Hypothesis

Dokumentace aplikace verze 1.0

Únor 2015

Obsah

Úvod

Hypothesis je webová aplikace pro přípravu testových baterií a následné provádění a vyhodnocení performačních testů.

Aplikace je vyvinuta za použití moderních technik dynamického webu. Výkonné jádro a uživatelské rozhraní je postaveno nad frameworkem Vaadin 7, práci s databází obstarává ORM Hibernate a za výchozí databázový stroj je použit PostgreSQL ve verzi 9.1 (případně vyšší).

Architektura aplikace je třívrstvá – klient, server a databáze. Klientská část slouží pro komunikaci a interakci s uživatelem a její běh zprostředkovává standardní systémový webový prohlížeč (tenký klient) nebo speciální prohlížeč distribuovaný v rámci aplikačního balíku – Hypothesis Browser. Tento prohlížeč je vystavený na komponentách Standard Widget Toolkit a umožňuje zabezpečit striktnější podmínky pro provádění testů. Klientská část komunikuje se serverem pomocí techniky Ajax RPC (vzdálené volání procedur) na pozadí.

Serverová část je implementována jako servlet aplikačního serveru (Apache Tomcat), který zpracovává požadavky klienta a aktualizuje uživatelské rozhraní. Servlet následně komunikuje s databázovým systémem metodami objektového mapování entit prostřednictvím knihovny Hibernate. Tato knihovna umožňuje připojení ke všem běžně používaným databázovým systémům (PostgreSQL, MySQL, MS SQL, Oracle aj.) na základě jednotného rozhraní.

Jednotlivé testové baterie (balíčky) jsou strukturovaně uloženy v databázi. Balíček je složen z větví, které obsahují jednu nebo více úloh a každá úloha obsahuje nejméně jednu scénu. Scéna je složena ze šablony a obsahu.

Spuštěním testu dojde k načtení zvoleného balíčku z databáze do serverové aplikace a k vytvoření nové instance testu. Při provádění testu jsou procházeny větve podle jejich definovaných podmínek na základě interakce probanda. V rámci větve jsou lineárně procházeny jednotlivé úlohy a v nich postupně vystavuje scény. Každá interakce pokusné osoby (např. stisknutí tlačítka, atd.) je zaznamenávána do logu událostí. Po provedení úkolu je vyhodnocena správnost řešení.

Scéna je vytvořena z jednotlivých komponent, které jsou rozděleny do šablony a obsahu. Šablona je vhodná pro obecné části, obsah může být variabilní. Obě části scény jsou popsány ve formátu XML. Při sestavování testu jsou obě části scény složeny do jednoho XML, podle kterého jsou vytvořeny prvky uživatelského rozhraní.

V následujících částech je vysvětlováno, jak jsou tvořeny konfigurační XML soubory testů. Ty jsou však často velmi dlouhé a komplikované. V následujících kapitolách jsou proto většinou použity jen kratší výseky XML kódu, které vysvětlují konkrétní elementy. Příklady kompletních a komplexních šablon a obsahů najdete v dokumentu Návod_zakázka_PřF.pdf. K dispozici jsou rovněž předpřipravené XML dokumenty šablon a obsahu.

Struktura databáze

Testová baterie je složena z dat obsažených v jednotlivých tabulkách, které tvoří hierarchickou strukturu. Nejvýše v této struktuře je tabulka ***tbl_pack*** (balíček, baterie), druhá je tabulka ***tbl_branch*** (větev), pod ní tabulka ***tbl_task*** (úloha), následuje tabulka ***tbl_slide*** (scéna) a na poslední úrovni jsou tabulky ***tbl_slide_content*** (obsah) a ***tbl_slide_template*** (šablona).

Záznam v tabulce ***tbl_slide*** představuje scénu a obsahuje odkaz na id patřičného obsahu v tabulce ***tbl_slide_content***, kde je navazující odkaz na šablonu v tabulce ***tbl_slide_template***.

Záznam v tabulce ***tbl_task*** představuje úlohu, která je tvořena 1..n scénami a tento vztah je uložen v propojovací tabulce ***tbl_task_slide***. V tabulce ***tbl_task*** lze pomocí atributu randomized nastavit, že pořadí scén může být libovolné.

Záznam v tabulce ***tbl_branch*** představuje větev a obsahuje 1..n úloh propojených tabulkou ***tbl_branch_task***.

Záznam v tabulce ***tbl_pack*** představuje testovou baterii tvořenou z 1..n větví se vztahem v propojovací tabulce ***tbl_pack_branch***.

Průběh testu, tj. jeho provádění pokusnou osobou, je ukládán do hierarchické struktury v tabulkách ***tbl_test*** a ***tbl_event*** s propojovací tabulkou ***tbl_test_event***. Záznam v tabulce ***tbl_test*** představuje jeden průběh testu, záznam v tabulce ***tbl_event*** představuje určitou událost, která nastala při vykonávání testu. Jedná se o události jak uživatelské (např. kliknutí na tlačítko) tak systémové (např. nahrání nové scény). Na jednu scénu testu může připadat více událostí v tabulce ***tbl_event***.

Další tabulkou, která obsahuje výsledky průchodu testu je ***tbl_slide_output***. Tato tabulka se odlišuje tím, že každý záznam obsahuje výsledek jen právě jedné scény. Do této tabulky se zapisuje finální stav scény – tedy např. údaje vyplněné do dotazníkových polí.

Tabulky ***tbl_user***, ***tbl_user_permission***, ***tbl_user_role***, ***tbl_group***, ***tbl_group_permission*** slouží pro správu uživatelů a jejich práv.

Tabulky ***tbl_branch_trek*** a ***tbl_branch_branch_trek*** se používají při větvení testu.

Tabulka ***tbl_branch_trek*** - se používá při větvení, obsahuje key (klíč definovaný uživatelem) který zastupuje branch_id pro dané pack_id, key se používá v xml_data v tabulce tbl_branch. Tabulka ***tbl_branch_branch_trek*** - join tabulka pro vazbu m:n mezi branch a branch_trek. Tabulka ***branch_output*** – zatím se nepoužívá, je připravena pro záznam výsledku celé větve.

Popis tabulek

Tato sekce znázorňuje seznam a obsah databázových tabulek.

Tabulky pro skládání testů:

Tabulka tbl_pack

id	bigint	primární klíč
description	varchar(255)	popis testu
name	varchar(255)	název testu
published	bool	test je/není zveřejněný
note	varchar(255)	případná poznámka

Tabulka tbl_branch

id	bigint	primární klíč
xml_data	text	popis ve formátu XML
note	varchar(255)	název větve v rámci testu

Tabulka tbl_task

id	bigint	primární klíč
note	varchar(255)	popis úlohy
name	varchar(255)	název úlohy
randomized	bool	scény úlohy mají/nemají pevné pořadí

Tabulka tbl_slide

id	bigint	primární klíč
slide_content_id	bigint	id obsahu v tabulce tbl_slide_content
note	varchar(255)	název scény

Tabulka tbl_slide_content

id	bigint	primární klíč
xml_data	text	XML definice obsahu
note	varchar(255)	název
slide_template_uid	varchar(255)	id šablony v tabulce tbl_slide_template

Tabulka tbl_slide_template

uid	bigint	primární klíč
xml_data	text	XML definice šablony
note	varchar(255)	název

Tabulka tbl_pack_branch

pack_id	bigint	id testu v tabulce tbl_pack
branch_id	bigint	id větve v tabulce tbl_branch
rank	int	Pořadí větve v testu. I větve mají pořadí, je nutné určit první větev.

Tabulka tbl_branch_task

branch_id	bigint	id větve v tabulce tbl_branch
task_id	bigint	id úlohy v tabulce tbl_task
rank	int	pořadí úlohy ve větvi

Tabulka tbl_task_slide

task_id	bigint	id úlohy v tabulce tbl_task
slide_id	bigint	id scény v tabulce tbl_slide
rank	int	pořadí scény v úloze

Tabulky používané při větvení:

Tabulka tbl_branch_trek

id	bigint	primární klíč
key	varchar(255)	klíč definovaný uživatelem, používá se ve sloupci xml_data v tabulce tbl_branch
branch_id	bigint	id větve v tabulce tbl_branch
pack_id	bigint	id testu v tabulce tbl_pack

Tabulka tbl_branch_branch_trek

branch_id	bigint	id větve v tabulce tbl_branch
branch_trek_id	bigint	id z tabulky tbl_branch_trek

Tabulky zaznamenávající průchod uživatele testem:

Tabulka tbl_test

id	bigint	primární klíč
created	timestamp	čas vytvoření testu prohlížečem
started	timestamp	čas spuštění testu uživatelem
broken	timestamp	čas předčasného ukončení testu
finished	timestamp	čas klasického ukončení testu
last_access	timestamp	čas posledního zápisu testu do databáze
production	bool	Ano – jde o ostrý test připravený k použití. Ne – jde o zkušební test.
status	int	Indikátor způsobu průběhu testu: 1- test vytvořen 2- test spuštěn 3- test úspěšně ukončen 4- test ukončen předčasně uživatelem 5- test ukončen předčasně chybou programu
last_branch_id	bigint	id poslední spuštěné větve v tbl_branch
last_task_id	bigint	id poslední spuštěné úlohy v tbl_task
last_slide_id	bigint	id poslední spuštěné scény v tbl_slide
pack_id	bigint	id testu v tabulce tbl_pack
user_id	bigint	id uživatele, který test spustil, v tabulce tbl_user

Tabulka tbl_event

id	bigint	primární klíč
xml_data	text	XML popis událostí – souřadnice kliku, název použitého tlačítka apod.
name	varchar(255)	název události – START TEST, BUTTON CLICK apod.
branch_id	bigint	id větve v tabulce tbl_branch
task_id	bigint	id úlohy v tabulce tbl_task
slide_id	bigint	id scény v tabulce tbl_slide
timestamp	bigint	časový otisk
type	bigint	kód typu události

Tabulka tbl_test_event

test_id	bigint	id spuštění testu v tabulce tbl_test
event_id	bigint	id události v tabulce tbl_event
rank	int	pořadí události v průběhu testu

Tabulka tbl_slide_output

id	bigint	primární klíč
xml_data	text	XML popis události – např. informace o vybraných formulářových prvcích
output	varchar(255)	XML popis události – výstupní hodnota, pokud je ve scéně definována
slide_id	bigint	id scény v tabulce tbl_slide
test_id	bigint	id spuštění testu v tabulce tbl_test

Tabulka tbl_branch_output

id	bigint	primární klíč
xml_data	text	XML popis události – např. informace o vybraných formulářových prvcích

output	varchar(255)	XML popis události – výstupní hodnota, pokud je ve scéně definována
branch_id	bigint	id větve v tabulce tbl_branch
test_id	bigint	id spuštění testu v tabulce tbl_test

Tabulky pro správu uživatelů:

Tabulka tbl_user

id	bigint	primární klíč
password	varchar(255)	heslo
username	varchar(255)	uživatelské jméno
enabled	bool	uživatel je/není povolen
expire_date	timestamp	datum, do kdy je uživatelské jméno platné
note	varchar(255)	popis uživatele
owner_id	bigint	id uživatele, který vytvořil daného uživatele

Tabulka tbl_role

id	bigint	primární klíč
name	varchar(255)	jméno uživatelské role – uživatel, administrátor…

Tabulka tbl_user_role – přiřazení role a uživatele

user_id	bigint	id uživatele
role_id	bigint	id role přiřazené uživateli

Tabulka tbl_user_permission – přiřazení práv na určitý test uživateli

id	bigint	primární klíč
enabled	bool	přiřazení práv uživteli je/není aktivní
pass	bigint	počet průchodů testem povolený danému uživateli
pack_id	bigint	id testu, k němuž je povolen přístup

user_id	bigint	id uživatele, jemuž je povolen přístup

Tabulka tbl_group – skupiny uživatelů pro hromadnou správu práv

id	bigint	primární klíč
name	varchar(255)	název skupiny
note	varchar(255)	poznámka
owner_id	bigint	id uživatele (z tabulky tbl_user), které skupinu vytvořil

Tabulka tbl_group_permission – přiřazení práv na určitý test skupině uživatelů

id	bigint	primární klíč
note	varchar(255)	popis uživatele
owner_id	bigint	kdo uživatele vytvořil

Tabulka tbl_group_user – přiřazení uživatelů do skupiny

group_id	bigint	id skupiny v tabulce tbl_group
user_id	bigint	id uživatele v tabulce tbl_user

Další tabulky:

Tabulka tbl_token – systémová tabulka

Popis konfiguračních souborů XML

XML soubory je doporučeno vytvářet v kódování UTF-8. Hlavička obsahuje standardní definici XML včetně kódování.

Šablona scény

Kořenový element šablony se musí jmenovat SlideTemplate a má povinný atribut UID, který představuje libovolný jedinečný identifikátor. Pro zaručení jedinečnosti doporučuji vygenerovat řetězec GUID např. na
http://createguid.com/ nebo http://www.guidgenerator.com/:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SlideTemplate UID="CA442B90-6C6B-11DE-8769-03E855D89593">
</SlideTemplate>
```

Kořenový element musí obsahovat element **Viewport**, který představuje obrazovku, a dále jsou nepovinné elementy: **Windows -** pro definici vyskakovacích oken, **Timers -** pro časovače, **Variables** – pro proměnné, **Actions** – definice akcí (subrutin), **Handlers**, které umožňují nastavit obsluhu událostí, a **OutputValue** pro definování celkového výsledku scény. Element Viewport pak obsahuje elementy jednotlivých komponent, ze kterých se scéna skládá – např. Vertical Layout.

```
<SlideTemplate UID="CA442B90-6C6B-11DE-8769-03E855D89593">
 <Viewport>
  <Panel Id="1">
   ...
  </Panel>
 </Viewport>
 <Windows>
  <Window Id="w1">
   ...
  </Window>
  ...
 </Windows>
 <Timers>
  <Timer Id="t1">
   ...
  </Timer>
  ...
 </Timers>
 <Variables>
  <Variable Id="v1">
   ...
  </Variable>
  ...
 </Variables>
 <Actions>
  <Action Id="a1">
   ...
  </Action>
  ...
 </Actions>
 <Handlers>
  <Init>
```

```
    ...
   </Init>
   ...
  </Handlers>
  <OutputValue>
   …
  <OutputValue>

</SlideTemplate>
```

Vzájemná souvislost sekcí Viewport, Variables, Windows a Actions je popsána např. v kapitole Window. Souvislost mezi Viewport, Variables, Actions a OutputValue je popsána např. v kapitole ButtonPanel.

Obsah scény

Kořenový element obsahu se musí jmenovat SlideContent a má povinný atribut TemplateUID, jehož hodnota se musí shodovat s atributem UID v šabloně. Při sestavování scény jsou porovnávány hodnoty atributů UID a TemplateUID a v případě, že se neshodují, nedojde k vytvoření scény.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SlideContent TemplateUID="CA442B90-6C6B-11DE-8769-03E855D89593">
</SlideContent>
```

Kořenový element musí obsahovat element Bindings, který obsahuje 0..n elementů Bind.

```
<SlideContent TemplateUID="CA442B90-6C6B-11DE-8769-03E855D89593">
  <Bindings>
    <Bind>
      <Panel Id="1">
      ...
      </Panel>
    </Bind>
    <Bind>
      <Panel Id="2">
      ...
      </Panel>
    </Bind>
  </Bindings>
</SlideContent>
```

Element Bind obsahuje element některého z povolených typů komponent, ze kterých se skládá scéna (např. Panel v příkladu výše). Blíže jsou typy komponent popisovány v následující kapitole. Element kterékoliv z těchto komponent obsahuje atribut Id, jehož hodnota (např. „1“) představuje Id daného elementu v šabloně.

Sestavování šablony a obsahu

Při sestavování scény je propojen obsah elementu komponenty z šablony s obsahem elementu z obsahu scény do výsledného elementu pomocí shodného Id. Je tedy nutné, aby v šabloně byla definována alespoň „obálka“ komponenty s daným Id.

Pokud element dané komponenty v šabloně obsahuje elementy Properties, Actions či Layers a stejně tak příslušný element komponenty v obsahu scény, potom jsou jednotlivé subelementy Property, Action, Layer porovnávány se stejnými subelementy v XML obsahu a v případě shody identifikátoru je **hodnota ze šablony přepsána hodnotou z obsahu**. Obsah scény je tedy nadřazen šabloně. Ostatní elementy jsou do výsledného dokumentu pouze zkopírovány:

Šablona:

```
<Panel Id="3">
  <Properties>
    <Border Value="False" />
    <Width Value="80%" />
      ...
  </Properties>
  ...
</Panel>
```

Obsah:

```
<Bind>
  <Panel Id="3">
    <Properties>
      <Border Value="True" />
      <Caption Value="Toto bude vypsáno v panelu" />
      ...
    </Properties>
    ...
  </Panel>
</Bind>
```

Výsledek:

```
<Panel Id="3">
  <Properties>
    <Border Value="True" />
    <Width Value="80%" />
    <Caption Value="Toto bude vypsáno v panelu" />
    ...
  </Properties>
  ...
</Panel>
```

Vhodný postup je tedy vložit do šablony prvky, které se v podobných scénách vzájemně neliší (rozmístění a funkce tlačítek, velikost obrázků) a do obsahu scény vložit vzájemné odlišnosti mezi podobnými scénami založenými na podobné šabloně (vysvětlující texty, hodnoty tlačítek, samotné obrázky). Např. následující dva panely tlačítek budou mít společnou šablonu a obsah scény každého z nich bude obsahovat jen popisky tlačítek.

V následujících kapitolách věnujících se jednotlivým elementům jsou použity ukázky XML kódu, které mají vysvětlit nastavení jednotlivých parametrů daných elementů. V těchto ukázkách se většinou nebude rozlišovat, zda je daný kód umístěn v šabloně nebo v obsahu scény, daný parametr bude fungovat tak či tak. Je už na tvůrci konkrétního testu, zda daný parametr umístí do šablony nebo do obsahu scény. Pro tyto účely může využít ukázky kompletních šablon a obsahů obsažené jednak v dokumentu Návod_zakázka_PřF.pdf a zároveň coby připravené samostané XML dokumenty.

Typy komponent

Kontejnerové komponenty

Panel	vykreslí základní panel
VerticalLayout	nevizuální komponenta, která rozdělí obsah vertikálně
HorizontalLayout	nevizuální komponenta, která rozdělí obsah horizontálně
Window	dialogové okno
FormLayout	určeno pro umístění různých komponent – součástí formuláře

Akční komponenty

Button	jednoduché tlačítko
ButtonPanel	panel s více tlačítky
Timer	časovač (stopky)
TimerLabel	panel zobrazující čas z časovače
Image	obrázek
ComboBox	výběrový editační prvek
TextField	textové pole
TextArea	textové pole pro delší text
DateField	pole pro zadávání data
SelectPanel	panel pro výběr z několika možností
Label	popisek, text
Video	vložený obrazový záznam
Audio	vložený zvukový záznam

Plugin komponenty

Map	mapa, obrázek s možností kreslit
- ImageLayer	vrstva mapy – rastrový obrázek
- ImageSequenceLayer	vrstva mapy – postupně se střídá několik rastrových obrázků
- WMSLayer	vrstva mapy – mapa získaná pomocí Web Map Service
o Pan	element pro nastavení posunování mapy myší
o Zoom	element pro nastavení změny měřítkakolečkem myši
- FeatureLayer	vrstva mapy – možnost kreslit a definovat vektorovou geometrii
o Feature	podčást FeatureLayer – vektorový objekt

o DrawPoint	element pro nastavení kreslení bodů do FeatureLayer
o DrawPath	element pro nastavení kreslení linií do FeatureLayer
o DrawPolygon	element pro nastavení kreslení polygonů do FeatureLayer
o Style	element pro nastavení vzhledu Feature a FeatureLayer

Všechny komponenty mají různé parametry definované elementem **Properties** obsahujícím jednotlivé elementy vlastností.

Kontejnerové komponenty

Komponenty typu Panel, VerticalLayout, HorizontalLayout a Window fungují jako tzv. kontejnery, takže mohou obsahovat další vnořené komponenty. Pro vložení vnořených komponent se používá element **Components**.

Panel

Panel lze považovat za základní komponentu, která slouží pro zobrazení HTML obsahu nebo jako kontejner pro další komponenty.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Border	True/False	rámeček	True
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Caption	text	nadpis	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc

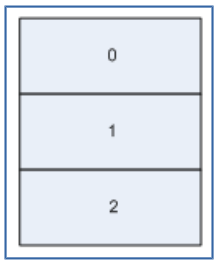
Ukázka:
Na obrázku jsou tři Panely. Do nich je vložen text ve formě komponenty Label, která bude probírána dále. Nastavení šířky na 70% způsobí, že panely bude zabírat jen část z šířky scény. První Panel má definován nadpis, druhý má vypnutý rámeček, třetí má všechny hodnoty ponechány implicitní – bez nadpisu a s rámečkem.

```
<Panel Id="1">
  <Properties>
    <Width Value="70%" />
    <Caption Value="Nadpis panelu se zobrazí nad ním" />
  </Properties>
</Panel>
<Panel Id="2">
  <Properties>
    <Width Value="55%" />
    <Border Value="False" />
    <Caption Value="Nadpis panelu se zobrazí nad ním" />
  </Properties>
</Panel>
<Panel Id="3">
  <Properties>
    <Width Value="55%" />
  </Properties>
</Panel>
```

Nadpis panelu se objeví nad ním
1. Problémy řeší rychle.
2. Projevuje pokročilé usuzovací schopnosti.
3. Má pronikavé myšlení, problémy chápe intuitivně.

VerticalLayout

Rozděluje obsah vertikálně a slouží jen jako kontejner



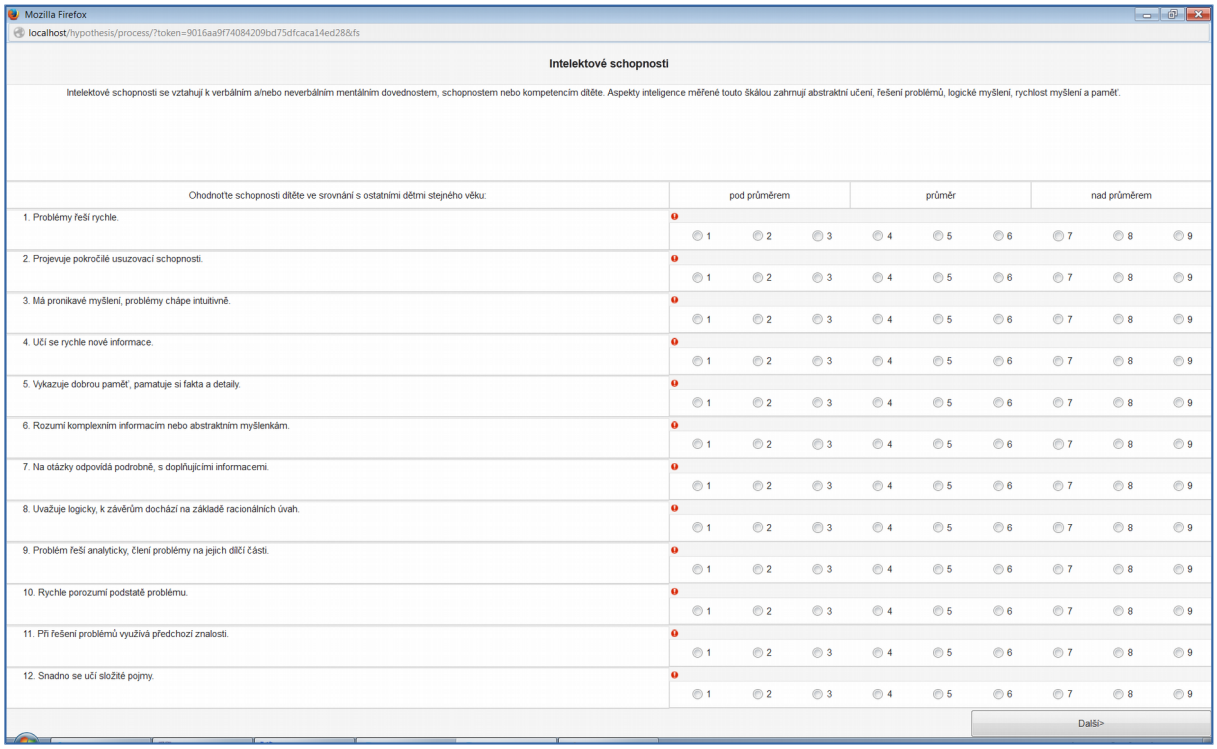
Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Width	text (100%, 230px, 230)	Šířka. V případě procent jde o relativní šířku z šířky nadřazené komponenty.	
Height	text (jako Width)	Výška. V případě procent jde o relativnívýšku z výšky nadřazené komponenty.	

Ukázka:
Základní komponentou této scény je VerticalLayout, který obsahuje několik komponent typu Panel, HorizontalLayout a dalších komponent, které jsou řazeny pod sebou – viz následující obrázek. VerticalLayout má nastaveny vlastnosti Height i Width na 100%, zabírá tedy celou scénu. Vnořené komponenty mají šířku rovněž nastavenu na 100%, ale výšky mají nastavené podle toho, jakou část scény mají zabírat.

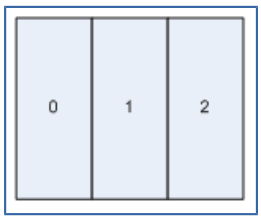
```
<VerticalLayout Id="vl1">
  <Properties>
    <Width Value="100%" />
    <Height Value="100%" />
  </Properties>
  <Components>
    <Panel Id="p1">
      ..
      <Height Value="6%" />
      ..
    </ Panel >
    <HorizontalLayout Id="hl1">
```

..
<Height Value="14%" />
..
</ HorizontalLayout >
..
</Components>
</**VerticalLayout**>



HorizontalLayout

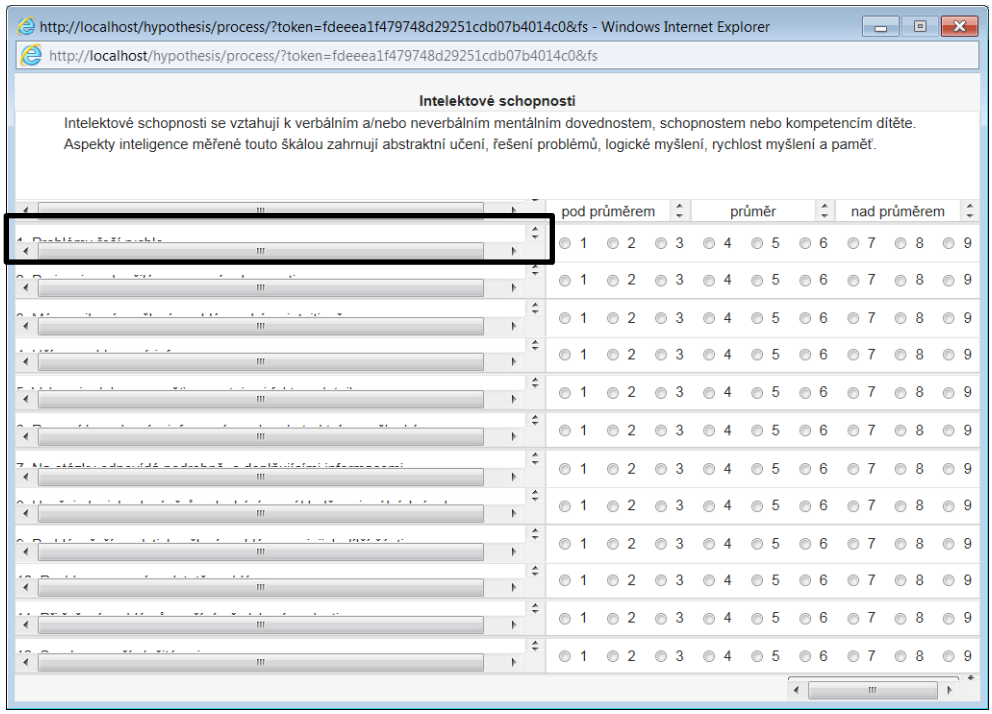
Rozděluje obsah horizontálně.



Parametry: shodné jako VerticalLayout.

Využití VerticalLayoutu a HorizontalLayoutu

Horizontal a VerticalLayout lze s výhodou použít i v případech, kdy nejsou nezbytně potřeba. Zabudované vlastnosti těchto kontejnerů ovlivňují zobrazení komponent v nich obsažených v některých zvláštních příležitostech – např. při nedostatku místa na obrazovce. Lze tak zlepšit zobrazování komponent v některých prohlížečích. Na následujícím obrázku je ukázka, kdy se nevhodně automaticky zobrazují posuvníky a zakrývají tak velkou část komponent.



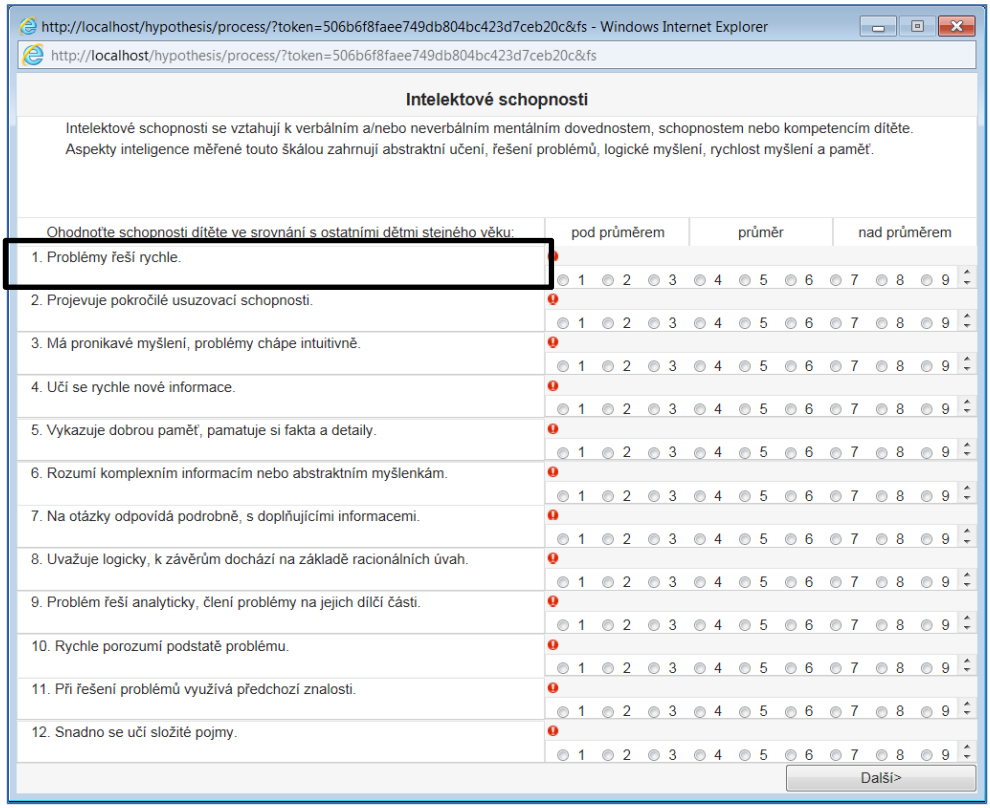
Následující kód je částí šablony, která popisuje část scény výše vyznačenou obdélníkem. Panel p1obsahuje text Label q1. Ten má nastavenu šířku 95% nadřazeného prvku, aby se text s rezervou vešel do panelu, nepřesáhl za jeho hranice. Přesto se ve výsledku objevil u panelu nevyžádaný posuvník, který zbytečně zakrývá většinu panelu a způsobuje nečitelnost textu.

```
<Panel Id="p1">
 <Properties>
 …
 </Properties>
 <Components>
  <Label Id="q1">
   <Properties>
    <Width Value="95%" />
   </Properties>
  </Label>
 </Components>
</Panel>
```

Řešením je vložit do panelu p1 nejprve VerticalLayout a teprve do něj Label q1 s textem. VerticalLayout má nastavenu šířku i výšku na 100% rozměru panelu, takže se vizuálně nijak neprojeví. Jeho vložení však zajistí, že se nevhodný posuvník nezobrazí. Jestliže použijeme

tuto úpravu na všechny panely scény, bude scéna vypadat jako na následujícím obrázku. HorizontalLayout by měl v tomto případě fungovat rovněž.

```
<Panel Id="p1">
 <Properties>
 …
 </Properties>
 <Components>
  <VerticalLayout Id="vl1 ">
   <Properties>
    <Width Value="100%" />
    <Height Value="100%" />
   </Properties>
   <Components>
    <Label Id="q1">
     <Properties>
      <Width Value="95%" />
     </Properties>
    </Label>
   </Components>
  </VerticalLayout>
 </Components>
</Panel>
```



FormLayout

Tato komponenta je kontejnerem určeným, aby se do ní umisťovaly různé další komponenty tvořící zejména např. formulář – tlačítka (Button) a jejich skupiny (ButtonPanel), textová pole (TextArea, TextField), popisky (Label) apod. Umístění všech těchto různých prvků, kterých bývá např. v dotaznících velké množství, do jedné kontejnerové komponenty umožní programu jejich účelnější rozmístění na slide.

K témuž účelu lze využít např. i VectorLayout nebo HorizontalLayout. FormLayout má velmi podobné parametry, ale je možné mu např. nadefinovat ohraničení linií. To umožňuje vizuálně více seskupit prvky (např. popisek a kolonku), které patří tématicky k sobě a usnadnit tak uživateli orientaci ve formuláři.

Název	Povolené hodnoty	Význam	Výchozí hodnota
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Border	True/False	rámeček	False
Width	text (100%, 230px, 230)	Šířka. V případě procent jde o relativní šířku z šířky nadřazené komponenty.	
Height	text (jako Width)	Výška. V případě procent jde o relativnívýšku z výšky nadřazené komponenty.	

Ukázka:
Následující kód ukazuje větší množství komponent TextField a ComboBox uložené do jednoho elementu FormLayout.

```
<FormLayout Id="ct001_ct002_ct001">
  <Properties>
    <Border Value="True" />
    <Width Value="80%" />
    <Height Value="100%" />
    <Alignment Value="mc" />
  </Properties>
  <Components>
    <TextField Id="field_name">
      …
    </TextField>
    <TextField Id="field_age">
      …
```

```
    </TextField>

    <ComboBox Id="field_graduate">
      …
    </ComboBox>
    <ComboBox Id="field_employment">
      …
    </ComboBox>
    <TextField Id="field_email">
      …
    </TextField>
  </Components>
</FormLayout>
```

Window

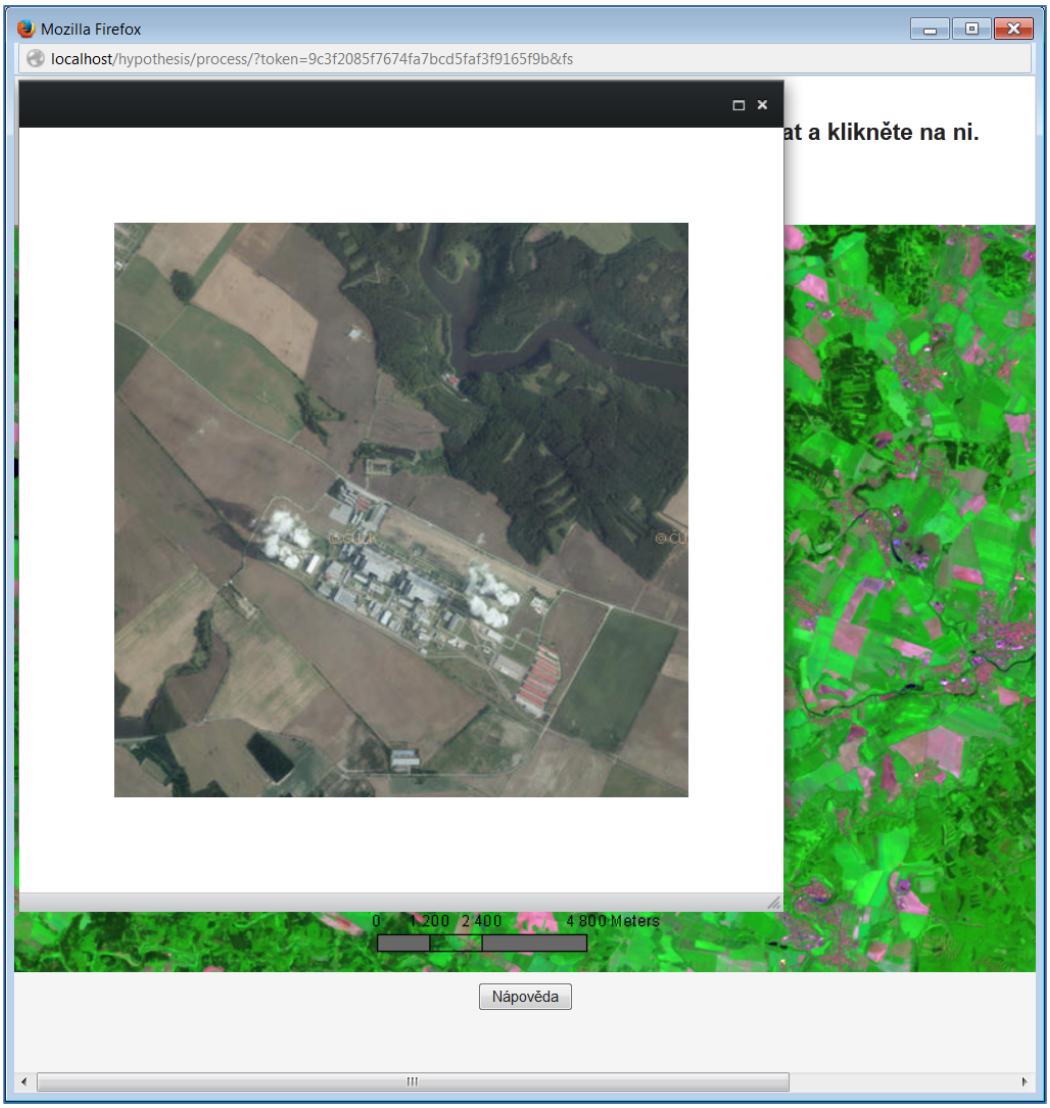
Vytváří dialogové okno, které slouží jako kontejner. Může sloužit např. pro zobrazení nápovědy k příslušné scéně na vyžádání.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Caption	text	nadpis	

Ukázka:
Definice samotného okna v elementu Window je podobná definicím ostatních komponent. Zde obsahuje jen HorizontalLayout s obrázkem. Ukázka okna zobrazeného nad scénou s tlačítkem je na následujícím obrázku.

```
<Window Id="help_window">
  <Properties>
    <Width Value="950" />
    <Height Value="400" />
  </Properties>
  <Components>
    <HorizontalLayout Id="ct002_ct001">
      …
      <Components>
        <Image Id="help_image" />
      </Components>
    </HorizontalLayout>
  </Components>
</Window>
```



Vyvolání a zavírání okna

Okno může být definováno v šabloně v elementu Windows. Může být vyvoláno např. kliknutím na tlačítko Nápověda typu Button, které spouští akci „showHelp“ nadefinovanou v elementu Actions. Ta obsahuje příkaz „napoveda->open()“ k otevření okna, přičemž „napoveda“ je proměnná definovaná elementu Variables, která odkazuje na objekt typu Window s id „help_window“. Okno se zavírá stejně jako jiná okna operačního systému (např. křížkem vpravo nahoře), nemusí být tedy definována žádná speciální akce nebo tlačítko na jeho zavření.

```
<SlideTemplate UID="CA442B90-6C6B-11DE-8769-03E855D89593">
  <Viewport>
    …
    <Button Id="ct001_ct003_ct003_ct001">
      <Properties>
        <Caption Value="Nápověda" />
      </Properties>
      <Handlers>
        <Click>
          <Call Action="showHelp" />
        </Click>
      </Handlers>
    </Button>
```

```
…
</Viewport>
<Windows>
  <Window Id="help_window">
    ...
  </Window>
</Windows>
<Variables>
  <Variable Id="napoveda" Type="Object">
    <Reference>
      <Window Id="help_window" />
    </Reference>
  </Variable>
</Variables>
<Actions>
  <Action Id="showHelp">
    <Expression>napoveda->open()</Expression>
  </Action>
</Actions>
</SlideTemplate>
```

Akční komponenty

Dosud zmíněné komponenty byly komponenty kontejnerové – slouží pro umístění jiných komponent. Komponenty typu Button, ButtonPanel, Map, Image, Timer a další jsou tzv. akční komponenty. Vyznačují se tím, že mají obsluhu událostí, které mohou být využity k nastavení chování testu při interakci s pokusnou osobou.

Událost (klik, přehrání videa, načtení nového slide apod.) je zachycena pomocí speciálních elementů. Příklad takového elementu Click je např. v ukázkovém kódu v kapitole Window. Každý typ komponenty může zachycovat určité typy událostí. V následujících kapitolách bude u každého typu komponenty uvedeno, jaké typy událostí může zachycovat.

Zachycovací element pak většinou volá některý element Action. Ten obsahuje kód, který definuje reakci programu na danou událost (např. kliknutí může způsobit start videa. V kódu definujícím reakci je možné využít následující metody:

Metoda	Význam
Navigator->next()	Přejít na další scénu. Navigator je speciální proměnná vytvářená ve scéně.
ComponentData->getButtonIndex()	Vrátit pořadí vybraného tlačítka ze skupiny tlačítek (ButtonPanel). První tlačítko má pořadí 1. ComponentData je proměnná automaticky referencovaná v Handleru.
ComponentData->getCoordinate()->x	získat souřadnici x kliku v mapě
ComponentData->getCoordinate()->y	získat souřadnici y kliku v mapě
X->start()	spustit časovač (Timer) s id hodnotou X
X->stop()	Zastavit časovač (Timer) s id hodnotou X. Pokud není zastaven, končí časovač automaticky po uplynutí nastaveného času.
X->activate()	aktivovat mapový ovládací prvek (Control) s id hodnotou X
X->deactivate()	deaktivovat mapový ovládací prvek (Control) s id hodnotou X
X->open()	zobrazit dialog (Window) s id hodnotou X
X->close()	zavřít dialog (Window) s id hodnotou X
X->play()	spustit video (Video) nebo zvuk (Audio) s id hodnotou X
X->pause()	zastavit video (Video) nebo zvuk (Audio) s id hodnotou X
X->mask()	zakrýt mapu (Map), obrázek (Image), video (Video) nebo i některé typy kontejnerů (HorizontalLayout a VerticalLayout) s id hodnotou X maskou - neprůhlednou šedivou vrstvou
X->unmask()	odstranit masku z mapy (Map), obrázku (Image), videa (Video) a některých typů kontejnerů (HorizontalLayout a VerticalLayout) s id hodnotou X

X->getTilesCount()	zjistit počet obrázků v ImageSequenceLayer s id hodnotou X
X->getTileIndex()	zjistit pořadí aktuálního obrázku v rámci ImageSequenceLayer s id hodnotou X
X->nextTile()	přejít na další obrázek v ImageSequenceLayer s id hodnotou X
X->setHidden(true/false)	nastavit prvek mapové vrstvy (Feature) s id hodnotou X na neviditelný/viditelný

Metody, které mají ve výrazu X, se spouští nad prvkem s id X. Tento příkaz může být vykonán pouze tehdy, když pod daným id existuje komponenta příslušného typu:
start(), stop() – Timer
activate(), deactivate() – Control – element umožňující kreslení prvků v mapě (Map)
open(), close() – Window
play(), pause() – Video, Audio
mask(), unmask() – Map, Image, Video, dokonce i HorizontalLayuot a VerticalLayout
setHidden(true/false) – Feature
getTilesCount(), getTileIndex(), nextTile() - ImageSequenceLayer

Události:

Komponenta	Event	Význam
Celý slide	Show	zobrazení slide
Celý slide	Init	sestavení slide

Button

Vytvoří tlačítko. Kromě parametrů Properties má tato komponenta definovánu i událost (Event), na kterou reaguje (klik na tlačítko).

Parametry:

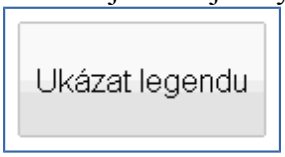
Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Caption	text	nadpis	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc

Povinné: Caption

Události:

Event	Význam
Click	stisknutí tlačítka

Ukázka:
Tlačítko typu Button může sloužit např. pro zobrazení legendy. Tlačítko má definovánu velikost (100 x 30 pixelů) a popisek. V elementu Handlers má nadefinovano, že reaguje na událost typu Click – kliknutí. Spouští akci „showLegend“ nadefinovanou v elementu Actions. Ta obsahuje příkaz „legend->open()“ k otevření okna, přičemž „legend“ je proměnná definovaná elementu Variables, která odkazuje na objekt typu Window s id „legend_window“.




```
<SlideTemplate UID="CA442B90-5C5B-11DE-8769-03E855D89593">
  <Viewport>
    …
    <Button Id="b1">
      <Properties>
        <Caption Value="Ukázat legendu"/>
        <Width Value="100" />
        <Height Value="30" />
      </Properties>
      <Handlers>
        <Click>
          <Call Action="showLegend" />
```

```
      </Click>
    </Handlers>
  </Button>
 …
 </Viewport>
 <Windows>
  <Window Id="legend_window">
    ...
  </Window>
 </Windows>
 <Variables>
  <Variable Id="legend" Type="Object">
    <Reference>
      <Window Id="legend_window" />
    </Reference>
  </Variable>
 </Variables>
 <Actions>
  <Action Id="showLegend">
    <Expression>legend->open()</Expression>
  </Action>
 </Actions>
</SlideTemplate>
```

ButtonPanel

Vytvoří skupinu tlačítek, ze kterých lze vždy zvolit jedinou hodnotu. Tlačítka mají určené pořadí, lze tedy zjistit, které bylo zvoleno.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Orientation	Horizontal, Vertical	orientace	Horizontal
Width	text (100%, 230px, 230)	šířka panelu	
Height	text (jako Width)	výška panelu	
ChildWidth	jako Width	výška tlačítek	
ChildHeight	jako Height	šířka tlačítek	
Captions	pole textů	popisy tlačítek	

Povinné: Captions

Události:

Event	Význam
Click	stisknutí tlačítka

Ukázka:
Kód definuje vzhled a funkcionalitu panelu na následujícím obrázku. Parametry Width a Height patří celému panelu – zabírá jen 60% šířky, kterou má k dispozici. ChildWidth a ChildHeight patří jednotlivým tlačítkům – tlačítka zabírají 100% výšky panelu, ale jen 90% šířky panelu, přičemž volný prostor se rovnoměrně rozdělí. Tímto způsobem lze dosáhnout mezer mezi tlačítky panelu.

Kromě vzhledu je v kódu šablony definována i funkcionalita v elementu Handlers. Všechna tlačítka reaguje na kliknutí, ale každé reprezentuje jinou hodnotu. Funkce getButtonIndex() proto zjistí, které tlačítko bylo zvoleno, uloží jeho pořadí do proměnné „buttonIndex“ a zavolá akci „vyber_tlacitkem“.

```
<ButtonPanel Id="vyber">
  <Properties>
    <Width Value="60%" />
    <Height Value="100%" />
    <ChildWidth Value="90%" />
    <ChildHeight Value="100%" />
  </Properties>
  <Handlers>
    <Click>
      <Expression>buttonIndex=ComponentData->getButtonIndex()</Expression>
      <Call Action="vyber_tlacitkem"/>
```

```
  </Click>
 </Handlers>
</ButtonPanel>
```

Pokud by se parametr ChildWidth změnil na 100%, pak by se vzhled panelu změnil takto:

Mezi properties nejsou uvedeny popisy tlačítek Caption. Ty jsou definovány v obsahu scény, protože se budou scéna od scény lišit. Spolu s nimi je zde mezi jiným také nastavena proměnná „spravnyVyber“, určující, volba kterého tlačítka je správná odpověď. Kód pro obojí může v obsahu scény vypadat např. takto.

```
<Bind>
 <ButtonPanel Id="vyber">
  <Properties>
   <Captions Value="'3','5','7','žádné'" />
  </Properties>
 </ButtonPanel>
</Bind>
<Bind>
 <Variable Id="spravnyVyber" Value="1"/>
</Bind>
```

Následující ukázka kódu šablony vysvětluje, jak se dále pracuje s pořadím tlačítka uloženým do proměnné „buttonIndex“. V rámci akce „vyber_tlacitkem“ je tato hodnota srovnána s proměnnou „spravnyVyber“ získanou z obsahu scény. Pokud jsou obě hodnoty shodné, pak je proměnná „vysledek“ nastavena na hodnotu 1. Původně jsou všechny proměnné nastaveny na hodnotu 0 – viz jejich definice v elementu Variables.

Ať už je proměnná „výsledek“ nastavena na hodnotu 0 nebo 1, je zavolána akce „nextSlide“. Její kód zajistí přechod na následující scénu. Proměnná „výsledek“ je nastavena coby výsledek scény (element OutputValue). Bude tedy po skončení scény zapsána do databáze do tabulky ***tbl_slide_output*** do sloupce Output.

```
<Variables>
 <Variable Id="vysledek" Type="Integer" Value="0" />
 <Variable Id="buttonIndex" Type="Integer" Value="0" />
 <Variable Id="spravnyVyber" Type="Integer" Value="0" />
</Variables>

<Actions>
 <Action Id="vyber_tlacitkem">
  <If>
   <Expression>buttonIndex==spravnyVyber</Expression>
   <True>
    <Expression>vysledek=1</Expression>
   </True>
  </If>
  <Call Action="nextSlide" />
 </Action>

 <Action Id="nextSlide">
  <Expression>Navigator->next()</Expression>
```

```
  </Action>
</Actions>

<OutputValue>
  <Expression>vysledek</Expression>
</OutputValue>
```

Timer

Vytvoří časovač, který slouží k měření času u některých typů úloh. Čas je měřen v milisekundách.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Time	číslo	čas v milisekundách	
Direction	up, down	čas se bude připočítávat od 0 nebo odpočítávat k 0	up

Povinné: Time

Události:

Event	Význam
Start	spuštění měření času
Stop	skončení měření času
Update	pravidelné opakování během chodu časovače

Ukázka:
Časovač se v šabloně scény definuje v elementu Timers. V ukázce dole je definován Timer s názvem „timer1“ a s hodnotou 10000 ms. Časovač se spustí akcí „startTimer1“, jakmile se scéna objeví (Handler Show). Handler Update spustí každou vteřinu akci „pricti“, která zvyšuje proměnnou x o 1. Jakmile časovač doběhne, je spuštěna akce „nextSlide“ – přejde se na další scénu. Samotný timer1 je uložen do stejnojmenné proměnné.

```
...
<Timers>
  <Timer Id="timer1">
    <Properties>
      <Time Value="10000" />
    </Properties>
    <Handlers>
      <Stop>
        <Call Action="nextSlide" />
      </Stop>
      <Update Interval="1000" />
        <Call Action="nextSlide" />
      </Update>
    </Handlers>
  </Timer>
</Timers>
…
<Variable Id="timer1" Type="Object">
  <Reference>
    <Timer Id="timer1" />
  </Reference>
</Variable>
<Variable Id="x" Type="Integer" Value="0" />
```

```
…
<Actions>
  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
  <Action Id="startTimer1">
    <Expression>timer1->start()</Expression>
  </Action>
  <Action Id="pricti">
    <Expression>x=x+1</Expression>
  </Action>
</Actions>

<Handlers>
  <Show>
    <Call Action="startTimer1" />
  </Show>
</Handlers>
```

TimerLabel

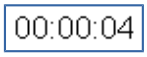
Zobrazí popisek zobrazující čas některého Timeru. Samotný Timer netvoří žádný viditelný prvek, běží neviditelně na pozadí. TimerLabel umožňuje chod času zviditelnit.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	Šířka panelu	
Height	text (jako Width)	Výška panelu	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
TimeFormat	např. ["HH:mm:ss"]	Způsob zápisu času.	včetně desetin vteřiny ["ss.S"]
TimerId	text	Id Timeru, jehož čas se bude zobrazovat.	

Ukázka:
TimerLabel se vlastnostmi podobá prvku Label (viz kapitola o tomto elementu). Lze ho vložit do libovolného kontejneru. Properties se liší jen tím, že je nutné mu přiřadit některý Timer z těch, které jsou definovány v elementu Timers. Implicitně je čas zobrazován včetně desetin vteřiny. Formát je možno přenastavit.

```
<TimerLabel Id="timerlabel">
  <Properties>
    <Width Value="90%" />
    <TimerId Value="odpocet"/>
    <TimeFormat Value=["HH:mm:ss"] />
  </Properties>
</TimerLabel>
```



Image

Vloží do scény obrázek, který je uložen na dostupné URL adrese. Jestliže je adresa nedostupná, pak se scéna vytvoří bez obrázku. Vkládat lze většinu obecně známých formátů. S využitím formátu animovaný GIF lze pomocí tohoto elementu vložit do testu dokonce i animaci.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Url	text	url adresa obrázku	

Povinné: Url

Události:

Event	Význam
Load	načtení obrázku
Click	kliknutí do obrázku

Ukázka:
U obrázku je vhodné nastavit šířku a výšku přímo v pixelech. Nastavení relativní může způsobit deformaci obrázku.

```
<Image Id="im1">
  <Properties>
    <Width Value="950" />
    <Height Value="589" />
    <Url Value="http://www.imagehosting.cz/images/divadlo.jpg" />
  </Properties>
</Image>
```

Nestandardní chování prohlížečů při nastavení velikosti obrázku
Test lze v každém internetovém prohlížeči spouštět různými způsoby. Více se o tomto lze dočíst v poslední kapitole dokumentu Návod_zakázka_PřF.pdf. V případě, že je test spuštěn standardním způsobem (jen kliknutím myší na tlačítko Spustit), pak nenastávají žádné problémy s deformováním obrázků.

Pokud však uživatel spustí test ve vyskakovacím okně prohlížeče (se stisknutou klávesou Alt), pak jediným prohlížečem, který zajistí zobrazení obrázků ve správné velikosti je Google Chrome. Mozilla Firefox a Internet Explorer mohou změnit velikost zobrazených obrázků. Např. na větších monitorech se obrázek často roztáhne. A to i v případě, že má nastavenu přesnou velikost v pixelech! To může způsobit problémy např. při odečítání souřadnic při kliku myší.

ComboBox

Vytvoří rolovací editační prvek s výběrem hodnot. Může být nastaven validátor kontrolující správnost vyplnění.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Caption	text	nadpis	
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc

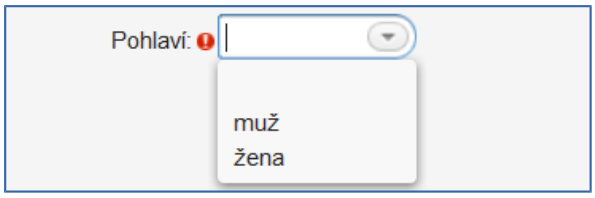
Validátory:

Název	Význam
Empty	ComboBox nesmí zůstat prázdný.

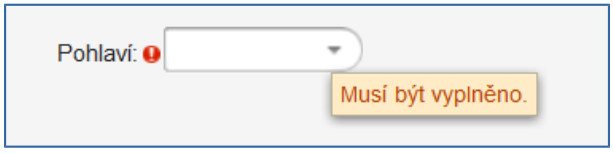
Další povinné elementy mimo Properties:
Items – Komponenta typu ComboBox musí obsahovat výběrové hodnoty, které jsou definovány elementem Items a jeho podelementy Item.

Ukázka:
Následující kód vytváří ComboBox s relativní výškou a pevně v pixelech nastavenou šířkou. To je výhodné, pokud si chceme být jistí, že bude rolovátko na jakémkoliv displeji dostatečně široké, aby všechny nabízené hodnoty byly čitelné. Nadpis se automaticky umisťuje před rolovátko.

```
<ComboBox Id="combo1">
  <Properties>
    <Caption Value="Pohlaví:" />
    <Width Value="100" />
    <Height Value="10%" />
  </Properties>
  <Items>
    <Item Value="1" Caption="muž" />
    <Item Value="2" Caption="žena" />
  </Items>
  <Validators>
    <Empty>
      <Message>Musí být vyplněno.</Message>
    </Empty>
  </Validators>
</ComboBox>
```



Nastavený Validator „Empty“ zajišťuje, že scénu nelze opustit, dokud není položka vyplněna. Červený vykřičník u položky symbolizuje její povinné vyplnění. Při přejetí myší se objeví zpráva z elementu Message.



Uživatelem zvolený údaj z rolovátka se v databázi ukládá do tabulky ***tbl_slide_output*** do sloupce Xml_data.

TextField

Vytvoří textové pole pro zadání krátkého textu. Text se nezalamuje.

Parametry:

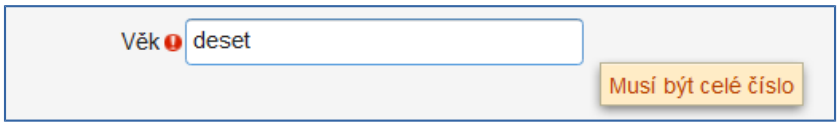
Název	Povolené hodnoty	Význam	Výchozí hodnota
Caption	text	nadpis	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	

Validátory:

Název	Význam
Empty	ComboBox nesmí zůstat prázdný.
Integer	Hodnota musí být celé číslo.
Number	Hodnota musí být číslo.
Range	Hodnota musí být číslo v určitém rozsahu.

Ukázka:
V této ukázce je kombinováno víc validátorů současně. Splněny musí být všechny současně.

```
<TextField Id="textfield1">
  <Properties>
    <Width Value="70%" />
    <Height Value="10%" />
    <Caption Value="Věk" />
  </Properties>
  <Validators>
    <Range>
      <Message>Musí být mezi 15 a 100.</Message>
      <Min Value="15" />
      <Max Value="100" />
    </Range>
    <Empty>
      <Message>Musí být vyplněno.</Message>
    </Empty>
    <Integer>
      <Message>Musí být celé číslo.</Message>
    </Integer>
  </Validators>
</TextField>
```



Uživatelem vyplněný údaj z pole se v databázi ukládá do tabulky ***tbl_slide_output*** do sloupce Xml_data.

TextArea

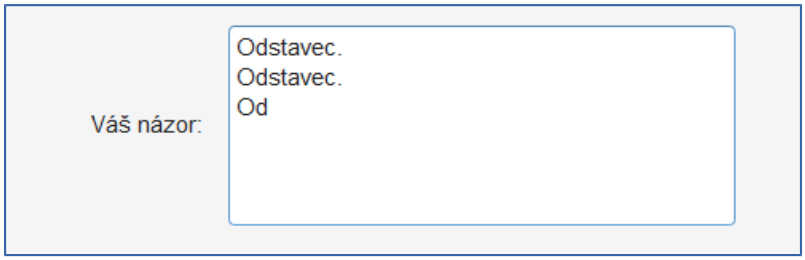
Vytvoří textové pole pro zadání delšího textu. Text se v poli zalamuje.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Caption	text	nadpis	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	

Ukázka:

```
<TextArea Id="textarea1">
  <Properties>
    <Width Value="70%" />
    <Height Value="10%" />
    <Caption Value="Váš názor:"/>
  </Properties>
</TextArea>
```



Uživatelem vyplněný údaj se v databázi ukládá do tabulky ***tbl_slide_output*** do sloupce Xml_data.

DateField

Vytvoří kolonku pro zadání data. Po rozkliknutí jejího pravého okraje se otevře předpřipravené okno pro výběr data – viz obrázek níže. Uživatelem zvolený údaj se v databázi ukládá do tabulky ***tbl_slide_output*** do sloupce Xml_data. Do tabulky se datum ukládá a do kódu zapisuje ve formátu rrrr-mm-dd (např. 2015-02-03).

Parametry:

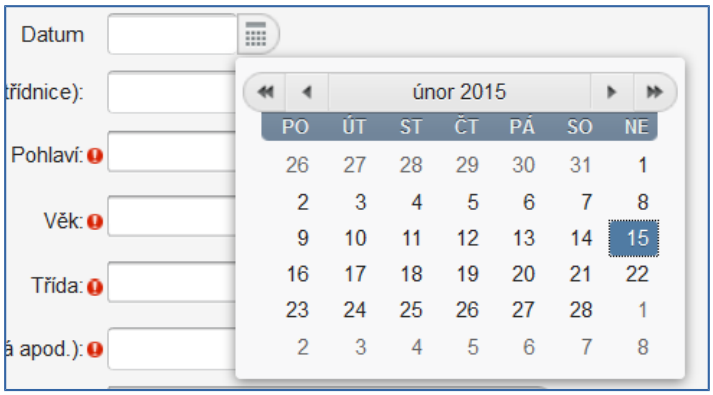
Název	Povolené hodnoty	Význam	Výchozí hodnota
Caption	text	nadpis	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	

Validátory:

Název	Význam
Empty	ComboBox nesmí zůstat prázdný.
Range	Hodnota musí být číslo v určitém rozsahu. Min a max value se zadávají ve tvaru rrrr-mm-dd.

Ukázka:
V kódu je definován validátor rozsahu, ale není definován validátor kontrolující vyplnění položky. Prázdná položka tedy nevadí, ale zadání data mimo požadovaný rozsah způsobí zobrazení vykřičníku.

```
<DateField Id="date_1">
  <Properties>
    <Caption Value="Datum" />
  </Properties>
  <Validators>
    <Range>
      <Message>Vybrané datum musí být z roku 2014.</Message>
      <Min Value="2014-01-01" />
      <Max Value="2014-12-31" />
    </Range>
  </Validators>
</DateField>
```


Datum
třídnice):
Pohlaví:
Věk:
Třída:
á apod.):
únor 2015
PO ÚT ST ČT PÁ SO NE
26 27 28 29 30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 1
2 3 4 5 6 7 8


Datum
15.2.15

SelectPanel

Vytvoří panel pro výběr z více možností. Podle nastavení parametru MultiSelect je možné zvolit jen jednu nebo více hodnot.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Caption	text	nadpis	
Captions	text	popis jednotlivých položek	
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc
Orientation	Horizontal, Vertical	orientace	Horizontal
MultiSelect	True, False	Je možné vybrat více hodnot?	False

Události:

Event	Význam
Click	výběr některé hodnoty kliknutím

Validátory:

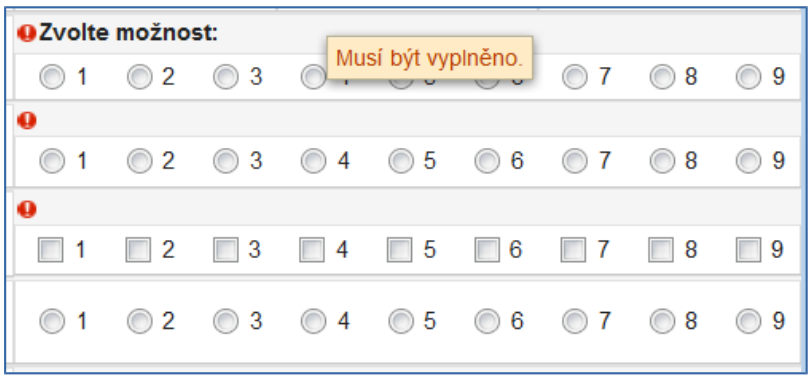
Název	Význam
Empty	Položka musí být vybrána.

Ukázka:
Tento kód definuje první ze čtyř SelectPanelů na následujícím obrázku. Je nastaven nadpis i validátor kontrolující, že nějaká hodnota byla vybrána.

```
<SelectPanel Id="rb1">
  <Properties>
    <Caption Value="Zvolte možnost" />
    <MultiSelect Value="False" />
    <Width Value="45%" />
    <Height Value="100%" />
    <Captions Value="'1','2','3','4','5','6','7','8','9'" />
  </Properties>
  <Validators>
    <Empty>
      <Message>Musí být vyplněno.</Message>
    </Empty>
  </Validators>
```

</SelectPanel>

Druhý SelectPanel se liší tím, že nemá nadpis – není definován parametr Caption. Třetí SelectPanel povoluje zaškrtnout i více než jednu hodnotu – parametr MultiSelect je nastaven na True. Poslední SelectPanel nemá definován Validator – není tedy povinné vybrat nějakou hodnotu, což symbolizuje neexistence červeného vykřičníku.



Label

Vytvoří textovou položku bez rámečku. Může být použita pro vložení libovolného textu do libovolného kontejneru.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Caption	text	Samotný text.	
Width	text (100%, 230px, 230)	Šířka. Pokud je parametr uveden, text se bude zalamovat.	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc

Ukázka:
Často se komponenta Label definuje v šabloně scény pouhou obálkou – prázdným elementem Label nebo se v elementu nadefinuje ještě nějaká vlastnost – např. šířka.

```
<Label Id="title">
</Label>
…
<Label Id="text1">
  <Properties>
    <Width Value="80%" />
  </Properties>
</Label>
```

Samotný text definovaný parametrem Caption se totiž skoro vždy liší scéna od scény a musí tedy být definován v obsahu scény.

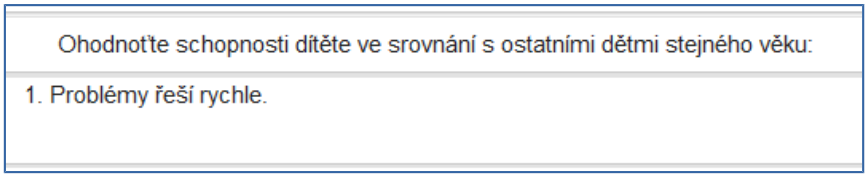
```
<Bind>
  <Label Id="title">
    <Properties>
      <Caption Value="&lt;h2&gt;Gifted Rating Scales – Pfeiffer and
Jarosewich&lt;/h2&gt;&lt;h3&gt;Posuzovací škály pro identifikaci nadání&lt;/h3&gt;" />
    </Properties>
  </Label>
</Bind>
<Bind>
  <Label Id="text1">
    <Properties>
      <Caption Value="&lt;p&gt;Dobrý den,&lt;/p&gt;&lt;p&gt; moc Vám děkujeme za Váš zájem vyplnit
posuzovací škálu GRS, která je určena k identifikaci nadání.&lt;/p&gt;" />
    </Properties>
  </Label>
</Bind>
```

Na následujícím obrázku je vidět, jak vypadá text, je li nedostatek místa. Label, kterému je nadefinován parametr Width=90% se zalamuje a je odsazen od okrajů svého panelu. Label, který vůbec parametr Width nemá, se nezalamuje, ani když pro něj není dostatek místa. Z tohoto důvodu je dobré definovat všem komponentám Label parametr Width, i kdyby měl být nastaven na 100%. Text tak sice nebude odsazen od okraj, ale bude umožněno jeho zalamování na menších displejích.

Obrázek také dokumentuje, jak je možné nastavit vzhled písma. Komponenta Label nepodporuje určování velikosti nebo typu písma. Proto je potřeba přímo v textu použít **elementy** z jazyka **HTML**. Např. velikost písma lze nastavit elementy <h1> až <h7>. První řádek je zapsán v elementu <h2>, druhý v elementu <h3>. Zbytek textu zase používá elementy <p> pro tvorbu odstavců.

Protože se v kódu šablony používají také znaky < a >, nesmí se tyto znaky používat v textu parametru Caption. Proto se musí elementy jazyka HTML použité pro formátování textu zapisovat pomocí náhradních znaků. Např. znak < se zapisuje jako < a znak > jako > Pak tedy <h2>abc</h2> se napíše jako <h2> abc</h2>

Nestandardní chování při zarovnávání
Komponenta Label vychází ze stejnojmenné třídy frameworku Vaadin 7. V některých případech se vlivem svého původního kódu chová jinak, než tvůrce testu očekává. Drobné obtíže mohou např. nastat u zarovnávání textů. Příklad je na následujícím obrázku. První řádek tvoří Label bez definovaného parametru Width, druhý řádek je Label s parametrem Width. Žádný z obou Labelů nemá definován parametr Alignment. Přesto je první řádek automaticky zarovnán na střed, zatímco druhý řádek je zarovnán doleva. Existence či neexistence parametru Width tedy ovlivňuje zarovnání textu!



Video

Video v různých formátech. Povolené jsou MP4, OGV a WEBM. Různé internetové prohlížeče však podporují jen určité formáty. Jediný formát podporovaný všemi významnějšími prohlížeči a tedy doporučeený pro použící je MP4. Více o podpoře formátů lze nalézt v dokumentu Návod_zakázka_MU.pdf na str. 48. Animaci lze do testu vložit i bez využití elementu Video. Animaci ve formátu animovaný GIF lze vložit pomocí elementu Image.

Video nemá ovládací panel, ale lze nadefinovat reakci na kliknutí myší. Umístit v kódu spouštění do Handleru Load zajistí, že se video spustí, až bude kompletně nahrána v paměti.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	Šířka. Pokud je parametr uveden, text se bude zalamovat.	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc

Události:

Event	Význam
Load	nahrání videa
Start	spuštění videa
Stop	ukončení videa
Click	kliknutí

Další povinné elementy mimo Properties:
Sources – adresy na video. Který z uvedených formátů je vybrán si určují internetové prohlížeče.

```
<Sources>
  <Source Url="..."/>
  <Source Url="..."/>
</Sources>
```

Ukázka:
V následujícím kódu z obsahu scény ukázce má element Video nadefinovánu přesnou velikost v pixelech. Tamtéž jsou uvedeny i adresy videa, která se má spustit.

<Bind>

```
  <Video Id="video">
    <Properties>
        <Width Value="854px" />
        <Height Value="480px" />
    </Properties>
    <Sources>
      <Source Url="http://media.w3.org/2010/05/sintel/trailer.mp4" />
      <Source Url="http://media.w3.org/2010/05/sintel/trailer.ogv" />
      <Source Url="http://media.w3.org/2010/05/sintel/trailer.webm" />
    </Sources>
  </Video>
</Bind>
```

V šabloně je video vloženo do kontejneru HorizontalLayout, který má nadefinovanou výšku na 70%. Pro případ, že by v obsahu scény nebyla velikost definována. Jsou definovány tři Handlery. Load je spuštěn po nahrání videa do paměti, Click reaguje na kliknutí a Stop spustí funkci automaticky po skončení videa. Load spouští video příkazem *play()*. Click umožňuje kliknutím video zastavit – používá příkaz *pause()*. Po skončení videa Handler Stop zajistí nahrání dalšího slidu.

```
<Viewport>
…
  <HorizontalLayout Id="hl1">
    <Properties>
      <Width Value="100%" />
      <Height Value="70%" />
      <Alignment Value="mc" />
    </Properties>
    <Components>
      <Video Id="video">
        <Handlers>
          <Load>
            <Call Action="video_load" />
          </Load>
          <Click>
            <Call Action="video_click" />
          </Click>
         <Stop>
            <Call Action="nextSlide" />
          </Stop>
        </Handlers>
      </Video>
    </Components>
  </HorizontalLayout>
…
</Viewport>

<Variables>
  <Variable Id="video" Type="Object">
    <Reference>
      <Component Id="video" />
    </Reference>
  </Variable>
</Variables>

<Actions>
  <Action Id="video_load">
    <Expression>video->play()</Expression>
```

```
  </Action>
  <Action Id="video_click">
    <Expression> video->pause()</Expression>
  </Action>
  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
</Actions>
```

Audio

Zvuková stopa v různých zvukových formátech. Parametry jsou stejné jako u elementu Video, kromě toho, že Audio nereaguje na kliknutí. Při přehrávání by měl být vidět ovládací panel, který je nastavitelný parametry Width a Height.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	Šířka. Pokud je parametr uveden, text se bude zalamovat.	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	Zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right).	mc

Události:

Event	Význam
Load	nahrání zvukového záznamu
Start	spuštění zvukového záznamu
Stop	ukončení zvukového záznamu

Další povinné elementy mimo Properties:
Sources – URL zvuku

```
<Sources>
  <Source Url="..."/>
  <Source Url="..."/>
</Sources>
```

Ukázka:

```
<Bind>
  <Audio Id="audio">
    <Properties>
        <Width Value="854px" />
        <Height Value="480px" />
    </Properties>
    <Sources>
      <Source Url="http://stampach.wz.cz/witchdrums.mp3" />
    </Sources>
  </Audio>
</Bind>
```

```
<Viewport>
…
  <HorizontalLayout Id="hl1">
    <Properties>
      <Width Value="100%" />
      <Height Value="70%" />
      <Alignment Value="mc" />
    </Properties>
    <Components>
      <Audio Id="audio">
        <Handlers>
          <Load>
            <Call Action="audio_load" />
          </Load>
          <Stop>
            <Call Action="nextSlide" />
          </Stop>
        </Handlers>
      </Audio>
    </Components>
  </HorizontalLayout>
…
</Viewport>

<Variables>
  <Variable Id="audio" Type="Object">
    <Reference>
      <Component Id="audio" />
    </Reference>
  </Variable>
</Variables>

<Actions>
  <Action Id="audio_load">
    <Expression>audio->play()</Expression>
  </Action>
  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
</Actions>
```

Další komponenty přiřazené formou pluginu

Jde o komponenty, které jsou součástí zvláštního balíku, který rozšiřuje původní funkcionalitu programu Hypothesis. Takového komponenty jsou často složité a mývají velké množství dalších podelementů, proto jsou popisovány v této samostatné kapitole, ačkoliv jde jinak o komponenty akční.

Map

Složitý element, který se skládá z mnoha součástí. Ty jsou popisovány v následujících kapitolách. Je součástí doplňkového balíku (pluginu). Mapa má široké možnosti použití. Připomíná element Image a stejně jako on může reagovat na kliknutí. Kromě toho do ní lze ale i kreslit, definovat jednotlivé vrstvy a skládat je na sebe, vkládat do nich symboly atd.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Width	text (100%, 230px, 230)	šířka	
Height	text (jako Width)	výška	
Alignment	tl, tc, tr, ml, mc, mr, bl, bc, br	zarovnání, kombinace hodnot svislého zarovnání (Top, Modele, Bottom) a vodorovného zarovnání (Left, Center, Right)	mc
CRS	Kódy ze seznamu EPSG. Zdroj: http://www.epsg-registry.org/	Souřadný systém. Musí být definován, pokud bude v mapě použit element WMSLayer.	
BoundingBox		Ohraničení obdélníku mapy, souřadnice v pořadí minx, miny, maxx a maxy. Musí být definován, pokud bude v mapě použit element WMSLayer.	

Další povinné elementy mimo Properties:
Layers – jednotlivé vrstvy mapy. Může nabývat hodnot **ImageLayer, ImageSequenceLayer** (rastrový obrázek nebo série obrázků) či **WMSLayer** (většinou používáno pro podklad) a **FeatureLayer** (vektorová vrstva, lze do ní např. i kreslit).

Controls – prvky pro editaci objektů vrstvy a ovládání mapy. Může nabývat hodnot **DrawPoint** (zákres bodu), **DrawPath** (zákres lomené linie) a **DrawPolygon** (zákres polygonu). Součástí této skupiny jsou i nástroje **Pan** (posun mapy) a **Zoom** (změnu měřítka).

Styles – nastaveni vizualizace některých prvků.

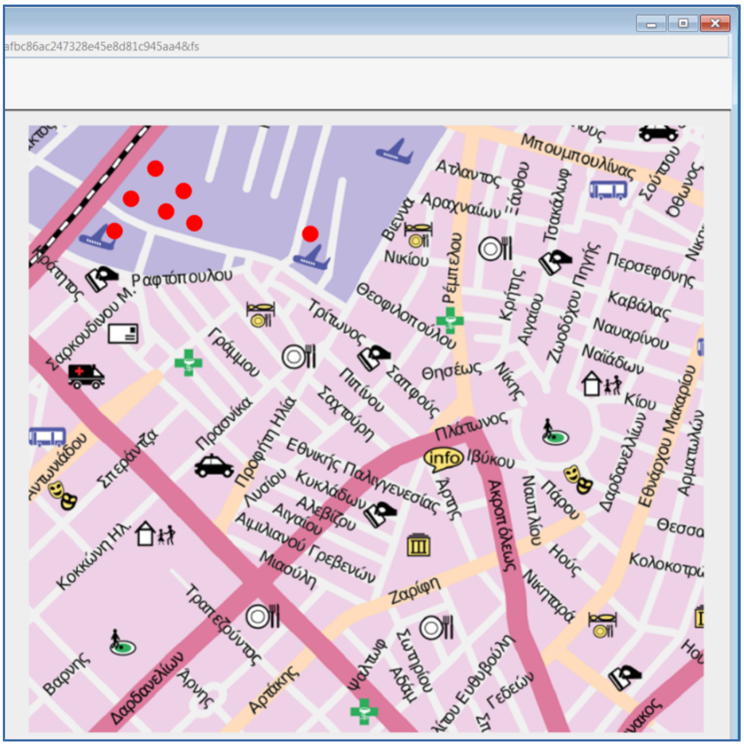
Jednotlivé prvky jsou podrobně popsány v následujících kapitolách.

Ukázka:
Je součástí doplňkového balíku (pluginu). Proto je názvům elementů tohoto balíku předřazen název jmenného prostoru *maps:* - např. <maps:Map Id="map">. Do záhlaví XML souborů (šablona, obsah), kde se tyto elementy použijí se musí zadat adresa tohoto jmenného prostoru:

```
<?xml version="1.0" encoding="UTF-8"?>
<SlideTemplate
 xmlns:maps="http://hypothesis.cz/xml/maps"
 UID="EC28FE19-9D33-4501-9156-909280B867C3">
 <Viewport>
  …
  <maps:Map Id="map">
   <Properties>
    <Width Value="990px" />
    <Height Value="585px" />
   </Properties>
   <Layers>
    …
   </Layers>
   <Controls>
    …
   </Controls>
   <Styles>
    …
   </Styles>
  </maps:Map>
  …
 </Viewport>
 …
</SlideTemplate>

<?xml version="1.0" encoding="UTF-8"?>
<SlideContent xmlns:maps="http://hypothesis.cz/xml/maps"
 TemplateUID="EC28FE19-9D33-4501-9156-909280B867C3">
 <Bindings>
  <Bind>
   <maps:Map Id="map">
    <Properties>
     <Width Value="640" />
     <Height Value="480" />
    </Properties>
   </maps:Map>
  </Bind>
 </Bindings>
</SlideContent>
```

Následující obrázek ukazuje mapu se dvěma vrstvami. Podkladový obrázek tvoří vrstva ImageLayer. Nad ní je položena průhledná vrstva FeatureLayer, která slouží ke kreslení červených bodů kliknutím myší (vlevo nahoře).


afbc86ac247328e45e8d81c945aa4&fs
Μπουμπουλίνας
Ατλαντος
Αραχναίων
Σούτσου
Οθωνος
Τσακάλωφ
Βιέννη
Νικίου
Ρεμπέλου
Κορίνης
Αιγαίου
Ζωοδόχου Πηγής
Περσεφόνης
Καβάλας
Ναυαρίνου
Ναϊάδων
Κίου
Ραφτόπουλου
Σαρκουδίνου Μ.
Γράμμου
Τρίτωνος
Θεοφυλοπούλου
Θησέως
Νίκης
Σαπφούς
Πιττίνου
Σαχτούρη
Πλάτωνος
Ιβύκου
Αρτής
Ακροπόλεως
Ναυπλίου
Πάρου
Δαρδανελλίων
Εθνάρχου Μακαρίου
Αρματωλών
Θεσσα
Κολοκοτρώ
Πρασονίκα
Προφήτη Ηλία
Εθνικής Παλιγγενεσίας
Κυκλάδων
Λυσίου
Αλεβίζου
Αιγαίου
Αιμιλιανού
Γρεβενών
Μιαούλη
Στεφανίτζα
Κοκκώνη Ηλ.
Τραπεζούντος
Ζαρίφη
Ηούς
Νικηταρά
Βαρνης
Δαρδανελλίων
Αρνης
Αρτάκης
Ψάλτου
Σωτηρίου
Αδάμ
Γεδεών
info

Sekce elementů Map – Layers

ImageLayer

Jeden ze dvou druhů vrstev tvořící mapu. Vrstvu tvoří rastrový obrázek uložený v parametru Url. Umisťuje se do elementu Layers. Vrstvy nemají parametr Width a Height. Jejich plocha je stejná jako plocha mapy, určená parametry elementu Map.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Url	text	url adresa obrázku tvořícího vrstvu	

Události:

Event	Význam
Load	nahrání vrstvy do slide
Click	kliknutí myší do vrstvy

Ukázka:
Následující kód šablony ukazuje případ, kdy element Layers obsahuje jednu vrstvu ImageLayer, která reaguje na kliknutí, tak že se nahraje následující slide.

```
<maps:Map Id="map">
  …
  <Layers>
    <ImageLayer Id="image_layer">
      <Properties>
        <Url Value="http://hypothesis.cz/gallery/albums/userpics/10001/01a.png" />
      </Properties>
      <Handlers>
        <Click>
          <Call Action="nextSlide" />
        </Click>
      </Handlers>
    </ImageLayer>
  </Layers>
  …
</maps:Map>
…
<Actions>
  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
</Actions>
```

ImageSequenceLayer

Jde o zvláštní případ elementu ImageLayer, kdy vrstva není tvořena jedním rastrovým obrázkem, ale postupně se ve vrstvě obrázky střídají. Nemá ve svých parametrech ani URL obrázků, ta jsou součástí elementu Images.

Události:

Event	Význam
Load	nahrání obrázků do slide
Click	kliknutí myší do vrstvy
Change	změna obrázku

Další elementy mimo Properties:
Images – element sdružující jednotlivé obrázky, které se postupně objeví v rámci vrstvy ImageSequenceLayer.

Ukázka:
V následujícím kódu je definován ImageSequenceLayer, který postupně zobrazí tři obrázky z adres uložených v elementu Images. U každého obrázku má uživatel možnost zvolit jednu z kláves Left či Right. To je zajištěno dvěma Handlery typu Shortcut, který bude blíže popisován v příslušné kapitole.

Po načtení celé slide se celá mapa zakryje neprůhlednou vrstvou. To zajišťuje kód „map->mask()“ spuštěný v Handleru Init, který patří celému slide. Naopak samotnému elementu ImageSequenceLayer náleží Handler Load. Ten naopak zakrytí zruší ve chvíli, kdy jsou nahrány obrázky patřící do ImageSequenceLayeru. Tak je zajištěno, že první obrázek bude vidět až ve chvíli, kdy jsou připraveny i ty následující a nedojde tak k prodlevám mezi obrázky a ovlivnění testu.

Jakmile uživatel zmáčkne jedno z tlačítek, tak se zavolá Action „key_press“. V ní se ověří, jestli šlo o pravé či levé tlačítko a podle toho se navýší proměnná leftCount nebo rightCount. Následně se spustí akce „tileChange“. Pořadí aktuálního obrázku je porovnáno s počtem všech obrázků v elementu ImageSequenceLayer (uložený v proměnné tileCount) a následně je buď načten další obrázek, nebo je zavolán následující slide, pokud už další obrázek není k dispozici.

```
<SlideTemplate>
 …
 <Viewport>
  …
  <maps:Map Id="map">
   …
   <Layers>
    …
    <ImageSequenceLayer Id="sequence_layer">
     <Images>
      <Image Url="http://hypothesis.cz/gallery/albums/userpics/10001/normal_cervena.png" />
      <Image Url="http://hypothesis.cz/gallery/albums/userpics/10001/normal_modra.png" />
      <Image Url="http://hypothesis.cz/gallery/albums/userpics/10001/normal_zelena.png" />
     </Images>
```

```
      <Handlers>
        <Load>
          <Expression>map->unmask()</Expression>
        </Load>
        <Change>
          <Call Action="tile_change" />
        </Change>
      </Handlers>
    </ImageSequenceLayer>
    …
  </Layers>
  …
 <maps:Map Id="map">
 …
</Viewport>

<Handlers>
 <Init>
  <Expression>map->mask()</Expression>
  <Expression>tilesCount=seqLayer->getTilesCount()</Expression>
 </Init>
 <Shortcut Key="Left">
  <Expression>key=1</Expression>
  <Call Action="key_press" />
  <Call Action="next_tile" />
 </Shortcut>
 <Shortcut Key="Right">
  <Expression>key=2</Expression>
  <Call Action="key_press" />
  <Call Action="next_tile" />
 </Shortcut>
</Handlers>

<Variables>
 <Variable Id="map" Type="Object">
  <Reference>
   <Component Id="map" />
  </Reference>
 </Variable>
 <Variable Id="rightCount" Type="Integer" Value="0" />
 <Variable Id="leftCount" Type="Integer" Value="0" />
 <Variable Id="key" Type="Integer" Value="0" />
 <Variable Id="tilesCount" Type="Integer" Value="0" />
 <Variable Id="seqLayer" Type="Object">
  <Reference>
   <Component Id="sequence_layer" />
  </Reference>
 </Variable>
</Variables>

<Actions>
 <Action Id="key_press">
  <If>
   <Expression>key==2</Expression>
   <True>
    <Expression>rightCount=rightCount+1</Expression>
   </True>
   <False>
    <Expression>leftCount=leftCount+1</Expression>
   </False>
```

```
    </If>
   </Action>

   <Action Id="next_tile">
    <If>
     <Expression>index&lt;tilesCount-1</Expression>
     <True>
      <Expression>seqLayer->nextTile()</Expression>
     </True>
     <False>
      <Call Action="nextSlide" />
     </False>
    </If>
   </Action>

   <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
   </Action>
  </Actions>

</SlideTemplate>
```

WMSLayer

Jeden z rastrových druhů vrstev tvořící mapu. Vrstvu tvoří rastrový obrázek, tentokrát však nejde o statický obrázek uložený na určité adrese. Obrázek je získáván z webové mapové služby (WMS). Jde o formát, jakým se ze specializovaných mapových serverů získávají mapy ve formě obrázku. Jde o službu standardizovanou dle specifikace ISO 19128 Geographic Information: Web Map Service.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Url	text	url adresa mapového serveru	
Layers	text	seznam vrstev požadovaných z mapového serveru	

Události:

Event	Význam
Load	nahrání vrstvy do slide
Click	kliknutí myší do vrstvy

Ukázka:
Následující kód ukazuje mapu, která má vektorovou vrstvu (element FeatureLayer) zachycující kliknutí a podkladovou rastrovou vrstvu řešenu pomocí elementu WMSLayer. Mapa proto má nastaven i souřadný systém a ohraničující obdélník mapy. Kód 26986 odpovídá systému Massachusetts Mainland.

Samotný element WMSLayer má v elementu URL nastavenu adresu mapového serveru a v elementu Layers požaduje po serveru tři vrstvy. Názvy požadovaných vrstev byly získány předem z informací poskytovaných daným mapových serverem. V elementu Controls je s využitím elementů Pan a Zoom povoleno posouvání i změna měřítka pomocí myši. Tato nastavení musí být po zobrazení celého slide (využije se Handler Show) aktivována pomocí Action activatePan a activateZoom, které obsahují kód pan->activate() a zoom->activate(). K tomu slouží i dvě proměnné „zoom“ a „pan“ definované v elementu Variables.

```
<SlideTemplate>
 …
 <Viewport>
  …
  <maps:Map Id="map">
   <Properties>
    <Width Value="900px" />
    <Height Value="500px" />
    <CRS Value="EPSG:26986" />
    <BBOX Value="232325.38526025353,898705.3447384972,238934.49648710093,903749.1401484597"/>
   </Properties>

   <Layers>
    <WMSLayer Id="wms_layer">
     <Properties>
      <Url Value="http://giswebservices.massgis.state.ma.us/geoserver/wms"/>
```

```
        <Layers
Value="massgis:GISDATA.TOWNS_POLYM,massgis:GISDATA.NAVTEQRDS_ARC,massgis:GISDATA
.NAVTEQRDS_ARC_INT"/>
        </Properties>
       </WMSLayer>
       <FeatureLayer Id="feature_layer">
        …
        <Handlers>
         <Click>
          <Call Action="nextSlide" />
         </Click>
        </Handlers>
       </FeatureLayer>
     </Layers>

     <Controls>
     …
      <Pan Id="pan">
      </Pan>
      <Zoom Id="zoom">
      </Zoom>
     </Controls>
     …
   </maps:Map>
   …
 </Viewport>

 <Handlers>
   <Show>
    …
    <Call Action="activatePan" />
    <Call Action="activateZoom" />
   </Show>
 </Handlers>

 <Variables>
   …
   <Variable Id="pan" Type="Object">
    <Reference>
     <Component Id="pan" />
    </Reference>
   </Variable>
   <Variable Id="zoom" Type="Object">
    <Reference>
     <Component Id="zoom" />
    </Reference>
   </Variable>
 </Variables>

 <Actions>
   …
   <Action Id="activatePan">
    <Expression>pan->activate()</Expression>
   </Action>
   <Action Id="activateZoom">
    <Expression>zoom->activate()</Expression>
   </Action>
   …
   <Action Id="nextSlide">
    <Expression>Navigator>next()</Expression>
   </Action>
```

```
  </Actions>
</SlideTemplate>
```

WMSLayer je poměrně složitý element s širokým využitím. Další podrobnosti a komplexnější ukázky lze najít v dokumentu Návod_zakázka_PřF.pdf na str. 55-96.

FeatureLayer

Jeden ze dvou druhů vrstev tvořící mapu. Umisťuje se do elementu Layers. Vrstva je „vektorová“ - obsahuje jednotlivé prvky (polygony, lomené linie, body), které jsou definovány pomocí elementů Feature sdružených do elementu Features. Celá samotná vrstva FeatureLayer má velikost stejnou jako element Map, proto mezi parametry nejsou Width a Height. Může do jedné vrstvy patřit linie, body a polygony zároveň?

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Style	text	Definice vizualizace prvků vrstvy. Odkazuje na některý styl definovaný v elementu Styles.	
HoverStyle	text	Styl, který je použit na vybraný nebo zvýrazněný prvek.	

Události:

Event	Význam
Click	kliknutí myší do vrstvy

Další elementy mimo Properties:
Features – element sdružující definici jednotlivých prvků, ze kterých se skládá vrstva FeatureLayer.

Ukázka:
Níže uvedený kód ukazuje, jak jsou ve vrstvě FeatureLayer definovány další prvky pomocí elementu Feature. Tomu se bude věnovat následující kapitola. Zde je definován prvek s názvem „area“.

Je dobré si všimnout, že v kódu se vyskytují hned dvě sekce Handlers zachycující klik myší. Svou sekci Handlers má totiž jak FeatureLayer, tak jemu podřazený Feature „area“. Tučně je zvýrazněna sekce náležící elementu FeatureLayer. Ta zachycuje všechny kliky kdekoliv na celé vrstvě, která má vždy stejnou rozlohu jako samotná mapa, a zobrazí další slide.

Naproti tomu sekce Handlers náležící elementu Feature zachycuje pouze kliknutí do prostoru samotného Feature „area“ a spustí akci „spravna_odpoved“ – nastaví proměnnou „result“ na hodnotu 1. Protože ale Feature je zároveň i součástí celé vrstvy FeatureLayer, tak je stejným kliknutím spuštěn zároveň i Handler celé vrstvy. To znamená, že při jakémkoliv kliknutí do mapy je zobrazen další slide, ale pouze v případě, že je kliknuto na Feature „area“, tak je zaznamenáno, že jde o správnou odpověď. Toho se využívá při rozlišování správných a nesprávných reakcí při testech s mapou.

Kód obsahuje elementy Style a HoverStyle, které odkazují na jednotlivé definice v rámci elementu Styles. Jeho podrobný popis je ve stejnojmenné kapitole o tomto elementu.

```
<maps:Map Id="map">
…
  <Layers>
    <FeatureLayer Id="feature_layer">
      <Properties>
        <Style Value="red_cross" />
        <HoverStyle Value="green_cross" />
      </Properties>

      <Features>
        <Feature Id="area">
          <Geometry Value="POINT (300 100)" />
          <Text Value="A">
            <Offset X="0" Y="-20" />
          </Text>

          <Handlers>
            <Click>
              <Call Action="spravna_odpoved" />
            </Click>
          </Handlers>
        </Feature>
      </Features>

      <Handlers>
        <Click>
          <Call Action="nextSlide" />
        </Click>
      </Handlers>
    </FeatureLayer>
  </Layers>
  …
  <Styles>
    …
  </Styles>
</maps:Map>
…
<Variables>
  <Variable Id="result" Type="Integer" Value="0" />
</Variables>
<Actions>
  <Action Id="spravna_odpoved">
    <Expression>result=1</Expression>
  </Action>
  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
</Actions>
```

Feature

Element slouží pro definování jednotlivých prvků vrstvy FeatureLayer. Může jít o bod, čáru, lomenou čáru nebo polygon. Typ geometrie a souřadnice se určuje v elementu Geometry ve formátu Well Known Text (WKT), což je standard ISO/IEC 13249-3:2011 pro textový zápis vektorové geometrie.

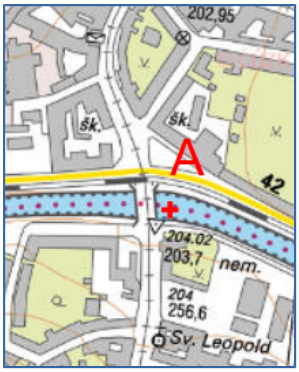
Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
Style	text	Definice vizualizace prvků vrstvy. Odkazuje na některý styl definovaný v elementu Styles.	
HoverStyle	text	Styl, který je použit na vybraný nebo zvýrazněný prvek.	
Hidden	True/False	Prvek je či není neviditelný.	False

Události:

Event	Význam
Click	kliknutí myší do vrstvy

Další elementy mimo Properties:
Text – Popisek prvku. Vnořený element Offset pak definuje umístění popisku (resp. jeho levého dolního rohu) pomocí souřadnic X a Y a to relativně k poloze popisované geometrie. Souřadnice X narůstají směrem doprava a souřadnice Y směrem dolů. Jednotkami jsou pixely.




```
<Text Value="A">
  <Offset X="0" Y="-20" />
</Text>
```

Souřadnice X popisku jsou stejné jako souřadnice popisovaného bodu, záporné souřadnice Y znamenají, že je popisek o 20 pixelů nad popisovaným bodem.

Geometry – WKT definice geometrie. Souřadnice se uvádějí v pixelech, souřadnice X narůstají směrem doprava, souřadnice Y směrem dolů.
Příklady:
<Geometry Value="POINT (30 10)"/>
<Geometry Value="LINESTRING (30 10, 10 30, 40 40)"/>
Souřadnice prvního a posledního bodu polygonu musí být totožné:
<Geometry Value="POLYGON ((**30 10**, 40 40, 20 40, 10 20, **30 10**))" />
Polygon může obsahovat i vnitřní hranici (vnitroblok):
<Geometry Value="POLYGON ((**35 10**, 15 40, 10 20, **35 10**), (**20 30**, 35 35, 30 20, **20 30**))" />

Program akceptuje i složitější geometrie - MultiPolygon. MultiLineString, MultiPoint, GeometryCollection. Jejich popis lze najít v dokumentaci standardu ISO/IEC 13249-3:2011.

Ukázka:
Následující kód je totožný s předešlým příkladem, jen je tučně vyznačena naopak ta část, která náleží elementu Feature a kterou ovládá Handler tohoto elementu. Feature je nastaven jako neviditelný. To znamená, že v mapě není vidět jeho ohraničení.

```
<maps:Map Id="map">
…
 <Layers>
  <FeatureLayer Id="feature_layer">
   <Properties>
    <Style Value="red_cross" />
   </Properties>

   <Features>
    <Feature Id="area">
     <Geometry Value="POINT (100 300)" />
     <Text Value="A">
      <Offset X="0" Y="-20" />
     </Text>
     <Properties>
      <Hidden Value="true" />
     </Properties>
     <Handlers>
      <Click>
       <Call Action="spravna_odpoved" />
      </Click>
     </Handlers>
    </Feature>
   </Features>

   <Handlers>
    <Click>
     <Call Action="nextSlide" />
    </Click>
   </Handlers>
  </FeatureLayer>
 </Layers>
 …
 <Styles>
  …
 </Styles>
</maps:Map>
…
<Variables>
 <Variable Id="result" Type="Integer" Value="0" />
</Variables>
<Actions>
 <Action Id="spravna_odpoved">
  <Expression>result=1</Expression>
 </Action>
 <Action Id="nextSlide">
  <Expression>Navigator->next()</Expression>
 </Action>
</Actions>
```

Sekce elementů Map – Controls

DrawPoint

Element definující kreslení bodů. Jeden ze způsobů kreslení prvků do určité vrstvy FeatureLayer. Je obsažen v elementu Controls.

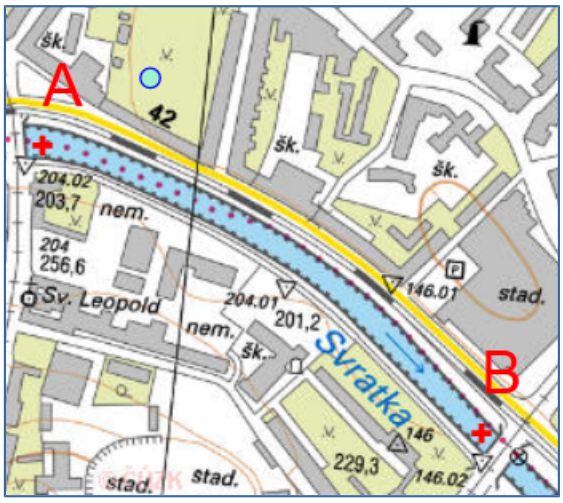
Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
LayerId	text	Vrstva FeatureLayer, do které se kreslí.	
CursorStyle	text	Vzhled kreslícího kurzoru.	

Události:

Event	Význam
Draw	Dokončení kreslení prvku.

Ukázka:
Umístění elementu DrawPoint do elementu Controls určuje, že kreslenými prvky budou body. Vnořený element LayerId určuje, do které vrstvy typu FeatureLayer obsažené v elementu Layers se bude nový prvek kreslit – v ukázce je to vrstva „feature_layer“. Element CursorStyle určuje vzhled kurzoru myši, kterým se kreslí. Na obrázku níže je to kroužek vlevo nahoře.



Vzhled samotných kreslených prvků (v ukázce jde o body) se však v elementu Controls nenastavuje! Ten je definován v elementu Style vrstvy, do které se kreslí. V ukázce je vrstvě „feature_layer“ nastaven Style „red_cross“.

Kreslení prvků musí být ale nejdříve aktivováno. Je kvůli tomu definována proměnná Variable „drawPoint“, která odkazuje na samotný element DrawPoint. Tato proměnná „drawPoint“ je využita v Action „activatePoint“, která obsahuje kód aktivující kreslení drawPoint>activate(). Handler Show (který pak slouží k tomu, aby se po načtení slide spustila Action activatePoint a

tak bylo kreslení pomocí elementu DrawPoint aktivováno. Všimněte si, že Handler Show náleží do sekce Handlers celé šablony, protože reaguje na její načtení.

Naproti tomu Handler Draw je umístěn v sekci Handlers elementu DrawPoint, protože se v něm definuje činnost po dokreslení prvku. V ukázce je volána Action „draw_finished“, která zobrazí následující slide.

```
<maps:Map Id="map">
 …
 <Layers>
  <FeatureLayer Id="feature_layer">
   <Properties>
    <Style Value="red" />
   </Properties>
   …
  </FeatureLayer>
 </Layers>

 <Controls>
  <DrawPoint Id="draw_point">
   <Properties>
    <LayerId Value="feature_layer" />
    <CursorStyle Value="cursor" />
   </Properties>
   <Handlers>
    <Draw>
     <Call Action="draw_finished" />
    </Draw>
   </Handlers>
  </DrawPoint>
 </Controls>

 <Styles>
  <Style Id="cursor">
   …
  </Style>
  <Style Id="red">
   …
  </Style>
 </Styles>
 …
</maps:Map>

<Handlers>
 <Show>
  <Call Action="activatePoint" />
 </Show>
</Handlers>

<Variables>
 <Variable Id="drawPoint" Type="Object">
  <Reference>
   <Component Id="draw_point" />
  </Reference>
 </Variable>
</Variables>

<Actions>
 <Action Id="nextSlide">
```

```
    <Expression>Navigator->next()</Expression>
  </Action>
  <Action Id="draw_finished">
    <Call Action="nextSlide" />
  </Action>
  <Action Id="activatePoint">
    <Expression>drawPoint>activate()</Expression>
  </Action>
</Actions>
```

DrawPath

Element definující kreslení linií. Jeden ze způsobů kreslení prvků do určité vrstvy FeatureLayer. Je obsažen v elementu Controls.

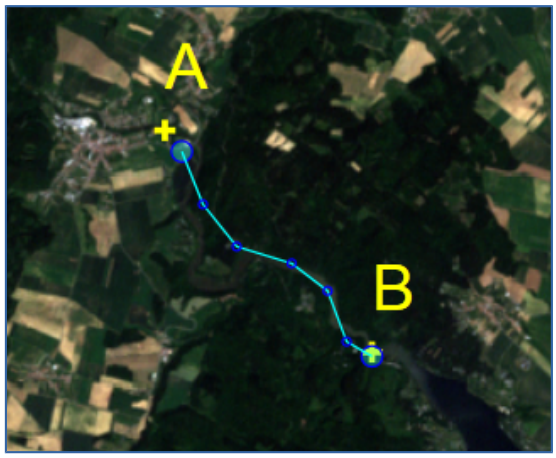
Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
LayerId	text	Vrstva FeatureLayer, do které se kreslí	
CursorStyle	text	Vzhled kreslícího kurzoru	
StartPointStyle	text	Vzhled úvodního bodu kreslené lomené čáry	
LineStyle	text	Vzhled linie kreslené lomené čáry	
VertexStyle	text	Vzhled lomového bodu čáry	
FinishStrategy	AltClick, DoubleClick	Způsob, kterým se ukončí kreslení čáry. Buď stisknutím klávesy Alt zároveň s kliknutím myší, nebo pomocí dvojkliku myši.	AltClick

Události:

Event	Význam
Draw	Dokončení kreslení prvku.

Ukázka:
Umístění elementu DrawPath do elementu Controls určuje, že kreslenými prvky bodou linie, včetně lomených. Vnořený element LayerId určuje, do které vrstvy typu FeatureLayer obsažené v elementu Layers se bude nový prvek kreslit – v ukázce je to vrstva „feature_layer“. Element CursorStyle určuje vzhled kurzoru myši, kterým se kreslí. Elementy StartPointStyle, VertexStyle a LineStyle nastavují vzhled úvodního bodu (kroužek u A), lomových bodů (malé kroužky) a čáry kreslené lomené linie. Na následujícím obrázku je počáteční bod linie kroužek u A, linie je značena bílou čarou s malými kroužky coby lomovými body a kurzor myši je znázorněn větším kroužkem u B.



Element FinishStrategy definuje způsob, jakým se ukončí kreslení linie. V ukázce není uveden, proto platí výchozí nastavení – poslední bod linie se nakreslí spolu se stisknutou klávesou Alt.

Vzhled samotných kreslených prvků (v ukázce jde o linie) se však v elementu Controls nenastavuje! Ten je definován v elementu Style vrstvy, do které se kreslí. V ukázce je vrstvě „feature_layer“ nastaven Style „red_line“. Kromě toho je definována i řada dalších stylů, které nastavují vzhled úvodního bodu, linie a lomových bodů.

Kreslení linie musí být nejdříve aktivováno. Je kvůli tomu definována proměnná Variable „drawPath“, která odkazuje na samotný element DrawPath. Tato proměnná „drawPath“ je využita v Action „activatePath“, která obsahuje kód aktivující kreslení drawPath>activate(). Handler Show (který pak slouží k tomu, aby se po načtení slide spustila Action activatePath a tak bylo kreslení pomocí elementu DrawPath aktivováno. Všimněte si, že Handler Show náleží do sekce Handlers celé šablony, protože reaguje na její načtení.

Handler Draw definuje činnost po dokreslení prvku. V ukázce je volána Action „draw_finished“, která zobrazí následující slide.

```
<maps:Map Id="map">
  …
  <Layers>
    <FeatureLayer Id="feature_layer">
      <Properties>
        <Style Value="red_line" />
      </Properties>
      …
    </FeatureLayer>
  </Layers>

  <Controls>
    <DrawPath Id="draw_path">
      <Properties>
        <LayerId Value="feature_layer" />
        <CursorStyle Value="cursor" />
        <StartPointStyle="start" />
        <LineStyle="line" />
        <VertexStyle="vertex" />
      </Properties>
      <Handlers>
        <Draw>
```

```
        <Call Action="draw_finished" />
       </Draw>
      </Handlers>
    </DrawPath>
  </Controls>

  <Styles>
    <Style Id="cursor">
      …
    </Style>
    <Style Id="red_line">
      …
    </Style>
    <Style Id="line">
      …
    </Style>
    <Style Id="start">
      …
    </Style>
    <Style Id="vertex">
      …
    </Style>
  </Styles>
  …
</maps:Map>

<Handlers>
  <Show>
    <Call Action="activatePath" />
  </Show>
</Handlers>

<Variables>
  <Variable Id="drawPath" Type="Object">
    <Reference>
      <Component Id="draw_path" />
    </Reference>
  </Variable>
</Variables>

<Actions>
  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
  <Action Id="draw_finished">
    <Call Action="nextSlide" />
  </Action>
  <Action Id="activatePath">
    <Expression>drawPath>activate()</Expression>
  </Action>
</Actions>
```

DrawPolygon

Element definující kreslení polygonů. Jeden ze způsobů kreslení prvků do určité vrstvy FeatureLayer. Je obsažen v elementu Controls.

Parametry:

Název	Povolené hodnoty	Význam	Výchozí hodnota
LayerId	text	Vrstva FeatureLayer, do které se kreslí	
CursorStyle	text	Vzhled kreslícího kurzoru.	
StartPointStyle	text	Vzhled úvodního bodu kresleného polygonu	
LineStyle	text	Vzhled linie kresleného polygonu	
VertexStyle	text	Vzhled lomového bodu polygonu	
FinishStrategy	AltClick, DoubleClick	Způsob, kterým se ukončí kreslení polygonu. Buď stisknutím klávesy Alt zároveň s kliknutím myší, nebo pomocí dvojkliku myši.	AltClick

Události:

Event	Význam
Draw	Dokončení kreslení prvku.

Ukázka:
Umístění elementu DrawPolygon do elementu Controls určuje, že kreslenými prvky bodou polygony. Vnořený element LayerId určuje, do které vrstvy typu FeatureLayer obsažené v elementu Layers se bude nový prvek kreslit – v ukázce je to vrstva „feature_layer“. Element CursorStyle určuje vzhled kurzoru myši, kterým se kreslí. Elementy StartPointStyle, VertexStyle a LineStyle nastavují vzhled úvodního bodu, lomových bodů a čáry kreslené lomené linie. Element FinishStrategy definuje způsob, jakým se ukončí kreslení polygonu. V ukázce není uveden, proto platí výchozí nastavení – poslední bod polygonu se nakreslí spolu se stisknutou klávesou Alt a polygon se následně automaticky uzavře.

Vzhled samotných kreslených prvků (v ukázce jde o polygony) se však v elementu Controls nenastavuje! Ten je definován v elementu Style vrstvy, do které se kreslí. V ukázce je vrstvě „feature_layer“ nastaven Style „red_polygon“.

Handler Draw definuje činnost po dokreslení prvku. V ukázce je volána Action „draw_finished“, která zobrazí následující slide.

```
<maps:Map Id="map">
  …
  <Layers>
    <FeatureLayer Id="feature_layer">
```

```
    <Properties>
      <Style Value="red_polygon" />
    </Properties>
    …
  </FeatureLayer>
 </Layers>

 <Controls>
  <DrawPolygon Id="draw_polygon">
   <Properties>
    <LayerId Value="feature_layer" />
    <CursorStyle Value="cursor" />
    <StartPointStyle="start" />
    <LineStyle="line" />
    <VertexStyle="vertex" />
   </Properties>
   <Handlers>
    <Draw>
     <Call Action="draw_finished" />
    </Draw>
   </Handlers>
  </DrawPolygon>
 </Controls>

 <Styles>
  <Style Id="cursor">
   …
  </Style>
  <Style Id="red_polygon">
   …
  </Style>
  <Style Id="line">
   …
  </Style>
  <Style Id="start">
   …
  </Style>
  <Style Id="vertex">
   …
  </Style>
 </Styles>
 …
</maps:Map>

<Handlers>
 <Show>
  <Call Action="activatePolygon" />
 </Show>
</Handlers>

<Variables>
 <Variable Id="drawPolygon" Type="Object">
  <Reference>
   <Component Id="draw_polygon" />
  </Reference>
 </Variable>
</Variables>

<Actions>
 <Action Id="nextSlide">
```

```
    <Expression>Navigator->next()</Expression>
  </Action>
  <Action Id="draw_finished">
    <Call Action="nextSlide" />
  </Action>
  <Action Id="activatePolygon">
    <Expression>drawPolygon>activate()</Expression>
  </Action>
</Actions>
```

Pan

Element ze skupiny Controls, který slouží k povolení posunu mapy pomocí myši. Ukázka jeho využití je v kapitole WMSLayer. Element, který vlastnost Pan definuje v rámci elementu Controls je velmi jednoduchý:

```
<Controls>
  <Pan Id="pan">
  </Pan>
</Controls>
```

Funkce Pan však musí být povolena i kódem „pan->activate() v některé z volaných Action:

```
<Action Id="activatePan">
  <Expression>pan->activate()</Expression>
</Action>
```

K tomu bývá potřeba definovat proměnnou, která se odkáže na element Pan v elementu Controls:

```
<Variable Id="pan" Type="Object">
  <Reference>
    <Component Id="pan" />
  </Reference>
</Variable>
```

Zoom

Element ze skupiny Controls, který slouží k povolení změny měřítka mapy pomocí kolečka myši. Ukázka jeho využití je v kapitole WMSLayer. Element, který vlastnost Zoom definuje v rámci elementu Controls je velmi jednoduchý:

```
<Controls>
  <Zoom Id="zoom">
  </Zoom>
</Controls>
```

Funkce Zoom však musí být povolena i kódem „zoom->activate() v některé z volaných Action:

```
<Action Id="activateZoom">
  <Expression>zoom->activate()</Expression>
</Action>
```

K tomu bývá potřeba definovat proměnnou, která se odkáže na element Zoom v elementu Controls:

```
<Variable Id="zoom" Type="Object">
  <Reference>
    <Component Id="zoom" />
  </Reference>
</Variable>
```

Sekce elementů Map – Styles

Style

Style je definován jako samostatná datová struktura s několika položkami. Slouží k definování vzhledu vektorových objektů – vektorové vrstvě FeatureLayer i jednotlivým jejím prvkům Feature.

Styl lze přiřadit vektorové vrstvě FeatureLayer i jednotlivým vektorovým objektům Feature. Styly jsou základní (Style) nebo zvýrazňovací (HoverStyle). Existují i speciální styly, které slouží k změně vizualizace kurzoru (CursorStyle), kreslené linie (LineStyle), počátečního bodu kreslené linie (StartPointStyle), lomového bodu linie (VertexStyle).

Tento element se od doposud popisovaných elementů liší. Nemá totiž žádné parametry v sekci Properties, ani Handlery, které by reagovaly na nějaké události. Elementy jednotlivých povolených parametrů se zapisují rovnou do elementu Style.

Tato kapitola je pouze krátkým shrnutím elementu Style. Podrobnější vysvětlení lze najít v kapitole 4 dokumentu Návod_zakázka_PřF.pdf, kde jsou i ukázky kódů.

Parametry:

Název	Typ	Výchozí hodnota	Povolené hodnoty	Popis
StrokeColor	text	black	CSS barvy, (http://www.w3schools.com/cssref/css_colorsfull.asp)	Barva čáry, ohraničení; barvy můžou být zadány jménem dle CSS, nebo jako hex hodnota #000000.
StrokeWidth	celé číslo	1	>= 0	Tloušťka čáry, ohraničení; 0 = bez ohraničení.
StrokeOpacity	číslo	1,0	0,0 - 1,0	Průhlednost čáry, ohraničení; 1 = úplně neprůhledná.
FillColor	řetězec	white	CSS barvy	Barva výplně plošných objektů.
FillOpacity	číslo	1,0	0,0 - 1,0	Průhlednost výplně plošných objektů.
TextColor	řetězec	black	CSS barvy	Barva písma.
TextStrokeColor	řetězec		CSS barvy	Barva ohraničení písma.
TextStrokeWidth	celé číslo	0	>= 0	Tloušťka ohraničení písma.
TextOpacity	číslo	1,0	0,0 - 1,0	Průhlednost písma.
TextFillOpacity	číslo	1,0	0,0 - 1,0	Průhlednost výplně písma.
FontFamily	řetězec	Arial	systémové fonty	Druh fontu písma; může se lišit v závislosti na systému.

FontSize	celé číslo	20	> 0	Velikost písma.
PointRadius	celé číslo	5	>= 0	Poloměr kruhu, který prezentuje bod; pouze ve spojení s tvarem Circle; 0 = žádný bod.
PointShape	řetězec	Circle	Circle, Square, Cross, XCross, Asterisk, TriangleUp, TriangleDown, Diamond	Tvar bodu; ○, □, +, ×, ✳, △, ▽, ◇
PointShapeScale	číslo	5,0	> 0,0	Měřítko tvaru.

Ukázka:
V následujícím výseku jsou definovány styly vrstvy i prvku. Vrstva bodů má nastaven styl „red_cross“. Ten je definován pouze červeným křížkem o síle 3, žlutým popisem atd. Jeden z jejích prvků má sice nastaven stejný styl „red_cross“, při zvýraznění (např. při přejetí myší) se však dočasně použije styl „green_cross“. Ten se liší jen zelenou barvou a velikostí. Všechny tyto styly musí být definovány v části Styles.

```
<maps:Map Id="map">
  ...
  <Layers>
    <FeatureLayer Id="feature_layer">
      <Properties>
        <Style Value="red_cross" />
        ...
      </Properties>
      <Features>
        <Feature Id="start_cross">
          <Properties>
            <Style Value="red_cross" />
            <HoverStyle Value="green_cross" />
            ...
          </Properties>
          <Geometry Value="POINT (445 125)" />
        <Feature>
    </FeatureLayer>
  </Layers>
</maps:Map>

<Styles>
  <Style Id="red_cross">
    <StrokeColor Value="red" />
    <StrokeWidth Value="3" />
    <FillOpacity Value="0" />
    <PointShape Value="Cross" />
    <FontFamily Value="Arial" />
    <FontSize Value="30" />
    <TextColor Value="yellow" />
  </Style>
  <Style Id="green_cross">
```

```
    <StrokeColor Value="green" />
    <StrokeWidth Value="5" />
    <FillOpacity Value="0" />
    <PointShape Value="Cross" />
    <PointShapeScale Value="10" />
    <FontFamily Value="Arial" />
    <FontSize Value="30" />
    <TextColor Value="yellow" />
  </Style>
</Styles>
```

Větvení v kódu

Aby bylo v rámci elementů Action možno definovat i složitější funkcionalitu, je možné využít větvení. Pro větvení v rámci slidy lze použít konstrukci příkazu If, který vyhodnocuje logický (boolean) výraz obsažený v uzlu Expression. Je li výraz pravdivý, provede se blok True, v opačném případě se provede blok False (není povinný):

```
<If>
  <Expression>...</Expression>
  <True>
    …
  </True>
  <False>
    ...
  </False>
</If>
```

Druhým způsobem větvení je konstrukce Switch. Narozdíl od If se vyhodnocuje shoda výrazu s hodnotou Value v jednotlivých větvích Case:

```
<Switch>
  <Expression>...</Expression>
  <Case Value=”value1”>
    ...
  </Case>
  <Case Value=”value2”>
    ...
  </Case>
  ...
</Switch>
```

Jeden příklad větvení již byl uveden v kapitole ButtonPanel u Action „vyber_tlacitkem“. V kódu jsou naprogramována čtyři tlačítka. Po kliknutí na některé z nich se do proměnné buttonIndex uloží pořadí zvoleného tlačítka, přičemž správnou odpovědí je kliknutí na první z nich (tedy na hodnotu 3).


```
<Variables>
  <Variable Id="vysledek" Type="Integer" Value="0" />
  <Variable Id="buttonIndex" Type="Integer" Value="0" />
  <Variable Id="spravnyVyber" Type="Integer" Value="1" />
</Variables>
```

Následující ukázka kódu šablony vysvětluje, jak se dále pracuje s pořadím tlačítka uloženým do proměnné „buttonIndex“. V rámci akce „vyber_tlacitkem“ je tato hodnota srovnána s proměnnou „spravnyVyber“ získanou z obsahu scény. Pokud jsou obě hodnoty shodné, pak je proměnná „vysledek“ nastavena na hodnotu 1. Ať už je proměnná „výsledek“ nastavena na hodnotu 0 nebo 1, je zavolána akce „nextSlide“. Její kód zajistí přechod na následující scénu. Proměnná „vysledek“ je nastavena coby výsledek scény (element OutputValue). Bude tedy po skončení scény zapsána do databáze do tabulky ***tbl_slide_output*** do sloupce Output.

```
<Actions>
  <Action Id="vyber_tlacitkem">
    <If>
```

```
    <Expression>buttonIndex==spravnyVyber</Expression>
    <True>
      <Expression>vysledek=1</Expression>
    </True>
   </If>
   <Call Action="nextSlide" />
  </Action>

  <Action Id="nextSlide">
   <Expression>Navigator->next()</Expression>
  </Action>
</Actions>

<OutputValue>
  <Expression>vysledek</Expression>
</OutputValue>
```

Další příklady lze najít v předpřipravených XML šablonách 1e-tpl.xml, 1h-tpl.xml, 1i-tpl.xml, 1k-tpl.xml.

Zachycení souřadnic kliku myší

Myší lze v testu například stisknout tlačítko. Jak zjistit, že došlo ke kliknutí na tlačítko a jak naprogramovat navázanou reakci programu, je popisováno v kapitole o elementu Button. Další možnosti pro použití myši se otevírají s použitím elementu Map. Jak se pomocí myši kreslí, je popisováno v kapitolách o elementech FeatureLayer a Feature a souvisejících kapitolách DrawPoint, DrawPath a DrawPolygon. V těchto kapitolách je i vysvětleno, jak nadefinovat určitou oblast (pomocí elementu Feature), ve které je kliknutí zachycováno.

Kromě toho, že se dá zjistit, že bylo kliknuto do určité oblasti, lze ale také zjistit přesné souřadnice kliku. To je potřeba například k měření přesné délky nakreslené čáry. Souřadnice jsou získávány v pixelech, takže i změřená vzdálenost bude pouze v pixelech. Používá se k tomu funkce getCoordinate(). Souřadnice budou uloženy do proměnných nadefinovaných v elementu Variables, což znamená, že je bude možné najít v tabulce tbl_slide_output.

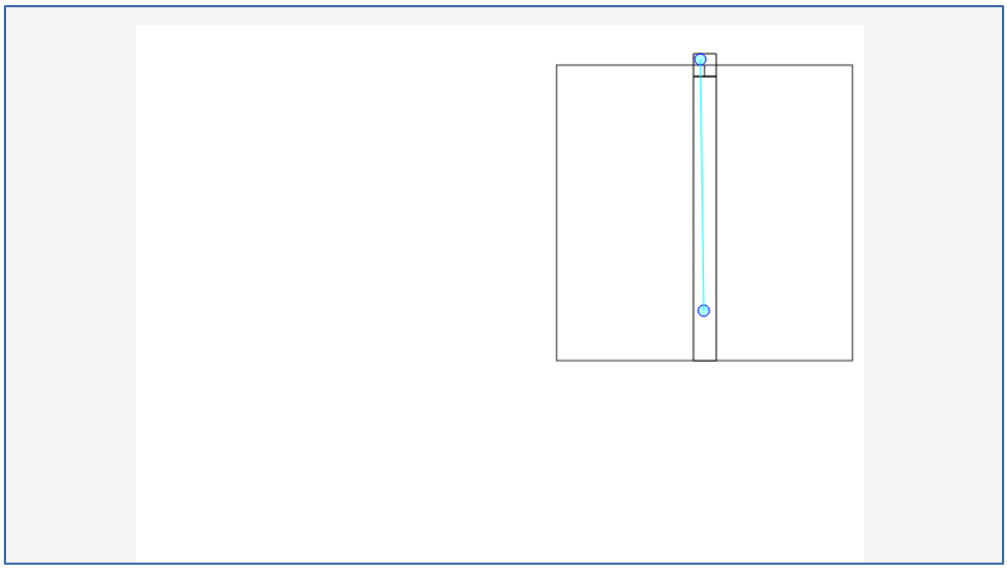
Pomocí funkce ComponentData->getCoordinate() se nejdříve získá proměnná coord představující souřadnice aktuálního kliku a z ní se následně získají obě souřadnice X a Y:
coord=ComponentData->getCoordinate()
xcoord=coord->x
ycoord=coord->y

Lze také oba kroky provést najednou:
xcoord=ComponentData->getCoordinate()->x
ycoord=ComponentData->getCoordinate()->y

Se získáváním souřadnic souvisí i problém, který již byl zmiňován v kapitole o elementu Image. Pokud není test spuštěn klasicky, pak jiné prohlížeče než Google Chrome mohou obrázek deformovat – například ho roztáhnout. Obrázek má následně více pixelů než které mu byly nadefinovány a tudíž i změřené kliknutí bude mít nesprávné souřadnice.

Ukázka:
V následujícím příkladu je ukázán kód (zkrácený), který vytváří element Map o velikosti 640x640 pixelů s dvěma vrstvami – ImageLayer, zobrazující obrázek čtverce, a FeatureLayer zachycující kliknutí. Úlohou uživatele je nakreslit čáru seshora dolů, která bude změřena.

Na následujícím obrázku je bílou barvou znázorněna oblast elementu Map. Vpravo nahoře je vidět obdélník, který je nakreslen coby rastrový obrázek v ImageLayer. V jeho prostřední části je vidět malý čtverec „start_point“ a úzký svislý obdélník „buffer“. Element „start_point“ definuje oblast, kde má uživatel začít kresbu, zatímco v oblasti „buffer“ má čáru ukončit, aby se výsledek považoval za platný.



V následujícím kódu je vidět, že element FeatureLayer má svůj Handler, který zajišťuje počítání všech kliků na celou plochu mapy pomocí volání Action „click_count“. Po dvou klicích se zavolá následující slide. Kromě toho jsou v kódu definovány dva elementy Feature pro čtverec „start_point“ a pro obdélník „buffer“. Oba mají rovněž své Handlery. Oba volají Action „buffer_click“, která počítá počet kliků do těchto vyhrazených částí mapy a start_point volá navíc Action „start_click“, která kontroluje, že první kliknutí se odehrálo v oblasti malého čtverce.

```
<SlideTemplate>
  …
  <Viewport>
    <maps:Map Id="map1">
      <Properties>
        <Width Value="640" />
        <Height Value="640" />
      </Properties>

      <Layers>
        <ImageLayer Id="image_layer">
          …
        </ImageLayer>

        <FeatureLayer Id="feature_layer">
          <Features>

            <Feature Id="start_point">
              …
              <Geometry Value="POLYGON ((490 25,490 45,510 45,510 25,490 25))" />
              <Handlers>
                <Click>
                  <Call Action="start_click" />
                  <Call Action="buffer_count" />
                </Click>
              </Handlers>
            </Feature>

            <Feature Id="buffer">
              …
              <Geometry Value="POLYGON ((490 45,490 295,510 295,510 45,490 45))" />
```

```
          <Handlers>
            <Click>
              <Call Action="buffer_count" />
            </Click>
          </Handlers>
        </Feature>
      </Features>

      <Handlers>
        <Click>
          <Call Action="click_count" />
        </Click>
      </Handlers>
    </FeatureLayer>
  </Layers>
  …
 </maps:Map>
</Viewport>
…
<Variables>
 …
 <Variable Id="clickCounter" Type="Integer" Value="0" />
 <Variable Id="bufferCounter" Type="Integer" Value="0" />
 <Variable Id="started" Type="Boolean" Value="false" />
 <Variable Id="c1" Type="Object" />
 <Variable Id="c2" Type="Object" />
 <Variable Id="x1" Type="Float" Value="0" />
 <Variable Id="x2" Type="Float" Value="0" />
 <Variable Id="y1" Type="Float" Value="0" />
 <Variable Id="y2" Type="Float" Value="0" />
</Variables>

<Actions>
 …
 <Action Id="click_count">
  <Expression>clickCounter=clickCounter+1</Expression>
  <If>
   <Expression>clickCounter==1</Expression>
   <True>
    <Expression>c1=ComponentData->getCoordinate()</Expression>
    <Expression>x1=c1->x</Expression>
    <Expression>y1=c1->y</Expression>
   </True>
  </If>
  <If>
   <Expression>clickCounter==2</Expression>
   <True>
    <Expression>c2=ComponentData->getCoordinate()</Expression>
    <Expression>x2=c2->x</Expression>
    <Expression>y2=c2->y</Expression>
    <Call Action="nextSlide" />
   </True>
  </If>
 </Action>

 <Action Id="start_click">
  <If>
   <Expression>clickCounter==0</Expression>
   <True>
    <Expression>started=true</Expression>
   </True>
```

```
    </If>
  </Action>

  <Action Id="buffer_count">
    <Expression>bufferCounter=bufferCounter+1</Expression>
  </Action>

  <Action Id="nextSlide">
    <Expression>Navigator->next()</Expression>
  </Action>
 </Actions>
</SlideTemplate>
```

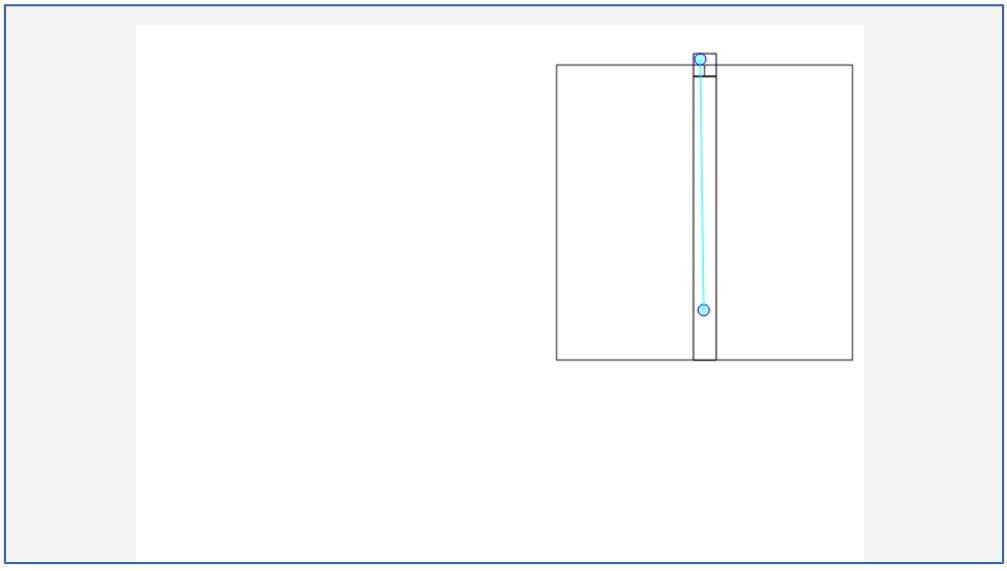
Nyní se konečně dostáváme k získávání souřadnic prvního a druhého kliknutí – tedy začátku a konce čáry. Získávají se v Action „click_count“. Pomocí funkce ComponentData->getCoordinate() se nejdříve získá proměnná c1 představující souřadnice aktuálního kliku a z ní následně souřadnice X a Y. Proto se musí v podmínce kontrolovat, který klik byl aktuální, zda první nebo druhý, podle toho se naplní proměnné x1 a y1 nebo x2 a y2.

```
<Expression>c1=ComponentData->getCoordinate()</Expression>
<Expression>x1=c1->x</Expression>
<Expression>y1=c1->y</Expression>
```

Souřadnice se měří od horního levého rohu mapy. Jestliže celý element Map (bílá plocha) má rozměry 640x640 pixelů, pak první a druhý bod čáry mohou mít např. následující souřadnice:

```
<Variable Id="x1" Type="Float">496.000</Variable>
<Variable Id="y1" Type="Float">30.0000</Variable>
<Variable Id="x2" Type="Float">499.000</Variable>
<Variable Id="y2" Type="Float">251.000</Variable>
```

Délku čáry v pixelech pak lze vypočítat např. pomocí Pythagorovy věty na 221,02 pixelů.



Možné komplikace při získávání souřadnic kliknutí

Výše zmíněný kód získává souřadnice obou bodů v Action „click_count“, která se volá v Handleru náležícím elementu Map. Avšak funkce se změní, když se volání kódu pro zjišťování souřadnic např. pro první bod přesune např. do Action „start_click“. Ta je volána z Handleru patřícímu elementu Feature „start_point“. Kód elementu Actions by pak vypadal následovně:

```
<Actions>
 …
 <Action Id="click_count">
  <Expression>clickCounter=clickCounter+1</Expression>
  <If>
   <Expression>clickCounter==1</Expression>
  </If>
  <If>
   <Expression>clickCounter==2</Expression>
   <True>
    <Expression>c2=ComponentData->getCoordinate()</Expression>
    <Expression>x2=c2->x</Expression>
    <Expression>y2=c2->y</Expression>
    <Call Action="nextSlide" />
   </True>
  </If>
 </Action>

 <Action Id="start_click">
  <If>
   <Expression>clickCounter==0</Expression>
   <True>
    <Expression>started=true</Expression>
    <Expression>c1=ComponentData->getCoordinate()</Expression>
    <Expression>x1=c1->x</Expression>
    <Expression>y1=c1->y</Expression>
   </True>
  </If>
 </Action>
 …
<Actions>
```

Změna se zdá logická, neboť souřadnice prvního bodu by se získávali v rámci Action, která je volána po kliknutí do malého čtverce, kam se má právě kliknout prvním klikem. Avšak výsledné souřadnice při stejném umístění kliků budou vypadat jinak:

```
<Variable Id="x1" Type="Float">6.00000</Variable>
<Variable Id="y1" Type="Float">10.00000</Variable>
<Variable Id="x2" Type="Float">499.000</Variable>
<Variable Id="y2" Type="Float">251.000</Variable>
```

Odlišují se souřadnice prvního bodu. Důvod je ten, že jeho souřadnice jsou získávány v Action, která je volána Handlerem elementu Feature „start_point“ (malý čtverec na horní hranici velkého čtverce) a ne Handlerem celé mapy. Souřadnice se přitom zjišťují relativně k prvku, ze kterého je příslušná Action volána! Souřadnice 6 a 10 jsou tedy počítány od levého horního rohu Feature „start_point“. Pokud bychom chtěli zjistit souřadnice bodu v rámci celé mapy, pak bychom museli k x1 a y1 připočíst souřadnice tohoto levého horního rohu malého čtverce. Souřadnice druhého bodu zůstaly stejné, neboť kód pro zjištění jeho souřadnic zůstal v Handleru náležícím elementu Map. Jsou tedy počítány v rámci celé mapy.

Zachycení reakce uživatele z klávesnice

Zpětná reakce uživatele na test nemusí využít jen myš ke klikání nebo výběru položek. Klávesnice nemusí sloužit jen pro psaní textu do políček připravených formulářů. Chování slide se dá ovládat i pomocí kláves na klávenici. Je k tomu potřeba nadefinovat speciální Handler s názvem Shortcut, který definuje, stisk které klávesy se bude monitorovat.

Každá klávesa klávesnice již má definován svůj Shortcut, který jediný může jedině právě tuto klávesu zachytit. Kdybychom proto chtěli monitorovat celou klávesnici, museli bychom použít příslušné elementy Shortcut pro všechny klávesy! Zde uvádíme seznam zachycovatelných kláves a všech příslušných jmen pro element Shortcut.

Kláve sa	Název elementu Shortcut	Klávesa	Název elementu Shortcut	Klávesa	Název elementu Shortcut
A	<Shortcut Key="A">	numerická 0	<Shortcut Key="Num0">	mezerník	<Shortcut Key="Spacebar">
B	<Shortcut Key="B">	numerická 1	<Shortcut Key="Num1">	Enter	<Shortcut Key="Enter">
C	<Shortcut Key="C">	numerická 2	<Shortcut Key="Num2">	Escape	<Shortcut Key="Esc">
D	<Shortcut Key="D">	numerická 3	<Shortcut Key="Num3">	Page Up	<Shortcut Key="PgUp">
E	<Shortcut Key="E">	numerická 4	<Shortcut Key="Num4">	Page Down	<Shortcut Key="PgDn">
F	<Shortcut Key="F">	numerická 5	<Shortcut Key="Num5">	Tab	<Shortcut Key="Tab">
G	<Shortcut Key="G">	numerická 6	<Shortcut Key="Num6">	šipka vlevo	<Shortcut Key="Left">
H	<Shortcut Key="H">	numerická 7	<Shortcut Key="Num7">	šipka vpravo	<Shortcut Key="Right">
I	<Shortcut Key="I">	numerická 8	<Shortcut Key="Num8">	šipka nahoru	<Shortcut Key="Up">
J	<Shortcut Key="J">	numerická 9	<Shortcut Key="Num9">	šipka dolů	<Shortcut Key="Down">
K	<Shortcut Key="K">	F1	<Shortcut Key="F1">	Backspac e	<Shortcut Key="Backspace ">
L	<Shortcut Key="L">	F2	<Shortcut Key="F2">	Delete	<Shortcut Key="Del">
M	<Shortcut Key="M">	F3	<Shortcut Key="F3">	Insert	<Shortcut Key="Ins">
N	<Shortcut Key="N">	F4	<Shortcut Key="F4">	End	<Shortcut Key="End">
O	<Shortcut Key="O">	F5	<Shortcut Key="F5">	Home	<Shortcut Key="Home">
P	<Shortcut Key="P">	F6	<Shortcut Key="F6">		
Q	<Shortcut Key="Q">	F7	<Shortcut Key="F7">		
R	<Shortcut Key="R">	F8	<Shortcut Key="F8">		

S	<Shortcut Key="S">	F9	<Shortcut Key="F9">
T	<Shortcut Key="T">	F10	<Shortcut Key="F10">
U	<Shortcut Key="U">	F11	<Shortcut Key="F11">
V	<Shortcut Key="V">	F12	<Shortcut Key="F12">
W	<Shortcut Key="W">		
X	<Shortcut Key="X">		
Y	<Shortcut Key="Y">		
Z	<Shortcut Key="Z">		

Ukázka:
Příklad s využitím tohoto typu Handleru je uveden v kapitole ImageSequenceLayer. Zde je pouze zkrácená část. Oba elementy Shortcut zapisují proměnnou a volají další funkce. Všimněte si, že nikde není definována funkcionalita elementu Shortcut a není v nich uvedena ani klávesa, která je zachycována. Klávesa je totiž definována již jménem elementu Shortcut, jména elementu Shortcut tedy jsou libovolná! Každé klávese klávesnice je již předem přiřazen název elementu Shortcut, který je potřeba pro její zachycení.

```
<SlideTemplate>
  <Viewport>
    …
  </Viewport>

  <Handlers>
    <Shortcut Key="Left">
      <Expression>key=1</Expression>
      <Call Action="key_press" />
      <Call Action="next_tile" />
    </Shortcut>
    <Shortcut Key="Right">
      <Expression>key=2</Expression>
      <Call Action="key_press" />
      <Call Action="next_tile" />
    </Shortcut>
  </Handlers>

  <Variables>
    <Variable Id="rightCount" Type="Integer" Value="0" />
    <Variable Id="leftCount" Type="Integer" Value="0" />
    <Variable Id="key" Type="Integer" Value="0" />
  </Variables>

  <Actions>
    <Action Id="key_press">
      …
    </Action>
    <Action Id="next_tile">
      …
    </Action>
  </Actions>
</SlideTemplate>
```