se jen informace o počátečním a koncovém bodu hrany v kartézských souřadnicích.

Některé referenční profily mohou být takzvanými „subprofily“ jiných referenčních profilů. Tuto informaci evidujeme také pro potřeby implementace vizualizace.

3.2.3 Postup automatizovaného skriptu

Obecný postup importního skriptu můžeme po předchozích podkapitolách shrnout do několika jednoduchých kroků.

- V první řadě se zkontroluje zda doběhl předešlý import, tj. zkontroluje se poslední záznam v tabulce evidující stavy cron jobu. Pokud stav není „Ve zpracování“ přejde k dalšímu bodu, jinak se pokusí spustit za další hodinu.
- V přehledové SQL tabulce balíčků dat k importu se nalezne záznam, který ještě nebyl zpracován a zapíše se nový záznam o stavu cron jobu, konkrétně „Ve zpracování“. Pokud není ke zpracování nový záznam, skript to zkusí znovu za hodinu jestli nepřišla nová data.
- Postupně pak importujeme všechna data zmíněná v 3.2.2. Přesný postup importu těchto dat nalezneme v následujících podkapitolách.
- V případě, že import neskončil chybou změním stav cron jobu na „Zpracovaný“ a v přehledové tabulce označíme příslušný balíček dat za naimportovaný. V případě chyby, změním stav na „Chyba“ a zapíšeme chybovou hlášku do logu pro programátora.



Obrázek 3.1: Diagram automatizovaného skriptu spouštějícího se každou hodinu.

3.2.4 Import úsekových měření

Jedná se o hlavní a časově nejnáročnější část celého importu. Data ze zdrojového SQLITE souboru obsahují na každém řádku společně obecné údaje (systémová id, údaje o poloze apod.) pro všechny čtyři čtvrtmetry včetně specifických informací týkající se konkrétního čtvrtmetru. Mezi tyto dílčí informace patří i laserová měření v podobě binárního řetězce.

Pro lepší přehlednost jsme vytvořili v SQL databázi dvě tabulky, první přehledovou tabulku prostorové průchodnosti, obsahující obecné informace a druhou tabulku s detailem laserového měření.

Každý řádek zdrojového souboru postupně převádíme na čtyři řádky, kde každý reprezentuje jeden čtvrtmetr a ukládáme je do přehledové tabulky. Některé údaje ze zdrojového souboru je zapotřebí formátovat, většinou se však jedná o uspořádání hodnot pod společné jednotky.

Po zapsání obecných údajů jednoho čtvrtmetru do přehledové tabulky pak zpracováváme příslušný záznam laserového měření. Máme tři tisíce polárních souřadnic na jednu otočku laseru. Vypočítáme úhel a převedeme polární souřadnice na souřadnice kartézské.

Pro každý čtvrtmetr pak máme uložených v tabulce laserových měření tři tisíce řádků kartézských souřadnic, které pak využíváme k pozdější vizualizaci.

3.2.5 Import dat referenčního profilu

Jak bylo zmíněno v 3.2.2 import dat referenčního profilu se dočkal za dobu vývoje modulu dvou verzí. Přestože se z technických důvodů využívá druhá iterace řešení, první přesto stojí za zmínku a tak si ji také přiblížíme.

Předtím, než si popíšeme jednotlivé metody zpracování, přiblížme si podobu zdrojových dat referenčních profilů. I zde rozdělujeme data na obecné informace, jakou jsou název, typ, systémové identifikační číslo atd. a na detailní informace každého profilu. Zde detailní informace jsou kartézské souřadnice vrcholů, reprezentující obrys referenčního profilu kolejového vozidla.

Pro data jsme opět navrhly dvě SQL tabulky, kde jedna obsahuje obecné informace a druhá hodnoty, dle kterých se pak provádí vizualizace. Hodnoty a struktura druhé tabulky se pak lišila v závislosti na verzi v jaké bylo řešení zpracování importu.

První iterace

Pro vizualizaci uceleného referenčního profilu nestačí samotné vrcholy a jelikož se u dat prostorové průchodnosti pracuje s body danými kartézskými souřadnicemi šlo se touto cestou i zde.

V praxi se tedy zamýšlelo reprezentovat celý referenční profil body, kterých mělo

být tolik, aby splynuly v jednotný obrys. V programovacím jazyce PHP se implementovala metoda, která vygenerovala body mezi jednotlivými vrcholy.

Vycházelo se přitom z obecné rovnice:

$$y = k \times x + q$$

Načteme si x a y pozici dvou po sobě jdoucích vrcholů referenčního profilu. Pro přehlednost označme první vrchol jako

$$A(x, y)$$

a druhý jako

$$B(x, y)$$

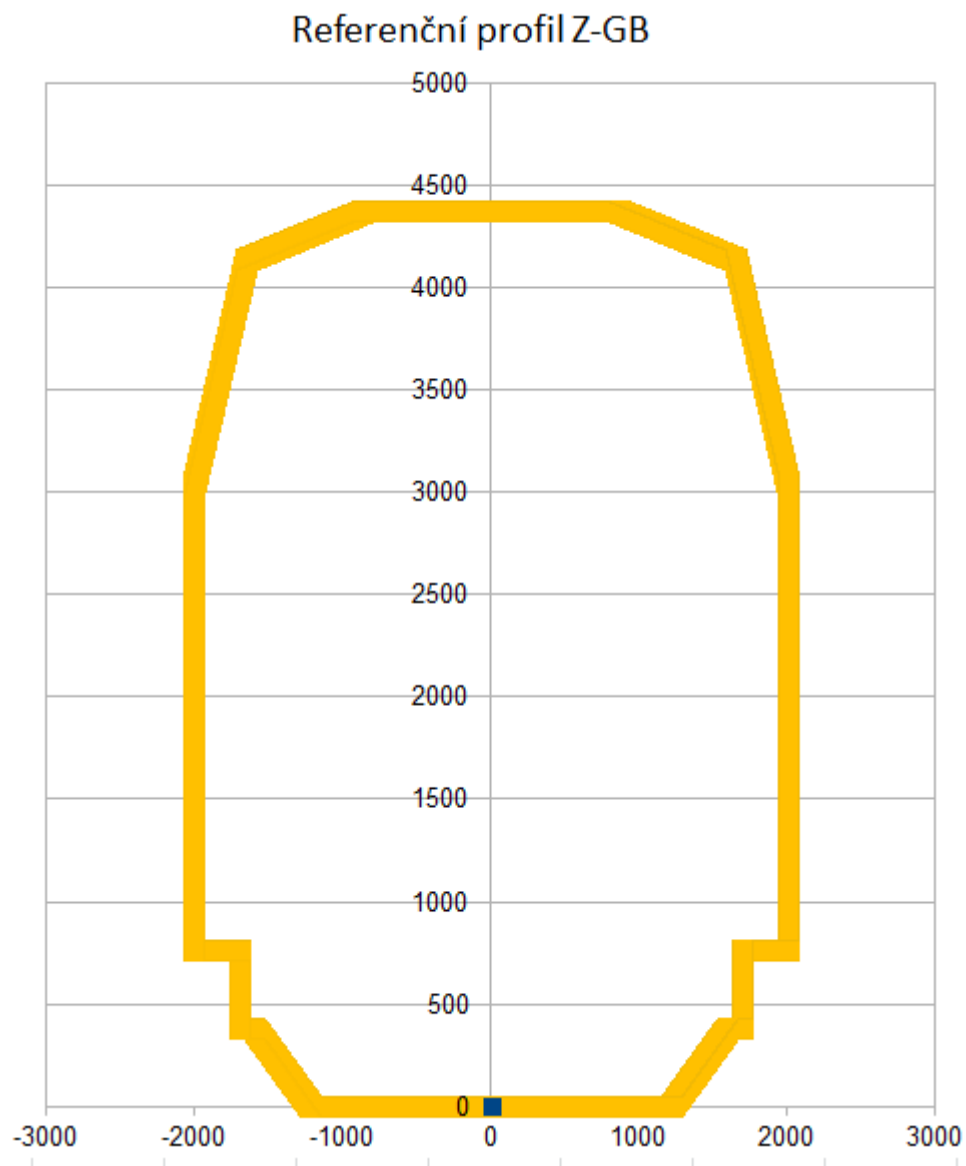
Vypočítáme x a y vzdálenosti odečtením souřadnic bodu A od bodu B. Pokud nejsou vzdálenosti rovné nule, využijeme je k vypočtu parametru k. Poté dopočítáme i parametr q.

$$k = \frac{y_vzdalenost}{x_vzdalenost}$$

$$q = A(y) - (k \times A(x))$$

Pomocí PHP funkce *range* získáme s hodnotou kroku rovného 1 pole x hodnot mezi body A a B. S nalezenými parametry k a q pak můžeme dopočítat veškerá y dosazováním v cyklu do rovnice. Dvojice souřadnic s vazbou na referenční profil pak reprezentovaly jeden řádek v SQL tabulce.

Na obrázku č. 3.2 pak můžeme vidět vizualizaci dopočítaných bodů referenčního profilu Z-GB v jednoduchém provedení v programu excel. Jednotky zobrazené na osách grafu jsou v centimetrech.



Obrázek 3.2: Ukázka testu dopočítaných bodů mezi vrcholy referenčního profilu jako bodového grafu v programu excel.

Druhá iterace

Novou myšlenkou, která nahradila dopočítávání bodů mezi jednotlivými vrcholy referenčních profilů, bylo uvažovat nad profilem jako nad polygonem. To znamenalo vytvořit si tabulku popisující strany polygonu.

V tomhle případě zůstává tabulka s obecnými informacemi v zásadě stejná a mění se tabulka, která uchovává detail profilu. Do ní se krom id odkazujícího na referenční profil ukládá informace o straně polygonu a její počáteční a koncové kartézské souřadnice.

Při importu referenčních profilů obě tabulky nejdříve vyprázdníme a pak zpracováváme data v cyklu, kde pak pro jednotlivé referenční profily procházíme jejich vrcholy a tvoříme z nich řádek po řádku strany polygonu.

Výslednou vizualizaci zpracovanou pro ukázkou v programu Excel můžeme vidět na obrázku č. 3.3..

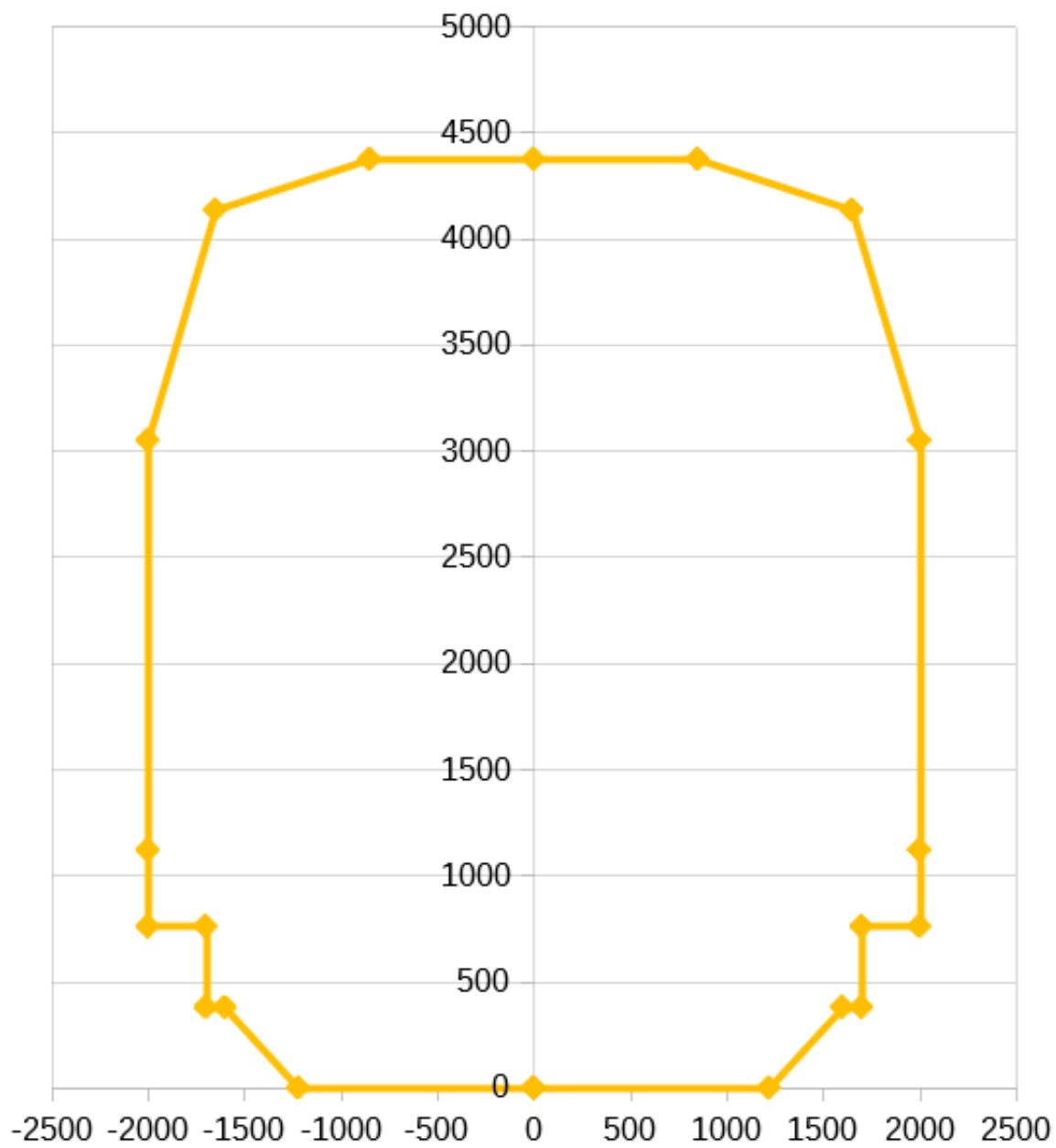
3.2.6 Import dat subprofilů

Informace o příslušícím subprofilu k referenčnímu profilu je jednou z informací zapisující se do přehledové tabulky. Vzhledem k tomu, že subprofil je sám o sobě referenční profil, není zapotřebí žádné samostatné tabulky.

3.2.7 Import dat překročení

Pro tato data vznikla SQL tabulka víceméně kopírující 1:1 zdrojovou tabulku. Ukládáme do ní informace o úseku, na kterém se překročení nachází s upřesněním na metry. Není zde potřeba provádět žádnou transformaci dat.

Referenční profil Z-GB



Obrázek 3.3: Ukázka testu vrcholů referenčního profilu Z-GB jako grafu v programu Excel.

3.3 Návrh uživatelského rozhraní

Pro plné využití potenciálu dat uložených v datovém skladu, je zapotřebí vytvořit adekvátní uživatelské rozhraní, ve kterém si uživatelé budou moci strukturovaná data prohlížet v podobě, která jim dále pomůže v práci. Rozhraní datového skladu je koncipováno jako webová aplikace.

3.3.1 Webová aplikace

Jak již bylo zmíněno jedná se o webovou aplikaci a ta sebou pojí i programovací jazyky, které se při její tvorbě využívají. Pojdme si před samotnou implementací projít jazyky a obecný princip, který jsme při implementaci použili.

Požadavky od uživatele, ať už se jedná o stisknutí tlačítka, nebo nějaké předem definované kombinace tlačítek, jsou zachycovány pomocí programovacího jazyka Javascript, respektive rozšiřující javascriptové knihovny jQuery.

Některé uživatelské akce vedou na načtení dat v různých podobách a tak je zapotřebí komunikovat s databází pro jejich získání. Někdy je data zapotřebí ještě i nějakým způsobem předzpracovat a přestože i tato akce je možná v jQuery, není to příliš vhodné.

Důvodem je, že jQuery běží na straně klienta a výpočetní výkon není vždy konzistentní, nebo dostatečný. Proto se načítání z databáze a předzpracování dat provádí pomocí programovacího jazyka PHP na straně serveru.

Pro komunikaci s PHP používáme takzvaných AJAXů (Asynchronous JavaScript and XML). Jejich výsledkem jsou předzpracovaná data z PHP, kterými pak můžeme pomocí jQuery dále aktualizovat prvky na stránce.

Ve shrnutí je princip následující:

- Nejprve zachytíme uživatelský požadavek pomocí jQuery.
- Pokud požadavek vede na načtení dat z databáze, vyšleme pro komunikaci se serverem AJAX.
- V programovacím jazyce PHP načteme a připravíme data do příslušící podoby a odešleme zpět jako výsledek AJAXu.
- Pomocí jQuery s daty aktualizujeme příslušný prvek (tabulka, graf, obrázek atd.).

3.3.2 Vyhledávání naměřených úseků

V konkrétním případě modulu prostorové průchodnosti vycházíme částečně z vlastních zkušeností ostatních modulů, které mají celoplošně společnou jednu funkcionalitu (v principu, ne nutně v implementaci).

Jedná se o funkcionalitu vyhledávání traťových úseků, které si chce uživatel prohlédnout či s nimi pracovat. Způsobů vyhledávání je několik a aplikují se v závislosti na potřebách vybraného modulu.

- Prvním ze způsobů je vyhledávání ve stromové struktuře, kde rozdělujeme větší úseky na menší dílky a k nim pak za případných dalších filtrů, jako například rok a číslo měření k nim dohledáváme záznamy. Hierarchie se dělí na tyto články:

- Supertrasa
- Složka supertrasy
- Historie

Ukázku rozhraní pak můžeme vidět na obrázku č. 3.4.

- Traťové úseky dále můžeme rozdělovat z pohledu správy, pod kterou patří. Opět se jedná o hierarchickou strukturu. Jejím průchodem opět profilujeme vybrané záznamy. Ukázku můžeme vidět na obrázku č. 3.5.

- Oblastní ředitelství
- Správa trati
- Traťový okrsek

- Dalším způsobem je vyhledávání pomocí GPS souřadnic a zadáním oblasti v metrech. Ukázka na obrázku č. 3.6.

- Ve poslední řadě se někdy využívá i vyhledávání přes TUDU (traťový úsek definiční úsek), můžeme vidět na obrázku č. 3.7.

Supertasy

Zobraz záznamů ▾

Označení ▾ **Supetrasa** ▾ **kol.č.** ▾ **ST** ▾ **Název** ▾

Žádné záznamy nebyly nalezeny

Zobrazuji 0 až 0 z 0 záznamů Předchozí Další

Složky supetrasy

Zobraz záznamů ▾

Repre_tudu ▾ **Od [km]** ▾ **Do [km]** ▾ **Jméno** ▾ **Délka** ▾

Žádné záznamy nebyly nalezeny

Zobrazuji 0 až 0 z 0 záznamů Předchozí Další

Historie

Zobraz záznamů ▾

Rok ▾ **Číslo měření** ▾ **Datum měření** ▾ **Týden** ▾ **Prostředek** ▾ **Postavení** ▾ **OD** ▾ **DO** ▾ **DM_délka** ▾ **Stavební délka** ▾

Žádné záznamy nebyly nalezeny

Obrázek 3.4: Ukázka vyhledávacího rozhraní přes supetrasy.

Oblastní ředitelství (OŘ)	Vyberte možnost ▾
Správa tratí (ST)	Vyberte možnost ▾
Traťový okrsek (TO)	Vyberte možnost ▾

Obrázek 3.5: Ukázka vyhledávacího rozhraní přes organizační strukturu.

Vyhledávání přes název supertrasy	Vyhledávání přes GPS souřadnice
GPS Lat a Lon: <input type="text"/>	vstupní formát např.: 50,083 14,436
Vzdálenost: <input type="text"/>	m

Obrázek 3.6: Ukázka vyhledávacího rozhraní pomocí GPS souřadnic.

The image shows a search interface with a blue header bar. On the left, there is a search bar labeled 'TUDU' with a text input field. To its right is a dropdown menu labeled 'Stanice'. Below the search bar, the text 'Úsek' is visible. To the right of the search bar, there is a unit label 'km' and a text input field. Below the dropdown menu, there is a label 'Měření' and a text input field. The entire interface is set against a light blue background.

Obrázek 3.7: Ukázka vyhledávacího rozhraní pomocí parametru TUDU.

Pro prostorovou průchodnost byla vybrána varianta vyhledávání pomocí supertrasy, jelikož byla v tomhle případě nejlogičtější z povahy práce s daty. Zde jsme se dostali do fáze, kdy se spojovala existující data datového skladu spolu s daty prostorové průchodnosti.

Uživatel si vyhledá seznam supertas, kde po zvolení jedné z nich pak následně ještě vybere její složku. S vazbou na identifikační číslo složky supertasy vyhledáváme v tabulkách úseků právě ty úseky, které mají stejné identifikační číslo měření jako ty importované v datech prostorové průchodnosti.

Důvodem toho je, že tabulka obsahující naměřené čtvrtmetry neobsahuje obecné informace měření, ale jen identifikační číslo měření, ke kterému patří. Obecné informace o měřeních jsou z uchovávané v samostatné tabulce. Zde můžeme vidět důraz na synergii s dalšími importy dat a vazbu modulu na celý datový sklad.

Po vybrání měření se pak přesouváme k vizualizaci dat prostorové průchodnosti.

3.3.3 Vizualizace dat

Vizualizace dat je vždy specifická k povaze dat, která se snažíme zobrazit, zároveň v případě, kdy se jedná o firemní projekt přichází určité požadavky od klienta. V tomto modulu se snažíme zobrazit data naměřená fotogrametrickým strojem.

Do rozhraní tak umísťujeme nástroj prohlížení fotografií úseků, canvas pro zobrazování bodových měření a ve poslední řadě graf celého úseku s užitečnými informacemi pro uživatele. Každý nástroj sebou přináší vlastní ovládací prvky přidávané do rozhraní.

Ke canvasu, na kterém se promítá většina naimportovaných dat, je přiřazeno několik ovládacích a informačních prvků:

- Tlačítka pro posouvání úseku po čtvrtmetrech vpřed a vzad.
- Tlačítka pro ovládání zoomu - přiblížení, oddálení a zrušení.
- Nástroj pro měření vzdáleností mezi dvěma body a canvasu.
- Export do několika formátů.
- Tabulka udávající informace, např. počty překročení vůči vybranému referenčnímu profilu.

Na canvas, ale nevykreslujeme jenom naměřené okolí, nýbrž také vybraný referenční profil. Do rozhraní tak přibyla volba zobrazovaného referenčního profilu. Významným nástrojem pro tento modul jsou operace, které mohou s referenčním profilem uživatelé provádět.

- Posouvat referenční profil po osách x a y.
- Simulace převýšení kolejí v mm. (jedna kolej je umístěná výše např. v zatáčce).

- Zvyšování, či snižování pojistné zóny.

Nástroj čelních fotek obsahuje také jednoduchý posun vpřed a vzad po fotkách nafocených přibližně po dvaceti metrech. Dále k nim přibyl nástroj „simulace jízdy“, který postupně projíždí fotky ve směru jízdy, čímž ji simuluje.

Tento nástroj je navíc synchronizovaný s posouváním čtvrtmetrů na canvasu. Vzhledem k tomu, že jsou fotky foceny s větší vzdáleností, přepnutí čelní fotky vždy čeká až ji data na canvasu „doběhnou“.

Posledním prvkem v rozhraní je graf celého úseku, zobrazující navíc i hodnoty naměřené jinými měřícími vozy, které pomáhají ilustrovat kontext pro vybraná bodová měření zobrazená na canvasu. Mezi hodnoty na grafu patří:

- Rozchod kolejí - horizontální vzdálenosti mezi dvěma kolejnicemi, údaj je uvedený v mm.
- Křivost kolejí - vychýlení ze správného úhlu kolejnic pro dané místo.
- Převýšení kolejí - vertikální vzdálenost mezi dvěma kolejnicemi, údaj uvedený v mm.
- Převýšení v oblouku.
- Poloměr v oblouku, údaj uvedený v m.
- Počet překročení na metr.
- Oblast, ve které se nachází překročení.

3.4 Implementace zobrazení čelních fotek

Jedním ze zdrojů dokumentujícím vizuální stav úseku jsou čelní fotografie, které při své jízdě fotogrametrický stroj pořizuje. Fotografie pak pomáhají k lepšímu přehledu o konkrétním místě, které si prohlížíme prostřednictvím bodových měření.

Čelní fota mají svůj samostatný import, jelikož se nevyužívají jen v tomto modulu. Fyzicky jsou fotky uloženy na vzdáleném serveru a v databázi datového skladu uchováváme jejich umístění a metadata.

Po vybrání úseku uživatelem pracujeme při načítání s identifikačním číslem úseku a jeho počátkem a koncem. Při první inicializaci identifikujeme fotku spadající pod počátek úseku, konkrétně k ní načteme cestu a vložíme ji do příslušného elementu v rozhraní.

Pro zrychlení operací prohlížení fotek načítáme zároveň s fotkou aktuálního metru i fotku následující a předchozí. Na základě jejich existence pak i upravujeme ovládací tlačítka pro jejich posun.

Princip zde je přitom jasný, pokud předchozí, nebo následující fotka neexistuje, tlačítko se zneaktivní. Z některých technických důvodů tak nemusí být jen na okrajích rozmezí úseku, přestože se to stává jen zřídka.

3.5 Implementace vykreslení bodových měření a referenčního profilu na canvas

Předtím, než se posuneme dále, věnujme chvíli prvku „canvas“ a jak se s ním pracuje. Canvas můžeme jednoduše definovat jako vymezený prostor na webové stránce, na který je možné vykreslovat jednoduchou grafiku.

Prvek vkládáme do html webové stránky a následně na něj pomocí jQuery kreslíme požadované tvary. Námi využívanými jsou především úsečky a obdelníky. Úsečkám předáváme v kartézských souřadnicích počáteční a koncový bod.

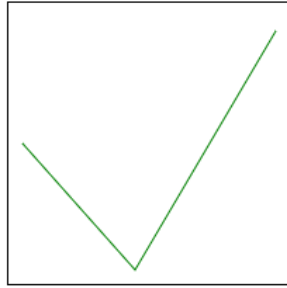
Obdelníky jsou náhrada za vykreslení bodu. Předáváme jim souřadnici levého horního rohu v kartézských souřadnicích a jejich výšku a šířku. Na ukázkách 3.1 a 3.8 můžeme vidět jednoduché použití canvasu.

```
// vytvoreni prvku canvas
<canvas id="canvas" width="200" height="200"
  style="border:1px_solid_#000000;">
</canvas>
<script>
// nacteni elementu
var c = $("#canvas");
var ctx = c.getContext("2d");
// nastaveni barvy
ctx.strokeStyle = "green";
// prvni usecka
ctx.moveTo(10, 100);
ctx.lineTo(90, 190);
// druha usecka
ctx.moveTo(90, 190);
ctx.lineTo(190, 20);
// vykresleni
ctx.stroke();
</script>
```

Ukázka kódu 3.1: Ukázka práce s canvasem

Do implementace týkající zobrazení dat na canvas zahrnujeme i vysvětlení implementace ovládacích prvků s ním spojených. Přiblížíme si zde blíže i důvody změny principů importování dat referenčních profilů.

Cílem této části modulu je umožnit uživateli porovnávat bodová měření vůči vybraným referenčním profilům a zkoumání počtů bodů překročení. Můžeme tak například



Obrázek 3.8: Výsledek ukázky práce s canvasem

zkoumat zda daný typ kolejového vozidla projede bez problému zúženým mostem, nebo tunelem.

Body vyskytující se uvnitř prostoru referenčního profilu pak obarvujeme červenou barvou pro jasnou distinkci oproti zeleným bodům, které dohromady vizuálně tvoří naměřené okolí.

Zprostředkováváme pro uživatele i nastavitelnou pojistnou zónu, při implementaci na základě pokynů nazvanou "Zóna nejistoty". Body zachycené v zóně nejistoty jsou pak pro rozlišení obarveny oranžově. Dále je pak možné manipulovat s umístěním referenčního profilu po osách x a y a také upravovat jeho úhel při převýšení.

Manipulace je určena vyjma testovacích účelů také ke korekci umístění referenčního profilu, který je defaultně umístěn do středu naměřeného úseku. Přestože je středem bodových měření zpravidla kolej, po které měřící vůz projíždí, je někdy potřeba nasimulovat si jízdu zatáčkou.

Při jízdě zatáčkou dochází k převýšení kolejnic a je nutné zkorigovat úhel a pozici referenčního profilu pro adekvátní výsledky. Údaj převýšení je udáván v milimetrech, které pak převádíme na úhel.

Převýšení je zároveň jednou z hodnot zobrazovaných na grafu pro každý metr viz. 3.7 a uživatel ho tak nemusí dohledávat v jiném modulu či jiném zdroji shrnujícím informace o daném úseku.

3.5.1 Načítání a vykreslení dat

Zmínili jsme si jakým způsobem obecně pracujeme při načítání dat ve webové aplikaci a také jsme si uvedli jednoduchý příklad jak se data vykreslují na canvas. Požadavek na načtení nových dat tedy odchyťáváme pomocí jQuery a akcí, které tento požadavek vyvolají je několik.

Jmenovitě:

- Posun na další metr úseku pomocí tlačítek.
- Změna nastavení referenčního profilu.
- Export canvasu. (viz. 3.5.5).

- Spuštění simulace jízdy. (viz. 3.6)
- Kliknutí do grafu. (viz. 3.7)

První dva body si přiblížíme v rámci této podkapitoly. Jak bylo zmíněno v podkapitole o importu dat 3.2.4, bodová měření jsou rozdělena na čtvrtmetry. Neposouváme se pomocí ovládacích tlačítek o celý metr, nýbrž o čtvrtmetr. Pro zrychlení přepínání načítáme pro každý metr najednou data všech čtyř příslušných čtvrtmetrů, díky čemuž je možné mezi nimi přepínat bez vyvolání dalšího načítání.

Součástí AJAXem získaných dat je kromě čtvrtmetrů i vybraný referenční profil s ohledem na nastavené parametry (obrázek č. 3.9). Akce přepočítat vyvolá nové vyslání AJAXu a následné překreslení canvasu aktuálními daty.

Referenční profil: Z_GC ▼ Přepočítat

Zobrazit subprofily:

Horizontální překlopení:

Zóna nejistoty: 25 mm

Posun x: 0 mm

Posun y: 0 mm

Převýšení: 0 mm (0°)

Obrázek 3.9: Rozhraní s operacemi referenčního procesu

Pro odůvodnění proč není v případě změny nastavení referenčního profilu vyměněn právě jen referenční profil si je potřeba přiblížit strukturu získávaných dat. Můžeme na ně pohlížet jako na vrstvy, které pomalu pokládáme přes sebe.

První vrstvou, tvořící podklad jsou naměřená bodová data čtvrtmetru, která jsou označena zelenou barvou. Další vrstvou jsou pak body (či vrcholy v druhé verzi zpracování) reprezentované referenční profily označené červeně a nakonec pak zóna nejistoty kopírující obrys referenčního profilu, označená oranžově.

Pokud bychom tedy měli první vrstvu staticky načtenou pro dané čtvrtmetry, museli bychom v případě změny, či posunu referenčního profilu a zóny nejistoty dopočítávat v javascriptu jaké body se mají obarvit a nakonec i první vrstvu překreslit.

Kromě toho by se jednalo o poměrně náročný výpočetní úkon na to, abychom ho prováděli na straně klienta. Z toho důvodu se přistoupilo na jiné řešení a to vytvořit vrstev více, respektive rozdělit první bodovou vrstvu dat na tři.

Princip si uvedeme na jednom čtvrtmetru, nicméně je aplikován na všechny čtyři naprosto stejně. Nejprve načteme do PHP z databáze všechna příslušná bodová měření z databáze a uložíme si je do asociativního pole viz. 3.2.

```
$array_name [ quarter ] [] = array (
    'x' => $x ,
    'y' => $y ,
    'barva' = "green "
);
```

Ukázka kódu 3.2: Ukázka struktury uchování dat čtvrtmetru

Body této vrstvy jsou defaultně nastaveny na zelené, reprezentující okolí. Postupně je pak porovnáme s vybraným referenčním profilem a jeho zónou nejistoty. Způsob vyhodnocení prošel vývojem a více detailněji se mu budeme věnovat v 3.5.2. a 3.5.3.

Při vyhodnocení se zjistí zda bod koliduje s referenčním profilem a pokud ano, je změněna hodnota parametru „barva“ na červenou. Stejně tak se vyhodnotí zda bod spadá do zóny nejistoty a v případě toho je změněna hodnota parametru „barva“ na oranžovou.

Díky tomu máme připravené už správně obarvené body a stačí je jen vykreslit jakožto první vrstvu. Přes ní pak ještě vykreslíme referenční profil a zónu nejistoty. Proto pokaždé, když se změní nastavení referenčního profilu, přenačítají se všechna data, pro získání adekvátního obarvení bodového měření.

Dále je pak vhodné zmínit, že operace měnící pozici referenčního profilu, oproti jeho výchozí, pak hrají při přípravě dat také velkou roli. Je nutné totiž adekvátně měnit hodnoty, kterými je referenční profil reprezentován.

Jedná se o delší blok a tak si k implementaci těchto operací řekneme více v 3.5.4. Zároveň přišly implementačně na řadu až po změně zpracování referenčních profilů a tak si nejdříve přiblížíme způsoby vyhodnocení bodů.

3.5.2 Vyhodnocení bodů překročení - první verze

Jak bylo vysvětleno v podkapitole o zpracování dat referenčního profilu 3.2.5 v raných fázích projektu byl referenční profil reprezentován body. Vzhledem k velkému počtu vygenerovaných bodů nebylo vizuálně poznat, že se jedná o bodovou reprezentaci.

Ve fázi vyhodnocení, zdali se bod nachází uvnitř, nebo vně referenčního profilu jsme pak porovnávali navzájem pozice bodů, konkrétněji všechny pozice bodů měření se porovnávali se všemi pozicemi bodů referenčního profilu.

V cyklu jsme pak procházeli jestli nějaký bod není na stejné x, nebo y pozici bodu referenčního profilu. Pokud ano, vyhodnocujeme, kde se nachází vůči druhé ose a to pomocí rozdělení naměřené oblasti na čtyři kvadranty, kde jejím středem (0,0) byl referenční profil (viz. obrázek č.3.2).

Postup vyhodnocení pomocí kvadrantu, ve kterém se bod nachází můžeme vidět na ukázce č. 3.3. Vyhodnocující metoda je společná pro test obou os a zároveň zahrnuje i případ, kdy se bod měření nachází přesně na bodu referenčního profilu.

Od tohoto způsobu se nakonec upustilo z důvodů, které jsou zmíněny v následující podkapitole 3.5.3 při představení druhé verze vyhodnocování.

```
$quadrant = self::detect_quadrant($inspected_x, $inspected_y);  
// we are comparing Y  
if ($reference_y != $inspected_y  
    && $reference_x == $inspected_x){  
    if ($quadrant == 1 || $quadrant == 2)  
        return $reference_y > $inspected_y;  
    if ($quadrant == 3 || $quadrant == 4)  
        return $reference_y < $inspected_y;  
    return null; // X=0 and Y=0  
}  
// we are comparing X  
else if ($reference_x != $inspected_x  
        && $reference_y == $inspected_y){  
    if ($quadrant == 1 || $quadrant == 4)  
        return $reference_x > $inspected_x;  
    if ($quadrant == 2 || $quadrant == 3)  
        return $reference_x < $inspected_x;  
    return null; // X=0 and Y=0  
}  
// the points are the same  
else if ($reference_x == $inspected_x  
&& $reference_y == $inspected_y) {  
    return true;  
}  
}
```

Ukázka kódu 3.3: Nastavení datové série

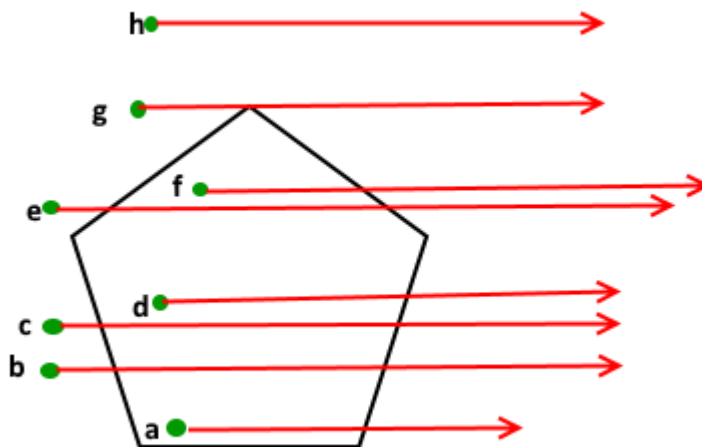
3.5.3 Vyhodnocení bodů překročení - druhá verze

Potřeba přejít na jinou verzi vyhodnocování bodů kolidující s referenčním profilem či zóně nejistoty vznikla ze dvou důvodů. Prvním z nich bylo velmi pomalé načítání dat pro celý metr. Způsobeno to bylo porovnáváním několika tisíc bodů dat prostorové průchodnosti spolu s několika tisíci body tvořícími referenční profil.

Druhý důvod byl odstranění chyb v identifikaci v případě naklopení referenčního profilu do strany, dle zadaného úhlu převýšení. Zavedla se proto na místo toho metoda „Vrhání paprsku“, neboli anglicky „Ray casting“.

Metoda vrhání paprsku je matematická metoda sloužící k určení toho zda se bod nachází uvnitř polygonu, anebo vně. Princip spočívá v tom, že od každého bodu natáhneme přímkou a sledujeme kolik stěn polygonu protne. Ukázka na obrázku č. 3.10.

Pokud přímkou protne lichý počet stěn polygonu, bod se nachází uvnitř. Pokud sudý



Obrázek 3.10: Zdroj: <https://www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon/>

tak se nachází mimo něj. Pro využití této metody jsme změnilí zpracování referenčních profilů tak, abychom je dostali definované jako strany polygonu viz. 3.2.5.

Přibližme si nyní jakým způsobem zjišťujeme, zda přímka protíná stranu polygonu. Tento postup opakujeme pro každou stěnu polygonu v cyklu a z výsledného součtu jsme schopni usoudit jeho umístění vůči referenčnímu profilu.

Vstupními parametry metody „test polygon side“ jsou v kartézských souřadnicích počáteční a konečný bod strany referenčního profilu, kartézské souřadnice testovaného bodu a ve poslední řadě kartézské souřadnice simulující nekočho z jedné či druhé strany. V použití se jedná o velkou zápornou či kladnou hodnotu pro x souřadnici, y zůstává stejné.

```
function test_polygon_side($v1x1, $v1y1, $v1x2, $v1y2, $v2x1,
    $v2y1, $v2x2, $v2y2)
```

Definice pomocných proměnných:

```
$d1; $d2;
$a1; $a2; $b1; $b2; $c1; $c2;
```

Převeďme vektor jedna (strana polygonu) na přímku a označme ji jako přímku jedna. Přitom chceme, aby odpovídala lineární rovnici ve tvaru $A \times x + B \times y + C = 0$.

```
$a1 = $v1y2 - $v1y1;
$b1 = $v1x1 - $v1x2;
$c1 = ($v1x2 * $v1y1) - ($v1x1 * $v1y2);
```

Každý bod, který rovnici řeší se nachází na přímce a obráceně. Výsledek rovnice bude kladné číslo pokud je na první straně přímky jedna a negativní pokud je na

druhé straně přímky. Dosadíme body vektoru dva (úsečka tvořená souřadnicemi testovaného bodu).

```
$d1 = ($a1 * $v2x1) + ($b1 * $v2y1) + $c1 ;  
$d2 = ($a1 * $v2x2) + ($b1 * $v2y2) + $c1 ;
```

Pokud mají d1 a d2 stejné znaménko, nacházejí se na stejné straně přímky a protnutí není možné. Metoda zde tak končí s výsledkem neprotnutí testované strany polygonu. Pokud však podmínky nejsou splněny pokračuje se v metodě dál.

```
if ($d1 > 0 && $d2 > 0) return 0 ;  
if ($d1 < 0 && $d2 < 0) return 0 ;
```

Fakt, že vektor dva protnul přímku jedna neznamená, že také protíná vektor jedna. Jinými slovy, vektor jedna (strana polygonu) je jen subset přímky jedna a nemůžeme vědět s jistotou, že ji vektor dva mohl protnout mimo interval vektoru jedna.

Provedeme stejný test, ale tentokrát pro vektor dva, který převedeme na přímku dva.

```
$a2 = $v2y2 - $v2y1 ;  
$b2 = $v2x1 - $v2x2 ;  
$c2 = ($v2x2 * $v2y1) - ($v2x1 * $v2y2) ;
```

Znovu vypočítáme rovnice d1 a d2. Stejný princip jako předtím, pokud mají stejné znaménko nedošlo k protnutí a metoda zde končí. Pokud podmínky nejsou splněny, pokračujeme dál.

```
if ($d1 > 0 && $d2 > 0) return 0 ;  
if ($d1 < 0 && $d2 < 0) return 0 ;
```

Zbývají už jen dvě možnosti, buď se vektory protínají přesně v jednom bodě, anebo jsou kolineární, což znamená, že se mohou protínat až v nekonečně mnoho bodech.

```
if (( $a1 * $b2) - ($a2 * $b1) == 0.0) return 1 ;
```

Pokud nejsou kolineární protínají se přesně v jednom bodě, další možnost není. Výsledek metody je pak potvrzení protnutí strany polygonu.

Toto vyhodnocování bodů probíhá jak pro referenční profil, tak i pro jeho zónu nejistoty, která je z hlediska programování v podstatě jeho zvětšená kopie. Každý naměřený bod je tedy testován až dvakrát.

O referenčním profilu můžeme říct, že je podmnožinou své zóny nejistoty. Proto nejdřív testujeme, zda se bod nachází uvnitř zóny nejistoty. Pokud se v ní vůbec nenachází, nemá smysl provádět test vůči referenčnímu profilu.

A naopak, pokud se nachází v zóně nejistoty, otestujeme jestli se bod nenachází ještě i uvnitř referenčního profilu. Bod pak příslušně obarvíme a přesouváme se na test dalšího.

3.5.4 Operace s referenčním profilem

Zde se budeme odkazovat na akce z rozhraní na obrázku č. 3.9, jejichž vyřízení předchází v kódu vypočítávání pozice bodů vůči referenčnímu profilu a jeho zóně nejistoty.

Důvodem je samozřejmě to, že pokud potřebujeme posunout či naklopit referenční profil vůči počátku, je zapotřebí přepočítat pozice jeho vrcholů, čímž změníme porovnáváný polygon.

První operaci na rozhraní je výběr referenčního profilu, která jednoduše na základě identifikačního načte z databáze příslušný profil. Zaškrťovací pole zobrazení subprofilů je parametr, díky kterému víme, zda vykreslovat subprofile příslušného referenčního profilu.

Tyto subprofile, které v databázi představují jiné samostatné referenční profile, pak označujeme jinými barvami pro rozlišení. Zde je důležité podotknout, že pro subprofile, se umístění bodů nevypočítává.

Další zaškrťovací pole, jako jediné z tohoto rozhraní neputuje jako parametr do PHP, ale zprostředkované pomocí javascriptu. Jelikož se jedná o horizontální překlopení, je jednoduší vykreslený canvas zrcadlově otočit.

Přepínač pro zobrazení zóny nejistoty navíc umožňuje nastavit její velikost v milimetrech, libovolně dle uživatele. Navýšení dosahujeme přičetním milimetrové hodnoty k vrcholům.

Další tři akce, posun x , y a převýšení se aplikují jak na referenční profil tak i na zónu nejistoty. Referenční profil je defaultně umístován na počátek měření (pozice laserového měřiče), bod $(0,0)$, čímž je jedna jeho půlka v záporném kvadrantu osy x a druhá v kladném.

Posuny po ose x a ose y jsou pak jednoduše provedeny připočtením hodnoty posunu k původní kartézské souřadnici všech vrcholů stran referenčního profilu. Žádné další operace nejsou nutné.

Při změně převýšení, neboli úhlu, potřebujeme před posunem vypočítat nové pozice bodů vrcholu. Nejdřív převedeme stupně na radiány a pak pomocí matematických vzorců vypočítáme nové pozice x a y .

```
$angle = deg2rad($move_angle);  
$x_new = ($x*cos($angle) - $y*sin($angle));  
$y_new = ($x*sin($angle) + $y*cos($angle));
```

3.5.5 Exporty canvasu

Další možností, kterou uživateli nabízíme jsou exporty canvasu do několika různých formátů. Export je jedním ze základních nástrojů pro uživatele, které jim umožňují promítnutí výsledků vizualizace měření do reportů či prezentací.

Exporty do PNG, SVG a PDF mají společný jeden aspekt a tou je příprava dat pro export. Vzhledem k tomu, že potřebujeme jinak formátovat data pro vykreslení na canvas a jinak pro export, necháváme data znovu vygenerovat se započítáním škálovací konstanty.

Takto upravenými daty pak nepřekresluje canvas, nýbrž je posíláme k dalšímu zpracování v závislosti na datovém typu zvoleného uživatelem. Níže si přiblížíme jejich jednotlivé implementace.

PNG

Vstupními daty pro javascriptovou funkci zpracovávající PNG export jsou canvas, název souboru a text sestávající s informací o názvu úseku, poloze metru, čtrtmetru a zvolený referenční profil.

Nejprve zkompletujeme informační text spolu s vykreslenými daty na canvas. Vytvoříme si dočasný větší canvas, do kterého obě sady dat vložíme. Takto připravený canvas už stačí jen stáhnout pomocí dočasného elementu „<a>“ s atributem „download“.

```
var a = $("<a>").attr("href",
canvas.toDataURL()).attr("download", filename)
.appendTo("body");
a[0].click();
a.remove();
```

SVG

Při exportu SVG nepřidáváme žádné dodatečné textové informace a vstupními parametry jsou jen název souboru a canvas (v ukázce kódu ctx). Z canvasu vytvoříme Blob a analogickým způsobem ke stažení PNG ho přes element <a> stáhneme.

```
var mySerializedSVG = ctx.getSerializedSvg();
var svgBlob = new Blob([mySerializedSVG],
{ type: "image/svg+xml; charset=utf-8" });
var svgUrl = URL.createObjectURL(svgBlob);
var downloadLink = document.createElement("a");
downloadLink.href = svgUrl;
downloadLink.download = filename;
document.body.appendChild(downloadLink);
downloadLink.click();
document.body.removeChild(downloadLink);
```

PDF

V tomto případě opět přikládáme ke canvasu dodatečné textové informace jako při exportu PNG. Dokonce postupujeme také vytvořením nového canvasu, kde kombinujeme informace spolu s výřezem canvasu.

Tentokrát se však jedná o PDF a nastavení se zde trochu liší, v PNG jsme kupříkladu nenastavovali font a okraje když jsme do nich vkládali text a výřez. Výsledkem kombinace je pak obrázek, který můžeme vložit do PDF.

PDF vytváříme pomocí javascriptové knihovny jsPDF. Pak už jen vyvoláme akci ke stažení.

```
var imgData = canvas.toDataURL("image/jpeg", 1.0);
var pdf = new jsPDF(options);
pdf.addImage(imgData, 'JPEG', 0, 0);
pdf.save(filename);
```

CSV

U exportování dat do CSV se postup zcela liší oproti předchozím implementacím exportů. Cílem tohoto exportu je uložit všechny kartézské souřadnice bodů reprezentující okolí, nehledě na to, jestli kolidují s referenčním profilem, nebo zónou nejistoty.

Výsledný soubor pak může sloužit na tvorbu různých grafů, nebo pro práci s externími softwary, které mají uživatelé k dispozici. Data získáváme vysláním AJAXu a dotazem do databáze, který nám načte souřadnice.

V javascriptu pak data připravíme pro stažení a opět pomocí elementu „<a>“ csv soubor stáhneme. Název souboru tvoří unikátní id úseku a čtvrtmetru.

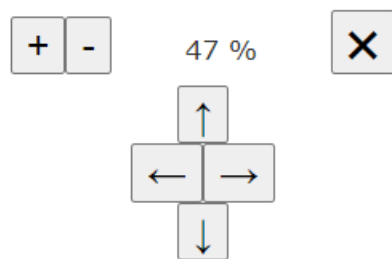
```
let csvContent = "data:text/csv;charset=utf-8,"
+ coordinates.map(e => Object.values(e).join(",")).join("\n")
var encodedUri = encodeURIComponent(csvContent);
var link = document.createElement("a");
link.setAttribute("href", encodedUri);
link.setAttribute("download", current_dm_id +
  '-' + current_qmp_id + '.csv');
document.body.appendChild(link);
link.click();
link.parentNode.removeChild(link);
```

3.5.6 Ovládací prvky

Ovládacími prvky canvasu jsou myšleny operace, které mění pohled na canvas, tj. různé posuny a přiblížení a oddálení výřezu. Vykreslená data na canvasu, jsou automaticky zaměřena kolem počátku s referenčním profilem.

Uživatele může ale zajímat širší okolí a tak se vložilo několik tlačítek a mechanismů, které mu umožňují se po canvasu „hýbat“. Na obrázku č. 3.11 můžeme vidět rozhraní s ovládacími prvky.

Všechny tyto operace jsou zprostředkovány pomocí javascriptu a po každé změně je canvas překreslen. Nedochozí však k žádnému znovunačítání dat, používáme nativní funkce škálování „scale“ a přesun „translate“.



- ❗ Uchopením myši se lze pohybovat po canvasu.
- ❗ Kolečkem myši se mění zoom.

Obrázek 3.11: Rozhraní s ovládacími prvky pro posouvání na canvasu.

Při posunech za pomoci tlačítek pak upravujeme proměnné uchovávající hodnoty x a y , které jsou defaultně nastaveny na $0,0$, což představuje počátek měření. Další možností pro pohyb po canvasu je uchopením pomocí myši. V javascriptu zachytáváme akce myši na canvasu:

- mousedown - stisknutí tlačítka myši
- mousemove - pohyb myši po canvasu
- mouseup - uvolnění stisku tlačítka

Když uživatel stiskne a okamžitě neuvolní tlačítko, připravujeme se na možný pohyb po canvasu. Nastavíme si booleanovou proměnnou, která je rovna `true` pokud je tlačítko stisknuto a `false` pokud je stisk uvolněn. Zároveň si uložíme pozice x a y kam uživatel klikl.

Poté zachycujeme pohyb myši, v případě, že je námi nastavená proměnná rovna `true`, pohybujeme se po canvasu, jinými slovy přepočítáváme souřadnice vůči pozici, kam uživatel klikl. Překreslování se odehrává téměř okamžitě.

Jakmile uživatel uvolní stisk tlačítka myši, canvas se ustálí a vyprázdníme, anebo resetujeme pomocné proměnné do původního stavu.

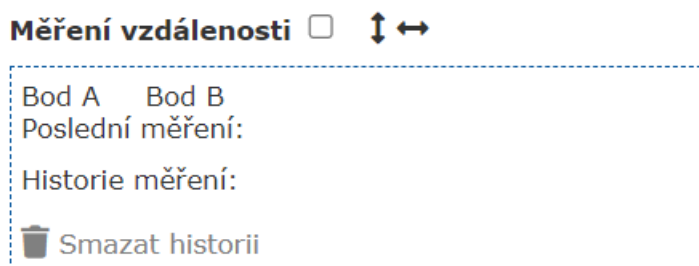
Další možností je měnit úroveň přiblížení canvasu a to už buď pomocí příslušných tlačítek, či kolečka myši. Pro tyto účely jsme vytvořily pole s hodnotami přiblížení, po kterých se uživatel postupně pohybuje jedním či druhým směrem.

Poslední ovládací prvek je resetování přiblížení a posunu do defaultního stavu. Zde se jedná jednoduše o přepsání hodnot do jejich původního stavu a překreslení canvasu.

3.5.7 Měření vzdáleností na canvasu

Měření vzdáleností na canvasu je spíše nástroj, než ovládací prvek a proto je oddělen do samostatné podkapitoly. Pomocí tohoto nástroje může uživatel měřit vzdálenost mezi dvěma vybranými body, přesněji pozicemi na canvasu.

Uživatel vybírá dva body pomocí kliknutí myši na canvas. Aby tato funkčnost nekolidovala s posunem po canvasu, je aktivovaná pouze když je zaškrtnut checkbox z rozhraní pro měření vzdáleností, které můžeme vidět na obrázku č. 3.12.



Obrázek 3.12: Rozhraní pro měření vzdálenosti mezi body.

Po aktivaci nástroje a kliknutí do canvasu označíme první bod za zaznamenaný. Jakmile uživatel vybere i druhý bod označíme i ten a spočítáme pomocí matematického vzorce vzdálenost mezi body.

$$\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

Obrázek 3.13: Vzorec pro výpočet vzdálenosti mezi dvěma body.

Vypočítanou hodnotu vepíšeme uživateli do rozhraní k textu „Poslední měření“ a uvolníme body k dalšímu měření. V případě, že už nějaké měření uživatel podnikl, posouváme předchozí měření pod „Historie měření“. Tlačítkem „Smazat historii“ pak hodnoty mažeme z paměti.

3.6 Simulace jízdy

Pro uživatele je výhodné mít možnost prohlížet si data, jako kdyby jel sám ve měřícím voze. Čelní fotografie a canvas mají na sobě nezávislé posuvníky a mohou být prohlíženy aniž by se ovlivňovali navzájem.

V případě simulace jízdy ale tyto dva prvky na stránce sjednotíme a posouváme je společně po úseku v časovém intervalu dvou vteřin. Každé dvě vteřiny posouváme čtvrtmetr na canvasu, kde při přechodu na další metr načítáme i další čelní fotografii.

Simulace jízdy v rozumném intervalu je možná jen díky úpravě vyhodnocování bodů canvasu pomocí metody vrhání paprsků, která markantně zrychlila načítání dat canvasu. Simulace pokračuje do konce úseku, nebo dokud ji uživatel nezastaví.

Tlačítko pro operaci se simulací má dva stavy: „Spustit simulaci“ a „Zastavit simulaci“. Tyto stavy mluví samy za sebe a není potřeba k nim dále něco víc dodávat.

3.7 Implementace grafu celého úseku

Předchozí dva prvky, jmenovitě prohlížení čelních fotografií a vykreslení bodového měření na canvas, jsme postupně načítali po částech, nikdy ne najednou pro celý úsek. Ani by to u nich nedávalo smysl z jejich podstaty.

Naopak na grafu vykreslujeme najednou data pro celý úsek, což pomáhá rychleji identifikovat místa s překročeními, či umožňuje identifikovat místa v zatáčce. Proklik do grafu pak načte pro příslušný metr čelní fotografii a bodové měření na canvas.

3.7.1 Knihovna Highcharts

Předtím, než si přiblížíme implementaci přípravy dat a jejich zobrazení, je důležité neopomenout hlavní nástroj, který pro zobrazení dat v podobě grafu v tomto modulu využíváme. A tím je javascriptová knihovna Highcharts.

Jak její název napovídá jedná se o knihovnu, pomocí které můžeme vytvářet různé typy grafů od nejjednodušších statických spojnicových po grafy překreslující se v reálném čase při přidání nových dat.

Nabízí spoustu možností jak grafy stylizovat a i kombinovat jejich typy v závislosti na konkrétních požadavcích pro vizualizaci. Tímto ji i vřele doporučujeme, pro větší i menší projekty.

3.7.2 Příprava dat v PHP

Výhodou, kterou nám knihovna Highcharts také přináší, je možnost kompletní přípravy dat v php, které pak v javascriptu předáme Highcharts jako parametr. Stačí jen znát správnou strukturu v jaké data předávat.

Když si představíme běžný spojnicový graf, vytanou nám na mysli dvě osy - x a y. Pokud ale potřebujeme více křivek, které se nemají navzájem překrývat, musíme si sami upravit měřítko a křivky na ose y posouvat a vyčlenit jim tak uměle jejich vlastní prostor.

Naštěstí knihovna Highcharts umožňuje pro jeden graf definovat několik samostatných os y, a nemusíme se tak starat o žádné posuny, aby se data nepřekrývala. Data pak jednoduše přiřadíme jejich příslušné ose y.

Dle potřeby tak můžeme i vymezit prostor, který každá osa na grafu zabírá. Můžeme tak například rozdělit tři osy tak, že jedna bude zabírat padesát procent výšky grafu a zbylé dvě si každá zaberou dvacet pět procent.

Graf na stránce prostorové průchodnosti rozděluje svůj prostor mezi pět ypsilonových os, které jsme uvedli v bodovém přehledu v podkapitole návrhu uživatelského rozhraní (viz. 3.3.3). V rámci této kapitoly si je dále přiblížíme.

Hodnoty na ose x

Než se k nim ale přesuneme, pojďme se ještě krátce věnovat ose x, která je pro všechny osy y společná. První tři křivky nabízejí svá data pro každý jednotlivý metr celého úseku.

Jak si však dovedeme představit, zobrazovat každý metr např. osmi kilometrového úseku by proměnilo osu x v nečitelnou směsici překrývajících se hodnot. I na toto má Highchart nativní řešení, kde můžeme nastavit periodu, po které se má hodnota na ose x objevit.

Problém nastává ve chvíli, kdy nechceme na ose x ukazovat její opravdové hodnoty, které si umí Highcharts samo rozdělit na požadované kusy. V případě našeho grafu se nám opravdové hodnoty právě nehodí, resp. potřebujeme zobrazovat jiné.

Přestože popisujeme implementaci, nevyhneme se zde troše železniční teorie. Obecně se používají pro železnice dva typy staničení (měření vzdáleností, kdy je vzdálenost místa měřena vždy k určitému bodu), interní staničení a externí staničení.

Interní staničení bývá zpravidla přesnější, jelikož se jedná o vzdálenosti jen měřeného úseku od začátku do konce. Externí staničení je pak počítáno přes více úseků a můžou se v něm tak vyskytovat drobnější odchylky.

Princip je pak takový, že chceme, aby byl graf a k němu příslušná data, orientován dle interního staničení, ale z důvodu praktičnosti pro uživatele zobrazovat staničení externí. Uveďme si na příkladu:

- Vezměme si úsek Pardubice - Přelouč, který dle externího staničení vede od 306.344 kilometru až po 317.046 kilometr. Pokud bychom uživateli zobrazovali na grafu interní staničení, překročení by se mohlo nacházet například na 2.312 kilometru.

Uživatel si by si jednoduše přičtením mohl vypočítat, že se překročení nachází na 308.656 kilometru, nicméně je to poněkud krkolomné uvažíme-li, že se má jednat o nástroj usnadňující práci.

Z tohoto důvodu se na grafu zobrazuje externí staničení, přestože jsou data orientovaná podle interního. Pro Highcharts, konkrétně pro parametr „ticklabels“, je nutno připravit asociativní pole ve tvaru *interní_km => externí_km*.

Jak již bylo zmíněno, Highcharts naše pole s náhradními hodnotami neumí sám rozdělit pro zobrazení, nicméně ale nabízí možnost, definovat asociativní pole s místy na ose x, na kterých má hodnotu zobrazit, pole opět připravujeme v PHP.

Dle délky úseku tuto hodnotu ukazujeme buď po sto, nebo pět seti metrech. Na příkladu s úsekem Pardubice - Přelouč bylo také vidět, že úseky nezačínají a nekončí vždy na celém kilometru.

To znamená, že je zapotřebí najít si první celý externí kilometr se zaokrouhlením buď na sto, či pět set metrů a od něj se pak posouvat o příslušnou hodnotu. Pro každý externí kilometr máme do dvojice interní kilometr.

Aby to nebylo tak jednoduché, úsek může být naměřen od začátku do konce, nebo od konce do začátku v závislosti na směru, kterým jel měřící vůz. Musíme tedy nejprve zjistit, zda jsou externí kilometry rostoucí, nebo klesající a podle toho hodnoty buď přičítat, nebo odečítat.

Po dokončení máme připravené druhé asociativní pole, opět e tvaru *interní_km => externí_km*, které předáme v Highcharts parametru „tickpositions“.

Křivky RK, KK a PK

Tyto křivky si sjednotíme pod stejnou sekci z důvodu stejné zdrojové SQL tabulky, což vede i na stejný způsob zpracování. Veličiny „Rozchod kolejí“, „Křivost kolejí“ a „Převýšení kolejí“ patří mezi základní parametry měření GPK (Geometrická poloha koleje).

Jejich hodnoty jsou měřené pro každý jednotlivý metr a tak tvoří velice jemná data. Po společném načtení z databáze je pak v cyklu rozdělujeme na jednotlivá dvojrozměrná pole. Vnitřní pole jsou asociativní pole ve tvaru *interní_km => hodnota*. S hodnotou kromě zaokrouhlení na tři desetinná místa dále nic nepodnikáme.

Výsledkem zpracování jsou tedy tři dvoujrozměrná pole.

Křivka převýšení v oblouku

Tato křivka v zásadě kopíruje data převýšení kolejí, s tím rozdílem, že data nejsou jemná a v poměru ku přibližně jedenácti tisícům hodnot (úsek Pardubice - Přelouč), máme hodnot jen několik desítek.

Po vzoru předchozích křivek je výsledkem dvojrozměrné pole, kde vnitřní pole jsou opět ve tvaru *interní_km => hodnota*. Na grafu sdílí křivky převýšení stejnou osu y a jsou rozlišeny barevně.

Křivka poloměru v oblouku

Tato křivka je strukturou dat velice podobná křivce převýšení v oblouku, jak jejich částečně společný název naznačuje. Rozdíl spočívá v tom, že data této křivky je potřeba pro vizualizaci formátovat.

Jsou totiž měřené v jiných jednotkách a oproti ostatním křivkám na grafu vizuálně neodpovídají. Vznikly tak dvě dvojrozměrná pole, kde jedno obsahuje nezměněná data, pro správné zobrazení hodnoty v tooltipu (přehledová tabulka grafu s hodnotami) a druhé s logaritmizovanými hodnotami pro vizualizaci křivky.

Body překročení

Na rozdíl od ostatních dat, které z jednotlivých bodů formují křivky, tato data zůstala čistě bodová a nebyla spojována. Jedná se zároveň o první data z importu

dat prostorové průchodnosti zobrazených na grafu.

Opět je formujeme jako dvoujrozměrné pole, kde pak hodnoty pro osu y jsou počty nalezených překročení pro daný metr úseku. Zde je dobré připomenout, že data překročení jsou rozlišována vůči vybranému referenčnímu profilu (viz. 3.2.7).

Úseky překročení

Součástí importovaných dat překročení jsou i informace o rozmezí od - do krátkých úseků, neboli oblastí, ve kterých se překročení nacházejí. Ty se na grafu zobrazují jako krátké úsečky a jsou umístěny na stejnou osu y jako „Body překročení“.

Highcharts v případě spojnicového grafu spojí všechny body spadající pod jednu datovou sérii, což se pro zobrazení několika úseček reprezentovaných v jedné datové sérii (jednom dvojrozměrném poli) moc nehodí.

Způsoby jak tento případ řešit jsou dva. Prvním z nich je rozdělení dat na několik samostatných datových sérií, kde každá z nich je jedna úsečka. To má nevýhodu v tom, že Highcharts je pak jednotlivě zapíše do legendy a obejít by se to muselo pracně formou vlastní legendy.

Druhý a nakonec i využitý způsob je rozřezat jednu datovou sérii na několik menších vložením null hodnot před počáteční a za koncový bod jednotlivých úseček. Zde bylo důležité brát i v potaz směr měření, který obrací který bod je počáteční a který koncový.

3.7.3 Zobrazení dat v javascriptu

V PHP jsme si připravili pole s hodnotami pro jednotlivé křivky, pole s pozicemi pro zobrazení externího kilometru a pole na převod interního kilometru na kilometr externí. Nyní stačí většinu údajů předat s dodatečným nastavením knihovně Highcharts, která nám graf vygeneruje.

V následujících částech si blíže projdeme vytvoření os y, nastavení vlastního tooltipu na míru, s čímž souvisí dopočítávání hodnot mezilehlých bodů pro křivky převýšení a poloměru v oblouku a další různá nastavení grafu.

Obecné nastavení grafu

Do stránky webové aplikace potřebujeme pro zobrazení grafu umístit nějaký element, nejlépe `<div>`, do kterého graf vložíme. Zároveň tím můžeme omezit velikost vytvářené grafu dle naší potřeby.

Grafu lze nastavit několik obecných vlastností, nicméně ty, které jsou pro tento graf důležité jsou jeho typ a možnosti přiblížení a posunu. Pokud není definováno jinak pro konkrétní křivku, typ bude spojnicový graf.

Dále vzhledem k tomu, že se nám jedná o několika kilometrový úsek trati, nemusí být vždy jednoduché trefit se na konkrétní metr, na který se uživatel chce podívat. Z toho důvodu je umožněno přibližování dle osy x. Jinými slovy je možno prohlížet část úseku detailněji.

Ve chvíli, kdy si uživatel nastaví vyhovující úroveň přiblížení, je užitečné, aby se mohl v úseku pohybovat, aniž by se nejdříve musel vrátit do defaultní úrovně zoomu a graf posunout a znovu přiblížit.

K tomu v Highcharts slouží parametry „panning“ a „panKey“. Jak je vidět na 3.6, nastavili jsme funkci pohybu na klávesu shift.

```
$( '#element' ).highcharts ({
  chart: {
    type: 'line',
    zoomType: 'x',
    panning: true,
    panKey: 'shift'
  },
```

Ukázka kódu 3.4: Obecné nastavení grafu

Mezi další obecné nastavení grafu pak můžeme zařadit i nadpis a podnadpis grafu. Přestože to v tomto případě není využito, je v Highcharts možné umístit nadpis, či podnadpis i pod graf, v případě, že je to více vyhovující.

Nadpisem grafu na prostorového průchodnosti je složení výsledku vyhledávání uživatele, tj. název úseku s kolejí a identifikačním číslem. Jeho podnadpis tvoří vybraný referenční profil.

```
title: {
  text: 'Nazev_useku'
},
subtitle: {
  text: 'Nazev_vybraneho_referencniho_profilu'
},
```

Ukázka kódu 3.5: Nastavení popisu grafu

Osa x

V nastavení osy x specifikujeme, kromě jejího popisku, hlavně jednotlivé pozice, na kterých se zobrazí hodnoty externího staničení. Pozice kilometru interního staničení jsme připravili v php v jednoduchém poli, které je předáno parametru „tickpositions“.

Parametr „labels“ s formátovací funkcí vypisuje místo kilometru interního staničení, staničení externí pomocí připraveného asociativního pole. Hodnota v „this.value“ představuje kilometr interního staničení.

```

xAxis: {
  title: {
    text: 'Staniceni',
  },
  tickPositions: tick_positions,
  labels: {
    formatter: function () {
      return km_labels[ this.value ];
    },
  },
},
},

```

Ukázka kódu 3.6: Nastavení osy x

Osa y

V předchozích částech jsme zmínili pět os y, technicky vzato se stále jedná o jednu osu y, kterou rozdělíme na několik menších. Každé menší ose pak udáváme kolik prostoru na grafu zabere, v našem případě při pěti osách je to dvacet procent.

Zde je důležité podotknout, že přestože zabírají určitý prostor osy y, každá z nich se pak škáluje dle hodnot, které jsou k ní přiřazeny, ať už jsou to hodnoty jen kladné, záporné či obojí.

Pak ještě všem osám kromě první určujeme jejich posun na celkové ose y, opět také v procentech. Druhá osa bude posunuta o dvacet procent, třetí o čtyřicet atd. Osám se pak nastavuje název, který se zobrazuje jako popisek nalevo od grafu v prostoru křivky.

Parametr „plotLines“ nám umožňuje definovat úsečky po celé délce grafu, v tomto případě na jedné z dílčích os y. Na ukázce 3.7 vytváříme černou úsečku pro zvýraznění hodnoty nula, která poslouží pro lepší orientaci na křivce.

Parametrem „gridLineDashStyle“ nastavuje styl zobrazení úseček, které oddělují dílčí osu y od ostatních.

```

yAxis: [{
  title: {
    text: "Nazev_krivky",
  },
  height: '20%',
  lineWidth: 2,
  plotLines: [{
    color: 'black',
    width: 2,
    value: 0
  }
]

```

```
    }],  
    gridLineDashStyle: 'longdash',  
  },  
},
```

Ukázka kódu 3.7: Nastavení osy y

Datové sady

V Highcharts se data nepředávají přímo při definici os y, ale slouží pro to samostatná sekce. Datovým sadám, které předáváme jako dvojrozměrné pole z php, můžeme totiž nastavit také několik parametrů, které mohou být rozdílné od jiných křivek okupujících stejnou osu y.

I zde definujeme popisek křivky, tento je však zobrazen v legendě. Parametrem „yAxis“ udáváme, na které ose se data zobrazí a obecně není žádný limit pro počet datových sad pro jednu osu y, snad jen výkonostní.

Velice užitečnou vlastností, kterou Highcharts nabízí, je definování vlastních akcí při kliknutí na bod křivky. V podkapitole 3.5. se zmiňujeme o přenačtení canvasu při kliknutí do grafu.

Funkce na ukázce č. 3.8. pojmenovaná „move_function“ získává jako parametr x pozici bodu, kterou pak využije pro identifikace metru k načtení do canvasu.

Vzhledem k tomu, že je graf orientován datově po metrech a ne po čtvrtmetrech, načte se do canvasu vždy první čtvrtmetr. Metoda pak také přenačte příslušnou čelní fotografii, doplňující tak vizuální přehled o vybraném místě.

Funkce „create_vertical_line“ využívá parametru „plotLines“, jehož jednoduchý příklad jsme mohli vidět na ukázce č. 3.7, s tím rozdílem, že ho nyní nastavuje pro osu x a hodnotou je x pozice bodu, na který uživatel klikl.

Vznikne nám tak vertikální úsečku označující místo, kam uživatel klikl, opět to slouží pro větší přehlednost. Při dalším prokliku na jiný bod se stará úsečka smaže a na novém místě se vytvoří nová.

```
series: [{  
  name: 'Nazev_krivky',  
  yAxis: 0,  
  data: data,  
  states: {  
    inactive: {  
      opacity: 1  
    }  
  },  
  point: {  
    events: {  
      click: function () {  
        move_function(this.x);  
      }  
    }  
  }  
}]
```



```

        create_vertical_line( this.x );
    }
}
},

```

Ukázka kódu 3.8: Nastavení datové série

Tooltip

Nativní tooltip Highcharts bohužel není stavěn na to, aby zobrazoval data všech křivek, dle místa na ose x, na kterém se uživatel myši nachází. Je schopen zobrazovat data jen pro křivku, na kterou má uživatel najeto myši.

Pro účely zobrazení dat všech křivek jen dle x souřadnice je potřeba definovat vlastní tooltip, což Highcharts také umožňuje. Využije se přitom také formátovací funkce tooltipu, kde pro nás směrodatným parametrem je hodnota na ose x.

Kompletujeme dohromady html tabulku, ve které zobrazujeme kilometr externího staničení, k čemuž opět využíváme asociativního pole pro převod mezi typy staničení, a pak hodnoty jednotlivých křivek.

Na ukázce č. 3.9 můžeme vidět, že hodnoty křivek kompletujeme ve dvou javascriptových funkcích. Funkce „add_pk_rad“ připravuje data do tabulky pro křivky převýšení a poloměru v oblouku.

Funkce „tooltip_creation“ připravuje data pro všechny zbylé křivky. Důvodem tohoto rozdělení je, že pro křivky v oblouku dopočítáváme hodnoty mezi jednotlivými body. Jak jsme popsali v podkapitole č. 3.7.2, jsou tvořeny oproti jiným křivkám značně menším počtem bodů.

```

tooltip: {
  backgroundColor: null,
  borderWidth: 0,
  shadow: false,
  useHTML: true,
  crosshairs: true,
  shared: true,
  headerFormat: '<small>{point.key}</small><br>',
  formatter: function () {
    var text = tooltip_creation( this.points, this.x, data );
    var text_pk_rad = add_pk_rad( this.x, data_pk_rad )
    var tooltip_html = ''
      + '<table_id="tooltip">'
      + '<tr><td_colspan="3">'
      + km_labels[ this.x ] + '_km_ext<td><tr>'
      + text
      + text_pk_rad

```

```

    + '</table>'
;
return tooltip_html;
},
footerFormat: '</table>',
  style: {
    padding: 0
  }
},

```

Ukázka kódu 3.9: Nastavení tooltipu

Princip funkce „tooltip_creation“ je poměrně jednoduchý. V cyklu procházíme všechny datové série, ve kterých hledáme hodnotu x , na které se právě uživatel nachází. Pokud hodnota existuje přidáme do návratové proměnné tři sloupce html tabulky.

První sloupec obsahuje bod obarvený na odpovídající barvu křivky, druhý sloupec obsahuje název datové série a poslední obsahuje hodnotu s jednotkami, ve kterých je uvedena (většinou mm). Data rozdělujeme na tři sloupce z důvodů formátování textu.

Ve funkci „add_pk_rad“ procházíme dvě datové série, kde u dat poloměru v oblouku využíváme skutečná nelogaritmizovaná data. Pro funkci mohou nastat dvě možnosti, buď se nacházíme přímo pozici bodu, nebo někde mezi dvěma body.

Vyjetí mimo interval je omezeno podmínkou a zároveň by se nemělo stávat, pokud někde nechybí data. Pokud se nacházíme na pozici bodu, jednoduše vrátíme jeho hodnotu, pokud ne hodnotu k zobrazení vypočítáme pomocí vzorce:

$$y = (p1_y - p2_y) \times (1 - (pos - p1_x) \div (p2_x - p1_x)) + p2_y$$

Obrázek 3.14: Vzorec pro výpočet y hodnoty mezi dvěma body.

Kde „pos“ je x souřadnice, na které se nacházíme a $p1$ a $p2$ představují dva body, mezi kterými se nacházíme. Vypočítanou hodnotu kompletujeme do tří sloupců stejně jako ve funkci „tooltip_creation“.

Celkovou html tabulku s hodnotami pak zobrazíme uživateli.

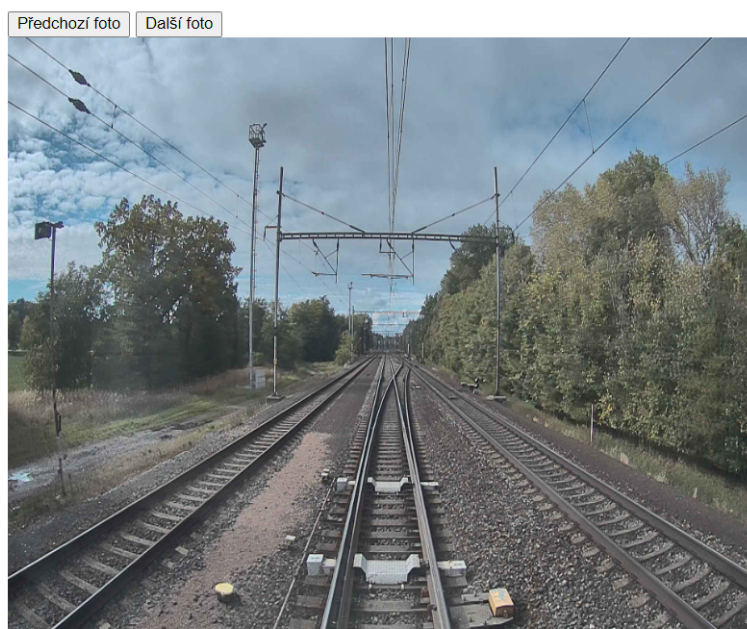
Kapitola 4

Ukázky vizualizace

V předchozí kapitole jsme si přiblížili implementaci jednotlivých nástrojů. V této kapitole se podíváme na ukázky hotových nástrojů a na příklady jejich použití. Zde bychom rádi poznamenali, že se zde nejedná o aktuální data a jejich stav vybraný úsek nereflektuje.

4.1 Čelní fotografie

Na obrázcích č. 4.1 a 4.2 můžeme vidět čelní fotografie nafocené v úseku Pardubice - Přelouč, dokreslující bodovou vizualizaci v různých místech úseku. U druhé fotografie se nám naskýtá i pohled na část úseku v mírném oblouku.



Obrázek 4.1: Ukázka čelní fotografie 1: úsek Pardubice - Přelouč.

Předchozí foto

Další foto



Obrázek 4.2: Ukázka čelní fotografie 2: úsek Pardubice - Přelouč. v mírném oblouku.

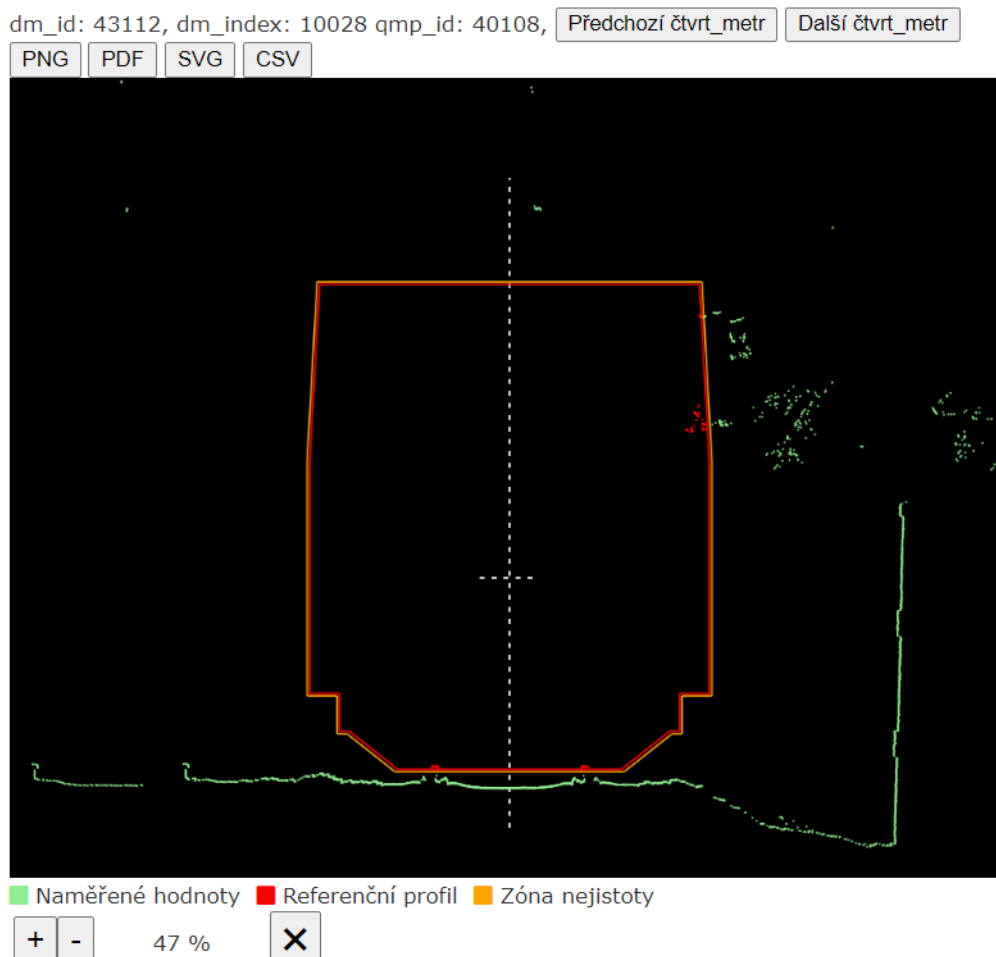
4.2 Zobrazení bodového měření na canvas

V rámci implementace jsme si popsali vykreslení bodového měření na canvas a k němu jeho příslušné ovládací prvky. V následujících částech na ukázkách uvidíme jejich realizaci.

4.2.1 Referenční profil

Na obrázku č. 4.3 můžeme vidět referenční profil posazený do středu měření na koleje. Zelenou barvou reprezentujeme bodové měření daného čtvrtmetru. Červená a oranžová pak slouží pro referenční profil a jeho zónu nejistoty.

Na ukázce si můžeme všimnout, že několik bodů koliduje s referenčním profilem a jsou obarveny červeně. V tomto případě se jedná poměrně zřetelně o listoví stromu. Posuny a další operace s referenčním profilem si ukážeme na stejném čtvrtmetru pro zachování reference.



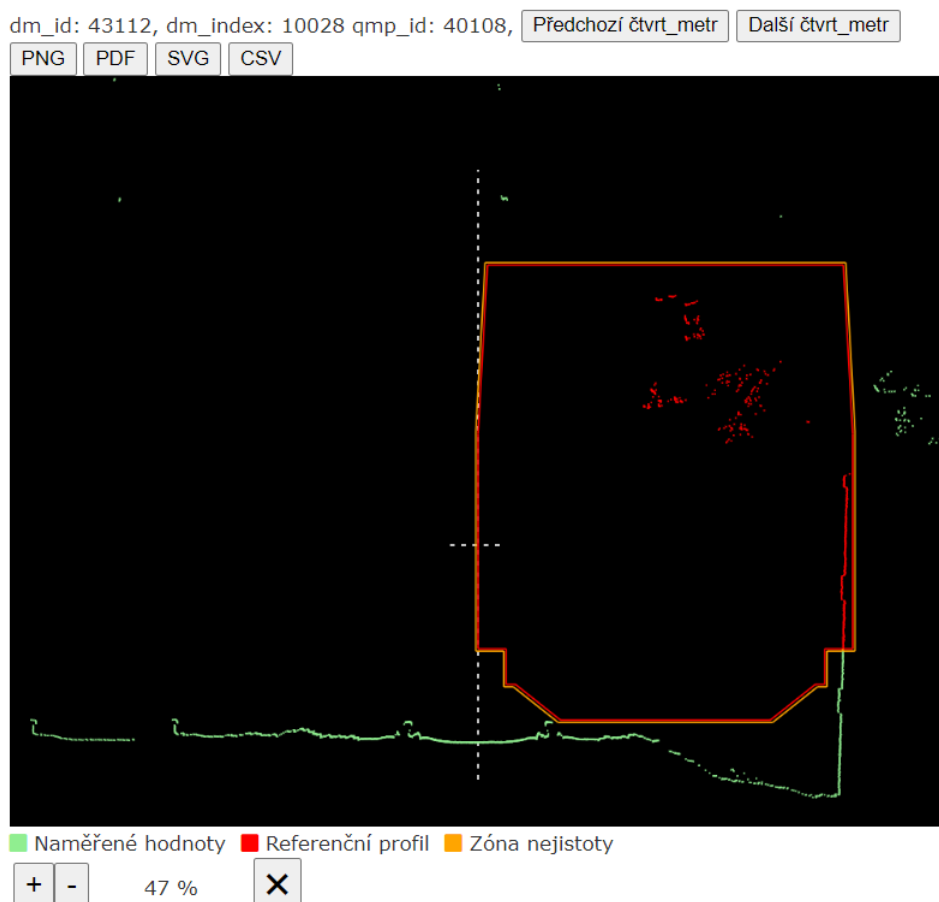
Obrázek 4.3: Vykreslení bodových měření na canvas spolu s referenčním profilem a zónou nejistoty.

Základní operací s referenčním pohybem je jeho posun po ose x a y. Na obrázku č. 4.4 můžeme vidět rozhraní, ve kterém jsme pro tuto ukázkou nastavili posun referenčního profilu o dva metry doprava a o padesát milimetrů vzhůru.

Po přepočítání jsou identifikovány nové body překročení a jsou adekvátně obarveny dle překročení zóny nejistoty, nebo referenčního profilu samotného. Ukázka na obrázku č. 4.5.

Referenční profil: Z_GC	Přepočítat
Zobrazit subprofily:	<input type="checkbox"/>
Horizontální překlopení:	<input type="checkbox"/>
Zóna nejistoty:	<input checked="" type="checkbox"/> 25 mm
Posun x:	2000 mm
Posun y:	50 mm
Převýšení:	0 mm (0°)

Obrázek 4.4: Rozhraní pro práci s referenčním profilem s nastavením posunu o dva metry doprava.

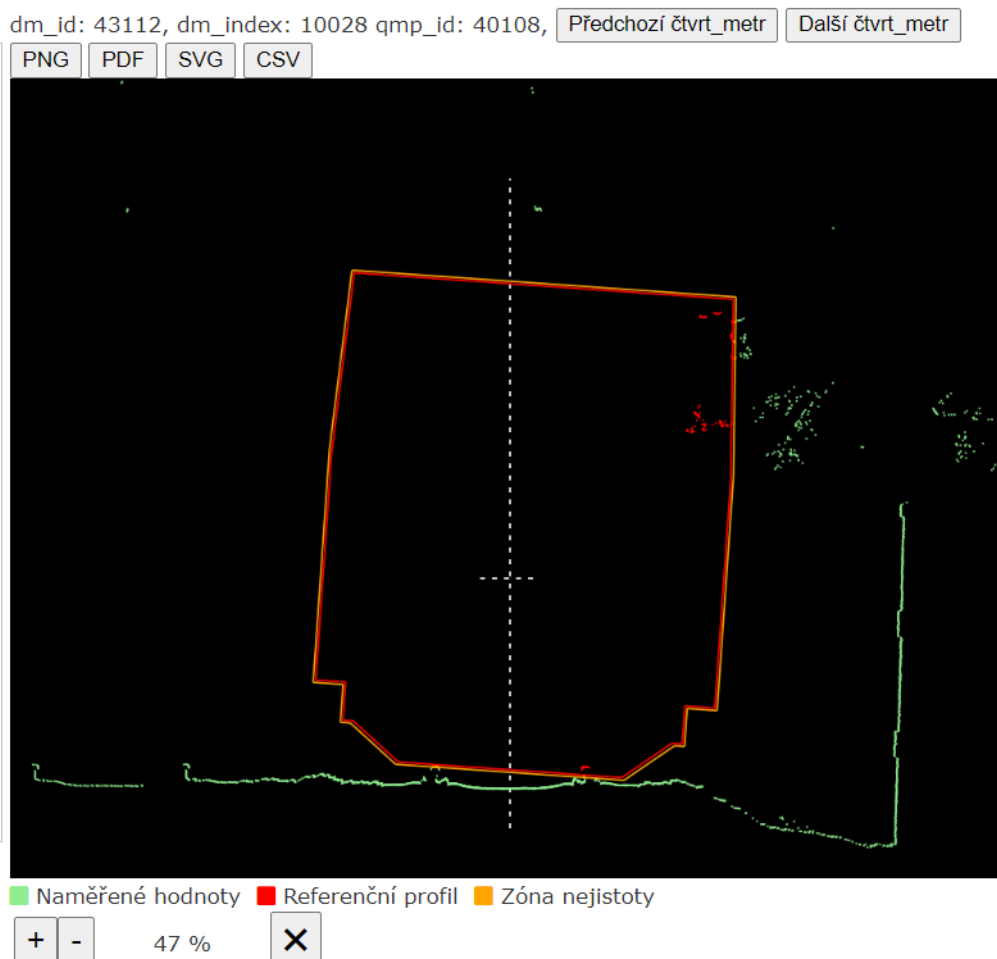


Obrázek 4.5: Vykreslení bodových měření na canvas spolu s referenčním profilem a zónou nejistoty posunutých o dva metry doprava.

Dalším z úkonů je nastavení převýšení. I zde vidíme na obrázku č. 4.6 nastavení v rozhraní, které automaticky přepočítá převýšení sto dvaceti milimetrů na pět stupňů. Na obrázku č. 4.7 pak vidíme realizaci přepočítání.

Referenční profil: Z_GC	<input type="button" value="Přepočítat"/>
Zobrazit subprofily:	<input type="checkbox"/>
Horizontální překlopení:	<input type="checkbox"/>
Zóna nejistoty:	<input checked="" type="checkbox"/> 25 mm
Posun x:	0 mm
Posun y:	0 mm
Převýšení:	120 mm (4.59°)

Obrázek 4.6: Rozhraní pro práci s referenčním profilem s nastavením převýšení o pět stupňů.

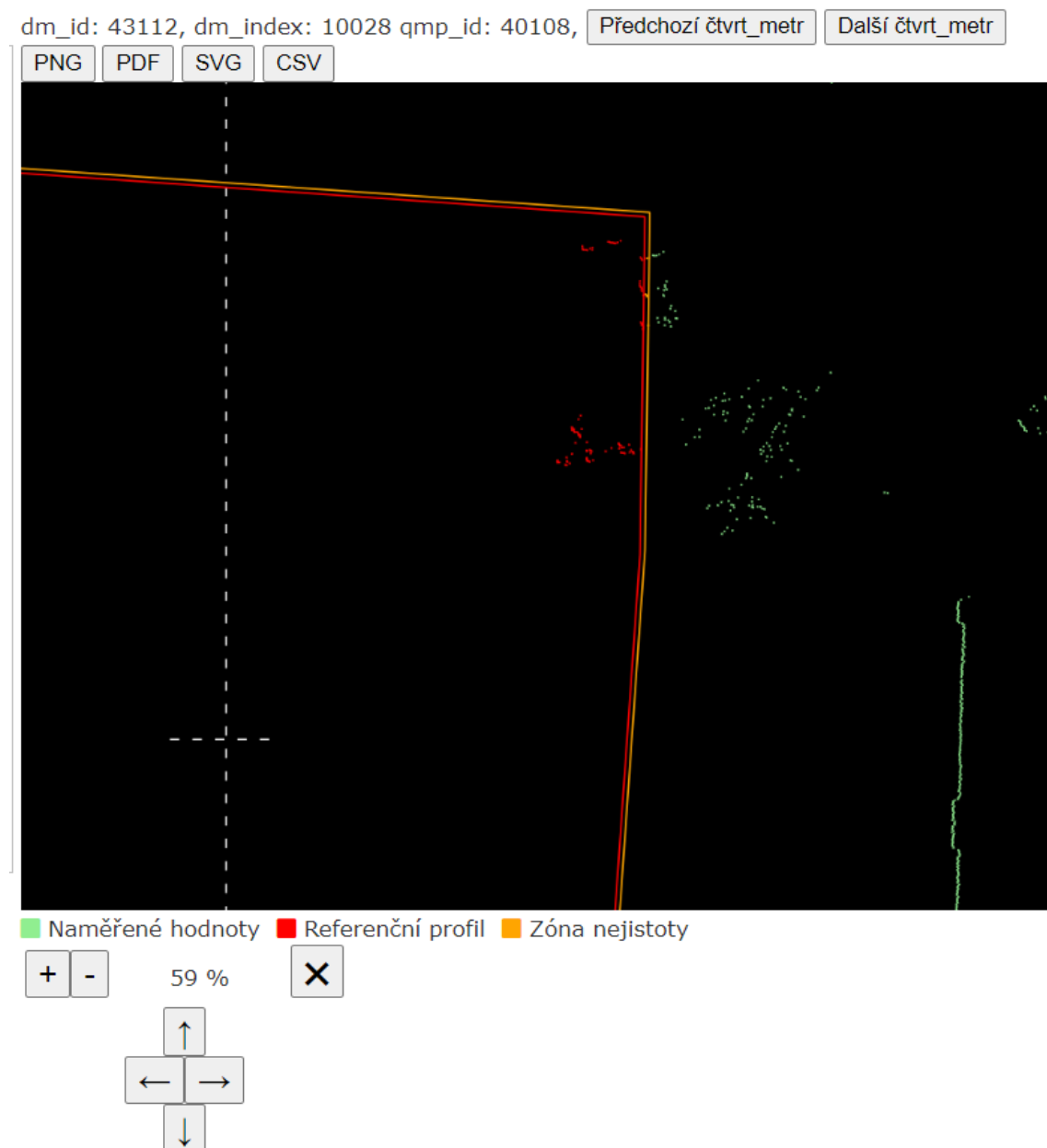


Obrázek 4.7: Vykreslení bodových měření na canvas spolu s referenčním profilem a zónou nejistoty s převýšení o pět stupňů.

4.2.2 Ovládání canvasu

Pro canvas je zprostředkováno několik operací umožňujících uživateli si zobrazenou oblast různě přiblížovat, či oddalovat, nebo posouvat všemi směry dle libosti. Tyto operace je možno provádět buď pomocí připravených tlačítek umístěných pod canvasem, či pomocí myši.

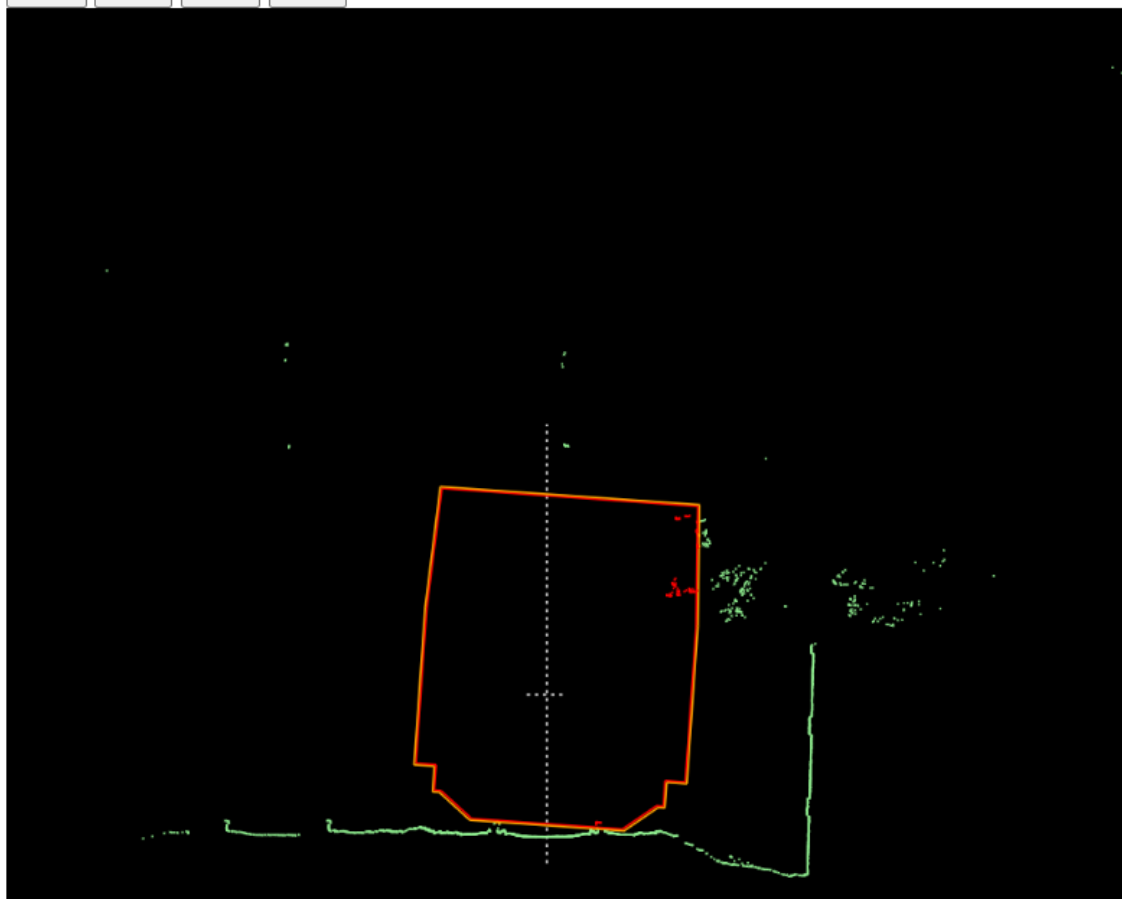
Na obrázcích č.4.8 a 4.9 pak můžeme vidět příklad právě těchto operací.



Obrázek 4.8: Ukázka přiblížení a posunu canvasu.

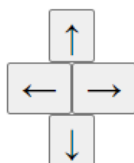
dm_id: 43112, dm_index: 10028 qmp_id: 40108, [Předchozí čtvrt_metr](#) [Další čtvrt_metr](#)

[PNG](#) [PDF](#) [SVG](#) [CSV](#)



■ Naměřené hodnoty ■ Referenční profil ■ Zóna nejistoty

[+](#) [-](#) 24 % [×](#)



Obrázek 4.9: Ukázka oddálení a posunu canvasu.

4.2.3 Měření vzdáleností na grafu

Jedním z nástrojů pro práci s canvasem je i měření vzdálenosti mezi dvěma body. Po zakliknutí checkboxu se při kliknutí myši do canvasu označí postupně dva body, tj. bod A a bod B, načež následuje výpočet vzdálenosti.

V paměti uchováváme několik těchto měření pro možnosti porovnání provedené uživatelem. Ve chvíli, kdy historii měření uživatel nepotřebuje, může ji uvolnit z paměti tlačítkem „smazat historii“.



Obrázek 4.10: Ukázka zapsání vzdáleností mezi dvěma body na canvasu.

4.3 Simulace jízdy

V předešlých podkapitolách jsme si ukázali samostatné komponenty čelních fotografií a bodového vykreslení na canvas. V implementaci jsme věnovali podkapitolu popisu simulace jízdy, která tyto dvě komponenty spojuje dohromady.

Na obrázku č. 4.11 můžeme vidět ukázkou synchronizace míst měření úseku na obou komponentách a zároveň také jejich umístění vůči sobě v celkovém rozhraní prostorové průchodnosti.

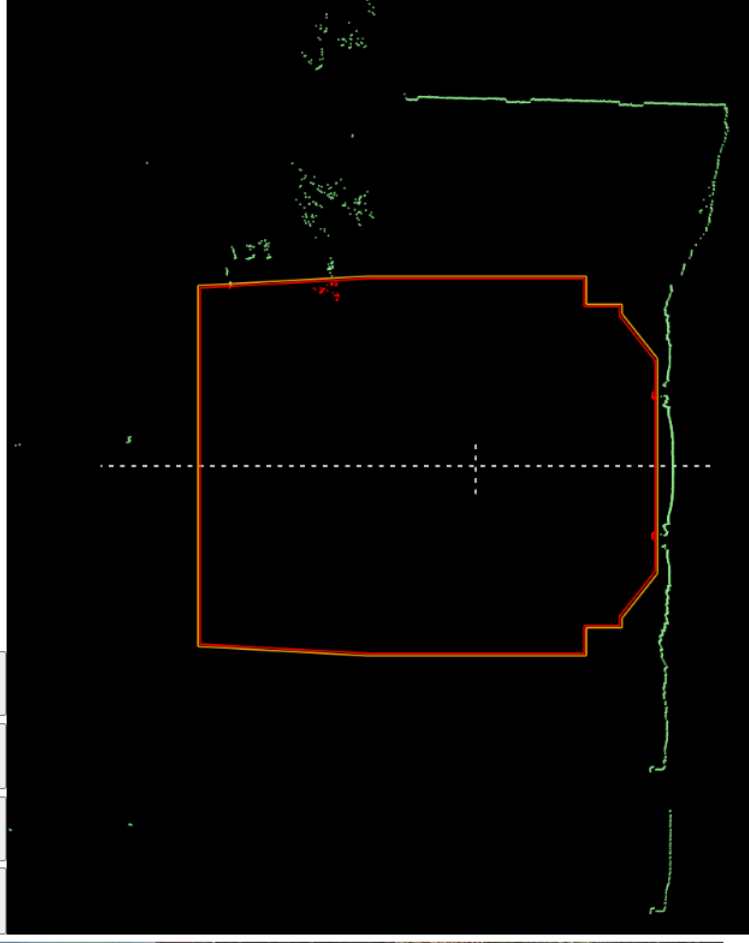
Předchozí foto

dm_id: 43112, dm_index: 10028 qmp_id: 40108, Předchozí čtvrt_metr

[PNG](#) [PDF](#) [SVG](#) [CSV](#)

[Dalsí foto](#)

[Dalsí čtvrt_metr](#)



[Naměřené hodnoty](#) [Referenční profil](#) [Zóna nejistoty](#) [Zóna nejistoty](#)

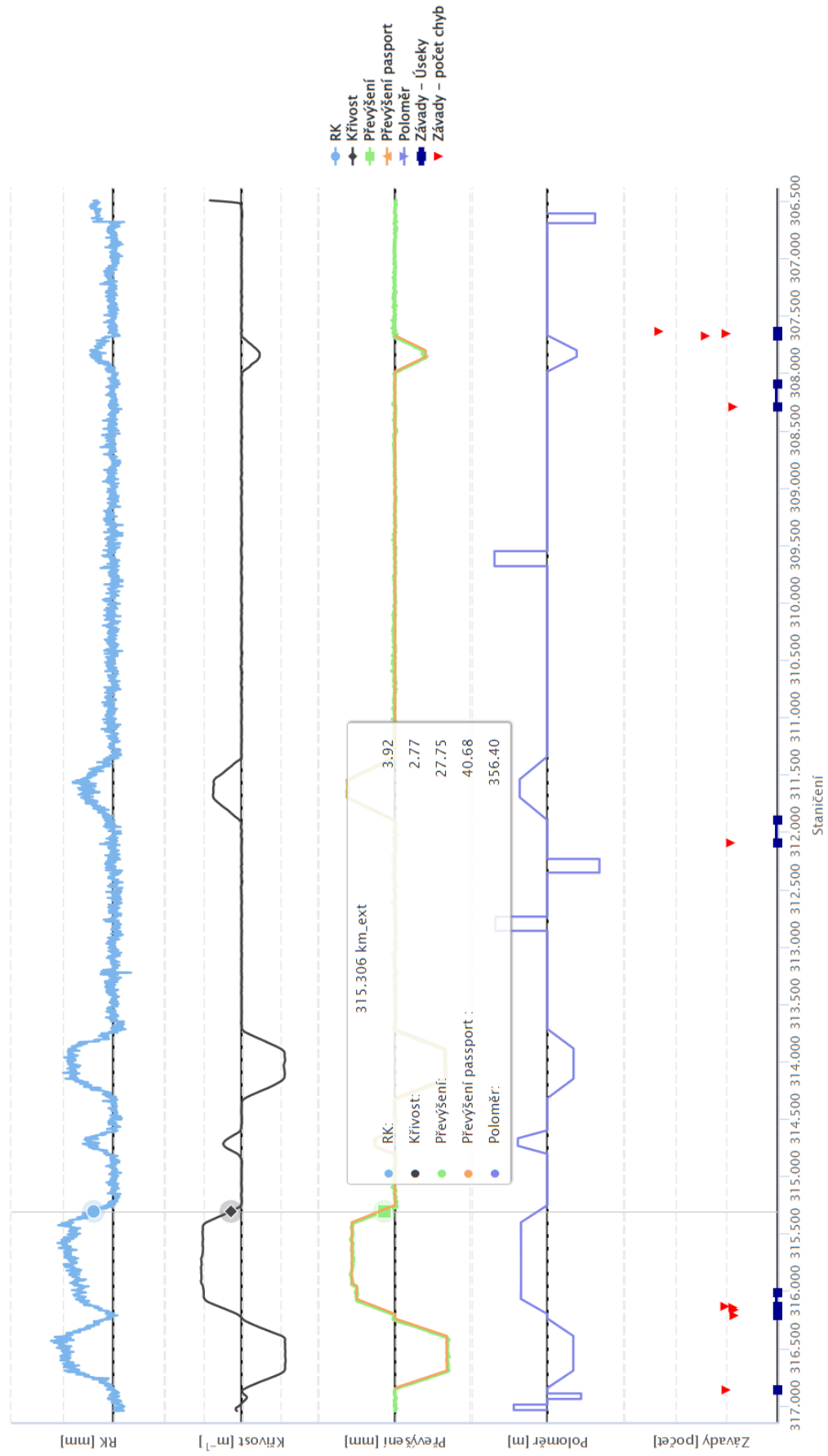
Obrázek 4.1.1: Ukázka simulace jízdy.

4.4 Graf měřeného úseku

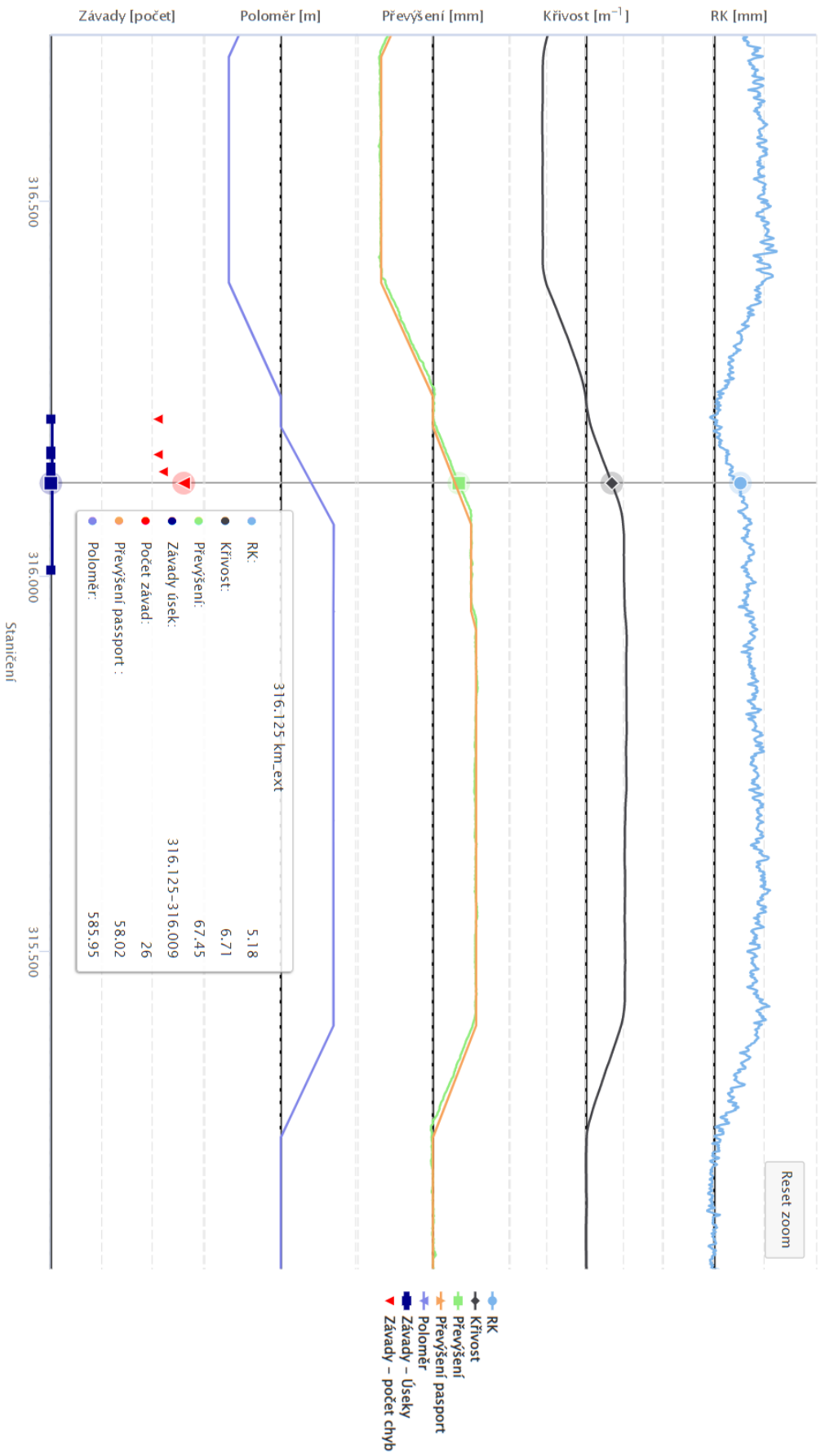
Na obrázku č. 4.12 můžeme vidět vizualizaci dat celého naměřeného úseku rozděleného do několika dílčích os y podle měřené veličiny. Rozchod, křivost, převýšení (obou typů) a poloměr jsou měřeny měřicím vozem pro železniční svršek (MVŽS).

Data závad představují data překročení, čili problematických míst prostorové průchodnosti, jsou měřena fotogrametrickým strojem, jehož obrázek nalezneme v kapitole č.2.

Detail přiblížení se zaměřením na oblast s překročením prostorové průchodnosti je pak možno vidět na obrázku č. 4.13.



Obrázek 4.12: Graf úseku Pardubice - Přelouč.



Obrázek 4.13: Graf úseku Pardubice - Přelouč přibližně.

Závěr

V této práci jsme si představili základní poznatky a principy k vytvoření podnikového datového skladu, blíže jsme se věnovali zejména procesu ETL, který je nedílnou součástí každého datového skladu.

V druhé kapitole jsme se věnovali přiblížení tématu praktické části z oboru železniční dopravy a to konkrétně měření prostorové průchodnosti. Vysvětlili jsme si zde jakým způsobem měření probíhají a objasnili jsme si dále pojmy referenčního profilu a pojistné zóny.

Samotnou implementaci pokrývá třetí kapitola. Uvádíme zde celý proces realizace nového rozšiřujícího modulu v praxi provozovaného datového skladu diagnostiky.

Proces implementace je pro přehlednost rozdělen do tří částí. V první části zabýváme samotným procesem importování dat prostorové průchodnosti do systému, který ztělesňuje všechny tři fáze procesu ETL, extrakci, transformaci a načtení dat.

V druhé části se zabýváme návrhem uživatelského rozhraní prostorové průchodnosti. Zde představujeme vhodné způsoby vyhledávání v železniční síti České republiky a nástroje pro práci s naměřenými daty.

V třetí části této kapitoly se zabýváme implementací nástrojů představených v popisu rozhraní. Věnujeme se zde vyhodnocováním měření vůči referenčním profilům, které bylo dotaženo k dokonalosti z hlediska výkonu pomocí metody vrhání paprsku (anglicky ray casting), zobrazením čelních fotografií a realizací grafů naměřených veličin pro celý úsek.

Finální kapitola je věnována ukázkám vizualizace a práce se všemi vytvořenými nástroji pro uživatele nad naměřenými daty. Nástroj se v praxi využívá a je součástí systému využívaného přibližně již dvanácti sty pracovníky oboru diagnostiky a správy trati u správce železniční infrastruktury v době psaní této práce.

Správnou analýzou zdrojových dat a potřeb uživatelů jsme sestavili efektivní modul diagnostiky prostorové průchodnosti a vytvořili jsme tak užitečné a přehledné rozšíření diagnostického datového skladu.

Literatura

- [1] DEVLIN, B. *Data warehouse: from architecture to implementation*, Reading, Mass: Addison-Wesley, c1997. ISBN 0201964252.
- [2] LIŠKA, T. *The Cluster and GRID Computing on Mass Data Production Systems for High Energy Physics Experiments*, CERN, ČVUT FJFI, 2009
- [3] *Nakládací směrnice UIC*, Internationaler Eisenbahnverband, 2021 ISBN 978-2-7461-3044-9
- [4] NOVÝ, J. *Zpracování velkých objemů dat v rámci fyzikálního experimentu COMPASS*, disertační práce, ČVUT FJFI, 2019
- [5] *Obrys vozidel a průjezdný průřez* [online; cit. 2022-10-20], Dostupné z WWW: <<http://www.trminek.cz/view.php?navezvclanku=obrys-voziel-a-prujezdnny-prurez&cislocclanku=2004080003>>
- [6] SPRÁVA ŽELEZNIC. *Prostorová průchodnost*, Správa železnic, státní organizace, 2021
- [7] SYCHROVSKÝ, P. *Systém diagnostiky prostorové průchodnosti tratí*, Vědecko-technický sborník ČD č26, 2008
- [8] ŠVAMBERG M. *Analýza a zpracování velkých objemů dat*, Praha, 2021. Výzkumný úkol. ČVUT, FJFI, KSI. Vedoucí práce Ing. Tomáš Liška, Ph.D.
- [9] TOMEK, R. TRUBÁK, A. *Měření prostorové průchodnosti tratí*, Hradec Králové, 1999
- [10] ŽABA, A. *Datový sklad s aplikací v letecké dopravě*, diplomová práce, ČVUT FJFI, 2017