

Vizualizace Workflow

Projekt ITU, 2018Z

Číslo projektu: 99

Číslo a název týmu: 126. Tým xvalka05

Autor: Miroslav Válka (xvalka05)

Další členové týmu: Trněný Jan (xtrnen03), Jablonický Jaroslav (xjablo03)

Technická zpráva

Cílem projektu je navrhnout a implementovat uživatelské rozhraní k editoru workflow diagramů. (Pojmem workflow se obecně myslí množina stavů a množina přechodů mezi nimi.) Ovládání aplikace by mělo být intuitivní a často prováděné operace efektivní.

1) Popis aplikace

Aplikace by měla uživatelům umožnit rychle a efektivně vytvořit workflow diagram, který potřebuje udělat. Měl by zvládat vymodelovat konečný automat, diagram případů užití a další diagramy tvořené ze stavů (různé tvary) a vztahů mezi stavy (čáry, šipky).

2) Průzkum kontextu použití

Cílová skupina uživatelů

Cílovou skupinou aplikace jsou lidé potřebující vytvořit diagramy pro srozumitelné a přehledné znázornění fungování nějakého procesu. Nejedná se primárně o jejich hlavní náplň práce, proto chtějí diagram vytvořit rychle a jednoduše. Pracují na stolním počítači či notebooku na projektu, prezentaci nebo dokumentaci a na vytvoření diagramu nechtějí trávit mnoho času a zabývat se složitými způsoby jak nakreslit diagram.

Od cílové skupiny se tedy očekává, že se jedná o dospělého člověka se znalostí práce na počítači. Tento jedinec nemá v plánu kreslit diagram v malování a hledá nástroj, který mu dá na výběr předpřipravené části ze kterých si poskládá potřebný diagram. Pro jedince se jedná o činnost, kterou musí udělat a nechce na ní ztrácet čas.

Příklady uživatelů:

- Pracovník firmy, jenž modeluje vnitřní procesy firmy do prezentace pro vedení firmy. (např. diagram BPMN)
- Student FITu, který potřebuje vymodelovat konečný automat pro jeho projekt na Fitkitu.
- Student/Programátor vytvářející dokumentaci k lexikálnímu analýzátoru.
- Student potřebující vytvořit diagram případů užití (např. diagram UML)

Cílová platforma

Cílová skupina bude pracovat na projektu, prezentaci, dokumentaci takže je očekáváno použití stolního počítače či notebooku. Nelze přesně říci, jaký operační systém budou používat, proto je nutné počítat se základními dostupnými operačními systémy jako je Windows, Linux, Mac.

Jelikož není specifikován konkrétní operační systém a uživatelé očekávají rychlé použití, tak nutnost instalovat aplikaci do počítače je odrazující záležitost. Navíc pro některé uživatele je tvorba diagramu záležitostí, kterou řeší jen jednou za čas.

Proto bude ideálním řešením aplikace ve formě webové aplikace.

Existující řešení

V současnosti jsou již dostupné nástroje s danou funkcionalitou.

Jeden z konkurenčních produktů je Draw.io. Jedná se o komplexní nástroj pro tvorbu diagramů, poskytující široké možnosti úpravy.

Jeden z používaných nástrojů je také tzv. Nákres v Google dokumentech. Nejedná se o samostatnou aplikaci, ale jako vložený nástroj v Google Dokumenty či Google Tabulky. Je to velmi jednoduchý a rychlý způsob vytváření diagramu. Bohužel má velké množství omezení.

Další zástupcem může být bpmn.io. Tento nástroj je zaměřen na tvorbu BPMN diagramů. Dává možnost vytvářet jen jeden typ diagramu, ale zároveň je optimalizován pro práci s tímto konkrétním typem diagramu.

Ještě vezmeme příklad placené konkurence online.visual-paradigm.com. Nástroj je placen měsíčně a umožňuje tvořit převážně UML diagramy a vývojové diagramy.

Co očekává typický uživatel

Typický uživatel bude hledat způsob, jak vytvořit diagram. V okamžiku, kdy najde náš nástroj, zaujme ho jednoduché a čisté rozhraní, které mu na první pohled napovídá, jak vytvořit diagram, který potřebuje.

Uživatel chce vytvořit rychle a s minimální námahou diagram, který potřebuje. Bude ho chtít uložit třeba jako png obrázek a tento obrázek následně použít ve svém projektu.

Typické situace, které uživatel bude řešit

- 1) Přidat do diagramu nový element.
 - a) Vyhledá tvar/prvek, který se mu hodí. (Základní tvary musí být viditelně na očích a další tvary musí být snadno dostupné.)
 - b) Vložit vybraný tvar/prvek do diagramu. (Uživatelé mohou být zvyklí na více způsobů. Přetáhne prvek do diagramu, klikne na prvek a následně do diagramu, nebo něco jiného.)
- 2) Přemístit element z pozice A na pozici B.
 - a) Vybere element, který chce přesunout.
 - b) Chytne element, přesune ho na vybrané místo a pustí ho. (Uživatel chce vidět, jak se element posouvá v reálném čase.)
- 3) Posouvání plochy na které se nachází diagram. (Při posouvání nechce, aby omylem rozbil svůj diagram.)
- 4) Smazat element, který již v diagramu nechce.
 - a) Vybere element, který chce odstranit.

- b) Vybere akci pro smazání elementu. (Někteří uživatelé mohou být zvyklí na zažité zkratky - klávesa Delete nebo Backspace, která běžně mažou vybraný obsah.)
- 5) Propojit dva elementy.
- a) Vybere, co má elementy propojovat. (čára, šipka a jejich vzhled. To co uživatel použil pro spojení jednou, tak bude chtít použít i pro další spojení a nebude chtít opakovaně vybírat typ propoje.)
 - b) Vybere 1. element od kterého povede propojení. (Uživatel chce vidět, že vše probíhá správně a očekává, že propoj bude vidět již při hledání 2. elementu)
 - c) Vybere 2. element do kterého povede propojení. (A nejspíše bude chtít propojit další dva elementy)
- 6) Uložit výsledný diagram.
- a) Klikne na uložení/export diagramu. (Na první pohled musí být jasné v jaké formě diagram uloží. Obrázek ve formátu SVG, PNG nebo jiném.)

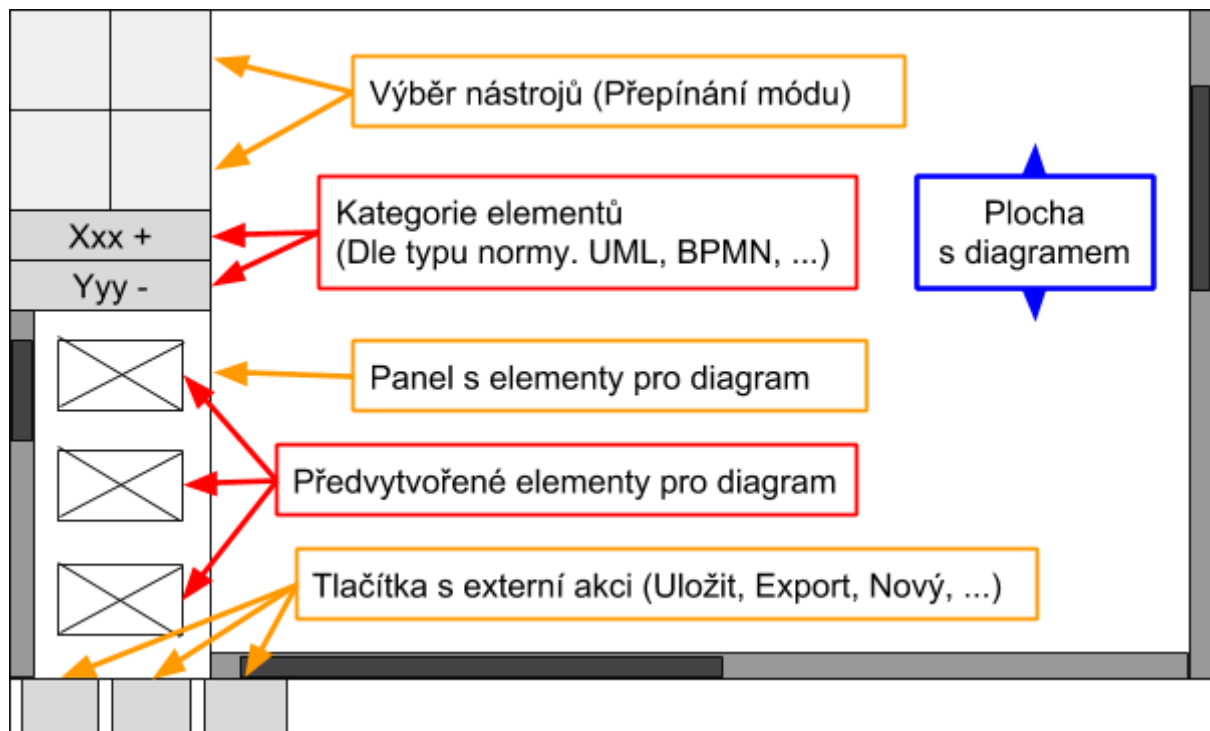
3) Návrh klíčových prvků UI

Prvky UI důležité pro zaujmutí uživatele

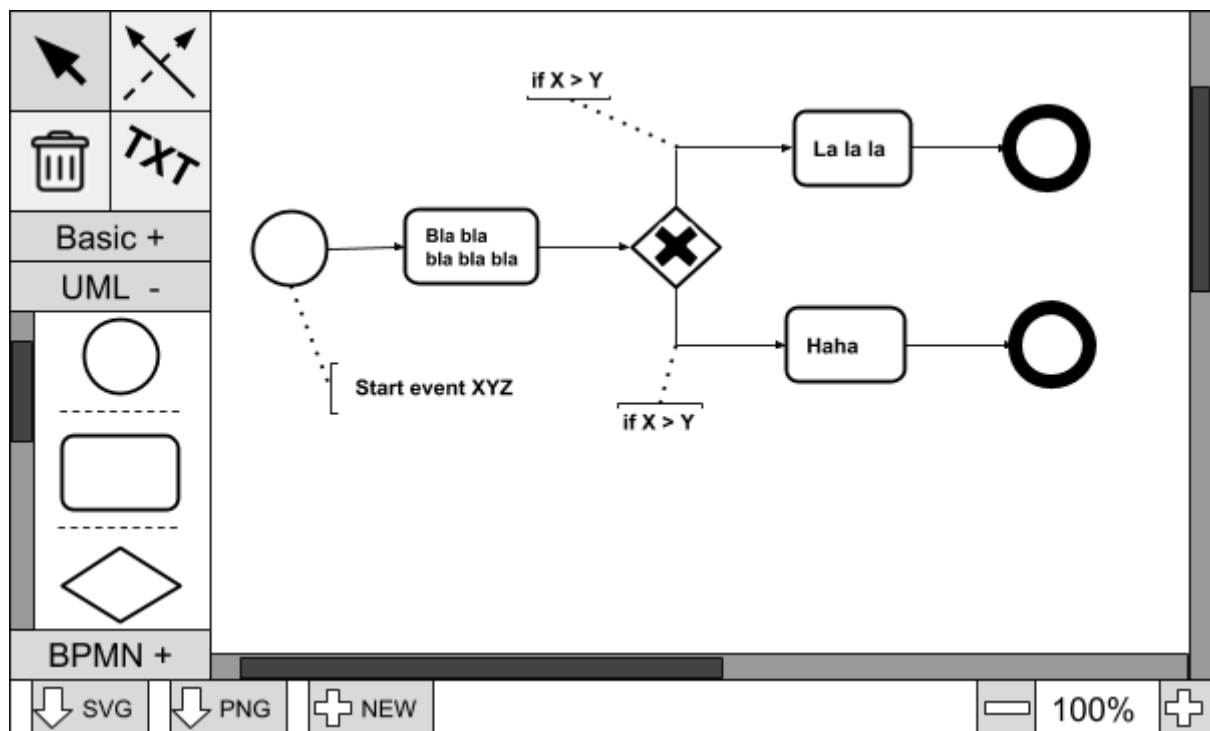
- 1) Viditelný panel nástrojů s malým počtem nástrojů.
 - a) Vybrání nástroje mění mód. Je nutné vybraný mód vizuálně indikovat a odlišit aktuálně vybraný nástroj od nevybraných nástrojů.
 - b) Mít pouze 3 až 4 nástroje. Nechceme uživatele nutit pamatovat si desítky módů s různou funkcionalitou. Naši uživatelé chtějí jednoduchost a nechtějí se dlouho učit, jak pracovat s naší aplikací.
- 2) Panel s elementy pro diagram.
 - a) Základní předvytvořené elementy musí uživatel na první pohled vidět.
 - b) Panel musí být hned vedle kreslicí plochy s diagramem. (Krátká vzdálenost mezi místem, kde si vybírá uživatel element a místem kam ho umísťuje.)
 - c) Elementy by měli být rozříděné do kategorií. (Základní, UML, BPMN, ...)
 - d) Více způsobů jak vkládat nový element.
 - i) Přetáhnout element z panelu na kreslicí plochu.
 - ii) Kliknout na element a následně kliknout na kreslicí plochu.
 - e) "Scrollování" by mělo být vertikální a scrollbar by neměl být mezi elementy a kreslicí plochou.
- 3) Panel s tlačítky vyvolávající externí akci. Tj. vyvolají dialogové okno. (Ukládání, Export, Nový projekt, ...)
 - a) Vzhled tlačítek musí indikovat akci, která bude vykonána.
 - b) Tyto tlačítka nesmí být nalepeny příliš blízko na sebe. (Překliknutí může být velmi otravné neboť vyvolá dialogové okno, které uživatel nechce.)
- 4) Manipulace s elementy na kreslicí ploše.
 - a) Uživatelem označené/vybrané elementy musí být vizuálně zvýrazněny. (Volil bych tradiční orámování elementu)
 - b) Elementy by měli jít přetahovat po kreslicí ploše tzv. "drag and drop".
 - c) Měnění velikosti elementu.
 - i) Buď 8 bodů ve kterých lze manipulovat s rozměry a pozicí elementu za pomoci "drag and drop". (Body v rozích obdélníku a v polovině stran) (Jedna se o známý prvek z existujících kreslicích nástrojů)
 - ii) Nebo 1 bod v pravém dolním rohu ve kterém lze manipulovat s šířkou a výškou za pomoci "drag and drop". (Větší jednoduchost a menší šance, že uživatel změní velikost omylem.)
 - d) Další možné akce vložit do "contextmenu". (Akce jako 'Odstranit', 'Editovat Text', ...)

4) Návrh GUI a prototyp

Návrh GUI aplikace - Mockup v1



Bližší ukázka GUI aplikace - Mockup v2



Ukázka zobrazení vybraného elementu. (Dvě varianty)



1) Klasický výběr s 8 body
pro změnu velikosti



2) Jednoduchý výběr s 1
bodem pro změnu velikosti

5) Testování prototypu GUI

Individuální návrh testování

Při navrhování GUI každý z týmu navrhnul vlastní GUI aplikace, proto je žádoucí podrobit testování všechny verze a zjistit preference uživatelů. Neshledávám za nutnost zjišťovat, které řešení je lepší, ale které části daného návrhu byli uživateli lépe přijaty a jaký způsob používání jim přijde intuitivnější.

Při testování bych preferoval dva způsoby zjišťování dat. První způsob, který se bude zaměřovat na kvantitativní výsledky, ve formě dotazníku. A druhý způsob, který bude zaměřen na konkrétní subjekt a jeho práci s rozhraním, ve formě testování fyzického prototypu rozhraní na menším vzorku uživatelů.

První způsob ve formě dotazníku by měl obsahovat ukázkou daného GUI nebo různé situace s daným GUI. Tyto situace by měly být doplněny sérií otázek k dané situaci. Měly by zjišťovat, jakým způsobem by uživatel danou část řešil.

Příklad pro testování s dotazníkem:

- 1) Máme obrázek GUI s rozpracovaným diagramem, který obsahuje jen jeden obdélník.
 - a) Chcete přidat kolečko vedle již existujícího obdelníku. Jak tuto akci provedete?
 - i) Najdu kolečko v levém sloupečku. Chytnu kolečko myší. Přejedu s kolečkem vedle obdelníku. Kolečko pustím.
 - ii) Najdu kolečko v levém sloupečku. Kliknu na kolečko. Kliknu myší na místo na kterém se má objevit kolečko.
 - iii) Najdu kolečko v levém sloupečku. Dvakrát na kolečko kliknu a očekávám, že se vloží do diagramu, Kolečko přetáhnu na správnou pozici.

Druhý způsob ve formě testování fyzického prototypu. Pro tento způsob testování je nutné vytvořit například z papíru prototyp dané GUI a další potřebné části, kterými budou modelovány odezvy na akce testovacího subjektu. Funkcionalitu dané aplikace nahradí testující, který bude upravovat GUI dle akcí testovacího subjektu. Pro tato testování je potřeba připravit scénáře, které budou testovanému subjektu zadávat, co je jeho cílem. Testující bude nejen simulovat odezvu GUI, ale i pozorovat akce subjektu a zapisovat informace. Informacemi je myšleno:

- 1) Při jaké činnosti se subjekt zasekl.
- 2) Jaké části subjektu nečinili problém.
- 3) Jak moc se subjekt přiblížil požadovanému cíli.
- 4) Postup jakým subjekt postupoval při řešení
 - a) Při tvorbě diagramu nejprve vložil všechny prvky a až následně je propojil.
 - b) Při tvorbě diagramu spojoval prvky ihned po vložení do diagramu.

Výsledný testovací protokol

Testování bude probíhat na subjektech z cílové skupiny (první část dokumentu) primárně formou dotazníku, který bude doplněn prototypem / mockupem rozhraní editoru. Cílem tohoto dotazníku bude zjistit, zda-li se subjekty rychle orientují v rozhraní a jsou schopni popsat, jak by s daným rozhraním pracovali, aby mohli vytvořit diagramy. Mezi klíčové body testování řadíme:

- 1) Celkové rozložení rozhraní
- 2) Přehlednost v prvcích diagramu tzn. barvy, zvýraznění, velikost a písmo textu
- 3) Kategorizaci komponent v menu a rychlá orientace v menu
- 4) Zpětná vazba na specifické prvky navržené členy týmu

Z pozorování a dat z dotazníku vyvodíme, zda-li rozložení prvků uživatelům vyhovuje a orientují se. Dále zjistíme, jestli je rozhraní dostatečně kontrastní a prvky jsou viditelné a neztrácejí se buď v pozadí nebo mezi sebou navzájem. Z dotazníku také získáme zpětnou vazbu na nově navržené prvky rozhraní a jsou-li tyto prvky pro uživatele zajímavé, či přínosné.

Pro testování každý člen týmu využije svůj dotazník, který následně celý tým využije pro zjištění, které rozhraní bylo pro subjekty přívětivější, popř. které prvky rozhraní lze od každého člena použít do výsledného skupinového rozhraní.

Realizace testů

Testování formou pouhého dotazníků:

Testovacím subjektům byl předložen dotazník a byli požádáni o jeho vyplnění.

Testování formou fyzického prototypu:

Moje testování prototypu probíhalo na třech pokusných subjektech. Tyto subjekty plnily zadané úkoly a probíhalo pozorování jejich počínání při práci.

Úkony prováděli nad papírovým prototypem velikosti A4 s pohyblivými částmi pro nabídku s elementy, kartičkami představující různé elementy diagramu (kolečka, obdélníky, kosočtverce, panáčky, obálky a další..) a provázky simulující propoje mezi elementy.

Zjištěné informace z pozorování jsou zaznamenány v [externím souboru^{\(1\)}](#).

Subjekty následně vyplnili i dotazník.

Výsledky a závěry

Testování probíhalo na studentech, jejichž zaměření je ekonomika či elektrotechnika a podobná aplikace by se jim mohla hodit. Při práci s prototypem nebyli vyrušováni a byla jim umožněno zvolit si způsob ovládání jim vlastní. Jejich počínání bylo sledováno a zaznamenáváno.

Z pozorování bylo zjištěno, že nejčastěji používaný způsob vkládání nových elementů do diagramu bylo “Přetáhnutí vybraného elementu do diagramu” (Odpovídá: Drag and Drop akci myši). Jediným používaným způsobem propojení dvou elementů bylo “Vybrání šipky, kliknutí na první element a následně kliknutí na druhý element” (Odpovídá: Vybrání nástroje a kliknutí na elementy v pořadí ve kterém je očekávána výsledná šipka.) Způsob mazání byl taktéž jasný a všichni používali “Výběr popelnice a následně elementu pro odstranění”. Nicméně všechny testovací subjekty uvedly, že by aplikace měla mít jednotný způsob ovládání.

Studijní zdroje

Při vytváření a návrhu byly využity zveřejněné prezentace z přednášek. Byly taktéž využity nabyté vědomosti z přednášek o způsobech testování a získávání informací od koncového uživatele.

Přílohy

- 1) Soubor s testováním fyzického prototypu.
 - a) <https://docs.google.com/spreadsheets/d/1laYLne-edipOBDEF67IS6p3587xOmRyGhqaKAUvwLUM/edit#gid=0>
- 2) Dotazník xvalka05
 - a) <https://docs.google.com/forms/d/e/1FAIpQLSdvHTCzrAxArkMDS3u0fWwJvbQqGVi7nYsW9cZKhWgumjkwQ/viewform>

6) Implementace

Výběr technologií

Probíhal vývoj webové aplikace, takže bylo použito HTML, CSS3, SVG 2 a Javascript. V Javascriptu byla použita knihovna svg.js a její pluginy jako svg.draggy.js, svg.foreignobject.js, svg.connectable.js. Pokud jde o plugin svg.connectable.js byla použita upravená verze od "loredanacirstea".

Cílem aplikace je vytvářet diagramy a umožňovat jejich použití v jiných dokumentech. Z tohoto důvodu je použití SVG v HTML vhodnou volbou. S svg prvky je možné pracovat stejně jako s jinými html prvky, ale oproti běžným html prvkům mají výhodu ve způsobu uložení. Celý výsledný svg obrázek je možné převést a uložit bez funkcionality nebo jej převést na rastrový obrázek. Tyto obrázky může poté uživatel aplikace použít ve svých dokumentech.

Knihovna svg.js byla zvolena pro své zaměření na manipulaci s svg obrázky. Poskytuje vhodnou abstrakci nad elementy a dodává k nim funkce pro jednodušší manipulaci s svg elementy. Jednotlivé pluginy následně tuto řadu funkcí rozšiřují.

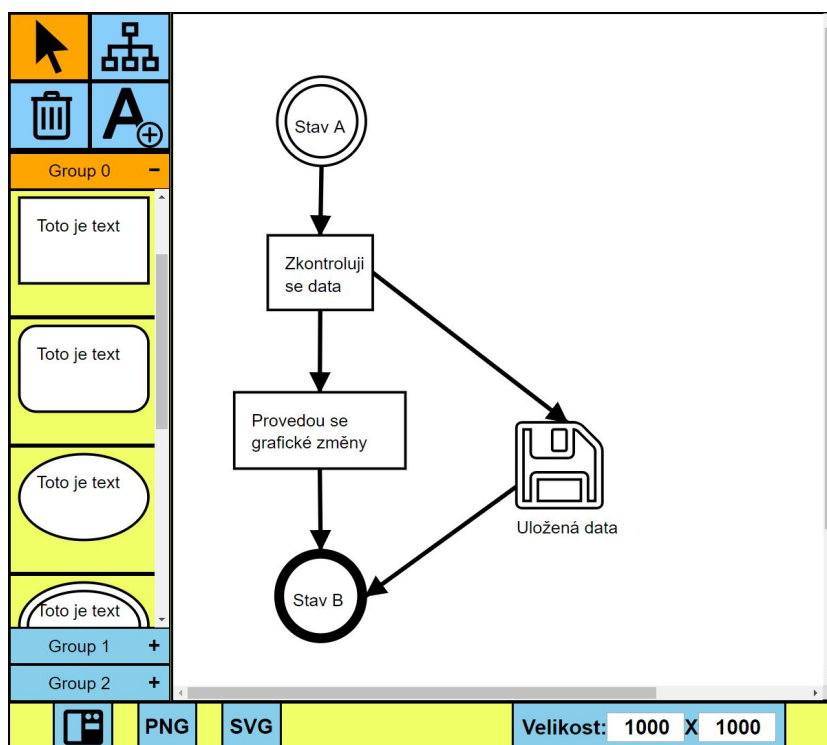
Aplikace

Jedná se o webovou singlpage aplikaci. Mezi funkční částí back-endu a zobrazovací částí front-endu není žádná dělící hranice. Kvůli velmi blízkému napojení na svg obrázek a jeho elementy se funkcionality i ovládání nachází na jednom místě.

V mé konkrétní implementaci je aplikace rozdělena na několik částí. V javascriptu ve velkém využívám tzv. "callbacků", které obstarávají chování elementů pro různé módy aplikace.

Hlavní částí je kreslicí plocha na níž se nachází tvořený diagram. Tato část zodpovídá za správu elementů či propojů a jejich manipulaci s nimi v diagramu.

Další velkou částí je samotné vytváření elementů, které je možné vkládat do diagramu a jaké chování budou mít (Mohou měnit velikost, mít text, pozicování textu, atd.).



7) Týmová spolupráce

Ohledně týmu byli problémy vzhledem k nepochopení účelu týmu. Z prvotního vysvětlení jsem měl za to, že týmy jsou pouze pro chycení daného zadání, které aktéři chtějí (Podobně jako cvičení.). Následné další vysvětlení na přednáškách nás vedlo k názoru, že každý má vytvořit vlastní GUI a sdílet můžeme jen informace a funkcionalitu pozadí aplikace (Nemluvím o tom, že ohledně tohoto byli neustále zmatky, kdy každý chápal spolupráci jinak.) Takže spolupráce u nás probíhala vzájemně si vyměňujícími informacemi a hlídáním termínů pro jednotlivá odevzdání.

8) Závěr

V projektu jsme byli donuceni dělat věci, které bychom normálně opomíjeli dělat. Tím mám na mysli zjišťování a rozebírání cílové skupiny aplikace nebo testování prototypu na osobách, které spadají do cílové skupiny.

Testování prototypu shledávám jako nejvíce přínosnou zkušenost. Testovací subjekty měli pro mne neočekávané smýšlení a pokud první způsob, který použili k manipulaci fungoval, tak nad jiným způsobem již nepřemýšleli. A přesně jak bylo řečeno na přednáškách, tak subjekty odmítali testovat prototyp pokud měli pocit, že jsou zkoušeni oni.

Další přínosnou zkušeností je pro mne přemýšlení nad způsobem, jak zkombinovat více typů ovládání, které plní stejnou úlohu způsobem, aby se navzájem nenarušovali.

Ovládání výsledné aplikace vychází z pozorování provedených na testovacích subjektech a vzhled aplikace je inspirován konkurenčními existujícími řešeními a snaží se o maximální jednoduchost a pochopitelnost.

Aplikace je dostupna na: itu.mwarcz.cz