

# Řízení zobrazování zastávkových informačních panelů

## Obsah

|     |   |    |
|-----|---|----|
| 1   | Úvod .....  | 3  |
| 2   | Základní koncepce .....   | 4  |
| 2.1 | Stromová struktura vstupních dat.....                                     | 4  |
| 2.2 | Oddělení obsahu od rozložení a formátu zobrazování .....                  | 5  |
| 2.3 | Vstupních data ve formátu XML.....  | 5  |
| 3   | Popis vstupních dat pro panely - XML .....                                | 6  |
| 3.1 | Rozložení obrazovek odjezdů - departureLayouts.xml .....                  | 6  |
| 3.2 | Definice scénářů - scenarios.xml .....                                    | 7  |
| 4   | Popis komunikačního protokolu mezi dispečinkem a informačními panely..... | 9  |
| 4.1 | Základní charakteristika komunikace .....                                 | 9  |
| 4.2 | Základní nastavení komunikace .....                                       | 9  |
| 5   | Popis služeb .....  | 10 |
| 5.1 | 0 – Alive_status .....  | 10 |
| 5.2 | 10 – UpdateDepartures: aktualizace informací o odjezdech .....            | 10 |
| 5.3 | 11 – DateTimeSync: synchronizace data a času .....                        | 12 |
| 5.4 | 12 – TextMsg: textová zpráva.....   | 12 |
| 5.5 | 13 – UpdateWindowContent: aktualizace obsahu okna.....                    | 13 |
| 5.6 | 14 – ActivateScenario: aktivace scénáře .....                             | 13 |
| 5.7 | 15 – Schedule: aktualizace rozvrhu zobrazování.....                       | 14 |
| 5.8 | 16 – AddOrEditScenario: dynamická aktualizace vstupních dat scénáře.....  | 14 |
| 6   | SW simulátor a skladač pro přípravu dat .....                             | 15 |

## 1 Úvod

Tento dokument popisuje způsob řízení zobrazování zastávkových informačních panelů, včetně popisu komunikačních protokolů, formátu a struktury vstupních dat. Při návrhu bylo snahou vytvořit co možná nejuniverzálnější řešení, které umožní rozmanité zobrazování na panelech a dynamické změny obsahu (řízené vzdáleně z dispečinku), bez nutnosti přehrávat firmware jednotek vykonávající zobrazení.

Součástí dokumentu je představení základní koncepce. Na toto navazuje popis vstupních dat, ve formátu XML, které tvoří základ celého řešení. Posléze je popsán komunikační protokol mezi dispečinkem a informačními panely (simulátorem). Na závěr je stručně představen SW skladač pro tvorbu vstupních dat, který slouží zároveň jako SW simulátor informačních panelů.

## 2 Základní koncepce

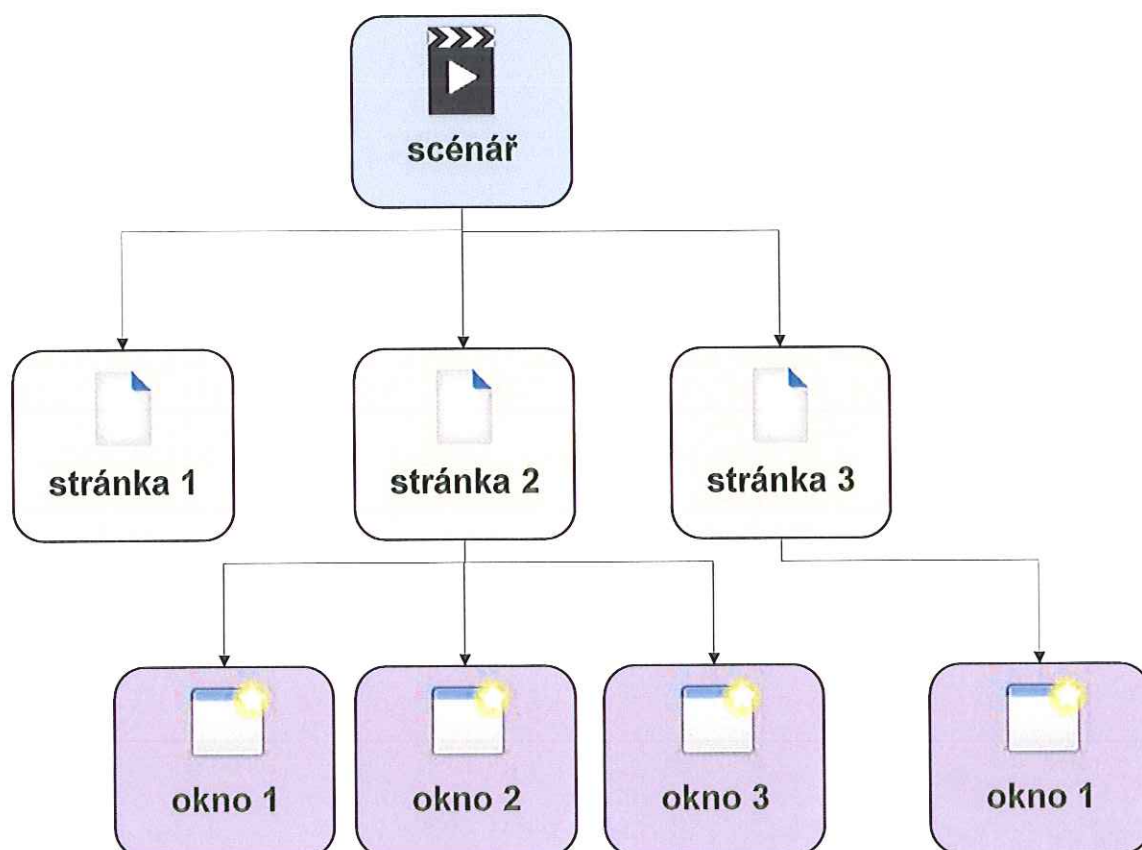
Cílem bylo navrhnout řešení, které umožní „klasické“ informace o odjezdech spojů prokládat s různými reklamními informacemi, mimořádnými informacemi ze strany dispečinku apod. Tzn. zobrazovat také libovolnou grafiku a texty včetně drobných animací.

Základní koncepce je shrnuta v následujících bodech:

### 2.1 Stromová struktura vstupních dat

Vstupní data formovaná do stromové struktury (viz obrázek) umožňují rozdělit komplexní problém zobrazování na nezávislé elementy, ke kterým je poté možno přistupovat samostatně. Tzn. přiřazovat jim parametry a řídit/aktualizovat obsah a animace na několika úrovních abstrakce.

Kořenovým elementem je **scénář**, ten obsahuje jednu a více **stránek**. Při vykonávání scénáře jsou stránky cyklicky střídány. **Stránky** mohou být *různého typu (odjezdy, komponované apod.)* a v závislosti na něm mohou obsahovat žádné nebo několik **oken**. Okna jsou v hierarchii nejnižší a mohou být také různého typu (textové, víceřádkový text, led mapa, ...) nicméně v případě že zobrazení stránky vyplývá už z její povahy (např. stránka typu *odjezdy*) nemusí poté okna obsahovat vůbec.



Obr. 1: Stromová struktura vstupních dat

## 2.2 Oddělení obsahu od rozložení a formátu zobrazování

Vstupní data slouží jako jistá forma šablon. Definují pomyslnou pozici, formát, fonty, způsob animace a rozložení budoucího obsahu. Obsah je poté možno měnit a řídit dynamicky. Oddělení rozložení a obsahu umožňuje minimalizovat množství přenášených dat, kdy datově objemnější šablony nemusí být často aktualizovány a naopak pouze dynamicky se měnící obsah (není tolik datově objemný) může být aktualizovaný často.

Navíc definice vstupních dat obsahuje i výchozí hodnotu obsahu pro jednotlivé elementy. Toho je možno využít při zobrazování statických dat, kdy není nutné (příp. žádoucí) obsah do šablon vždy dodávat.



Obr. 2: Ukázka koncepce oddělení formátu (šablony stránek) od obsahu

## 2.3 Vstupních data ve formátu XML

Vstupní data jsou ve formátu XML, a to zejména z následujících důvodů:

- Celosvětově rozšířený standard pro výměnu dat mezi aplikacemi.
- Existující podpora parserů v běžných programovacích jazycích.
- Jasně definovaná gramatika (možnost validace na *well-formed document*).
- Možnost validovat obsah dokumentu (DTD, XSD,...).
- Otevřenost a rozšiřitelnost formátu.
- Přehlednost – význam dat je možné číst a chápat i v jejich zdrojové podobě.

### 3 Popis vstupních dat pro panely - XML

V této kapitole budou představeny XML soubory, které jsou součástí vstupních dat. Každý XML soubor má přiložený XSD soubor (XML Schema Definition), jenž umožňuje validovat obsah dokumentu pomocí standardu W3C XML Schema.

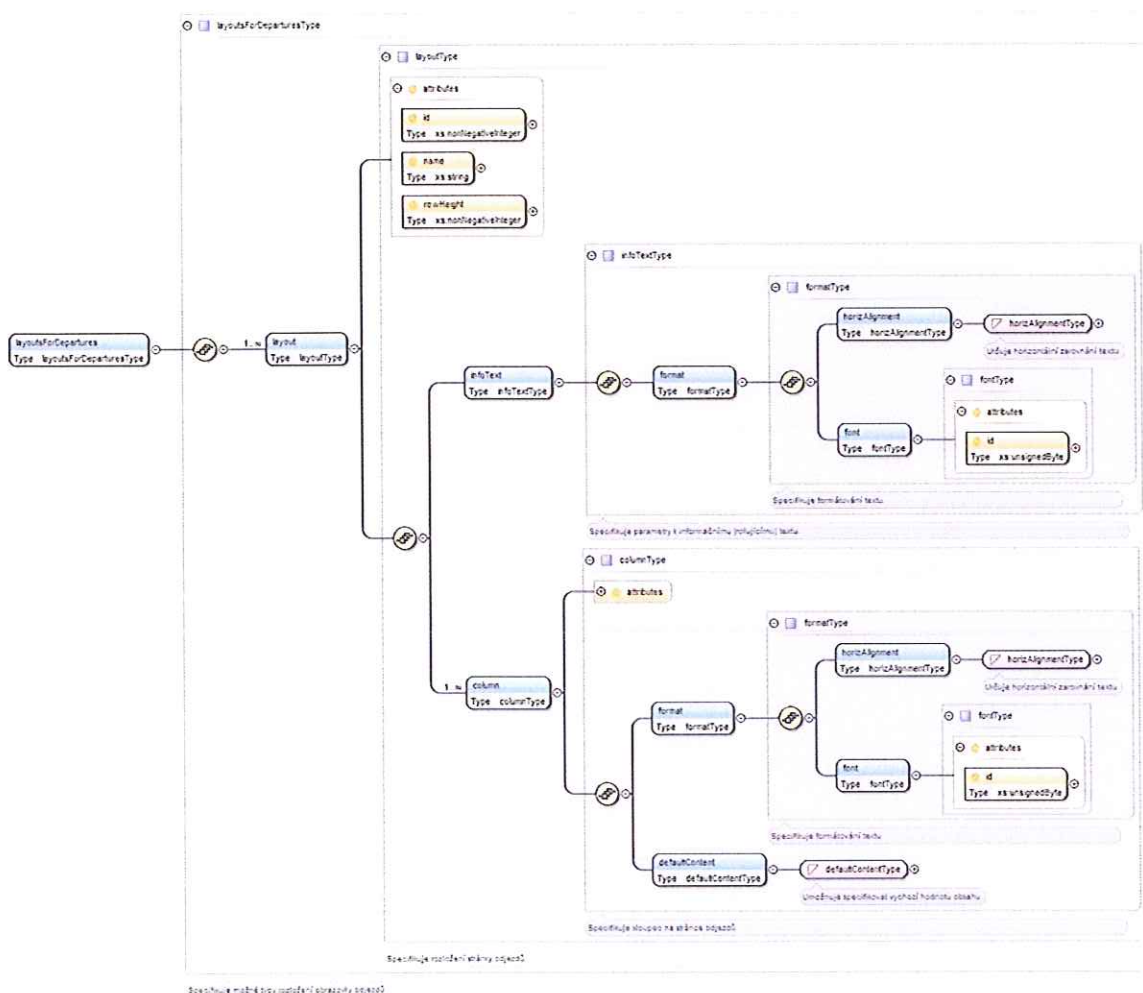
#### 3.1 Rozložení obrazovek odjezdů - departureLayouts.xml

Soubor *departureLayouts.xml* obsahuje definici rozložení obrazovek odjezdů. Toto zahrnuje rozměry a typ sloupců, formátování textů na řádcích, výchozí hodnotu obsahu sloupců apod.

Byl vytvořen XSD soubor *validDepartLayouts.xsd*, jenž specifikuje povinné atributy a elementy, datové typy a rozsah jejich hodnot včetně slovního popisu.

Kromě samotného XSD souboru je možné detailní popis struktury XML najít v souboru *validDepartLayouts.pdf*, který je generovaný přímo z XSD souboru.

Struktura souboru je zachycena také na obrázku níže.



Obr. 3: Struktura souboru departureLayouts.xml

### 3.2 Definice scénářů - scenarios.xml

Soubor *scenarios.xml* obsahuje kompletní definici scénářů.

Byl vytvořen XSD soubor *validScenarios.xsd*, jenž specifikuje povinné atributy a elementy, datové typy a rozsah jejich hodnot včetně slovního popisu.

Kromě samotného XSD souboru je možné detailní popis struktury XML najít v souboru *validScenarios.pdf*, který je generovaný přímo z XSD souboru.

Struktura souboru je zachycena také na obrázku níže.



## 4 Popis komunikačního protokolu mezi dispečinkem a informačními panely

### 4.1 Základní charakteristika komunikace

Komunikace je realizována mezi serverem dispečinku (dále Sprinter) a zastávkovými informačními panely případně SW simulátorem zastávkových informačních panelů (dále simulátor).

SW simulátor i reálný zastávkový informační panel využívají stejný komunikační protokol, pouze přenosová cesta se může lišit (sít' na bázi TCP/IP vs. radiová sít').

Komunikace mezi Sprinterem a simulátorem je realizována v síti využívající rodinu protokolů TCP/IP. Jako protokol transportní vrstvy je použit bezstavový - nespojovaný, nespolehlivý protokol UDP (User Datagram Protocol), který v případě ztráty paketu nezajišťuje spolehlivost opakovaným odesláním.

V případě komunikace na se simulátorem je v současné době komunikace pouze jednosměrná, ve směru server -> simulátor.

### 4.2 Základní nastavení komunikace



Simulátor přijímá příchozí zprávy na portu 5080. IP adresa simulátoru je 10.0.0.66.

Základní struktura dat je zobrazena na obr. 1.

| Počet bajtů pole | Adresa začátku pole | Název pole                     |
|------------------|---------------------|--------------------------------|
| 2                | 0                   | Adresa panelu                  |
| 2                | 2                   | Služba                         |
| 2                | 4                   | Čítač                          |
| 0-n              | 5                   | Data (CP-1250, oddělovač 0x1E) |

Obr. 2: Základní struktura dat

| Název pole           | Popis  |
|----------------------|--|
| <i>Adresa panelu</i> | Udává adresu panelu (binárně). Endianita: Big-endian.<br>Adresa s hodnotou 0xFFFF značí generální adresu (broadcast na všechny panely) |
| <i>služba</i>        | Udává číslo služby (binárně). Endianita: Big-endian.   |
| <i>čítač</i>         | Udává číslo čítače (binárně). Ten je inkrementován s každou novou zprávou.<br>Endianita: Big-endian.                                   |
| <i>data</i>          | Data jsou v textovém formátu s kódováním CP-1250. Oddělovačem jednotlivých polí je hodnota 0x1E.                                       |

## 5 Popis služeb

### 5.1 0 – Alive\_status

**Číslo služby:** 0

**Charakteristika:** Zatím neimplementováno.

**Formát dat:**

### 5.2 10 – UpdateDepartures: aktualizace informací o odjezdech

**Číslo služby:** 10

**Charakteristika:** Smyslem služby je aktualizovat obsah pro stránky typu „odjezdy“.. Zpráva obsahuje ID rozložení (layout) a dále informace specifické k danému rozložení. Rozložení stránky (layout) je definován v souboru *departureLayouts.xml* pod příslušným ID. Zřetěžené informace obsahují informace pro jednotlivé sloupce jednoho řádku. Informace je interpretována na základě typu sloupce (znovu dáno definicí rozložení v xml souboru), viz také dále. Zalamování řádků není řešeno, jelikož počet sloupců (polí) v řádku je dán již definicí rozložení (layoutu).

### Formát dat:

Server -> Simulátor:

|           |    |            |    |  |
|-----------|----|------------|----|--|
| layout ID | 1E | poč. řádků | 1E | zřetězené informace (obsah) odjezdů (viz dále) |
|-----------|----|------------|----|--|

V současné době jsou definovány následující typy sloupců:

| Typ sloupce      | Popis   |
|------------------|---|
| text             | Zaslaná informace je zobrazena tak jak byla získána (bez další transformace)                |
| finDestinationID | Zaslaná informace obsahuje ID kódu cíle a je překládáno na název dle souboru „findest.des“. |

### Výchozí hodnota sloupce:

Každý sloupec může obsahovat v definici (soubor .xml) i výchozí hodnotu obsahu. V případě, že je ve zprávě na odpovídající pozici prázdný řetězec je použita výchozí hodnota z definice.

Konkrétní příklady řádků pro obsahu odjezdů:

**Layout 1 – základní zobrazení odjezdů: informace pro jeden řádek**

|                 |    |                          |    |                    |    |                  |    |               |
|-----------------|----|--------------------------|----|--------------------|----|------------------|----|---------------|
| linka<br>(text) | 1E | kód cíle<br>(finDestin.) | 1E | Invalida<br>(text) | 1E | Symbol<br>(text) | 1E | Čas<br>(text) |
|-----------------|----|--------------------------|----|--------------------|----|------------------|----|---------------|

**Layout 2 – nepravidelný provoz: informace pro jeden řádek**

|                 |    |                           |
|-----------------|----|---------------------------|
| linka<br>(text) | 1E | Prázdný<br>řetězec (text) |
|-----------------|----|---------------------------|

- na straně panelu je doplněn řádek, je nahrazen prázdný řetězec textem „nepravidelný provoz“

### Poznámka:

Některé používané speciální znaky:

- Symbol invalidy je uložen v základním fontu a je určen znakem „t“ (dle CP-1250 je to kód 0xFE)
- symbol „±“ (dle CP-1250 je to kód 0xB1)
- Symbol autobus je uložen v základním fontu a je určen znakem „J“ (dle CP-1250 je to kód 0xDE)

### 5.3 11 – DateTimeSync: synchronizace data a času

**Číslo služby:** 11

**Charakteristika:** Smyslem služby je synchronizovat čas na panelu.

**Formát dat:**

Server -> Simulátor:

dateTime

| Název pole      | Popis  |
|-----------------|--|
| <i>dateTime</i> | Textový řetězec specifikující aktuální datum/čas. Formát: "ddMMyyyyHHmmss" |

### 5.4 12 – TextMsg: textová zpráva

**Číslo služby:** 12

**Charakteristika:** Smyslem služby je zobrazit na panelu (v některém ze spodních řádků) informační rolující text. Text na panelu je smazán buď na základě požadavku na zobrazení nového textu nebo po uplynutí nastavené doby zobrazení.

**Formát dat:**

Server -> Simulátor:

řádek 1E timeMin 1E text

| Název pole     | Popis   |
|----------------|---|
| <i>řádek</i>   | Index řádku, počítáno od spodního (posledního) řádku na stránce, místo kterého se má informační text zobrazit                                       |
| <i>timeMin</i> | Číselná hodnota (dekadický formát). Specifikuje dobu v minutách po kterou má být text zobrazen. <u>Hodnota 0 znamená neomezenou dobu zobrazení.</u> |
| <i>text</i>    | Text, který má být na panelu zobrazen. <u>Prázdný textový řetězec ruší aktuálně zobrazený text bez ohledu na hodnotu <i>timeMin</i>.</u>            |

## 5.5 13 – UpdateWindowContent: aktualizace obsahu okna

Číslo služby: 13

**Charakteristika:** Smyslem služby je aktualizovat obsah konkrétního okna.

**Formát dat:**

Server -> Simulátor:

|                    |    |     |    |       |
|--------------------|----|-----|----|-------|
| deskriptor<br>okna | 1E | typ | 1E | obsah |
|--------------------|----|-----|----|-------|

| Název pole             | Popis  |
|------------------------|--|
| <i>deskriptor okna</i> | Jednoznačná identifikace okna napříč všemi scénáři a v nich obsažených stránek.<br>Formát: „scenarioID,pageID,windowID,“<br>Oddělovačem je čárka, hodnota je složena z identifikace scénáře, stránky i okna na stránce. Vychází se z předpokladu, že ID je v rámci dané viditelnosti (scénáře, stránky ve scénáři a okna na stránce) unikátní. |
| <i>type</i>            | Znak popisující typ okna: <ul style="list-style-type: none"><li>- T.....textové okno</li><li>- M....led mapa (zatím neimplementováno)</li><li>- L.....víceřádkové textové okno</li></ul>   |
| <i>obsah</i>           | Obsah je závislý na typu okna. Pro textově založená okna je obsahem textový řetězec.   |

## 5.6 14 – ActivateScenario: aktivace scénáře

Číslo služby: 14

**Charakteristika:** Smyslem služby je aktivovat některý ze scénářů na panelu.

**Formát dat:**

Server -> Simulátor:

|             |    |         |
|-------------|----|---------|
| scenario ID | 1E | restart |
|-------------|----|---------|

| Název pole         | Popis   |
|--------------------|---|
| <i>scenario ID</i> | Číselná hodnota (dekadický formát). Specifikuje id scénáře.   |
| <i>restart</i>     | Určuje jestli s přijetím zprávy má rovněž dojít k restartu vykonávání scénáře. <u>Hodnota 0 znamená, že k restartu dojít nemá. Hodnota 1 značí restart scénáře.</u> |

## 5.7 15 – Schedule: aktualizace rozvrhu zobrazování

*Číslo služby:* 15

*Charakteristika:* Zatím neimplementováno. Časová posloupnost zobrazování, aktivace scénářů, vyplňování obsahů oken apod.

## 5.8 16 – AddOrEditScenario: dynamická aktualizace vstupních dat scénáře

*Číslo služby:* 16

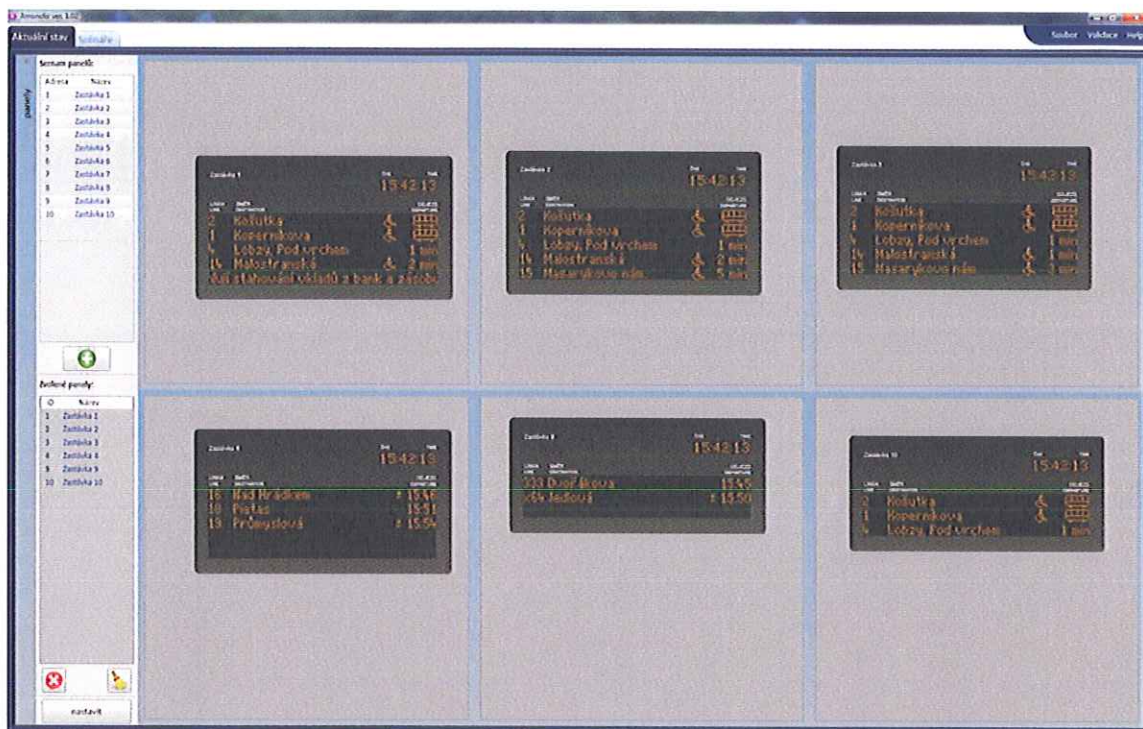
*Charakteristika:* Zatím neimplementováno. Umožňuje dynamicky zaslat komprimovaný XML soubor s definicí scénáře.

## 6 SW simulátor a skladač pro přípravu dat

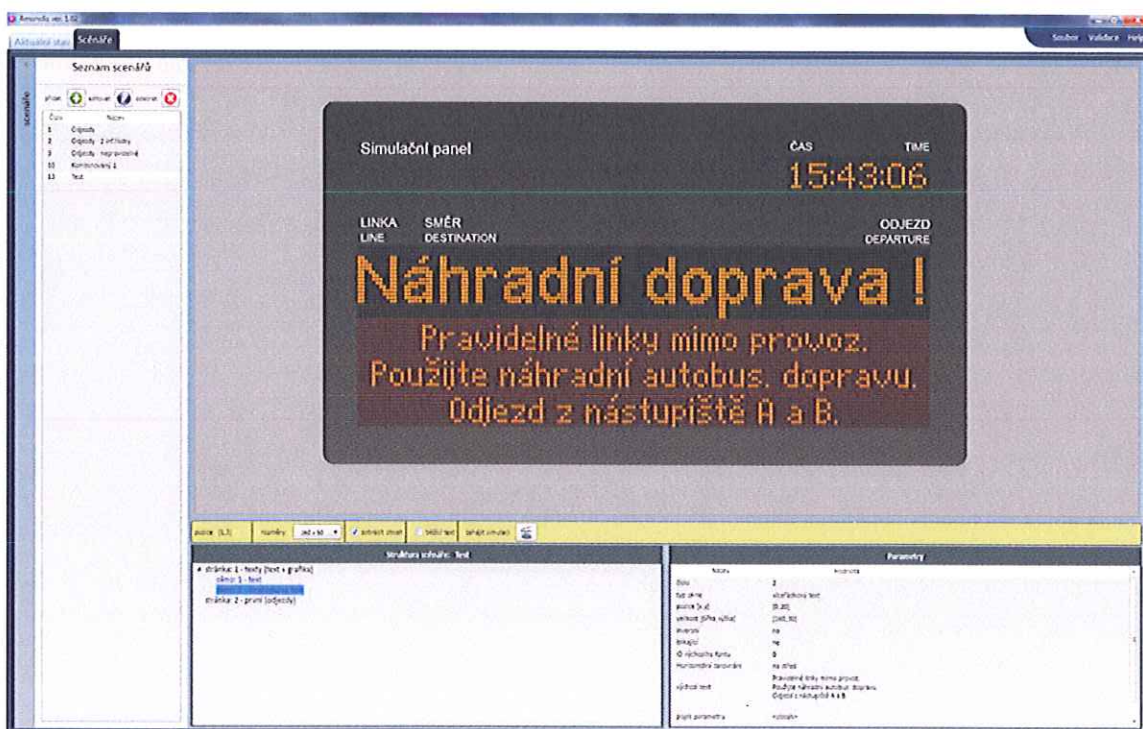
Bude doplněno....

SW umožňuje:

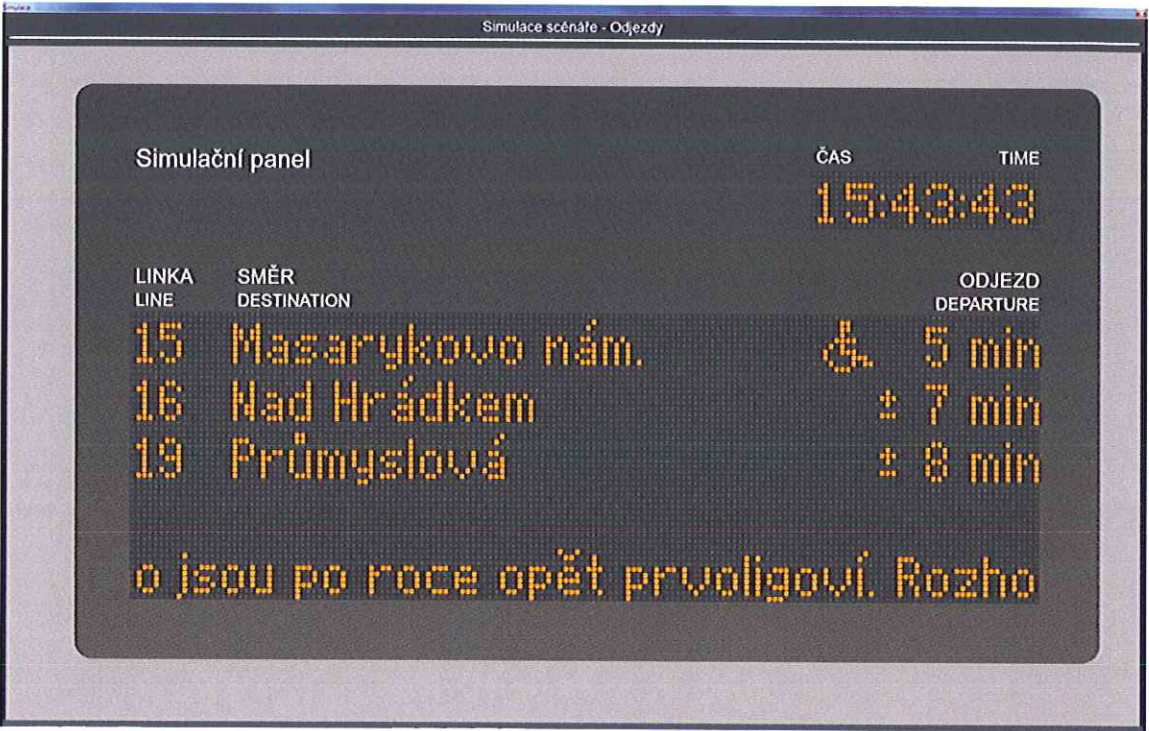
zobrazovat aktuální stav panelů:



vytvářet scénáře a vstupní data pro panely:



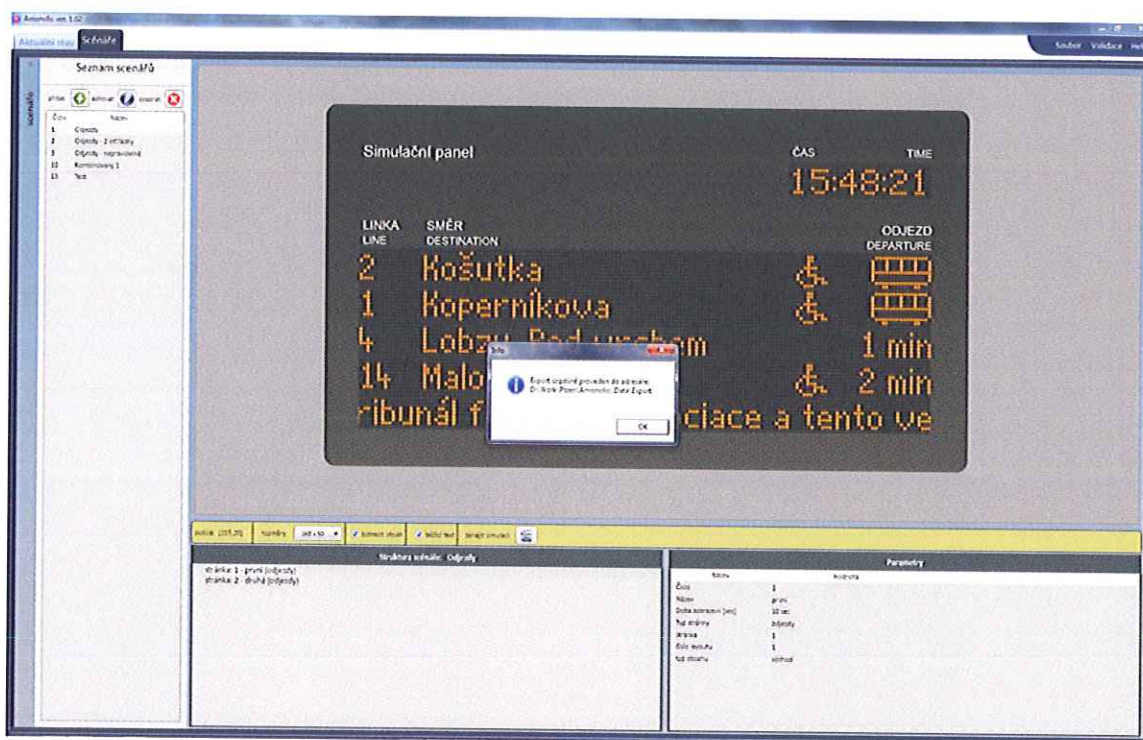
simulovat zobrazení v procesu tvorby scénáře:



validovat vytvářená data:



Exportovat složená data do XML formátu (tyto jsou dále nahrávány na dispečink a zastávkové panely):



Prohlížet, modifikovat, vytvářet a testovat fonty:

