



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

GRAFICKÉ VIZUALIZAČNÍ KOMPONENTY PRO SYSTÉM COMES

GRAPHICAL VISUALIZATION COMPONENTS FOR COMES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LADISLAV LEBEDA

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. FRANTIŠEK ZEŽULKA, CSc.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Ladislav Lebeda

ID: 146886

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Grafické vizualizační komponenty pro systém COMES

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření grafických vizualizačních komponent pro systém operativního řízení výroby COMES. Je požadováno, aby student

1. Provedl rešerši komerčních i nekomerčních komponent pro vytvoření vizualizace
2. Vybral a navrhl vhodné typy komponent
3. Provedl naprogramování komponent v jazyce SVG
4. Provedl prezentaci realizovaných komponent

DOPORUČENÁ LITERATURA:

[1] Mgr. Martin Dostál, Ph.D. - <http://dostal.inf.upol.cz/uro.html>

[2] COMPAS AUTOMATIZACE. Výrobní informační systém COMES verze 3 – úvod do systému. [s.l.], Compas Automatizace spol. s r. o., ©2010

Termín zadání: 10.2.2014

Termín odevzdání: 26.5.2014

Vedoucí práce: prof. Ing. František Zezulka, CSc.

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce je zaměřena na vytvoření grafických vizualizačních komponent pro aplikaci COMES Dashboard, systému operativního řízení výroby COMES.

Nejprve se zabývá posouzením uživatelského rozhraní systému COMES, analýzou komerčních a nekomerčních komponent pro tvorbu dashboard vizualizace a výběrem vhodných typů elementů, jež jsou na řídicím panelu použity.

Dále je zaměřena na vlastní návrh grafické podoby jednotlivých komponent v jazyce SVG. Funkcionalita těchto objektů je zajištěna pomocí programovacího jazyka JavaScript. Pro zjednodušení programové části komponent a rozšíření podpory webových prohlížečů je využita javascriptová knihovna jQuery.

Klíčová slova

COMES, dashboard, vizualizační komponenty, MES, SVG, JavaScript, jQuery.

Abstract

This work is aimed to enable creation of graphic visualization components for application COMES Dashboard, which is part of operative control system COMES.

It deals with the assessment of the user interface system COMES, the analysis of commercial and non-commercial component for creating dashboard visualization and selecting the most appropriate element types, which are used on the dashboard.

Furthermore, the thesis is focused on the design of individual objects in a graphical form of the SVG language. The functionality of this object is provided by the JavaScript programming language. To simplify the programming of component and extension support for Web browsers is used the JavaScript library jQuery.

Keywords

COMES, dashboard, visualization components MES, SVG, JavaScript, jQuery.

Bibliografická citace:

LEBEDA, L. *Grafické vizualizační komponenty pro systém COMES*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 65 s. Vedoucí bakalářské práce byl prof. Ing. František Zezulka, CSc.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Grafické vizualizační komponenty pro systém COMES jsem vypracoval samostatně pod vedením vedoucího bakalářské práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 20. května 2014

.....
podpis autora

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce panu prof. Ing. Františku Zezulkovi, CSc. za věnovaný čas a cenné rady. Také bych rád poděkoval panu Ing. Aleši Stehnovi za hodnotné konzultace. Dále bych rád poděkoval mé rodině a přátelům za podporu a pomoc při studiu.

V Brně dne: 20. května 2014

.....
podpis autora

Obsah:

1.	Úvod.....	11
2.	COMES.....	13
2.1	Funkce nabízené systémem COMES.....	13
2.2	Moduly systému COMES.....	14
2.2.1	COMES Logon.....	14
2.2.2	COMES Historian	15
2.2.3	COMES Modeller.....	16
2.2.4	COMES Traceability.....	16
2.2.5	COMES Batch.....	16
2.3	COMES OEE	17
2.3.1	Ukazatel OEE.....	17
2.3.2	Funkce COMES OEE.....	17
3.	Dashboard.....	19
3.1	COMES Dashboard.....	19
3.2	Příklady dashboardů v systému COMES.....	19
3.3	Dashboards používané v jiných systémech	20
4.	Analýza komponent pro tvorbu dashboard vizualizace	22
4.1	Graf.....	22
4.2	Měřidlo.....	23
4.3	Tabulka	23
4.4	Datové pole	24
4.5	Akční komponenty	24
4.6	Vhodné typy komponent	24
5.	Jazyk SVG.....	25
5.1	Vektorová a bitmapová grafika	25
5.1.1	Bitmapová (pixelová) grafika	25
5.1.2	Vektorová grafika.....	25
5.2	Výhody SVG.....	26
5.3	Skladba SVG dokumentu	26
5.3.1	Základní kostra.....	27
5.3.2	Základní tvary objektů.....	27
5.3.3	Vlastnosti SVG objektů.....	28
5.4	SVG Editory.....	28
5.5	SVG a scriptování	28
6.	JavaScript	29
6.1	Tvorba skriptu	29
6.2	Obsah skriptu	29
6.3	JavaScriptové knihovny	30
6.4	jQuery	30
6.4.1	Načtení jQuery do HTML	30
6.4.2	Funkce symbolu „\$“	31
6.4.3	CSS v jQuery.....	31

7.	Návrh komponent	32
7.1	Skladba komponent	32
7.2	Příklad struktury programové části komponenty	33
8.	Prezentace realizovaných komponent	34
8.1	Button	34
8.2	Start-Stop Buttons	36
8.3	Reset Button	37
8.4	ONxOFF Switch	39
8.5	Linear Potenciometer	40
8.6	Column Signalization	41
8.7	RedGreen Signalization	43
8.8	Pilot Light	45
8.9	Beacon 1	46
8.10	Gauge_1	48
8.11	Gauge_2	49
8.12	Gauge_3_thermometer	51
8.13	Gauge_5	53
8.14	Gauge_6	56
9.	Závěr	59

Seznam obrázků

Obrázek 1: Hierarchie podnikového výrobního systému.....	11
Obrázek 2: Přihlašovací obrazovka klienta systému COMES (modul COMES Logon) [5]	14
Obrázek 3: Příklad trendu v COMES Historian [6].....	15
Obrázek 4: Příklad alarmů v COMES Historian [6].....	15
Obrázek 5: Výpis materiálů v COMES Traceability [8].....	16
Obrázek 6, Obrázek 7: Vizualizace na velkoplošné TV ve výrobní hale [10]	18
Obrázek 8: Příklad Andon obrazovky systému COMES OEE [12].....	19
Obrázek 9: Dashboard, vytvořen firmou Compas, zobrazující stav výroby	20
Obrázek 10: Dashboard zobrazující souhrn práce stroje [13].....	20
Obrázek 11: Dashboard vyjadřující důležité parametry strojů [14]	21
Obrázek 12: Příklad provedení grafu [15].....	22
Obrázek 13: Příklad provedení měřidla [16]	23
Obrázek 14: Příklad tabulky použité v COMES v zakázce pro Huhtamaki.....	23
Obrázek 15: Příklad provedení datového pole.....	24
Obrázek 16: Vlevo bitmapa, vpravo vektorový obrázek	26
Obrázek 17: Komponenta Button.....	35
Obrázek 18: Komponenta Start-Stop Buttons	36
Obrázek 19: Komponenta Reset Button	38
Obrázek 20: Komponenta ONsOFF Switch	39
Obrázek 21: Komponenta Linear Potenciometer.....	40
Obrázek 22: Komponenta Column Signalization	42
Obrázek 23: Komponenta RedGreen Signalization.....	44
Obrázek 24: Možné barvy rozsvícené komponenty Pilot Light.....	45
Obrázek 25: Komponenta Beacon 1	47
Obrázek 26: Komponenta Gauge_1	48
Obrázek 27: Komponenta Gauge_2	50
Obrázek 28: Komponenta Gauge_3_thermometer	52
Obrázek 29: Komponenta Gauge_5	54
Obrázek 30: Komponenta Gauge_6	56

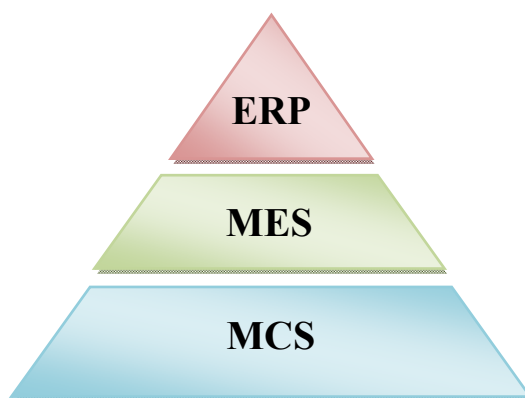
Seznam příkladů

Příklad 1: Kostra SVG dokumentu.....	27
Příklad 2: Definice obdélníku v SVG.....	27
Příklad 3: Definice kruhu v SVG	27
Příklad 4: Definice čáry v SVG.....	27
Příklad 5: Definice křivky v SVG	27
Příklad 6: Definice cesty v SVG	28
Příklad 7: Seskupení objektů v SVG.....	28
Příklad 8: Import skriptu přímo do HTML.....	29
Příklad 9: Načtení skriptu ze souboru	29
Příklad 10: Vložení skriptu do odkazu	29
Příklad 11: Vložení jQuery knihovny do HTML pomocí nahrání z vlastního souboru	30
Příklad 12: Načtení jQuery knihovny do HTML z Google	30
Příklad 13: Použití \$ - pro objekt "BACKGROUND_BT1" je nastavena barva výplně na červenou.....	31
Příklad 14: Použití CSS v jQuery-nastavení šířky objektu StartStopButtons na 50 pixelů	31
Příklad 15: Možná struktura kódu pro deklaraci komponent v HTML.....	33
Příklad 16: Obslužná funkce komponenty Button.....	35
Příklad 17: Obslužná funkce komponenty Start-Stop Buttons	37
Příklad 18: Obslužná funkce komponenty Reset Button	38
Příklad 19: Obslužná funkce komponenty ONxOFF Switch.....	39
Příklad 20: Obslužná funkce komponenty Linear Potenciometer.....	41
Příklad 21: Obslužná funkce vlastností komponenty Column Signalization	42
Příklad 22: Obslužná funkce světelných efektů komponenty Column Signalization.....	43
Příklad 23: Obslužná funkce vlastností komponenty RedGreen Signalization.....	44
Příklad 24: Obslužná funkce světelných efektů komponenty RedGreen Signalization	45
Příklad 25: Obslužná funkce vlastností komponenty Pilot Light.....	46
Příklad 26: Obslužná funkce barvy a světelných efektů komponenty Pilot Light	46
Příklad 27: Obslužná funkce komponenty Beacon 1	47
Příklad 28: Obslužná funkce vlastností komponenty Gauge_1	48
Příklad 29: Obslužná funkce komponenty Gauge_1	49
Příklad 30: Obslužná funkce vlastností komponenty Gauge_2	50
Příklad 31: Obslužná funkce komponenty Gauge_2	51
Příklad 32: Obslužná funkce vlastností komponenty Gauge_3_thermometer_properties.....	52
Příklad 33: Obslužná funkce komponenty Gauge_3_thermometer	53
Příklad 34: Obslužná funkce vlastností komponenty Gauge_5_properties.....	54
Příklad 35: Obslužná funkce komponenty Gauge_5	55
Příklad 36: Obslužná funkce vlastností komponenty Gauge_6_properties.....	56
Příklad 37: Obslužná funkce komponenty Gauge_6	57

1. ÚVOD

Již od počátku 20. století se převážně ve strojírenské výrobě objevují první automatizované stroje, vykonávající zprvu jednoduché činnosti. V dnešní době je automatizována většina výrobních systémů v mnoha průmyslových odvětvích, jako jsou například chemické, strojírenské, potravinářské atd.

Automatizovaný výrobní systém lze rozdělit na tři dílčí vrstvy (viz Obrázek 1): ERP (podnikový informační systém), MES (výrobní informační systém) a MCS (výrobní řídicí systém). K tomu, aby bylo dosaženo co nejlepších výrobních výsledků, je zapotřebí, aby všechny tyto tři zmiňované vrstvy fungovaly správně a pokud možno co nejefektivněji.



Obrázek 1: Hierarchie podnikového výrobního systému

Dále se v této práci budu věnovat pouze MES, a to konkrétně systému COMES, vytvořenému ve firmě Compas automatizace, spol. s.r.o..

Od takového systému se očekává, že bude elektronicky, optimálně řídit a plánovat výrobu, spravovat výrobní zdroje a postupy, sledovat samotnou výrobu, zprostředkovávat sběr a uchování výrobních dat, zobrazovat výkonnostní analýzy pro vyhodnocení úspěšnosti výroby a v neposlední řadě bude také jednoduše modifikovatelný. Dále je vhodné přehledné a logické řešení, které umožní snadnou správu. Velmi důležitým parametrem je pochopitelně také cena.

Ke splnění těchto požadovaných vlastností přispívá také grafická vizualizace komponent, které jsou použity na dashboardu, neboli řídicím panelu. Je tedy zapotřebí, aby všechny vizualizační prvky byly přehledné a fungovaly tak, jak je od nich požadováno. Jejich navržení by mělo být logické a vycházet z moderních trendů, které jsou pro všechny uživatele srozumitelné a na něž jsou zvyklí. Informace, které komponenta vyjadřuje, musí být snadno a především správně pochopitelné. Zásah do výrobního procesu tak bude rychlejší a přesnější, což znamená větší efektivitu podniku.

Výsledkem této bakalářské práce by mělo být alespoň sedm vizualizačních komponent, které v případě potřeby použijí pracovníci firmy Compas automatizace, spol. s.r.o. na dashboard informačního systému COMES.

Před vlastní tvorbou jednotlivých komponent je nejdůležitější stanovit, jaké typy elementů budeme vytvářet. Podle toho se bude odvíjet způsob vytváření grafické podoby objektů, vhodný editor, umožňující dostatečnou kreativitu, a programovací jazyk, s kterým bude zprostředkována interaktivita elementů. Dále je zapotřebí brát ohled na softwarovou podporu používaných technologií. COMES Dashboard je zobrazován prostřednictvím Internet Exploreru od firmy Microsoft. Je tedy naprosto nezbytné, aby veškeré vizualizační komponenty byly vytvořeny na základě principů, jež jsou podporovány právě tímto webovým prohlížečem.

Požadavek konzultanta, zadávajícího tuto práci, bylo vytvořit signalizační prvky, výstražné prvky, měřidla pro zobrazení číselných hodnot, objekt pro zadávání hodnot, ovládací tlačítka a vypínače. Grafy, tabulky a jiné textové způsoby prezentace informace nejsou od této práce vyžadovány.

Jako velmi výhodný grafický formát se jeví SVG. Tento formát se používá především pro tvorbu vektorové grafiky pro webové účely. Umožňuje jednoduchou tvorbu a opětovnou editaci parametrů pomocí skriptovacích jazyků (např. JavaScript). Další výhodou je libovolná změna velikosti (vhodné i pro velké obrazovky) a malá datová velikost. V dnešní době je SVG a JavaScript podporován všemi webovými prohlížeči jako například Opera, Mozilla Firefox, Google Chrome, Safari a také Internet Explorer.

2. COMES

COMES je výrobní informační systém úrovně MES. Byl vyvinut firmou Compas automatizace, především k řízení a optimalizaci výrobních procesů. V dnešní době je nabízen ve verzi 3. Má multijazykovou modulární strukturu vycházející z mezinárodních standardů MESA, ANSI/ISA 95 a ANSI/ISA 88. Využívá IT technologie společnosti Microsoft: Windows Server, SQL Server, IIS, Internet Explorer, .NET Framework, Silverlight, ASP.NET, C#. Architektura je stavěna na principu klient/server, kde klientem je webový prohlížeč Internet Explorer. [1]

Velmi užitečnou vlastností COMES je jeho flexibilita. Díky ní je možné tento systém aplikovat ke konkrétním uživatelům za krátký čas a zároveň za nízkou cenu, což zvyšuje konkurenceschopnost v tomto oboru. Další částí COMES je integrované vývojové rozhraní. To umožňuje neomezenému počtu programátorů práci i za „běhu“ aplikace. Jakékoliv změny lze tedy provádět bez zastavení systému. Nedochozí tak ke zbytečným prostojům při výrobě. [2]

Jelikož snahou firmy Compas automatizace je rozšíření COMES i na zahraniční trh, je tento MES systém standardně nabízen i v angličtině a slovenštině. Jiné jazykové lokalizace jsou pak doplněny na přání klienta. V případě zájmu je možné sestavu systému vytvořit vícejazyčně. Což se hodí nadnárodním společnostem, kde v pracovním týmu pracují lidé různých národností. [2]

Pro výrobní IT systém tohoto druhu je taktéž velmi důležité, aby dokázal komunikovat s dalšími zařízeními, jako jsou řídicí systémy, systémy značení materiálů, výstupní zařízení (tiskárny) apod. Tuto schopnost v COMES zaručuje modul CCI (COMES Communication Interface), v němž jsou obsaženy veškeré potřebné ovladače pro komunikace s ostatními zařízeními. [2]

Základem COMES je také jeho univerzálnost. Lze totiž použít na jakékoliv automatizační techniky jako například systémy od výrobců SIEMENS, ABB, B & R, Schneider, Bosch, Rockwell, Beckhoff, Omron a další. [2]

2.1 Funkce nabízené systémem COMES

- operativní plánování a řízení výroby,
- sledování výroby a materiálních toků,
- sběr, zobrazení a archivace technologických a výrobních dat,
- traceability,
- analýza dat, vyhodnocování KPI a zpracování protokolů,
- on-line vizualizace výrobních technologií,
- označování výrobků čárovými kódy (RFID, 2D),
- řízení údržby,

- řízení lidských zdrojů,
- komunikace s jinými systémy (podniková síť IT, ERP, řídicí systémy)

[3]

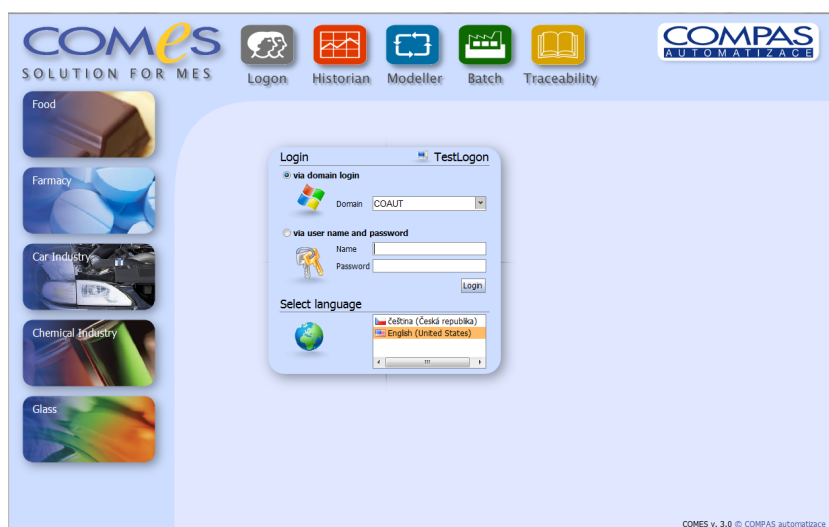
2.2 Moduly systému COMES

Jak již bylo zmíněno, systém COMES je modulárního typu. Skládá z jednotlivých modulů: COMES Logon, COMES Historian, COMES Modeller, COMES Traceability, COMES Batch. Každý z těchto modulů má jinou funkci. Zákazník může jednotlivé moduly používat samostatně, nebo je libovolně kombinovat. [4]

2.2.1 COMES Logon

Je základní stavební kámen COMES, který musí být nainstalován vždy. Zajišťuje přihlášení a správu uživatelů. Umožňuje spolupráci s ostatními moduly, přičemž nabízí také jejich diagnostiku. Dále má za úkol správu vzniklých událostí v systému. Prostřednictvím modulu Logon lze vytvářet model výrobních zařízení a technologie. Všechna data jsou zde operátorovi přístupná z jednoho místa. Obsahuje společné číselníky všech modulů vyhodnocující materiály, měrné jednotky, výčtové typy a mnoho dalších. Ve verzi COMES 3 získal modul Logon další vlastnosti jako: centrální konfigurace uživatelů a skupin, společné číselníky, centrální zálohování, časová oprávnění. Došlo také k rozšíření diagnostiky. Výhody přinesla možnost používání oblíbených odkazů. [5]

Používání modulu COMES Logon verze 3 má ale také určité softwarové požadavky. Pro server jsou nezbytné: Windows Server 2008 R2, SQL Server 2008 R2, .NET Framework 4.0 a IIS. Od klienta jsou požadovány: Internet Explorer 8, Silverlight 4 a doporučené rozlišení 1280x1024. [5]

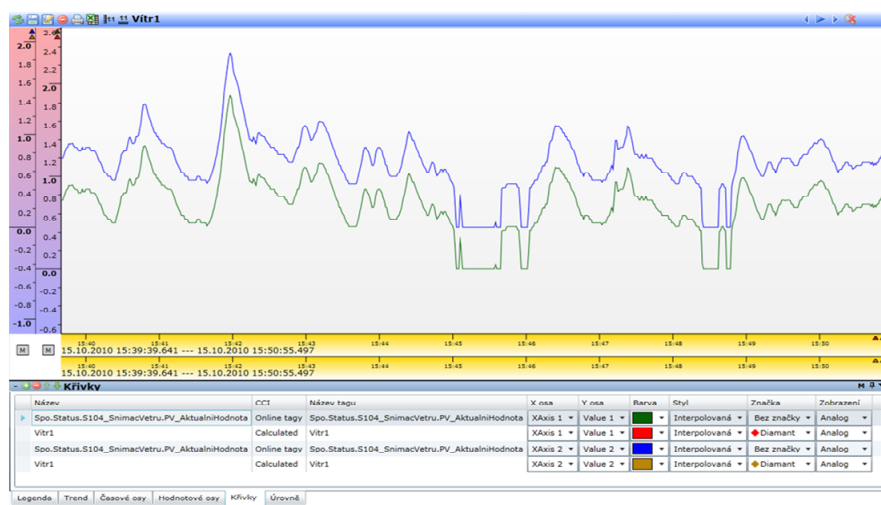


Obrázek 2: Přihlašovací obrazovka klienta systému COMES (modul COMES Logon) [5]

2.2.2 COMES Historian

Tento modul zajišťuje sběr a dlouhodobou archivaci procesních dat z řídicích systémů. Vytváří analýzu získaných dat a jejich prezentaci. Pracuje se dvěma typy dat: trendy (grafy), alarmy a hlášení (tabulky). [6]

Trendy umožňují ukládání řetězcových proměnných, provádění výpočtů s uloženými daty, různé zobrazení a práce s trendy (barvy pozadí a os, zoom, schodovité zobrazení křivek, záměrné kříže pro odčítání hodnot a nastavení úrovní, časová oprávnění, více časových a hodnotových os, export převzorkovaných/surových dat do formátu .CSV a tisk do formátu .pdf. [6]



Obrázek 3: Příklad trendu v COMES Historian [6]

U alarmů lze definovat patřičné filtry, přičemž data jsou opět exportovány do formátu .CSV a tisknuty do formátu .pdf. [6]

time	typname	text1	text3	disconnname	tv4	tv6
15.6.2010 7:52:46	Failure	SWT08XP	Terminal adapter 00:19:D1:17:28:C4 disconnected	PLC process control messages		
15.6.2010 7:52:38	Failure	SWT08XP	Terminal adapter 00:19:D1:17:28:C4 disconnected	PLC process control messages		
15.6.2010 7:52:38	Failure	SWT08XP	Terminal adapter 00:19:D1:17:28:C4 disconnected	PLC process control messages		
15.6.2010 7:52:16	Failure	SWT08XP	Terminal adapter 00:19:D1:17:28:C4 disconnected	PLC process control messages		
15.6.2010 7:52:14	Failure	SWT08XP	Terminal adapter 00:19:D1:17:28:C4 disconnected	PLC process control messages		
15.6.2010 6:00:14	Process control system	AM:SWT08XP:Backup:Error on writing to backup path	System, need not be acknowledged	ArchiveManager	Backup:invalid path for type TLG_S:\kft	
15.6.2010 6:00:14	Process control system	AM:SWT08XP:Backup:Error on writing to backup path	System, need not be acknowledged	ArchiveManager	Backup:invalid path for type TLG_F:\kft	
15.6.2010 6:00:14	Process control system	AM:SWT08XP:Backup:Error on writing to backup path	System, need not be acknowledged	ArchiveManager	Backup:invalid path for type TLG_F:\kft	
15.6.2010 6:00:14	Process control system	AM:SWT08XP:Backup:Error on writing to backup path	System, need not be acknowledged	ArchiveManager	Backup:invalid path for type TLG_F:\kft	
15.6.2010 6:00:14	Process control system	AM:SWT08XP:Backup:Error on writing to backup path	System, need not be acknowledged	ArchiveManager	Backup:invalid path for type ALG:\kftcz	
15.6.2010 6:00:07	Process control system	AM:SWT08XP:Backup:Error on writing to backup path	System, need not be acknowledged	ArchiveManager	Backup:invalid path for type ALG:\kftcz	

Obrázek 4: Příklad alarmů v COMES Historian [6]

Vstupem pro COMES Historian jsou data zprostředkovaná modulem CCI, což jsou data z řídicí úrovně a informace z webových služeb, tedy data z modulu COMES Modeller a data z aplikací třetích stran. Modul také poskytuje uložená data dalším COMES modulům. [6]

2.2.3 COMES Modeller

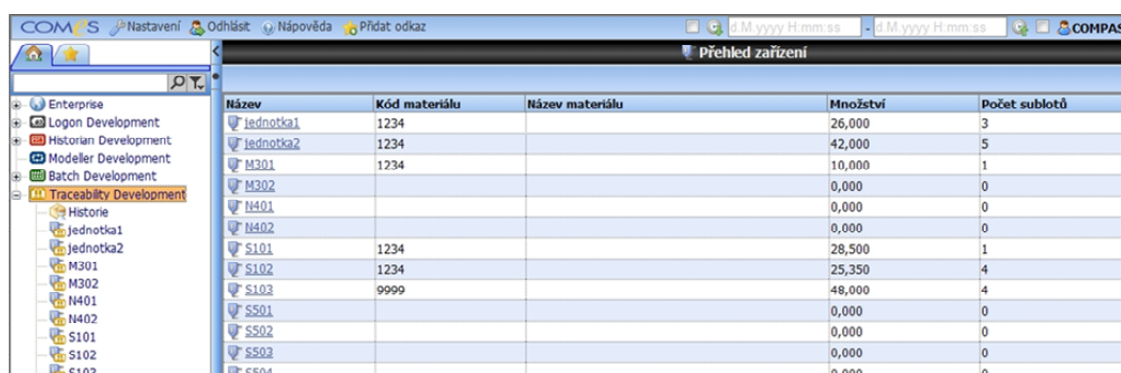
Tento modul přepracovává data na informace v podobě ukazatelů, tabulek, datových sestav a bilancí. Plní funkce: vytváření zákaznických aplikací, reporting (dat z Modeller či ostatních modulů), interface na externí systémy (např. ERP), výpočty KPI a OEE, identifikace a označování výroby (čárové kódy, RFID), vytváření přehledových vizualizací, motivačních obrazovek a operátorských terminálů. [7]

Data mohou být zobrazena v podobě tabulek, grafů, protokolů, uživatelských HTML stránek nebo ve formě vizualizace. [7]

Vstupem pro Modeller mohou být jak ostatní moduly COMES, tak externí databáze. Modul lze uživatelsky modelovat, z čehož vyplynul jeho název. [7]

2.2.4 COMES Traceability

Zajišťuje sledování výroby z hlediska pohybu materiálů (suroviny, meziprodukty, obaly), včetně jejich určení (množství atd.), dlouhodobou archivaci veškerých dat s tímto spojenými a vytváření rodokmenu výrobků. Je integrován s COMES Modeller. Umožňuje online kontrolu. [8]



The screenshot shows the COMES Traceability application window. The title bar includes 'COMES', 'Nastavení', 'Odhlásit', 'Nápověda', and 'Přidat odkaz'. The main window has a menu bar with 'Přehled zařízení'. On the left is a tree view with 'Enterprise' expanded, showing 'Logon Development', 'Historian Development', 'Modeller Development', 'Batch Development', and 'Traceability Development'. Under 'Traceability Development' is 'Historie', which is further expanded to show 'Jednotka1', 'Jednotka2', 'M301', 'M302', 'N401', 'N402', 'S101', 'S102', and 'S103'. The main area displays a table titled 'Přehled zařízení' with the following data:

Název	Kód materiálu	Název materiálu	Množství	Počet sublotů
Jednotka1	1234		26,000	3
Jednotka2	1234		42,000	5
M301	1234		10,000	1
M302			0,000	0
N401			0,000	0
N402			0,000	0
S101	1234		28,500	1
S102	1234		25,350	4
S103	9999		48,000	4
S501			0,000	0
S502			0,000	0
S503			0,000	0
S504			0,000	0

Obrázek 5: Výpis materiálů v COMES Traceability [8]

2.2.5 COMES Batch

Jeho úkolem je řízení dávkových neboli šaržových výrobních procesů. Vychází z mezinárodního standardu ANSI/ISA 88. Využívá technologie Microsoft Silverlight pro editor a klientské rozhraní. Nabízí možnosti jednotlivých fyzických a logických fází, které lze dále nastavovat. Průběžně kontroluje vytvářený předpis, nebo operaci.

Lze zde také využít testovací režim. V průběhu výrobního procesu vytváří elektronický záznam o šarži. [9]

2.3 COMES OEE

2.3.1 Ukazatel OEE

Je hlavní ukazatel pro podniky, na jehož základě se zlepšuje výrobní efektivita. Udává se v procentech využití výrobní kapacity. Obvykle má hodnotu okolo 85% až 100%. V šaržové výrobě bývá menší vlivem prostojů při přechodu mezi dávkami. [10]

Počítá se jako součin tří faktorů:

$OEE = \text{Dostupnost zařízení} * \text{Výkon zařízení} * \text{Kvalita výroby} * 100 [\%]$

Dostupnost zařízení: skutečný/plánovaný čas výroby.

Výkon zařízení: vyrobené/normované výrobky.

Kvalita výroby: shodné/vyrobené výrobky.

COMES OEE je aplikace, která slouží k vyhodnocení efektivit výrobních linek podniků, nejčastěji prostřednictvím KPI. Zahrnuje v sobě COMES Dashboard. Poskytuje během výroby užitečné informace, na základě kterých se dělají rozhodnutí přispívající k lepším výsledkům.

COMES OEE poskytuje přehled o: efektivitě výrobních linek, prostojích při výrobě, počtu správně vyrobených kusů či naopak zmetků. Dále vypočítává a zobrazuje důležité statistiky (KPI) a směnové protokoly. Výsledkem použití COMES OEE je snížení nákladů na výrobu. [10]

2.3.2 Funkce COMES OEE

Sběr dat - automatický či ruční sběr dat o výrobních linkách a o počtu vyrobených výrobků. Vyhodnocení prostojů a neshod výrobků.

Vizualizace - data z jakýchkoliv zařízení mohou být zobrazena buď prostřednictvím operátorských terminálů ve výrobě či vizualizací na PC manažerů.



Obrázek 6, Obrázek 7: Vizualizace na velkoplošné TV ve výrobní hale [10]

Analýzy - důležité výrobní hodnoty lze vyhodnocovat pomocí grafů, tabulek a protokolů, které slouží jako podklad pro optimalizaci výroby.

Komunikace - je možné jej plnohodnotně začlenit do podnikového systému řízení výroby a ERP. COMES OEE také umožňuje připojení téměř jakéhokoliv měřicího zařízení, z něhož získáváme data z výrobního procesu.

Zálohování dat - provádí se automaticky každý den na disk serveru, kde se uchovává pouze posledních deset záloh. V případě potřeby lze zálohovat i ručně.

[10]

3. DASHBOARD

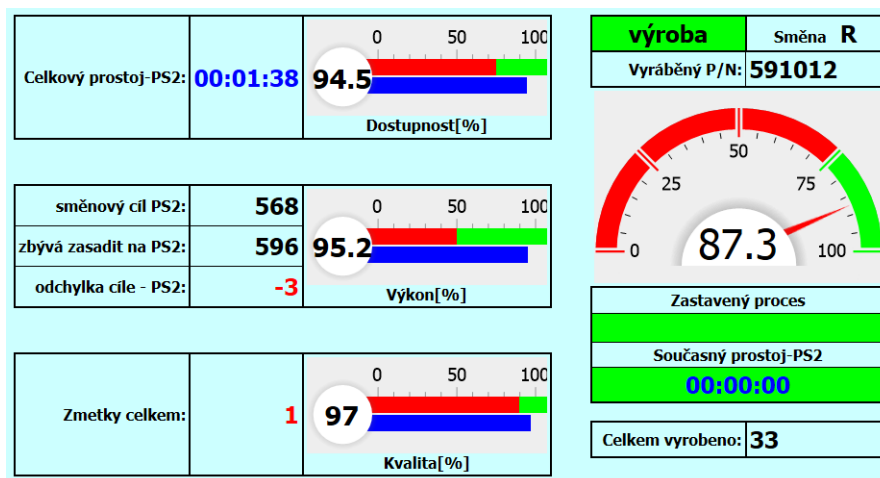
Dashboard lze v českém jazyce označovat jako řídicí panel. Slouží k přehlednému grafickému zobrazení klíčových parametrů o řízení podniku či výrobního procesu. Vše bývá vhodně uspořádáno na jedné stránce, případně obrazovce. Dashboard poskytuje manažerům či obsluze užitečné informace, díky kterým dochází ke správným rozhodnutím a tím i k dosažení stanovených cílů.

Slovo dashboard pochází z anglického jazyka. V doslovném překladu se jedná o palubní, nebo přístrojovou desku v automobilech či letadlech, kde je zapotřebí zobrazit mnoho informací nezbytných k řízení, na jednom místě. Pro řidiče automobilu jsou to například: rychlost, otáčky, teplota. Důležité jsou zde také indikace odchylek od žádoucího stavu: nedostatek paliva, oleje. Stejně tak i podnikový dashboard poskytuje informace o výrobě, prodejích, zisku. A indikuje odchylky od žádaného stavu: poruchy či jakékoliv nedostatky. Důvod inspirace v použití tohoto slova při řízení podniku, nebo procesu je tedy zřejmý. [11]

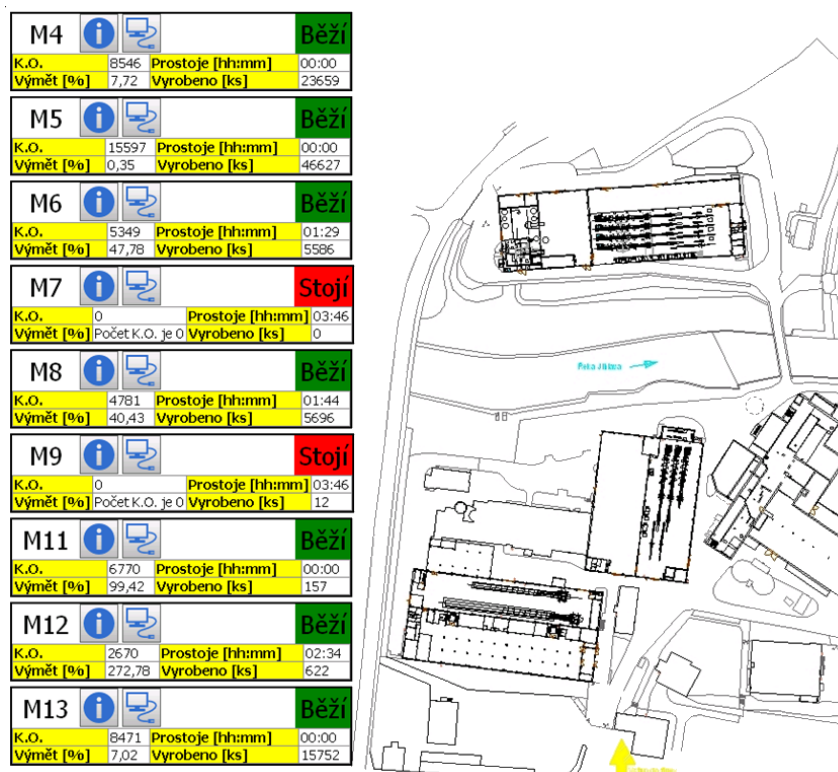
3.1 COMES Dashboard

COMES Dashboard je zahrnut v aplikaci COMES OEE. Používá se ve výrobně informačním systému COMES pouze v případě zájmu kupní strany. Prozatím nemá přesně stanovenou podobu ani pravidla při jeho vytváření. Návrh je sestaven na základě požadavků investora.

3.2 Příklady dashboardů v systému COMES

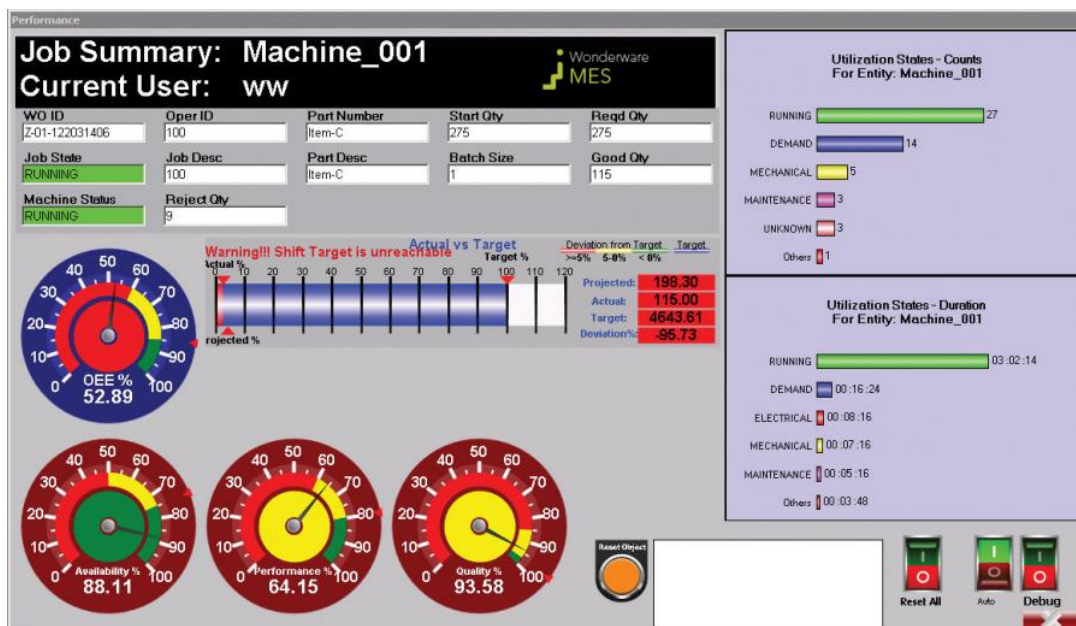


Obrázek 8: Příklad Andon obrazovky systému COMES OEE [12]

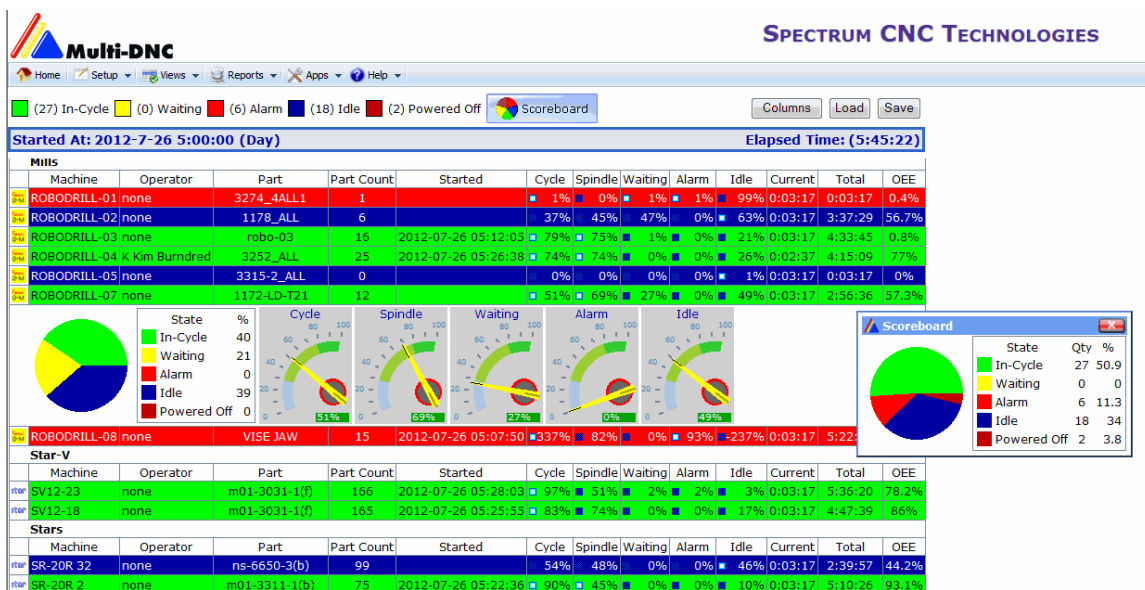


Obrázek 9: Dashboard, vytvořen firmou Compas, zobrazující stav výroby

3.3 Dashboardy používané v jiných systémech



Obrázek 10: Dashboard zobrazující souhrn práce stroje [13]



Obrázek 11: Dashboard vyjadřující důležité parametry strojů [14]

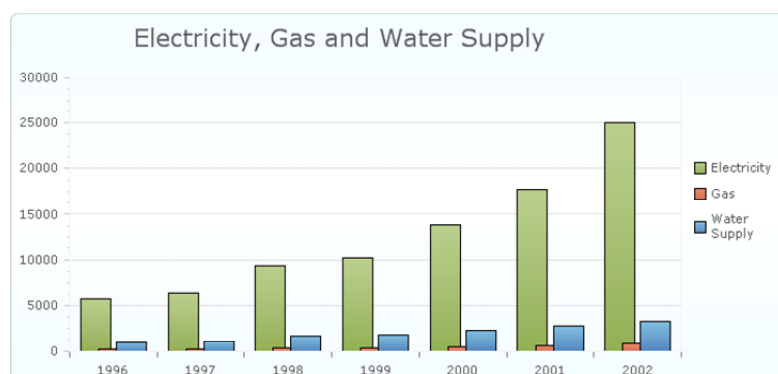
4. ANALÝZA KOMPONENT PRO TVORBU DASHBOARD VIZUALIZACE

Na řídicí panel pro komerční i nekomerční účely lze použít téměř jakýkoliv vizualizační prvek. Hlavní je, aby byl indikátor snadno pochopitelný. Nejčastěji vídané jsou grafy, měřidla a tabulky.

Velmi důležité jsou také barevné aspekty prvků, či jejich částí. Červená barva většinou představuje problém. Zelená, že je vše v pořádku (plán výroby je plněn). Oranžová vyjadřuje mezní hodnoty a žlutá ve většině případů nese výstražnou informaci.

4.1 Graf

Je velmi často používán. Zobrazuje data ve strukturované grafické podobě. Lze si vybrat z mnoha typů grafů: sloupcové, spojnicové, prstencové, výsečové, pruhové, plošné, XY grafy, bublinové a případné jejich modifikace.



Obrázek 12: Příklad provedení grafu [15]

4.2 Měřidlo

Lze nazývat také jako ukazatel či „budík“. Graficky zobrazuje jednu hodnotu v daném rozsahu. Může být lineární nebo radiální. Existuje nesčetné množství typů a podob měřidel. Významné hodnoty mohou být barevně odlišeny. Lze použít i prvky s více ručičkami.

Jako příklad použití můžeme uvést měřidlo zisků za určité časové období. Pokud je aktuální zisk nad hranicí plánovaného zisku, není třeba zásahu do procesu. Pokud ovšem zisk poklesne pod určitou kritickou úroveň, musí se vykonat patřičné změny.



Obrázek 13: Příklad provedení měřidla [16]

4.3 Tabulka

Souhrn dat vynesných ve sloupcích a řádcích. Je vhodná pro větší množství zobrazovaných dat. Pomocí algoritmů lze z hodnot v tabulce vypočítat další ukazatele, které jsou důležité například pro chod výrobního procesu. Kritické či důležité hodnoty mohou být v tabulce barevně odlišeny.

Den: 2.4.2013		Pracoviště: M13	
HUHTAMAKI		Report výmětu po sekcích	
	Ranní	Odpolední	Noční
	[ks] [%]	[ks] [%]	[ks] [%]
Stroj - před mlžením (sušárna)	276 1.09	9 0.03	15 0.04
Před mlžením - za mlžením (mlžení)	702 2.80	298 0.83	365 1.00
Za mlžením - před potiskem (mlžení - denestry)	4992 20.47	-3871 -10.90	852 2.35
Před potiskem - zabalené (potisk)	1248 6.43	320 0.81	1568 4.42

Obrázek 14: Příklad tabulky použité v COMES v zakázce pro Huhtamaki

4.4 Datové pole

Zobrazuje pouze jednu danou hodnotu. Většinou je tato hodnota důležitější než ostatní. Například celkový počet vyrobených předmětů.

Celkem vyrobeno:	33
------------------	----

Obrázek 15: Příklad provedení datového pole

4.5 Akční komponenty

Do této skupiny spadají signalizační a zadávací prvky. Vzhled těchto komponent se většinou odvíjí od podobnosti s reálnými předměty. Mohou to být semaforey a jiné signalizace, které graficky znázorňují stav příslušného atributu. Majáky, které blikají v případě vzniklého problému. Uplatnění zde také nacházejí různá tlačítka a vypínače pro vykonávání akcí, případně potenciometry pro nastavení hodnot.

4.6 Vhodné typy komponent

Určit, jaké typy komponent jsou vhodné, je velmi obtížný úkol. Pro každou problematiku se hodí jiný typ, vzhled a uspořádání. Pokud vytváříme řídicí panel pro výrobní linku, použijeme jiné prvky než pro dashboard celého výrobního procesu. V dnešní době je trend využívat především grafy a tabulky, které doplníme o měřidla, datová pole a případně další prvky, zvýrazňující nedostatky. Pro dosažení dostatečné efektivity je tedy vhodné volit právě z těchto zmiňovaných typů komponent.

5. JAZYK SVG

Scalable Vector Graphics je značkovací jazyk pro tvorbu dvojrozměrné vektorové grafiky využívající XML (XML je obecný značkovací jazyk, ze kterého vycházejí konkrétní značkovací jazyky). Byl navržen především k tvorbě webu. Lze se s ním setkat ale i jinde. Postupně se totiž stává průmyslovým standardem pro přenos vektorové grafiky mezi různými aplikacemi. Slouží ke kreslení jednodušších grafických obrazců. Jeho výhody lze uplatnit i na složitější ilustrace. Veškerá grafika se skládá ze základních geometrických útvarů, jako jsou čáry, kružnice či obdélníky. Posléze se jim zadají vlastnosti jako umístění, barva, průhlednost, atd. [17]

5.1 Vektorová a bitmapová grafika

5.1.1 Bitmapová (pixelová) grafika

Bitmapový obrázek je složen z jednotlivých pixelů, které mají jednoznačně určeny polohu, barvu a průhlednost. Typickým zástupcem bitmapové grafiky je digitální fotografie. Důvodem, proč se pro digitální fotografie nevolí vektorový formát je fakt, že na fotografiích nenalezneme pravidelně se opakující tvary, které by bylo možné definovat geometrickými obrazci. Z toho vyplývá nutnost popisu každého pixelu zvlášť.

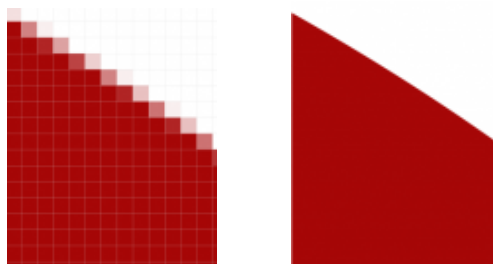
Tento způsob záznamu obrazu má ovšem několik významných nedostatků. Především je to velká datová velikost, závislá na rozlišení a barevné hloubce (až jednotky megabajtů) a velice omezená změna velikosti obrázku. Při vícenásobném zvětšení dochází k podstatnému zhoršení kvality obrázku.

Nejčastěji se pro uchování rastrové grafiky používají formáty JPEG, GIF a PNG.

[18]

5.1.2 Vektorová grafika

Je složena ze základních geometrických obrazců, jako jsou čáry, kružnice, křivky atd. Tyto obrazce jsou určeny vždy alespoň počátečním bodem, směrem (čáry), poloměrem (kružnice). Díky tomu, že jsou jednotlivé tvary popsány obecně, nikoliv na úrovni jednotlivých pixelů, je možná libovolná změna velikosti grafiky, aniž by se tato transformace negativně podepsala na jejich kvalitě. Vektorový formát umožňuje také práci s každým objektem zvlášť, což je vhodné pro tvorbu animací a interaktivit. Datová velikost je v porovnání s rastrovým formátem řádově menší (jednotky kilobajtů). U složitých ilustrací je tomu ovšem naopak. To znamená, že vektorové vyjádření fotografie je téměř nemožné. [19]



Obrázek 16: Vlevo bitmapa, vpravo vektorový obrázek

5.2 Výhody SVG

- Volitelná velikost obrázku (ZOOM).
- Malá datová velikost obrázku.
- Kompatibilita - jelikož je SVG založen na XML.
- Obrázky mohou být interaktivní či dynamické.
- Podporuje kaskádové styly.
- Kreslení ve vrstvách.
- Vkládání textu.
- Jednoduchá přenositelnost.
- Podpora ostatními standardy (HTML aj.)

SVG umožňuje integraci s dalšími technologiemi. K nejdůležitějším patří scriptování (nejznámější je JavaScript), DOM (manipulace se všemi elementy), ICC (barevné profily). Umožňuje také CSS (kaskádové styly), díky nimž lze měnit velikosti a umístění celých skupin prvků.

Zdrojový kód obrázku je prostý text umožňující vyhledávání objektů například podle „id“.

Další výhodou je také podpora tohoto formátu většinou nových verzí webových prohlížečů, jako jsou Opera, Mozilla Firefox, Google Chrome, Safari a také Internet Explorer.

[20]

5.3 Skladba SVG dokumentu

Jak již bylo zmíněno, SVG je formát založený na XML (Extensible Markup Language). To znamená, že veškeré grafické objekty jsou popsány textem.

5.3.1 Základní kostra

```
<svg width="200px" height="100px"
    version="1.1" xmlns="http://www.w3.org/2000/svg" >
    <!--Vlastní kód-->
</svg>
```

Příklad 1: Kostra SVG dokumentu

[21]

Tag <svg> vymezuje prohlížeči část, která patří k SVG dokumentu. Parametry width a height určují šířku a výšku tohoto dokumentu. Druhý řádek definuje XML namespace.

5.3.2 Základní tvary objektů

Jazyk SVG definuje několik základních tvarů pro tvorbu grafiky.

Obdélník - Je definován příkazem „rect“ s odsazením od levého horního rohu (parametry x a y), barvou, šířkou a výškou.

```
<rect x="2" y="2" fill="#999999" width="200" height="300"/>
```

Příklad 2: Definice obdélníku v SVG

Kruh - Je definován příkazem „circle“ se středem na souřadnicích cx a cy, barvou a radiusem r.

```
<circle cx="20" cy="20" fill="#999999" r="10"/>
```

Příklad 3: Definice kruhu v SVG

Čára - Je definována příkazem „line“ se začátkem v souřadnicích (x1,y1) a koncem v souřadnicích (x2,y2).

```
<line x1="2" y1="2" x2="10" y2="10" />
```

Příklad 4: Definice čáry v SVG

Křivka - Je definována příkazem „polyline“ a tvoří ji série čar se souřadnicemi x a y.

```
<polyline points="2,2 30,60 70,140" />
```

Příklad 5: Definice křivky v SVG

Element cesta - Je definován příkazem „path“. Jde o grafický objekt definovaný sekvencí příkazů. Objekt může být složen z čar, křivek, segmentů atd. M určuje počáteční bod a L vykreslí čáru do bodu x=20, y=20.

```
<path d="M 0 0 L 50 50" />
```

Příklad 6: Definice cesty v SVG

Seskupení objektů - Je zprostředkováno párovým tagem <g>. Skupina dědí veškeré vlastnosti definované v tagu <g>.

```
<g fill="#FF0000">  
<circle cx="20" cy="20" fill="#999999" r="10"/>  
<rect x="2" y="2" fill="#999999" width="200" height="300"/>  
</g>
```

Příklad 7: Seskupení objektů v SVG

[22]

5.3.3 Vlastnosti SVG objektů

Každý objekt či skupina objektů mohou mít nastaveny doplňující informace, jako jsou například:

- Identifikační název objektu - ID
- Vnitřní barva objektu - fill
- Barva vnějšího okraje - stroke
- Tloušťka vnějšího okraje - stroke-miterlimit

5.4 SVG Editory

K vytváření grafických objektů v SVG existuje velké množství komerčních a nekomerčních editorů. Mezi nejznámější patří Adobe Illustrator a Corel Draw. Lze pracovat i v EvolGrafix XStudio 2, Jasc Web Draw 1, EvolGrafix X Studio, WebDraw, Inkscape, SVG Editor a mnoha dalších.

5.5 SVG a scriptování

Jelikož je SVG založeno na jazyce XML lze parametry jednotlivých či seskupených objektů měnit programově, pomocí jakéhokoliv skriptovacího jazyka, jako je například JavaScript či Java samotná. Mohou se tak vytvářet SVG objekty, jejichž podoba se bude dynamicky měnit v závislosti na jiných událostech. [23]

6. JAVASCRIPT

Jedná se o objektově orientovaný, programovací, skriptovací jazyk, nacházející uplatnění především při tvorbě WWW stránek a aplikacích jim podobným. Scriptovací znamená, že se nevytváří samostatně fungující programy, ale skripty, které se vkládají například do struktury HTML. Obvykle ovládá různé interaktivní prvky jako tlačítka, textová pole aj. Použit může být na jakémkoliv operačním systému (Windows, Linux, MacOS X či jiném).

Javascript pracuje na straně klienta, musí být tedy nejprve stažen do příslušného počítače a až poté spuštěn prohlížečem. Z toho tedy vyplývá nutnost povolení javascriptových efektů v prohlížeči. Syntaxe patří do rodiny jazyků C, C++ a Java. JavaScript využívá tzv. DOM (Document Object Model), rozhraní umožňující přistupovat k jednotlivým prvkům stránky.

[24]

6.1 Tvorba skriptu

Script můžeme importovat přímo do HTML pomocí párového tagu `<script>`, načítat ze samostatného .js souboru či vložit skript do odkazu.

```
<script>
//vlastní kód
</skript>
```

Příklad 8: Import skriptu přímo do HTML

```
<skript src ="functions.js" >/skript>
```

Příklad 9: Načtení skriptu ze souboru

```
<a href="javascript:alert('Zavolej hlášku!!!' );">Kliknutí
zde, zobrazí hlášku.</a>
```

Příklad 10: Vložení skriptu do odkazu

[25]

6.2 Obsah skriptu

Jak již bylo zmíněno, syntaxe JavaScriptu patří do rodiny jazyků C, C++, Java atd. Jsou zde ovšem některé odlišnosti. Například zde není rozlišována definice metody

a funkce. Dědičnost zde má určité odlišnosti. Dále JavaScript obsahuje velké množství vlastních událostí, které slouží k příslušným účelům (onClick, onMouseDown...). [25]

6.3 JavaScriptové knihovny

V současnosti se kvůli usnadnění práce při vytváření HTML a zlepšení vlastností celkového skriptu využívá velká řada JS knihoven. Ty mohou být určeny k vytváření odkazů, změně částí stránky, úpravě vzhledu objektů či interakci objektů.

Pro tvorbu grafických vizualizačních komponent v rámci této bakalářské práce je velice výhodné implementovat knihovnu, která zefektivní interaktivitu vytvářených prvků. Pro tento účel se nejvíce hodí knihovna jQuery.

6.4 jQuery

Je nejznámější JavaScriptová knihovna. Klade důraz na interakci mezi JS a HTML. Rozšiřuje podporu prohlížečů. Zjednoduší programový kód skriptu. Umožňuje tvorbu animací, efektů, práci s CSS, selektory, události a mnohé další.

Tuto knihovnu lze bezplatně stáhnout a načíst do vlastních souborů.

[26]

6.4.1 Načtení jQuery do HTML

Tuto knihovnu můžeme do vlastních stránek nahrát několika způsoby. Jedním je stažení jQuery souboru do PC a následné nalinkování v HTML.

```
<script src="js/jquery-1.11.0.min.js"></script>
```

Příklad 11: Vložení jQuery knihovny do HTML pomocí nahrání z vlastního souboru

Dalším způsobem je například načtení ze serverů Google. Sníží se tak odezva při volání funkcí. Není nutnost mít jQuery knihovnu staženou v PC.

```
<skript  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jque  
ry.min.js"></skript>
```

Příklad 12: Načtení jQuery knihovny do HTML z Google

[27]

6.4.2 Funkce symbolu „\$“

Pro tuto knihovnu je symbol „\$“ klíčový, jelikož vymezuje prostor jQuery. „\$“ musí být uveden před každou událostí či funkcí, volanou z knihovny.

```
$( "#BACKGROUND_BT1" ).attr("fill", "red");
```

Příklad 13: Použití \$ - pro objekt "BACKGROUND_BT1" je nastavena barva výplně na červenou

6.4.3 CSS v jQuery

jQuery také zprostředkovává práci s Cascading Style Sheets (kaskádovými styly). Ty určují vzhled HTML dokumentu, případně jejich částí (barvy, pozicování aj).

```
$( "#StartStopButtons" ).css({ "width": 50 });
```

Příklad 14: Použití CSS v jQuery-nastavení šířky objektu StartStopButtons na 50 pixelů

7. NÁVRH KOMPONENT

Nyní se dostáváme k vlastnímu návrhu a realizaci jednotlivých grafických vizualizačních komponent pro COMES Dashboard. Koncept spočíval ve dvou krocích.

Nejprve bylo zapotřebí vytvořit grafickou podobu objektu, k čemuž jsem použil Adobe Illustrátor. Tento program slouží pro tvorbu vektorové grafiky. Zároveň nabízí rozsáhlou řadu funkcí s profesionálním výstupem, například v SVG.

K programování funkcionality komponent bylo možno využít pouze textový editor. Ten ale nenabízí žádné podpůrné funkce pro práci s JavaScriptem a HTML. Z tohoto důvodu jsem využil volně šiřitelný PSpad.

7.1 Skladba komponent

Každá komponenta se tedy skládá z SVG části, která zastupuje grafickou podobu, a JavaScriptových funkcí. Ty vytvářejí příslušné efekty a mají za úkol změnu velikosti či pozicování celých objektů pomocí CSS.

Obslužné funkce komponent, s vhodnými parametry, jsou volány ve skriptu dashboardu. U některých prvků jsou tyto funkce dvě, kdy jedna slouží k nastavení obecných vlastností komponenty (velikost atd.) a druhá má za úkol vykonání příslušných akcí, jako může být například rozsvícení signalizačního světla.

Pro zjednodušení použití vytvořených komponent je obsah všech funkcí vložen do samostatného JavaScriptového souboru s názvem „functions.js“. Stačí tedy pouze načíst tuto knihovnu ve zdrojovém kódu vytvářeného dashboardu a zavolat příslušné funkce, které zprostředkovávají veškeré náležitosti komponent. Použití se tedy velice zjednoduší. Soubor „functions.js“ lze nalinkovat kamkoliv do kódu dashboardu, přičemž nejvhodnější je umístit link do hlavičky, kde zároveň načteme knihovnu jQuery. Volání javascriptových funkcí jednotlivých komponent s nastavenými parametry musí být provedeno v těle dokumentu, konkrétně v obsahu vymezeném párovým tagem <skript>. Pokud se jedná o objekt, který umožňuje vykonávání dalších událostí (tlačítka, vypínače, potenciometr), lze ve skriptu také vyčíst a následně zpracovat proměnné hodnoty, které zastupují stav komponenty (zapnuto - true, vypnuto - false).

Další nezbytností pro použití daného vizualizačního prvku je vložení SVG obrázku (v textové podobě) například do hlavičky kódu dashboardu. Pro čerpání těchto částí slouží HTML soubory, nacházející se ve složce „Components“ (viz CD). Pokud si tedy jeden z těchto souborů spustíme v některé z novějších verzí prohlížeče, zobrazí se nám konkrétní grafická podoba tohoto vizualizačního prvku. Následně lze ve zdrojovém kódu nalézt SVG kód komponenty a to i se vzorovým způsobem použití, včetně úseku se scriptem. Kompletní SVG část pro daný vektorový obrázek je vymezena párovým

tagem `<div>`, přičemž v prvním z těchto tagů je ID celé komponenty (např.: `<div id="Gauge_6">`) a my tak přesně víme, s kterým prvkem pracujeme. Díky tagům `div` lze také uplatnit CSS.

Veškerá programová část této bakalářské práce byla na doporučení konzultanta vypracována v anglickém jazyce. Důvodem je možnost použití vytvořených komponent i na zahraničních projektech.

7.2 Příklad struktury programové části komponenty

COMES Dashboard bývá zobrazen na klientu systému COMES, prostřednictvím webového prohlížeče. Proto jsou veškeré vytvořené komponenty vloženy do samostatných HTML souborů. Pokud bychom tedy chtěli jednu z vytvořených komponent použít například na dashboardu umístěném na WWW stránce, existuje několik způsobů, jak vhodně tuto strukturu kódu uspořádat. Jeden z možných způsobů je uveden na příkladu 15.

```
<html>
  <head>
    <title>Dashboard</title>
    <script src="cesta/jquery-1.11.0.min.js"></script>
    <script src="cesta/functions.js"></script>
    <div id="Nazev">
      <!--SVG kód komponenty-->
    </div>
  </head>
  <body>
    <script>
      //Volání obslužných funkcí ze souboru functions.js
    </script>
  </body>
</html>
```

} kód grafické části

Příklad 15: Možná struktura kódu pro deklaraci komponent v HTML

8. PREZENTACE REALIZOVANÝCH KOMPONENT

V rámci této bakalářské práce bylo vytvořeno čtrnáct grafických vizualizačních komponent. Pět z nich je navrženo pro zobrazování jedné hodnoty v daném rozsahu (typ měřidlo). Jedná se o prvky „Gauge_1“, „Gauge_2“, „Gauge_3_thermometer“, „Gauge_5“ a „Gauge_6“. Ostatních devět spadá do podskupiny akčních komponent. Jeden z nich je výstražný maják, pojmenován jako „Beacon 1“. Další tři jsou také signalizační, konkrétně tedy „RedGreen Signalization“, „Column Signalization“ a „Pilot Light“. Ostatních pět komponent plní funkci vstupů pro události na dashboardu. Jsou zastoupeny tlačítky, vypínačem a lineárním potenciometrem, s názvy „Start-Stop Buttons“, „Button“, „ONxOFF Switch“, „Reset Button“ a „Linear Potenciometer“.

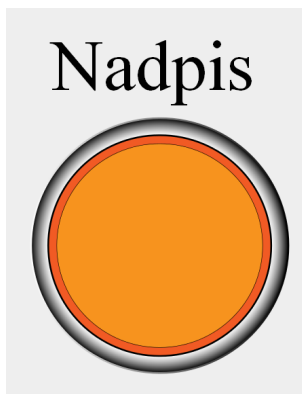
Všechny tyto komponenty jsou v samostatných HTML souborech uloženy ve složce „Components“ (viz CD). Dále se v této složce nachází také soubor s názvem „js“. Uvnitř něhož pak najdeme, pro správnou funkci nezbytnou, knihovnu jQuery, knihovnu „functions“ a několik doplňujících efektů pro signalizační komponenty.

V nadcházející části bakalářské práce jsou prezentovány jednotlivé vizualizační komponenty. Každá z nich má uvedeny údaje jako název, charakteristiku, grafickou podobu, funkci obslužnou (v některých případech i dvě) a u vstupních objektů také funkci návratovou. SVG kód obrázku a obslužný program komponent zde, s ohledem na rozsah práce, uveden není. Tyto části nalezneme v příloze (viz příložené CD).

Každá funkce je doplněna o popis všech volitelných parametrů, které budou číslovány od levé strany, tedy v posloupnosti nadefinování.

8.1 Button

Button komponenta zastává funkci běžného tlačítka (viz Obrázek 17). V případě stisku dojde k pomyslnému rozsvícení střední části. Na první pohled je tedy zřejmé, že je tlačítko sepnuto. Dále nastane také překlopení výstupní proměnné VALUE_Button na hodnotu „true“. S touto komponentou lze vykonávat téměř jakoukoliv událost na dashboardu.



Obrázek 17: Komponenta Button

Button umožňuje velmi dobrou modifikovatelnost. Lze měnit velikost celého objektu, odsazení od horní a levé hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barva pozadí komponenty (šedý rámeček na Obrázku 17), nadpis, typ tlačítka, barvy vnitřní části tlačítka při zapnutém a vypnutém stavu. Veškeré změny zprostředkovává obslužná funkce Button s jedenácti parametry.

```
Button(180, 0, 0, 'show', "#EEEEEE", 'Nadpis', 'click',
"#F19024", "#F7C81E", "#F15A24", "#F7931E");
```

Příklad 16: Obslužná funkce komponenty Button

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
6. **Parametr:** Nadpis komponenty.
Hodnota: Libovolný krátký text ['Nadpis'].
7. **Parametr:** Volba typu tlačítka.
Hodnota: ['click'] (zapnutí při stisku, vypnutí při opětovném stisku) nebo ['updown'] (zapnutí při „zmáčknutí“, vypnutí při „odmáčknutí“ tlačítka).
8. **Parametr:** Barva okraje střední části (tmavší oranžová část na Obrázku 17) při vypnutém stavu.

Hodnota: Anglický překlad barvy ["black"], nebo hex. číslo barvy ["#000000"].

- 9. Parametr:** Barva středu (světlejší oranžová část na Obrázku 17) při vypnutém stavu.

Hodnota: Anglický překlad barvy ["black"], nebo hex. číslo barvy ["#000000"].

- 10. Parametr:** Barva okraje střední části (tmavší oranžová část na Obrázku 17) při sepnutém stavu.

Hodnota: Anglický překlad barvy ["black"], nebo hex. číslo barvy ["#000000"].

- 11. Parametr:** Barva středu (světlejší oranžová část na Obrázku 18) při sepnutém stavu.

Hodnota: Anglický překlad barvy ["black"], nebo hex. číslo barvy ["#000000"].

Výstup tlačítka je reprezentován proměnnou `VALUE_Button`, která v případě stisknutého stavu nabývá hodnoty „true“ a nestisknutého stavu hodnoty „false“. `VALUE_Button` je obsažena v návratové funkci `Output_Button`. Tato funkce je nadefinována ve skriptu dashboardu a slouží k vykonání příslušných akcí spjatých s tlačítkem.

8.2 Start-Stop Buttons

Dvojice těchto tlačítek je určena k zapínání a vypínání jakýchkoliv událostí na dashboardu (viz Obrázek 18). Stisknutím START tlačítka dojde k jeho „rozsvícení“ a „zhasnutí“ STOP tlačítka. Pokud stiskneme STOP tlačítko, nasává opačný jev. Jelikož u této komponenty je vždy jedno tlačítko „rozsvíceno“, je při prvním pohledu jasné, zdali se aktuálně nacházíme v zapnutém či vypnutém stavu. Při přepínání ze STOP na START a obráceně dochází k překlápění proměnné `VALUE_StartStopButtons`.



Obrázek 18: Komponenta Start-Stop Buttons

U Start-Stop Buttons lze měnit velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barva pozadí (šedý rámeček na obrázku 18). Veškeré tyto změny zprostředkovává obslužná funkce StartStop_Buttons s pěti parametry.

```
StartStop_buttons(180, 0, 0, 'show', "#EEEEEE");
```

Příklad 17: Obslužná funkce komponenty Start-Stop Buttons

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezení rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

Výstup dvojice tlačítek je reprezentován proměnnou VALUE_StartStopButtons, která v případě stisknutého zeleného START tlačítka nabývá hodnoty „true“ a naopak stisknutého STOP tlačítka hodnoty „false“. VALUE_StartStopButtons je obsažena v návratové funkci Output_StartStopButtons. Tato funkce je nadefinována ve skriptu dashboardu a slouží k vykonání příslušných akcí spjatých s tlačítky.

8.3 Reset Button

Reset Button je určen výhradně jako resetovací tlačítko (viz Obrázek 19). Za jeho pomoci můžeme resetovat různé hodnoty či události na dashboardu. Při kliknutí na tlačítko dojde k sepnutí. Pokud odklikneme, nastane rozepnutí. Při přepínání dochází k překlápění proměnné VALUE_ResetButton.



Obrázek 19: Komponenta Reset Button

Upravit je u této komponenty možné velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barva pozadí (šedý rámeček na obrázku 19). Veškeré tyto změny zprostředkovává obslužná funkce `Reset_button` s pěti parametry.

```
Reset_button(180, 0, 0, 'show', "#EEEEEE");
```

Příklad 18: Obslužná funkce komponenty Reset Button

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

Výstup komponenty Reset Button je reprezentován proměnnou `VALUE_ResetButton`, která v případě stisknutého tlačítka nabývá hodnoty „true“ a naopak po uvolnění tlačítka („odkliknutí“) hodnoty „false“. `VALUE_ResetButton` je obsažena v návratové funkci `Output_ResetButton`. Tato funkce je nadefinována ve skriptu dashboardu a slouží k vykonání příslušných akcí spjatých s tlačítkem.

8.4 ONxOFF Switch

ONxOFF Switch je objekt určený k přepínání hodnoty zapnuto/vypnuto jako reálný přepínač (viz Obrázek 20). Velkou výhodou přepínače je stálá informací o stavu, ve kterém se zrovna nachází. Pokud tedy na objekt ONxOFF Switch klikneme, dojde k přepnutí na druhou pozici, než je pozice původní. Zároveň se překlápí proměnná VALUE_ONxOFFSwitch.



Obrázek 20: Komponenta ONsOFF Switch

Modifikovat je možné velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), nadpis, viditelnost a barva pozadí (šedý rámeček na obrázku 20). Veškeré tyto změny zprostředkovává obslužná funkce ONxOFFSwitch s šesti parametry.

```
ONxOFFSwitch(180, 0, 0, 'Nadpis', 'show', "#EEEEEE");
```

Příklad 19: Obslužná funkce komponenty ONxOFF Switch

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Nadpis komponenty.
Hodnota: Libovolný krátký text ['Nadpis'].
5. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].

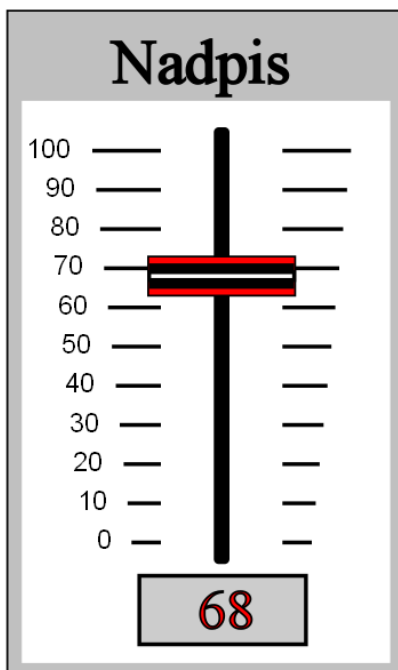
6. **Parametr:** Barva vymezovacího rámečku na pozadí.

Hodnota: Anglický překlad barevy ["red"], nebo hex. číslo barvy ["#000000"].

Výstup komponenty ONxOFF Switch je reprezentován proměnnou VALUE_ONxOFFSwitch, která v případě překlopení přepínače na stav „I“ nabývá hodnoty „true“ a naopak na stavu „O“ hodnoty „false“. VALUE_ResetButton je obsažena v návratové funkci Output_ONxOFFSwitch. Tato funkce je nadefinována ve skriptu dashboardu a slouží k vykonání příslušných akcí spjatých s přepínačem.

8.5 Linear Potenciometer

Jak z názvu vyplývá, komponenta Linear Potenciometer zastupuje funkci lineárního potenciometru (viz Obrázek 21). Při uchopení ukazatele můžeme najet na libovolnou hodnotu ze zvoleného rozsahu a následně tuto hodnotu zpracovat k dalším akcím na dashboardu. Po „odkliknutí“ či opuštění kurzoru myši prostor vyhrazený potenciometrem, dojde k uložení zvolené hodnoty do proměnné VALUE_LinearPotenciometer. Pokud na dashboardu potřebujeme, z jakýchkoliv důvodů, zadávat číselné hodnoty, je lineární potenciometr nejpresnější možný prvek.



Obrázek 21: Komponenta Linear Potenciometer

Měnit na tomto potenciometru je možné odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barva pozadí (šedý rámeček na obrázku

21), nadpis a rozsah. Veškeré tyto změny zprostředkovává obslužná funkce `Linear_potenciometer` s šesti parametry.

Koncepce návrhu je založena na vyčítání y souřadnice kurzoru myši na dashboardu, při uchopení jezdce potenciometru. Změnou velikosti objektu by došlo ke vzniku chyb velikostního poměru, jež by korektně nebylo možno opravit. Změna velikosti pouhým nastavením parametrů obslužné funkce tedy není možná.

```
Linear_potenciometer(0,0,'show',"#EEEEEE",'Nadpis',100);
```

Příklad 20: Obslužná funkce komponenty Linear Potenciometer

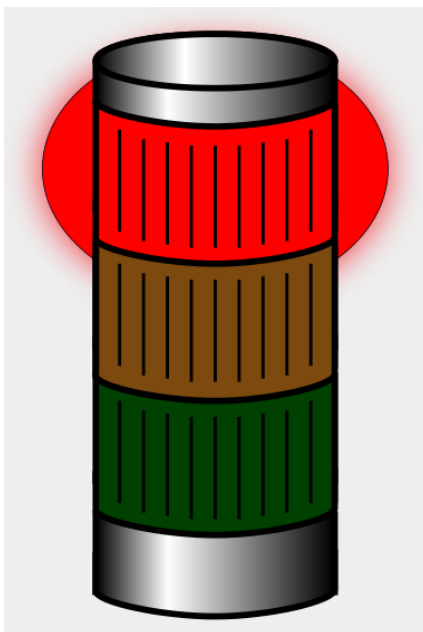
1. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
4. **Parametr:** Barva vymezení rámečku.
Hodnota: Anglický překlad barevy ["red"], nebo hex. číslo barvy ["#000000"].
5. **Parametr:** Nadpis komponenty.
Hodnota: Libovolný krátký text ['Nadpis'].
6. **Parametr:** Číslo, představující maximální hodnotu rozsahu potenciometru.
Ostatní hodnoty rozsahu se ekvivalentně přepočítají.

Výstup komponenty Linear Potenciometer je reprezentován proměnnou `VALUE_LinearPotenciometer`, která představuje hodnotu, na níž se právě ukazatel potenciometru nachází. `VALUE_LinearPotenciometer` je obsažena v návratové funkci `Output_LinearPotenciometer`. Tato funkce je nadefinována ve skriptu dashboardu a slouží k vykonání příslušných akcí spjatých s potenciometrem.

8.6 Column Signalization

Column Signalization slouží jako vizualizační signalizace událostí na dashboardu (viz Obrázek 22). Je inspirována průmyslovou sloupcovou signalizací. Skládá se ze tří, na sobě nezávisle pracujících, „světelných“ částí. Každá z těchto částí umožňuje svícení či blikání, přičemž blikání je nadřazeno svícení. V praxi to znamená, že pokud je vykonána akce, která rozsvítí například zelené světlo, a následně dojde ke splnění podmínky pro jeho rozblikání, tak zelené světlo bliká. Naopak pokud bude světlo nejprve rozblikáno a posléze bude vykonána akce pro jeho rozsvícení, tak bude světlo

stále blikat. Důvodem je odlišení důležitosti blikání a svícení světel, kdy blikání vždy znamená stav více důležitý. Budeme-li chtít přepnout z blikajícího světla na svítící, musíme nejprve blikání úmyslně vypnout. Podstatné také je, že při sepnutí blikání neztrácíme informaci o tom, že světlo původně svítilo. Po vypnutí blikání tedy původně svítící světlo opět svítit, pokud ovšem nebylo v průběhu blikání vypnuto.



Obrázek 22: Komponenta Column Signalization

U celé této komponenty můžeme opět libovolně měnit velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 22). Všechny tyto změny jsou zapouzdřeny v obslužné funkci `ColumnSignalization_properties`, která zprostředkovává pouze vlastnosti. Druhá funkce, `ColumnSignalization`, slouží k přímému rozsvícení a zhasínání jednotlivých světelných částí.

```
ColumnSignalization_properties(400,0,0,'show','#EEEEEE');
```

Příklad 21: Obslužná funkce vlastností komponenty Column Signalization

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.

3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymežovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

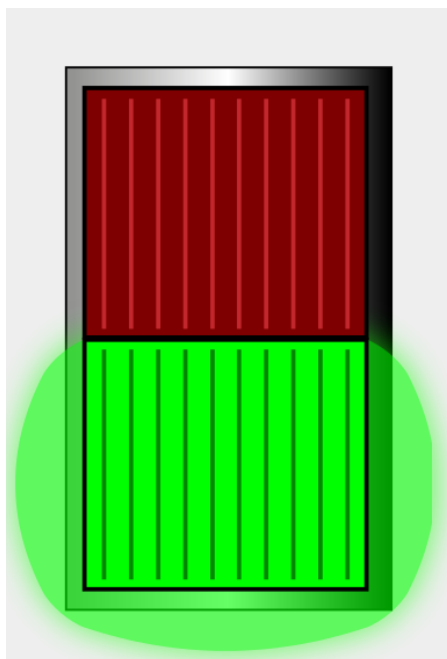
```
ColumnSignalization('REDLIGHT', 'SET');
```

Příklad 22: Obslužná funkce světelných efektů komponenty Column Signalization

1. **Parametr:** Zde si vybíráme, s kterou světelnou částí budeme pracovat a zdali se bude jednat o blikání či svícení.
Jednotka: ['REDLIGHT']-svítící červené světlo, ['ORANGELIGHT']-svítící oranžové světlo, ['GREENLIGHT']-svítící zelené světlo, ['REDBLINK']-blikající červené světlo, ['ORANGEBLINK']-blikající oranžové světlo, ['GREENBLINK']-blikající zelené světlo, ['TURNOFF']-vypnutí všech světelných částí.
2. **Parametr:** Nastavení zapnutí či vypnutí světelného efektu.
Jednotka: ['SET']-zapnutí, ['RESET']-vypnutí.

8.7 RedGreen Signalization

RedGreen Signalization je taktéž navržena jako vizualizační signalizace událostí na dashboardu (viz Obrázek 23). Skládá se ze dvou, na sobě nezávisle pracujících, „světelných“ částí. Každá z nich umožňuje svícení či blikání, přičemž blikání je nadřazeno svícení. V praxi to znamená, že pokud je vykonána akce, která rozsvítí například zelené světlo, a následně dojde ke splnění podmínky pro jeho rozblikání, tak zelené světlo bliká. Naopak pokud bude světlo nejprve rozblikáno a posléze bude vykonána akce pro jeho rozsvícení, tak bude světlo stále blikat. Důvodem je odlišení důležitosti blikání a svícení světél, kdy blikání vždy znamená stav více důležitý. Budeme-li chtít přepnout z blikajícího světla na svítící, musíme nejprve blikání úmyslně vypnout. Podstatné také je, že při sepnutí blikání neztrácíme informaci o tom, že světlo původně svítilo. Po vypnutí blikání tedy původně svítící světlo opět svítit, pokud ovšem nebylo v průběhu blikání vypnuto.



Obrázek 23: Komponenta RedGreen Signalization

U RedGreen Signalization lze libovolně měnit velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 23). Všechny tyto změny jsou zapouzdřeny ve volací funkci `RedGreenSignalization_properties`, která zprostředkovává pouze vlastnosti. Druhá funkce, `RedGreenSignalization`, slouží k přímému rozsvícení a zhasínání jednotlivých světelných částí.

```
RedGreenSignalization_properties(400,0,0,'show','#EEEEEE');
```

Příklad 23: Obslužná funkce vlastností komponenty RedGreen Signalization

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixels.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixels.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixels.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymežovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

```
RedGreenSignalization('REDLIGHT','SET');
```

Příklad 24: Obslužná funkce světelných efektů komponenty RedGreen Signalization

1. **Parametr:** Zde si vybíráme, s kterou světelnou částí budeme pracovat a zdali se bude jednat o blikání či svícení.

Jednotka: ['REDLIGHT']-svítící červené světlo, ['GREENLIGHT']-svítící zelené světlo, ['REDBLINK']-blikající červené světlo, ['GREENBLINK']-blikající zelené světlo, ['TURNOFF']-vypnutí všech světelných částí.

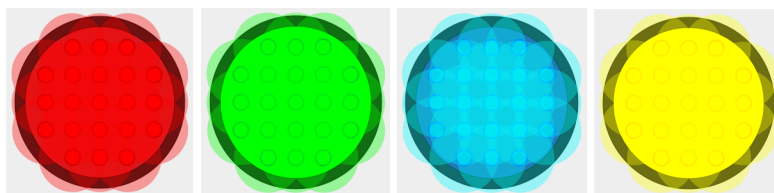
2. **Parametr:** Nastavení zapnutí či vypnutí světelného efektu.

Jednotka: ['SET']-zapnutí, ['RESET']-vypnutí.

8.8 Pilot Light

Pilot Light je komponenta, jež plní funkci kontrolky (viz Obrázek 24). Světelným efektem upozorňuje na události vzniklé na dashboardu, a to konkrétně svícením či blikáním, v předem zvolené barvě. Tyto barvy jsou předdefinovány a lze si tedy vybrat ze čtyř možností (červená, zelená, modrá a žlutá). Blikání je zde opět nadřazeno svícení. V praxi to znamená, že pokud je vykonána akce, která rozsvítí kontrolku na zeleno, a následně dojde ke splnění podmínky pro její rozblikání, tak zelené světlo bliká. Naopak pokud bude světlo nejprve rozblikáno a posléze bude vykonána akce pro jeho rozsvícení, tak bude světlo stále blikat. Důvodem je odlišení důležitosti blikání a svícení světla, kdy blikání vždy znamená stav více nebezpečný. Budeme-li chtít přepnout z blikajícího světla na svítící, musíme nejprve blikání úmyslně vypnout.

Pokud na dashboard vložíme tuto komponentu, není možná dynamická změna barev (například z modré na červenou) při blikajícím režimu. Zvolíme-li si pouze světelný režim kontrolky, je dynamická změna barev kontrolky možná.



Obrázek 24: Možné barvy rozsvícené komponenty Pilot Light

Tato komponenta umožňuje změnu velikost celého objektu, odsazení od levé a horní hrany prohlížeče, viditelnost a barvu pozadí (šedé rámečky na obrázku 24). Všechny tyto změny jsou zapouzdřeny ve volací funkci `PilotLight.properties`, která zprostředkovává pouze vlastnosti. Druhá funkce, `PilotLight`, slouží k přímému rozsvícení a zhasínání kontrolky v dané barvě.

```
PilotLight_properties(75,0,0,'show','#EEEEEE');
```

Příklad 25: Obslužná funkce vlastností komponenty Pilot Light

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

```
PilotLight('YELLOW','SHINE');
```

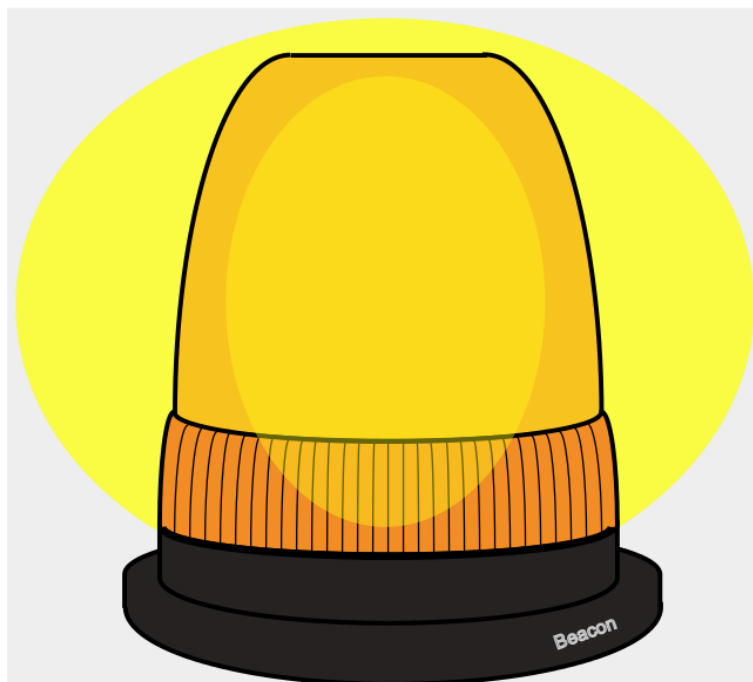
Příklad 26: Obslužná funkce barvy a světelných efektů komponenty Pilot Light

1. **Parametr:** Výběr barvy kontrolky.
Jednotka: ['RED']-červená, ['GREEN']-zelená, ['BLUE']-modrá, ['YELLOW']-žlutá.
2. **Parametr:** Nastavení svícení, blikání, či naopak zhasnutí kontrolky.
Jednotka: ['SHINE']-rozsvícení, ['BLINK']-rozblikání, ['RESET']-zhasnutí.

8.9 Beacon 1

Beacon 1 představuje výstražný maják (viz Obrázek 25), jehož úkolem je grafické znázornění nevhodné, nebo dokonce nebezpečné situace na dashboardu. Při vzniku takovéto situace dojde k rozblikání majáku s definovatelnou periodou blikání.

Lidské oko je velmi citlivé na žlutou barvu. Pokud výstražný maják pravidelně bliká světlem žluté barvy, je pouze malá pravděpodobnost, že bude obsluhou přehlédnut.



Obrázek 25: Komponenta Beacon 1

I u této komponenty lze měnit velkou řadu vlastností jako velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 25), zapnutí či vypnutí blikání a periodu, s kterou mají blikat. Veškeré tyto změny zprostředkovává obslužná funkce Beacon_1 se sedmi parametry.

```
Beacon_1(450, 0, 0, 'show', "#EEEEEE", 'START', 300);
```

Příklad 27: Obslužná funkce komponenty Beacon 1

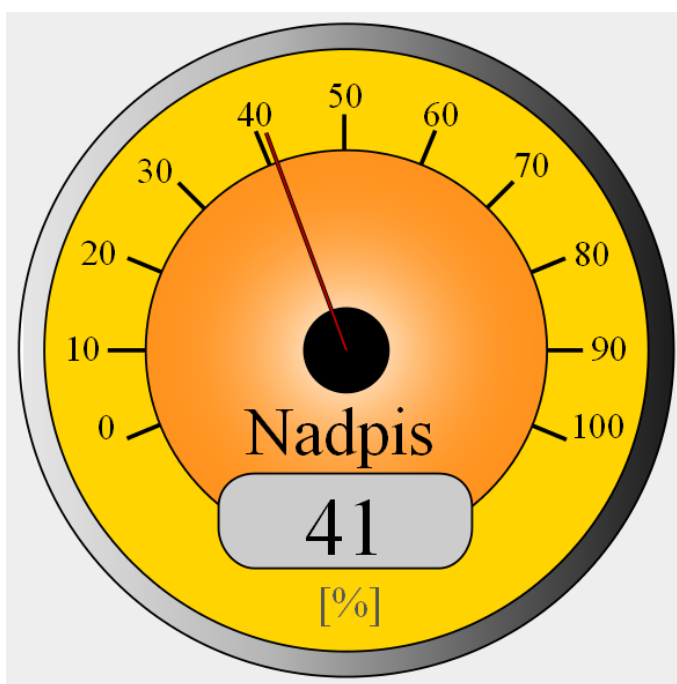
1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barevy ["red"], nebo hex. číslo barvy ["#000000"].
6. **Parametr:** Zapínání a vypínání blikání majáku.

Hodnota: ['START'] nebo ['STOP'].

7. **Parametr:** Číslo, které představuje periodu blikání majáku v milisekundách.

8.10 Gauge_1

Gauge_1 je vizualizační prvek typu měřidlo (viz Obrázek 26). Pomocí ručičkového ukazatele zobrazuje hodnotu v předem zvoleném rozsahu. Zobrazovaná hodnota je v číselné podobě zapsána ve spodní části komponenty, kde se nachází také měnitelný nadpis a popisek. V popisku je možno uvést například jednotku, ve které je zobrazená hodnota udána (na Obrázku 26 jako [%]).



Obrázek 26: Komponenta Gauge_1

U Gauge_1 je měnitelná velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 26). Všechny tyto změny jsou zapouzdřeny ve volací funkci Gauge_1_properties, která zprostředkovává pouze vlastnosti. Druhá funkce, Gauge_1, slouží k nastavení nadpisu, popisku, rozsahu hodnot a zadání konkrétní zobrazené hodnoty. Touto hodnotou může být pochopitelně také dynamicky měnící se proměnná.

```
Gauge_1_properties(600,0,0,'show','#EEEEEE');
```

Příklad 28: Obslužná funkce vlastností komponenty Gauge_1

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

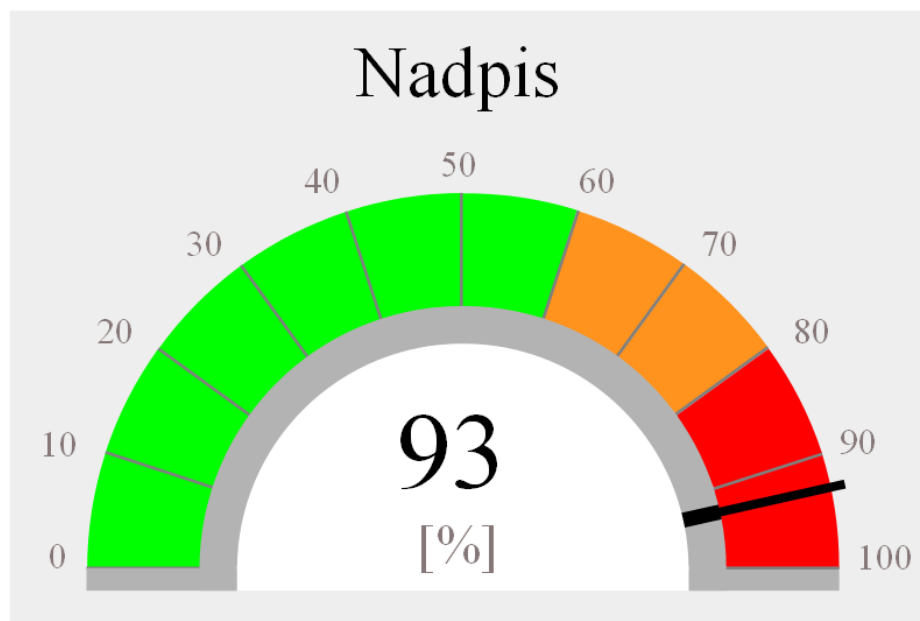
```
Gauge_1 ( 'Nadpis', 'Popisek', 100, 41 ) ;
```

Příklad 29: Obslužná funkce komponenty Gauge_1

1. **Parametr:** Nadpis zobrazené hodnoty.
Jednotka: Libovolný krátký text.
2. **Parametr:** Popisek zobrazené hodnoty.
Jednotka: Libovolný krátký text.
3. **Parametr:** Číslo, představující maximální hodnotu rozsahu měřidla. Ostatní hodnoty rozsahu se ekvivalentně přepočítají.
4. **Parametr:** Číslo, představující zobrazovanou hodnotu na měřidle.

8.11 Gauge_2

Gauge_2 je taktéž vizualizační prvek typu měřidlo (viz Obrázek 27). Pomocí ručičkového ukazatele zobrazuje hodnotu v předem zvoleném rozsahu. Tato hodnota je také v číselné podobě zapsána ve spodní části komponenty, kde se také nachází měnitelný popisek. V horní části komponenty je libovolně nastavitelný nadpis. Popisek je určen například k zobrazení jednotky, ve které je hodnota udána (na obrázku 27 jako [%]).



Obrázek 27: Komponenta Gauge_2

Gauge_2 je navržen tak, aby ho bylo možno přizpůsobit na jakýkoliv dashboard. Je takřka celý modifikovatelný. Měnit lze velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 27) a barvy všech deseti rozsahových úseků (na Obrázku 27 je zvoleno 6 zelených, 2 oranžové a 2 červené úseky). Všechny tyto změny jsou zapouzdřeny ve volací funkci Gauge_2_properties, která zprostředkovává pouze vlastnosti. Dále je možno změnit nadpis, popisek, rozsahu hodnot a pochopitelně zobrazovanou hodnotu, pomocí druhé obslužné funkce s názvem Gauge_2.

```
Gauge_2_properties(800, 0, 0, 'show', "#EEEEEE", "#00FF00",
"#00FF00", "#00FF00", "#00FF00", "#00FF00", "#00FF00",
"#FF931E", "#FF931E", "red", "red");
```

Příklad 30: Obslužná funkce vlastností komponenty Gauge_2

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].

5. **Parametr:** Barva vymezení rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
6. **Parametr:** Barva první rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
7. **Parametr:** Barva druhé rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
8. **Parametr:** Barva třetí rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
9. **Parametr:** Barva čtvrté rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
10. **Parametr:** Barva páté rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
11. **Parametr:** Barva šesté rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
12. **Parametr:** Barva sedmé rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
13. **Parametr:** Barva osmé rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
14. **Parametr:** Barva deváté rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
15. **Parametr:** Barva desáté rozsahové části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

```
Gauge_2('Nadpis', ' [%]', 100, 93);
```

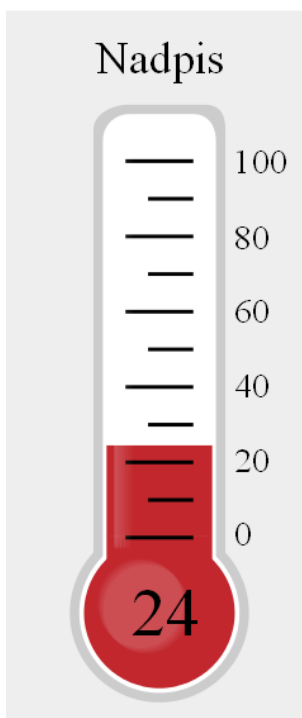
Příklad 31: Obslužná funkce komponenty Gauge_2

1. **Parametr:** Nadpis zobrazené hodnoty.
Jednotka: Libovolný krátký text.
2. **Parametr:** Popisek zobrazené hodnoty.
Jednotka: Libovolný krátký text.
3. **Parametr:** Číslo, představující maximální hodnotu rozsahu měřidla. Ostatní hodnoty rozsahu se ekvivalentně přepočítají.
4. **Parametr:** Číslo, představující zobrazovanou hodnotu na měřidle.

8.12 Gauge_3_thermometer

V případě Gauge_3_thermometer jde o lineární svislé měřidlo (viz Obrázek 28), založené na podobnosti se rtuťovým teploměrem. Při stoupající zobrazované hodnotě bude médium uvnitř virtuálního teploměru taktéž stoupat. Tato komponenta se hodí

především pro vizualizaci významné teploty na dashboardu. Pro zobrazení různých výkonnostních ukazatelů není příliš vhodná.



Obrázek 28: Komponenta Gauge_3_thermometer

Změnit u této komponenty můžeme velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 28) a barvu virtuálního měřicího media. Všechny tyto změny jsou zapouzdřeny v obslužné funkci `Gauge_3_thermometer_properties`, která zprostředkovává pouze vlastnosti. Dále je možno změnit nadpis, rozsahu hodnot a pochopitelně zobrazovanou hodnotu, a to pomocí druhé obslužné funkce s názvem `Gauge_3_thermometer`.

```
Gauge_3_thermometer_properties(250, 0, 0, 'show',  
"#EEEEEE", "#C1272D");
```

Příklad 32: Obslužná funkce vlastností komponenty `Gauge_3_thermometer_properties`

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixels.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixels.

3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezení rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
6. **Parametr:** Barva virtuálního měřicího media.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

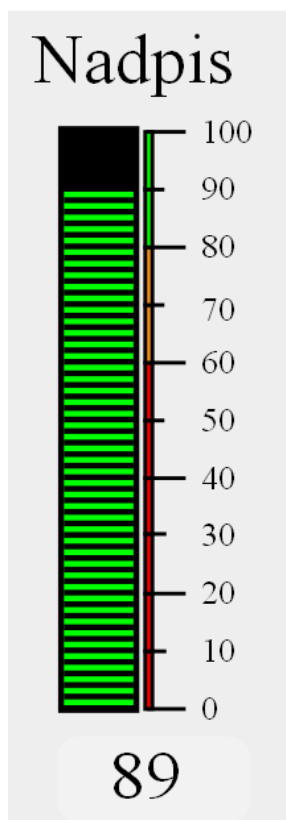
```
Gauge_3_thermometer('Nadpis', 100, 24);
```

Příklad 33: Obslužná funkce komponenty Gauge_3_thermometer

1. **Parametr:** Nadpis zobrazené hodnoty.
Jednotka: Libovolný krátký text.
2. **Parametr:** Číslo, představující maximální hodnotu rozsahu měřidla. Ostatní hodnoty rozsahu se ekvivalentně přepočítají.
3. **Parametr:** Číslo, představující zobrazovanou hodnotu na měřidle.

8.13 Gauge_5

Gauge_5 je lineární svislé měřidlo přizpůsobené na zobrazování jakýchkoliv hodnot na dashboardu, včetně veškerých výkonnostních ukazatelů (viz obrázek 29). Stoupá-li zobrazovaná hodnota, bude taktéž stoupat i virtuální vizualizační medium. Jelikož se mění barva vizualizačního media v závislosti na zobrazované hodnotě, je ihned z Gauge_5 jasné, zda je hodnota ve správných mezích či nikoli.



Obrázek 29: Komponenta Gauge_5

Tato komponenta umožňuje změnu rozsahu zobrazované hodnoty jak v číselné, tak i v barevné podobě (změna barev obdélníků mezi čísly rozsahu). Počet barev vymezující rozsah je, s ohledem na přehlednost komponenty, omezen na tři barvy. Vizualizační medium vždy nese barvu v závislosti na barevném rozsahu, kde se právě nachází (viz Obrázek 29). Pokud tedy bude, za stejného nastavení parametrů komponenty, jako je na obrázku 29, například zobrazovaná hodnota $x \in <0;60>$ medium je červené, $x \in (60;80>$ medium je oranžové, $x \in (80;100>$ medium je zelené.

Dále zde můžeme měnit velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 29). Všechny doposud zmiňované změny jsou zapouzdřeny volací funkcí `Gauge_5_properties`. Dále je možno pomocí druhé funkce `Gauge_5` měnit nadpis, rozsahové úrovně, maximální rozsah a pochopitelně zobrazovanou hodnotu.

```
Gauge_5_properties(225, 0, 0, 'show', "#EEEEEE",
"#FF0000", "#F7931E", "#00FF00");
```

Příklad 34: Obslužná funkce vlastností komponenty `Gauge_5_properties`

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
6. **Parametr:** Barva první části rozsahového vymezení (od spodní strany) a virtuálního media.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
7. **Parametr:** Barva druhé části rozsahového vymezení (od spodní strany) a virtuálního media.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
8. **Parametr:** Barva třetí části rozsahového vymezení (od spodní strany) a virtuálního media.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

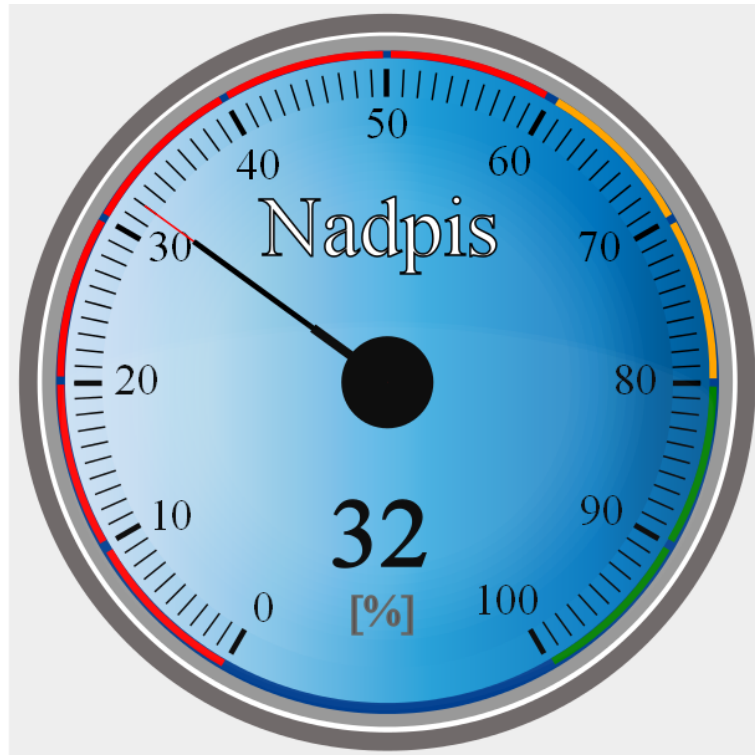
```
Gauge_5('Nadpis', 60, 80, 100, 24);
```

Příklad 35: Obslužná funkce komponenty Gauge_5

1. **Parametr:** Nadpis zobrazené hodnoty.
Jednotka: Libovolný krátký text.
2. **Parametr:** Číslo, představující první hodnotu vymezení rozsahu
3. **Parametr:** Číslo, představující druhou hodnotu vymezení rozsahu
4. **Parametr:** Číslo, představující maximální hodnotu rozsahu měřidla. Ostatní hodnoty rozsahu se ekvivalentně přepočítají.
5. **Parametr:** Číslo, představující zobrazovanou hodnotu na měřidle.

8.14 Gauge_6

Gauge_6 spadá do skupiny stupnicových indikátorů s ručičkovým ukazatelem (viz Obrázek 30). Při grafickém návrhu této komponenty bylo snahou docílit co možná nejpřesnějšího a nejpropracovanějšího vzhledu doplněného o dostatečnou modifikovatelnost.



Obrázek 30: Komponenta Gauge_6

V rámci modifikovatelnosti lze měnit velikost celého objektu, odsazení od levé a horní hrany prohlížeče (pozice objektu na dashboardu), viditelnost a barvu pozadí (šedý rámeček na obrázku 30), jednotlivé barvy deseti rozsahových výsečí (viz Obrázek 30: 6 červených, 2 oranžové a 2 zelené). Všechny tyto změny jsou zapouzdřeny ve volací funkci `Gauge_6_properties`, která zprostředkovává pouze vlastnosti. Druhá obslužná funkce, `Gauge_6`, umožňuje měnit nadpis, popisek (na obrázku 30 jako [%]), maximální rozsah hodnot a pochopitelně také zobrazovanou hodnotu.

```
Gauge_6_properties(615, 0, 0, 'show', "#EEEEEE", "red",  
"red", "red", "red", "red", "red", "orange",  
"orange", "green", "green");
```

Příklad 36: Obslužná funkce vlastností komponenty `Gauge_6_properties`

1. **Parametr:** Číslo, které nastavuje šířku celého objektu (vymezen šedým rámečkem). Pro zachování velikostního poměru je výška přepočítávána a nastavena automaticky.
Jednotka: Pixely.
2. **Parametr:** Číslo, znamenající odsazení celého objektu od horní hrany prohlížeče.
Jednotka: Pixely.
3. **Parametr:** Číslo, znamenající odsazení celého objektu od levé hrany prohlížeče.
Jednotka: Pixely.
4. **Parametr:** Zapínání a vypínání viditelnosti rámečku na pozadí.
Hodnota: ['show'] nebo ['hide'].
5. **Parametr:** Barva vymezovacího rámečku na pozadí.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
6. **Parametr:** Barva rozsahového vymezení první části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
7. **Parametr:** Barva rozsahového vymezení druhé části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
8. **Parametr:** Barva rozsahového vymezení třetí části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
9. **Parametr:** Barva rozsahového vymezení čtvrté části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
10. **Parametr:** Barva rozsahového vymezení páté části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
11. **Parametr:** Barva rozsahového vymezení šesté části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
12. **Parametr:** Barva rozsahového vymezení sedmé části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
13. **Parametr:** Barva rozsahového vymezení osmé části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
14. **Parametr:** Barva rozsahového vymezení deváté části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].
15. **Parametr:** Barva rozsahového vymezení desáté části zleva.
Hodnota: Anglický překlad barvy ["red"], nebo hex. číslo barvy ["#000000"].

```
Gauge_6('Nadpis', '[%]', 100, 32);
```

Příklad 37: Obslužná funkce komponenty Gauge_6

1. **Parametr:** Nadpis zobrazené hodnoty.
Jednotka: Libovolný krátký text.
2. **Parametr:** Popisek zobrazené hodnoty.
Jednotka: Libovolný krátký text.

3. **Parametr:** Číslo, představující maximální hodnotu rozsahu měřidla. Ostatní hodnoty rozsahu se ekvivalentně přepočítají.
4. **Parametr:** Číslo, představující zobrazovanou hodnotu na měřidle.

9. ZÁVĚR

Cílem této bakalářské práce bylo vytvoření grafických vizualizačních komponent pro systém COMES Dashboard.

Nejprve bylo zapotřebí charakterizovat systém COMES, tedy že se jedná o výrobně informační systém úrovně MES, který má za úkol sběr a archivaci dat, na jejichž základě řídí a optimalizuje výrobu. Nezbytností pro tuto práci je také začlenění dashboardu do systému COMES.

V další části je provedena rešerše komerčních i nekomerčních komponent pro tvorbu dashboard vizualizace, a z ní plynoucí vhodné typy.

Následně se dostáváme k vlastnímu návrhu a realizaci zmiňovaných komponent. V souladu se zadáním, jsou všechny vizualizační prvky vytvořeny v SVG. Interaktivita je zprostředkována použitím JavaScriptu, doplněného o knihovnu jQuery.

Inspirací při vytváření vzhledu komponent byly, na doporučení konzultanta, prvky z dashboardů volně přístupných na internetových stránkách či vzhled reálných objektů.

Při tvorbě grafické podoby bylo snahou docílit co nejvhodnějšího kompromisu mezi propracovaností a názorností uspořádání, z kterého bude na první pohled zřejmé, co je vyjadřováno. Objekty jsou založeny na rysech reálných předmětů. Pokud by totiž například tlačítka na dashboardu měla stejnou podobu jako tlačítka používaná v systémech Windows (velká většina lidí je zvyklá při zobrazení hlášky automaticky stisknout OK), docházelo by často k chybným zásahům do činností na dashboardu. Zároveň nesmí být návrh příliš složitý. Pozornost obsluhy by nebyla věnována vyjadřované informaci, ale samotnému vzhledu, což není žádoucí. Sestavování grafického vzhledu bylo provázeno myšlenkou: „ Čím důležitější vyjadřovaná informace, tím výraznější a přehlednější musí být!“.

Důraz byl, kvůli možnému použití na různých řídicích panelech, kladen také na modifikovatelnost vzhledu komponent. U všech objektů lze bez ztráty kvality měnit celkovou velikost, pozici na dashboardu, barvu a viditelnost bezprostředního pozadí. Výjimkou je pouze prvek Linear Potenciometer, u kterého, jak z principu koncepce vyplývá, nelze měnit pouze celkovou velikost. Dále komponenty nabízejí možnost změny některých grafických částí (především jejich barev či číselných rozsahů), pro zlepšení objektivnosti a orientace na vizualizaci. Jelikož je SVG založeno na XML, kdy je každý obrázek zastoupen textovým kódem, lze v těchto prvcích jednoduše měnit jakýkoli grafický atribut. Stačí mít pouze dostatečné znalosti SVG.

Tvorbě komponent Gauge_1, Gauge_2, Gauge_3_Thermometer, Gauge_5 a Gauge_6 byla věnována velká pozornost, jelikož takovéto typy komponent jsou v praxi velmi oblíbeny. Ručičkové ukazatele a lineární měřidla disponují rychlým orientačním přečtením údaje. Doplněním o číslíkový formát hodnoty je výrazně zlepšena také přesnost odečtu. V seznamu komponent chybí element Gauge_4. Tento

objekt byl zkonstruován a naprogramován, ale kvůli své složitosti programové části nefunguje se stoprocentní spolehlivostí. Proto zde není uveden.

Programová část komponent byla vytvořena co nejefektivněji, aby bylo dosaženo spolehlivého a přesného chování prvků. Program všech komponent je umístěn v samostatném JavaScriptovém souboru s názvem „functions“ (viz CD – ve složce „js“). Při použití některé z komponent se tedy nemusí kopírovat složitý kód funkcí objektů do kódu dashboardu, ale stačí pouze nalinkování knihovny. Pro usnadnění pochopení a jednodušší práci s veškerým programem, bylo vše zpracováno se stejnou posloupností parametrů funkcí komponent. Například první parametr obslužných funkcí je vždy velikost objektu. U složitějších vizualizačních prvků jsou obslužné funkce dvě. Jedna slouží pro nastavení obecných vlastností (velikost, umístění atd.). Stačí ji tedy inicializovat pouze jednou. Druhá je určena k přímému vykonávání akcí, jako je například u komponenty RedGreen Signalization zapnutí světelného efektu. Může být tedy volána mnohonásobně vícekrát.

Zobrazení grafických vizualizačních komponent je možné pouze na novějších verzích webových prohlížečů (viz Příloha 1). Způsobeno je to kvůli použití poměrně moderních technologií, především tedy SVG. Dále musí být v prohlížeči povoleno scriptování.

V rámci této bakalářské práce bylo vytvořeno celkem čtrnáct plně funkčních grafických vizualizačních komponent, určených pro případné použití na COMES Dashboard. Všechny prvky lze použít na jednom řídicím panelu, aniž by se navzájem negativně ovlivňovaly. Pokud by bylo zapotřebí na jednom dashboardu zobrazit několik stejných komponent, je nezbytné nakopírovat a správně indexovat jak SVG část v kódu dashboardu, tak programovou část příslušného prvku v souboru „functions“.

Použitá literatura

- [1] ING. BRÁZDA, Roman a Vlastimil BRAUN. Výrobní informační systém COMES 3. *Výrobní informační systém COMES 3* [online]. 1-2. 2011, s. 1-2 [cit. 2013-12-08]. Dostupné z: http://www.compas.cz/UserFiles/Informace/odborne_clanky/Comes-verze3.pdf
- [2] BRAUN, Vlastimil. COMES pro systémové integrátory automatizace. *COMES pro systémové integrátory automatizace* [online]. Březen 2013 [cit. 2013-12-08]. Dostupné z: http://www.compas.cz/UserFiles/Informace/odborne_clanky/ControlEngineering/COMES%20pro%20systemove%20integratory%20automatizace.pdf
- [3] ING. BRÁZDA, Roman. Výrobní informační systém COMES a jeho použití. *Výrobní informační systém COMES a jeho použití* [online]. 10/2010 [cit. 2013-12-08]. Dostupné z: http://www.compas.cz/UserFiles/Informace/odborne_clanky/Technika-clanek-2010-10-COMES-SK.pdf
- [4] COMPAS S.R.O. *Výrobní informační systém COMES verze 3: Úvod do systému*. Roman Brázda. [firemní dokument]. Žďár nad Sázavou, 2011. MES11-D-10-08-AA-COMES_Uvod_CZ.pptx.
- [5] COMPAS S.R.O. *Výrobní informační systém COMES verze 3: Modul Logon*. Roman Brázda. [firemní dokument]. Žďár nad Sázavou, 2010. MES10-D-10-02-01-COMES_Logon_CZ.pptx.
- [6] COMPAS S.R.O. *Výrobní informační systém COMES verze 3: Modul Historian*. Roman Brázda. [firemní dokument]. Žďár nad Sázavou, 2010. MES10-D-10-03-01-COMES_Historian_CZ.pptx.
- [7] COMPAS S.R.O. *Výrobní informační systém COMES verze 3: Modul Modeller*. Roman Brázda. [firemní dokument]. Žďár nad Sázavou, 2011. MES10-D-10-07-01-COMES_Modeller_CZ.pptx.
- [8] COMPAS S.R.O. *Výrobní informační systém COMES verze 3: Modul Traceability*. Roman Brázda. [firemní dokument]. Žďár nad Sázavou, 2010. MES10-D-10-05-01-COMES_Traceability_CZ.pptx.

- [9] COMPAS S.R.O. *Výrobní informační systém COMES verze 3: Modul Batch*. Roman Brázda. [firemní dokument]. Žďár nad Sázavou, 2010. MES10-D-10-06-01-COMES_Batch_CZ.pptx.
- [10] COME OEE: Vlastnosti COMES OEE. COMPAS AUTOMATIZACE. *COMES OEE* [online]. 2013 [cit. 2013-12-11]. Dostupné z: <http://www.oee.cz/comes-oee>
- [11] GREIF, Tomáš: *Analytik dat* [online]. Aktualizováno 2013-2-15. [cit. 2013-11-13]. Dostupné z: <http://www.analytikdat.cz/index.php/entry/co-je-dashboard>
- [12] COMPAS automatizace, spol.s.r.o.: Měření efektivity výroby OEE v praxi. [online]. 9/2013 [cit. 2013-12-11]. Dostupné z: http://www.compas.cz/UserFiles/Informace/odborne_clanky/IT-Systems/Mereni_efektivita_vyroby_OEE_v_praxi_09-13.pdf
- [13] WONDERWARE: Dashboard. In: . *OEE numbers shine a light on packaging machine performance* [online]. 2013 [cit. 2013-12-11]. Dostupné z: <http://www.packworld.com/controls/networking/oee-numbers-%E2%80%A8shine-light-packaging-machine-performance>
- [14] MULTI-DNC. In: *Automatic Machine Data Collection for Machine Monitoring* [online]. 2012 [cit. 2013-12-11]. Dostupné z: <http://www.multi-dnc.com/products/monitoring/>
- [15] TELERIK. RadChart - Telerik's ASP.NET Chart. TELERIK. *ASP.NET* [online]. 2013, aktualizováno 11/14/2013 [cit. 2013-12-09]. Dostupné z: <http://demos.telerik.com/aspnet-ajax/chart/examples/overview/defaultcs.aspx>
- [16] Oracle Developer Tools and Thoughts. In: *A migration dashboard with Forms JavaBeans* [online]. 2008 [cit. 2013-12-09]. Dostupné z: <http://www.degenio.com/2008/09/a-migration-dashboard-with-forms-pjcs/>
- [17] KOSEK Jakub: *SVG tvůrce, stručný průvodce SVG grafikou* [online]. [cit. 2013-11-13]. Dostupné z: <http://svg.jkoweb.cz/coje.xhtml>
- [18] Adaptic. *Rastrová grafika* [online]. 2014, č. 1 [cit. 2014-04-20]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/rastrova-grafika/>

- [19] Adaptic. *Vektorová grafika* [online]. 2014, č. 1 [cit. 2014-04-20]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/vektorova-grafika/>
- [20] HOUSE, Lorien a PEARLMAN. InformIT: *Introductino to SVG* [online]. 2003, s. 6, 2003-07-25 [cit. 2014-04-18]. Dostupné z: <http://www.informit.com/articles/article.aspx?p=99036>
- [21] W3Schools.com. *SVG Basic* [online]. 2014 [cit. 2014-04-20]. Dostupné z: http://www.w3schools.com/svg/svg_inhtml.asp
- [22] W3Schools.com. *SVG Shapes* [online]. 2014 [cit. 2014-04-20]. Dostupné z: http://www.w3schools.com/svg/svg_inhtml.asp
- [23] TIŠNOVSKÝ, Pavel. ROOT.CZ. *Podpora skriptování v grafickém formátu SVG* [online]. 2007, č. 1 [cit. 2014-04-20]. Dostupné z: <http://www.root.cz/clanky/podpora-skriptovani-v-grafickem-formatu-svg/>
- [24] Jakpsátweb.cz. *JavaScript* [online]. 2014 [cit. 2014-04-20]. Dostupné z: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [25] Pěstujeme web. *Úvod do JavaScriptu* [online]. 2010 [cit. 2014-04-20]. Dostupné z: <http://www.pestujemeweb.cz/obsah/javascript/javascript-uvod.php>
- [26] JADRNÝ, Tomáš. JQuery návod. *Úvodní článek* [online]. 2010, č. 1 [cit. 2014-04-20]. Dostupné z: <http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek>
- [27] Devbook. *5. díl - Tvorba funkcí a úvod do jQuery* [online]. 2014, č. 1 [cit. 2014-04-20]. Dostupné z: <http://www.devbook.cz/javascript-tutorial-funkcionalni-programovani-a-jquery-webova-kalkulacka/>

Seznam zkratek

ERP - Enterprise Resource Planning (Podnikový informační systém)

MES - Manufacturing Execution Systems (Výrobní informační systém)

MCS - Manufacturing Control Systems (Výrobní řídicí systém)

IT - Information Technology (Informační Technologie)

CCI - COMES Communication Interface (COMES komunikační rozhraní)

CSV - Comma-separated values (Hodnoty oddělené čárkami)

KPI - Key Performance Indicators (Klíčový výkonnostní ukazatel)

OEE - Overall equipment effectiveness (Celková efektivita výroby)

SVG - Scalable Vector Graphics (Škálovatelná vektorová grafika)

XML - Extensible Markup Language (Rozšiřitelný značkovací jazyk)

DOM - Document Object Model (Objektový model dokumentu)

ICC - International Color Consortium (Mezinárodní konsorcium pro barvu)

CSS - Cascading Style Sheets (Kaskádové styly)

HTML - HyperText Markup Language (hypertextový značkovací jazyk)

JS - JavaScript

WWW - World Wide Web

PC - Personal Computer (Osobní počítač)

Seznam příloh

Příloha 1: Tabulka podpory vytvořených komponent webovými prohlížeči

Příloha 2: Obsah CD

Příloha 1:

Tabulka podpory vytvořených komponent webovými prohlížeči

	Internet Explorer	Mozilla Firefox	Google Chrom	Opera	Safari
JavaScript	od verze 3	všechny verze	všechny verze	všechny verze	všechny verze
jQuery	od verze 6 (9)	od verze 2.0	od verze 8.0	od verze 10.6	od verze 3.0
SVG	od verze 9	od verze 3.6	od verze 8.0	od verze 11.0	od verze 3.0
Vytvořené komponenty	od verze 9	od verze 3.6	od verze 8.0	od verze 11.0	od verze 3.0

Příloha 2:

Obsah CD

- PDF formát bakalářské práce
- složka „Components“ zahrnující: HTML soubory jednotlivých komponent (Button, Start-Stop Buttons, Reset Button, ONxOFF Switch, Linear Potenciometer, Column Signalization, RedGreen Signalization, Pilot Light, Beacon 1, Gauge_1, Gauge_2, Gauge_3_thermometer, Gauge_5 a Gauge_6), příklad použití všech komponent na jednom místě (HTML soubor s názvem „ALL“) a složku „js“ (uvnitř „js“ se nachází JavaScriptové knihovny „functions“, „jquery-1.11.0.min“ a soubor SSZareni, kde jsou doplňující efekty pro některé signalizační komponenty)