



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Vysoká škola báňská – Technická univerzita Ostrava

Fakulta strojní



NÁVODY KE CVIČENÍ Z PŘEDMĚTU PROCESNÍ SYSTÉMY

Učební text k předmětu „Procesní systémy“

Lenka Landryová

Ostrava 2011



Tyto studijní materiály vznikly za finanční podpory Evropského sociálního fondu (ESF) a rozpočtu České republiky v rámci řešení projektu OP VK CZ.1.07/2.3.00/09.0147 „Vzdělávání lidských zdrojů pro rozvoj týmů ve vývoji a výzkumu“.

Název: Učební text předmětu „Procesní systémy“

Autor: Lenka Landryová

Vydání: první, 2011

Počet stran: 91

Náklad: <xx (minimum je 5)>

Studijní materiály pro studijní obor Strojní inženýrství Fakulty strojní

Jazyková korektura: nebyla provedena.



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tyto studijní materiály vznikly za finanční podpory Evropského sociálního fondu a rozpočtu České republiky v rámci řešení projektu Operačního programu Vzdělávání pro konkurenceschopnost.



Název: Vzdělávání lidských zdrojů pro rozvoj týmů ve vývoji a výzkumu

Číslo: CZ.1.07/2.3.00/09.0147

Realizace: Vysoká škola báňská – Technická univerzita Ostrava

© Lenka Landryová

© Vysoká škola báňská – Technická univerzita Ostrava

ISBN 978-80-248-2765-0

POKYNY KE STUDIU

Procesní systémy

Pro předmět 1. semestru oboru Strojní inženýrství jste obdrželi studijní balík obsahující:

- integrované skriptum pro distanční studium obsahující i pokyny ke studiu,
- přístup do e-learningového portálu obsahující doplňkové animace vybraných částí kapitol.

Prerekvizity

Pro studium této opory se předpokládá znalost na úrovni absolventa bakalářského studia oboru Strojírenství.

Cíl učební opory

Cílem je poskytnutí učebních materiálů pro seznámení se se základními pojmy automatizace procesního průmyslu. Většina informací pochází z aplikovaného výzkumu a vývoje a případových studií aplikací těchto výsledků v praxi. Po prostudování modulu by měl student být schopen zvládnout analýzu a návrh specifikací funkcí pro uživatelské rozhraní procesních systémů a dokumentovat vlastní projekt tak, aby se uvedené informace týkaly technických funkcí a nikoli obchodních výhod, které projekt může přinášet.

Pro koho je předmět určen

Modul je zařazen do magisterského studia většiny oborů studijního programu Strojní inženýrství, ale může jej studovat i zájemce z kteréhokoliv jiného oboru, pokud splňuje požadované prerekvizity.

Skriptum se dělí na části, kapitoly, které odpovídají logickému dělení studované látky, ale nejsou stejně obsáhlé. Předpokládaná doba ke studiu kapitoly se může výrazně lišit, proto jsou velké kapitoly děleny dále na číslované podkapitoly, a těm odpovídá níže popsaná struktura.

Při studiu každé kapitoly doporučujeme následující postup:



Čas ke studiu: xx hodin

Na úvod kapitoly je uveden čas potřebný k prostudování látky. Čas je orientační a může vám sloužit jako hrubé vodítko pro rozvržení studia celého předmětu či kapitoly. Někomu se čas může zdát příliš dlouhý, někomu naopak. Jsou studenti, kteří se s touto problematikou ještě nikdy nesetkali a naopak takoví, kteří již v tomto oboru mají bohaté zkušenosti.



Cíl: Po prostudování tohoto odstavce budete umět

+ Popsat ...

+ Definovat ...

Okamžitě potom jsou uvedeny cíle, kterých máte dosáhnout po prostudování této kapitoly – konkrétní dovednosti, znalosti.



Výklad

Následuje vlastní výklad studované látky, zavedení nových pojmů, jejich vysvětlení, vše doprovázeno obrázky, tabulkami, řešenými příklady, odkazy na animace.



Shrnutí pojmů

Na závěr kapitoly jsou zopakovány hlavní pojmy, které si v ní máte osvojit. Pokud některému z nich ještě nerozumíte, vraťte se k nim ještě jednou.



Otázky

Pro ověření, že jste dobře a úplně látku kapitoly zvládli, máte k dispozici několik teoretických otázek.



Úlohy k řešení

Protože většina teoretických pojmů tohoto předmětu má bezprostřední význam a využití v praxi, jsou Vám nakonec předkládány i praktické úlohy k řešení. V nich je hlavním významem předmětu schopnost aplikovat čerstvě nabyté znalosti pro řešení reálných situací.



Klíč k řešení

Řešení zadaných příkladů, které doprovázejí výklad, je k dispozici v doprovodných animacích.

Úspěšné a příjemné studium s tímto učebním textem Vám přeje autorka.

Lenka Landryová

OBSAH

1	ŘÍZENÍ PROCESŮ A JEHO ÚROVNĚ	8
1.1	Funkční hierarchie procesů.....	8
1.1.1	Informace a informační proces	9
1.1.2	Modely procesů.....	10
1.1.3	Koncepce kooperace informačních systémů CPM	11
1.2	Časový aspekt v procesních systémech.....	13
1.2.1	Proces dávkování směsi a řízení zásoby sypkých hmot.....	13
1.2.2	Integrační smyčková regulace	14
1.2.3	Kaskádová regulace v tepelné technice	15
1.3	Řízení s podporou grafického rozhraní	17
1.3.1	Základní znaky rozhraní člověk-stroj	17
1.3.2	Typický uživatel SCADA/HMI	18
2	VIZUALIZACE PROCESŮ	20
2.1	Koncepce vývoje aplikací.....	20
2.1.1	Koncepce vývoje podle definovaných proměnných.....	20
2.1.2	Koncepce vývoje definováním komponent objektů.....	20
2.2	Značky a písmenné kódy v projektové dokumentaci.....	22
2.2.1	Význam značek projektové dokumentace pro tvorbu knihoven objektů.....	22
2.2.2	Písmenné kódy v projektové dokumentaci.....	24
2.2.3	Objekty pro návrh vizualizačních obrazovek	25
3	VIZUALIZAČNÍ PROSTŘEDÍ	29
3.1	Tvorba aplikací ve vizualizačním prostředí.....	29
3.1.1	Manažer aplikací	29
3.1.2	Vývojové prostředí aplikací.....	30
3.1.3	Běhové – runtime - prostředí aplikací	31
3.2	Návrh obrazovky operátorského pracoviště.....	32
3.3	Kreslení objektů	34
3.3.1	Jednoduché a složené objekty.....	34
3.3.2	Komplexní a předem definované objekty	35
4	ANIMACE OBJEKTŮ A VAZBY NA PROMĚNNÉ VELIČINY	39
4.1	Sběr dat	39
4.1.1	Vzorkování a logování hodnot.....	39

4.1.2	Interní a externí zdroje sběru dat	41
4.2	Datové typy proměnných	44
4.2.1	Volba typu veličin	44
4.2.2	Uživatелеm definované veličiny	44
4.2.3	Systémové veličiny	45
4.3	Animace objektů	48
4.3.1	Konfigurace animovaného objektu	48
4.3.2	Skript pro animace objektů	48
4.4	Simulace procesu	51
4.5	Komunikace s externím prostředím	53
5	ALARMY A JEJICH KONFIGURACE	56
5.1	Zprávy, hlášení, události a alarmy	56
5.1.1	Účel zpráv a hlášení	56
5.1.2	Priorita zpráv a reakce operátora na zprávy	57
5.1.3	Událost a alarm	57
5.1.4	Ukládání zpráv	58
5.2	Typy alarmu	59
5.2.1	Více-stavové alarmy	59
5.2.2	Potvrzené alarmové stavy	59
5.2.3	Alarmy aktuální a historické	60
5.3	Notifikace událostí na runtime obrazovky operátorských pracovišť	62
5.3.1	Uživatelé grafického rozhraní	62
5.3.2	Moderní směry grafického rozhraní	63
5.3.3	Komunikace událostí v architekturách klient – server	65
5.3.4	Rozhraní webových prohlížečů	65
5.3.5	Služby pro notifikaci událostí	66
6	TRENDY A JEJICH KONFIGURACE	68
6.1	Typy vizualizací průběhů	68
6.1.1	Reálné trendy	68
6.1.2	Historické trendy	69
6.2	Logování a ukládání logů	71
7	ANALÝZA PROCESŮ	73
7.1	Sledování ztrát ve výrobní procesu a celkové efektivity výroby	73

7.1.1	Ztráty ve výrobním procesu	73
7.2	Statistické řízení procesů	76
7.2.1	Regulační diagramy.....	77
7.2.2	Histogram.....	78
7.2.3	Pareto diagram	80
8	ZNALOSTI V PROCESNÍM ŘÍZENÍ	82
8.1	Znalosti	82
8.1.1	Získávání znalostí	83
8.1.2	Metody prezentace znalostí.....	83
8.1.3	Heuristické řešení problémů.....	84
8.2	Znalosti při posouzení rizika procesu.....	85
9	SPOLEHLIVOST A ZABEZPEČENÍ PROCESNÍCH SYSTÉMŮ.....	86
9.1	Spolehlivost systému	87
9.2	Bezpečnost přístupu k aplikaci	88
9.2.1	Přihlašování uživatelů a jejich identifikace, autentizace a autorizace.....	88
9.2.2	Způsob identifikace uživatelů v operačních systémech počítačů	89
9.2.3	Přihlašování uživatelů do aplikace procesního systému	89

1 ŘÍZENÍ PROCESŮ A JEHO ÚROVNĚ

Kapitola vysvětluje pojem supervizní řízení a význam použitých zkratk SCADA/HMI, MES, OOP, OEM, KPI a 7/24. Pohlíží na řízení procesů z hlediska hierarchie jednotlivých úrovní a času. Vysvětluje rozdíl mezi automatickým řízením a řízením automatizovaným s podporou grafického rozhraní.



Čas ke studiu: 4 hodiny



Cíl: Po prostudování této kapitoly budete umět

- ✚ definovat pojem supervizní řízení
- ✚ popsat charakteristické znaky rozhraní člověk-stroj
- ✚ popsat informační proces
- ✚ vysvětlit rozdíl mezi informačním a řídicím systémem
- ✚ určit role uživatele systému SCADA/HMI



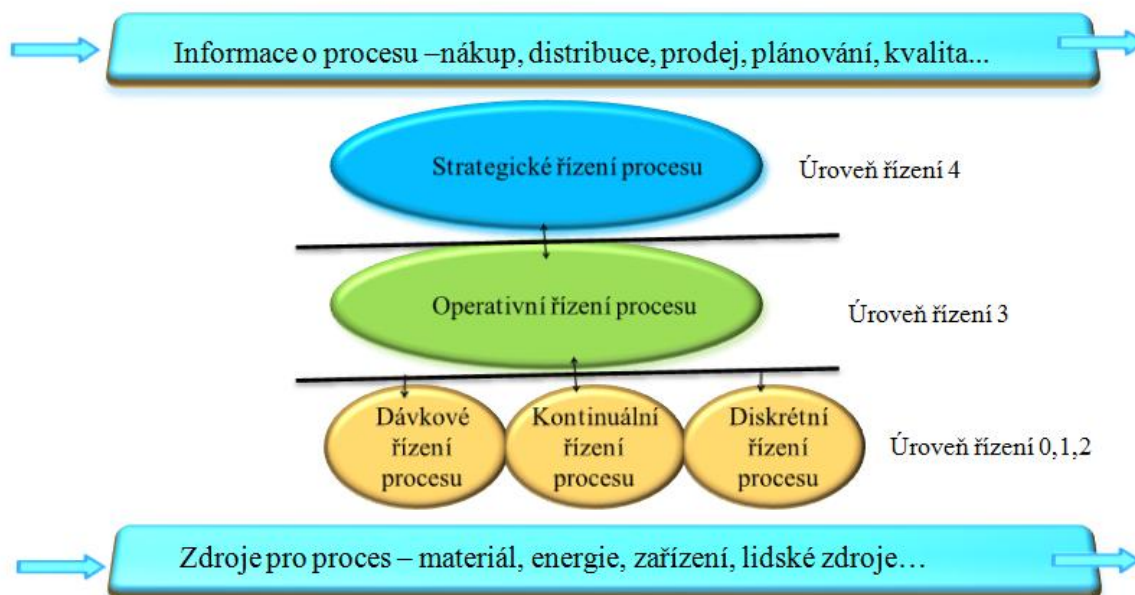
Výklad

SCADA představuje zkratku pro systémy supervizního řízení a sběr dat z angličtiny *Supervisory Control and Data Acquisition*, MES označuje systémy řízení výroby z anglického *Manufacturing Execution Systems*. Z hlediska řízení procesu, technologie nebo výroby tak rozlišujeme různé úrovně řízení nebo supervize procesů (z anglického *to supervise*, tj. dohlížet). V jednotlivých úrovních řízení pak dochází k interakci řízeného a řídicího objektu, systému nebo jeho prvku. Mluvíme –li o základní úrovni řízení, v níž se převážně vyskytuje působení stroje na stroj, nazveme tuto úroveň automatickým řízením. Vyznačuje se zejména automatickým chodem předem naprogramovaným, odladěným a spuštěným v reálném prostředí, běžícím bez zásahu člověka. Úroveň řízení, která vyžaduje dohled obsluhy a jinou interakci člověka nazveme řízením automatizovaným s podporou rozhraní člověk – stroj, a tedy supervizním řízením.

1.1 Funkční hierarchie procesů

Podniky bývají organizačně uspořádané, ať už účelově nebo historicky, do úseků a oddělení s rozdělenými systémy, zodpovědnostmi a kompetencemi. Hranice mezi těmito systémy bývají často i bariérami mezi automatizovanými a jinak organizovanými činnostmi. A někdy už jen název oddělení nebo úseku bývá zavádějící. Například distribuce může zodpovídat za fyzickou distribuci pitné vody do domácností, ale hlavními aktivitami nebude administrace této distribuce smluvními podmínkami s koncovými odběrateli (úroveň 4), ale spíše údržba distribuční sítě pitné vody (úroveň 0,1,2). Název oddělení nebo úseku nestačí. Pod úsek výroby v jednotlivých podnicích může spadat i dlouhodobé plánování výroby a záleží jen na daném podniku, zda jej bude řídit z úrovně 3, a nebo také z úrovně 4.

Obrázek 1.1 znázorňuje schéma pro úrovně řízení podle standardu ISA95, který formalizuje procesy z hlediska funkcí identifikovaných v procesních systémech typu SCADA/HMI a MES. Formalizace a standardizace těchto funkcí a aktivit je důležitým předpokladem pro identifikaci informačních toků.



Obrázek 1.1 - Úrovně řízení procesu podle standardu ISA95

1.1.1 Informace a informační proces

Pojem informace pochází z latinského *informare*, uvádět v tvar, dávat tvar, formovat, tvořit, zobrazovat, představovat, vytvářet představu a nebo také *informatio*, instruovat, dát někomu znalosti, dát něčemu formu. Podle výkladu pojmu lze tedy definovat význam informace jako míru organizace systému a jeho uspořádání nebo proces v lidském vědomí.

Samotný informační proces se skládá z následujících informačních činností, které lze definovat jako aktivity informačního systému:

- Vznik informace
- Akvizice – sběr
- Vstupní zpracování
- Uložení informace
- Vyhledání informace
- Výstupní zpracování
- Vyhodnocení
- Akční zásah
- Užití informace

1.1.2 Modely procesů

S pojmem model a modelování se v inženýrské praxi setkáváme často, zejména pak v souvislosti:

- s datovými modely v databázích, jejichž úlohou je popis struktury dat, elementů, jejich atributů a vztahů;
- se simulačními metodami používanými v odborné i vědecké praxi opřenými o simulační model, jenž je ve výsledku počítačovým programem, algoritmem;
- s objektovými modely definovanými pro objektově orientované programování, jinými slovy modely v programovacím jazyce orientovaném okolo takzvaných objektů.

Modely procesů mohou být důležité už v případě, kdy se seznamujeme s výrobním procesem a procházíme výrobními halami, skladovými prostory, laboratořemi, údržbou daného podniku a mapujeme funkce a zodpovědnosti jednotlivých úseků. Model v takovém případě poskytuje obecný pohled na podnik důležitý i z hlediska jeho řízení. Podle toho, zda úsek pracuje na úrovni řízeného technologického procesu (úrovně 0,1,2, 3) nebo spadá pod strategické řízení podniku (úroveň 4) lze určit i činnosti přímo spojené s výrobou. Mohou to být i činnosti týkající se bezpečnosti, spolehlivosti, efektivity, kvality a údržby, záleží však na vnitřním uspořádání podniku, které může být různé i pro podniky působící ve stejném výrobním sektoru.

Modely řízení procesů výrobních podniků byly definovány v minulosti v pěti úrovních:

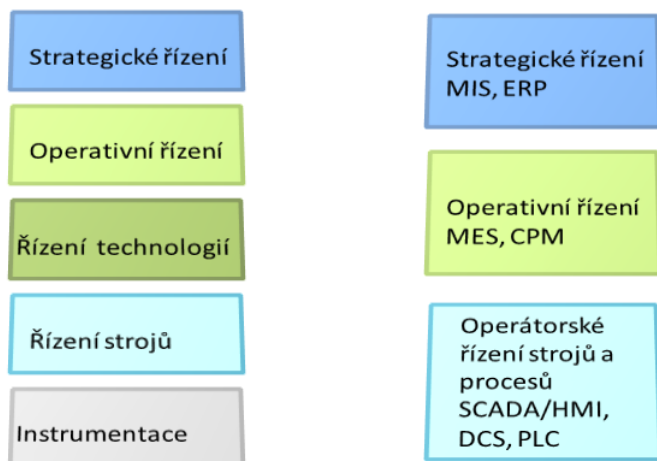
- Strategické řízení
- Operativní řízení
- Taktické řízení
- Operátorské řízení strojů a procesů
- Stroje a procesy

Modely řízení procesů výrobních podniků v současnost jsou již tři úrovně:

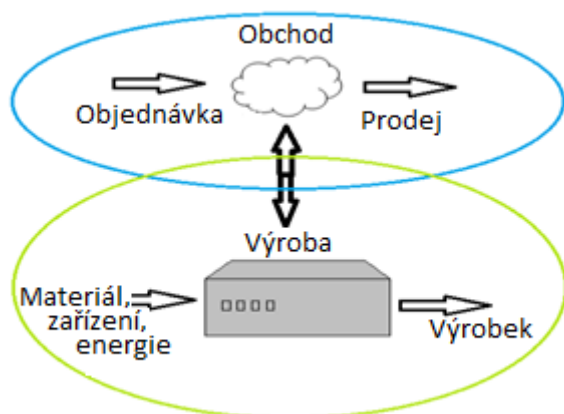
- Strategické řízení s podporou informačních systémů MIS (*Management Information System*), systémů plánování podnikových zdrojů ERP (*Enterprise Resource Planning*), on-line systémů eBusiness;
- Operativní řízení s podporou výrobních informačních systémů MES (*Manufacturing Execution Systems*) a kooperativních informačních systémů CPM (*Collaborative Production Management*) pro zajištění operativního plánování, přípravy a monitorování výroby;
- Operátorské řízení, logické řízení, instrumentace, ovládání, regulace, umělá inteligence, měření, monitorování výroby s podporou distribuovaných řídicích systémů DCS (*Distributed Control Systems*), programovatelných logických automatů PLC (*Programmable Logic Controllers*), supervizních řídicích systémů SCADA, HMI pro řízení výrobních technologií a strojů (viz. výše vysvětlené zkratky na začátku Kap. 1.1).

1.1.3 Koncepce kooperace informačních systémů CPM

Výhoda současného řešení oproti minulosti vězí v redukci počtu řídicích úrovní z pěti na tři a odstranění bariéry, zejména komunikační, mezi těmito úrovněmi řízení za současného zautomatizování řízení funkcemi procesních systémů pro plánování, přípravu a monitorování výroby, viz. *Obrázek 1.2* a *Obrázek 1.3*.



Obrázek 1.2 – Redukce úrovní řízení podle modelu ISA95 s koncepcí CPM



Obrázek 1.3 – Komunikační bariéra mezi úrovněmi řízení z pohledu informačního a procesního systému

Koncepce kooperace informačních systémů CPM (*Collaborative Production Management*) podporuje výměnu dat a vzájemnou komunikaci a odstraňuje komunikační bariéry. Informace se mezi různými informačními systémy vyměňují na základě XML dokumentů, které jsou obecným formátem pro reprezentaci dat schváleným standardem W3C, jsou čitelné počítačem i lidmi, používají UNICODE znakovou sadu a jsou nezávislé na platformě. viz. také Kap. 5.3.3. Tato koncepce není omezena jen na výrobní podnik, ale zahrnuje i dodavatele, partnery a zákazníky:

- Dodavatel sleduje, co a v jakém množství výrobci chybí, upravuje cenu, výrobu;
- Zákazník zjistí, kdy je jeho požadavek vyřízen.

Technické a softwarové prostředky již problémem v zavádění moderních procesních systémů nejsou. Překážkou jsou samotní lidé, uživatelé systémů. Neumí systémy využívat, neví, k čemu systémy a systémová řešení jsou, mají negativní zkušenosti v podniku, například s nedokončenými projekty, s nevratnými investicemi nebo s nepřátelským uživatelským rozhraním systémů, což představuje výzvu pro vývojáře, programátory a architekty systémových řešení, více v Kap. 0.

Požadavky kladené na MES a CPM systémy z hlediska jejich uživatelů jsou:

- Sběr dat, sledování toku materiálu
- Řízení výrobních jednotek: aktuální stav výroby
- Řízení dokumentace: návody, receptury
- Analýza výkonnosti
- Řízení lidských zdrojů: školení, docházka, znalosti
- Řízení údržby
- Řízení kvality: SPC
- Plánování a rozvrhy, harmonogramy
- Přidělování kapacit a zdrojů

Typickými uživateli MES a CPM systémů z hlediska hierarchického uspořádání výrobního podniku shora dolů jsou:

- Management podniku: plán a realizace zakázek, nákup materiálu a snížení nákladů;
- Vedoucí pracovníci: ušetření na skladu, rychlejší zavádění nových výrobků, genealogie výrobku, v jakém množství, v jaké kvalitě co a kde vyrábí;
- Obsluha linek a výrobních strojů: přehled o pořadí výrobních požadavků, o dokumentaci, plánu v nejbližší době;
- Oddělení údržby: používání strojů, plán preventivní údržby;
- Plánovači: kolik materiálu je na skladě, kolik je vyrobeno, zrušení a změny v dodávkách.



Shrnutí pojmů 1.1.

Supervizní řízení (SCADA) je nasazováno na úrovni řízení vyžadující interakci člověka s technickými prostředky. Tuto úroveň řízení označujeme za automatizovanou - úroveň řízení 2 a 3 - *Level 2, Level 3*. Úroveň řízení, na které řídí stroj jiný stroj nebo zařízení je úroveň automatického řízení – *Level 0, Level 1*.



Otázky 1.1.

Co znamenají zkratky SCADA/HMI, MES a CPM?

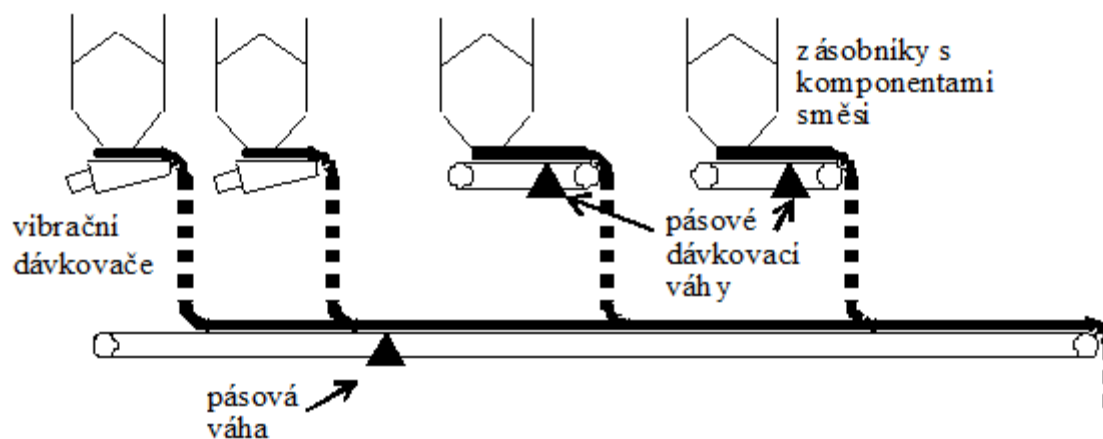
Jaký je rozdíl mezi automatickým a supervizním řízením?

1.2 Časový aspekt v procesních systémech

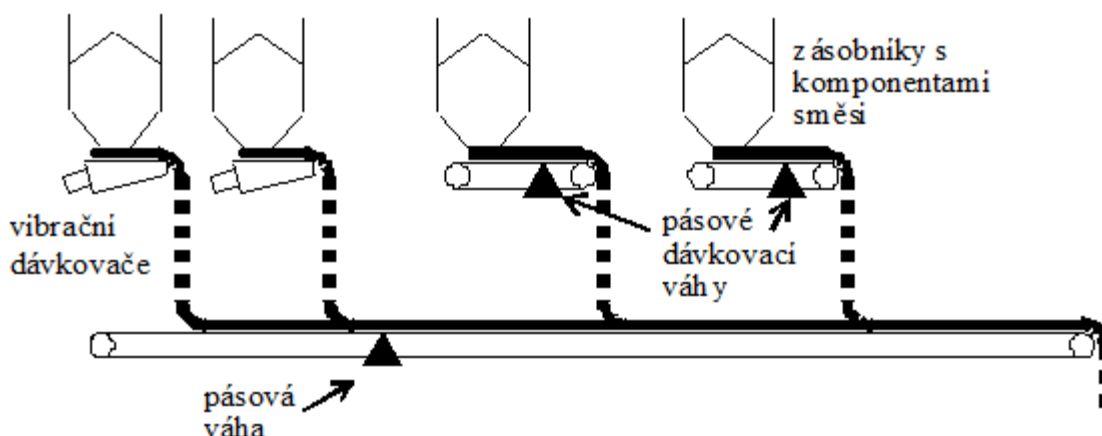
SCADA systémy řídí aplikace, které jsou provozovány v plánovaném nebo v požadovaném časovém rozsahu, a mají tak částečně charakter jak dávkového, tak real-time systému. Řízení v reálném čase je definováno jako "řízení týkající se výpočetního výkonu v reálném čase, v němž probíhá související fyzikální proces" [Boyer, S.A., 1993]. Vztahuje se k odezvě řídicího systému na změny souvisejícího fyzikálního procesu. Signál požadované veličiny je přiveden na vstup řídicího systému bez časového zpoždění. Řídicí systém pracuje s tímto signálem na provedení řídicího algoritmu v nejkratším možném čase a posílá akční signál, který vstupuje do procesu okamžitě. Do důsledku je tedy real-time řídicí systém pouze takový, který řídí bez časového zpoždění nebo prodlení v čase mezi sběrem měřených dat z procesu a exekucí řídicího signálu. Ve skutečnosti však všechny řídicí systémy pracují s nějakým časovým zpožděním. Ty z nich, které řídí systém bez měřitelného časového zpoždění lze považovat za systémy pracujícími v reálném čase. Většinu systémů řízení kontinuálních procesů můžeme považovat za systém řízení v reálném čase, protože časová zpoždění na straně řízení jsou velice krátká. Systémy dávkového řízení procesu jsou v tomto protikladem systému řízení v reálném čase.

1.2.1 Proces dávkování směsi a řízení zásoby sypkých hmot

Regulace průtoku sypkých hmot a materiálů má široké uplatnění při vytváření směsí nebo pro dodávku materiálu do technologického zařízení. Příkladem může být zavážení paliva do kotle nebo spékání aglomerátu v hutích či výroba stavebních hmot.



Obrázek 1.4 - Dávkování směsi sypkých hmot

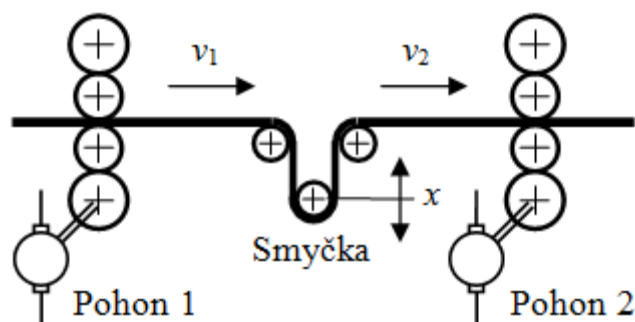


Obrázek 1.4 znázorňuje vytváření směsi jako nepřetržitý kontinuální technologický proces. Dávkuje se obvykle na dopravní pás, přičemž směs je dále promíchána v průtočném míchacím zařízení, například v otáčejícím se bubnu s mírně skloněnou osou, a pokračuje k dalšímu zpracování. Kontinuální dávkování vyžaduje průběžné měření průtoku směsi po dopravníku, a řadí se tak z pohledu řízení procesu mezi kontinuální procesy, i když jde o technologii dávkování.

1.2.2 Integrační smyčková regulace

Jedná se o regulaci používanou v případech, kdy je zapotřebí synchronizovat otáčky dvou různých zařízení. Příkladem může být pohon navíjecího bubnu, přičemž drát nebo pás je podáván svou vlastní rychlostí. Je evidentní, že průměr cívky se při navíjení mění, což vyžaduje přizpůsobení otáček navíjecího bubnu. Jiným příkladem je papírenský stroj, u kterého se předává pás papíru od jedné dvojice válců k jiné dvojici válců, přičemž každá dvojice válců má vlastní pohon. Nedokonalým řešením regulace pohonu by bylo společné zadání otáček oběma regulátorům nebo odvození žádané hodnoty otáček druhého regulátoru od skutečných otáček prvního regulátoru. I sebemenší chyba v měření otáček by vedla k integrování jejich rozdílu a posléze k roztržení pásu papíru nebo naopak k jeho havarijnímu nakupení mezi válci.

Tomuto jevu se čelí vložení smyčky, jejíž délka se reguluje změnou rychlosti pohybu jednoho z pohonů. Obrázek 1.5 ukazuje uspořádání obvodu integrální smyčkové regulace. Znázorněn je regulovaný pás s válci, dvěma pohony a vloženou smyčkou. Pohon 2 je řízen nezávisle na pohonu 1. Tento pohon určuje rychlost pohybu pásu v_2 . Délka smyčky x je výsledkem integrace rozdílu rychlostí v_1 a v_2 . Regulátor určuje otáčky pohonu 1, a tím i rychlost v_1 . Rychlost v_2 je měřenou poruchou. Regulační odchylkou je velikost prodloužení nebo zkrácení smyčky x od nulové polohy.

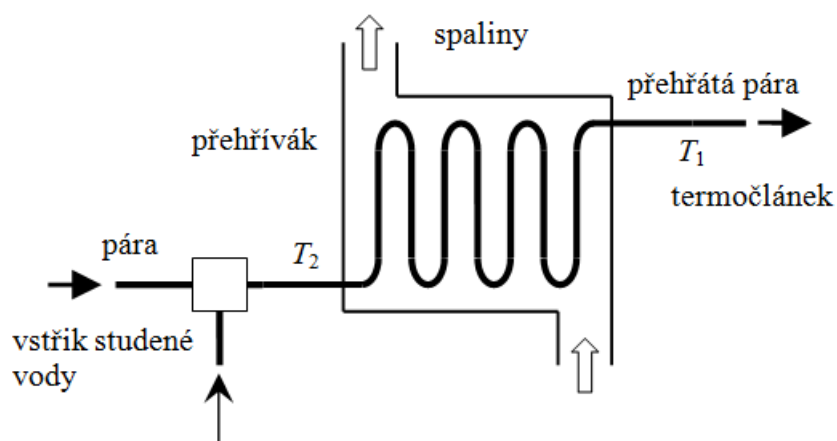


Obrázek 1.5 - Integroční smyčková regulace

1.2.3 Kaskádová regulace v tepelné technice

V tepelné technice je běžné, že regulovaná soustava má pomalou odezvu, tj. relativně velké dopravní zpoždění, a pracuje s velkou časovou konstantou. Po změně akční veličiny následuje velmi pomalá odezva v časovém průběhu regulované teploty. Těmto pro regulaci nevýhodným podmínkám se čelí tzv. kaskádovou regulací. V regulované soustavě se vyhledá teplota, jejíž odezva je podstatně rychlejší, než odezva regulované teploty.

Příklad kaskádové regulace je ilustrován na přehřívačů tlakové páry. Znázorněný přehřívač je průtočný kotel. Teplota páry pro turbínu je například požadována velmi stálá. Kolísání teploty páry, a tím i turbíny, může urychlit únavu materiálu, ze kterého je vyrobena. Teplota výstupní páry se reguluje změnami průtoku studené vody, která je vstřikovávána do páry před přehřívačem. Odezva na vstřik studené vody ve změně teploty páry před ohřívákem je téměř okamžitá.



Obrázek 1.6 - Kaskádová regulace

Kaskádové zapojení pracuje se dvěma regulátory. První regulátor reguluje teplotu páry na výstupu přehříváku. Akční veličina tohoto regulátoru je žádanou hodnotou teploty páry na vstupu do přehříváku pro druhý regulátor. Tento druhý regulátor pracuje s regulovanou veličinou, kterou je teplota páry na vstupu do přehříváku, a s akční veličinou, kterou je průtok studené vody. První regulátor musí pracovat bez trvalé regulační odchylky, druhý regulátor musí pracovat s co nejrychlejší odezvou.



Shrnutí pojmů 1.2.

Řízení v reálném čase je definováno jako "řízení týkající se výpočetního výkonu v reálném čase, v němž probíhá související fyzikální proces". Real-time řídicí systém je pouze takový systém, který řídí bez časového zpoždění nebo prodlení v čase mezi sběrem měřených dat z procesu a exekucí řídicího signálu.



Otázky 1.2.

Jak je definováno řízení v reálném čase?

Jaký typ procesu představuje zavážení paliva do kotle nebo spékání aglomerátu v hutích či výroba stavebních hmot?

1.3 Řízení s podporou grafického rozhraní

Předchozí příklady popisovaly procesní děje slovně a také schematicky v jednoduchých obrázcích. Grafická znázornění jsou pro člověka a jeho rozhodování v řízení procesů velice důležitá. Pro vizuální vnímání člověka však velkou úlohu hraje rozhraní, které mu zprostředkovává grafickou podobu systému a jeho prvků, procesu, technologie a stroje.

1.3.1 Základní znaky rozhraní člověk-stroj

Rozhraní člověk stroj (HMI) je rozhraní takového systému, ve kterém člověk pro svou práci používá prostředky pro interakci se stroji, s technickým zařízením (Level 2, Level 3). Jedná se o rozhraní pro lepší a efektivnější řízení a ovládání strojů a zajištění zpětné vazby důležité pro rozhodování a řízení. Důležité je při tom zachovat systém přátelský pro práci člověka a přinést mu vhodné pracovní prostředí s dostatkem (nikoli přebytkem) informací a v přívětivé formě.

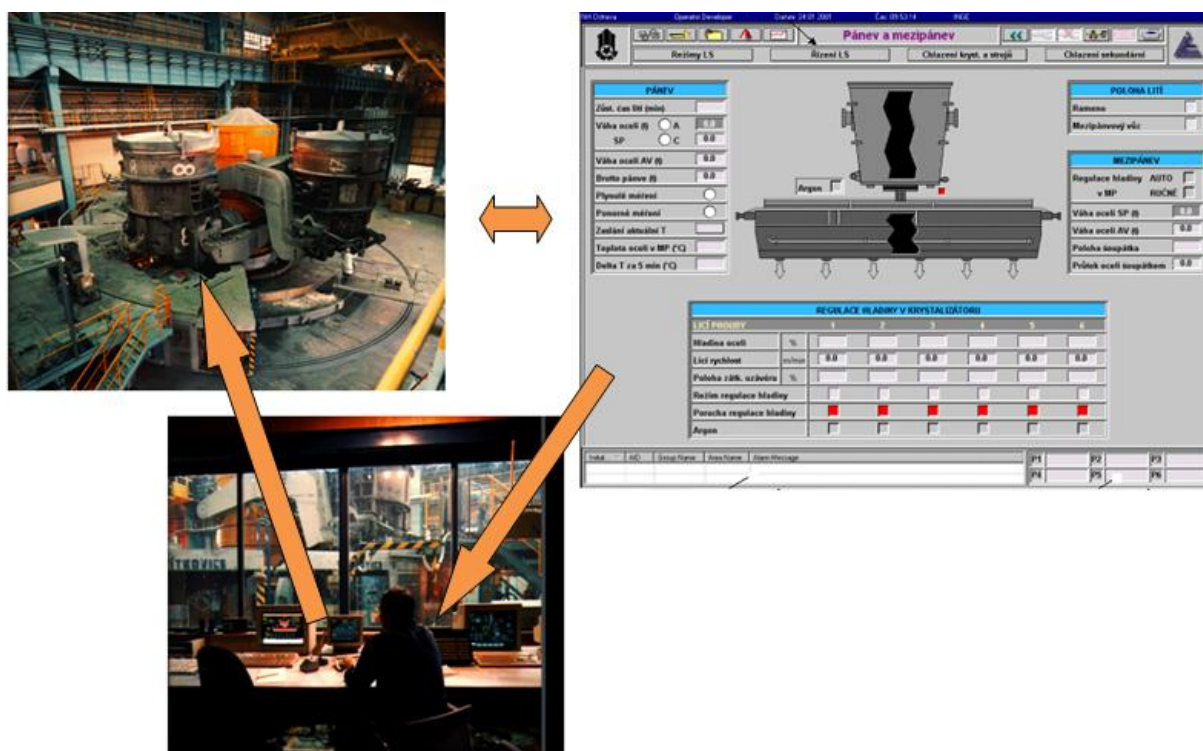
Systémy supervizního řízení a sběru dat poskytují díky HMI grafické možnosti pro rychlé vytváření názorných aplikací vizualizací procesů a jejich případné modifikace. Vyznačují se **snadným použitím, objektově orientovanou grafikou, účinnou komunikací a flexibilní architekturou systémů.**

Snadné použití lze definovat jako vlastnost produktu, prostředí nebo aplikace, pro jejichž ovládání a práci s nimi není vyžadována žádná schopnost nad rámec intuitivního uživatelského přístupu. *Obrázek 1.7* přibližuje typické pracoviště obsluhy komplexního systému, které umožňuje ovládat a řídit procesy vizualizační aplikací instalovanou pro člověka do jeho počítače a z jeho obrazovky monitoru ve velínu uprostřed těžkého provozu.

Objektově orientovaná grafika vychází z objektově orientovaného programování neboli OOP. Jde o model v programovacím jazyce orientovaném okolo takzvaných objektů a dat, na rozdíl od programování logických funkcí vycházejících z matematických modelů a Booleovy algebry. V minulosti byly programy zobrazeny jako logické procedury, které si vzaly vstupní data, zpracovaly je a vytvořily výstupní data. Programování bylo o tom, jak zapsat logiku, a nikoli jak nadefinovat data. Oproti tomu, objektově orientované programování pracuje s objekty, se kterými chceme manipulovat. Objekty zobrazené na obrazovce monitoru mohou být co do tvaru, barev, velikosti jednoduché nebo mohou být složené a konfigurovatelné pro komunikaci a pro vstupní a výstupní data.

Účinná komunikace umožňuje zpracovat velké množství dat a informací a srozumitelným a dostupným způsobem je zobrazit na velké obrazovky a/nebo efektivně navržené displeje. Online data automaticky reprezentují proces na obrazovku pomocí dynamických displejů, kontrolních přepínačů, blikajících světel, tabulek s aktuálními hodnotami, křivkami průběhů atd. To vše je umožněno sdílením a předáváním dat v rámci systémové architektury.

Flexibilní architektura systému je systémová architektura, která podporuje sběr dat z procesu, technologie nebo výroby a umožňuje dále data integrovat a šířit do podnikového informačního systému a vyšší úrovně řízení.



Obrázek 1.7 - Online data z výrobního procesu prezentovaná na obrazovku operátorského pracoviště

1.3.2 Typický uživatel SCADA/HMI

Zatímco na úrovni automatického řízení jsou monitorována a ukládána data jako žádaná hodnota, odchylka, poruchová veličina, na úrovni automatizovaného řízení s podporou supervize jsou tyto hodnoty již posílány podle jejich užitečnosti na různá místa a různým uživatelům. Například tlak, teplota, hladina, hustota apod. směřují na rozhraní supervizního řízení systému nádrží. Na vyšší úrovni řízení a rozhodování jsou z těchto hodnot počítány klíčové indikátory výkonu (Key Performance Indicators neboli KPIs) jako je kvalita, efektivita, aktiva, výnos, zisk a tyto údaje přeměňují data v informace poskytované a sdílené ve stejné architektuře a ve stejné síti se systémem SCADA/HMI.

Kdo pracuje se systémy SCADA/HMI:

- Procesní inženýr
- Výrobní ředitel
- Koncový uživatel
- Systémový integrátor
- Výrobce originálních zařízení (OEM)

Procesní inženýr se zabývá ve své praxi různými typy procesů a systémy SCADA/HMI mu umožňují procesy modelovat, analyzovat a simulovat v podmínkách blízkých provozním podmínkám, nikoli laboratorním, s cílem procesy a jejich parametry optimalizovat pro splnění daného cíle řízení.

Výrobní ředitel má ve své praxi za úkol zajištění efektivního využívání výrobní techniky během výroby konkrétního produktu. Také jemu SCADA/HMI přináší nástroje pro vizualizaci, testování a diagnostiku dílčích strojů i výrobních komplexů.

Koncový uživatel obsluhuje technologii a/nebo techniku a je v bezprostředním kontaktu s rozhranním člověk-stroj systémů SCADA/HMI podle typu provozu 7/24 i sedm dní v týdnu 24 hodin denně.

Systémový integrátor je klíčovým partnerem při vývoji a dodávkách zakázkových zařízení pro průmyslové výrobní komplexy a zajišťuje realizaci systémové integrace jednotlivých komponent, datových serverů a sítí.

Výrobce originálních zařízení (OEM) používá základní komponenty vhodné k integraci do softwarového prostředí SCADA/HMI od různých výrobců a nabízí systémové řešení výrobním podnikům a provozům.

Vzhledem k charakteru systémového řešení se mohou role uživatelů postupem času a získáváním zkušeností s konkrétními podmínkami nasazení SCADA/HMI ve výrobním procesu dále vyvíjet nebo překrývat. Koncový uživatel může mít přímý vliv na vývoj, rozšíření realizace a inovaci systému nasazeného v provozu. Systémový integrátor vyvine na základě požadavků koncového uživatele, procesního inženýra či výrobního ředitele nové komponenty odpovídající inovativním a ověřeným teoretickým laboratorním metodám pro uplatnění v praxi.



Shrnutí pojmů 1.3.

Systémy SCADA/HMI se vyznačují snadným použitím a objektově orientovanou grafikou podporovanou uživatelským rozhranním (HMI), účinnou komunikací a flexibilní architekturou systémů.

Typickými uživateli systémů SCADA/HMI jsou v průmyslové praxi procesní inženýři, výrobní ředitelé, koncoví uživatelé, systémoví integrátoři a výrobci originálních zařízení (OEM).



Otázky 1.3.

Které znaky jsou charakteristické pro HMI?

Kdo pracuje se systémy SCADA/HMI?

Co znamenají zkratky OOP, OEM, KPI a 7/24?

2 VIZUALIZACE PROCESŮ

Kapitola vysvětluje význam vizualizace v systémech SCADA/HMI a MES. Seznamuje s koncepcí vývoje vizualizační aplikace. Představuje nástroje pro její vývoj.



Čas ke studiu: 2 hodiny



Cíl: Po prostudování tohoto odstavce budete umět

- + Vysvětlit rozdíly mezi koncepcemi vývoje vizualizační aplikace
- + Seznámit se s významem základních značek projektové dokumentace pro tvorbu objektů vizualizačních aplikací



Výklad

2.1 Koncepce vývoje aplikací

První aplikace HMI a SCADA vyvíjeli vývojáři v softwaru kreslením grafiky, podobně jako v jiných grafických a vizualizačních programech, a napojením na proměnné-tagy- definované podle veličin, které reprezentují adresy v programovatelných automatech (PLC) nebo řídicích systémech. Tato koncepce vývoje může být tedy nazývaná *tag-based* neboli *podle definovaných proměnných*. Koncepce vývoje *component-object based* definováním komponent objektů pochází původně z oblasti informačních technologií. Jejím cílem je poskytovat vývojářům aplikací nástroje, které odstraní opakované programování objektů při jejich násobném použití v aplikaci a zároveň časově optimalizuje vývoj aplikace pomocí knihoven s předem definovanými základními komponentami-objekty.

2.1.1 Koncepce vývoje podle definovaných proměnných

Aplikace vyvíjená koncepcí podle definovaných proměnných je vytvářena v daném okamžiku na jednom počítači jedním vývojářem. Ten pro aplikaci kreslí objekty a vytváří obrazovky. Pro každou z obrazovek je použitými objekty vytvořena grafika a na tyto objekty jsou vývojářem definovány proměnné veličiny pro import z PLC nebo pro manuální konfigurace. Alarmy a událostní skripty jsou programovány pro každou proměnnou veličinu zvlášť. Proměnné veličiny jsou napojeny na grafické objekty vazbami, které jsou rovněž konfigurovatelné. Vývojář této aplikace oživí objekty animacemi napsáním logiky ve skriptovacím jazyce a napojením vstupních a výstupních (I/O) veličin na grafické objekty v aplikaci. Každá změna v aplikaci vyžaduje vypnutí aplikace v systému, což může být v procesu prováděno například v rámci údržby celého systému.

2.1.2 Koncepce vývoje definováním komponent objektů

Aplikaci vyvíjenou definováním komponent objektů lze vytvářet v daném okamžiku na více než jednom počítači a práci rozdělit mezi tým vývojářů. Aplikační objekt vzniká podle

šablony, která již od počátku její definice obsahuje parametry pro připojení na zařízení, které graficky reprezentuje, například objekt Ventil již obsahuje kromě základní charakteristiky objektu co do velikosti, tvaru, barvy apod. také všechny události, alarmy, zabezpečení, komunikaci s veličinami atd. Každá změna ve vývoji aplikace nebo její modifikace po nasazení aplikace do provozu pak na rozdíl od předchozí koncepce vývoje nevyžaduje vypnutí aplikace v systému, protože se ve vývojovém prostředí provede změna původního objektu (šablony) a cestou progresu změn jsou vlastnosti dědičně propojené na šablonu modifikovány za provozu zařízení.



Shrnutí pojmů 2.1.

Aplikace vyvíjená koncepcí podle definovaných proměnných je vytvářena v daném okamžiku na jednom počítači jedním vývojářem.

Aplikaci vyvíjenou definováním komponent objektů lze vytvářet v daném okamžiku na více než jednom počítači a práci rozdělit mezi tým vývojářů.



Otázky 2.1.

Jaké koncepce vývoje vizualizačních aplikací již znáte?

Co je tag, komponenta, objekt?

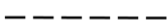
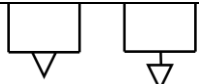
2.2 Značky a písmenné kódy v projektové dokumentaci

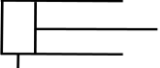
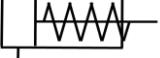
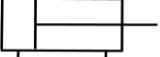
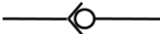
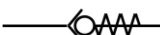
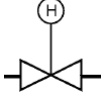
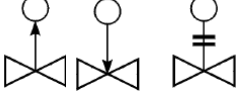
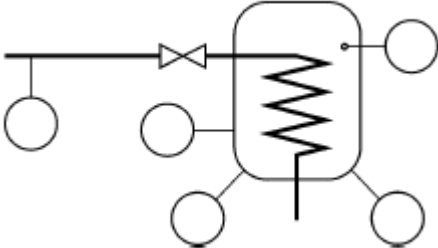
Značky používané v projektové dokumentaci určují umístění snímačů nebo přístrojů v technologickém procesu.

2.2.1 Význam značek projektové dokumentace pro tvorbu knihoven objektů

Úkolem značek je lokalizovat místo přenášených dat, specifikovat chování a přiřazovat funkcionalitu zařízení k měřicím a řídicím přístrojům. Jejich základem je obvykle kružnice pro označení přístroje, tenká čára připojená k obrysu zařízení pro vyznačení místa měření a malý kroužek na konci čáry pro určení přesné polohy měřicího místa. Některé vybrané značky pro tvorbu projektové dokumentace jsou uvedeny v Tab. 2.1.

Tab. 2.1 – Tabulka vybraného značení používaného v projektové dokumentaci

Název	Značka
provozní, zpětné a napájecí vedení, přívod energie	
řídící vedení k přenosu tlakových řídicích signálů	
proudění kapaliny	
proudění vzduchu	
výstup média (odlehčovací kanál, odfuk)	
regulace teploty s identifikačním číslem vázaným na další podrobnosti k dokumentaci	
regulace tlaku	
Ukazatel a signalizace maximální hladiny	
zásobník stlačeného vzduchu	
filtr k oddělení pevných nečistot	
zvlhčovač (olejovač), rozprašovač maziva	
chladič plynu nebo kapaliny bez označení přívodů	
elektromotor	
automatický pohon	
servopohon s točivým motorem	

Název	Značka
membránový servopohon	
Elektromagnetický servopohon	
pneumatický motor pro jeden směr pohybu	
pneumatický motor pro oba směry pohybu	
jednočinný pneumatický válec - zpětný pohyb vnější silou	
jednočinný pneumatický válec - zpětný pohyb vratnou pružinou	
dvojčinný pneumatický válec s jednostrannou pístní tyčí (pístnicí)	
všeobecná značka regulačního členu bez rozlišení typu zařízení	
ventil	
zpětný ventil	
zpětný ventil s pružinou	
škrticí prvek, clona, dýza	
stavitelný škrticí ventil	
regulační člen (obecně a ventil) s pohonem	
ruční ovladač prvek (polokružnice s vepsaným písmenem H)	
regulační člen (ventil) s automatickým pohonem (plná kružnice) doplněný ručním ovládáním (písmeno H)	
chování regulačního členu při přerušení dodávky pomocné energie automatického pohonu - regulační člen otevírá, zavírá, zůstává v dosažené poloze	
obecná značka přístroje a vyznačení místa měření	

2.2.2 Písmenné kódy v projektové dokumentaci

Písmenné kódy podle normy ČSN ISO 3511-1 ve značkách vyjadřují měřenou veličinu, případně další funkce přístroje s ohledem na zpracování informace získané měřením. První písmeno pak vyjadřuje druh monitorované veličiny, další písmena upřesňují význam, například zda se měří rozdíl, poměr, součet apod. a způsob zpracování informace, například signalizace, indikace, regulace, spínání, zabezpečení apod. Písmenka mimo kroužek nahoře mohou označovat maximální (H) nebo minimální (L) signalizovanou hodnotu, případně mimo kroužek dole označuje vhodná zkratka měření konkrétní koncentrační veličiny. Přístroje umístěné na panelu operátorských pracovišť jsou označeny vodorovným jednoduchým proškrtnutím a přístroje uvnitř rozvaděče jsou označeny proškrtnutím dvojitou čarou. Některé písmenné kódy jsou uvedeny v Tab. 2.2 a Tab. 2.3.

Tab. 2.2 – Tabulka vybraných kódů prvního písmene značení používaného v projektové dokumentaci

Písmeno	Druh měřené veličiny
D	Hustota
E	Obecně elektrická veličina
F	Průtok
G	Poloha, délka
K	Čas
L	Hladina
M	Vlhkost
P	Tlak
R	Radioaktivní záření
S	Rychlost, frekvence
T	Teplota
V	Viskozita
W	Tíhová síla, síla

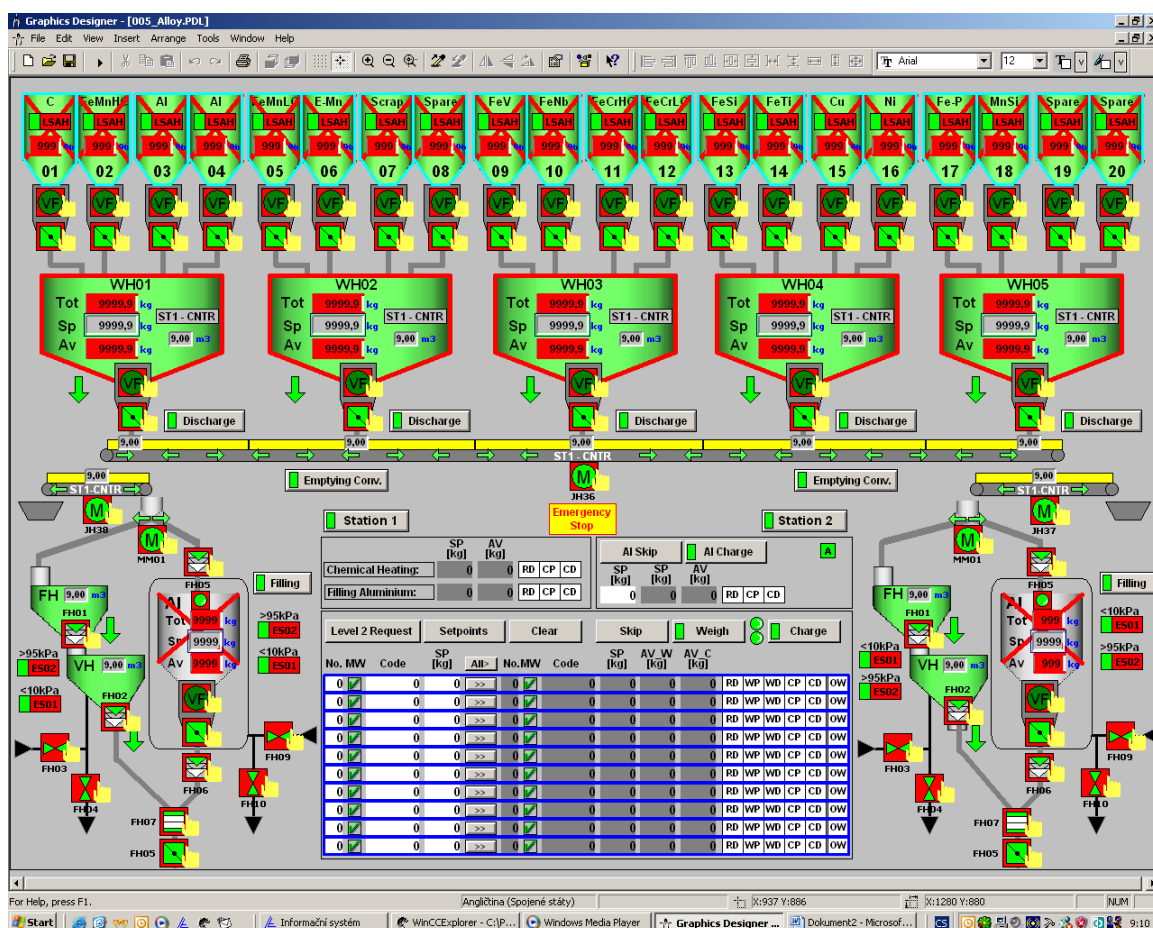
Tab. 2.3 – Tabulka vybraných kódů dalšího písmene značení používaného v projektové dokumentaci

Písmeno	Význam a způsob zpracování signálu
A	Signalizace
B	Indikace
C	Regulace
D	Rozdíl hodnot
E	Čidlo
F	Poměr
I	Ukazatel
J	Snímání
Q	Sčítání, sumace, integrace
R	Zapisování
S	Spínání
T	Vysílání
Z	Nouzová, zabezpečovací funkce

2.2.3 Objekty pro návrh vizualizačních obrazovek

Značky a písmenné kódy se staly základem pro tvorbu jednoduchých a složených objektů, které na obrazovkách vizualizovaných aplikací reprezentují strojní zařízení a jejich ovladačí a ovládané prvky. Ve vizualizačních aplikacích jsou k dispozici nástroje ke kreslení objektů, jejichž vzhled na obrazovce vychází ze zavedeného značení.

Samotná vizualizace se provádí většinou změnou vzhledu, velikosti nebo polohy příslušného objektu, který reprezentuje danou technologickou nebo výrobní součást.

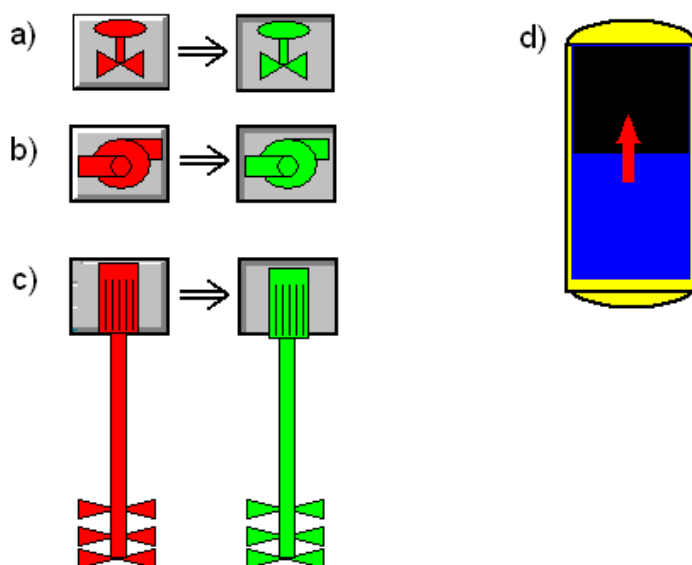


Obrázek 2.1 – Obrazovka vizualizace procesu využívající objekty vytvořené podle značek projektové dokumentace

Na příkladě ventilu se funkce objektu vizualizuje změnou jeho barvy a stínováním. Červená barva a vytlačené tlačítko symbolizují uzavřený ventil. Kapalina tedy neprotéká. Pokud je ventil zelený a tlačítko zatlačeno, kapalina protéká.

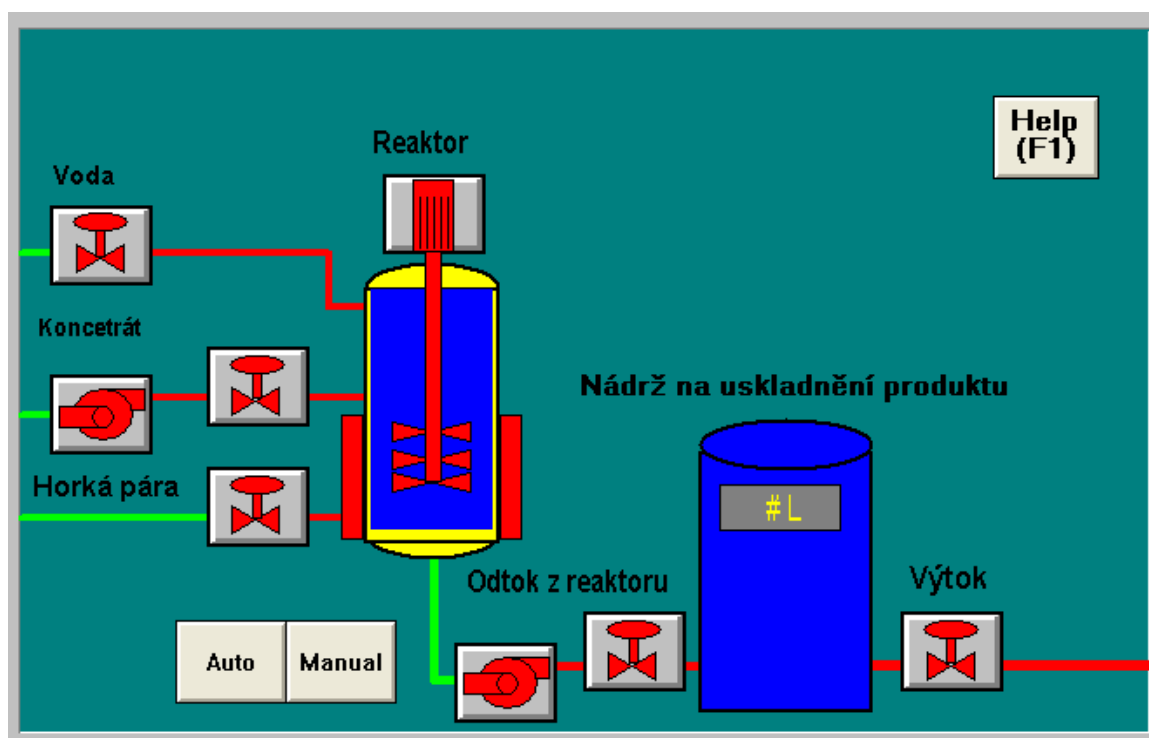
U čerpadla je funkce vizualizovaná podobně jako u ventilu. Funkce míchání je vizualizovaná lopatkami mixéru, které aktivitu znázorňují blikáním.

Plnění nádrže je vizualizováno opět změnou barvy. Na rozdíl od předchozích objektů je však řízeno analogovou proměnnou a ke změně tedy dochází postupně, od spodu nahoru, viz. Obrázek 2.2. Navíc je vhodné použít blikající rám pro signalizaci přetečení nádrže nebo přiřadit číselnou hodnotu ukazující aktuální stav objemu.



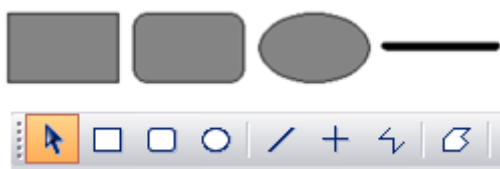
Obrázek 2.2 – Vizualizace některých vybraných zařízení: (a)ventil, (b)čerpadlo, (c)mixér,(d) nádrž

Celkově se vizualizace skládá z posloupnosti: uzavřený ventil se otevře, čerpadlo se spustí, obsah se promíchá, nádrž se plní, viz. Obrázek 2.3.



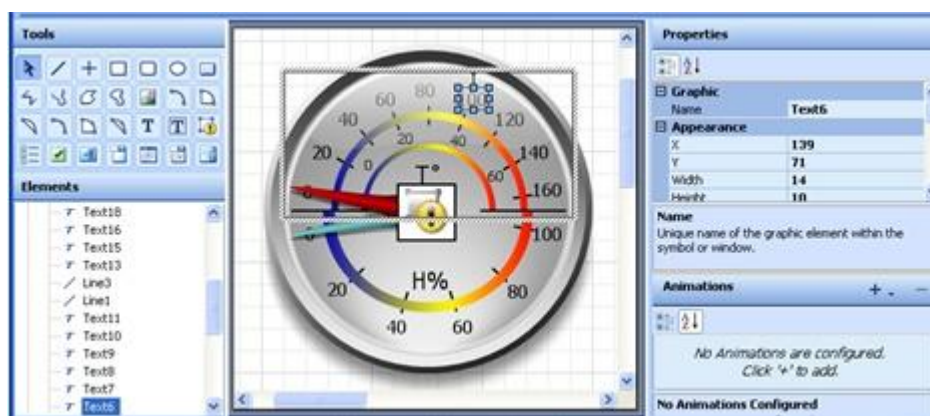
Obrázek 2.3 – Vizualizace procesu s použitím zvolených prvků

Objekty lze například graficky znázornit ze základních geometrických prvků, jejichž vlastnosti lze konfigurovat. Ke konfigurovatelným základním vlastnostem patří velikost, barva, ohraničení, tvar apod.



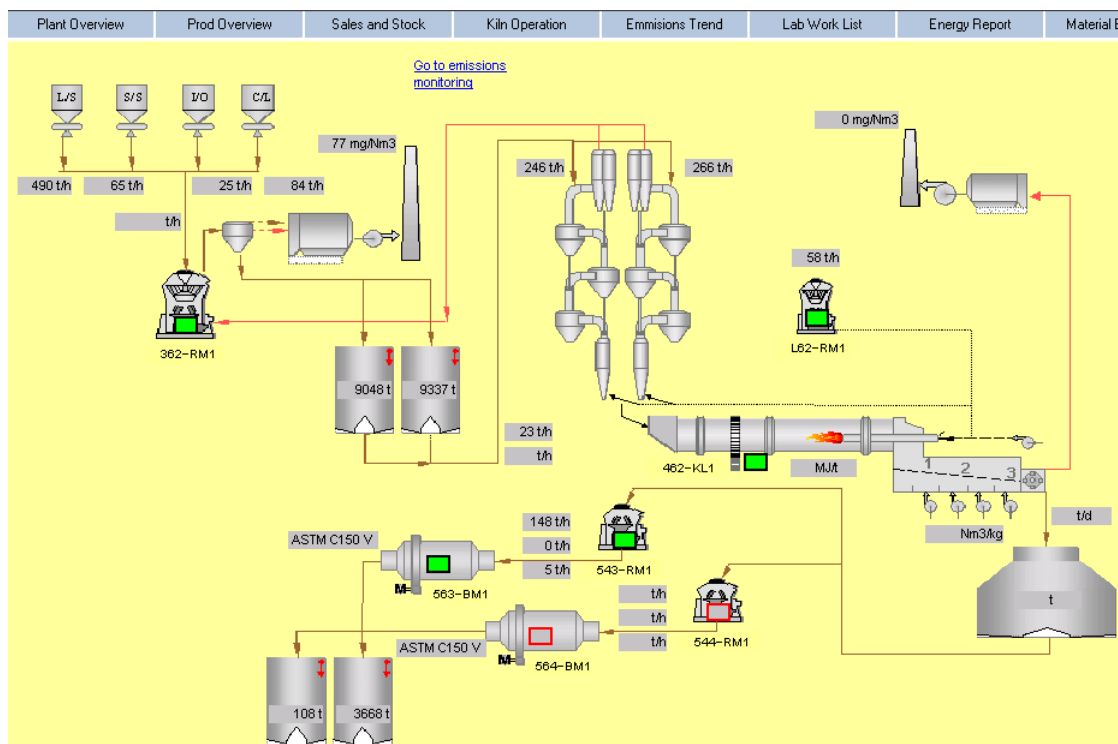
Obrázek 2.4 - Kreslení objektů ze základních geometrických prvků a lišta s odpovídajícími nástroji ve vývojovém prostředí

Ze základních prvků lze pak složením vytvářet složitější objekty, jejichž vzhled závisí na použitých nástrojích pro jejich kreslení. Příkladem složitějších objektů jsou tlačítka, numerické a grafické displeje, panely apod. Pro opakované použití se z těchto objektů sestavují knihovny, příklad ukazuje Obrázek 2.5.



Obrázek 2.5 - Objekt z knihovny předem konfigurovaných objektů a nástroje v liště vývojového prostředí

Používáním takto vyspělých grafických nástrojů se práce na složitějších projektech a komplexech velice zjednodušuje, zrychluje a zpříjemňuje, viz. Obrázek 2.6.



Obrázek 2.6 – Obrazovka vizualizace procesu využívající předem konfigurované objekty



Shrnutí pojmů 2.2.

Značky používané v projektové dokumentaci určují umístění snímačů nebo přístrojů v technologickém procesu. Jejich úkolem je lokalizovat místo přenášených dat, specifikovat chování a přiřazovat funkcionalitu zařízení k měřicím a řídicím přístrojům.

Značky používané v projektové dokumentaci se staly základem pro tvorbu jednoduchých a složených objektů, které na obrazovkách vizualizovaných aplikací reprezentují strojní zařízení a jejich ovladačí a ovládané prvky.

Ve vizualizačních aplikacích jsou k dispozici nástroje ke kreslení objektů, jejichž vzhled na obrazovce vychází ze zavedeného značení. Samotná vizualizace se provádí většinou změnou vzhledu, velikosti nebo polohy.

Objekty lze graficky znázornit ze základních geometrických prvků, jejichž vlastnosti lze konfigurovat. Ke konfigurovatelným základním vlastnostem patří velikost, barva, ohraničení, tvar apod.



Otázky 2.2.

Jaké vlastnosti má objekt reprezentovaný v objektově orientované grafice?

Které jsou základní konfigurovatelné vlastnosti jednoduchých objektů?

Jaký je rozdíl mezi objektem nakresleným a objektem vloženým z knihovny objektů?

3 VIZUALIZAČNÍ PROSTŘEDÍ

Prostředí pro vývoj vizualizačních aplikací je mnoho a liší se některými svými funkcemi. Pro vývojáře je ale rozhodující, zda bude aplikace vznikat na lokálním počítači nebo se vývoj rozdělí mezi tým vývojářů. Následující text zohledňuje práci a způsob vývoje v prostředí Window Maker a koncepci vývoje založenou na objektu propojeném vazbou na veličinu – tzv. *tag based development*, viz. kapitola 2.1.1.



Čas ke studiu: 2 hodiny



Cíl: Po prostudování tohoto odstavce budete umět

- + Popsat charakteristické znaky vývojového a runtime prostředí
- + Používat objekty, objektově orientovanou grafiku
- + Rozumět struktuře návrhu procesních obrazovek

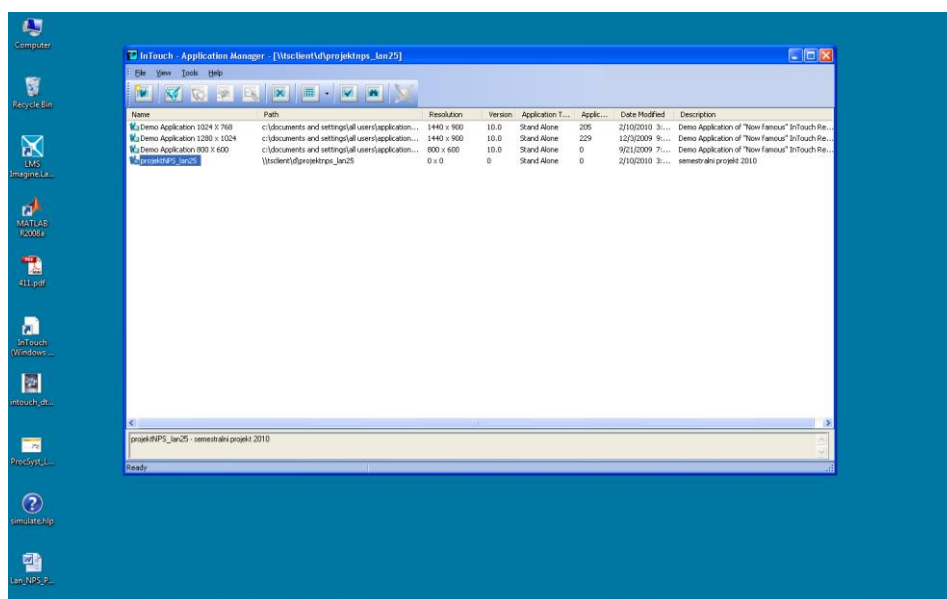


Výklad

3.1 Tvorba aplikací ve vizualizačním prostředí

3.1.1 Manažer aplikací

Manažer aplikací - Application Manager - umožňuje založení nového projektu nebo vyhledání a otevření již existujícího projektu. Úloha aplikačního manažera je při organizaci projektu, jelikož vývojové prostředí neumožňuje editovat více než jeden projekt na jednom lokálním počítači. Projekt může být otevřen do vývojového nebo runtime prostředí.



Obrázek 3.1 - Založení nového projektu



Úlohy k řešení

Řešte následující úlohy.

Příklad 3.1. Otestujte demo aplikaci

Vyhledání existujícího demo projektu – použijte příkazy z menu File/Find Application.

Příklad 3.2. Vytvořte nový projekt pro vizualizační aplikaci

Založte nový projekt s novým názvem, jeho popisem a definovanou cestou pro složku projektu tak, abyste mohli provést zálohu na jiné médium a projekt uschovat do příštího cvičení.

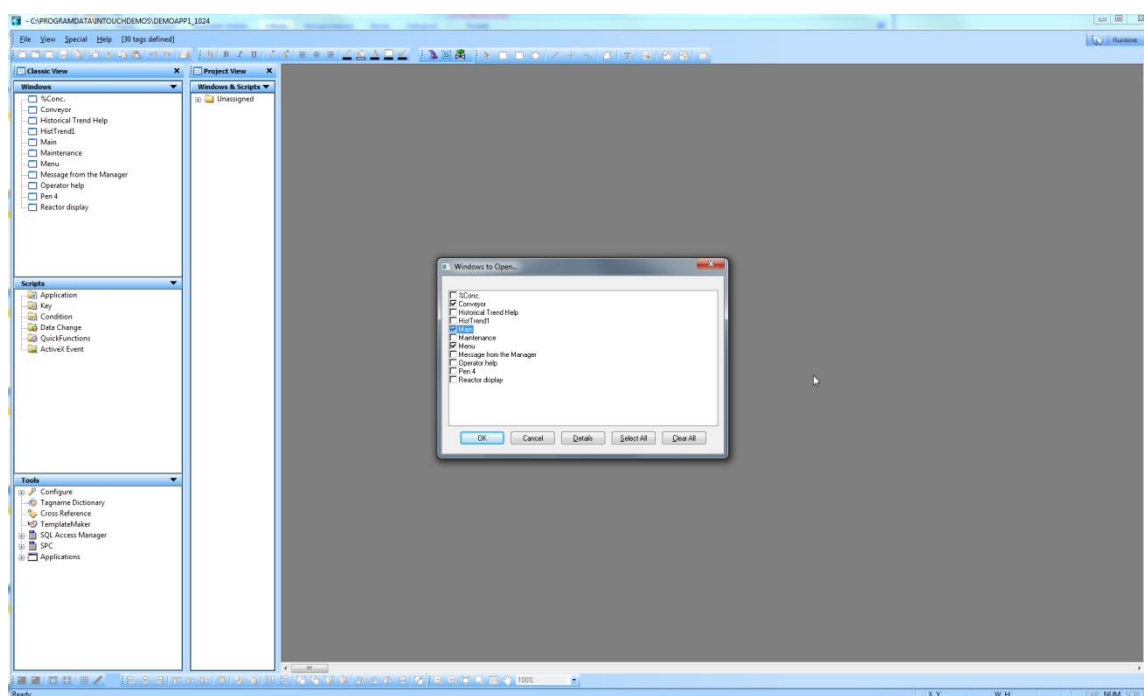


Řešení příkladů

Nový projekt – založíte pomocí průvodce, zadáním cesty, názvu složky a názvu projektu s popisem. Podrobný postup práce na příkladu je uveden v animaci „Práce s Aplikačním managerem“.

3.1.2 Vývojové prostředí aplikací

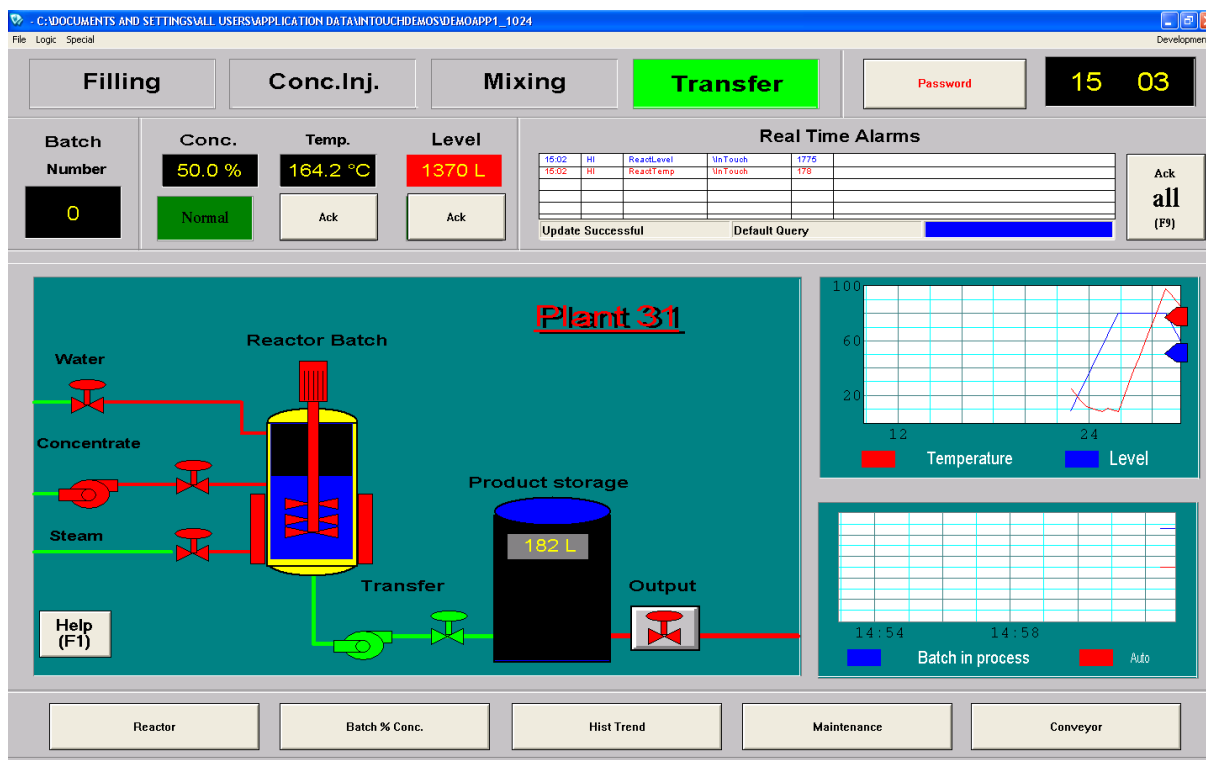
Vývojové prostředí – Window Maker – slouží pro konfiguraci, návrh vzhledu a kreslení objektů, viz *Obrázek 3.2*. Po otevření projektu ve vývojovém prostředí vznikne soubor *appedit.lok* v adresáři projektu, jehož existence brání editaci projektu druhým uživatelem a který po ukončení práce na projektu sám zanikne. V případě kolize systému, nestandardního zavření projektu nebo kopírování složky projektu před ukončením práce s ním je tento soubor zachován a nedovolí projekt znovu opakovaně spustit.



Obrázek 3.2 – Vývojové prostředí projektu

3.1.3 Běhové – runtime - prostředí aplikací

Runtime – Window Viewer – zprostředkuje pohled do technologie, do výroby, do procesu v reálném čase, viz. *Obrázek 3.3*. Mezi jednotlivými pohledy do technologie se uživatel přepíná formou tlačítek v dolní části obrazovky formou nabídky/menu.



Obrázek 3.3 – Běhové – runtime - prostředí projektu



Shrnutí pojmů 3.1.

Manažer aplikací - Application Manager - umožňuje založení nového projektu nebo vyhledání a otevření již existujícího projektu.

Vývojové prostředí – Window Maker – slouží pro konfiguraci, návrh vzhledu a kreslení objektů.

Runtime – Window Viewer – zprostředkuje pohled do technologie, do výroby, do procesu v reálném čase.



Otázky 3.1.

Co je označováno jako runtime?

Jak se liší vývojové prostředí od runtime?

3.2 Návrh obrazovky operátorského pracoviště

Zobrazení na obrazovkách má být jednoduché, bez zbytečných, neužitečných informací. Jinak vzniká riziko, že nedůležitá informace odvede pozornost. Dobré rozhraní umožňuje uživateli soustředit se na řízení procesu, bez odvádění pozornosti svým vzhledem.

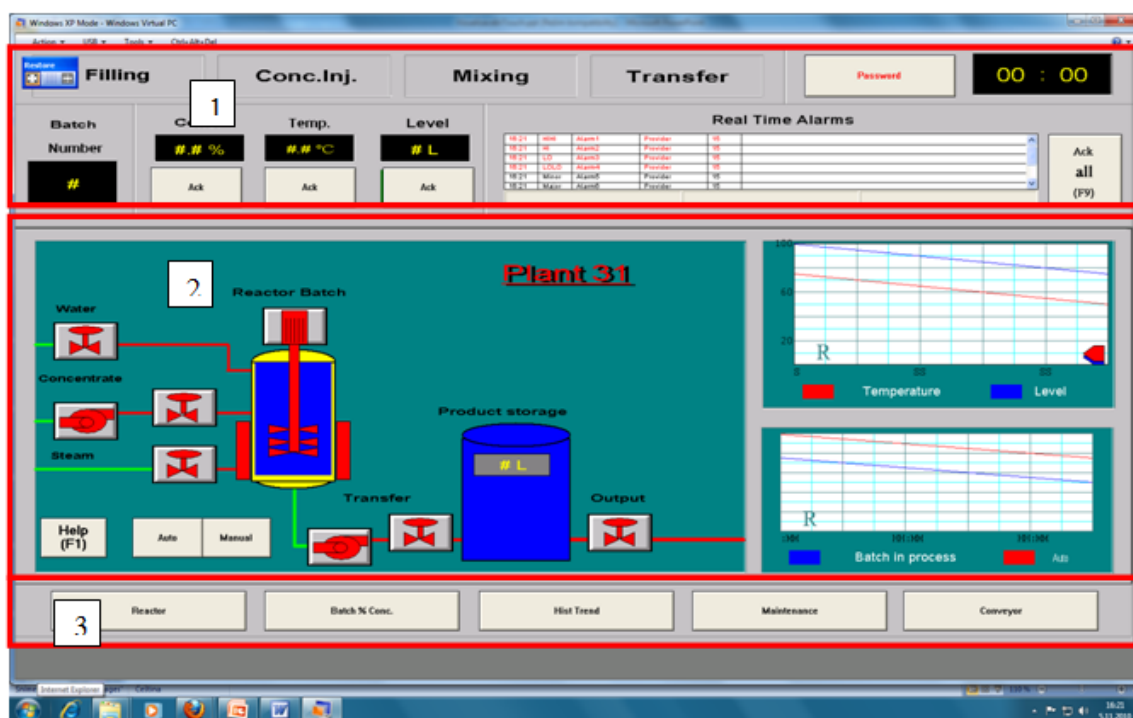
Pro dobré zpracování obrazovky je důležité dodržet tato pravidla:

- nezobrazovat více informací než je třeba,
- nepřesáhnout pět výrazně rozlišitelných položek do stejné abstrakční úrovně,
- informace musí být logicky organizována, pokud možno samo vysvětlující,
- stejné použití symbolů a barev v různých úrovních a v různých obrazovkách.

Při vývoji je nutné kromě standardu pro volbu barev zachovávat také princip návrhu pro umístění obrazovek. Obrazovka má mít jednoduchou koncepci. Údaje je nutné rozdělit do menších skupin, kde každá skupina je prezentována vlastním oknem. Důraz se klade na nejdůležitější informace. Například uživatel musí být vždy informován o závažných poruchách. V případech, kdy je nutné získat další důležitou informaci, se musí přepnout do jiného okna.

Hlavní obrazovka (1) – poskytuje koncovému uživateli souhrnné informace o procesu, a ty musí zůstat na obrazovce i při přepínání do jiných pohledů. Je to tedy přehledová (nepřepisovatelná) část, ke kterým patří procesní hlášení, hlášení systému, přihlášení se do systému apod.

Pohled do různých částí technologie, hlášení, trendy, informativní texty, apod. (2) jsou zprostředkovány přepínáním do různých obrazovek zpravidla formou nabídky/menu, virtuální klávesnicí, ovládacími vstupy, panely (3), viz. také Obrázek 3.4.



Obrázek 3.4 – Princip uspořádání oken v aplikaci pro monitorování procesu

Okno lze navrhovat typu:

- replace – při překrytí zavře okno na pozadí
- overlay – překrývající okna zůstávají otevřená
- popup – výhradně pro výstrahu, musí být zavřeno manuálně

V našem projektu navrhujeme sled obrazovek takto:

- info – zobrazuje důležité indikátory, čas
- objekty, animace - střední plocha pro náhledy do různých částí technologií
- menu – pro přepínání mezi pohledy.



Úlohy k řešení

Řešte následující úlohu.

Příklad 3.3. *Vytvořte okno s tlačítky pro přepínání do různých obrazovek.*



Řešení příkladů

Podrobný postup práce na příkladu je uveden v animaci „Práce s okny v prostředí InTouch“.



Shrnutí pojmů 3.2.

Zobrazení na obrazovkách má být jednoduché, bez zbytečných, neužitečných informací, aby nevznikalo riziko, že nedůležitá informace odvede pozornost. Dobré rozhraní umožňuje uživateli soustředit se na řízení procesu, bez odvádění pozornosti svým vzhledem.

Oblast pro hlavní okno obrazovky soustřeďuje nejdůležitější data v horní části obrazovky.

Oblast určená pro všechna ostatní okna zpřístupňující pohled do technologie je uprostřed obrazovky.

Nabídka pro tlačítka přepínající obsluhu do různých pohledů je k dispozici ve spodní části obrazovky.



Otázky 3.2.

Jak je rozdělena obrazovka operátorského pracoviště?

Jaké typy oken lze konfigurovat?

Jak se liší funkčnost okna podle toho, jakého je typu?

3.3 Kreslení objektů

Pokud je výměna informací mezi uživatelem a řízeným systémem jednoduchá a efektivní, nejen to zpřijemňuje práci, ale také zmenšuje chyby a omezuje vznik škod. Výstupní údaje musí být lehce srozumitelné i ve složitých situacích.

Příjem informace je v procesním systému nejčastěji akustický nebo vizuální. Informace shromažďované smyslovými orgány jsou přenášeny do krátkodobé paměti. Krátkodobá paměť je rychlá na rozpomenutí i na zapomenutí, ale informace v ní je k dispozici okamžitě. Dlouhodobá paměť má téměř nekonečný potenciál, ale zapamatování a rozpomenutí vyžaduje delší dobu. Krátkodobá paměť může trvat sekundy, dlouhodobá celý život.

Pro vizualizaci by měly být vybrány jednoduché modely, které souvisejí s všedními zkušenostmi. Například hodnota 54 sama o sobě nic neznamená. Pokud ji však nazveme teplotou a přirovnáme k maximální dovolené teplotě 67°C, získáme o mnoho výstižnější informaci. Je však rozdíl 13°C přijatelný?

Uživatel získává informace třemi způsoby, je to:

- vnímání,
- kódování,
- organizace.

Schopnost **vnímání** má fyzickou podobu. Pro obrazovku terminálu je důležitý jas, kontrast barev, velikost symbolů. Není vhodný takový přístroj, na kterém uživatel musí vynakládat velké úsilí, aby získal nějaké informace.

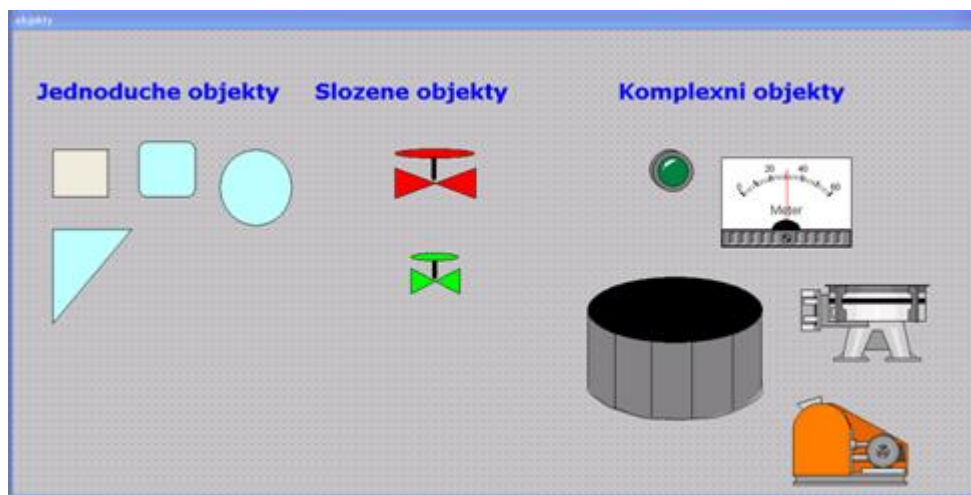
Kódování je způsob přenášení informace pomocí grafických objektů. Správné zakódování zprávy napomáhá přenášet velké množství informací. Je možné použít stejné symboly s rozdílnými barvami nebo polohami pro různé stavy.

3.3.1 Jednoduché a složené objekty

Objekty kreslíme ve vývojovém prostředí pomocí nástrojů, viz. také *Obrázek 2.4* a *Obrázek 2.5* v předchozí kapitole.

Jednoduché objekty mají definované základní vlastnosti, jako tvar, velikost, barva apod. Složené objekty vzniknou seskupením jednoduchých objektů stejným způsobem, na jaký jsme zvyklí z jiných kreslicích prostředí. Seskupený objekt lze nadefinovat jako:

- Symbol – u něho je možná editace základních vlastností celého seskupení,
- Cell – drží základní vlastnosti seskupení bez možnosti editace



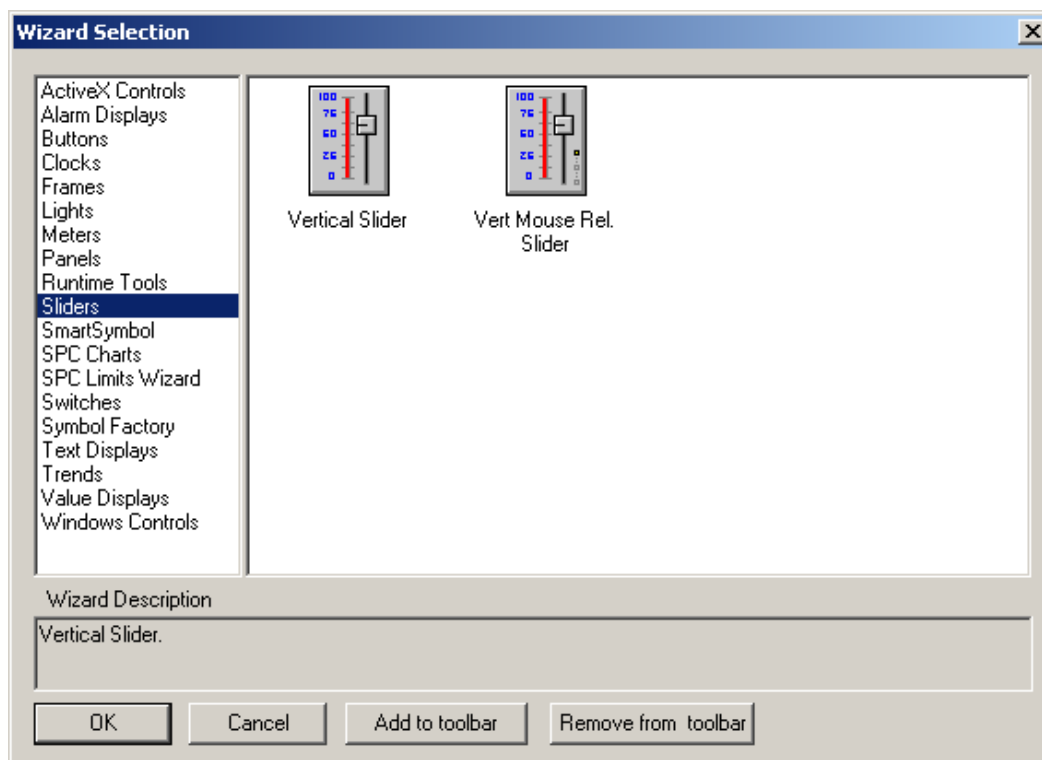
Obrázek 3.5 – Typy objektů

3.3.2 Komplexní a předem definované objekty

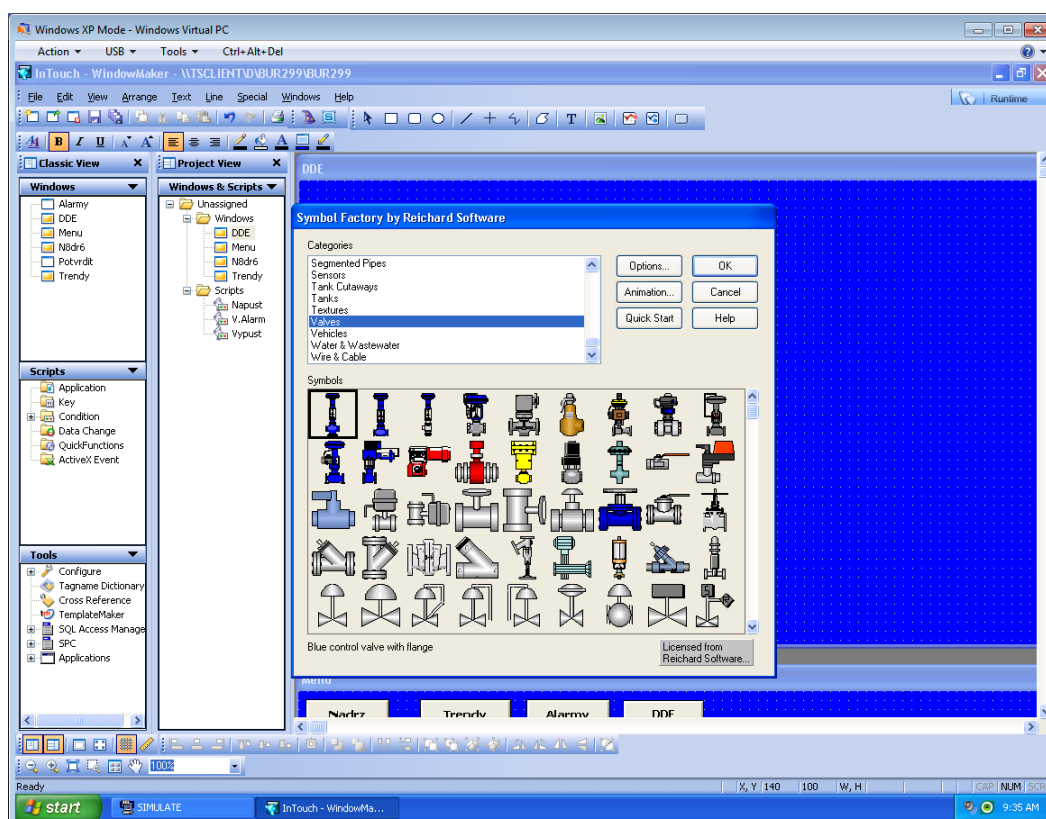
Komplexní objekty představují variantu předem definovaných objektů s možností jejich vložení z dostupných knihoven, příkladem jsou:

- Wizards
- Symbol Factory

K nejpoužívanějším objektům patří Buttons (tlačítka), Clocks (hodiny), Lights (kontrolní světla), Sliders (posuvníky), Switches (vypínače) a Trends (ukládání a zobrazování hodnot v grafu).



Obrázek 3.6 – Volba objektů z knihovny Wizards



Obrázek 3.7 – Volba objektů z knihovny Symbol Factory

Volba barev objektů je doporučena dodržováním standardů, viz. např. IEC73. Výběr barvy by měl tedy být v souladu s platnými normami a směrnicemi. Barvy použité na objekty a obrazovky pro řízení procesu lze rozdělit podle cíle použití:

- barvy pro podklad,
- barvy pro kódování informace,
- barvy pro vyjádření stejného významu.

Barvy pro podklad, pozadí obrazovky se musí volit tak, aby byla dosažena maximální rozlišitelnost všech barev v popředí. Použité barvy musí poskytovat vzhledem k pozadí vysoký barevný kontrast. Podklad nemá být pestrý, ale achromatický.

Barvy pro kódování mají za úkol každému stavu, kterého objekt může nabývat, přiřadit jednu barvu.

Barvy jsou pro kódování informace důležité. Čtyři až pět barev je možné vnímat bez většího úsilí. Barvami je třeba rozlišovat tyto diskrétní stavy:

- poruchy-stavy částečné nebo úplně omezené funkce zařízení,
- stavy, při kterých zařízení fungují podle příkazů (např. motor je zapnutý, klapka vypnutá).

Doporučené dvojice barev a význam barev pro kódování uvádí tabulky Tab. 3.1 a Tab. 3.2.

Tab. 3.1 – Tabulka pro použití dvojice barev pro objekty, texty, podklady

Barva objektu	Barva podkladu						
	černá	bílá	purpurová	modrá	zelenomodrá	zelená	žlutá
Černá		velmi dobrá	čitelné	špatná	velmi dobrá	dobrá	velmi dobrá
Bílá	dobrá		dobrá	dobrá	nečitelné	nečitelné	nečitelné
Purpurová	čitelné	dobrá		špatná	čitelné	čitelné	velmi dobrá
Modrá	špatná	velmi dobrá	špatná		dobrá	dobrá	velmi dobrá
Zelenomodrá	velmi dobrá	špatná	čitelné	velmi dobrá		nečitelné	špatná
Zelená	velmi dobrá	špatná	čitelné	dobrá	nečitelné		špatná
Žlutá	velmi dobrá	nečitelné	čitelné	velmi dobrá	nečitelné	nečitelné	
Červená	čitelné	velmi dobrá	špatná	špatná	dobrá	dobrá	dobrá

Tab. 3.2 – Význam kódování barev podle IEC 73

Barva	Stav	Stav podle IEC 73	Požadavky na operátora
Červená	alarm	naléhavý případ ¹	nevyhnutelná okamžitá reakce
Žlutá	varování	odchylka od žádaného stavu	věnovat pozornost a dále sledovat
Zelenožlutá ²	výstražné varování	odchylka od želaného stavu	věnovat pozornost
Zelená	normální	normální	žádná
Modrá	výzva k zásahu		potřebný zásah
bílá, šedá, černá	bez speciálního určení (volné použití)		

¹V některých normách se neuvažuje pojem nebezpečí, protože je nutný zvláštní blokovací systém.

²Pokud je nutné vizualizovat výstražné varování jako stav, musí se barva odlišovat od žluté pro varování.

Diskrétní stavy nejlépe vyjadřují přirozené barvy. Zelená jako stav bezpečnosti, souhlasu a správnosti, červená vyjadřuje alarmové stavy, zákazy, nebezpečí. Žlutá je vhodná pro označení přítomnosti menšího problému.

Pro analogové stavy je možno použít barvy následovně:

- skutečné hodnoty, např. naměřené hodnoty, nebo regulační odchylky – světle zelená,
- vstup – bílá,
- žádaná hodnota – modrá,
- nastavování hodnoty – okrová.



Úlohy k řešení

Příklad 3.4. *Nakreslete jednoduchý objekt tak, aby splňoval doporučení pro kódování informace pomocí grafických objektů.*

Příklad 3.5. *Nakreslete složený objekt seskupením jednoduchých objektů.*

Příklad 3.6. *Vložte vybrané objekty z knihovny dostupných objektů a konfigurujte je jako objekt ovládaný a ovladací.*



Řešení příkladů

Nový objekt nakreslíte nebo vložíte z knihovny pomocí nástrojů dostupných v liště vývojového prostředí metodou *drag and drop*. Podrobný postup práce na příkladech je uveden v animaci „Kreslení objektů v prostředí InTouch Maker“.



Shrnutí pojmů 3.3.

Kódování je způsob přenášení informace pomocí grafických objektů.

Jednoduché objekty mají definované základní vlastnosti, jako tvar, velikost, barva apod. Složené objekty vzniknou seskupením jednoduchých objektů stejným způsobem, na jaký jsme zvyklí z jiných kreslicích prostředí. Seskupený objekt lze nadefinovat jako symbol nebo cell.

Volba barev objektů je doporučena dodržováním standardů, viz. např. IEC73.



Otázky 3.3.

Jaké jsou základní vlastnosti jednoduchých konfigurovaných objektů?

Jak lze vytvářet složitější objekty?

4 ANIMACE OBJEKTŮ A VAZBY NA PROMĚNNÉ VELIČINY

V této části je prezentován způsob, jak oživit objekty napojením na monitorované veličiny. Opět je dodržena koncepce viz. kapitola 2.1.1.



Čas ke studiu: 6 hodin



Cíl: Po prostudování tohoto odstavce budete umět

- + Popsat datové typy proměnných veličin
- + Používat animační vazby objektů



Výklad

4.1 Sběr dat

Součástí systému SCADA/HMI je sběr dat - *data acquisition*. Data jsou ukládána do databáze a dále výpočetně zpracovávána – *data processing*. Umístění databáze s uloženými daty závisí na architektuře celého procesního systému. Zpracování dat je dáno aplikační logikou software aplikací procesního systému. Nejjednodušší architektura procesního systému je dvou-vrstvá tvořená klientem a serverem, mezi kterými jsou data komunikována. Klient je zařízení nebo počítač, který žádá data. Server je zařízení nebo počítač, který data poskytuje. Více-vrstvé architektury sbírají a zpracovávají data ze složitějších systémů. Kromě sběru dat ze zařízení poskytovaných I/O servery a ukládaných do databáze na databázových serverech vyžadují složitější systémy zpracovávání dat na aplikačních serverech. Pokud je klientem vzdálené zařízení nebo počítač, který pro vyžádání dat potřebuje pouze webový prohlížeč, jedná se o tenkého klienta. Takový klient pak může žádat data z procesů a do vícevrstvé architektury klient server pak bude patřit taky webový server.

Sběr dat může probíhat automatickým nebo manuálním způsobem. Automaticky mohou být čteny údaje z měřených veličin, zapisovány odezvy funkcí systému nebo hlášení z údržby, apod. Ručně lze vkládat data zjištěná analýzami v laboratořích, odpozorované obsluhou apod.

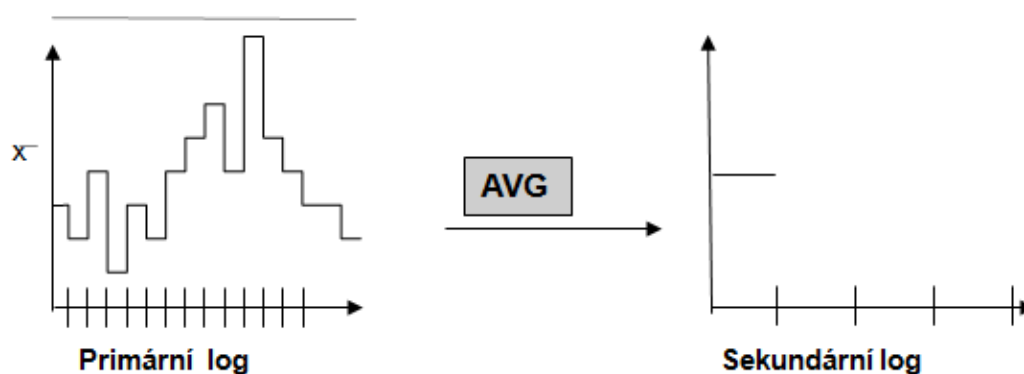
4.1.1 Vzorkování a logování hodnot

Odečítání hodnot sledované veličiny v procesu v daném časovém intervalu nazýváme vzorkování. Čtená hodnota sledované veličiny je pro sběr dat označována jako signál. Uložená hodnota v databázi ze čteného signálu je ve většině případů označována jako log. Pokud není vstupem logu signál, může se jednat o počítaný log, jehož vstupem je jiný log nebo může být vstupem logu také ručně zadaná hodnota operátorem. Na jeden signál je možné definovat více logů, např. průměrnou, maximální nebo minimální hodnotu

vzorkovaného signálu v určitém časovém intervalu, atd. Primární log je pak uložena hodnota ze čteného signálu bez dalšího zpracování, například výpočetních úprav.

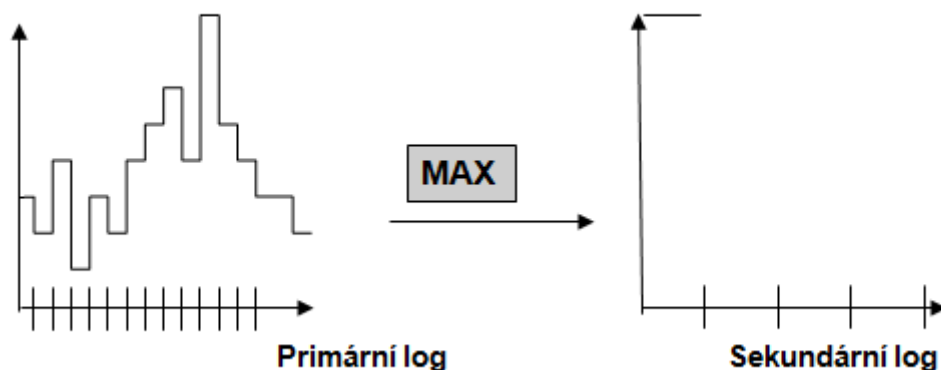
Obrázek 4.1 – Konfigurace primárního logu v prostředí procesního systému

Konsolidovaný log je vypočtená hodnota pro uživatelem definovaný časový interval, například je-li z minutových logů zapsána jedna hodnota reprezentující čtvrt nebo půl hodiny (15M, 30M), hodinu (HRS), směnu (SHT), den (DAY), týden (WEK), měsíc (MTH), rok YER). Zapsaná hodnota konsolidovaného logu do databáze je sekundární log. Konsolidační funkce je funkce pro výpočet konsolidovaného logu (matematická funkce, např. průměr, maximum, kvadratická odchylka apod.) ze sledované veličiny v daném intervalu. Obrázek 4.2. zobrazuje princip použití AVG funkce, která ukládá průměr z hodnot do příští hodnoty logu.



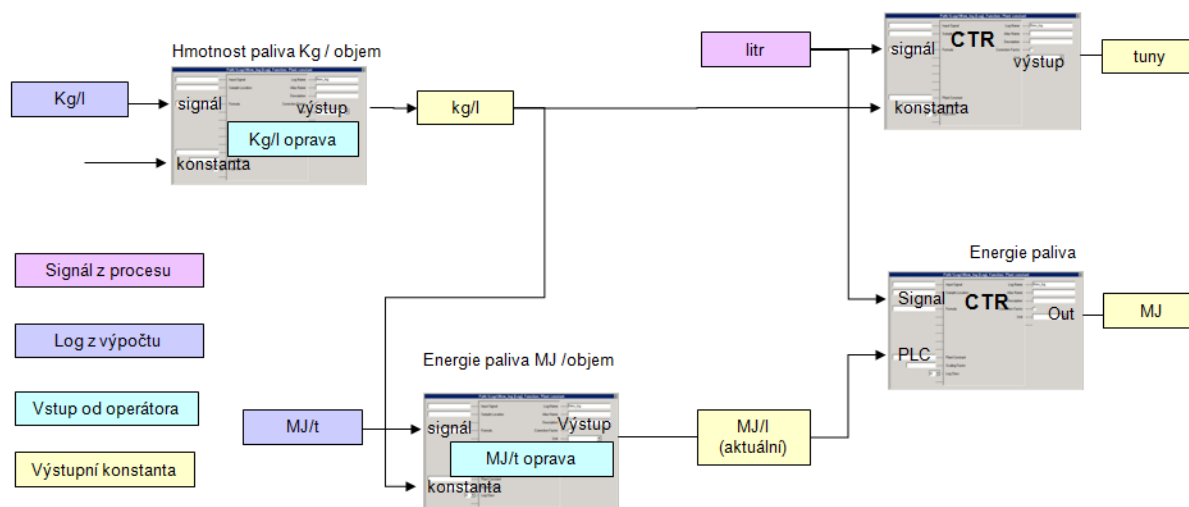
Obrázek 4.2 – Sekundární log konsolidovaný funkcí výpočtu průměrné hodnoty (average)

Podobně pak funkce MAX ukládá maximální hodnotu z hodnot do příští hodnoty logu, viz. Obrázek 4.3.



Obrázek 4.3 – Sekundární log konsolidovaný funkcí výpočtu maximální hodnoty (maximum)

Vypočtená hodnota množství a energie paliva, ze zadané hodnoty hmotnosti paliva a energetické hodnoty vložené manuálně operátorem, tvoří počítaný log viz. Obrázek 4.4.



Obrázek 4.4 – Výpočet logů postupnou konsolidací signálu z procesu, z vypočteného logu a vstupního údaje od operátora

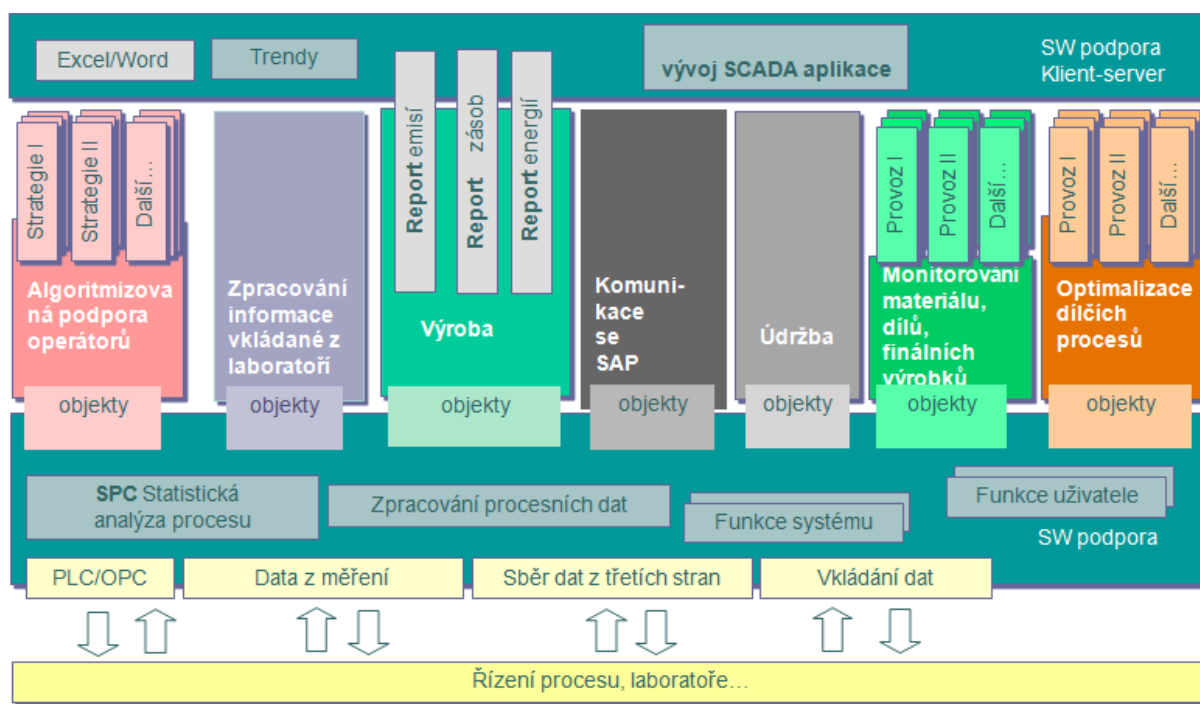
4.1.2 Interní a externí zdroje sběru dat

Interní data prostředí SCADA/HMI zůstávají lokálně v tomto softwarovém prostředí a slouží jako algoritmizovaná podpora operátorů nebo jsou to data z funkcí systému nebo funkcí uživatelů. Jedná se o data v lokálních proměnných, která nejsou komunikována jinde, než mezi softwarovým prostředím a na obrazovku operátora.

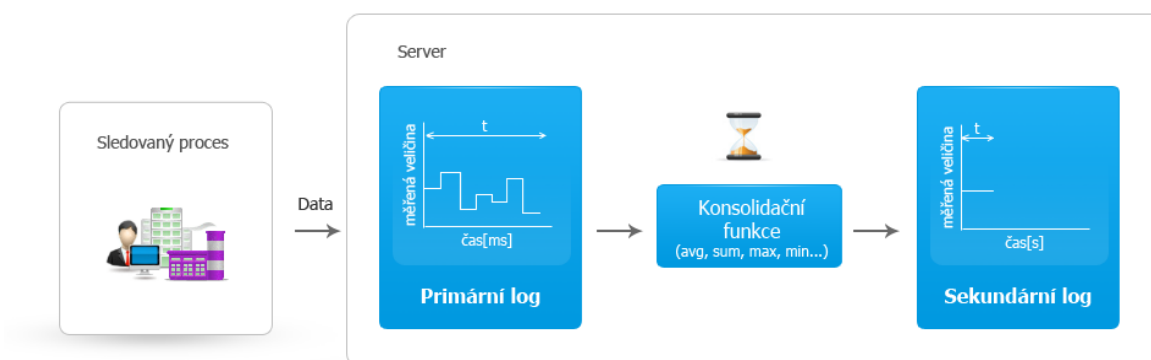
Data, která pochází ze zpracování výrobních procesů, z měření, z programovatelných (PLC) automatů a/nebo třetích stran (i nadřazených informačních systémů, se kterými jsou

integrovány systémy SCADA) jsou prostřednictvím proměnných veličin komunikována v celé architektuře systému SCADA/HMI.

Obrázek 4.5 znázorňuje možné zdroje dat pocházející z řízení procesů v architektuře systému typu klient-server do jednoho schématu. Různorodost dat je dána parametry dat, jako jsou jejich datové typy, jednotky, rozsahy, možnost komunikace s externím prostředím atd.



Obrázek 4.5 – Schéma znázorňující zdroje sběru dat během řízení procesů



Obrázek 4.6 – Schéma grafického zobrazení dat z klasických datových zdrojů



Shrnutí pojmů 4.1.

Data jsou získávána sběrem (akvizicí - *data acquisition*), kdy je sledován určitý proces, ať už výrobní, nebo jsou získávána data od zaměstnanců ve formě např. výsledků z hodnotících dotazníků atd. Data jsou ukládána do databáze a dále výpočetně zpracovávána – *data processing*.

Nejjednodušší architektura procesního systému je tvořená klientem a serverem, mezi kterými jsou data komunikována. Klient je zařízení nebo počítač, který žádá data. Server je zařízení nebo počítač, který data poskytuje.

Odečítání hodnot sledované veličiny v procesu v daném časovém intervalu nazýváme vzorkování. Čtená hodnota sledované veličiny je pro sběr dat označována jako signál. Uložená hodnota v databázi ze čteného signálu je označována jako log. Data jsou uložena do klasické databáze na serveru ve formě primárního logu. Primární log je na serveru zpracován podle požadavků pomocí konsolidačních funkcí a uložen do sekundárního logu, se kterým se pak pracuje dále. Konsolidační funkce je funkce pro výpočet konsolidovaného logu (matematická funkce, např. průměr, maximum, kvadratická odchylka apod.) ze sledované veličiny v daném intervalu. Konsolidovaný log je tedy vypočtená hodnota pro uživatelem definovaný časový interval. Zapsaná hodnota konsolidovaného logu do databáze je sekundární log.

Různorodost dat je dána parametry dat, jako jsou jejich datové typy, jednotky, rozsahy, možnost komunikace s externím prostředím atd.



Otázky 4.1.

Jak probíhá komunikace v architektuře klient-server?

Co je to klient, a jak je definován server?

Co je to vzorkování dat a vzorkovací perioda?

Jaký je rozdíl mezi signálem a logem?

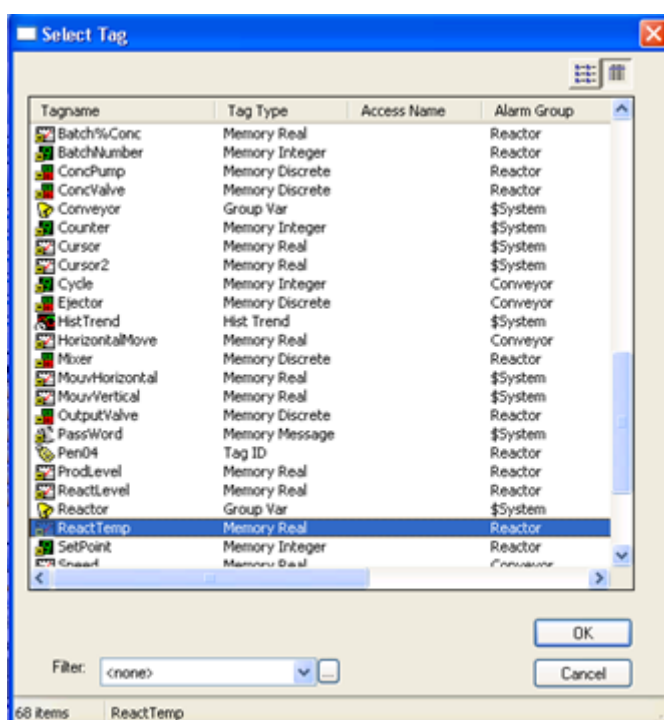
Co pro log znamená konsolidace?

4.2 Datové typy proměnných

Při běhu procesní aplikace obsahuje databáze proměnných veličin vždy aktuální hodnotu všech položek uložených v databázi. Aby bylo možné tuto databázi vytvořit při běhu programu, vyžaduje program informace o všech vytvářených proměnných. Každé proměnné musí být přiřazen název a typ. Vývojář definuje novou proměnnou buď pomocí dialogového okna v knihovně proměnných zadáním názvu proměnné a jejího typu nebo během vývoje a návrhu samotného objektu, viz. Kapitola 0. v souladu s *tag-based* koncepcí vývoje, viz. také Kapitola 2.1.1.

4.2.1 Volba typu veličin

Databáze proměnných veličin, viz. *Obrázek 4.7*, podporuje mechanismus používaný pro zadávání těchto informací. Databáze je rozdělena podle typů veličin. Může se jednat o proměnné veličiny systémové (\$AccessLevel, \$InactivityTimeout, \$Operator) či speciálně určené (HistTrend), které slouží funkcím systému, nebo o proměnné definované uživatelem jako lokální veličiny typu memory nebo veličiny komunikující s externím prostředím typu I/O konvertované pro interní použití.



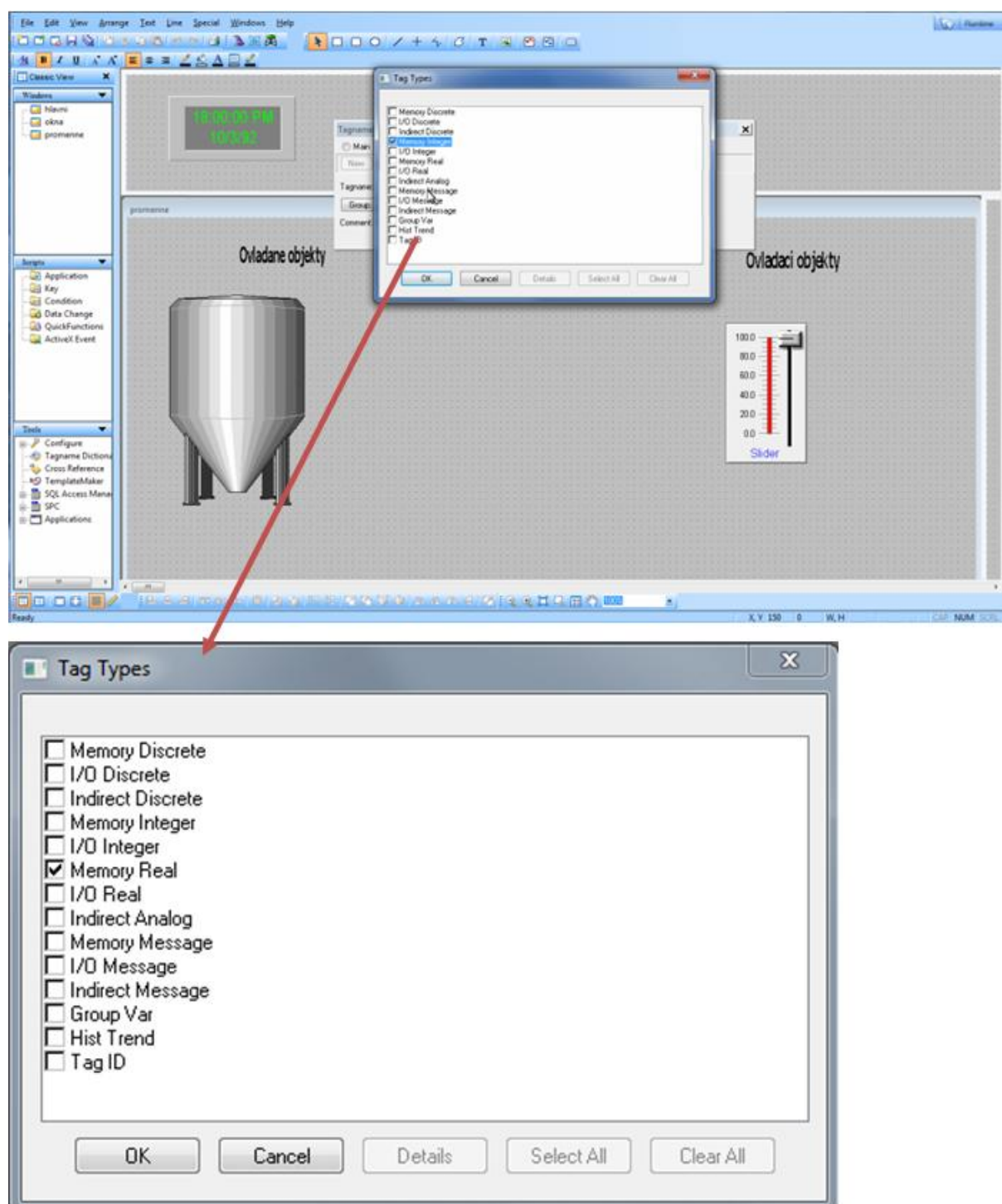
Obrázek 4.7 - Databáze – knihovna proměnných

4.2.2 Uživatelem definované veličiny

Parametr veličiny pro datové typy se týká definice, zda se jedná o veličinu:

- analogovou (typu integer, tedy celočíselnou nebo real s plovoucí desetinnou tečkou),
- binární (discrete nabývající hodnot buď 0 nebo 1),
- textovou, která může vyjadřovat zprávu, hlášení, jméno (message).

Volbu typu ukazuje Obrázek 4.8.



Obrázek 4.8 – Definice datového typu proměnné

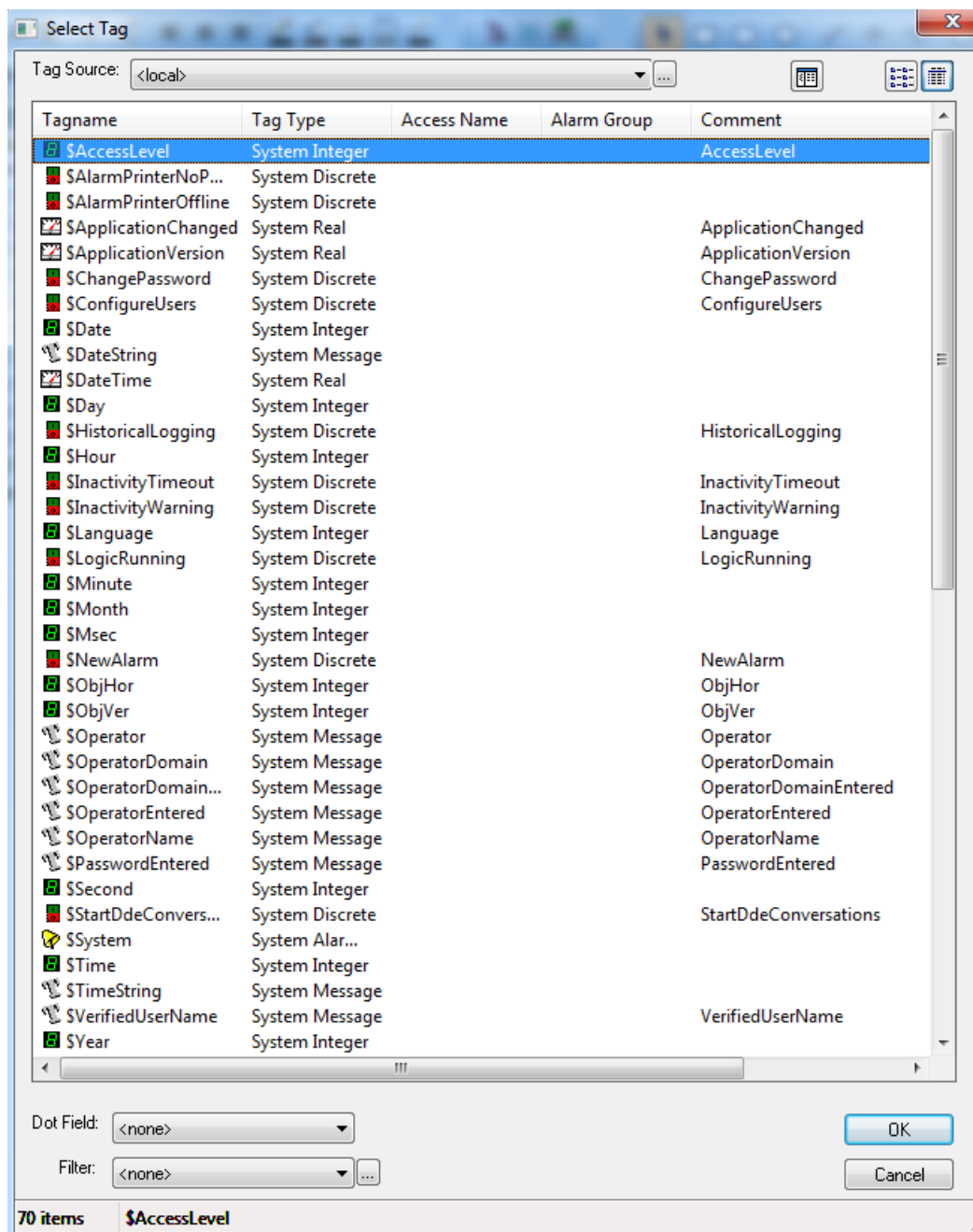
4.2.3 Systémové veličiny

Systémové proměnné bývají označeny \$_____system a patří mezi ně:

- \$Operator (ukládá jméno uživatele, proto je typu message),
- \$AccessLevel (ukládá číslem vyjádřené přístupové úrovně–práva, proto typu integer),

- \$InactivityTimeout (pro kontrolu aktivity nastavené na určitý časový interval, proto je typu discrete).

Uživatelé konfigurované proměnné se přidávají do databáze až za systémové proměnné (podle abecedy) během konfigurace objektů a jejich počet limituje funkci demo režimu InTouch aplikace (max. 31 uživatelem definovaných proměnných).



Obrázek 4.9 – Volba typu systémové proměnné



Shrnutí pojmů 4.2.

Databáze proměnných veličin obsahuje informace o všech vytvářených proměnných a aktuální hodnotu všech položek uložených v databázi aplikace. Každé proměnné je přiřazen název a typ, a to buď během definice nové proměnné pomocí dialogového okna v knihovně proměnných, nebo během vývoje a návrhu samotného objektu.

Databáze proměnných je rozdělena podle typů veličin na systémové či speciálně určené nebo definované uživatelem jako lokální veličiny typu memory nebo veličiny komunikující s externím prostředím typu I/O.

Parametr veličiny pro datové typy se týká toho, zda se jedná o veličinu analogovou, logickou nebo textovou.



Otázky 4.2.

Co je to databáze proměnných?

Jaké typy veličin znáte?

Jaký je rozdíl mezi analogovou, binární a textovou veličinou?

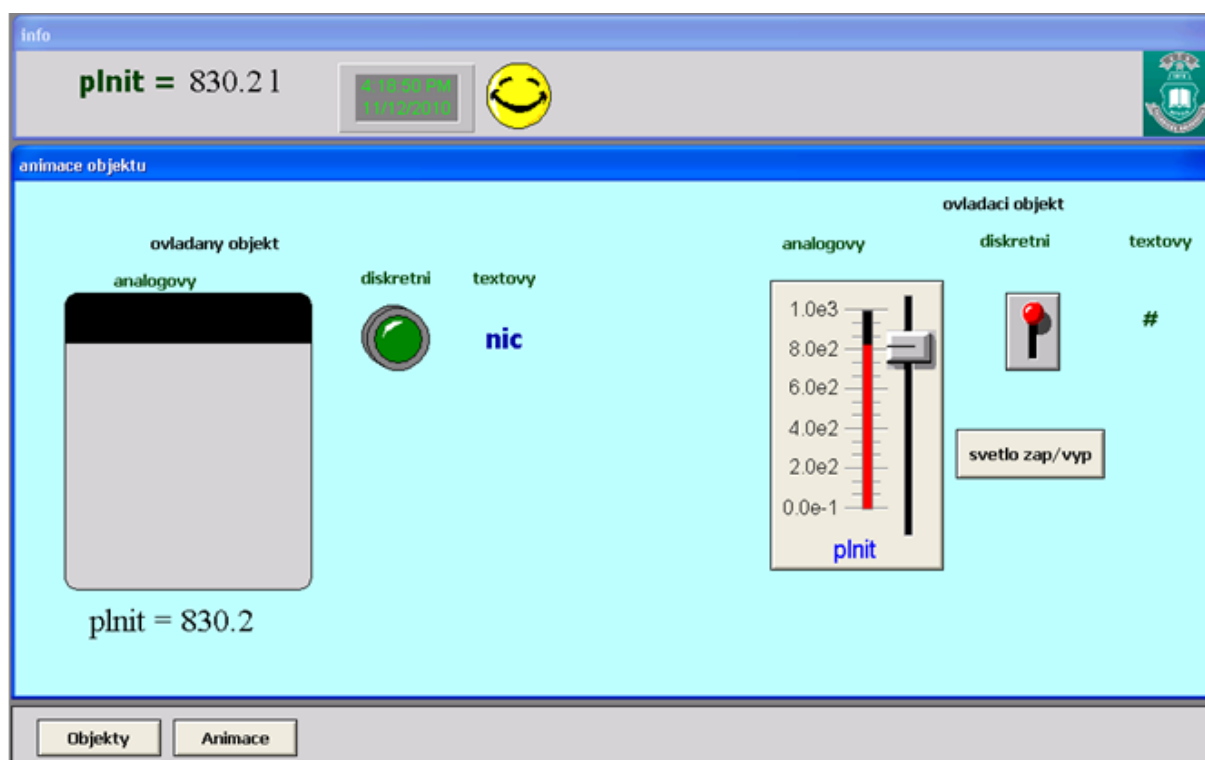
4.3 Animace objektů

Animovat lze objekty připojením na animační vazby, které lze konfigurovat na dvojklik objektu.

4.3.1 Konfigurace animovaného objektu

Při konfiguraci se v konfiguračním okně objektu nabídnou možnosti – vazby na ovládané nebo ovládací objekty, viz. *Obrázek 4.10*, podle typu proměnné, např.:

- pro analogový typ proměnné (objekt typu obdélník):
 - fill color/analog
 - percent fill/vertical
 - value display/analog
- u binární proměnné, (objekt z knihovny předem definovaných objektů typů light):
 - fill color/discrete (on/off)
- u textové proměnné (objekt typu text):
 - value display/string



Obrázek 4.10 - Animace objektů analogovou, diskretní a textovou veličinou

4.3.2 Skript pro animace objektů

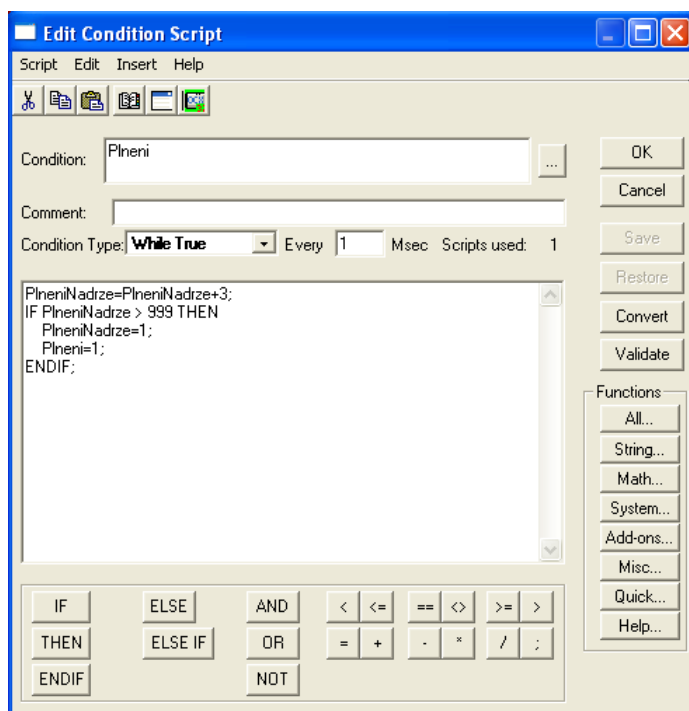
Animovat-rozhýbat objekty je také možné využitím skriptu napsaného do editoru skriptu. Vykonání skriptu na daném objektu může být nadefinováno různými způsoby, např.:

- při splnění určitých procesních podmínek,
- při změnách datových údajů,

- při událostech svázaných s aplikací nebo jednotlivými okny aplikace,
- při stisku kláves, událostech objektu apod.

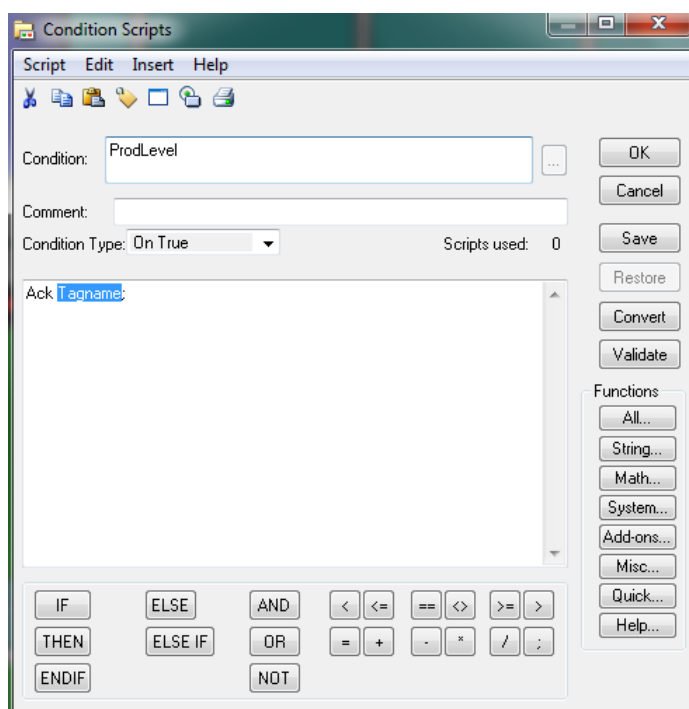
Skriptový editor umožňuje pracovat i s funkcemi, které si vývojář nadefinuje pouze jednou, uloží je do knihovny funkcí a potom je může opakovaně volat jako uživatelské funkce. Tím se zjednodušuje nejen vlastní vytváření aplikací, ale zkracuje i čas potřebný pro jejich rozšiřování.

Při vytváření skriptů lze prostřednictvím tlačítek v editoru pohodlně vybírat často používané výrazy jako „větší než“, „menší než“, „if-then-else“ aj., viz. *Obrázek 4.11*.



Obrázek 4.11 – Editor podmínkového skriptu

Pokročilé funkčnosti jako matematické funkce, operace s řetězci, potvrzení parametru operátorem apod. jsou přístupné pomocí průvodce, který vyzývá uživatele pro zadání potřebných parametrů a zajišťuje správnou syntaxi pro bezchybnou funkčnost, viz. *Obrázek 4.12*.



Obrázek 4.12 – Editor podmínkového skriptu při potvrzení alarmu funkcí Ack s nápovědou pro zadání názvu proměnné



Úlohy k řešení

Řešte následující úlohy.

Příklad 4.1. Vytvořte animaci na objekt s analogovou veličinou.

Příklad 4.2. Vytvořte animaci na objekt s diskrétní veličinou.

Příklad 4.3. Vytvořte animaci na objekt s textovou veličinou.



Řešení příkladů

Podrobný postup práce na příkladu je uveden v animaci „Proměnné a animace objektů“.



Shrnutí pojmů 4.3.

Animovat lze objekty připojením na animační vazby, které lze konfigurovat na dvojklik objektu, nebo využitím editorů skriptu. Vykonání skriptu na daném objektu může být nadefinováno při splnění určitých procesních podmínek, změnách datových údajů, událostech svázaných s aplikací nebo jednotlivými okny aplikace, stisku kláves, událostech objektu apod.



Otázky 4.3.

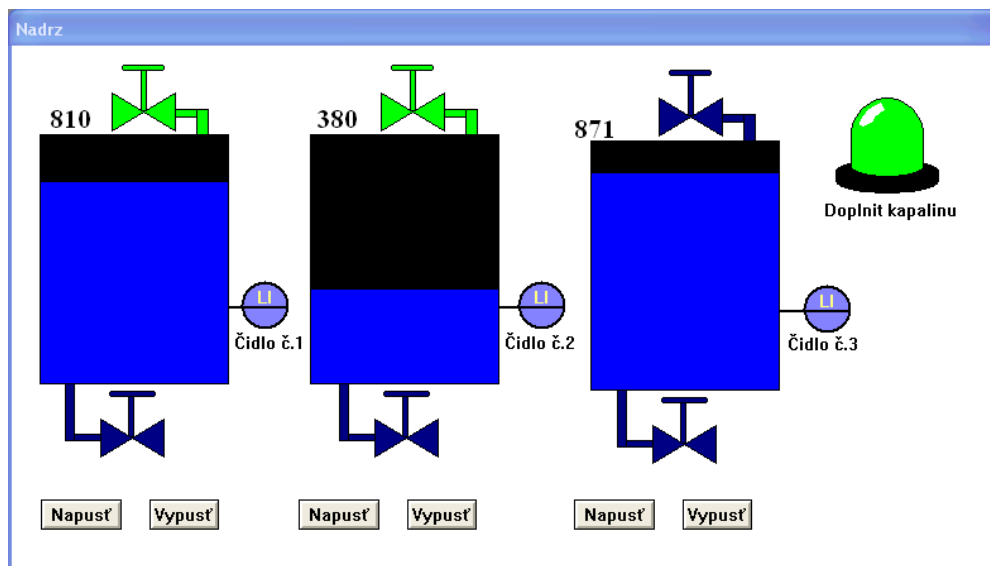
Co jsou animační vazby a k čemu slouží?

Jakým způsobem se konfiguruje animace na objekt?

4.4 Simulace procesu

Simulace procesu je v praxi využívána pro účely ověření správné funkce před nasazením do reálného prostředí. Simulace odhalí možné chyby v programech a validuje vývojové práce na projektech.

Příkladem pro simulaci může být vizualizace situace, kdy jsou pro potřeby dílny ve výrobním procesu umístěny tři nádrže s technologickými kapalinami, viz. Obrázek 4.13.



Obrázek 4.13 – Vizualizace procesu plnění nádrží

V každém zásobníku je umístěno čidlo indikující minimální množství kapaliny nutné pro provoz výrobního procesu. V případě, že alespoň ve dvou zásobnících je již méně kapaliny, než je minimální množství, je potřeba kontrolkou na panelu stav indikovat, aby byla kapalina doplněna.

Tento slovní popis lze přepsat do tabulky vstupních a výstupních veličin, viz. Tab. 4.1 a jim odpovídajícím logickým stavům pro funkci vizualizace a automatického řízení, viz. Tab. 4.2.

Výsledný výpis $AL = N1.N2 + N1.N3 + N2.N3$ logické funkce příkladu vizualizace tří nádrží je možné umístit do skriptu vizualizovaných objektů.

Tab. 4.1 Vstupy a výstupy pro indikaci stavu tří nádrží

Název	Zkratka	Logická hodnota
Vstupy		
Čidlo hladiny v nádrži č.1	N1	Hladina pod úrovní čidla: N1=1
Čidlo hladiny v nádrži č.2	N2	Hladina pod úrovní čidla: N2=1
Čidlo hladiny v nádrži č.3	N3	Hladina pod úrovní čidla: N3=1
Výstupy		
Alarm optický	AL	AL1=1 : Kontrolka svítí

Tab. 4.2 Logické stavy simulační úlohy indikace tří nádrží

N1	N2	N3	AL1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Výpis logické funkce:

$$AL = N1.N2 + N1.N3 + N2.N3$$



Shrnutí pojmů 4.4.

Simulace procesu je v praxi využívána pro účely ověření správné funkce před nasazením do reálného prostředí. Slovní popis úlohy lze přepsat do tabulky vstupních a výstupních veličin a jim odpovídajícím logickým stavům pro funkci vizualizace a automatického řízení. Výsledný výpis logické funkce úlohy je zapsán do skriptu vizualizovaných objektů.



Otázky 4.4.

Jak lze zapsat simulační funkci?

K čemu je simulace využívána v praxi?

4.5 Komunikace s externím prostředím

Vizualizační aplikace lze připojit k průmyslovým automatizačním řídicím nebo informačním zařízením prostřednictvím různých komunikačních programů, tzv. I/O (DDE, OPC) serverů. Aplikace pak mohou pracovat jako klient i server. V aplikacích jsou podporovány různé komunikační standardy, technologie OPC, SuiteLink, Microsoft DDE (Dynamic Data Exchange).

Definice protokolu DDE (Dynamic Data Exchange) je dána třemi úmluvou stanovenými názvy:

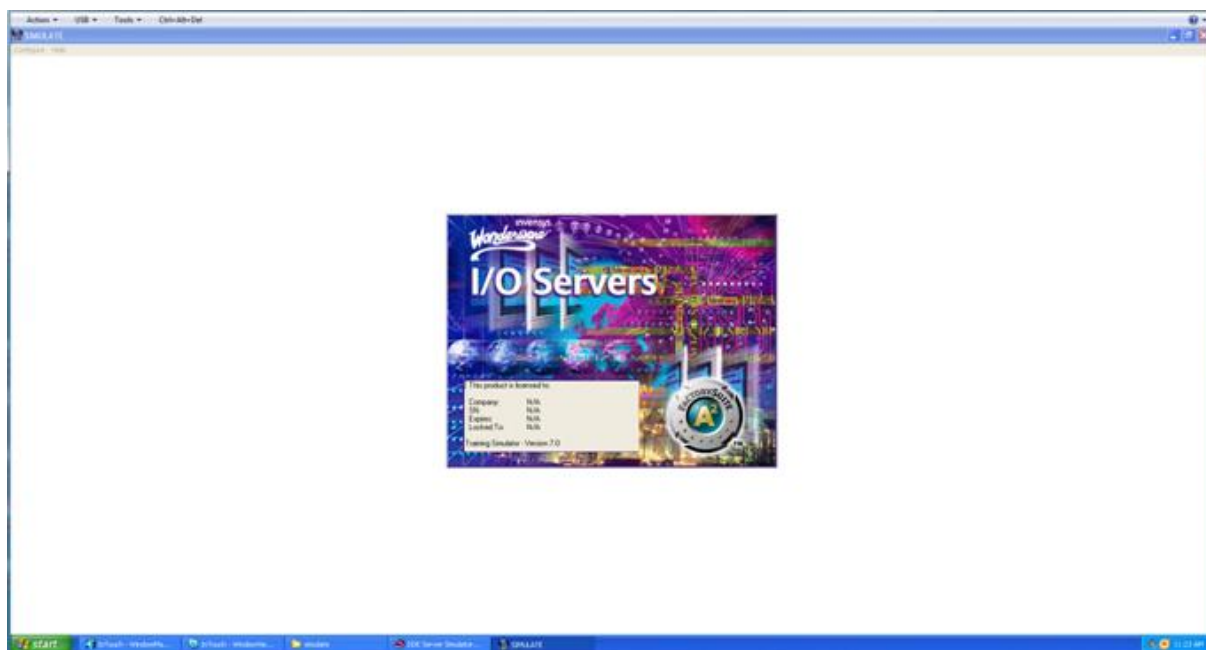
- Název aplikace, která poskytuje data
- Název tématu, pro které běží výměna dat
- Název položky, která je zároveň názvem komunikující proměnné veličiny

Tab. 4.3 uvádí příklady definic DDE komunikace ve vizualizační aplikaci.

Tab. 4.3 Definice DDE protokolu pro komunikaci typu klient-server

Klient	Server	Aplikace	Téma	Položka
View	Simulate	Simulate	PLC1	Ventily napouštěcí V1,V2, ventil vypouštěcí V3, tlačítko Start, analogová veličina L1
View	Excel	Excel	DDE.xls	R2C2,R3C2
Excel	View	View	Tagname	Jakákoliv proměnná z InTouch například pro ExcelSlider

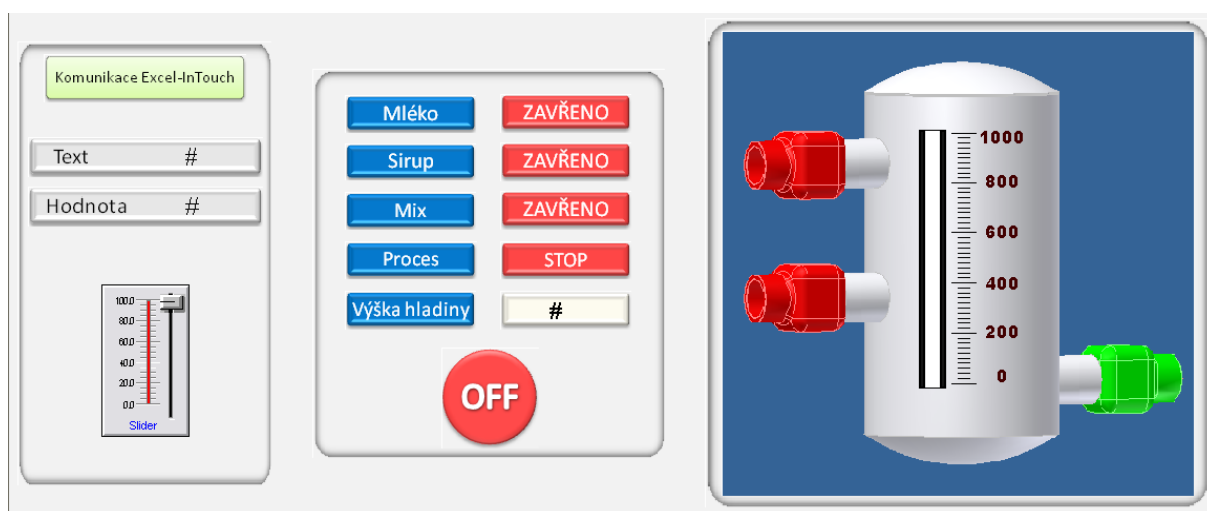
Před spuštěním klienta - runtime vizualizační úlohy - musí být spuštěn i DDE server, aby mohla být komunikace navázána, viz. Obrázek 4.14.



Obrázek 4.14 – Okno po spuštění DDE serveru

Po navázání komunikace mezi klientem – vizualizační aplikací viz. Obrázek 4.15 - a DDE serverem- aplikací Simulate, nebo aplikací Excel:

- Veličiny ukazují hodnoty a stav, které simuluje DDE server nebo posílá aplikace Excel - ukazatel výšky hladiny simulované DDE serverem (Vyskahladiny1), ventily zobrazují aktuální stav ventilů pomocí barvy, ventily jsou napojeny na funkce (1Ventil, 2Ventil, 3Ventil)
- Tlačítkem se spouští a zastavuje DDE simulace
- Hodnoty zapsané v sešitě Excelu (řádek 2 sloupec 2 v sešitu DDE.xls) se zobrazí ve vizualizační aplikaci a hodnota nastavená na posuvném ovládání se zobrazí v sešitě Excelu (řádek 3, sloupec 2).



Obrázek 4.15 – Vizualizace pro komunikace s aplikací Windows/MSEcel a DDE serverem

**Shrnutí pojmů 4.5.**

Vizualizační aplikace lze připojit k externím zařízením pomocí komunikačních programů, tzv. I/O (DDE, OPC) serverů. Aplikace pak mohou pracovat jako klient i server.

Definice protokolu DDE je dána třemi názvy: názvem aplikace, která poskytuje data, názvem tématu, pro které běží výměna dat a názvem položky.

**Otázky 4.5.**

Co je komunikační protokol?

Jak je definován DDE protokol?

5 ALARMY A JEJICH KONFIGURACE

V předchozí části jsme se seznámili s možností vizualizace čidel indikujícími minimální množství kapaliny nutné pro provoz výrobního procesu. V této části bude výklad zaměřen na monitorování různých stavů veličin s možností upozorňování na mezní hodnoty a na způsob jejich definice.



Čas ke studiu: 6 hodin



Cíl: Po prostudování tohoto odstavce budete

- umět definovat alarm, událost a rozlišit rozdíly mezi nimi
- rozumět způsobu potvrzení alarmových hlášení
- umět vysvětlit, jak Internet změnil komunikaci s řídicími systémy a přiblížil je více vizualizačním systémům a rozhraní typu člověk-stroj (HMI)



Výklad

5.1 Zprávy, hlášení, události a alarmy

5.1.1 Účel zpráv a hlášení

Zobrazení zpráv a hlášení na obrazovce pomáhá operátorovi při řízení procesu. Každá zpráva se objevuje při specifických událostech. Zprávy nebo hlášení se mohou vyhodnocovat i v delších časových intervalech, často odděleně od aktuální události, a to zejména s cílem vyloučit systematické chyby.

Zprávy a hlášení můžeme klasifikovat podle následujících hledisek:

- původ a místo vzniku (procesní zprávy, zprávy řídicího systému),
- priorita, důležitost (předvýstraha, výstraha, alarm),
- nutnost potvrzení (zprávy vyžadující potvrzení a zprávy bez potvrzení),
- směr doručení zprávy (došlé – *received*, odesílané – *sent*).

Procesní zprávy a zprávy procesního systému mohou mít původ v technologickém zařízení, např. překročený limit, nebo v řídicím systému, např. porucha sběru dat. Procesní zprávy slouží k řízení procesu.

Zprávy řídicího systému vypovídají o jeho poruchách a jsou určeny pro údržbu.

5.1.2 Priorita zpráv a reakce operátora na zprávy

Pokud ve stejném čase přijde několik zpráv nebo hlášení se stejnou prioritou, o jejich zpracování musí rozhodnout operátor. Priorita vychází z rychlosti a stupně, s jakým každá z hlášených událostí ovlivňuje bezpečnost, hospodárnost a kvalitu výrobku.

Operátor by měl vždy mít na obrazovce vizualizovány pouze ty zprávy, které jsou v daném okamžiku důležité pro proces. Redukci zpráv zobrazovaných na obrazovku operátorského pracoviště lze provést:

- v průběhu návrhu vizualizace aplikace,
- zhuštěním zpráv do skupin,
- filtrací – která brání hodnotám signálu oscilovat s malou amplitudou okolo limitních stavů a dávat tak opakované výsledky bez informační hodnoty,
- automatickou reakcí na událost, pokud jsou některé reakce na události řešeny systémem, vyloučí se informace o těchto zásazích obsluhou,
- operátorem v průběhu jeho činnosti, pokud má volby pro:
 - výběr,
 - potlačení,
 - blokování nepoužitelných zpráv.

Protože má každá zpráva různou důležitost, musí být pro ulehčení práce operátorovi nabízeny zprávy uspořádané podle důležitosti. Priorita musí být vybrána podle toho, jaká reakce se od operátora vyžaduje.

Tab. 5.1 – Priority zpráv, odstupňované podle požadované reakce operátora

	Priorita	Reakce operátora
Nebezpečné zprávy	-	podle bezpečnostních instrukcí
Zprávy procesu		
• Alarm	1	požadovaná okamžitá reakce
• Výstraha	2	zpozornění
• Predvýstraha	3	registrování
Zprávy řídicího systému	-	informování obsluhy

5.1.3 Událost a alarm

Událost je jakákoli změna v pracovním rozsahu hodnot monitorované veličiny v čase. Podle typu veličin lze monitorovat změnu v hodnotě analogových veličin, jejich růst či pokles, změnu stavu diskrétní veličiny z klidu na rozběh nebo opačně z běhu do zastavení, apod. Pokud nadefinujeme limitní hodnoty, tzv. meze, je po jejich překročení vyvolán alarm.

Alarm je tedy překročení definované mezní hodnoty monitorované veličiny. Na takto vyvolaný alarm musí zareagovat obsluha nebo dozor v podobě potvrzení, kterým alarm bere na vědomí. V praxi je důležité sledovat datum a čas, kdy alarm vznikl, ale také kdy na něj bylo zareagováno dozorující osobou nebo obsluhou zařízení. Pokud se jedná o krátkodobé překročení nebo výkyvy hodnoty s následným návratem do normálního rozsahu, i tento stav je

monitorován pro další analýzy nebo posouzení v archivu, v historii alarmů. Může se jednat o opakující se stavy, které indikují možné selhání nebo vznikající problém, který ještě nebyl diagnostikován jiným způsobem.

5.1.4 Ukládání zpráv

Zprávy a hlášení se tedy musí po jejich výskytu na obrazovku ukládat, a to kvůli zamezení problému nebo výrobní ztráty v budoucnu. Podle cíle ukládání a časového intervalu se zprávy rozlišují na:

- aktuální a bezprostřední zprávy – slouží k řízení procesu. Údaje jsou automaticky ukládány do vyrovnávací paměti,
- archiv, historické zprávy – slouží pro ukládání zpráv, které mohou být uloženy na delší dobu. Jsou to hlavně zprávy určené pro údržbu, pro diagnostiku poruch.



Shrnutí pojmů 5.1.

Zprávy a hlášení můžeme klasifikovat podle původu a místa vzniku (procesní zprávy, zprávy řídicího systému), priority a důležitosti (předvýstraha, výstraha, alarm), nutnosti potvrzení (zprávy vyžadující potvrzení a zprávy bez potvrzení), směru doručení zprávy (došlé – *received*, odesílané – *sent*).

Událost je jakákoli změna hodnoty monitorované veličiny v čase. Pokud nadefinujeme limitní hodnoty, meze, po jejich překročení je vyvolán alarm. Alarm je tedy překročení definované mezní hodnoty monitorované veličiny.

Zprávy se rozlišují podle cíle ukládání a časového intervalu na aktuální a bezprostřední zprávy, které slouží k řízení procesu a archiv s historií zpráv.



Otázky 5.1.

Jaký je účel zpráv a hlášení v procesních systémech?

Jaký je rozdíl mezi událostí a alarmem?

Podle čeho se zprávy dají rozlišovat?

5.2 Typy alarmu

5.2.1 Více-stavové alarmy

Alarm může mít definovány sub-stavy. V takovém případě se jedná o více-stavový alarm. Například alarm od analogové proměnné má obvykle několik limitů, limity "High (Vysoký)" a "Low (Nízký)" vymezující normální provozní rozsah a limity "HiHi (Velmi vysoký)" a "LoLo (Velmi nízký)" označující extrémní odchylky od normálu. Může také přecházet mezi dvěma sub-stavy, přičemž stále zůstává v celkovém alarmovém stavu. Je důležité rozlišovat mezi stavem a událostí. Stav může trvat minuty, hodiny, dny nebo týdny. Událost je přechod; uskuteční se a okamžitě zaniká. Alarm je stav; upozornění na alarm je událost.

5.2.2 Potvrzené alarmové stavy

Alarmy představují výstražná hlášení o stavech sledovaných procesů, které by mohly způsobit problémy a vyžadují odezvu operátora, takže je obvyklé, že zprávy vyžadující reakce operátora doprovázejí systémová potvrzení. Potvrzením operátor vyjádří, že zprávu přijal. Potvrzovány by měly být všechny zprávy, které jsou podstatné. Navíc je výhodné, pokud potvrzení doprovází reakce operátora s komentáři. Vztah komentáře ke zprávě musí být jasný.

Typický alarm je aktivován, když některá hodnota přesáhne uživatelem stanovenou mez; např. analogová hodnota přesáhne svůj horní limit. Tím se spustí alarmový stav typu unacknowledged (nepotvrzený alarm), který má upozornit operátora na vzniklý problém. Jakmile operátor alarm potvrdí, alarm přejde do stavu acknowledged (potvrzený alarm). Obrázek 5.2 ukazuje alarmy v alarmových objektech jako tabulku, která umožňuje ke každému alarmu zobrazit informaci o jeho vzniku, hodnotě, typu, dozorující osobě, která jej má potvrdit apod. Barvami jsou pak rozlišeny tyto stavy:

- nepotvrzený alarm - **červená barva**
- potvrzený alarm - černá barva
- veličina se vrátila z kritické hodnoty do normálního pracovního rozsahu - **modrá barva**.

Lokální alarmy lze potvrzovat pomocí .Ack (atribut proměnné) ve skriptu akce, klávesovým skriptem nebo prostřednictvím kontextové nabídky aktivované pravým tlačítkem myši v objektu pro zobrazování distribuovaných alarmů. *Obrázek 5.1* znázorňuje možnost potvrzení alarmu konfigurací na tlačítko pomocí .Ack ve skriptu akce.



Obrázek 5.1 – Okno s tlačítkem pro potvrzení alarmu

Některé signalizační systémy mohou automaticky spustit určitou činnost, např. vytáčení telefonního čísla, zasílání e-mailu. Pak se jedná o automatické reakce na zprávy.

5.2.3 Alarmy aktuální a historické

Pro práci na projektu vizualizační aplikace budeme rozlišovat:

- Aktuální alarm- označení alarmů, které jsou "aktuálně aktivní"
- Historický alarm- označení alarmů, které již zanikly
- Alarm při překročení určité odchylky od požadované monitorované hodnoty, aktuální hodnota se odečte od nastavené hodnoty a absolutní hodnota vypočteného rozdílu se porovnává s jedním nebo dvěma limitními hodnotami vyjádřenými jako procento rozsahu hodnoty proměnné
- Alarm na příliš velkou změnu hodnoty monitorované veličiny v časovém úseku, započítává se aktuální hodnota a předchozí hodnota společně s aktuálním časem a časem předchozí aktualizace. Pokud absolutní hodnota rychlosti změny překročí stanovený limit, alarmový stav je vyhlášen.

Aktuální alarmy								
Date	Time	State	Type	Priority	Name	Value	Limit	Operator
15 I	11:54	UNACK	HIHI	1	Alarm1	15	14	OperNode/Oper...
15 I	11:54	UNACK	HI	250	Alarm2	15	14	OperNode/Oper...
15 I	11:54	UNACK	LO	500	Alarm3	15	14	OperNode/Oper...
15 I	11:54	UNACK	LOLO	750	Alarm4	15	14	OperNode/Oper...
15 I	11:54	ACK	Minor	1	Alarm5	15	14	OperNode/Oper...
15 I	11:54	ACK	Major	250	Alarm6	15	14	OperNode/Oper...
15 I	11:54	ACK	1	500	Alarm7	15	14	OperNode/Oper...
15 I	11:54	ACK	1	750	Alarm8	15	14	OperNode/Oper...

Všechny alarmy								
Date	Time	State	Type	Priority	Name	Value	Limit	Operator
15 I	11:54	UNACK	HIHI	1	Alarm1	15	14	OperNode/Oper...
15 I	11:54	UNACK	HI	250	Alarm2	15	14	OperNode/Oper...
15 I	11:54	UNACK	LO	500	Alarm3	15	14	OperNode/Oper...
15 I	11:54	UNACK	LOLO	750	Alarm4	15	14	OperNode/Oper...
15 I	11:54	ACK	Minor	1	Alarm5	15	14	OperNode/Oper...
15 I	11:54	ACK	Major	250	Alarm6	15	14	OperNode/Oper...
15 I	11:54	ACK	1	500	Alarm7	15	14	OperNode/Oper...
15 I	11:54	ACK	1	750	Alarm8	15	14	OperNode/Oper...

Obrázek 5.2 – Objekty pro zobrazení alarmů (1.) alarmy aktuální, (2.) historie alarmů

Záměrem je, aby operátor viděl na obrazovce odděleně "souhrn" všech aktuálních alarmů, které čekají na potvrzení, a v jiné obrazovce informace o alarmech, které jsou zajímavé až z hlediska historie a jejich analýza je tedy méně naléhavá.



Úlohy k řešení

Řešte následující úlohy.

Příklad 5.1. *Vytvořte alarmový objekt pro zobrazení aktuálních alarmů v reálném čase.*

Příklad 5.2. *Definujte alarmové meze pro monitorovanou veličinu. Nastavte alarmy při dosažení příliš vysoké a nízké limitní hodnoty. Vizualizujte mezní stavy v objektu aktuálního alarmu.*

Příklad 5.3. *Nastavte alarm při překročení určité odchylky od požadované monitorované hodnoty*

Příklad 5.4. *Nastavte alarm tak, aby oznamoval příliš velkou změnu hodnoty monitorované veličiny v časovém úseku*



Řešení příkladů

Podrobný postup práce na příkladech je uveden v animaci „Nastavení alarmu“.



Shrnutí pojmů 5.2.

Alarm může mít definovány sub-stavy, limity "High (Vysoký)" a "Low (Nízký)" vymezující normální provozní rozsah a limity "HiHi (Velmi vysoký)" a "LoLo (Velmi nízký)" označující extrémní odchylky od normálu.

Barvami jsou rozlišeny tyto stavy: pro nepotvrzený alarm - **červená barva**, potvrzený alarm - černá barva a návrat veličiny do normálního pracovního rozsahu - **modrá barva**.



Otázky 5.2.

Jaké typy alarmů již znáte?

Jaké stavy mohou alarmy nabývat?

Jak lze alarmy vizualizovat?

Co znamená potvrzení alarmu operátorem?

5.3 Notifikace událostí na runtime obrazovky operátorských pracovišť

Notifikací rozumíme písemné sdělení skutečnosti, tedy sdělení nějaké zprávy. Další kapitola popisuje nové aspekty a moderní směry v komunikaci mezi člověkem a strojem nebo procesem a porovnává je s tradičními a současnými přístupy v průmyslové automatizaci. U tradičních grafických uživatelských rozhraní (GUI – *graphical user interface*) a v tradičních informačních technologiích jsou významy a vztahy předem definovány a zapisovány do datových formátů a kódů programových aplikací již během vývojové fáze projektu, viz. Kapitola 3.1.2.



Obrázek 5.3 – Informace a vstupy od operátorů na tradiční runtime obrazovce panelu

To má za následek, že pokud později potřebují dva programy v systému komunikovat jiným novým způsobem nebo pokud se něco v systému řízení mění během jeho runtime, viz. Kapitola 3.1.3, a vyvstane požadavek na komunikaci části dat a informací, které předtím komunikovány nebyly, potom musí programátoři aktualizovat kódy aplikačních programů. Tyto zásahy do kódů probíhají off-line, je nutné znovu definovat, co a komu bude posíláno, přeprogramovat datové struktury databází a programovací logiku, pak změny promítnout do grafiky aplikace.

5.3.1 Uživatelé grafického rozhraní

Grafická zobrazení dat mohou v praxi využívat tři typy uživatelů:



První skupinou mohou být tzv. **nonstop uživatelé**, nejčastěji provozní inženýři, kteří kontrolují aktuální zobrazené hodnoty a podle nich zasahují do provozu.



Druhou skupinou jsou **občasní uživatelé**, které zajímají historická data, např. z důvodu servisu nebo údržby.



Posledním typem uživatelů jsou **obchodníci a manažeři** jednotlivých oddělení společností, kterým pomáhá např. srovnání stavu před několika dny nebo měsíci jako podpora při rozhodování.

U některých uživatelů může být neekonomické vybavovat je kompletními systémy SCADA/HMI nebo jinými speciálními klientskými aplikacemi, aktuální, historická nebo statistická data přímo z výroby lze poskytovat prostřednictvím mnohem jednoduššího rozhraní, a to pomocí webového prohlížeče.

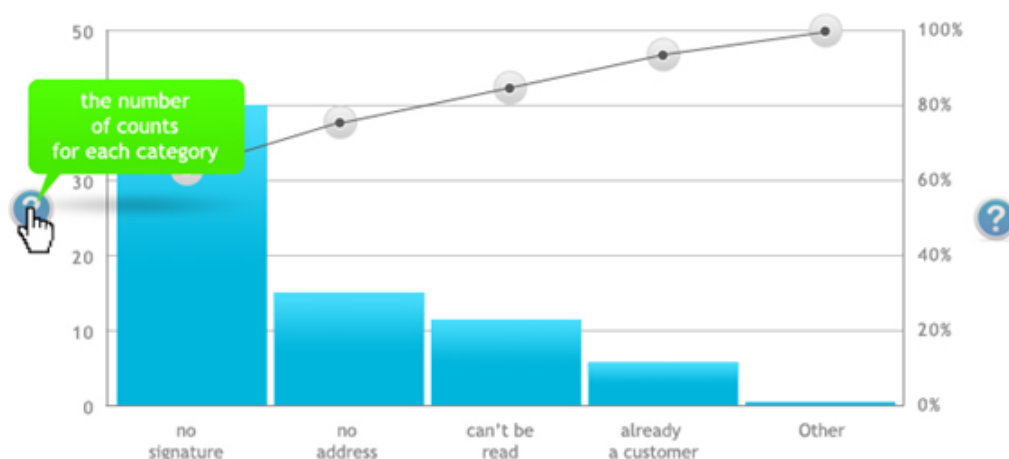
5.3.2 Moderní směry grafického rozhraní

Data v grafické podobě zobrazovaná na webových stránkách podléhají moderním designérským trendům stejně jako jiné webové prezentace. Představme si, že se nacházíme na místě, kde neznáme vůbec nic a máme u sebe jen PDA s připojením k internetu. Na tomto PDA spustíme příslušnou aplikaci, do které zadáme svou pozici. Teď si jej dáme před svůj výhled a díváme se skrz PDA na svět kolem sebe jako přes chytrý dalekohled. PDA aplikace nám ukáže, kde jsme, na co se díváme a různá místa jsou stejně jako např. na Google mapách opatřeny popisky, co se na dané pozici nachází, např. zastávka tramvaje a můžeme se podívat na jízdní řád nebo restaurace a jestli a co se tam vaří.



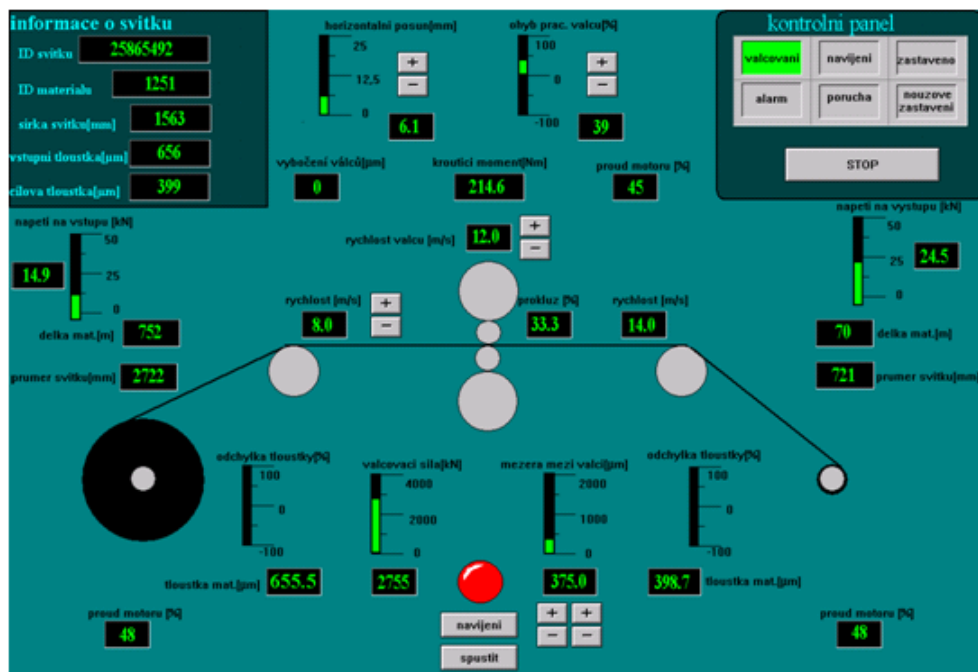
Obrázek 5.4 – Využití vizualizace dat pro webové prezentace

Podobně lze tuto technickou pomůcku využít i v řešení grafů, vyskakující nápovědy je možné umístit nad méně srozumitelná data, popisy souřadnic bodů nebo jako doplňující informace.



Obrázek 5.5 – Využití vyskakující nápovědy pro grafické zobrazení dat

Moderní informační technologie umožňuje posílat změny na displeje některých grafických rozhraní podle změn informace v runtime systému řízení. Moderní software prostředí umožňují programovat významy a obsah dat odděleně od aplikačního kódu. To přináší koncovým uživatelům a operátorům výhody při jejich práci s runtime obrazovkami a dává jim možnosti sdílet zkušenosti a znalosti přímo v reálném čase procesu, viz. Kapitola 1.2. Se software prostředím tohoto typu je přidávání, změna a realizace nového vztahu nebo významu dat a propojení s jiným programem jednoduchá asi jako změna a nasazení externího modelu, který tyto programy sdílejí. Uživatelé tak získávají požadovanou flexibilitu při manipulaci s daty na obrazovkách svých grafických rozhraní.



Obrázek 5.6 – Příklad runtime obrazovky válcovny plechu

5.3.3 Komunikace událostí v architekturách klient – server

Průmyslové aplikace sbírají velké objemy dat, které požadují stále větší počet uživatelů na různých úrovních výrobního, technologického, pracovního procesu (viz. Kapitola 1.1). Tito uživatelé potřebují efektivní nástroje pro provoz monitorovaných a řízených procesů. Z hlediska efektivity není vždy nutné data přenášet neustále, ale podle potřeby uživatelů. Jak už bylo zmíněno dříve, někteří uživatelé vyžadují sumarizovaná data, výsledky výroby, historii průběhu výroby a záznamy o její kvalitě. Tito uživatelé používají pouze jednoduchá rozhraní pro přístup, kontrolu a analýzu dat, například internetové prohlížeče. Internet změnil komunikaci s řídicími systémy na interaktivní a graficky velmi užitečný nástroj, který je nyní ještě bližší vizualizačním systémům a rozhraní typu člověk-stroj (HMI), protože umí zobrazit data koncovým uživatelům nejen do počítačů a i-Pad zařízení, ale i na běžné prostředky, jako je PDA nebo mobilní telefon. Data jsou přenášena mezi web klienty a servery, kde klientem je počítač nebo zařízení dotazující se po záznamech, a serverem je počítač nebo zařízení, které je schopné záznam poslat, protože má pro tohle nainstalován specifický software, viz. také Kap.4.1. V případě, že server pošle dokumenty napsané v některém z podporovaných standardních formátů, například HTML (*HyperText Markup Language*- značkovací jazyk pro hypertext) nebo XML (*eXtensible Markup Language* - rozšiřitelný značkovací jazyk), klient už pro jejich přečtení potřebuje pouze prohlížeč, jedná se tedy o tenkého klienta.



Obrázek 5.7 – Grafické zpracování dat pro tenkého klienta

Web server umí poslat data také v jiných formátech, ale klient v tomto případě potřebuje specifický software pro jejich přečtení, zpracování a zobrazení. V takovém případě se jedná o tlustého klienta.

5.3.4 Rozhraní webových prohlížečů

Průmyslové aplikace vyžadují schopnosti zpracovávat události a zobrazovat je na obrazovky a displeje koncových uživatelů, obsluhuje zařízení s pomocí tenkých klientů řídicích systémů. Zprávy a hlášení jsou definovány pro data aktualizovaná na pozadí výrobních procesů. Ale HTML komunikační protokol je postaven na principu požadavek-odezva. Tenký klient musí požádat webový prohlížeč o “novou stránku”. Bez tohoto požadavku nelze spustit žádný default mechanismus notifikace událostí. Takže v chování uživatelských rozhraní pracujících pro webové prohlížeče aplikací tenkých klientů není podporována automatická reakce na externí události.

Aby se průmyslové aplikace řídicích systémů dostaly až k datům z procesů (řízených s podporou rozhodování obsluhy), potřebují k tomu webové služby. Webové služby zde

reprezentují vrstvu mezi průmyslovou aplikací a tenkým klientem (v rukou člověka) a jsou právě tímto rozhraním k průmyslovým aplikacím (API – *application programming interface*) vyvinutým podle koncepce standardních internetových technologií. API je vlastně soubor pravidel, kódů a software specifikací, podle kterých software programy komunikují. Slouží jako rozhraní mezi různými software programy a zprostředkovává jejich interakce podobným způsobem, jako uživatelský interface zprostředkovává interakci mezi člověkem a počítačem, strojem, procesem.

Datová komunikace zpráv v XML jazyce probíhá na komunikačním protokolu SOAP (*Simple Object Access Protocol*). SOAP obsahuje obálku, soubor kódů a úmluvu pro vzdálené volání procedur RPC (*remote procedure calls*).

5.3.5 Služby pro notifikaci událostí

Notifikace událostí je služba, která rozesílá události na jednu nebo více supervizích stanic. Události mohou být generovány z průmyslové aplikace pomocí:

- business logiky (BL), která běží na aplikačním serveru (AS). Aplikační server má za úkol konsolidaci dat, autorizaci uživatelů, zprostředkovávat přístup k datům a generovat reporty podle daného plánu (časového rozvrhu nebo prováděných aktivit). Tyto služby jsou realizovány jako soubor objektů a procesů na serveru.
- skriptováním (psanými skripty) v počítačovém jazyce, např. C#. Skriptovací prostředí tvoří aplikační programové rozhraní (API).
- komunikací s externími aplikacemi.

Požadavky na aktualizaci dat mohou být generovány celou řadou zdrojů událostí, například:

- zprávami, které jsou vstupem pro požadavek na aktualizaci dat, jako například manuálním zadáním dat, pokud uživatel, obsluha ručně opravuje parametry v rámci zásahů supervize a rozhodování.
- automatickou synchronizací během komunikace mezi různými vrstvami systému (publish-subscribe message).

Každá zpráva musí mít své téma (topic). Příjemci zpráv si vyžádají (subscribe) pouze témata, která je zajímají, odesílatelé zprávu pošlou, a tím je zaručena efektivita komunikace.

Event Time	Object Na	Object Description	Condition	Message Description
21/09/07 12:05:42:675	Presetting	Presetting 6 High	CoilIDAlarm	Error sending telegram CoilID to level 2
21/09/07 12:05:42:675	Presetting	Presetting 6 High	L1AliveAlarm	Error sending telegram L1Alive to level 2
21/09/07 12:05:42:675	Presetting	Presetting 6 High	StandStillAlarm	Error sending telegram StandStill to level 2

Obrázek 5.8 – Zobrazení událostí a alarmů v aplikaci supervizního řídicího systému válcovny



Shrnutí pojmů 5.3.

Grafická zobrazení dat mohou v praxi využívat pomocí GUI přímo z aplikací procesních systémů tři typy uživatelů: provozní inženýři a operátoři (non-stop uživatelé), servis nebo údržba (příležitostní uživatelé) a management (podpora rozhodování).

Moderní informační technologie umožňuje posílat změny na displeje některých grafických rozhraní podle změn informace v runtime systému řízení. Moderní software prostředí umožňují programovat významy a obsah dat odděleně od aplikačního kódu. To přináší koncovým uživatelům a operátorům výhody při jejich práci s runtime obrazovkami a dává jim možnosti sdílet zkušenosti a znalosti přímo v reálném čase procesu.

Internet změnil komunikaci s řídicími systémy na interaktivní a graficky velmi užitečný nástroj, který je nyní ještě bližší vizualizačním systémům a rozhraní typu člověk-stroj (HMI), protože umí zobrazit data koncovým uživatelům nejen do počítačů a i-Pad zařízení, ale i na běžné prostředky, jako je PDA nebo mobilní telefon.

V případě, že server pošle dokumenty napsané v některém z podporovaných standardních formátů, například HTML nebo XML, klient už pro jejich přečtení potřebuje pouze webový prohlížeč.

Zprávy a hlášení jsou definovány pro data aktualizovaná na pozadí výrobních procesů. HTML komunikační protokol je postaven na principu požadavek-odezva. Bez tohoto požadavku nelze spustit žádný default mechanismus notifikace událostí. V chování uživatelských rozhraní pracujících pro webové prohlížeče aplikací tenkých klientů není podporována automatická reakce na externí události.

API – *application programming interface* je soubor pravidel, kódů a software specifikací, podle kterých software programy komunikují a slouží jako rozhraní, které zprostředkovává interakci mezi člověkem a počítačem, strojem, procesem.

Notifikace událostí je služba, která rozesílá události na jednu nebo více supervizích stanic. Události mohou být generovány z průmyslové aplikace např. pomocí business logiky, která běží na aplikačním serveru.

Efektivita komunikace je zaručena tím, že si příjemci zpráv vyžádají (subscribe) pouze témata, která je zajímají a odesílatelé zprávu pošlou.



Otázky 5.3.

Co je to grafický uživatelský interface?

Které podporované standardní formáty dokumentů pro tenkého klienta znáte?

Co je to notifikace událostí?

K čemu je definováno téma zprávy?

6 TRENDY A JEJICH KONFIGURACE

V další kapitole bude popsán způsob, jak zaznamenávat veličiny v čase.



Čas ke studiu: 2 hodiny



Cíl: Po prostudování tohoto odstavce budete

- vědět, co je to log a trend
- analyzovat historii průběhů veličin



Výklad

Účel vizualizace trendů je pomáhat operátorovi v řízení procesu, hlavně při monitorování průběhu minulých, současných a žádaných veličin. Křivky ukazují průběh proměnných hodnot vzhledem k času jako posloupnost bodů nebo čar.

6.1 Typy vizualizací průběhů

Křivky v trendech mohou zobrazovat stavy:

- normální, tj. naměřené hodnoty,
- pomocné, náhradní stavy,
- chybové, například když hodnoty chybějí,
- vybrané intervaly rozsahu hodnot.

Podle reálného času se trendy dělí na:

- současné, reálné – zobrazují aktuální hodnoty a hodnoty v krátkém časovém úseku před reálným časem.
- historické, minulé – zobrazují průběhy z archivních, minulých úseků, hlavně pro analýzu a ověření. Tyto údaje mohou využít i jiné aplikace v architektuře systému.

6.1.1 Reálné trendy

Reálné trendy jsou vizualizovány dynamickými objekty v čase. Jsou nepřetržitě aktualizovány při běhu vizualizačního programu. Zakreslují změny až čtyřech proměnných nebo výrazů. Zobrazují data konfigurovatelná ve vývojovém prostředí, viz. *Obrázek 6.1*.

Real Time Trend Configuration

Comment:

Time Span: ☐ Sec ☒ Min ☐ Hr

Sample Interval: ☐ Msec ☒ Sec ☐ Min ☐ Hr

Color: Chart Color: Border Color:

Time Divisions: Number of Major Div: Minor Div/Major Div: ☐ Top Labels ☒ Bottom Labels Major Div/Time Label: HH:MM:SS Display: ☐ HH ☐ MM ☒ SS

Value Divisions: Number of Major Div: Minor Div/Major Div: ☒ Left Labels ☐ Right Labels Major Div/Value Label: Min Value: Max:

Pen: Expression: Color: Width:

1	Hladina	Blue	1
2	Teplota	Red	1
3	Beh_cerpadla	Green	1
4	Start	Black	1

OK Cancel Clear Select Display Font ... ☐ Only update when in memory

Obrázek 6.1 – Konfigurovatelné parametry reálného trendu ve vývojovém prostředí

6.1.2 Historické trendy

Historické trendy poskytují "snímek" dat z určitého data a času z minulosti. Na rozdíl od reálných trendů historické trendy nejsou dynamické, aktualizují se pouze, když dostanou instrukci k aktualizaci.

Historical Trend Configuration

Historical Tag:

Chart time: Initial Time Span: ☐ Secs ☒ Mins ☐ Hrs ☐ Days

Initial Display Mode: ☒ Min/Max ☐ Average

Color: Chart Color: Border Color:

Time Divisions: Number of Major Div: Minor Div/Major Div: ☐ Top Labels ☒ Bottom Labels Major Div/Time Label: Time: ☐ MM ☐ DD ☐ YY ☒ HH ☒ MM ☒ SS

Value Divisions: Number of Major Div: Minor Div/Major Div: ☒ Left Labels ☐ Right Labels Major Div/Value Label: Min Value: Max:

Pen	Tagname	Color	Width
1	Hladina	Blue	1
2	Teplota	Red	1
3	Beh_cerpadla	Yellow	1
4	Start	Black	1

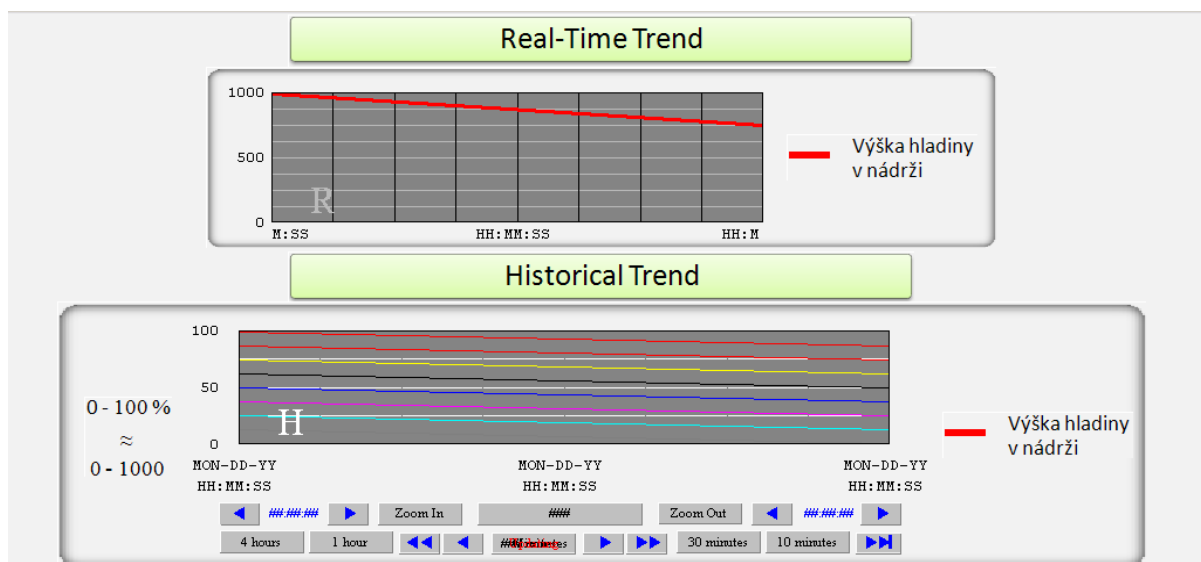
OK Cancel Clear Select Display Font ... ☒ Allow runtime changes

Obrázek 6.2 – Konfigurovatelné parametry historického trendu

Konfigurovatelné jsou i v runtime. Je však třeba povolit vytvoření souboru *Special/Configure/Historical Logging* v dialogu *Historical Logging Properties* „enable“. Je

také nutné vytvořit proměnnou typu HistTrend pro volání dat do zobrazeného trendu. Všechny proměnné pak musí být povoleny k zápisu: „Log data“, což je volba u každé proměnné v dialogovém okně *TagName Dictionary*.

Obrázek 6.3 ukazuje objekty pro vizualizaci hodnoty výšky hladiny v nádrži pomocí reálného i historického trendu ještě ve vývojovém prostředí před spuštěním vizualizace v prostředí runtime.



Obrázek 6.3 – Objekty pro vizualizace reálného a historického trendu konfigurované hodnotami výšky hladiny v nádrži



Shrnutí pojmů 6.1.

Účel vizualizace trendů je pomáhat operátorovi v řízení procesu, hlavně při monitorování průběhu minulých, současných a žádaných veličin v čase.

Reálné trendy jsou nepřetržitě aktualizovány, historické trendy poskytují náhled na data, ale až po zadání příkazu pro aktualizaci na zvolený časový úsek.



Otázky 6.1.

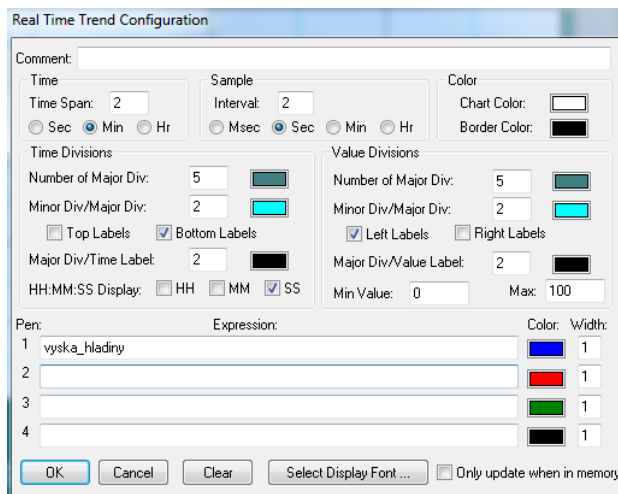
Co je to trend?

Jaké typy trendů vzhledem k reálnému času znáte?

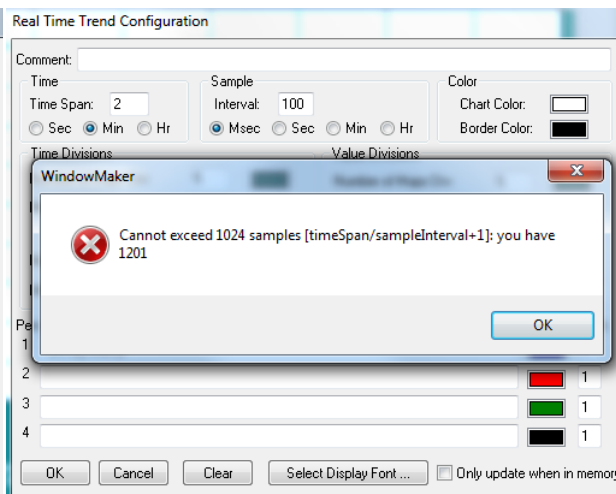
6.2 Logování a ukládání logů

Vizualizovaný průběh musí odpovídat realitě, a proto se při konfiguraci trendu, viz. *Obrázek 6.4*, volí vzorkovací frekvence - *sample interval* - alespoň dvakrát větší, než je maximální požadovaná frekvence monitorované veličiny, pro vzorkovací frekvenci musí platit Shannonův teorém.

Rozhodující při konfiguraci trendu je rovněž volba délky časového úseku - *time span*, který má být vizualizován. Některá vizualizační prostředí mají technicky limitovaný počet, který je při konfiguraci nutné respektovat, viz. *Obrázek 6.5*.



Obrázek 6.4 – Konfigurace reálného trendu hodnoty výšky hladiny v nádrži



Obrázek 6.5 – Chybně zvolená vzorkovací frekvence pro časovou osu v reálném trendu

Je vhodné, aby se křivky prezentovaly a ukládaly jako fyzikální proměnné, a ne v procentech, aby se nemusely ukládat také platné rozsahy. Při zvoleném časovém intervalu zobrazení nemusí být vždy zobrazovány všechny hodnoty, ale jen některé prezentativní vzorky, konsolidované logy, viz také Kapitola 4.1. Aby operátor dokázal správně pochopit trend, musí být zřejmé, jaká metoda konsolidace byla použita, např.:

- min – vybraná a zobrazená hodnota byla minimální,
- max – vybraná a zobrazená hodnota byla maximální,
- mean, avg – vybraná a zobrazená hodnota byla zprůměrována.

Trendy jsou zobrazovány jako barevné spojitě čáry nebo posloupnosti bodů. Doporučuje se, aby v jednom poli bylo maximálně 6 křivek. Pomůckami pro jejich čtení jsou mřížky, pravítka, případně dvě osy hodnot. Pomocí nástrojů pro zoom a filtraci časových úseků by posuv trendů podél časové osy měl umožňovat:

- stránkování,
- rolování, pomocí rolovacího tlačítka,
- zadání žádaného času pro nejnovější bod.



Úlohy k řešení

Řešte následující úlohu.

Příklad 6.1. *Vytvořte objekt trendu pro zobrazení analogové a diskrétní veličiny v čase.*

Příklad 6.2. *Konfigurujte monitorované veličiny pro logování hodnot do historie zobrazené v trendu.*



Řešení příkladů

Podrobný postup práce na příkladech je uveden v animacích „Reálný trend“ a „Historický trend“.



Shrnutí pojmů 6.2.

Při konfiguraci trendů je rozhodujícím parametrem vzorkovací frekvence monitorované veličiny a časový úsek, ve kterém mají být vzorky vykresleny do trendu.

Trendy mohou být spojitě čáry průběhů monitorovaných veličin nebo jen posloupnosti bodů.



Otázky 6.2.

Co je to vzorkovací frekvence?

Jak se volí vzorkovací frekvence?

7 ANALÝZA PROCESŮ

V další kapitole bude popsán způsob, jak zaznamenané veličiny v čase analyzovat.



Čas ke studiu: 2 hodiny



Cíl: Po prostudování tohoto odstavce budete

- vědět, co znamená OEE
- vědět, k čemu je využíván regulační diagram, Pareto, Histogram
- umět spočítat efektivitu procesu z měřených veličin



Výklad

K tomu, aby bylo možné optimalizovat výrobní procesy, zvyšovat využití strojů, materiálů, schopností lidí a dalších zdrojů je potřeba najít příčiny vzniku ztrát a získávat správné, úplné a aktuální informace o událostech ve výrobním procesu. Pokud chceme cokoli zlepšovat, musíme to umět změřit. Celková efektivita zařízení označovaná zkratkou OEE (*Overall Equipment Effectiveness*) je známý koncept v údržbě a určuje způsob jak měřit efektivitu výrobních zařízení.

- „Pokud umíte změřit to, o čem mluvíte a vyjádřit to v číslech, tak o tom něco víte; pokud to změřit neumíte, tak je vaše znalost chudá a nedostatečná.“ – Lord Kelvin

7.1 Sledování ztrát ve výrobním procesu a celkové efektivity výroby

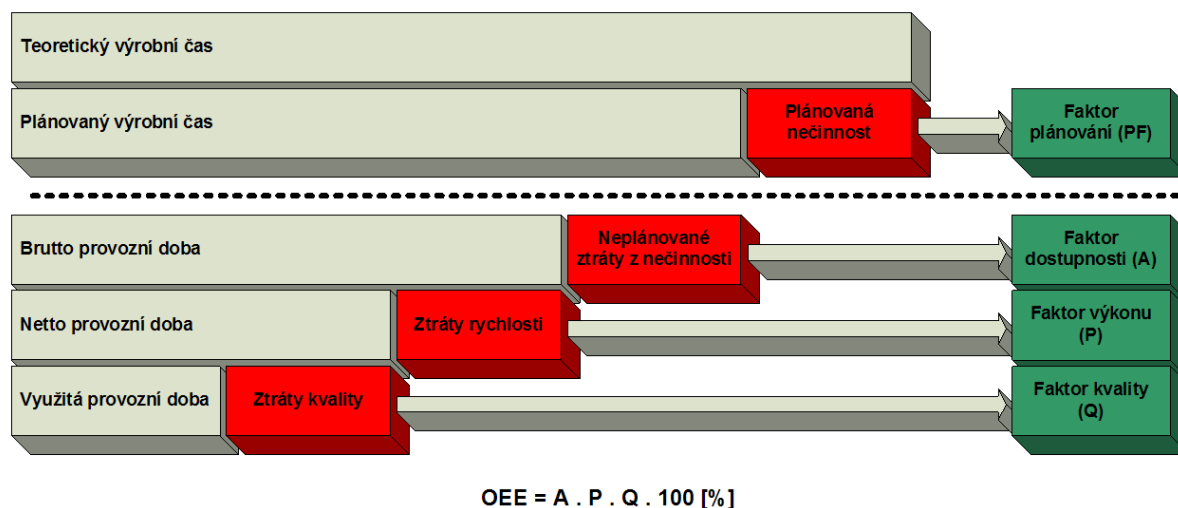
Cílem monitorování v reálném čase je zaznamenat příčiny prostojů různého typu, které mohou vzniknout i s nepatrným časovým odstupem v různých výrobních technologiích a oblastech výrobního podniku, a tím přesně nalézt skutečné příčiny prostojů strojů. Procesní systémy uplatňující koncept OEE pomáhají výrobcům sledovat a analyzovat prostoje, zjistit skrytou kapacitu výroby pro lepší využití výrobních zařízení nebo zdůvodnění nákupu nových strojů, měřit efektivitu zařízení a upřesnit intervaly údržby. Využitím standardních IT nástrojů usnadňují integraci s ostatními informačními systémy v podniku - MES, ERP, aj.

7.1.1 Ztráty ve výrobním procesu

V každé výrobě vznikají ztráty, které je možné rozdělit do několika kategorií:

- Plánované ztráty
- Operativní ztráty
- Výkonové ztráty
- Nekvalita výroby

Některé ztráty ve výrobním procesu, jako jsou například plánované nečinnosti (státní svátky, dovolené zaměstnanců, povinné pracovní přestávky), nelze odstranit. Ostatní ztráty pak představují skrytou kapacitu výroby. K jejich zjištění a analýzám jsou využita data z procesu v kombinaci s informacemi o dostupnosti a výkonnosti výrobních zařízení a kvalitě výroby na těchto zařízeních. Výsledné údaje umožňují jednoznačné a porovnatelné hodnocení, jak jsou jednotlivá výrobní zařízení využívána.



Obrázek 7.1 – Schéma výpočtu podle metodiky OEE – celková efektivita výroby

Obrázek 7.1 zobrazuje jednoduchý princip výpočtu celkové efektivity výroby ze zjištěných dat a informací týkajících se teoretického výrobního času, například 1 kalendářní měsíc, tj. 30 dní a 24 hodin denně, a zaznamenaných ztrát. Dílčí výpočty vedou na stanovení:

- Dostupnosti (A) vypočtené jako podíl hrubé (brutto) provozní doby a využitelného (plánovaného) výrobního času. Plánované nečinnosti, jak již bylo zmíněno, zahrnují například přestávky v dané směně, státní svátky v daném roce;
- Výkonu (P) vypočteného jako podíl čisté (netto) a hrubé (brutto) provozní doby. Hrubá provozní doba je definována například jedno- až tří-směnným provozem po 8 hodinách;
- Kvality (Q) vypočtené jako podíl využití a čisté (netto) provozní doby, přičemž využitá pracovní doba nezahrnuje dobu strávenou výrobou zmetků.

Výsledný koeficient OEE se uvádí v procentech a je dán součinem dostupnosti, výkonnosti a kvality vynásobené 100.

Příklad 7.1. Plánovaný výrobní čas

Na pracovišti s 8 hodinovou pracovní směnnou je sledováno několik indikátorů, z nichž sestavíme výpočet celkové efektivity výroby pracoviště za dané období. Nejprve určíme plánovanou dobu výroby. Pro každou směnu jsou předem dané 2 přestávky, každá 15 minut a jedna pauza na oběd v délce 30 minut. Plánovaná doba výroby pro jeden den s jednou směnou je tedy celková pracovní doba, od které se odečtou plánované pracovní přestávky a pauza na oběd, tj. $(8 \cdot 60 \text{ minut}) - (2 \cdot 15 \text{ minut} + 30 \text{ minut}) = 420 \text{ minut}$.

Příklad 7.2. Hrubá provozní doba

Další sledované indikátory jsou prostoje výrobního zařízení, např. v této směně obsluha zaznamenala 47 minut, kdy zařízení nepracovalo. Odečtením 47 minut prostojů od 420 minut plánovaného výrobního času dostáváme 373 minut provozní doby.

Příklad 7.3. Stanovení počtu kvalitně vyrobených kusů

Pro výrobní zařízení bylo normou stanoveno vyrobit 60 kusů výrobků za minutu. V této směně však bylo skutečně vyrobeno 19271 kusů, přičemž kontrolou kvality neprošlo 423 kusů, tj. $19271 - 423 = 18848$ vyrobených kusů.

Příklad 7.4. Dostupnost výrobního zařízení (A)

Koeficient dostupnosti A je dán poměrem $373/420 = 0.88$, tj. 88%

Příklad 7.5. Výkon výrobního zařízení (P)

Normovaný výkon umožňuje vyrobit 60 kusů za minutu, tedy za 373 minut skutečné doby by podle této normy mohlo být vyrobeno 22380 kusů. V dané směně však bylo skutečně vyrobeno jen 19271 kusů (tj. asi 51,66 kusů za minutu). Výkon P výrobního zařízení je dán podílem skutečného a normovaného výkonu zařízení $(19271/373)/(60/1) = 0,8611$, tj. 86.11%

Příklad 7.6. Kvalita výrobního zařízení (Q)

V dané směně bylo vyrobeno 19271 kusů, ale jen 18848 kusů prošlo výstupní kontrolou kvality, proto je koeficient kvality dán poměrem počtu dobrých kusů k celkovému počtu vyrobených kusů, tj. $18848/19271 = 0,9780$, tj. 97,80%

Příklad 7.7. Výpočet celkové efektivity zařízení OEE

Pro výpočet celkové efektivity použijeme koeficienty dostupnosti, výkonu a kvality daného výrobního zařízení v dané směně a podle vztahu pro výpočet $OEE = A * P * Q * 100$ [%] vychází celkové $OEE = 0,8881 * 0,8611 * 0,9780 = 0.7479$, tj. 74,79%.

**Shrnutí pojmů 7.1.**

Procesní systémy uplatňující koncept OEE získávají přesná data a poskytují informace o prostojích výrobních zařízení a jejich příčinách. Ukazatel OEE indikuje, jak efektivní je výroba na daném výrobním zařízení. Výpočet OEE se provádí ze zjištěných dat a informací týkajících se teoretického výrobního času a zaznamenaných ztrát.

**Otázky 7.1.**

K čemu je využíván indikátor OEE?

Jaké typy ztrát jsou při výpočtu efektivnosti výroby rozlišovány?

7.2 Statistické řízení procesů

Statistické řízení procesů SPC (*Statistical Process Control*) představuje metody a nástroje, které využívají údaje získávané v reálném čase z výrobních procesů s cílem předcházení zmetkovitosti a zachování vysoké a neměnné kvality provozovaných výrobních procesů a vyráběných produktů.

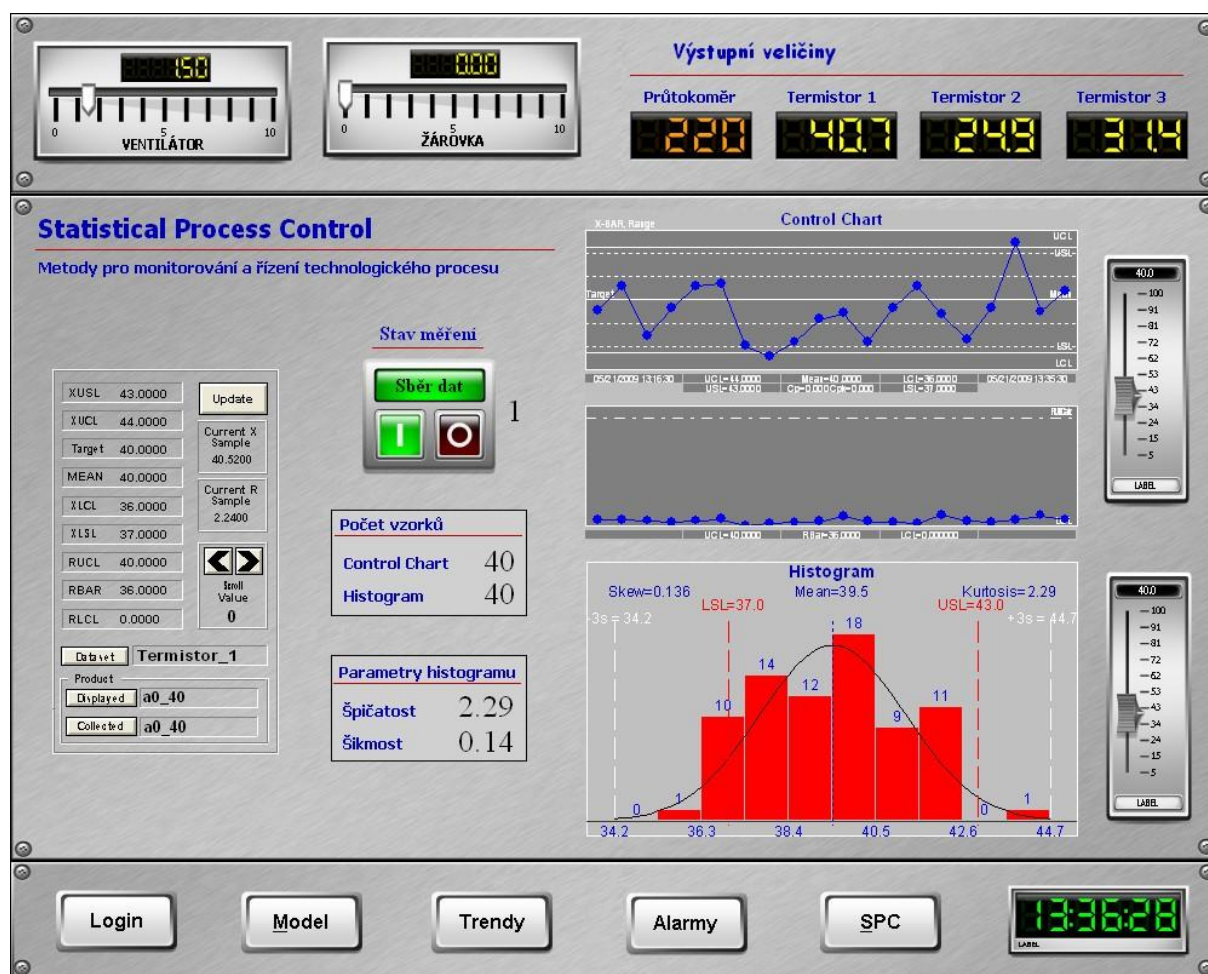
Data jsou analyzována i z dřívějších procesů a statistickými funkcemi (průměr, pohyblivý průměr, směrodatná odchylka) porovnávána s daty z procesů v reálném čase s cílem nastavení limitních parametrů vedoucích ke zlepšení provozu. Sledovaná a porovnávaná data jsou zpracovávána statistickými metodami, které jsou schopny odhalit odchylku od předepsaných parametrů, vypočítat trend dalšího vývoje a upozornit alarmovým hlášením na obrazovku nebo zasláním zprávy na mobilní zařízení operátora nebo obsluhu výrobního zařízení, aby se vznikající problémy mohly okamžitě začít řešit.

Statisticky řízený proces je takový, ve kterém nepůsobí žádné systematické vlivy na proces a nezpůsobují změny. Příčiny změn v řízených procesech mohou být dvojího druhu:

- náhodné - např. změna dodavatele, nestálá kvalita materiálu, změna parametrů, opotřebení nástroje;
- systematické - např. použití vadného materiálu, nesprávně nastavený nástroj nebo stroj.

K metodám statisticky řízených procesů řadíme celou řadu nástrojů běžně používaných v praxi. Mezi nejznámější patří:

- Regulační diagramy
- Histogram
- Pareto



Obrázek 7.2 – Příklad návrhu procesní obrazovky s nástroji SPC

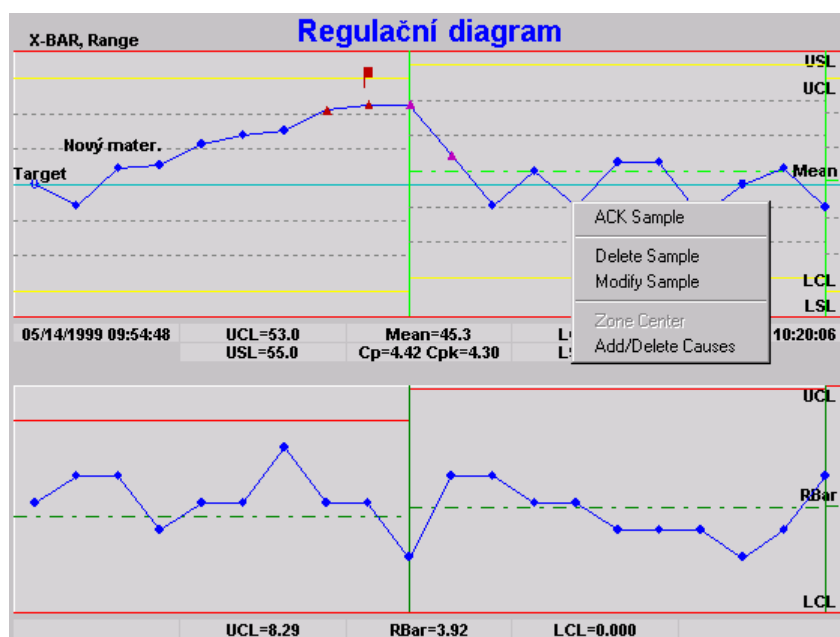
7.2.1 Regulační diagramy

Regulační diagramy dávají v přehledné grafické formě údaj, v jakém stavu se proces nachází a jestli tento stav potrvá. Jednoduše zobrazují samotný průběh procesu v čase, centrální přímkou charakterizující předepsaný průběh, definované mezní přijatelné hodnoty a pak kolik a které body té skutečné charakteristiky leží mimo takto definované rozmezí, tedy $\pm 3s$ od centrální přímky, kde s označuje směrodatnou odchylku.

Rozhoduje se porovnáním současných hodnot se dvěma regulačními mezemi nastavenými od centrální přímky ve vzdálenosti trojnásobku směrodatné odchylky:

- $(3s) - s$ je nazývána jako dolní regulační mez (*Lower Control Limit - LCL*)
- $(3s) + s$ je horní regulační mez (*Upper Control Limit - UCL*),

viz. Obrázek 7.3. Rozmezí mezi těmito mezemi znázorněnými přímkami $\pm 3s$ dává monitorovanému procesu pravděpodobnost 0,9973 (tj. 99,73 %), že hodnoty sledované charakteristiky splňují nároky kladené na statisticky stabilní proces.



Obrázek 7.3 – Regulační diagram

V praxi je využíváno více druhů regulačních diagramů, každý z nich je určený pro různé typy rozhodnutí a různé typy údajů, např.:

- Diagram individuální X vykresluje vzorky, které jsou tvořené jedním měřením
- Diagram pro výběrový průměr ($\bar{x}$ - R) zobrazí průměrný střed procesu na základě více měření a rozpětí R (šířku) jednotlivých měření pro každý vzorek
- Diagram pro výběrový průměr s výběrovou směrodatnou odchylkou ($\bar{x}$ - s) zobrazuje směrodatnou odchylku pro každý vzorek
- Diagram pro pohyblivý průměr (*moving X* , *moving R*) používá pohyblivý průměr n vzorků na zobrazení středu a šířky daného procesu
- Diagram EWMA (*exponentially-weighted moving average*) dává větší váhu posledním vzorkům, aby se nejdříve zjistily změny v procesech

Důležitým faktorem v používání regulačních diagramů je jednoduchost jejich konstrukce a použití jako součásti celého procesu řízení na požadovanou hodnotu.

7.2.2 Histogram

Histogram představuje grafické znázornění rozdělení četností, které se mohou týkat například rozměrů výrobků, chemického složení, pevnosti, napětí apod. nebo hodnot parametrů výroby ovlivňujících kvalitu, jako jsou rychlost, teplota, tlak, atd.

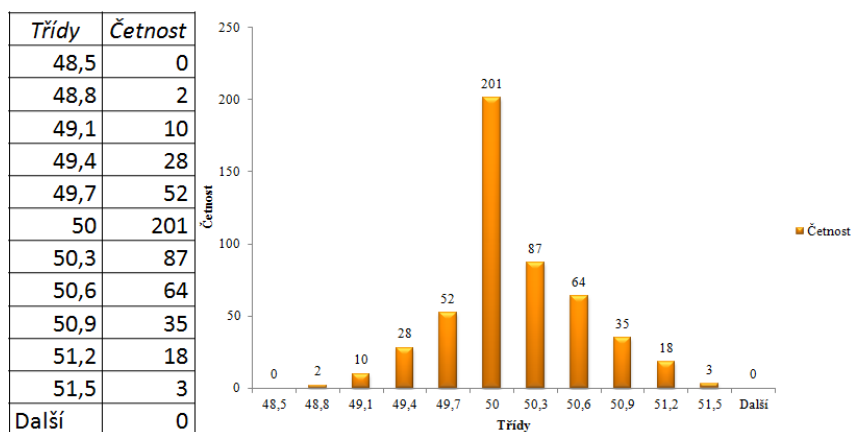
Základem pro sestavení histogramu mohou být naměřené údaje, pro názornost sestavené do tabulky souboru měření, z níž jsou patrné četnosti výskytu naměřeného parametru rozměru výrobku v definovaném rozsahu tohoto souboru měření.

Normální rozvrstvení hodnot kolem cílové hodnoty ukazuje Tab. 7.1.

Tab. 7.1 Normální rozdělení kolem cílové hodnoty

Číslo výrobku	ProductXYZ	
Procesní inženýr	Petr Pavel	
Dávka:	Lot-2013-1234	
Měřeno:	3.4.2013	
Rozměr	Výskyt	Celkem
98.8		
98.9		
99		
99.1		
99.2	X	1
99.3	X	1
99.4	XX	2
99.5	XX	2
99.6	XXXXX	5
99.7	XXXXXXXXXXXXXX	13
99.8	XXXXXXXXXXXXXXXXXXXXXX	19
99.9	XXXXXXXXXXXXXXXXXXXXXXXXXX	21
100.0	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	22
100.1	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	21
100.2	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	18
100.3	XXXXXXXXXXXXXXXXXXXXXX	13
100.4	XXXXXXX	6
100.5	XXX	3
100.6	XX	2
100.7	X	1
100.8		
100.9		
101.0		
101.1		
101.2		

Histogram je sloupkový graf, jehož základnu-osu x- tvoří sloupce stejné šířky a na osu y jsou vyneseny četnosti hodnot sledované veličiny v daném intervalu vyjádřené výškou sloupců.

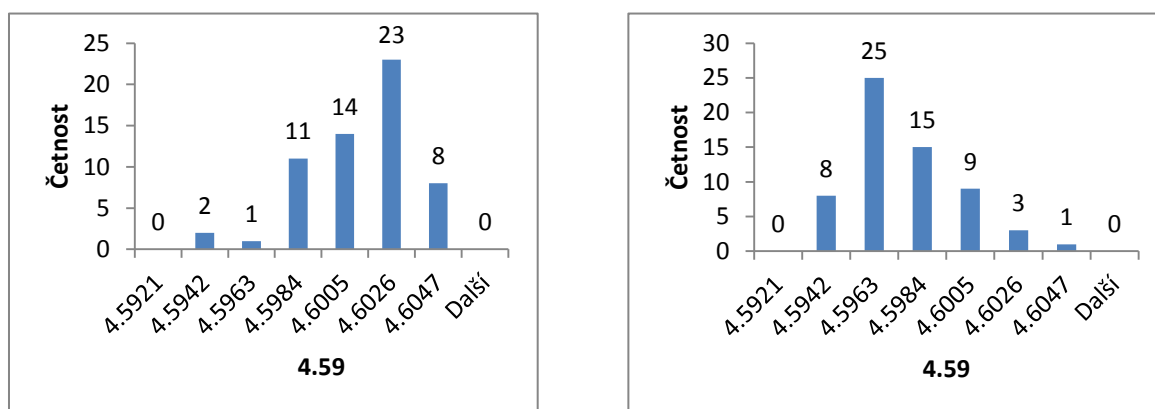


Obrázek 7.4 – Histogram

Vzhled a tvar histogramu vypovídá hlavně o systematických vlivech na proces, které působí na určitý parametr kvality. Histogram může nabývat různých tvarů, přitom odchylky od zvonovitého tvaru ukazují na problém, jehož příčiny je nutné dále analyzovat.

Tvar histogramu je posuzován z hlediska jeho špičatosti (kurtosis) a šikmosti (skewness):

- Špičatost vyjadřuje míru rozložení distribuční funkce Gaussova rozdělení.
- Šikmost charakterizuje protažení křivky na stranu levou-negativní protažení-nebo pravou (pozitivní protažení) od normálního rozložení. Pokud je křivka symetrická, je šikmost nulová.



Obrázek 7.5 – Histogramy se špičatostí vlevo (negativní) a vpravo (pozitivní)

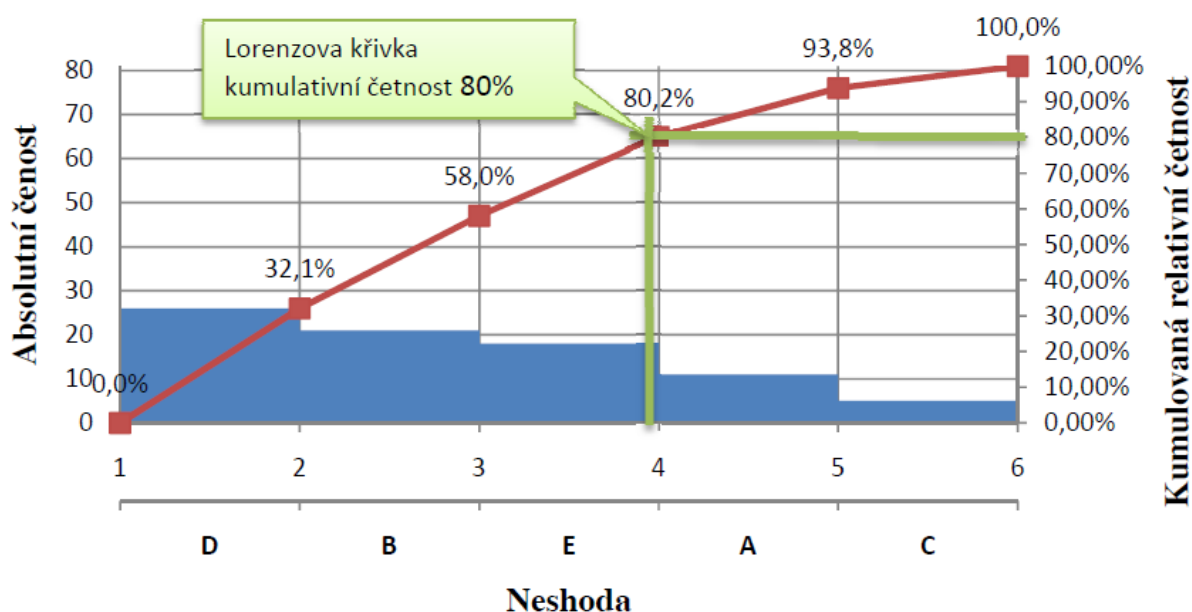
7.2.3 Pareto diagram

Pareto analýza hodnotí výskyt vad a je podkladem a nástrojem pro návrh nápravných opatření nebo vymezení příčin problémů. Název analýzy pochází z principu popsaném a publikovaném významným italským ekonomem Vilfredem Paretem v práci o vztahu jednoduchých faktorů k celkovému efektu, kde mimo jiné prokázal, že nepatrná část obyvatel má významný podíl na celkovém majetku charakterizovaném pravidlem "*the vital few and the trivial many*" (podstatná menšina a triviální většina).

Později tyto statistické vztahy znázornil křivkou M. O. Lorenza, proto se také této křivce dnes říká Lorenzova křivka. V managementu kvality se tento nástroj začal používat po roce 1970 díky J.M. Juranovi, který využil těchto poznatků k vytvoření tzv. Paretova diagramu.

Joseph M. Juran zastával názor, že většina (80-95%) problémů s kvalitou výroby je způsobena pouze malým počtem příčin (5-20%), jež se na nich podílejí a vystihl jej tvrzením "*the vital few and the useful many*" (podstatná menšina a užitečná většina). Dnes je tento princip znám jako Paretův princip 80:20 a v praxi používán pro analýzy:

- počtu a příčin výroby neshodných výrobků a jejich druhů
- časových a finančních ztrát spojených s neshodnými výrobky
- reklamací z hlediska finančních ztrát nebo důvodů reklamací
- příčin prostojů strojů



Obrázek 7.6 – Pareto diagram



Shrnutí pojmů 7.2.

Statistické řízení procesů podporuje svými nástroji a metodami okamžité informování obsluhy výrobního zařízení tak, aby na vznikající problémy mohla reagovat okamžitě a zachovat plynulý chod výroby bez zmetků, viz. *Obrázek 7.2*.

Statisticky řízený proces je takový, ve kterém nepůsobí žádné systematické vlivy na proces a nezpůsobují změny. Mezi nástroje SPC patří regulační diagramy, histogram a Paretův diagram.

Regulační diagramy poskytují informace, v jakém stavu se proces nachází a jestli tento stav potrvá. Histogram reprezentuje grafické znázornění rozdělení četností hodnot sledované veličiny. Paretův princip 80:20 vystihuje zásadu, že většina problémů s kvalitou výroby je způsobena pouze malým podílem příčin, jinými slovy odstraněním 20% zásadních příčin bude řešeno 80% všech poruch.



Otázky 7.2.

Co reprezentuje zkratka SPC?

Jaké nástroje do SPC patří?

8 ZNALOSTI V PROCESNÍM ŘÍZENÍ

V další kapitole bude popsán způsob, jak pracovat s veličinami, které nemusí mít měřicími přístroji a instrumentací měřitelnou a konkrétní hodnotu, ale jsou v praxi hodnoceny a zahrnuty do rozhodování a řízení díky formě znalostí, zkušeností a intuice managementu podniků nebo obsluhy výrobních zařízení.



Čas ke studiu: 2 hodiny



Cíl: Po prostudování tohoto odstavce budete

- vědět, jak popsat znalosti a zkušenosti z praxe
- umět vysvětlit způsob prezentace znalostí do řídicího systému



Výklad

V praxi se při rozhodování a řízení setkáváme s událostmi, které časem vytvářejí naše zkušenosti, potvrzují naše znalosti a podporují intuici běžnou pro lidské chování, podle nichž pak v určitých situacích jednáme. Znalosti a zkušenosti mohou mít dvě odlišné dimenze:

- mohou se týkat faktických údajů, zdrojových dat, které mají být prezentovány nebo jsou to
- formální vztahy mezi těmito faktickými údaji.

V programech a algoritmech jsou fakta, tvrzení a vztahy mezi nimi zapisovány matematicky, s využitím operátorů logiky předpokladů – predikátové logiky, např. AND, OR, ekvivalence, implikace, a výroky a kvantifikátory. Z předpokladů a známých tvrzení a faktů jsou odvozovány důsledky a akční zásahy programované do algoritmů procesních systémů.

8.1 Znalosti

Pro zpracovávání znalostí v řídicích systémech je důležitá jejich strukturovaná prezentace tak, aby znalosti mohly a uměly vyjadřovat logickou formu pro reprezentaci jednoduchých faktů v systému. Mezi požadavky kladené na prezentace znalostí v řídicích systémech patří:

- Schopnost adekvátně reprezentovat jakoukoli znalost
- Manipulovat, pracovat se strukturou reprezentace znalostí tak, aby mohly být odvozeny struktury pro reprezentaci nových znalostí
- Rozšiřovat znalosti, zapracovat do stávající struktury novou znalost, což je typické pro učící se systémy
- Získat novou znalost, odvodit novou znalost ze známých skutečností.

8.1.1 Získávání znalostí

Tak jako u sběru dat i znalosti jsou sbírány či získávány, ale proces získávání znalostí - *knowledge acquisition* - je odlišný, protože souvisí se zjišťováním znalostí ze zdrojů znalostí, tedy od expertů v oboru, dokumentovaných textů, schémat, tabulek a jiných forem již zpracovaných a uložených informací.

Jedním způsobem získávání znalostí je forma přímé spolupráce a konzultace s experty a znalostními inženýry, metoda *brain storming*, strukturovaných interview atd.

Druhý způsob je automatizovaný a využívá metody strojového učení - *machine learning* - od expertů, z textů a dalších dat.

8.1.2 Metody prezentace znalostí

Pro ukládání znalostí v jejich strukturované prezentaci do databáze, schémat v rámcích, skriptech, nebo pravidlech a dále pro vyjadřování vztahů mezi subjekty znalostí jsou využívány dvě základní metody prezentace:

1. Deklarativní
2. Procedurální

Deklarativní prezentace znalostí prezentuje znalosti statickou formou. Výhodou je, že lze každý fakt uložit jen jednou, i když je pak využit vícekrát a že lze do systému snadno přidat nová fakta. Mezi mechanismy deklarativní prezentace patří:

- Sémantické sítě: vyjadřují popis objektů a vztahů a podporují jejich dědičnost
- Koncepční závislost: popisují vztahy mezi komponentami v akci
- Rámce: vyjadřují obecné znalosti o třídách objektů, znalosti pravdivé pro většinu případů z komplexní reprezentace objektů
- Skripty: popisují sekvence událostí
- Pravidla: zapisují závislost mezi příčinou a důsledkem ve tvaru *jestliže->pak*.

Příkladem pravidla, které zapisuje určitou znalost může být:

IF rozměr (X) <= 30mm THEN hmotnost (X) = do_1kg

Pravidla, stejně jako ostatní formy prezentace znalostí jsou však užitečná i při zápisu neurčitosti nebo nejistoty:

IF rozměr (X) = velký THEN hmotnost (X) = velká

Většina znalostních systémů je založena na pravidlech, nebo kombinuje pravidla s jiným způsobem reprezentace.

Procedurální prezentace znalostí jsou metodou, která ukládá znalosti do procedury, vyjadřující jak něco provést. Výhodou je, že lze prezentovat znalosti typu procesu „Jak zpracovat...“, lze zpracovat znalosti předem dané jako default a později upřesňované nebo znalosti pravděpodobnostního charakteru odvození či heuristické znalosti opřené o zkušenosti a expertízu lidí.

8.1.3 Heuristické řešení problémů

Heuristika z řečtiny heuriskó, εὐρίσκω – nalézt, objevit - znamená zkusmé řešení problémů, pro něž neznáme algoritmus nebo přesnější metodu. Heuristické řešení je často jen přibližné, založené na poučeném odhadu, intuici, zkušenosti nebo jednoduše na zdravém rozumu.

Heuristický postup řešení problémů využívá principů evolučních algoritmů k nalezení řešení složitých problémů, postup, při němž není nalezeno exaktní řešení, ale výsledkem je pole nebo oblasti řešení, jež jsou nejbližší řešení ideálnímu (očekávanému).



Shrnutí pojmů 8.1.

Proces získávání znalostí je odlišný od procesu sběru dat a vyznačuje se různými technikami spolupráce s experty a znalostními inženýry.

Mezi požadavky kladené na prezentace znalostí v řídicích systémech patří schopnost adekvátně reprezentovat jakoukoli znalost, manipulovat, pracovat se strukturou reprezentace znalostí tak, aby mohly být odvozeny struktury pro reprezentaci nových znalostí, rozšiřovat znalosti, zapracovat do stávající struktury novou znalost, což je typické pro učící se systémy a získat novou znalost, odvodit novou znalost ze známých skutečností.

Deklarativní prezentace znalostí prezentuje znalosti statickou formou. Většina znalostních systémů je založena na pravidlech, nebo kombinuje pravidla s jiným způsobem reprezentace. Procedurální prezentace znalostí jsou metodou, která ukládá znalosti do procedury, vyjadřující jak něco provést.



Otázky 8.1.

Jaké požadavky jsou kladeny prezentaci znalostí?

Jak lze v systému popsat znalost?

8.2 Znalosti při posouzení rizika procesu

Kritické faktory sledované v procesech mohou být vyjadřovány jako riziko, pravděpodobnost, že dojde k havárii, že nastane situace, jejímž důsledkům a škodám bude vystaveno okolí procesního systému.

Riziko převedené na kritičnost mění koncepci pravděpodobnosti na předpověditelnosti s cílem kritickým situacím a událostem s vyšší pravděpodobností předcházet.

Při úvahách pravděpodobnosti či předpověditelnosti události velmi záleží na spolehlivosti dat, ze kterých jsou vypočteny. Spolehlivost dat může značně ovlivnit např. předpověditelnost místa vzniku poruchy, které je pro tento účel monitorováno s cílem poruše zabránit.

Kategorie spolehlivosti se mohou dělit například takto:

- Vysoce spolehlivá data:
 - inspekce byla provedena na základě kvalitní inspekční rozvahy a jsou posuzována jako plně postačující,
 - měření nejsou starší než 20 % odhadnuté zbytkové životnosti
- Reprezentativní data:
 - odpovídající množství měření provedeno náhodným přístupem bez předchozí korozní studie
 - rozptyl naměřených výsledků měření je poměrně malý
 - měření nejsou starší než 30 % odhadnuté zbytkové životnosti
- Dobrá data:
 - ne zcela postačující množství měření s náhodným přístupem
 - rozptyl naměřených výsledků měření poměrně malý
 - měření nejsou starší než 40 % odhadnuté zbytkové životnosti
- Orientační data:
 - data neodpovídající předchozím kritériím.



Shrnutí pojmů 8.2.

Procesy lze posuzovat z hlediska rizika nebo kritičnosti. Riziko vyjadřuje pravděpodobnost, že k události dojde. Kritičnost pracuje s předpověditelností, že k dané události může dojít.

Vyhodnocení rizika nebo kritičnosti procesu spoléhá na kvalitu poskytovaných dat o procesu. Spolehlivost dat je různá, od velmi spolehlivých až po orientační, každá kategorie spolehlivosti je hodnocena z pohledu kvalifikovaného odborníka.



Otázky 8.2.

Jaký je rozdíl mezi posuzováním procesu z hlediska rizika a kritičnosti?

Jak se posuzuje spolehlivost dat?

9 SPOLEHLIVOST A ZABEZPEČENÍ PROCESNÍCH SYSTÉMŮ

V další kapitole jsou zmíněny pojmy spolehlivosti a zabezpečení systému procesního průmyslu, možnosti identifikace a ověřování uživatelů v průmyslové automatizaci v závislosti na potřebách vyvolaných nutností dokladování technologických dat spojených s kvalitou výroby a certifikací podniků.



Čas ke studiu: 2 hodiny



Cíl: Po prostudování tohoto odstavce budete umět

- ✚ Rozlišit rizika při manipulaci s daty
- ✚ Popsat rozdíl mezi bezpečností přístupu a zabezpečení proti výpadku



Výklad

Důležitou úlohou procesních systémů je zajištění důsledného dodržování správných výrobních postupů, vysoké kvality výrobků, konzistentního provozu výrobních strojů nebo linek a zlepšení spolehlivosti a opakovatelnosti výchozích nastavení předepsaných parametrů výrobních zařízení.

Jejich funkce spočívá také v:

- automatickém záznamu všech definovaných výrobních událostí, meziproductových a dokončených sérií nebo dávek,
- přenosu materiálů během operací,
- okamžité zjištění odchylek od předepsaných údajů pro spotřebu materiálů, energií a ostatních zdrojů a
- identifikace přítomnosti odpovědné obsluhy.

Díky strukturovaným databázím dle normy ISA-95, viz. Kap. 1.1, zajišťují možnost zpětného dohledání všech výrobních procesů a oblastí, kterými výrobek prošel a umožňují propojení zaznamenaných výrobních informací s obchodními procesy.

9.1 Spolehlivost systému

Spolehlivost systému je schopnost systému plnit dané funkce v daném časovém intervalu při zachování hodnot stanovených provozních ukazatelů v daných mezích.

Pravděpodobnost bezporuchového provozu 24/7 zařízení nebo systému vyjádřena v čase je patrná v Tab. 9.1:

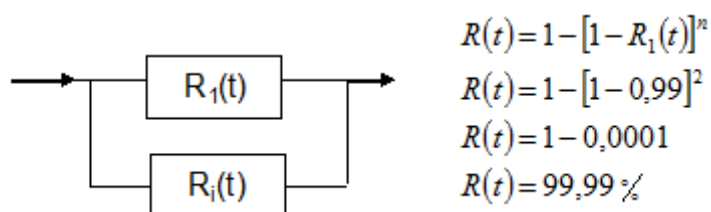
Tab. 9.1 Spolehlivost provozu vyjádřená v procentech a odpovídající výpadek v roce (365 dní)

Zaručená funkčnost [%]	Max. roční funkční prostoj
99,999	5 minut
99,995	26 minut
99,990	53 minut
99,950	4 hodiny
99,900	9 hodin
99,000	88 hodin

Redundance je v daném kontextu procesního systému forma zálohování kritických součástí, které by svým selháním mohly ovlivnit stabilitu či spolehlivost celého řízeného systému. Počet záloh jednotlivých komponent zvyšuje spolehlivost systému.

- Hardwarová redundance (tzv. strukturální zálohování, studená záloha – *cold backup*, horká záloha – *hot backup*) - využívá nadbytečného technického vybavení ke zvyšování spolehlivosti.
- Softwarová redundance se vyznačuje použitím tzv. algoritmické nadbytečnosti a nadbytečnosti softwarového vybavení. SW se podílí na obnově funkčnosti systému.
- Časová redundance systému přiděluje delší dobu, než je nezbytné na provedení výpočtu, algoritmu, příkazu.

Příkladem hardwarové redundance je paralelní zapojení dvou zařízení. Dvě zařízení, jejichž každého pravděpodobnost bezporuchového provozu je 99 % zvýší svým paralelním řazením zálohovaného a záložního prvku bezporuchový provoz na 99,99%, což představuje snížení maximálního funkčního prostoje z 88 hodin/rok na 53 minut/rok, viz. Obrázek 9.1.



Obrázek 9.1 – Redundance $R(t)$ systému, celková spolehlivost paralelní funkcionality dvou jeho částí vyjádřená procentem ve „čtyřech devítkách“

Spolehlivost systému je důležitá zejména z ekonomického hlediska a dále z hlediska bezpečnosti.



Shrnutí pojmů 9.1.

Spolehlivost systému je schopnost systému plnit dané funkce v daném časovém intervalu při zachování hodnot stanovených provozních ukazatelů v daných mezích.

Redundance je forma zálohování kritických součástí, které by svým selháním mohly ovlivnit stabilitu či spolehlivost celého řízeného systému a může se řešit z hlediska zajištění času, softwaru nebo hardwaru.



Otázky 9.1.

Co je spolehlivost systému?

Co je redundance systému?

9.2 Bezpečnost přístupu k aplikaci

Pojem bezpečnost vždy uznávaly všechny významné veřejné i soukromé instituce. Skládal se v podstatě ze tří složek. Jednalo se o:

- bezpečnost dat, která má za úkol zajistit především ochranu dat před neautorizovaným přístupem,
- zálohování dat, zabývající se ochranou dat před jejich ztrátou a
- dostupnost dat popisující požadavek na bezpečnost dat s ohledem na jejich spolehlivost.

9.2.1 Přihlašování uživatelů a jejich identifikace, autentizace a autorizace

Obzvláště velký důraz je v dnešní době kladen na zajištění bezpečnosti při identifikaci uživatelů přihlašujících se do operačního systému počítače nebo přímo do aplikace. Bezpečnost je jednou ze základních lidských potřeb, ale jde také o důležité téma, které bezprostředně souvisí i s problematikou týkající se průmyslových aplikací. Se stále rostoucími nároky na bezpečnost vyvíjených aplikací se zvyšují i požadavky na software pro jejich zajištění. Podpora nejmodernějších informačních technologií, otevřenost a snadnost používání je pro systémy používané v průmyslové automatizaci stále velmi důležitá.

V praxi rozlišujeme několik situací, s nimiž jsou spojeny tyto pojmy:

- Identifikace je zjištění identity uživatele, např. zadáním uživatelského jména a hesla.
- Autentifikace, autentikace nebo také autentizace, podle toho, z jakého jazyka překládáme, jestli z francouzštiny *authentification*, angličtiny *authentication* nebo němčiny *Authentifizierung*, je ověření toho, zda je uživatel skutečně tou osobou, za kterou se vydává např. použitím uživatelského jména a hesla. Ověřování existuje tedy na základě předchozího přihlášení a vyžaduje dodatečné informace. Je to proces ověření identity subjektu. Proběhne-li proces autentizace, dojde k autorizaci (souhlas, schválení, umožnění přístupu, atp.).

- Autorizace je tedy zjištění, zda má uživatel právo provést určitou akci, získat data či využít službu apod. Autorizace je udělována na základě předem definovaných kompetencí a kvalifikace uživatele.

9.2.2 Způsob identifikace uživatelů v operačních systémech počítačů

Nejdéle a nejčastěji používaným způsobem pro identifikaci uživatelů v operačních systémech počítačů jsou uživatelská jména a hesla. Uživatel se při přihlašování do systému identifikuje svým uživatelským jménem a teprve po zadání správného hesla mu je umožněna práce v systému.

Jednou z velkých nevýhod tohoto způsobu je možnost, že uživatel napíše heslo na místo svého uživatelského jména. V tomto případě se znaky hesla objeví přímo na obrazovce a ve většině případů budou zaprotokolovány do souboru systémových událostí či chyb jako pokus o přihlášení neznámého uživatele. Poté je pro osobu, která má přístup k systémovým datům jednoduché z tohoto souboru hesla přečíst. Mezi zdokumentované případy patří, kdy se v zachycených heslech objevovalo křestní jméno uživatele, hovorová podoba křestního jména, zřetězení iniciál uživatele s rokem narození, první čtyři znaky příjmení, samotný název aplikace, označení pracoviště, příjmení a křestní jméno napsané pozpátku. U většiny běžných uživatelů je možné odhalit jejich heslo po ne více než deseti pokusech o přihlášení, i když hesla uživatelů mohou v tomto případě obsahovat libovolnou kombinaci číslic a velkých písmen anglické abecedy v délce 4 až 8 znaků, nikoli však uživatelské jméno. K dalším nevýhodám patří pomalost identifikace při překlepech a změnách klávesnice z anglické na českou a naopak pro zadávání jména a hesla a také nutnost pamatovat si heslo zejména při časté vynucené změně hesla systémem nebo administrátorem systému, kdy dochází ke kontraproduktivě takovýchto zabezpečení a hesla si nakonec uživatelé píšou na štítky a lepí na monitory nebo klávesnice.

9.2.3 Přihlašování uživatelů do aplikace procesního systému

Přihlášením do aplikace jsou přidělena přístupová práva k aplikaci. S těmito právy jsou konfigurovány např. viditelnosti objektů aplikace pro různé skupiny uživatelů. Cílem je zviditelnit data na správném místě té správné osobě ve správný čas. Podle toho jsou také konfigurovány skupiny uživatelů.



Obrázek 9.2 – Návrh obrazovky pro přihlášení do vizualizační aplikace k zajištění bezpečného přístupu



Shrnutí pojmů 9.2.

Identifikace je zjištění identity uživatele.

Autentifikace nebo autentizace je ověření toho, zda je uživatel skutečně tou osobou, za kterou se vydává.

Autorizace je zjištění, zda má uživatel právo provést určitou akci, získat data či využít službu apod.

Nejdéle a nejčastěji používaným způsobem pro identifikaci uživatelů jsou uživatelská jména a hesla.

Přihlášením do aplikace jsou uživatelům přidělena přístupová práva k aplikaci a s těmito právy jsou pak konfigurovány vlastnosti objektů v této aplikaci.



Otázky 9.2.

Co je identifikace uživatele?

Jaké nevýhody způsob identifikace pomocí jména a hesla uživatele přináší?

Co je autorizace uživatele?



Úlohy k řešení

Příklad 9.1. Definujte alespoň dva uživatele s různými kompetencemi a právy přístupu

Příklad 9.2. Použijte vazbu na viditelnost objektu a práva přístupu pro přihlášeného uživatele.



Další zdroje

Boyer, S.A. (1993). SCADA: Supervisory Control and Data Acquisition. Instrument Society of America, p.239.



CD-ROM

Text, který vám byl předložen je doplněn těmito soubory animací:

Proměnné a animace objektů

Práce s okny v prostředí InTouch

Kreslení objektů v prostředí InTouch Window Maker

Práce s Aplikačním managerem

Reálný trend

Historický trend

Nastavení alarmu