



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Softwarový systém pro efektivní vytvá ení prezentací
Student:	Lukáš Haberzettl
Vedoucí:	Ing. Robert Pergl, Ph.D.
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Cílem práce je návrh a implementace systému pro vytvá ení škálovatelných vektorových prezentací. ešení se bude skládat z následujících ástí:

- 1) P íprava textového obsahu: vyberte vhodný formát a navrhnete strukturu souboru.
- 2) P íprava grafických podklad pro prezentaci bude probíhat v programu Inkscape. Vyžadována je implementace pluginu pro:
 - a) navázání textového obsahu z bodu 1) na grafické elementy,
 - b) vymezení rámc a p echod mezi nimi.
- 3) Export výsledné prezentace do HTML + CSS + Javascript + SVG. Ve výsledné prezentaci budou vyrenderovány texty z bodu 1) a bude možné ovládat prezentaci z hlediska p echod mezi rámci dle bodu 2b), nebo volného zoomování a posouvání.

Prove te analýzu, návrh a implementaci ešení. Výsledek ádn zdokumentujte, otestujte a demonstруйте na p íkladu.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 23. b ezna 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Softwarový systém pro efektivní vytváření vektorových prezentací

Lukáš Haberzettl

Vedoucí práce: Ing. Robert Pergl, Ph.D.

15. května 2017

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce, panu Ing. Robert Pergl, Ph.D. za jeho ochotu, čas a odborné rady, které mi velmi pomohly. Dále bych chtěl poděkovat mé rodině, přátelům a mé přítelkyni za jejich oporu v průběhu celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Lukáš Haberzettl. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Haberzettl, Lukáš. *Softwarový systém pro efektivní vytváření vektorových prezentací*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací systému pro efektivní vytváření vektorových prezentací. Nejprve byla provedena analýza problému zahrnující rozbor existujících řešení, detailnější specifikaci požadavků na systém a rešerši použitelných technologií. Na základě této analýzy byl navrhnout jednotný a efektivní proces přípravy vektorové prezentace kladoucí důraz především na separaci přípravy obsahové a grafické části. Výsledkem práce je rozšíření programu Inkscape, které umožňuje takový proces aplikovat a jehož výstupem je vektorová prezentace ve formě HTML souboru spustitelná v prostředí webového prohlížeče.

Klíčová slova softwarový systém, vektorová prezentace, Inkscape, plugin

Abstract

This thesis deals with design and implementation of software system for creating vector-based presentations. Based on analysis of the problem, including description of existing solutions, system requirements specification and research of useful technologies, presentation preparation process was designed with focus on separation of content. Result of this thesis is an extension developed for Inkscape software, that can apply this process. The output of the extension is vector-based presentation in form of HTML file executable in web browser.

Keywords software system, vector presentation, Inkscape, plugin

Obsah

Úvod	1
1 Cíl práce	3
1.1 Nefunkční požadavky	3
1.2 Funkční požadavky	3
2 Analýza	5
2.1 Existující řešení	5
2.2 Funkční požadavky	10
2.3 Analýza případů užití	15
2.4 Technologie	18
2.5 Shrnutí	22
3 Návrh	23
3.1 Volba technologií	23
3.2 Proces přípravy prezentace	24
3.3 Návrh obsahového souboru	25
3.4 Plugin programu Inkscape	28
3.5 Prezentace	35
3.6 Shrnutí	39
4 Realizace	41
4.1 Plugin programu Inkscape	41
4.2 Prezentace	43
4.3 Shrnutí	44
5 Testování	45
5.1 Současný stav řešení	45
5.2 Nedostatky	45
5.3 Shrnutí	46

Závěr	47
Možná vylepšení implementovaného řešení	47
Literatura	49
A Seznam použitých zkratk	51
B Výstup závěrečného testování	53
B.1 Výstup č. 1	53
B.2 Výstup č. 2	54
B.3 Výstup č. 3	54
B.4 Výstup č. 4	56
C Instalační příručka	57
C.1 Podporované operační systémy	57
C.2 Požadavky	57
C.3 Instalace	57
C.4 Další nástroje nutné pro stoprocentní využití pluginu	59
D Použité programy	61
E Obsah příloženého CD	63

Seznam obrázků

2.1	Prostředí pro vytvoření grafických podkladů v programu Prezi . . .	7
2.2	Náhled výsledné flash prezentace v prostředí programu Prezi . . .	7
2.3	Diagram případů užití	15
3.1	Diagram aktivit znázorňující proces přípravy prezentace	24
3.2	Návrh sekce Order	29
3.3	Návrh sekce Group	30
3.4	Návrh sekce Export	30
3.5	Návrh sekce s nápovědou Help	31
3.6	Návrh uživatelského rozhraní prezentační komponenty	36
4.1	Implementované uživatelské rozhraní pluginu	42
4.2	Výsledná prezentace spuštěná ve webovém prohlížeči	44
B.1	Výstup bez specifikace dodatečných stylů pro snímky reprezentované kruhem	53
B.2	Výstup bez specifikace dodatečných stylů pro snímky reprezentované obdélníkem	54
B.3	Výstup bez specifikace dodatečných stylů pro snímky reprezentované kruhem	55
B.4	Výstup systému pro snímky reprezentované kruhem po aplikaci stylů	55
B.5	Výstup systému pro topic s oběma druhy obsahu bez aplikování dodatečných stylů	56
B.6	Výstup systému pro topic s oběma druhy obsahu po aplikování dodatečných stylů	56
C.1	Nabídka rozšíření v programu po úspěšném nahrání pluginu Farslide	59

Seznam tabulek

2.1	Porovnání existujících řešení	9
3.1	Definice obsahové gramatiky pomocí EBNF	27
3.2	Dostupné UI prvky dle specifikace INX	28
3.3	Atributy sloužící k propojení obsahu s grafickým dokumentem . . .	32
3.4	Objekt reprezentující jeden snímek v prezentaci	37

Úvod

Téma tvorby elektronických prezentací je dnes velmi obsáhlé. Je mnoho způsobů jak připravit podpůrné prezentační materiály pro vlastní výklad. Nejčastější formou je dokument se stránkami (snímky), mezi kterými se přednášející v průběhu své přednášky pohybuje lineárně. Ačkoliv tento typ k vysvětlení vykládané problematiky často bohatě stačí, existuje mnoho případů, kdy tato forma už tak vhodná není. Jde především o prezentace s výskytem různých diagramů nebo postupů, u kterých je potřeba pro jejich pochopení znát vazby mezi jednotlivými fázemi a mít možnost mezi nimi procházet ve více směrech, popř. oddálit pohled a vidět větší celek (více snímků) probíraného materiálu. Takových prezentací však dnes vidíme po málu zejména proto, že neexistuje mnoho nástrojů, které by se touto problematikou zabývali.

Tato práce je určena jako řešení pro akademickou skupinu „Centrum pro konceptuální modelování a implementace“, fungující na půdě fakulty, které by takový nástroj přinesl větší možnosti a efektivitu v přípravě prezentačních materiálů. Krom toho bude však výsledek práce určený pro všechny, kterým při jejich výkladu a vysvětlení obsahu nestačí nebo nevyhovuje klasická lineární forma prezentace.

Téma jsem si zvolil zejména kvůli mému zájmu o multimediální a grafický design. Zmiňovaná interaktivní forma prezentace se mi zdá jako velice dobrý prostředek pro pochopení složitějších problematických celků. Zejména i pro absenci uceleného systému, který by tvorbu takových materiálů pohodlně a flexibilně umožňoval, jsem se rozhodl na tématu pracovat.

V rámci bakalářské práce se zabývám návrhem a implementací rozšíření programu Inkscape, které bude sloužit k efektivní tvorbě vektorových interaktivních prezentací.

Cíl práce

Cílem teoretické části práce je seznámit se s technologiemi a principy, které se dají využít při tvorbě interaktivních vektorových prezentací. V rámci literární rešerše je nutné zvolit vhodný značkovací jazyk použitelný pro tvorbu obsahové části dokumentu, vymyslet vhodnou metodiku pro vytvoření grafického základu prezentace včetně specifikace její struktury a propojení s obsahovými celky, a nastudovat a zanalyzovat možnosti transformace těchto instancí do výsledné vektorové interaktivní prezentace. Důležitým aspektem této části je také podrobný rozbor existujících řešení, která se danou problematikou zabývají, a analýzou jejich případného využití.

Praktická část se zabývá návrhem a implementací systému, který bude sloužit k efektivnímu vytváření vektorových prezentací. Cílem je, v závislosti na nabytých znalostech z teoretické části, navrhnout ucelený proces, využívající příslušné softwarové prvky, který připraví obsahovou i grafickou část prezentace a tyto vstupy transformuje do výsledné interaktivní formy.

K přesné definici požadavků na výslednou práci slouží následující seznam funkčních a nefunkčních požadavků, který byl vytvořen ve spolupráci s vedoucím práce.

1.1 Nefunkční požadavky

N1: Systém bude otevřený (Open Source¹).

1.2 Funkční požadavky

F1: Systém bude pracovat s obsahovým souborem, specifikujícím strukturu a obsah prezentace.

¹Software s legálně dostupnou licencí a otevřeným zdrojovým kódem.

1. CÍL PRÁCE

F2: Systém umožní vytvořit grafické podklady pro vektorovou prezentaci.

F3: Systém bude obsahovat nástroj pro propojení obsahu s grafickými podklady.

F4: Systém umožní specifikovat snímky a přechody prezentace.

F5: Systém umožní vyexportovat prezentaci, kterou bude možné procházet 2 standartními směry (dopředu, dozadu) včetně **zanořování**².

F6: Systém umožní vyexportovat prezentaci do interaktivní podoby, zobrazitelné ve webovém rozhraní.

F7: Systém bude obsahovat nástroj pro formátování a umístění definovaného obsahu do oblastí vymezených slidů.

²Funkce přechodu do vnitřních sekcí daného snímku.

Analýza

V této kapitole se zabývám analýzou řešeného problému. Nejprve se věnuji průzkumu existujících řešení a možností jejich využití pro můj projekt. Následně popisuji proces analýzy vlastního řešení z hlediska funkčních požadavků a použitých technologií.

2.1 Existující řešení

K řešení problému této bakalářské práce existuje již řada projektů, které však neřeší daný problém v plném rozsahu, nebo neplní některý z požadavků kladených na tuto práci. U každého řešení uvádím obecný popis projektu, jeho použitelnost a nedostatky s ohledem na požadavky mého tématu. Diskutuji také jeho možné využití v praktické části této bakalářské práce.

2.1.1 Sozi [1]

Sozi je jednoduchý program pro tvorbu vektorových prezentací, který může být použit buď jako samostatná aplikace, nebo jako rozšíření programu Inkscape. Na vstupu očekává soubor s připravenou vektorovou grafikou, ve které pomocí jeho uživatelského rozhraní lze specifikovat prezentaci. V případě samostatné aplikace se při každé změně automaticky ukládá aktuální soubor s výslednou prezentací do vybrané složky.

2.1.1.1 Použitelnost

- Jedná se o volný software, distribuovaný pod licencí **Mozilla Public License Version 2.0** ³.
- Instalace je dostupná pro všechny 3 OS (Windows, Linux, Mac OS).

³Mozilla Public License Version 2.0 [online]. mozilla.org contributors, 2017. Dostupné také z: <https://www.mozilla.org/en-US/MPL/2.0/>

2. ANALÝZA

- Možnost definice snímků prezentace (tzn. vymezení oblastí vstupního souboru, které se budou zobrazovat v rámci procházení prezentace).
- Možnost nastavit přechody mezi jednotlivými snímky prezentace.
- Automatické ukládání aktuálního stavu prezentace.
- Výsledná prezentace je ukládána jako **HTML**⁴ soubor.

2.1.1.2 Nevýhody

- V rámci práce s programem není možné vytvořit grafické podklady pro prezentaci. Před použitím Sozi je nutné dané podklady vytvořit pomocí jiného softwaru.
- Program se nezabývá oddělením obsahové a grafické části. Texty prezentace je nutné vytvořit společně s grafickými podklady.
- Snímky a přechody prezentace lze specifikovat pouze ve dvou směrech (vpřed, vzad). Program nepodporuje zanořování.

2.1.1.3 Shrnutí

Sozi svou jednoduchostí může posloužit jako vhodný prostředek pro rychlé vytvoření prezentace. Díky absenci možnosti vytvoření vlastních grafických podkladů a přílišné jednoduchosti UI na úkor některých komplexnějších nástrojů však není jako celek vhodným řešením pro problém, kterým se tato práce zabývá. Program může tedy posloužit pouze jako inspirace pro návrh vlastního řešení, zejména co se týče uživatelského rozhraní.

2.1.2 Prezi [2]

Prezi je dalším projektem, který se zabývá problematikou vektorových prezentací. Na rozdíl od předešlého zmiňovaného projektu se již jedná o daleko komplexnější a složitější nástroj. Program je dostupný ve formě webové i desktopové aplikace a nabízí velmi sofistikovaný systém přípravy a vytvoření vektorové prezentace.

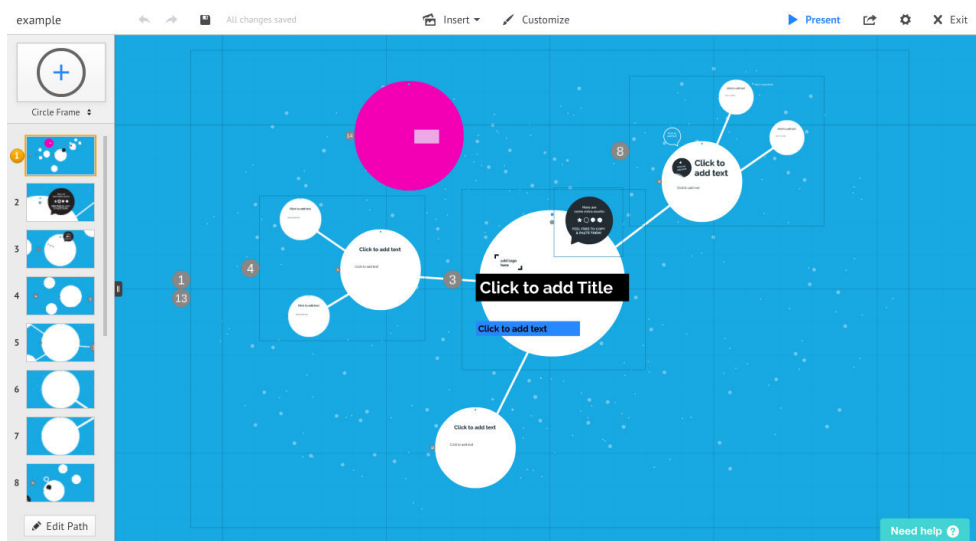
2.1.2.1 Použitelnost

- Možnost přípravy grafických podkladů pro prezentaci v prostředí programu.
- Díky webovému rozhraní je použití nezávislé na OS.

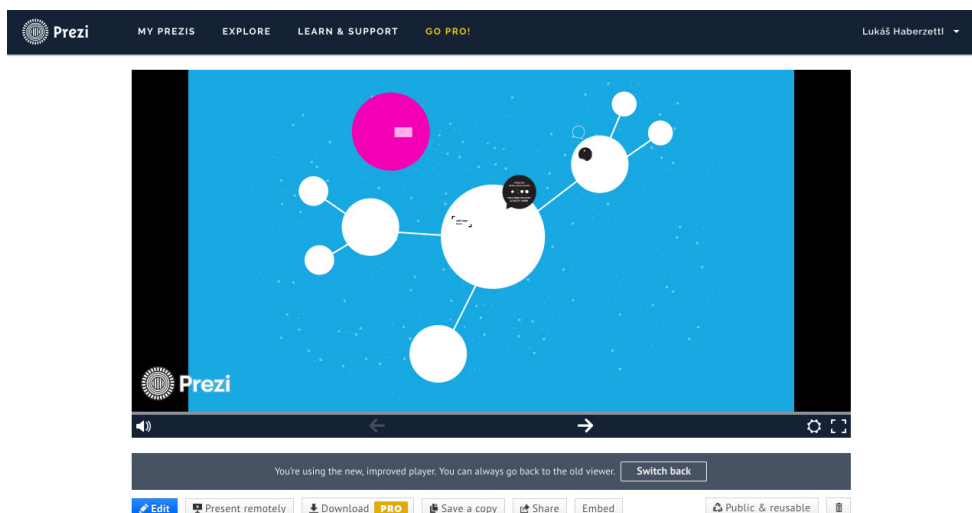
⁴značkovací jazyk, používaný pro tvorbu hypertextových dokumentů na webu

2.1. Existující řešení

- Výslednou prezentaci lze procházet všemi požadovanými směry včetně možnosti zanoření.
- Možnost výběru z velkého množství přednastavených šablon.



Obrázek 2.1: Prostředí pro vytvoření grafických podkladů v programu Prezi



Obrázek 2.2: Náhled výsledné flash prezentace v prostředí programu Prezi

2.1.2.2 Nevýhody

- Pokud chce uživatel software využívat v plném rozsahu, je nucen si vybrat některý z placených programů.
- V případě neplaceného plánu je k tvorbě prezentace potřeba internetové připojení.
- Uživatel nemá absolutní volnost při tvorbě grafických podkladů. Lze použít pouze přednastavené vektorové prvky.
- I přesto, že program umožňuje zanoření v rámci snímku prezentace, možnost pohybu v prezentaci je lineární (uživatel má na výběr pohyb pouze dopředu a dozadu. Zanoření musí být přesně specifikované v posloupnosti přechodů a nenechává uživateli možnost volby, zda se chce zanořit, nebo chce pokračovat na snímek ve stejné úrovni)
- Výsledkem je **flash animace**⁵, kterou lze prezentovat buď v rámci rozhraní Prezi, vložením na svých webových stránkách nebo ve formě staženého souboru na vlastním disku počítače. Hlavní nevýhodou je povaha technologie flash a díky tomu i nízká záruka, že bude prezentace fungovat dobře na všech zařízeních a ve všech webových prohlížečích.

2.1.2.3 Shrnutí

I přes velkou profesionalitu a propracovanost systému má Prezi nevýhody, které mohou uživatele limitovat. Především svou komerční politikou a uzavřeností systému se nejeví jako vhodné řešení problému mé bakalářské práce.

2.1.3 JessyInk [3]

Jedná se o komplexní rozšíření programu Inkscape. Je založené na principu vytváření vrstev dokumentu, definujících snímky prezentace. Díky podpoře zomomování⁶ a dalších efektů se vygenerovaná prezentace blíží výstupu předešlého probíraného řešení Prezi.

2.1.4 Použitelnost

- Jedná se o rozšíření programu Inkscape.
- Vzhledem k povaze řešení může uživatel k tvorbě grafických podkladů plně využívat prostředí programu Inkscape.

⁵Animace vytvořená pomocí Adobe Flash, nebo jiné podobné platformy, často distribuovaná ve formátu SWF.

⁶Přiblížení

- Podporuje zoomování vybraných oblastí, čímž se dá ve výsledném dokumentu docílit zanořování.
- Jedná se o Open Source program.

2.1.5 Nevýhody

- Neřeší formátování obsahu prezentace. Textové obsahy jednotlivých snímků je uživatel nucen vytvořit sám prostřednictvím programu Inkscape.
- Proces specifikace zanoření v prezentaci je příliš složitý.
- Uživatel má možnost, stejně jako u Prezi, procházet snímky pouze lineárně.
- Neposkytuje oddělení přípravy obsahové a grafické části prezentace.

2.1.6 Shrnutí

JessyInk nespĺňuje celou řadu požadavků kladených na tuto bakalářskou práci. Především díky nemožnosti oddělit jednotlivé části procesu přípravy prezentace a absenci čtyřsměrné navigace se nejeví jako vhodné řešení problému. Jakožto Inkscape plugin může však být velmi cennou inspirací při návrhu a implementaci vlastního řešení.

2.1.7 Porovnání řešení

Následující tabulka slouží k porovnání klíčových vlastností popisovaných existujících řešení.

Tabulka 2.1: Porovnání existujících řešení

Řešení	Otevřený software	Výsledkem HTML soubor	Podpora zanoření	4 směry ovládání
Sozi	ANO	ANO	NE	NE
Prezi	NE	NE	ANO	NE
JessyInk	ANO	NE	ANO	NE

2.1.8 Shrnutí

Průzkum existujících řešení ukázal, že ačkoliv již existují projekty, zabývající se danou tematikou, žádný z nich uspokojivě nespĺňuje požadavky kladené na cíl této bakalářské práce a jsou tedy jako její řešení nepoužitelné. Lze se jimi však při vlastním návrhu částečně inspirovat.

2.2 Funkční požadavky

2.2.1 F1

Tvorba obsahu prezentace musí probíhat nezávisle na procesu přípravy grafických podkladů. Obsah prezentace se dělí na dvě části. To jest:

Obsah snímků prezentace

Jedná se o veškerý text, který je umístěn uvnitř snímku.

Struktura prezentace

Specifikuje pořadí jednotlivých slidů včetně informace o míře zanoření. Může také obsahovat různé meta informace jako je název prezentace, datum vytvoření atd.

Pro popis výše zmíněných obsahových částí bude použit soubor s předdefinovanou strukturou, který bude načítán pluginem a využit pro výsledný export. Pro samotnou specifikaci struktury prezentace se jeví jako nejlepší možnost použít některý ze standartizovaných formátů pro definici dat. Pro obsah slidů se naopak dá lépe použít značkovací jazyk (viz 2.4.1.3). Pro vytvoření obsahu prezentace je tedy vhodné tyto dva formáty zkombinovat.

2.2.2 F2

Díky faktu, že řešení, kterým se v této bakalářské práci zabývám, má formu rozšíření pro program, který slouží k tvorbě **vektorové grafiky**⁷, je tento funkční požadavek splněn již samotnou integrací rozšíření do programu. Příprava grafické části prezentace tedy bude probíhat v rámci programu Inkscape, který je pro tento účel velice vhodným nástrojem.

2.2.3 F3

Jelikož se v rámci procesu přípravy předpokládá separace obsahové a grafické části, je nutné navrhnout způsob propojení těchto dvou částí, které zajistí veškeré vstupy nutné pro export prezentace. Toto propojení bude probíhat v rámci pluginu programu Inkscape. Plugin bude na vstupu očekávat externí soubor s obsahem prezentace a poskytne sadu nástrojů pro provázání obsahu a vytvořené grafiky. Pro úspěšný export prezentace je zapotřebí synchronizace následujících prvků.

⁷Jeden ze způsobů reprezentace obrazových informací v počítačové grafice. Narozdíl od rastrové grafiky je obrázek popsán matematicky a lze ho tak zmenšovat bez ztráty kvality

2.2.3.1 Synchronizované prvky prezentace

Identifikace snímku prezentace

Ke každému snímku, specifikovanému v obsahovém souboru, je zapotřebí přiřadit grafický element, který ho bude ve výsledné prezentaci zastupovat. Tímto přiřazením dojde k navázání veškerých obsahových atributů snímku na grafický element.

Pořadí snímku prezentace

U každého snímku musí být definováno jeho pořadí ve výsledné prezentaci.

Zanoření snímku prezentace

V případě zanoření snímku musí být jasně specifikován jeho rodič.

2.2.3.2 Způsob propojení

Abych mohl zvolit vhodný způsob propojení, musel jsem nejprve zanalyzovat princip reprezentace použité grafiky v programu a vymyslet vhodný způsob ukládání informací o propojení jednotlivých prvků. Hlavním otázkou zde bylo kam a jakým způsobem informace ukládat tak, aby byly použitelné pro export.

V rámci práce s programem Inkscape se pracuje s vektorovou grafikou. Ta je, na rozdíl od té rastrové, reprezentována matematicky popsanými křivkami. Je tedy libovolně škálovatelná bez ztráty kvality. Hlavním formátem, který se v programu Inkscape pro reprezentaci a ukládání vektorové grafiky používá, je **SVG**⁸. Tento formát ukládá data ve formě značkovacího jazyka **XML**⁹, díky kterému můžou být v jednotlivých vektorových objektech uložena dodatečná data ve formě atributů. Jako nejlepší možnost pro propojení se tedy jeví ukládání dodatečných informací o identifikaci, pořadí a zanoření snímku ve formě atributu přímo do xml elementu reprezentujícího daný grafický objekt. Tímto způsobem bude vždy jasně specifikováno, k jakému snímku informace náleží. Předpokládejme tedy, že již máme vytvořen externí soubor obsahující strukturu naší prezentace. Proces zapsání propojovacích a obsahových atributů do vytvořené grafiky může probíhat následujícími způsoby.

Manuálně

Jednou z možností je samozřejmě zapsat potřebné atributy do grafických objektů manuálně. Inkscape disponuje nástrojem pro úpravu XML kódu vytvářené grafiky. Uživatel má tak možnost si každý objekt upravit dle svých vlastních preferencí, přidat nebo odebrat atributy popř. změnit jejich hodnoty. Toto řešení nevyžaduje žádnou novou implementaci, avšak z hlediska použitel-

⁸Jeden z formátů pro popis vektorové grafiky, dnes ve velké míře používaný na internetu.

⁹Značkovací jazyk používaný zejména pro serializaci dat, nebo vytváření jiných značkovacích jazyků.

nosti pro koncového uživatele je velice neintuitivní a složité na ovládání.

V prostředí pluginu Inkscape

Další možností je vytvořit rozhraní pluginu Inkscape, které by uživatele abstrahovalo od přímého vkládání atributů a poskytlo mu uživatelsky přívětivější nástroj pro propojení. Uživatel by pomocí **UI**¹⁰ prvků pluginu označoval a propojoval objekty a vytvářel tím výslednou strukturu prezentace. Nevýhodou tohoto řešení je především velká omezenost uživatelského prostředí, které pro vytváření pluginů program Inkscape poskytuje. Vzhledem k nedostatku možností UI by bylo obtížné navrhnout jednoduchý uživatelský proces, kterým by se propojení docílilo. Nabízí se využití některé z **GUI**¹¹ knihoven jazyka pluginu, díky které by bylo možné použít i komplexnější prvky UI a docílit tím lepší uživatelské přístupnosti prostředí vytvářeného pluginu.

Ze zmíněných možností se vzhledem k provedené analýze jeví jako nejlepší způsob provádět propojování v prostředí pluginu, avšak bez použití GUI knihovny. Ukázalo se, že program nepodporuje využití takových knihoven, protože v rámci jejich spuštění není možná interakce s grafickým prostředím programu. Je tedy potřeba navrhnout co nejefektivnější prostředí pro navázání obsahu a grafiky s ohledem na dostupné UI prvky pluginu.

2.2.4 F4

V rámci práce s pluginem musí mít uživatel možnost definovat oblasti, které budou ve výsledné prezentaci chápány jako slidy. Při prohlížení výsledné HTML prezentace pak bude tato definovaná oblast, v momentu prohlížení slidu, pokrývat celou plochu náhledového okna prohlížeče. Dále musí být možné definovat posloupnost přechodů mezi takto určenými snímky. Proces propojení snímků prezentace s definicemi v obsahovém souboru včetně určení jejich pořadí je popsán v předchozí sekci (viz 2.2.3).

2.2.5 F5

Výsledný dokument s prezentací bude možné procházet následujícími směry.

- **Dopředu (na další snímek)**
Přechod na snímek, který ve struktuře prezentace bezprostředně následuje snímku aktuálnímu.
- **Dozadu (na předchozí snímek)**
Přechod na snímek, který ve struktuře prezentace bezprostředně předchází snímku aktuálnímu.

¹⁰Uživatelské prostředí

¹¹Označení pro grafické uživatelské prostředí. Zhlediska programu jde o vrstvu, která uživateli zobrazuje data

- **dolů**
Přechod na snímek, který je ve struktuře prezentace prvním synem aktuálního snímku (V pořadí první snímek ležící uvnitř aktuálního snímku)
- **Nahorů**
Přechod na snímek, který je ve struktuře prezentace rodičem aktuálního snímku.
V prezentaci se bude možné pohybovat všemi uvedenými směry, dovolí-li to struktura prezentace. Počet zanořených snímků v prezentaci není ničím omezen. Díky použití vektorové grafiky se tedy lze zanořovat (s ohledem na strukturu, kterou uživatel vytvoří) v libovolném rozsahu.

2.2.6 F6

Export prezentace lze provádět buď v rámci vytvářeného pluginu programu Inkscape, nebo pomocí externího skriptu. V obou případech musí být na vstupu programu, provádějícího export, 2 soubory, ze kterých se vygeneruje výsledný HTML soubor prezentace.

2.2.6.1 Vstupní soubory

- **Obsahový soubor**
Soubor obsahující specifikaci struktury prezentace včetně obsahů jednotlivých snímků.
- **Soubor s grafikou**
Soubor, nejspíše ve vektorovém formátu SVG, obsahující grafiku prezentace.

Vstupní soubory spolu musí být v momentě exportu kompatibilní. Tedy informace o propojení obsahu s grafickými elementy prezentace již musí být zaneseny v obsahu grafického souboru a musí odpovídat struktuře obsahového souboru. V opačném případě export nelze provést.

2.2.6.2 Výstupní soubory

Výstupem exportu bude jeden hlavní soubor, sloužící k zobrazení výsledné prezentace, včetně všech potřebných závislostí¹². V rámci exportu se také vygeneruje SVG soubor se současným stavem grafické části prezentace, spustitelný v programu Inkscape.

- **Soubor s prezentací**
Soubor obsahující výslednou prezentaci. Bude umožňovat interaktivní průchod prezentací a lze ho otevřít ve webovém rozhraní.

¹²Jako závislost se zde myslí jakýkoliv soubor, který je potřebný ke spuštění hlavního prezentačního souboru.

- **Knihovny**

Veškeré ostatní soubory potřebné pro správný chod hlavního souboru s prezentací. Tyto soubory zprostředkovávají manipulaci s SVG prezentací a vytvářejí prostředí pro interakci.

Pro zajištění všech požadavků kladených na formu prezentace budou použity technologie HTML, CSS¹³, Javascript a SVG (viz 2.4).

2.2.7 F7

Vzhledem k tomu, že je obsah jednotlivých snímků specifikován odděleně od grafické části prezentace, je potřeba vytvořit nástroj, který bude zprostředkovávat naformátování a umístění tohoto obsahového textu do oblasti snímku, který je s ním propojen. Předpokládá se, že obsah snímku bude specifikován v externím souboru pomocí některého ze známých značkovacích jazyků. Od nástroje se očekává konverze z tohoto značkovacího jazyka do naformátovaného textu čitelného pro běžného člověka a jeho umístění do oblasti snímku ve výsledné prezentaci. S ohledem na prostředí programu Inkscape a formát prezentace lze obsahový text konvertovat do dvou formátů.

- **SVG**

SVG formát nabízí definici textových objektů a jejich formátování za použití stylů. Největší výhodou zvolení tohoto formátu je především jeho podpora v programu Inkscape. Jelikož bude grafika prezentace vytvářena ve stejném formátu, byl by stav po konverzi stále zobrazitelný v programu. Uživatel by tak v průběhu práce díky konverznímu nástroji získal náhled na výsledný formát prezentace již s vloženými texty.

Velkou nevýhodou je fakt, že tento formát není primárně určen pro definici a stylování delších textových úseků. Nedisponuje tedy základními textovými elementy jako je nadpis, podnadpis, seznam a další. Nelze díky němu navíc vytvářet víceřádkové textové elementy. Každý řádek je chápán jako samostatný grafický element a nelze jej zarovnávat například podle parametrů rodičovského elementu, ve kterém je text umístěn. Díky tomu by bylo velice obtížné naformátovat a naškálovat vstupní text ve značkovacím jazyce tak, aby byl umístěn v oblasti vybraného snímku a nedošlo například k jeho přetečení do oblasti mimo něj.

- **HTML**

HTML je dalším formátem, který lze pro konverzi použít. Disponuje velkým množstvím textových elementů, podporuje zarovnání podle vymezené oblasti a lze pomocí něj tedy snadno naformátovat jakýkoliv text.

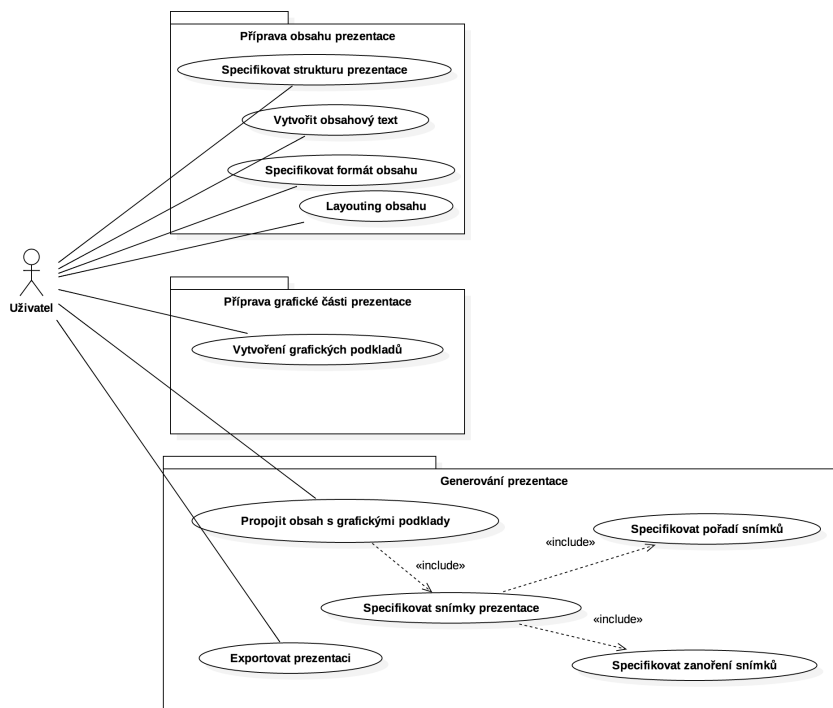
¹³Kaskádové styly

Jeho výhodou použitelnou pro mé řešení je také fakt, že jej lze snadno vložit do SVG formátu. Výsledkem konverze by tedy v tomto případě byl HTML objekt vložený do vytvořené grafiky a naformátovaný podle vstupního obsahového souboru. Jedinou nevýhodou použití tohoto formátu je absence jeho podpory v programu Inkscape. Zhlediska procesu přípravy prezentace by tedy uživatel získal náhled nad obsahovým textem až ve vyexportované prezentaci ve webovém prohlížeči.

S ohledem na značné nedostatky formátu SVG spojené s definicí textových elementů a jejich zarovnáním se jako daleko lepší řešení jeví použití pro konverzi formát HTML. I přes nemožnost zobrazit takto naformátovaný text v prostředí Inkscape je velice vhodným formátem pro reprezentaci obsahu ve výsledné prezentaci.

2.3 Analýza případů užití

S ohledem na specifikované funkční požadavky jsem provedl analýzu a modelování případů užití.



Obrázek 2.3: Diagram případů užití

Z diagramu je vidět rozdělení případů užití na tři oblasti. Na následujících řádcích budou popsány pouze případy užití související s první a poslední oblastí. To jest "Příprava obsahu prezentace" a "Generování prezentace". V oblasti "Příprava grafických podkladů prezentace" je využito již existujících nástrojů a analýza případů užití by zde byla v rámci této bakalářské práce redundantní.

2.3.1 Účastníci

Účastník případů užití je pouze jeden. Je jím uživatel, který systém používá. Uživatel připraví obsahový soubor prezentace, vytvoří grafickou část v programu Inkscape a pomocí pluginu tyto dvě části propojí a vyexportuje prezentaci do výsledného HTML souboru.

2.3.2 Případy užití

Popisy případů užití dle rozdělení do funkčních oblastí.

2.3.2.1 Příprava obsahu prezentace

Proces přípravy obsahu prezentace bude probíhat v prostředí libovolného textového editoru dle preferencí uživatele. V rámci mého řešení bude vytvořena gramatika pro popis struktury a obsahu prezentace, pomocí které budou aplikovatelné následující případy užití.

UC1 - Specifikovat strukturu prezentace

Umožní uživateli popsat a specifikovat strukturu prezentace. To zahrnuje především počet, pořadí a zanoření jednotlivých snímků.

UC2 - Vytvořit obsahový text

Umožní uživateli vytvořit obsahový text pro jednotlivé snímky prezentace.

UC3 - Specifikovat formát obsahu

Obsahový text v sobě ponese informaci o formátování, které bude použito při generování prezentace.

UC4 - Layouting obsahu

Umožní uživateli u každého snímku aplikovat dodatečné formátování, které bude zohledněno při generování prezentace.

2.3.2.2 Generování prezentace

UC5 - Specifikovat pořadí snímků prezentace

Umožní uživateli definovat pořadí označeného snímku v rámci výsledné prezentace.

UC6 - Specifikovat zanoření snímků prezentace

Umožní uživateli určit rodiče nebo potomky snímku a definovat tím zanoření označených snímků.

UC7 - Specifikovat snímky prezentace

Umožní uživateli označit oblasti ve vektorové grafice, které budou ve výsledné prezentaci figurovat jako snímky. Tyto oblasti jsou specifikované pomocí případů UC5 a UC6

UC8 - Propojit obsahový soubor s grafickými podklady

Pomocí specifikovaných informací z UC7 plugin propojí externí obsahový soubor s grafickými podklady a připraví vstupy pro export prezentace.

UC9 - Exportovat prezentaci

Plugin zpracuje veškeré vstupy získané pomocí UC8 a vygeneruje výslednou prezentaci do uživatelem zvoleného adresáře.

2.3.3 Scénáře případů užití

Na následujících řádcích se budu věnovat popisu scénářů dvou hlavních případů užití UC8 a UC9. U ostatních případů není potřeba popisovat jejich scénář díky malé míře složitosti a nebo díky tomu, že je jejich scénář součástí scénářů popisovaných případů.

UC8

1. Případ užití začíná v momentě, kdy má uživatel vytvořený obsahový soubor se strukturou prezentace a grafické podklady v prostředí programu Inkscape.
2. Program zobrazí uživateli nabídku pro určení pořadí a zanoření jednotlivých snímků.
3. Uživatel vybere možnost pro určení pořadí a pomocí uživatelského prostředí pluginu postupně označuje grafické elementy, reprezentující snímky prezentace, a určuje jejich pořadí.

2. ANALÝZA

4. V případě, že uživatel narazí na snímky u kterých chce specifikovat zanoření, zvolí nabídku pro zanoření a opět pomocí uživatelského prostředí pluginu specifikuje zanoření označených snímků.
5. Body 3. a 4. může uživatel libovolně střídat tak, aby docílil úspěšné specifikace celé struktury prezentace.
6. Snímky prezentace jsou v obsahovém souboru jednoznačně určeny svým pořadím a zanořením. Prostřednictvím bodů 3. a 4. je tedy uživatel schopen specifikovat a namapovat snímky z grafické části na tu obsahovou.
7. Jakmile má uživatel hotovu definici struktury prezentace, zadá prostřednictvím rozhraní pluginu lokaci externího obsahového souboru a spustí propojovací funkci, která namapuje veškeré atributy z obsahového souboru na grafické objekty příslušící daným snímkům a připraví tím veškeré vstupy potřebné pro export prezentace.

UC9

1. Příklad užití začíná po skončení případu UC4.
2. Program uživateli nabídne zadání lokace, kde bude vygenerována výsledná prezentace.
3. Uživatel zadá programu lokaci a spustí export.
4. Program zpracuje veškeré vstupy a vygeneruje adresář s prezentací ve zvolené lokaci.

2.4 Technologie

V souvislosti s analýzou funkčních požadavků řešení této bakalářské práce jsem provedl rešerši použitelných technologií.

2.4.1 Vytvoření obsahu

Pro vytvoření obsahové části prezentace jsem provedl průzkum následujících technologií.

2.4.1.1 YML

Jedná se o takzvaný **human-readable**¹⁴ formát používaný pro reprezentaci dat a jejich struktury. Jeho účel je podobný formátu XML, avšak oproti tomuto značkovacímu jazyku je jeho syntaxe¹⁵ značně minimalistická. Skvěle se hodí pro popis jakékoliv datové struktury a proto je i vhodným kandidátem pro realizaci definice obsahové části v mém řešení.

¹⁴Formát přirozeně čitelný člověkem

¹⁵Soubor pravidel, které definují strukturu dokumentu nebo fragmentu daného jazyka.

2.4.1.2 JSON

JSON je, stejně jako YAML, formát pro definici a výměnu dat. Je snadno čitelný člověkem a zároveň snadno zpracovatelný i generovatelný strojem.

2.4.1.3 Markdown [4]

Markdown je odlehčený značkovací jazyk, používaný pro formátování prostého textu. Původně byl navržen jako formát pro převod prostého textu do validního XHTML¹⁶ nebo HTML kódu, ale postupem času se stal oblíbeným nástrojem pro převod i do jiných formátů. Jeho síla je především v jednoduché syntaxi, díky které se dá snadno a rychle definovat sémantika výsledného textu. V mém projektu je využitelný pro definici obsahů jednotlivých snímků prezentace.

2.4.2 Inkscape plugin

Program Inkscape je **Open Source**¹⁷ projekt. Lze tedy legálně upravovat jeho zdrojový kód nebo psát pluginy, rozšiřující jeho funkce. Pro účely mého řešení jsem provedl analýzu problematiky vytváření takových pluginů pro program Inkscape a nastudoval technologie, které se v rámci tvorby používají.

2.4.2.1 Extension Definition File (INX)

INX je označení pro formát souboru, který se používá pro definici pluginu v prostředí programu Inkscape. Jak se uvádí na oficiálních stránkách programu v sekci věnované vývoji pluginů, *“All extensions are defined through Inkscape Extension Definition Files”* [5]. Soubor formátu INX je tedy hlavním prvkem, který je programem zpracován a na základě kterého se vytváří definovaný plugin. Lze pomocí něj definovat uživatelské prostředí pluginu a soubory se zdrojovým kódem, které tvoří logiku pluginu.

Listing 2.1: Ukázka formátu souboru INX

```
<?xml version="1.0" encoding="UTF-8"?>
<inkscape-extension xmlns="http://www.inkscape.org/
namespace/inkscape/extension">
  <_name>RadioButton example</_name>
  <id>org.inkscape.effect.radiobuttontest</id>
  <param name="radio1" type="optiongroup" _
gui-text="Select option: ">
  <option value="string1">translatable string 1</option>
  <option value="string2">string 2</option>
  <option value="string3">test 3!</option>
```

¹⁶značkovací jazyk, používaný pro tvorbu hypertextových dokumentů na webu

¹⁷Projekt se zdrojovým kódem dostupným pro veřejnost

```
</param>
<param name="radio2" type="optiongroup" _
gui-text="Select second option: ">
  <option value="string11">string1</option>
  <option value="string22">string2</option>
  <option>test3!</option>
</param>
<effect>
  <object-type>all</object-type>
  <effects-menu>
    <submenu _name="Developer Examples"/>
  </effects-menu>
</effect>
...
</inkscape-extension>
```

2.4.2.2 C++

C++ je programovací jazyk, ve kterém je program Inkscape implementován. Jednou z možností, jak vytvořit plugin, je přímá úprava zdrojového kódu. Taková úprava však vyžaduje velmi dobrou znalost jak daného programovacího jazyka, tak i zdrojových kódů programu, a její rozsah by přesahoval rámec této práce.

2.4.2.3 Python

Python je interpretovaný skriptovací jazyk používaný pro tvorbu Inkscape pluginů. Pluginy lze implementovat i v jiných skriptovacích jazycích, jako je například Perl, nicméně Python je momentálně zcela nejpoužívanější. Proto, aby Python skript správně fungoval v prostředí programu Inkscape, musí implementovat rozhraní definované speciálními knihovnamy vytvořenými právě pro tvorbu pluginů. Cesta k použitým Python skriptům musí být také definována v souboru INX.

2.4.2.4 Tkinter

Uživatelské prostředí definované souborem INX je velmi limitované. V případě potřeby složitějších uživatelských prvků se nabízí použít některou z GUI knihoven jazyka, ve kterém je plugin vytvářen. V případě jazyka Python je jednou z nejpoužívanějších GUI knihoven Tkinter. Ačkoliv by její využití v případě mého řešení mohlo být výrazným usnadněním ve tvorbě UI pluginu, s ohledem na nemožnost interakce s programem Inkscape v rámci jejího chodu, to nelze.

2.4.3 Export prezentace

Jedním z hlavních požadavků na výslednou prezentaci je její použitelnost ve webovém rozhraní. S ohledem na tento fakt zde popisují zejména webové technologie, které mají využití při tvorbě výsledného exportovaného souboru.

2.4.3.1 HTML

Jde o značkovací jazyk, používaný pro tvorbu webových stránek. Proto, aby byla prezentace čitelná a použitelná ve webovém rozhraní, je tedy nutné aby výsledný soubor byl ve formátu HTML.

2.4.3.2 CSS

CSS, jinak označované jako kaskádové styly, je jazyk, který popisuje způsob a styl zobrazení jednotlivých částí webové stránky, definované v jazyce HTML, XHTML, nebo XML.

2.4.3.3 Javascript

Při prohlížení prezentace se od uživatele očekává interakce. Javascript je multiplatformní skriptovací jazyk, který takovou interakci zprostředkovává. Lze ho vložit do HTML souboru a pomocí něj definovat akce, které může uživatel uskutečnit (interakcí s webovou stránkou/prezentací), a **callbacky**¹⁸, které se po provedení takové akce mají zavolat.

2.4.3.4 TweenMax.js [9]

V rámci výsledné prezentace bude potřeba vytvořit nástroj pro animaci přechodů mezi jednotlivými snímky. K takovému účelu se hodí právě knihovna TweenMax.js. Jedná se javascript knihovnu obsahující funkce pro animaci DOM¹⁹ elementů.

2.4.3.5 SVG

SVG je formát vektorové grafiky, dnes ve velké míře používaný právě na webových stránkách. Jde o specifikaci XML, pomocí které se dají popisovat grafické tvary a elementy. Hlavní výhodou tohoto formátu je právě jeho použitelnost na webu. Díky jeho struktuře a faktu, že se jedná o XML kód, lze formát (například pomocí Javascriptu) programově upravovat přímo v prohlížeči. Lze do něj vkládat HTML elementy nebo upravovat jeho vlastnosti.

¹⁸funkce, které se vykonají v předem definovaný moment (po stisknutí tlačítka, načtení stránky atd.)

¹⁹Strom objektů, vytvořen internetovým prohlížečem v momentě načtení stránky

2.4.3.6 Node.js [6]

Node.js je vysoce výkonné **event-driven**²⁰ prostředí pro jazyk Javascript. Skládá se z javascriptového interpretu V8, nad kterým je tenká vrstva C++ kódu. Pro řešení bakalářské práce může posloužit jako vrstva pro export prezentace, nebo také jako hlídací skript, který bude detekovat změny vstupních souborů a v závislosti na tom přegenerovávat prezentaci.

2.5 Shrnutí

V rámci analýzy byl proveden průzkum všech aspektů, důležitých pro zorientování se v tématu a upřesnění požadavků na řešení. Znalosti nabyté v rámci jednotlivých bodů této analýzy poskytnou odborný náhled na řešenou problematiku a poslouží jako základ pro návrh výsledného systému.

²⁰Událostmi řízené. Jedná se o programovací paradigma, ve kterém je běh programu tvořen událostmi, jako je uživatelská akce, zpráva od jiného programu nebo výstup nějakého senzoru, detekujícího určitou změnu.

Návrh

3.1 Volba technologií

V závislosti na provedené analýze jsem vybral technologie, které jsou nejlépe použitelné pro můj projekt. Po zmapování všech možností řešení jsem nakonec od některých technologií probíraných v analytické části této práce ustoupil. Následuje souhrn navržených technologií uskupených podle funkčních oblastí.

3.1.1 Obsahový soubor

- **YML** - Jazyk pro reprezentaci dat. V případě mého řešení bude určen pro reprezentaci struktury vytvářené prezentace.
- **Markdown** - Značkovací jazyk, který bude použit pro definici formátovaného obsahu jednotlivých snímků.

3.1.2 Inkscape plugin

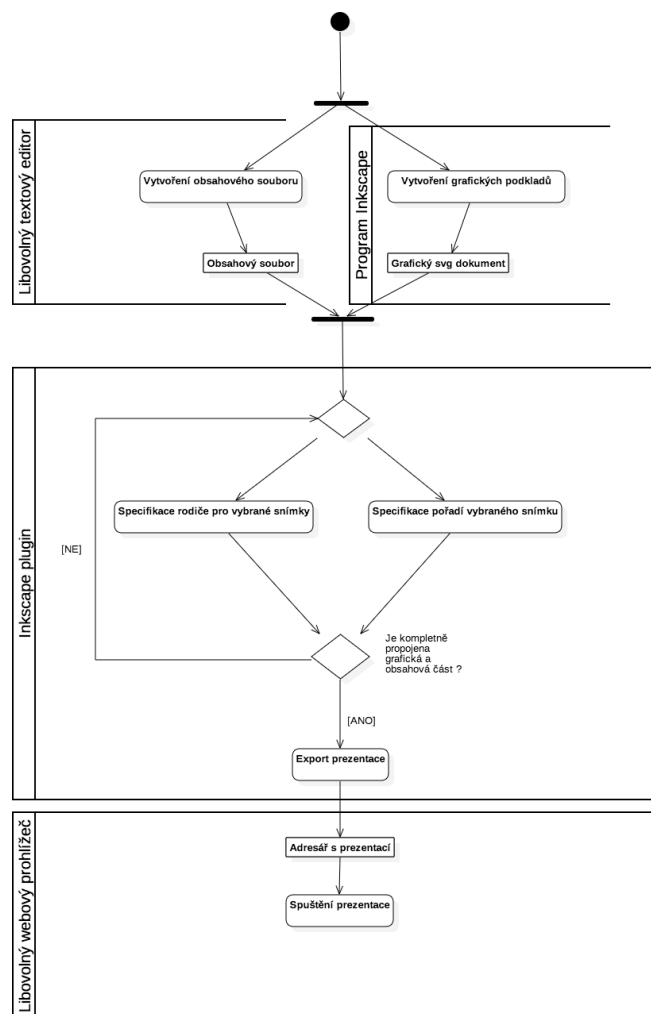
- **Python** - Jazyk, ve kterém bude naprogramováno jádro pluginu.
- **INX** - Formát použitý pro definiční soubor Inkscape pluginu.

3.1.3 Výsledná prezentace

- **HTML** - Formát výsledného souboru s prezentací, zobrazitelný ve webovém rozhraní.
- **CSS** - Formát pro popis stylů, použitých v rozhraní výsledné prezentace.
- **Javascript** - Jazyk zprostředkující veškerou obsluhu a interakci vygenerované prezentace.

3.2 Proces přípravy prezentace

Pro získání detailní představy nad životním cyklem vytvářeného řešení a jeho použití byl vytvořen diagram aktivit znázorňující veškeré aktivity a jejich posloupnosti, které se v rámci procesu vykonávají.



Obrázek 3.1: Diagram aktivit znázorňující proces přípravy prezentace

Ze schématu je patrné jakým procesem musí uživatel projít aby vytvořil výslednou prezentaci. K tomuto účelu je nutné navrhnout několik komponent. Je potřeba vytvořit syntaktickou specifikaci obsahového souboru, navrhnout uživatelské prostředí Inkscape pluginu včetně logiky pro propojení obsahové a grafické části, komponentu pro export prezentace a také strukturu výsledného souboru prezentace a logiku zprostředkující její interakci.

3.3 Návrh obsahového souboru

Obsahový soubor bude sloužit pro specifikaci struktury a obsahu prezentace. Pro tento účel jsem navrhl jednoduchý formát zápisu, který plně poslouží k zaznamenání všech potřebných informací obsahové části. Následující řádky popisují problematiku navrhovaného formátu včetně všech syntaktických pravidel, která musí být při vytváření souboru dodržena. Soubor se bude skládat ze dvou částí. Sekundární část, neboli hlavička slouží k zapsání meta informací, kdežto hlavní část obsahuje informace o jednotlivých snímcích včetně jejich obsahových informací, které budou později využity pluginem pro propojení a export.

3.3.1 Sekundární část / hlavička

Tato sekce je v souboru reprezentována klíčovým slovem **header** a slouží, jak už bylo řečeno, pro definici meta informací. Jedná se o údaje, které nebudou přímo viditelné v oblasti výsledné prezentace ale mají určitý sémantický význam. V této sekci lze definovat následující atributy.

- **name** - Název prezentace.
- **author** - Autor prezentace.
- **date** - Datum prezentace.

Informace v této sekci budou zpracovány komponentou na straně Inkscape pluginu a budou zobrazeny jako titulek²¹ výsledné prezentace.

3.3.2 Hlavní sekce

Tato sekce je v souboru reprezentována klíčovým slovem **presentation** a obsahuje definici struktury a obsahu prezentace. Sestává z dvou hlavních prvků reprezentovaných klíčovým slovem **node** a **topic**, které jsou z hlediska YML syntaxe definovány prostřednictvím listu.

- **node** - Snímek prezentace.
- **topic** - Skupina snímků. Může obsahovat libovolné množství prvků typu **node** nebo **topic**. Z této specifikace vyplývá, že může skupina obsahovat libovolný počet vrstev skládajících se z prvků typu **topic** s prvkem typu **node** na konci obsahové větve. Zvláštním případem snímku typu **topic** je **topic** neobsahující žádné další snímky.

Každý z těchto prvků obsahuje atributy ve formě definičního listu. Prvek může obsahovat následující atributy.

²¹Element `<title>` v hlavičce výsledné webové stránky s prezentací.

3. NÁVRH

- **style** - Nepovinný atribut. Lze pomocí něj definovat dodatečné styly aplikovatelné ve vyexportované prezentaci. Obsahuje definiční list, neboli slovník, atributů a hodnot. Styly obsažené v tomto definičním listu musí být validní z hlediska specifikace jazyka CSS.
- **content** - Nepovinný atribut. Definuje obsah daného prvku. U prvku typu **node** jde o textový obsah definován pomocí syntaxe značkovacího jazyka Markdown. V případě prvku **topic** může atribut obsahovat jak textový obsah, tak další snímky nebo skupiny snímků prezentace.

V případě atributu **content** prvku **topic**, u kterého jsou požadovány oba druhy obsahů (tedy textový obsah a list zanořených snímků) musí být specifikovány následující atributy.

- **node_content** - Atribut specifikující textový obsah prvku pomocí syntaxe značkovacího jazyka Markdown, stejně jak je tomu u prvku **node**.
- **topic_content** - Tento atribut obsahuje libovolný počet dalších snímků (**node**) nebo skupin snímků (**topic**) prezentace.

3.3.3 Syntaktická pravidla

- Každý atribut v definičním listu, specifikovaném v rámci prvku **topic** a **node** musí obsahovat vždy svou hodnotu. V opačném případě skončí export prezentace chybovým hlášením a ukončí se.
- Specifikace stylů obsažených v atributu **style** je rozšířena o atribut **size-ratio** s hodnotou reálného čísla. Toto číslo slouží k specifikaci poměru, v jakém se má obsah snímku v exportované prezentaci naškálovat oproti původní vypočtené velikosti. Výpočet poměrů a velikostí naformátovaného obsahu a jeho zasazení do oblasti snímku je velmi obtížný proces a může se stát, že výsledná forma exportu nemusí uživateli vždy vyhovovat. Atribut byl přidán především proto, aby si mohl uživatel v takovém případě výsledný export upravit podle svých potřeb.

3.3.4 Gramatika

Pro přesnou definici formátu obsahového souboru jsem vytvořil gramatiku pomocí **Rozvinuté Backus-Naurovy formy**²². Gramatiku popisuje následující tabulka. Atributy **nodeContent** a **topicContent** zde ve skutečnosti představují jeden a ten samý atribut **content**, který v závislosti na svém rodiči může nabývat odlišného obsahu.

²²Notace využívaná pro zápis bezkontextových gramatik

Tabulka 3.1: Definice obsahové gramatiky pomocí EBNF

atribut		obsah
document	::=	header , presentation
header	::=	[name], [author], [date]
presentation	::=	{ node topic }
node	::=	[style], [nodeContent]
nodeContent	::=	? string written in Markdown syntax ?
topic	::=	[style], [topicContent]
topicContent	::=	[nodeContent], [{ node topic }]
style	::=	? list of CSS styles ?

3.3.5 Ukázka formátu

Pro lepší představu o navrhovaném formátu obsahového souboru slouží následující ukázka kódu.

Listing 3.1: Ukázka formátu souboru INX

```
#Hlavicka
header:
  name: presentation name #navez prezentace
  author: Lukas Haberzettl #autor prezentace
  date: 01.05.2017 #datum vytvoreni prezentace

#Hlavni sekce
presentation:
  - node: #prvek typu node
    style: #definice stylu
      color: red
      margin: 0 20 0 30
    content: Obsah snimku v markdown syntaxi

  - topic: #prvek typu topic
    content: #obsah topicu
      - node: #node obsazen v rodicovskem topicu
        content: Obsah snimku v markdown syntaxi
```

3.3.6 Shrnutí

Navržený formát slouží jako jednoznačný popis struktury a obsahu prezentace a je jedním ze vstupů nutných pro její úspěšné vygenerování.

3.4 Plugin programu Inkscape

Zřejmě nejobsáhlejší komponentou celého řešení je plugin programu Inkscape. Předpokládá se, že je, z hlediska procesu přípravy prezentace, použit až ve chvíli, kdy jsou uživatelem vytvořeny obsahové a grafické podklady. Plugin zprostředkuje uživateli sadu nástrojů pro propojení daných vstupních částí a bude obsahovat funkci pro export prezentace do vybrané lokality.

Tato sekce se zabývá návrhem pluginu jak z pohledu uživatelského rozhraní, tak i po algoritmické stránce.

3.4.1 Návrh uživatelského rozhraní

Jak již bylo řečeno v rámci analýzy, program Inkscape poskytuje pro vytvoření uživatelského rozhraní pluginu jen omezené množství prvků. Bohužel využití některé z GUI knihoven jazyka Python také nepřipadá v úvahu díky nemožnosti interakce s programem v průběhu jejího běhu. Návrh prostředí pluginu tedy bylo nutné přizpůsobit této skutečnosti a i když díky tomu výsledné rozhraní možná není ideální, s ohledem na zvolené technologie se jedná zřejmě o nejefektivnější řešení.

Následující tabulka popisuje veškeré vstupy, které lze při návrhu uživatelského prostředí pluginu využít.

Tabulka 3.2: Dostupné UI prvky dle specifikace INX [7]

typ	popis
boolean	zaškrtačací pole pro nastavení pravdivostní hodnoty
color	nástroj pro vybrání hodnoty barvy
description	popisný textový prvek. Nezprostředkovává žádný výstup
enum	prvek pro výběr hodnoty z předdefinovaných možností
int	textový vstup pro zadání číselné hodnoty
notebook	skupina stránek, mezi kterými je možné se pohybovat pomocí navigace v záhlaví
optiongroup	podobně jako u typu enum vytvoří skupinu radio tlačítek s předdefinovanými hodnotami.
string	textový vstup pro zadání číselné hodnoty

Ve spolupráci s vedoucím mé práce bylo uživatelské rozhraní rozděleno do tří hlavních sekcí, ve kterých probíhá práce s pluginem. Tyto tři sekce včetně

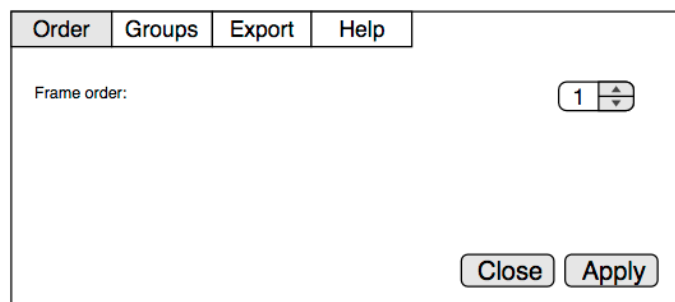
sekce s nápovědou budou seskupeny pomocí prvku **notebook** (viz tabulka 3.2).

3.4.1.1 Order

Sekce pro specifikaci pořadí snímků prezentace. V rámci této sekce bude uživatel přiřazovat pořadí jednotlivým snímkům pomocí UI prvku **int**.

Scénář užití

1. Uživatel označí grafický objekt v programu Inkscape, kterému chce přiřadit pořadí.
2. Uživatel zvolí pořadí které chce objektu přiřadit pomocí textového vstupu.
3. Tlačítkem **apply** se provede funkce přiřazení pořadí vybranému objektu.
4. V případě že se uživatel snaží přiřadit pořadí, které již v dané vsrtvě zanoření existuje, zobrazí se uživateli informující chybová hláška.



Obrázek 3.2: Návrh sekce **Order**

3.4.1.2 Groups

Tato sekce umožní uživateli určovat skupiny snímků a jejich rodiče²³ a tím specifikovat jejich zanoření v rámci prezentace.

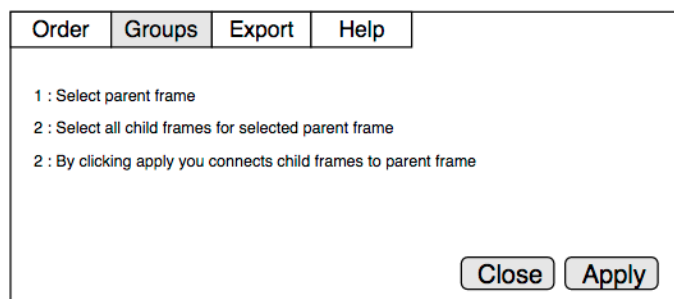
Scénář užití

1. Uživatel označí grafický objekt v programu Inkscape u kterého chce specifikovat jemu podřazené snímky.
2. K již označenému objektu označí uživatel skupinu dalších objektů, u kterých chce specifikovat jejich zanoření oproti prvnímu objektu.

²³jim nadřazené snímky

3. NÁVRH

3. Tlačítkem **apply** se provede funkce zanoření vybraných objektu.



The image shows a dialog box with a title bar containing four buttons: "Order", "Groups", "Export", and "Help". The "Groups" button is highlighted. The main area of the dialog contains three lines of text: "1 : Select parent frame", "2 : Select all child frames for selected parent frame", and "2 : By clicking apply you connects child frames to parent frame". At the bottom right, there are two buttons: "Close" and "Apply".

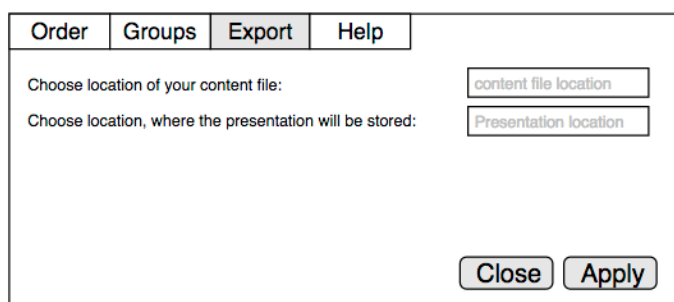
Obrázek 3.3: Návrh sekce **Group**

3.4.1.3 Export

Sekce slouží k exportování prezentace do zvoleného systémového adresáře. Očekává dva vstupy od uživatele. Jedním je lokace obsahového souboru a druhým již zmiňovaný cílový adresář, do kterého se výsledná prezentace uloží.

Scénář užití

1. Uživatel zadá lokaci obsahového souboru a cílového adresáře.
2. Tlačítkem **apply** se provede export prezentace do zvoleného adresáře.
3. V případě že se při exportu vyskytla chyba, zobrazí se uživateli informující chybová hláška.

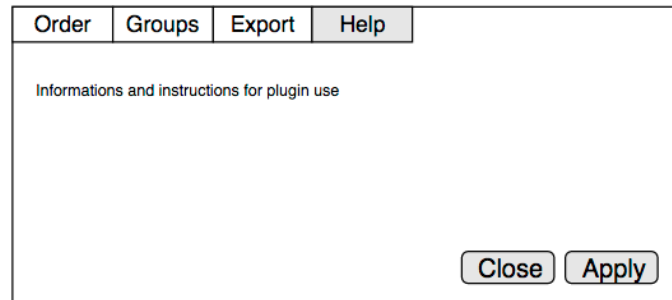


The image shows a dialog box with a title bar containing four buttons: "Order", "Groups", "Export", and "Help". The "Export" button is highlighted. The main area of the dialog contains two labels and two text input fields: "Choose location of your content file:" followed by a text box containing "content file location", and "Choose location, where the presentation will be stored:" followed by a text box containing "Presentation location". At the bottom right, there are two buttons: "Close" and "Apply".

Obrázek 3.4: Návrh sekce **Export**

3.4.1.4 Help

Plugin bude obsahovat také informující sekci, která bude sloužit k podrobnějšímu popisu a návodu k použití uživatelského prostředí.



Obrázek 3.5: Návrh sekce s nápovědou **Help**

3.4.2 Aplikační návrh

Vytvářený plugin je, dle specifikace Inkscape [5], typu **input**, **effect** a **output**. Čte data ze vstupního souboru, upravuje obsah Inkscape dokumentu a výsledná data zapisuje na disk. Návrh pluginu sestává z logiky pro propojení obsahové a grafické části a logiky transformace a exportu vstupů do výsledné prezentace.

3.4.2.1 Čtení a úprava SVG dokumentu

Hned na úvod je potřeba popsat, jakým způsobem bude plugin získávat a upravovat informace z vytvořeného grafického SVG dokumentu. K tomuto účelu slouží v prostředí Inkscape knihovna **inkex** [8], která obsahuje funkce pro vytváření, pohyb a úpravu v hierarchické struktuře dokumentu. Krom toho také poskytuje informace o objektech označených uživatelem v rámci programu.

3.4.2.2 Princip přípravy propojení obsahové a grafické části

Pro správné propojení těchto dvou částí je prostřednictvím pluginu potřeba propsat do grafických SVG objektů²⁴ informace o pořadí a zanoření snímků. Výsledkem procesu propojení bude grafický dokument, obsahující tyto informace, díky kterým bude v rámci exportu snadno propojitelný s atributy obsahového souboru (bude popsáno v další sekci). Jak již bylo popsáno v analýze, propojující informace budou do SVG dokumentu propisovány ve formě XML atributů. K přesnému popisu pořadí a zanoření snímků byly navrženy následující atributy.

²⁴ které jsou obsaženy v grafické části prezentace

3. NÁVRH

Tabulka 3.3: Atributy sloužící k propojení obsahu s grafickým dokumentem

atribut	hodnota	popis
farslide-order	celé číslo	Atribut popisující pořadí snímku v rámci aktuální úrovně jeho zanoření. Každý grafický objekt mající tento atribut se automaticky považuje za snímek prezentace.
farslide-parent	hodnota identifikátoru rodiče	Atribut přiřazující k danému snímku snímek jemu nadřazený.
class	farslide-is-parent	Každý snímek s tímto atributem a hodnotou se chápe jako topic a očekává se tedy, že obsahuje další snímky prezentace.

Vkládání atributů do grafických objektů budou obstarávat dvě komponenty pluginu rozdělené podle typu informace, se kterou pracují.

Order resolver

Komponenta pracující s atributem **farslide-order**. Jejím vstupem je grafický objekt a celé číslo reprezentující jeho pořadí. Komponenta zkontroluje zda se v dané úrovni zanoření nevyskytuje objekt se stejným pořadím a pokud ne, vloží informaci ve formě atributu do vstupního objektu. Následuje pseudo kód znázorňující logiku komponenty²⁵.

²⁵Pseudo kód je psán kompletně v anglickém jazyce. Vzhledem k tomu že klíčová slova pseudokódu jsou v angličtině, použití českého jazyka by zde bylo matoucí.

Algoritmus 1: Order resolver

Data: (selected svg object) **selected**, (order number) **order**
Result: Updated svg document containing new ordering information
objectsToControl = all svg objects with farslide-order attribute located
in same layer as selected object;
for *each object from objectsToControl* **do**
 if *object.farslide-order == order* **then**
 print error message;
 return;
 end
end
add farslide-order attribute to selected object;

Groups resolver

Komponenta pro specifikaci zanoření objektů. Jejím vstupem je skupina označených grafických objektů²⁶. Prvnímu objektu komponenta přiřadí atribut **class** s příslušnou hodnotou (viz tabulka 3.4). Další označené objekty jsou chápány jako potomci zanořené v rámci rodiče (prvního označeného objektu) a je jim přiřazen atribut **farslide-parent** s hodnotou identifikátoru rodiče. Pseudo kód zde, pro jednoduchost komponenty, neuvádím.

3.4.2.3 Princip exportu prezentace

Před spuštěním exportu prezentace se předpokládá, že jsou již obsahové a grafické podklady plně připravené pro propojení. Export zahrnuje dva logické celky.

1. **Příprava grafiky pro export** - Synchronizace obsahové a grafické části, na jejímž základě jsou do grafického SVG dokumentu propsány atributy z obsahového souboru (viz 3.3). Takto upravený dokument je použit pro generování prezentace.
2. **Vygenerování adresáře s prezentací** - Vygenerování výsledného HTML souboru s prezentací do cílového adresáře spolu se všemi podpůrnými soubory (bude popsáno v další sekci).

Veškerou logiku exportu bude zprostředkovávat jedna hlavní komponenta.

Export resolver

Tato komponenta má na vstupu obsahový soubor, grafický dokument a lokaci adresáře, ve kterém má být uložena výsledná prezentace. V rámci návrhu komponenty byl vytvořen pseudo kód popisující synchronizační funkci, jejímž

²⁶Minimální počet vstupních objektů je 2.

3. NÁVRH

výstupem je grafický svg dokument obsahující namapované atributy z obsahového souboru.

Algoritmus 2: Synchronizační funkce Export resolveru

Data: (One layer of objects from content file) **layer**, (Id attribute of svg parent) **parentId** = null

Result: Updated SVG document containing additional informations from content file

svgNodes = all SVG objects with farslide-parent attribute containing given parentId value ;

if *size(layer) != size(svgNodes)* **then**

 | print error("Content file does not match svg content.");
 | return;

end

// sort svgNodes by their order attribute (farslide-order)

sort(svgNodes);

index = 0;

for *key, frame in layer* **do**

 | **if** *frame has content attribute* **then**

 | **if** *frame is of type node* **then**

 | add content attribute with value = frame.content to
 | svgNodes[index] object;

 | **end**

 | **else if** *frame is of type topic* **then**

 | // Recursive call
 | synchronization(frame.content, svgNodes[index].getId());

 | **end**

 | **end**

 | **if** *frame has style attribute* **then**

 | add style attribute to svgNodes[index] object;

 | **end**

 | index++;

end

Uvedená funkce pro synchronizaci vstupních dat je založena na rekurzivním volání sebe sama. Export resolver tuto funkci zavolá poprvé s první vrstvou objektů a nulovým id atributem rodiče na vstupu. Tím je zajištěno, že se pomocí rekurzivity propojí všechny vrstvy obsahového souboru s grafickými objekty. V případě nekonzistence obsahových a grafických dat je vysána chybová hláška a funkce je přerušena.

Jakmile je úspěšně provedena synchronizace vstupních dat, resolver vygeneruje prezentaci do zadaného cílového adresáře. Export je proveden zapsáním synchronizovaného grafického dokumentu do připravené HTML šablony a jeho

překopírování do cílového adresáře včetně všech dalších souborů potřebných pro běh prezentace.

Obsah cílového adresáře

Struktura vyexportovaného adresáře je následující.

```

| index.html ..... spustitelný soubor s prezentací
| presentation.svg ..... kopie svg dokumentu použitého při generování
| prezentace
| img
| | * ..... soubory s grafikou uživatelského rozhraní
| js
| | * ..... javascriptové soubory (závislosti prezentačního souboru)

```

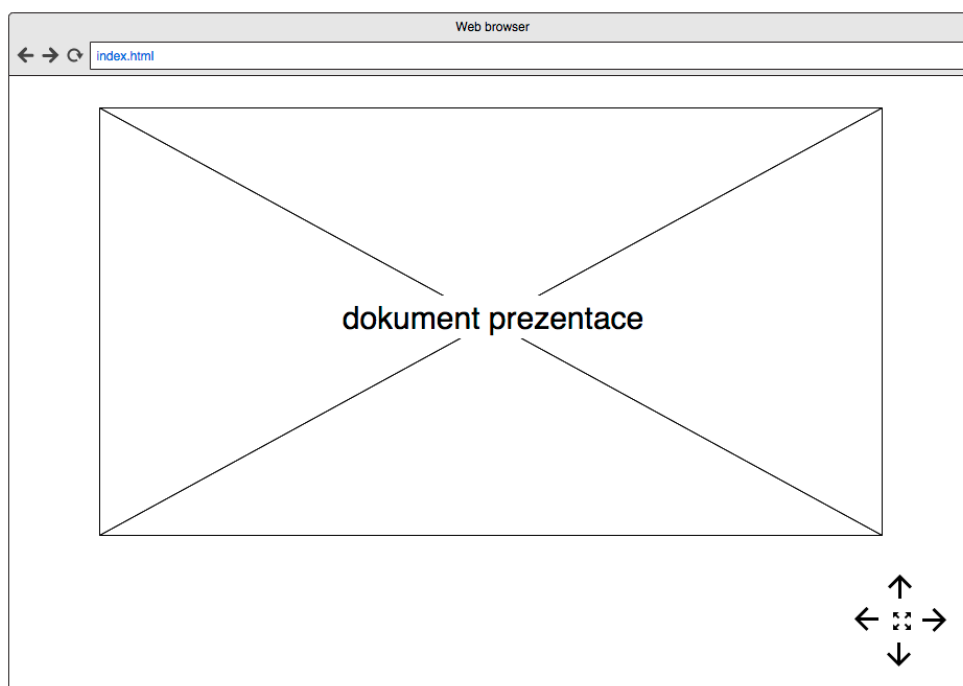
3.5 Prezentace

V rámci této kapitoly byl proveden návrh uživatelského rozhraní a aplikační návrh výsledné prezentační komponenty.

3.5.1 Uživatelské rozhraní

Uživatelské rozhraní výsledné prezentace bude velmi jednoduché. Bude se skládat z náhledu SVG dokumentu prezentace, ve kterém bude možné se pohybovat pomocí navigačních tlačítek umístěných v pravém dolním rohu. Navigace umožní pohyb na další / předchozí snímek v rámci aktuální vrstvy a nebo do nadřazené / podřazené vrstvy (šipky nahoru / dolů). Navigační tlačítko se uživateli zobrazí pouze ve chvíli, kdy je daný pohyb možný. Uživatel tak má vždy informaci o tom, jakými všemi směry se může v prezentaci pohybovat. Navigace bude také obsahovat speciální tlačítko pro oddálení pohledu na celý svg dokument.

3. NÁVRH



Obrázek 3.6: Návrh uživatelského rozhraní prezentační komponenty

3.5.2 Aplikační návrh

Veškerá inicializace prezentace a interakční logika bude zprostředkována javascriptovou komponentou. Tato komponenta bude rozdělena do funkčních bloků podle typu úlohy, kterou vykonávají.

3.5.2.1 Inicializace snímků prezentace

Při spuštění prezentačního souboru komponenta projde SVG dokument prezentace a vytvoří datovou strukturu obsahující všechny snímky dokumentu reprezentované speciálními objekty. Tato struktura bude sloužit k orientaci v prezentaci v rámci navigace. Objekt snímku bude mít následující strukturu.

Tabulka 3.4: Objekt reprezentující jeden snímek v prezentaci

atribut	hodnota	popis
id	string	jednoznačný identifikátor snímku
order	int	pořadí snímku v rámci vrstvy, ve které se nachází
left	string	identifikátor snímku, který následuje po tomto snímku
right	string	identifikátor snímku, který předchází tomuto snímku
childs	array	pole obsahující snímky, které jsou v prezentaci podřazené tomuto snímku
parent	string	identifikátor snímku, který je v prezentaci nadřazen tomuto snímku

3.5.2.2 Aktivace navigační komponenty

Pro možnost pohybu v prezentaci je potřeba aktivace navigační komponenty. Tento proces zahrnuje zaregistrování callback funkcí pro všechna navigační tlačítka a inicializaci globální proměnné, která ponese informaci o aktuální poloze v prezentaci. Tato proměnná bude obsahovat identifikátor objektu snímku z datové struktury snímků. Při stisknutí navigačního tlačítka se vykoná příslušná funkce, která upraví tuto globální proměnnou a spustí animaci přechodu na cílový snímek.

K naprogramování animace přechodu se perfektně hodí javascript knihovna **tweenMax.js** (viz 2.4.3.4), která poskytuje funkce pro jednoduché procházení DOM elementů ve formě animací.

3.5.2.3 HTML obsah

Poslední důležitou úlohou v rámci inicializace prezentace je vygenerování obsahových HTML bloků do oblastí jednotlivých snímků. V rámci této úlohy bude zkontrolován každý snímek a v případě že obsahuje atribut z obsahového souboru ²⁷, bude hodnota daného atributu přetřansformována do podoby HTML

²⁷příklad: content='Markdown content'

elementů a vložena do oblasti snímku v svg dokumentu.

Transformace obsahového atributu

Předpokládá se, že obsah atributu je popsán v syntaxi jazyka Markdown. K jeho transformaci do jazyka HTML poslouží jedna z dostupných javascriptových knihoven, zabývajících se tímto problémem.

Vložení HTML obsahu do oblasti snímku

K vložení HTML kódu do SVG dokumentu slouží element `foreignObject`. Tento element umožňuje do oblasti SVG dokumentu vložit elementy z jiného jmenného prostoru (V mém případě HTML). V rámci každého snímku bude tedy nejprve na stránce dynamicky vytvořen tento element, do kterého se posléze vloží transformovaný HTML obsah. Díky tomuto faktu se bude vložený obsah v průběhu animací škálovat společně s celým SVG dokumentem.

Vkládaný element bude mít následující strukturu.

Listing 3.2: Struktura vloženého elementu s HTML obsahem

```
<foreignObject>
  <body xmlns='http://www.w3.org/1999/xhtml' style='... '>
    <div style='... '>
      Transformed HTML content...
    </div>
  </body>
</foreignObject>
```

Pro správné pozicování obsahu je nutné nastavit příslušné atributy elementu `foreignObject` tak, aby byl prohlížečem zobrazen na správném místě. Atributy dostupné pro nastavení pozice a velikosti elementu jsou následující.

- **x** - Horizontální souřadnice v rámci SVG dokumentu.
- **y** - Vertikální souřadnice v rámci SVG dokumentu.
- **width** - Šířka elementu.
- **height** - Výška elementu.

Všechny potřebné hodnoty lze získat pomocí tzv. bounding boxu²⁸ objektu reprezentujícího daný snímek. Informace z bounding boxu lze získat pomocí javascriptové funkce `getBBox()` zvané na daném objektu.

Nastavením těchto atributů však proces vložení HTML obsahu nekončí. Zřejmě nejzásadnějším problémem je úprava velikosti písma jednotlivých textových elementů. Jelikož parametry jednotlivých snímků se mohou velmi lišit, je nutné vytvořit nástroj, který bude na základě těchto parametrů dynamicky vypočítávat a nastavovat velikost písem. Tento proces je velice náročný především

²⁸Pomyslný obdélník ohraničující grafický prvek podle jeho krajních bodů.

proto, že snímek může často obsahovat mnoho textových prvků různých velikostí. U textového elementu navíc nelze nastavit jeho výšku a šířku. Jeho velikost se nastavuje pomocí atributu **font-size**, který specifikuje velikost jednoho znaku textového fontu.

Vzhledem k povaze problému bude nejjednodušší způsob použit tzv. **hidden-div**. Jedná se o HTML kontejner²⁹, do kterého bude vložen obsah snímku v základní nastavené velikosti. Tato velikost bude danou komponentou u každého textového elementu cyklicky snižována. V závislosti na této úpravě velikosti textů se bude zmenšovat i obalový kontejner. Pomocí tohoto procesu tedy bude možné nastavit velikosti textových elementů tak, aby vyplňovaly prostor daného snímku. Aby v průběhu výpočtu nebylo narušeno uživatelské prostředí prezentace, bude obalový kontejner v rámci HTML stránky skrytý.

Algoritmus 3: Dynamické nastavení velikosti textových elementů

Data: String htmlContent, Frame frame

Result: Html document with properly scaled frame content

resizer = hidden-div handler;

resizer.html = htmlContent;

while *resizer dimensions* > *frame dimensions* **do**

for *each html element from resizer.html* **do**

 | reduce size of html element by 1 unit;

end

end

updatedContent = resizer.html;

frame.html = updatedContent;

3.6 Shrnutí

Kapitola začíná volbou technologií pro realizaci jednotlivých funkčních celků řešení. Následuje popis procesu přípravy prezentace. S ohledem na aspekty tohoto procesu je proveden návrh všech potřebných komponent.

Pro přípravu obsahových materiálů byla vytvořena gramatika popisující formát obsahového souboru, díky které je struktura a obsah prezentace snadno a jednoznačně specifikovatelná. Pro zpracování obsahových a grafických podkladů a jejich export do výsledné prezentace byl navržen Inkscape plugin obsahující sadu nástrojů dělících se podle jejich funkce do samostatných bloků. Bohužel díky omezené sadě UI prvků, které program Inkscape poskytuje pro tvorbu pluginů, není navržené uživatelské prostředí ideální. Všechny funkční požadavky jsou však splněny.

Závěr kapitoly je věnován návrhu prezentační komponenty vygenerované plu-

²⁹HTML element, jehož jediná funkce je seskupit další elementy.

3. NÁVRH

ginem. Tato komponenta se stará o vykreslení prezentace s navigací.

Realizace

V rámci této kapitoly je stručně popsána implementace jednotlivých komponent systému navržených v kapitole 3. Příprava obsahové části prezentace pomocí obsahového souboru není v této kapitole zahrnuta protože v jejím případě jde pouze o návrh gramatiky (viz 3.3).

Pro jasnou identifikaci jsem celý systém realizovaný v rámci této práce nazval **Farslide**.

4.1 Plugin programu Inkscape

Jádro pluginu se skládá ze dvou hlavních souborů.

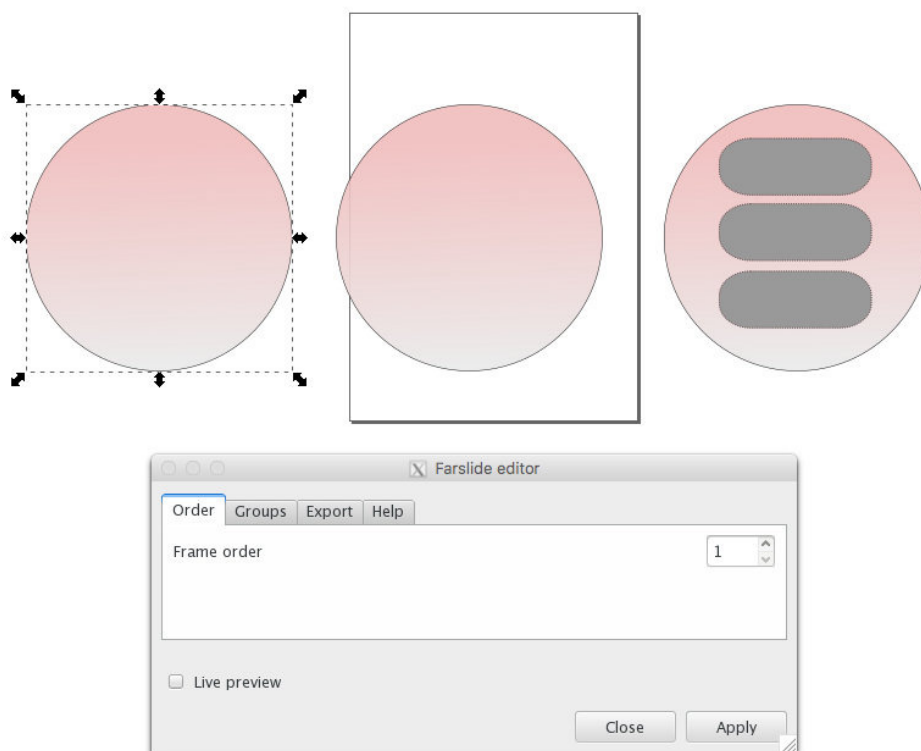
1. farslide.inx
2. farslide.py

Tyto soubory se dělí podle jejich účelu na dvě oblasti.

4.1.1 Uživatelské rozhraní

Uživatelské rozhraní pluginu je specifikované souborem **farslide.inx**. Tento soubor obsahuje veškeré informace o tom pod jakým klíčovým slovem je plugin umístěn v nabídce programu Inkscape, jakou má strukturu a z jakých prvků se skládá jeho uživatelské rozhraní. Samozřejmě také obsahuje vazbu na soubory s aplikační logikou.

Při implementaci uživatelského rozhraní pluginu nebylo použito žádné GUI knihovny jazyka Python. Rozhraní se skládá z uživatelských prvků dostupných v rámci specifikace INX souboru a v celé míře odpovídá navrhovanému uživatelskému rozhraní v kapitole 3. V sekci pro export prezentace byla navíc přidána možnost (zaškrťovací pole) pro vygenerování minimalizované verze prezentace.



Obrázek 4.1: Implementované uživatelské rozhraní pluginu

4.1.2 Aplikační logika

Aplikační jádro pluginu se skládá z jednoho Python souboru **farslide.py**, obsahujícího veškerý programový kód pluginu. Soubor obsahuje jednu hlavní třídu **Farslide** dědící ze třídy **Effect**³⁰. Tato třída obsahuje 3 funkce (`order_resolver()`, `groups_resolver()`, `export_resolver()`) obsahující logiku jednotlivých funkčních bloků pluginu.

Všechny resolver funkce byly implementovány s ohledem na vytvořený návrh. Pro propojení obsahových a grafických materiálů v rámci funkce `export_resolver` bylo nutné implementovat funkci pro parsování³¹ obsahového souboru. K tomuto účelu bylo využito knihovny **pyYAML** [10], která obsahuje funkce pro parsování YAML formátu (viz 2.4.1.1).

³⁰Třída zprostředkovaná prostředím Inkscape, která poskytuje nástroje pro vytvoření pluginu a funkce pro čtení a úpravu vektorového dokumentu.

³¹Proces získání informací ze zdroje s ohledem na předem specifikovaný formát.

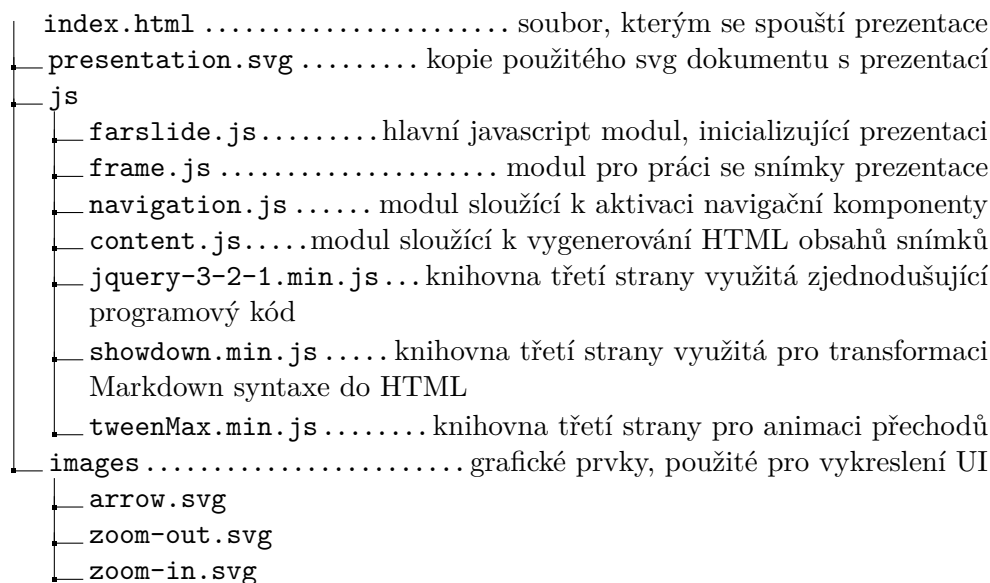
4.1.2.1 Generování prezentace

Pro účel vygenerování prezentace obsahuje plugin adresář **farslide**. Tento adresář obsahuje soubory, které slouží jako tzv. šablony, které v rámci exportu plugin přkopíruje do cílového adresáře prezentace.

Jediný soubor³², který je v rámci exportu prezentace upraven³³, je soubor **index_template.html** (případně **index_template_min.html** pokud uživatel zvolí minimalizovanou verzi prezentace). Jak již z názvu vyplývá, jedná se o hlavní soubor, určený ke spuštění prezentace. Tato šablona obsahuje HTML kód se zástupným tagem **{svg}**, který je v rámci exportu nahrazen SVG elementem s výslednou prezentací. Dále obsahuje tagy **{name}**, **{author}** a **{date}**, které jsou v případě vyplnění taktéž nahrazeny hodnotami s obsahového souboru. Takto upravený soubor je zapsán do cílového adresáře pod názvem **index.html**.

4.2 Prezentace

Adresář s vygenerovanou prezentací má následující strukturu.



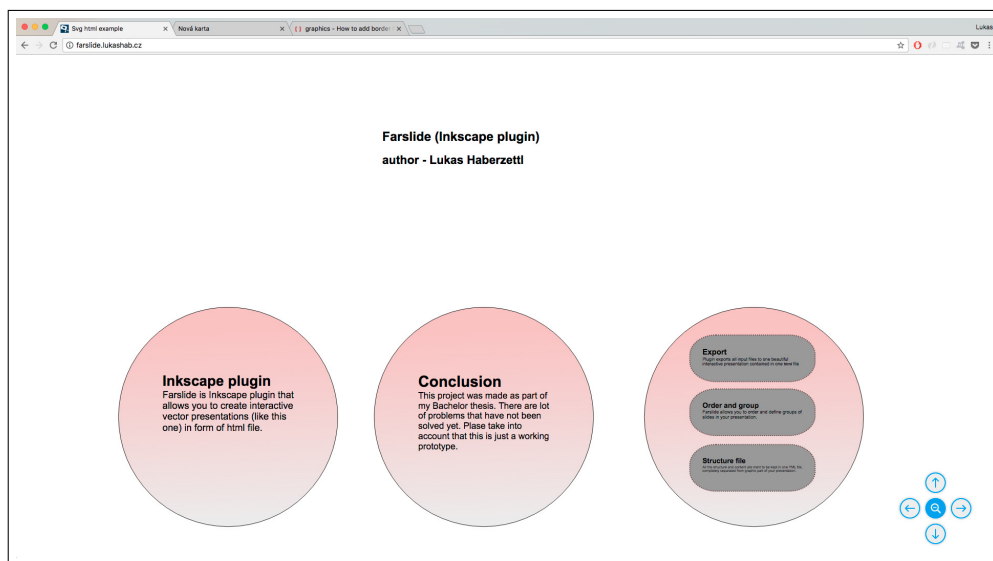
Jak je zřejmé ze schématu adresářové struktury, javascriptový kód byl pro přehlednost rozdělen do samostatných souborů podle funkčních oblastí. Pro transformaci obsahových atributů snímků do HTML kódu bylo využito knihovny **showdown** [11]. Pro pohyb a animaci přechodů mezi snímky bylo použito

³²Šablona

³³Ostatní soubory jsou pouze přkopírovány, protože u nich není potřeba nastavovat speciální informace týkající se vytváření prezentace.

4. REALIZACE

knihovny **TweenMax** (viz 2.4.3.4). Pro získání funkcí zjednodušujících manipulaci s DOM elementy byla využita knihovna **jQuery** [12]. V případě, že si uživatel zvolí v rámci exportu vygenerovat minimalizovanou verzi prezentace, výsledná adresářová struktura nebude obsahovat složku s javascriptovými soubory. Skripty v těchto souborech jsou v tomto případě obsaženy přímo v hlavním souboru **index.html**.



Obrázek 4.2: Výsledná prezentace spuštěná ve webovém prohlížeči

4.3 Shrnutí

V rámci této kapitoly byla popsána implementace jednotlivých komponent systému. Implementace zahrnuje vytvoření pluginu programu Inkscape a výsledné prezentační komponenty sloužící ke spuštění prezentace. Tyto dva celky byly implementovány v souladu s vytvořeným návrhem popsáním v kapitole 3 jak z pohledu uživatelského rozhraní tak aplikační logiky. Výsledkem implementace je tedy funkční systém aplikující proces přípravy prezentace popsány v sekci 3.2.

Testování

V průběhu implementace systému bylo v rámci iterací³⁴ vývoje průběžně prováděno testování klíčových funkcí. Pro toto testování nebyl vytvořen žádný sofistikovaný systém a aktérem testování byl pouze vývojář. I přesto však výstupy provedených testů posloužili jako dobrý zdroj informací o vlastnostech řešení. Na základě provedených testů byly na konci každé iterace identifikovány chyby a případné nedostatky současného stavu aplikace, které byly řešeny v další iteraci vývoje.

Na závěr implementace bylo provedeno celkové testování vytvořených komponent systému, jehož výstupem je popis současného stavu řešení a jeho nedostatků. Výstupy vzniklé v průběhu testování jsou popsány v příloze B.

5.1 Současný stav řešení

Implementovaný systém je v současné chvíli funkční a splňuje všechny funkční i nefunkční požadavky, které byly specifikovány v úvodu této práce. Řešení také pokrývá všechny případy užití tak, jak byly navrženy v sekci 2.3. Složitost procesu přípravy prezentace v rámci vytvořeného systému je větší než se předpokládalo a systém tak nemusí vyhovovat všem skupinám uživatelů. I přes tento fakt byl však cílovou skupinou označen jako dostatečně efektivní řešení, jehož výstupy jsou pro potřeby uživatelů uspokojující.

5.2 Nedostatky

V souvislosti s testováním výsledného řešení byly identifikovány následující nedostatky.

- Systém nepodporuje náhled nad obsahovými texty snímků v rámci přípravy prezentace. Vyrenderované obsahy vidí uživatel až ve formě ekspor-

³⁴etapy vývoje systému

tované prezentace. Tento fakt může působit problémy hlavně v případě tvorby rozsáhlejších prezentací.

- Způsob propojení obsahových a grafických podkladů je na úkor prostředí programu Inkscape málo intuitivní, což snižuje efektivitu procesu přípravy prezentace.
- Obsahy jednotlivých snímků se v některých situacích mohou vygenerovat v nežádoucích rozměrech. Tento nedostatek však řeší integrace stylů v obsahovém souboru, díky kterým si může uživatel výsledný obsah dostatečně upravit.
- Při hodně velkém počtu zanořených snímků se obsahy nevygenerují. Tento fakt je způsoben velmi malým měřítkem fontů obsahových textů, které webový prohlížeč zkrátka nevyrenderuje. V případě tohoto scénáře se dá obsah prezentace však vytvořit přímo v programu Inkscape.

5.3 Shrnutí

V rámci této kapitoly byl popsán způsob testování a identifikován současný stav implementovaného řešení včetně jeho nedostatků. Výsledný systém splňuje zadání této bakalářské práce a i přes jeho nedostatky aplikuje proces přípravy prezentace s uspokojujícím výsledkem dostatečně efektivně pro cílovou skupinu.

Závěr

Cílem mé práce bylo navrhnout a implementovat systém pro efektivní tvorbu interaktivních vektorových prezentací. Pro jasnou specifikaci vlastností řešení byly ve spolupráci s vedoucím vytvořeny funkční a nefunkční požadavky na výsledný projekt. Tyto požadavky byly detailně rozebrány v rámci analýzy, ve které jsem se také zabýval diskuzí nad existujícími řešeními a jejich použitelností pro můj projekt, vytvořením případů užití na základě funkčních požadavků, a rešerší použitelných technologií. Tato analýza posloužila jako dostatečný základ pro návrh a implementaci systému.

Výsledné řešení zahrnuje gramatiku obsahového souboru, pomocí které se vytváří obsahová část prezentace, a plugin programu Inkscape sloužící k propojení obsahové a grafické části a exportu těchto vstupů do výsledné prezentace. Ta má podobu jednoho HTML souboru, který spolu s ostatními podpůrnými soubory tvoří prezentační komponentu, zprostředkující náhled a interakci s vytvořeným dokumentem vektorové prezentace. Při implementaci jsem se snažil o maximální čitelnost a rozšiřitelnost. Veškerý kód je tedy řádně zdokumentován a, pokud je to možné, rozdělen do nezávislých modulů podle jeho funkce. Pro usnadnění práce jsem také využil několik knihoven, obsahujících potřebnou funkčnost. Tyto knihovny byly vybrány a použity s ohledem na jedinný nefunkční požadavek kladený na projekt a ve všech případech jde tedy o Open Source řešení.

Výsledný projekt splnil všechny specifikované požadavky a všechny definované případy užití. Zadání včetně všech cílů, kterých měla tato bakalářská práce dosáhnout, jsem tedy splnil.

Možná vylepšení implementovaného řešení

Současná verze systému má mnoho nedostatků, které jsou ve velké míře ovlivněny prostředím programu Inkscape. To ve značné míře omezuje možnosti pluginu, díky čemuž není proces přípravy prezentace dostatečně efektivní. Pro zlepšení efektivity se nabízí větší část přípravy prezentace přemístit do webo-

vého prostředí. V ideálním případě by mohlo jít o jednu aplikaci, která by umožnila aplikovat všechny části procesu. Toto řešení by mělo velkou výhodu ve své nezávislosti a jednoznačně větší efektivitě, nicméně jeho implementace by byla značně náročná a mnohonásobně by převýšila rozsah této bakalářské práce.

Dalším méně náročným vylepšením efektivity procesu by mohlo být vytvoření hlídacího skriptu, který by v průběhu tvorby prezentace monitoroval změny v zdrojových souborech prezentace a automaticky spouštěl funkci pro přegenerování včetně případného obnovení stránky s prezentací ve webovém prohlížeči.

Literatura

- [1] Sozi [online]. 2010. Dostupné také z: <http://sozi.baierouge.fr/>
- [2] Prezi [online]. Prezi, ©2017. Dostupné také z: <https://prezi.com/>
- [3] JessieInk. Launchpad [online]. 0017n. 1. [cit. 2017-05-02]. Dostupné z: <https://launchpad.net/jessyink>
- [4] Markdown. Daring Fireball [online]. The Daring Fireball Company LLC., ©2002-2017 [cit. 2017-04-08]. Dostupné z: <https://daringfireball.net/projects/markdown/>
- [5] Extensions. Inkscape [online]. [cit. 2017-04-08]. Dostupné z: <https://inkscape.org/en/develop/extensions/>
- [6] Node.js: about. Node.js [online]. Node.js Foundation, ©2017 [cit. 2017-04-08]. Dostupné z: <https://nodejs.org/en/about/>
- [7] INX_Parameters. Inkscape wiki [online]. [cit. 2017-05-04]. Dostupné z: http://wiki.inkscape.org/wiki/index.php/INX_Parameters
- [8] Inkex Namespace Reference [online]. [cit. 2017-05-05]. Dostupné z: <https://fossies.org/dox/inkscape-0.92.1/namespaceinkex.html>
- [9] TweenMax. <https://greensock.com>: Ultra high-performance, professional-grade HTML5 animation for the modern web [online]. GreenSock, c2017 [cit. 2017-05-07]. Dostupné z: <https://greensock.com/tweenmax>
- [10] PyYAML [online]. [cit. 2017-05-07]. Dostupné z: <http://pyyaml.org/>
- [11] Showdown. Github [online]. c2017 [cit. 2017-05-07]. Dostupné z: <https://github.com/showdownjs/showdown>
- [12] JQuery [online]. The jQuery Foundation, c2017 [cit. 2017-05-09]. Dostupné z: <https://jquery.com/>

Seznam použitých zkratk

OS Operační Systém

HTML Hypertext Markup Language

XHTML eXtensible Hypertext Markup Language

UI User interface (uživatelské rozhraní)

SWF Small Web Format

SVG Scalable Vector Graphics

XML eXtensible Markup Language

CSS Cascading Style Sheets

JSON JavaScript Object Notation

EBNF Extended Backus–Naur form (neboli Rozvinutá Backusova-Naurova forma)

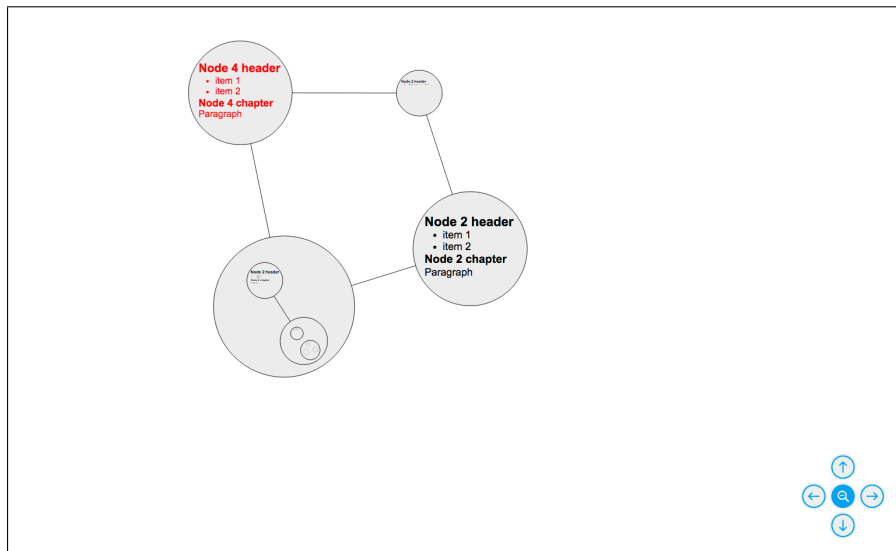
DOM Document Object Model

Výstup závěrečného testování

V rámci testování bylo prostřednictvím implementovaného řešení vytvořeno několik výstupů. Tyto výstupy se značně lišily zejména díky odlišným parametrům grafických a obsahových částí.

B.1 Výstup č. 1

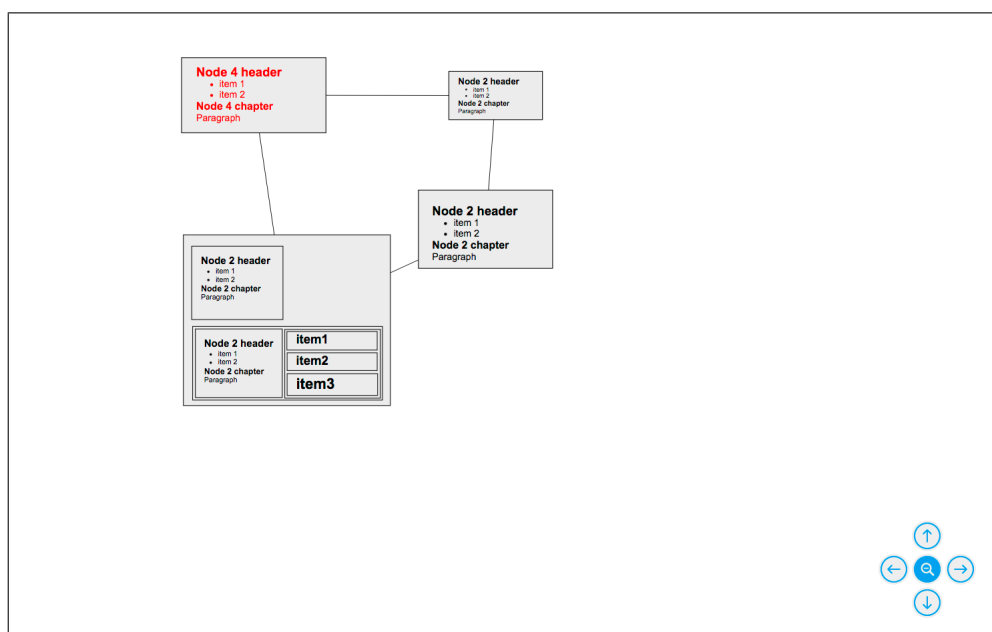
První výstup dopadl uspokojivě. Obsahy se vyrenderovaly do vymezených oblastí snímků ve správných velikostech. Díky absenci dodatečných stylů v obsahovém souboru jsou však texty neesteticky vsazené do snímků a na některých místech se dotýkají okraje snímku.



Obrázek B.1: Výstup bez specifikace dodatečných stylů pro snímky reprezentované kruhem

B.2 Výstup č. 2

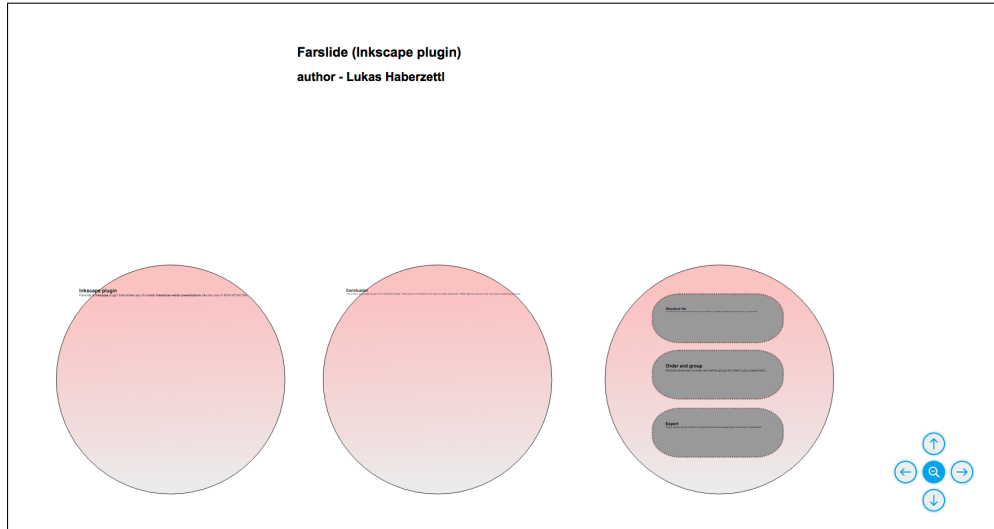
Druhý výstup dopadl velmi dobře. Hlavním faktorem úspěchu je zde nejspíš grafický formát snímků, které jsou reprezentovány obdélníkovým objektem. Vzhledem k tomu, že i HTML kontejner, ve kterém je výsledný obsahový text vyrenderovaný, má tvar obdélníku, je tento formát pravděpodobně nejvhodnější pro úspěšné vygenerování textových elementů bez použití dodatečných stylů.



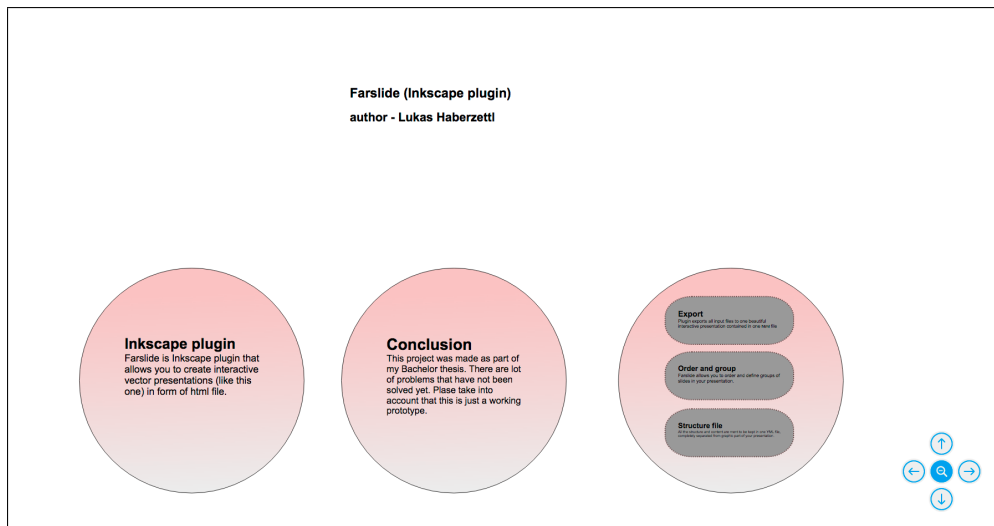
Obrázek B.2: Výstup bez specifikace dodatečných stylů pro snímky reprezentované obdélníkem

B.3 Výstup č. 3

Tento výstup dopadl bez použití dodatečných stylů bohužel neuspokojivě. Vzhledem k formátu snímků a délce textových elementů se sice obsah vsadil na správné pozice, avšak v nevyhovujících velikostech. I přes tento fakt však bylo možné po jednoduché úpravě v obsahovém souboru pomocí stylovacího atributu **size-ratio** vygenerovat prezentaci se správnými parametry.



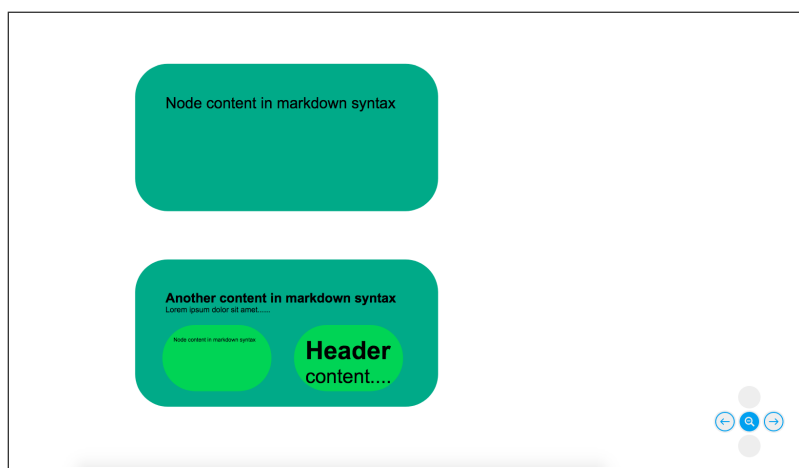
Obrázek B.3: Výstup bez specifikace dodatečných stylů pro snímky reprezentované kruhem



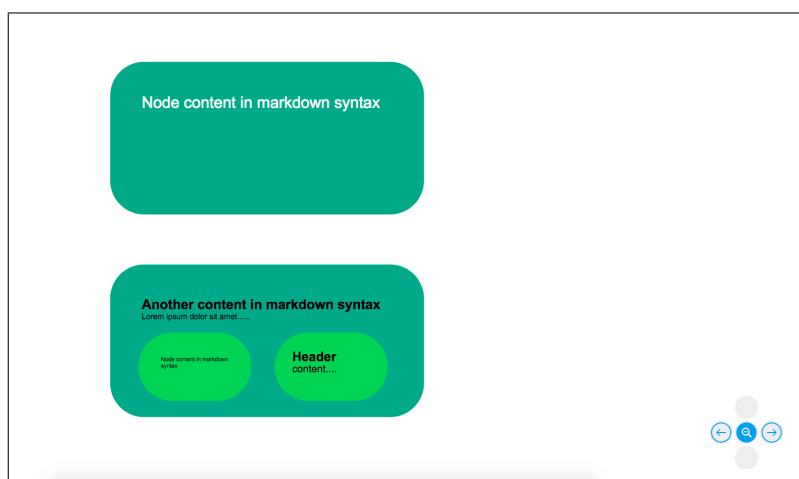
Obrázek B.4: Výstup systému pro snímky reprezentované kruhem po aplikaci stylů

B.4 Výstup č. 4

V rámci tohoto výstupu bylo otestováno vložení textového obsahu spolu se skupinou snímků do prvku typu **topic**. Výstup dopadl uspokojivě již bez použití dodatečných stylů. Textový obsah se vyrenderoval na správném místě spolu se snímkem. Pro větší estetiku byly následně pomocí stylů upraveny velikosti a zarovnání textů u některých snímků.



Obrázek B.5: Výstup systému pro topic s oběma druhy obsahu bez aplikování dodatečných stylů



Obrázek B.6: Výstup systému pro topic s oběma druhy obsahu po aplikování dodatečných stylů

Instalační příručka

C.1 Podporované operační systémy

Plugin Farslide lze nainstalovat a spustit v operačních systémech Windows, Max OS X a GNU/Linux. Vzhledem k tomu, že se jedná pouze o plugin a ne plnohodnotnou aplikaci, jeho instalace není nijak složitá.

C.2 Požadavky

Pro úspěšnou instalaci, spuštění a použití pluginu je nutné, aby byly na daném operačním systému nainstalovány následující nástroje.

- Program Inkscape
- Python knihovna PyYAML

C.3 Instalace

Instalace je stejná pro všechny operační systémy. Pro bezproblémovou instalaci pluginu je doporučen následující instalační postup.

1. Instalace programu Inkscape

- a) Nainstaujte program Inkscape na svůj operační systém.
- b) Podrobný návod instalace programu včetně všech instalačních souborů v závislosti na zvoleném operačním systému je umístěn na oficiálních stránkách programu (*inkscape.org*).

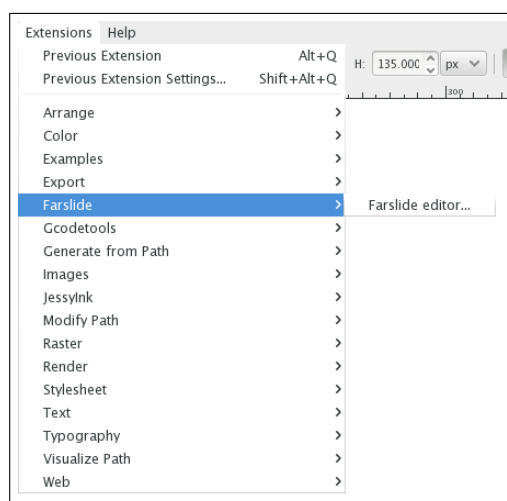
2. Instalace knihovny PyYAML

- a) Pro instalaci PyYaml knihovny je nutné, aby již na operačním systému bylo nainstalované prostředí programovacího jazyka Python (Toto prostředí je již zahrnuto v instalaci programu Inkscape).
- b) Pokud máte při instalaci problém s použitím Python prostředí poskytnuté programem Inkscape, nainstalujte si prostředí samostatně (viz python.org/downloads/).
- c) Návod instalace včetně všech nutných souborů je umístěn na oficiálních stránkách knihovny (pyyaml.org/wiki/PyYAMLDocumentation).

3. Instalace pluginu Farslide

- a) Překopírujte si na systémový disk aktuální verzi pluginu. Toto lze provést buď překopírováním pluginu z příloženého CD nebo stáhnutím git repozitáře z github profilu vývojáře - github.com/Lukashab/Farslide.
- b) Spusťte program Inkscape
- c) Pomocí horní nabídky programu přejděte do *Edit / Preferences / System*. V bloku *System info* je vypsána cesta k pluginům programu.
- d) Do této cesty zkopírujte obsah adresáře *plugin*, který naleznete v adresáři stažené aktuální verze pluginu.
- e) Restartujte program Inkscape.
- f) Pokud jste plugin zkopírovali na správné místo, bude po restartu vidět v nabídce *Extensions* položka *Farslide / Farslide editor*.
- g) Zvolením této položky spustíte plugin Farslide.

C.4. Další nástroje nutné pro stoprocentní využití pluginu



Obrázek C.1: Nabídka rozšíření v programu po úspěšném nahrání pluginu Farslide

C.4 Další nástroje nutné pro stoprocentní využití pluginu

K realizaci celého procesu přípravy prezentace se předpokládá, že má uživatel k dispozici také následující nástroje.

- Libovolný textový editor, umožňující tvorbu YML dokumentů (Příprava obsahové části prezentace).
- Libovolný webový prohlížeč (Spuštění výsledné prezentace).

Použité programy

TeXShop vytvoření bakalářské práce v \LaTeX a její překlad do formátu PDF

PyCharm psaní v programovacím jazyce Python

GIT verzování kódu

StarUML tvorba UML diagramů

Moqups tvorba aplikačních wireframů

Overleaf sdílení a záloha \LaTeX dokumentů

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
text	text práce
_ thesis.pdf	text práce ve formátu PDF
_ thesis.ps	text práce ve formátu PS
doc	dokumentace pluginu
src	
_ farslide.....	zdrojové kódy pluginu
_ thesis	zdrojová forma práce ve formátu \LaTeX