

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

SCADA SYSTÉM STANIC PRO KONTROLU TĚSNOSTI

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Rak

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření** Ústav automatizace a měřicí techniky

Student: Bc. Jan Rak

ID: 147645

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

SCADA systém stanic pro kontrolu těsnosti

POKYNY PRO VYPRACOVÁNÍ:

1. Popište základní principy a požadavky na finální kontrolu kvality těsnosti ve výrobě.
2. Proveďte rešerši a diskutujte různá vývojová prostředí pro realizaci SCADA systémů.
3. Navrhněte datový model pro sběr dat z jedné nebo více stanic pro měření těsnosti.
4. Srovnajte různé databázové systémy a vyberte řešení pro danou aplikaci.
5. Realizujte databázi pro měření těsnosti.
6. Realizujte příslušnou SCADA aplikaci pro více stanic.
7. Ověřte a vyhodnoťte své řešení.

DOPORUČENÁ LITERATURA:

HERNANDEZ, Michael J. Návrh databází. Přeložil Jan Bouda. 1. vyd. Praha: Grada, 2006. 408s. ISBN 80-24-0900-7

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání: 6.2.2017

Termín odevzdání: 15.5.2017

Vedoucí práce: Ing. Radek Štohl, Ph.D.

Konzultant: Ing. Povolný Petr

doc. Ing. Václav Jirsík,
CSc. předseda
oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Cílem této práce bylo vytvořit SCADA/HMI systém stanic pro kontrolu těsnosti. Tento software bude schopen komunikovat se stanicí přes OPC server, pomocí něhož bude stanice řídit a ukládat naměřená data. Za tímto účelem bylo použito více vláknového zpracování dat, jak pro samotnou komunikaci, tak i pro chod programu a práce s databází na ukládání výsledků. Výsledný program bude umístěn na osobní PC, jenž je součástí stanice pro kontrolu těsnosti.

Klíčová slova

SCADA, HMI, OPC server, Databáze, WPF, XAML, S7 .NET, C#

Abstract

The goal of this thesis is to develop SCADA/HMI system for stations on leak test. The software will be able to communicate with the station via the OPC server, by which the control station and store measured data. For this purpose, it was used more fiber processing, as for the actual communication, as well as to run the program and work with databases to store the results. The resulting program will be placed on the personal computer, which is part of the station to check for leaks.

Keywords

SCADA, HMI, OPC server, Database, WPF, XAML, S7 .NET, C#

Bibliografická citace

Rak, J. *SCADA systém stanic pro kontrolu těsnosti*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2017. 49s. Vedoucí semestrální práce byl Ing. Radek Štohl.

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma: SCADA systém stanic pro kontrolu těsnosti, jsem vypracoval samostatně pod odborným vedením a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 15. Května 2017

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Radku Štohlovi a konzultantovi diplomové práce panu Ing. Petru Povolnému z firmy Labtech s.r.o. za účinnou metodickou, a odbornou pomoc, trpělivost a další cenné rady při zpracování mé diplomové práce.

V Brně dne 15. Května 2017

.....
podpis autora

Obsah

1	Úvod.....	8
2	Základy teorie testování těsnosti	9
2.1	Důvody testování těsnosti výrobků pomocí helia.....	9
2.2	Technologie detekce netěsností pomocí helia	9
2.3	Metody testování netěsností	10
2.3.1	Vakuová metoda testování (zvenku dovnitř).....	10
2.3.2	Lokalizace netěsnosti	10
2.3.3	Měření netěsností	11
2.3.4	Tlaková metoda testování (zevnitř ven)	11
2.3.5	Měření netěsností	11
2.3.6	Lokalizace netěsnosti	12
2.3.7	Akumulační metoda testování	12
2.3.8	Testování netěsností systémů	12
2.3.9	Vakuové systémy	12
2.3.10	Tlakové systémy.....	12
2.3.11	Typické aplikace testování těsností	13
3	Vývojová prostředí SCADA systémů.....	13
3.1	Základní popis SCADA systému.....	13
3.2	Struktura SCADA systémů	14
3.3	Wonderware InTouch.....	15
3.4	Control WEB.....	16
3.5	Reliance.....	17
3.6	Promotic	19
3.7	WinCC flexible	20
3.8	Shrnutí	21
4	Databázové systémy	22
4.1	Oracle Database 12c.....	22
4.2	MySQL.....	23
4.3	Microsoft SQL server.....	23
4.4	PostgreSQL	23
4.5	SQLite	24
4.6	Porovnání DBMS	24
4.6.1	Základní údaje.....	24
4.6.2	Podpora operačních systémů	25
5	Datový model	25
5.1	Stanovení entit.....	25
5.1.1	Metody měření	26
5.2	Relace.....	26
5.3	Atributy	27
6	Technologie OPC	28
6.1	Přenos dat.....	28
6.2	Architektura OPC.....	30
6.3	Specifikace OPC	31
6.4	OPC UA	31
7	Realizace aplikace	33
7.1	Visual Studio.....	33
7.2	.NET Framework.....	33
7.3	WPF a C#.....	34
7.3.1	WPF.....	34
7.3.2	XAML	35
7.3.3	C#.....	35
7.4	Koncept aplikace	36
7.5	Obrazovka měření	37
7.6	Obrazovka servis	37
7.7	Obrazovka recept.....	38

7.8	Obrazovka parametry stanice	39
7.9	Obrazovka výsledky	39
7.10	Správa a přihlášení uživatelů.....	40
7.11	Komunikace s PLC.....	40
8	Realizace databázového systému.....	42
8.1	Implementace SQLite.....	42
8.2	Vytváření tabulek	42
9	Závěr.....	44
10	Seznam literatury.....	45
11	Seznam použitých zkratk.....	47

Seznam obrázků

Obr. 1	Princip magnetické separace.....	10
Obr. 2	Lokalizace netěsnosti.....	11
Obr. 3	Měření netěsnosti (zvenku dovnitř).....	11
Obr. 4	Měření netěsností (zevnitř ven).....	11
Obr. 5	Lokace netěsností (zevnitř ven).....	12
Obr. 6	Akumulace.....	12
Obr. 7	Základní funkce SCADA/HMI.....	14
Obr. 8	Struktura SCADA/HMI systému [1].....	15
Obr. 9	Vývojové prostředí Wonderware InTouch [2].....	16
Obr. 10	Control Web 7 [5].....	17
Obr. 11	Reliance 4 Design [6].....	19
Obr. 12	Domácí použití systému Promotic.....	20
Obr. 13	Přehled verzí prostředí WinCC [12].....	20
Obr. 14	Datový model pro stanici měření těsnosti.....	27
Obr. 15	Komunikace bez OPC [7].....	29
Obr. 16	Technologie OPC [7].....	29
Obr. 17	Architektura OPC [8].....	30
Obr. 18	OPC Universal Access [13].....	32
Obr. 19	Data binding princip [21].....	35
Obr. 20	Základní obrazovka aplikace.....	37
Obr. 21	Obrazovka manuálního ovládání stanice.....	38
Obr. 22	Obrazovka správy metod.....	38
Obr. 23	Obrazovka parametrů stanice.....	39
Obr. 24	Obrazovka výsledků.....	39
Obr. 25	Obrazovka přihlášení uživatelů.....	40

1 Úvod

Cílem diplomové práce je sestavení aplikace pro stanice na kontrolu těsnosti. Tato výsledná aplikace bude schopna komunikovat se stanicí přes průmyslový server, pomocí kterého bude stanici řídit a ukládat z ní data o měřených výrobcích.

Jedním z dílčích bodů práce je popsat finální kontrolu kvality těsnosti, tento bod je hlavně vztažen ke zkoušce heliem, která se právě na těchto stanicích provádí. Další část podává přehled o SCADA vizualizačních systémech, které se v průmyslu v dnešní době nejvíce používají.

Dále jsou zde rozebrány databázové systémy, jejich výhody a nevýhody. Z této rešerše vyplývá návrh datového modelu, do kterého výsledná aplikace bude ukládat data, a tento model je následně realizován v jazyce C#.

Následně je popsána technologie OPC, která je standardem v dnešním průmyslu a na které tato práce bude stát. V neposlední řadě jsou porovnány některé databázové systémy, s kterými se lze setkat na dnešním trhu a je vybráno nejschůdnější řešení.

Finálním bodem práce je samotná realizace příslušné aplikace a její realizace a ověření.

2 Základy teorie testování těsnosti

2.1 Důvody testování těsnosti výrobků pomocí helia

Z mnoha důvodů je helium nejpoužívanějším značkovacím plynem pro velmi přesné určování těsnosti výrobků a zařízení.

Helium je:

- netoxické
- inertní a nekondenzující
- v běžných podmínkách přítomné v atmosféře pouze ve stopových množstvích
- relativně levné
- díky malým rozměrům molekuly schopné proniknout velmi malými otvory (netěsnostmi)
- nehořlavé a nevýbušné

Jediná molekula, která je menší než helium (atomová hmotnost 4) je vodík (atomová hmotnost 2), který ale není inertní (nereaktivní). Helium má mnohem menší hmotnost než je následující inertní molekula vzácného plynu - neon (atomová hmotnost 20), který je také mnohem dražší.

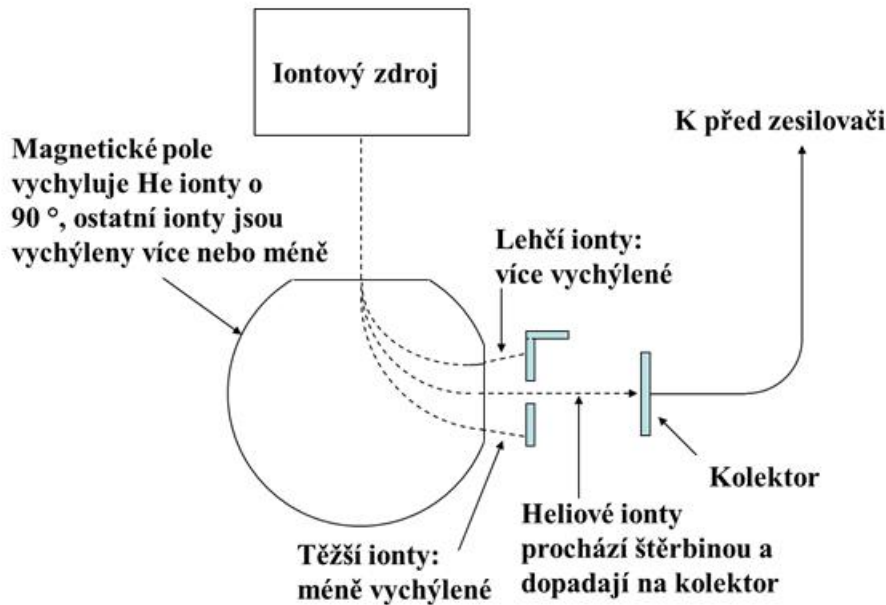
2.2 Technologie detekce netěsností pomocí helia

Detektor netěsností na principu hmotnostního spektrometru MSLD (Mass Spectrometer Leak Detector) je komplexní systém pro vyhledání a/nebo změření velikosti netěsností zařízení nebo nádob. Značkovací plyn, helium, je zavedeno do testované součástky, která je připojena k systému MSLD. Helium, které uniká ze součástky, se šíří systémem, v detekčním systému je měřen jeho parciální tlak, který je převeden na jednotky úniku. [10]

Princip MSLD spočívá v ionizaci plynů ve vakuu a jejich zrychlení v elektromagnetickém poli (Obr. 1). Ionty helia jsou odděleny a posbírány, výsledný iontový tok je zesílen a zobrazen na měřiči nebo na obrazovce.

Hmotnostní spektrometr se skládá z následujících součástí:

- Spektrometr, který zjišťuje množství helia
- Vakuový systém pro udržení odpovídajícího nízkého tlaku ve spektrometru
- Mechanická vývěva sloužící k odčerpání testovaného výrobku
- Ventily umožňující různá stadia cyklu detekce netěsností, od odčerpání přes testování až k vyvětrání
- Zesilovač a čtecí zařízení, které monitoruje výstupní signál spektrometru
- Zdroj napětí a ovládání, které řídí ventily, ochranné okruhy atd.
- Přípravky, pomocí kterých se připevňují testované součástky k zařízení pro detekci netěsností



Obr. 1 Princip magnetické separace

2.3 Metody testování netěsností

Helium jako značkovací plyn nachází uplatnění v mnoha aplikacích. Metoda testování se zpravidla volí podle aktuálních pracovních podmínek. Doporučuje se, aby byl během testování zachován v testované součástce stejný rozdíl tlaku a ve stejném směru jako během skutečného používání. To znamená, že vakuový systém je testovaný s vakuem uvnitř komory zatímco tlaková nádoba na plyn by měla být testována pod vysokým tlakem.

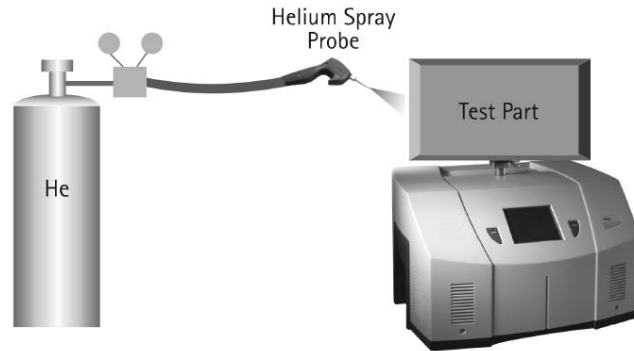
V praxi se využívají dva základní přístupy v testování netěsností. První je lokalizace netěsnosti a druhý je měření celkové rychlosti úniku ze součástky, protože jistá netěsnost může být akceptovatelná. Také je možné součástky nejprve testovat na přítomnost akceptovatelné netěsnosti, a pokud výrobek testem neprojde, je podroben lokalizaci netěsnosti. Mnoho součástek je možné testovat v sadách. Pokud sada neprojde, pak jsou testovány jednotlivé součástky, a zjistí se, které z nich jsou vadné. [10]

2.3.1 Vakuová metoda testování (zvenku dovnitř)

V testované součástce je vytvořeno vakuum pomocí zvláštního velkoobjemového čerpacího systému, nebo jen pomocí vlastního detektoru. Jakmile je dosaženo požadovaného vstupního tlaku, detektor se spojí se systémem pomocí ventilu nebo je přemístěn do testu a testovaná součástka je testována jednou z následujících metod:

2.3.2 Lokalizace netěsnosti

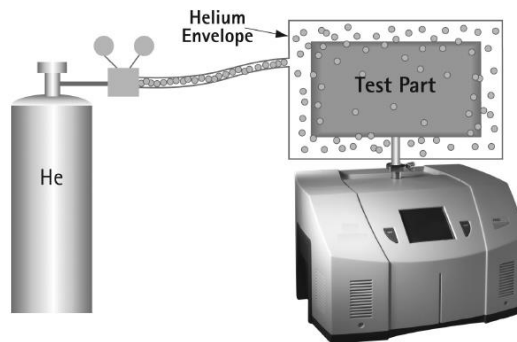
Zjistit polohu netěsnosti (ale ne celkový průtok – její velikost) je možné pomocí ofukovací sondy. Helium je přivedeno k předpokládané poloze netěsnosti a výrobek je „ofukován“ sondou s nastavitelným průtokem (Obr. 2).



Obr. 2 Lokalizace netěsnosti

2.3.3 Měření netěsností

Pro zjištění celkového úniku (ale ne počtu nebo poloh netěsností) je součástka připojena na detektor netěsností a obklopena prostředím helia. Prostředí může být tvořeno různými způsoby – od obyčejného plastového sáčku až po komplexnější uspořádání zvonovitých nádob (Obr. 3)



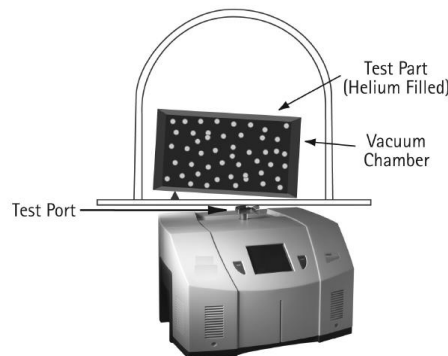
Obr. 3 Měření netěsnosti (zvenku dovnitř)

2.3.4 Tlaková metoda testování (zvenitř ven)

Testovaná součástka je naplněna heliem nebo směsí helia a vzduchu. Netěsnost je detekována jednou z následujících metod:

2.3.5 Měření netěsností

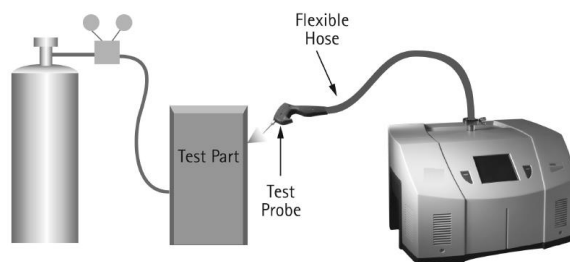
Pro zjištění celkového úniku (ale ne počtu nebo poloh netěsností) je součástka naplněna heliem (nebo heliem a vzduchem/dusíkem). Naplněná součástka je pak umístěna do komory, která je evakuována pomocí detektoru. Helium, které unikne ze součástky, je detektorem změřeno a vyhodnoceno (Obr. 4)



Obr. 4 Měření netěsností (zvenitř ven)

2.3.6 Lokalizace netěsnosti

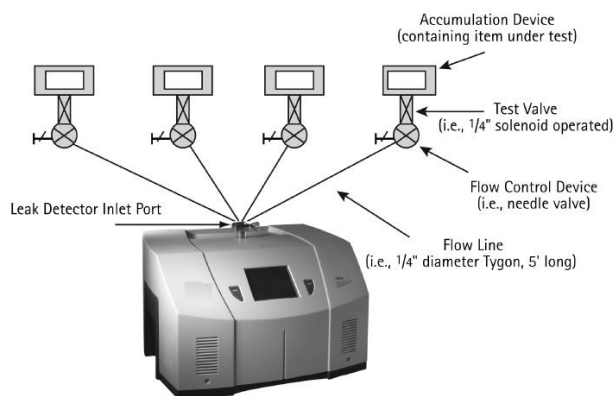
Zjistit polohu netěsnosti (ale ne celkový průtok – její velikost) je možné pomocí čichací sondy, která je připojena na detektor (Obr. 5)



Obr. 5 Lokace netěsnosti (zevnitř ven)

2.3.7 Akumulační metoda testování

Touto metodou je možné zjistit polohu i velikost úniku. Kolem předpokládané polohy úniku je umístěn kryt, který utěsní jeho okolí. Je stanovena určitá doba, která je ponechána, aby helium mohlo uniknout ze součástky a naplnit tak kryt utěsněné oblasti. K ní je pak připojen pomocí ventilu detektor. Je-li potřeba testovat více součástí nebo je předpokládán více násobný výskyt netěsností, mohou být oblasti připojovány následně po sobě (Obr. 6)



Obr. 6 Akumulace

2.3.8 Testování netěsností systémů

Systémy by stejně jako výrobky měly být testovány se stejným rozdílem tlaku, jaký bude v systému bude při skutečném používání. Vakuové systémy by měly být testovány ve vakuu, kdežto systémy, které pracují pod tlakem, by měly být testovány pod tímto tlakem.

2.3.9 Vakuové systémy

Obecně jsou vakuové systémy testovány pomocí přenosného detektoru netěsností. Ten je připojen pomocí t-kusu mezi vstupní potrubí pumpy pro vysoké vakuum a vstup jeho primární pumpy. Systém by měl být schopen udržet dostatečně nízký tlak na vstupním potrubí tak, aby zde mohl detektor pracovat. Helium je na místo předpokládaného úniku dopraveno ofukovací sondou. Pokud je netěsnost přítomna, helium se rychle začne systémem šířit a odpověď detektoru by neměla přijít později než za několik vteřin. Citlivost detektoru netěsností je snížena v systémech s velkými primárními pumpami.

2.3.10 Tlakové systémy

Mnoho tlakových systémů musí být těsných. Tyto systémy mohou být naplněny heliem nebo směsí helia a jiného plynu jako je třeba dusík. Je-li použita směs, signál helia je úměrně snížen. Například, je-li použita směs 10% helia a 90% dusíku, signál odpovídá

10% skutečné velikosti netěsnosti. Vzhledem k tomu, že většinou se hledají polohy netěsností spíše než jejich velikost, je to akceptovatelné. Jakmile je systém naplněn odpovídajícím množstvím helia, netěsnost je možné vyhledat pomocí čichací sondy nebo pomocí vytvoření krytu v okolí předpokládaného výskytu a detekci helia po naplnění oblasti na detekovatelnou úroveň. [10]

2.3.11 Typické aplikace testování těsností

Detekce a lokalizace úniků je kritická při výrobě mnoha produktů od jednotlivých součástek přes dílčí sestavy po kompletní systémy. V mnoha případech je nutné znát jak polohu, tak i velikost netěsnosti. Naše produkty poskytují jistotu v oblasti netěsností Vašich výrobků, ať už se jedná o testování kvalitativní nebo kvantitativní. Některé příklady využití:

Součástky a sestavy ve vakuu:

- Hermeticky uzavřené elektronické pakety
- Ventily a potrubí
- Těsnění přívodů a spojů sklo/kov

Součástky a sestavy pod tlakem:

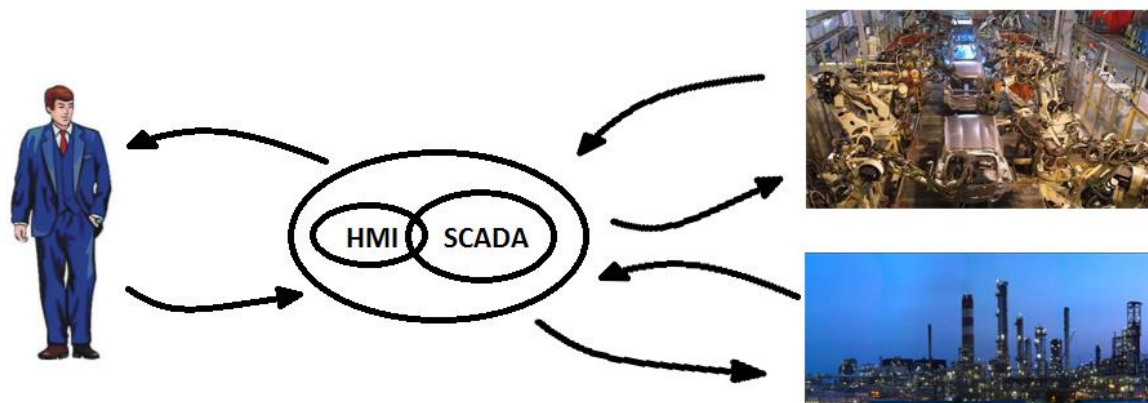
- Systémy klimatizace a chladicí okruhy
- Radiátory, tepelné výměníky a kondenzátory
- Vedení brzd, paliva a hydraulických kapalin
- Plynové tlakové nádoby
- Zásobníky potravin a obalové materiály
- Zdravotnická zařízení pro vnitro tělní implantace
- Potrubí vysoké čistoty

3 Vývojová prostředí SCADA systémů

3.1 Základní popis SCADA systému

Zkratka SCADA je z anglického názvu „Supervisory Control And Data Acquisition“, což můžeme volně přeložit jako „nadřazené řízení a sběr dat“. Spolu s těmito systémy se také často setkáváme se zkratkou HMI, neboli „Human Machine Interface“. V češtině by se to dalo říci jako rozhraní mezi člověkem a strojem, v praxi jsou takto často zastoupeny ovládací panely (v dnešní době průmyslové dotykové obrazovky) strojů, s vizualizační animací. Obecný pohled na tento systém je znázorněn na Obr. 7. Hlavním úkolem SCADA systémů je interaktivní ovládání technologie, sběr dat a jejich zpracování, ukládání a historizace.

Jedná se tedy zde o zdokonalení prezentace a uchovávání těchto dat. Dříve probíhala prezentace dat nejčastěji pomocí velkých panelů zabírajících jednu stěnu dispečerského pracoviště, skládající se z kontrolek stavů a analogových měřáků. Tyto moderní systémy zabírají pouze několik obrazovek počítače a využívá se zde především možnosti skrytí méně důležitých dat, které se zobrazují až v případě potřeby. Kritická data, jako alarmové stavy, mohou být zviditelněny tak, aby nešly přehlédnout nejčastěji v podobě pop-up oken.



Obr. 7 Základní funkce SCADA/HMI

Tyto systémy nejsou určeny k zastoupení řídicího systému, tuto funkci plní spíše PLC automaty. Nespornou výhodou těchto systémů je možnost vizualizace na libovolném PC, chytrém telefonu, tabletu nebo jinou platformou, touto možností disponuje většina moderních SCADA systémů. Hlavním cílem je zde umožnit centrální řízení a mít plnou kontrolu nad malými i velmi rozsáhlými technologiemi. Jedná se především o technologie jako jednoúčelové stroje, malé vodní elektrárny, výrobní linky až po komplexní řešení výrobních závodů jako jsou například automobilky. Využití tedy zde nalezneme napříč celým průmyslem. V současné době je na našem trhu dostupná celá řada programových systémů SCADA/HMI, které podporují obě hlavní funkce, tj. řízení technologických procesů a vizualizaci. Většinou se jedná o heterogenní otevřené systémy (InTouch, HP VEE, Citect) pracující v operačním systému UNIX, Windows.

Vyvíjené systémy lze dělit do dvou základních skupin, na uzavřené a otevřené. Uzavřené systémy jsou vyvíjeny výrobcem PLC, pouze pro jejich vlastní výrobky jako je např. Simatic WinCC, nebo Factory Talk View Studio. Otevřené systémy se naopak snaží o kompatibilitu s co největším množstvím výrobků a typů zařízení jako jsou např. Realiance 4, Control Web, Wonderware InTouch.

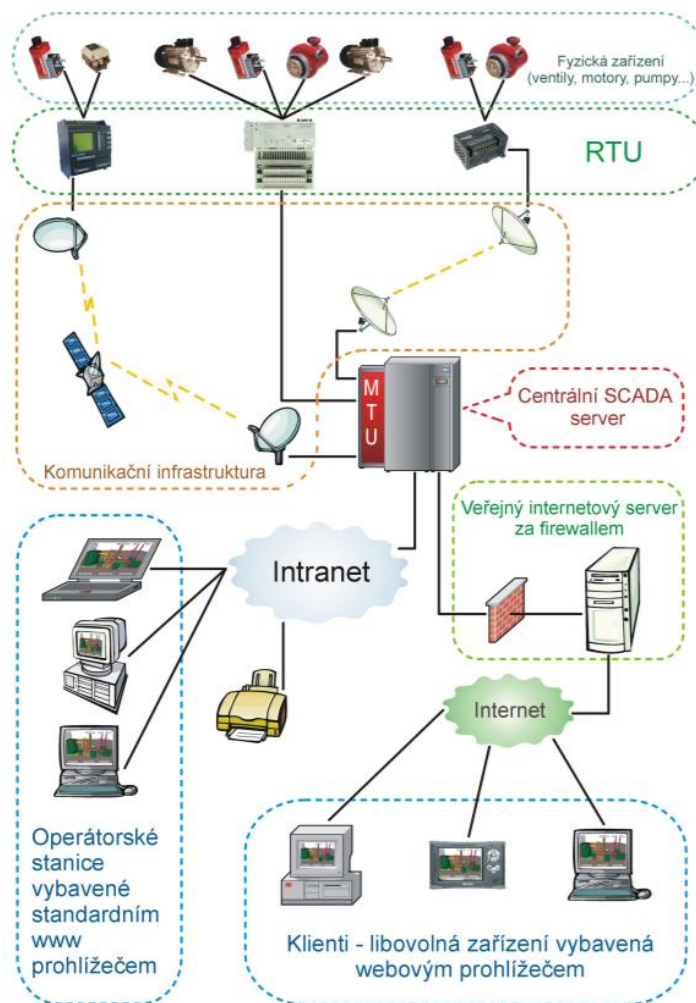
3.2 Struktura SCADA systémů

Základními komponenty každého SCADA systému jsou:

- RTU (Remote Terminal Unit)
- MTU (Main Terminal Unit)
- Centrální řízení se serverem (velín)
- Komunikační infrastruktura

RTU zařízení propojují fyzickou vrstvu technologie (motory, ventily, pumpy) s MTU. Tyto jednotky jsou realizovány jako mikropočítače nebo PLC. Tato vrstva řeší jednodušší dílčí úkony procesu a to bez zásahu vyšších vrstev, autonomnost těchto procesů přidává celému systému na modulárnosti a rozšiřitelnosti.

MTU komunikuje se všemi RTU, centrálním serverem a HMI. Data z fyzické vrstvy se zde zpracovávají, popřípadě předají dalším systémům nebo se zobrazí v HMI. Ve většině případů se jedná o PC, který přes komunikační vrstvu získává data a rovnou je zpracovává a zobrazuje. V jedné jednotce je tedy zastoupeno jak HMI tak i třeba OPC server.



Obr. 8 Struktura SCADA/HMI systému [1]

3.3 Wonderware InTouch

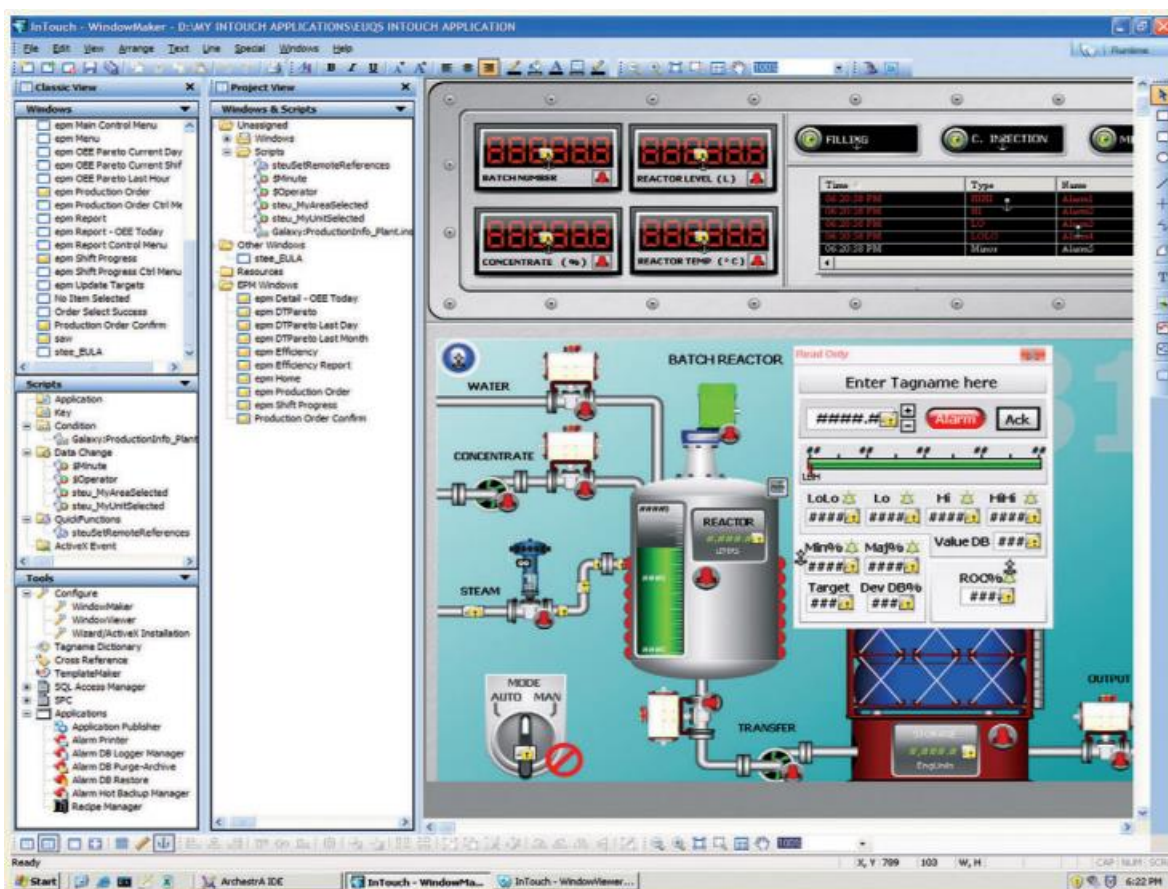
Jeho přední vlastností je zejména robustní a značná flexibilita v podobě možnosti doprogramování funkcí v prostředí .NET. Jeho převážné využití je v rozsáhlých a komplexních procesech avšak jeho použití je možné i na obyčejné aplikace. InTouch umožňuje rychle a snadno vyvíjet grafické zobrazení real-time průmyslových procesů. Podle společnosti Wonderware se systém InTouch používá ve třetině světových továren a téměř ve všech průmyslových sektorech.

Grafická flexibilita a jednoduchost vytváření animací je důvodem proč se tento systém stal tak populární mezi uživateli. Pro grafické rozhraní InTouch poskytuje komplexní řadu grafických primitiv (obdélník, čára, kruh) ze kterých mohou být vytvořeny grafické symboly, dále umožňuje vytvářet i složitější tvary, jako jsou křivky, kruhové nebo obloukové výseče, úseče, a běžné ovládací prvky Microsoft Windows (text box, combo box aj.), jedná se o vlastnost vytvářet složité tvary kombinováním jednodušších tvarů dohromady.

Pokročilá manipulace s grafickými atributy umožňuje mimořádně výkonné a přesné ovládání, pomocí vestavěných animací nebo skriptování. Těmito atributy se dá ovládat velká škála zobrazení těchto grafických prvků, jako je průhlednost, hladké stínování, styl výplně, styl čáry, orientace, velikost a poloha. To umožňuje návrhářům lehké vytváření jejich animací s potřebnou funkčností a efektivitou. To je podpořeno i možností kombinovat vektorovou i bitmapovou grafiku.

Další předností tohoto prostředí je rozsáhlá knihovna s již vytvořenými grafickými prvky (čerpadla, ventily, tlačítka, displeje, atd.), které jsou velmi dobře propracovány (více než 500 symbolů).[2] Tuto knihovnu lze editovat a tak přizpůsobit na požadavky vlastního použití. Důležitou vlastností, která urychluje vytváření animací v prostředí je opakovatelnost a dědičnost jednotlivých projektů tak i objektů, jde tedy lehce rozkopírovávat dílčí bloky použité vícekrát. Nejpodstatnější výhodou je však tato dědičnost mezi verzemi systému.

K sběru dat a komunikaci jsou zde k dispozici I/O servery, přímo od Wonderware nebo od nezávislých firem. Stejně tak se zde dají použít ke komunikaci OPC servery od různých výrobců. Pro práci s databázemi InTouch podporuje standardy, jako jsou ActiveX, ADO/ODBC, .NET. Existují i rozšiřitelné moduly pro standardy SQL Access, SPC a Recip Manager. [4]



Obr. 9 Vývojové prostředí Wonderware InTouch [2]

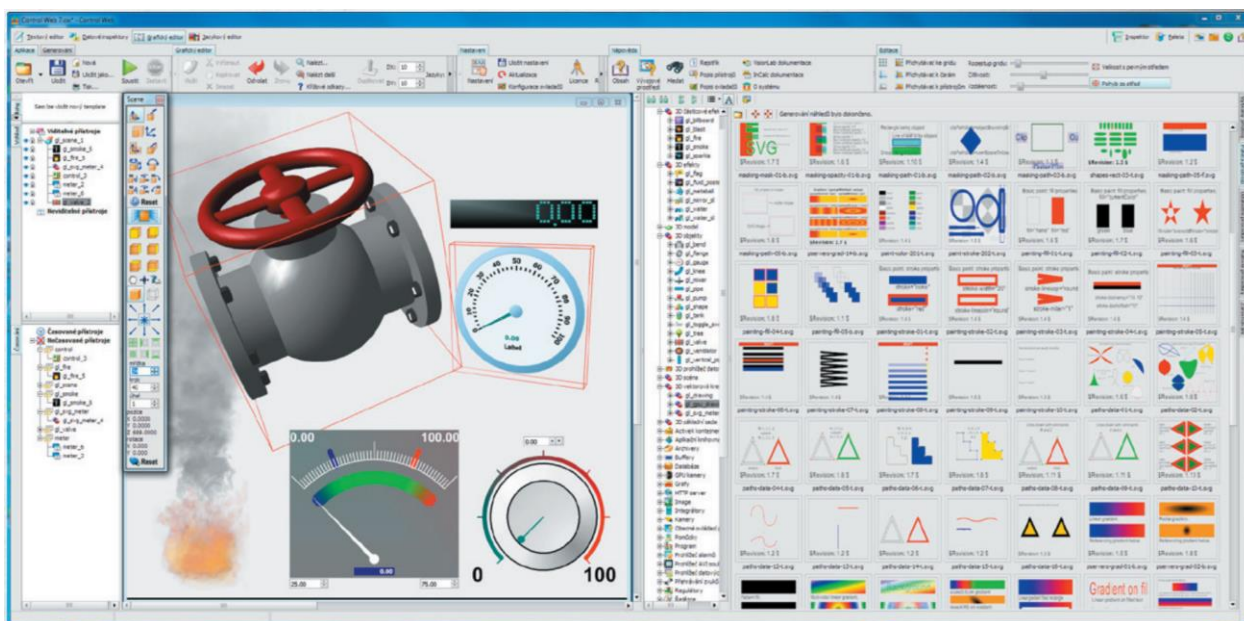
3.4 Control WEB

Control Web je univerzální nástroj pro vývoj vizualizačních a řídicích aplikací, pro sběr, vyhodnocování a ukládání dat. Objektově komponentová architektura aplikace zajišťuje systému Control Web široký rozsah nasazení od časově nenáročných aplikací až po aplikace reálného času. Hlavní cíl systému je určení běžných i komplikovaných úkolů. Samozřejmostí je respektování všech existujících standardů pro běh programu a jejich uživatelské rozhraní, přístup k databázím a výměna dat, spolu s komunikací po počítačových sítích a spolupráce s hardwarem pro sběr dat a řízení. Control Web je systém vyvíjený společností Moravské přístroje.

Control Web koncepčně vychází z osvědčené architektury jeho předchůdců, kterými jsou Control Panel a Control Web 2000. Tyto systémy byly na svoji dobu vybaveny kvalitním grafickým uživatelským rozhraním pracujícím s libovolně složitou hierarchickou strukturou oken na obrazovce. O kvalitě těchto systémů také vypovídá fakt, že aplikace použité v nepřetržitých provozech pracovali i několik let bez odstávky a stále si udrželi svoji stabilitu a výkon. Je však nutno zdůraznit že tyto aplikace pracovali na starších systémech Windows.

Vývojové prostředí nabízí grafickou nebo textovou možnost tvorby aplikace. Grafická metoda je založena na funkci drag-and-drop (táhni a pusť). Tato forma tvorby aplikace je výhodná kvůli toleranci vůči chybě při psaní textu, vývojové prostředí automaticky doplňuje všechny potřebné textové kódy. Textový editor je však také důležitý. Mnohdy je nezbytné jej použít a určité příkazy dopsat ručně. V praxi se používají oba editory.

Možnost archivace dat je dnes již běžným standardem a součástí vizualizačních systémů. Jedná se především o uchování měřených nebo vypočtených hodnot, alarmů a dalších dat vzniklých za běhu aplikace. Tyto činnosti jsou prováděny různými způsoby například pomocí virtuálního přístroje, atributem datového elementu nebo datovou sekci. Data je možné archivovat periodicky, voláním procedury nebo při vyvolání určité události. Archivovaná data se defaultně ukládají do formátu Microsoft Access. [3]



Obr. 10 Control Web 7 [5]

3.5 Reliance

Reliance je systém vyvíjený společností GEOVAP s.r.o. [6], a to již od roku 1997, jde o nástupce systému EP-DRAW. Jedná se o celosvětově rozšířený systém, o němž výrobce tvrdí, že podle vývojářů aplikací, kteří přešli na Reliance z jiných SCADA/HMI systémů je vývoj jejich aplikací 2-3 krát rychlejší, než se systémy konkurenčních firem. Výrobce si dále zakládá na jeho jednoduché přehlednosti, intuitivnosti a přátelském vývojovém prostředí. Tento systém je hlavně určen pro monitorování a ovládání průmyslových technologií ale i pro automatizaci inteligentních budov. Je tedy připraven na použití jak na malých projektech, tak i na rozsáhlých průmyslových aplikacích.

Reliance runtime moduly zajišťují nepřetržitý běh vizualizační aplikace na koncovém počítači klienta. Tyto moduly umožňují získávat data z komunikačních driverů, tzn. OPC a DDE serverů, dále mohou získávat data a alarmy z jiných runtime modulů, zpracovávat a generovat alarmy, to samé mohou udělat s jejich archivací a poskytováním těchto dat po síti jiným runtime modulům. Při práci s proměnnými se data dají navázat pomocí automatického mapování přímo z OPC serveru, není-li to však možné je nutno zadat OPC ItemID, což je adresa proměnné která je zpřístupněna přes OPC. Reliance taktéž podporuje komunikaci přes protokol TCP/IP, zde je nutné nastavit komunikační parametry jako IP adresa, název počítače v síti atd..

Poslední aktuální verze Reliance 4 je softwarovým balíčkem několika následujících programů umožňující plnou realizaci SCADA/HMI systému

- **Reliance 4 Server** – bez grafického rozhraní, určený pro sběr dat, které jsou dále zpřístupněny runtime modulům
- **Reliance 4 Control Server** – stejné služby jako u předchozího, plus obsahuje grafické rozhraní uživatele
- **Reliance 4 Control** – klientské zobrazení vizualizace, umožňuje náhled na všechny potřebná data, alarmy atd., dále je zde ovládání dané technologie plus detekce poruch
- **Reliance 4 View** – stejné možnosti jako u předchozího až na ovládání technologie, program je určen pouze pro sledování procesu
- **Reliance 4 Design** – Vývojové prostředí pro vývoj aplikací
- **Reliance 4 OPC server** – server pro komunikaci s PLC nebo jinou SCADA aplikací.
- **Reliance 4 OPC server monitor client** – umožní připojení k OPC a zobrazení jeho údajů

Reliance systém je vydáván ve dvou vývojových a třech runtime verzích. Vývojovými verzemi jsou Enterprises a Desktop. Desktop verze umožňuje tvorbu aplikací a správu, nastavení projektu. Zde se například nastavují proměnné (aliasy), které budou přistupovat ke klientovi přes OPC. Zde vytvořené aplikace propojují jeden PC s libovolným množstvím vstupně/výstupních HW zařízení, nebo s PLC automaty. Síťové aplikace nebo aplikace určené pro OPC servery zde však nevytvoříme. K tomuto účelu slouží verze Enterprises, která má stejné funkce jako Desktop verze, ale obsahuje navíc export aplikací do formátu přístupného OPC klientům. Jedná se zde tedy o programy jako Reliance 4 Web Client a Mobile Client. Výsledné aplikace tedy propojují libovolné množství PLC tak i PC a zároveň ještě umožňuje předávání dat tenkým klientům. Tenci klienti jsou např. webové rozhraní na PC nebo na chytrém telefonu.

Úplná verze systému je přístupná pouze s licenčním klíčem. Existuje i trial verze, ta je však omezena na tzv. 25 datových bodů, což jsou v podstatě proměnné. Celá licenční politika kolem tohoto systému se týká právě těchto datových bodů. S jejím nárůstem logicky narůstá i cena. Krom datových bodů se i rozlišuje počet tenkých klientů, kteří se budou moci připojit na server. Licenční klíč je poskytován ve dvou formách, buď jako HW nebo SW. HW klíč je možné objednat jako LPT nebo jako USB, jeho nespornou výhodou je přenositelnost mezi počítači, to může být zvláště užitečné, je-li potřeba vyvíjet aplikaci na více počítačích. [6]



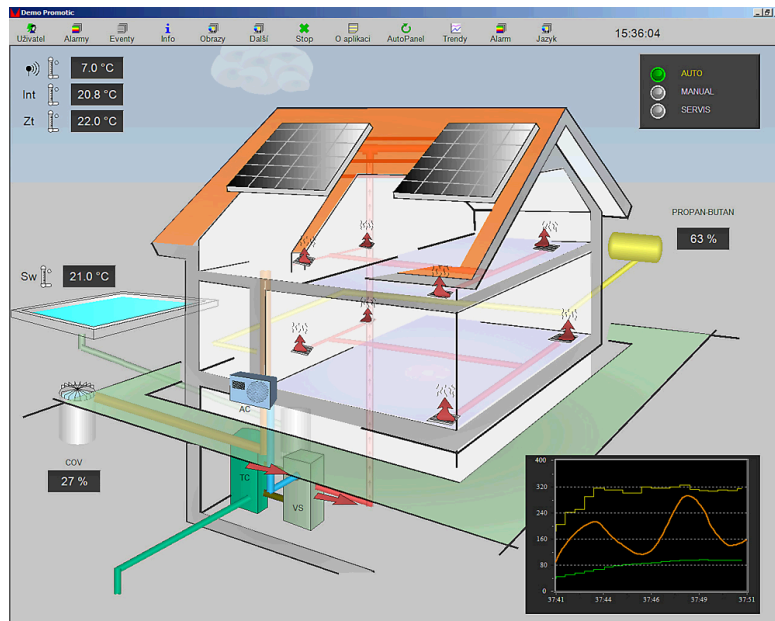
Obr. 11 Reliance 4 Design [6]

3.6 Promotic

Systém Promotic je vizualizační software vyvíjený firmou Microsys, jedná se o českou firmu se sídlem v Ostravě. Aktuální verze programu je dostupná v osmé verzi Promotic 8, jehož vývoj začal roku 2009. S poslední verzí tohoto softwaru se zároveň začala poskytovat freeware vývojové prostředí PmDevFree. Tato freeware verze umožňuje provozovat a vyvíjet aplikace zdarma. Výhodou také je, že jsou zde přístupny všechny vývojové prvky jako u komerčního balíčku, je zde pouze omezení velikosti aplikace a maximálního počtu proměnných a to na 100. PmFree je spíše výrobcem doporučována pro tvorbu aplikací na domácí užití a pro menší průmyslové aplikace. Pro větší aplikace je zapotřebí zakoupit licenci PmDevelop. Real time licenci PmRtFree je také možné využívat bezplatně a bez časového omezení, ale je zde opět zaveden limit 30 proměnných.

Základním nástrojem pro tvorbu je editor umožňující definování stromové struktury a algoritmů využívající jazyka Visual Basic Script, používaný pro zápis uživatelských algoritmů, přístupu k vlastnostem a metodám objektů nebo jiných aplikací. Dalším z podstatných nástrojů programu Promotic, je editor obrázků sloužící pro grafický návrh aplikace. Systém také podporuje databázová rozhraní SQL a ODBC, spolu s tím lze ukládat data do formátů MySQL, Oracle, FireBird aj.. Pro komunikaci s ostatním systémovým softwarem jsou zde zabudována rozhraní OPC, DDE, XML, TCP/IP, AcitveX. Všechna tato rozhraní jsou kompatibilní se softwarem jiných firem a tak možnosti komunikace s okolním softwarovým systémem jsou téměř neomezené. Další komunikační částí systému je podpora technologií intranet/internet, správa uživatelů a přihlašovací systém. Pro internetové rozhraní je systém vybaven automatickým generováním HTML stránek. Výhodou celého systému je jeho zabezpečení a široká škála obrázků vytvořena ve vektorové grafice.

Zajímavou možností, kterou systém Promotic nabízí je možnost vzdáleného ladění aplikace přes internet. Pro tuto možnost je však nutné použít programy Promotic Web nebo LapLink. U grafických prvků navržených projektantem aplikace dále nechybí možnost upravovat jejich statické vlastnosti nebo propojení jednotlivých prvků datovou vazbou a tím tak oživit vytvářenou vizualizaci. Přes tyto datové vazby lze napojit nejrůznější vlastnosti prvků jako je např. barva, velikost, hodnota aj..

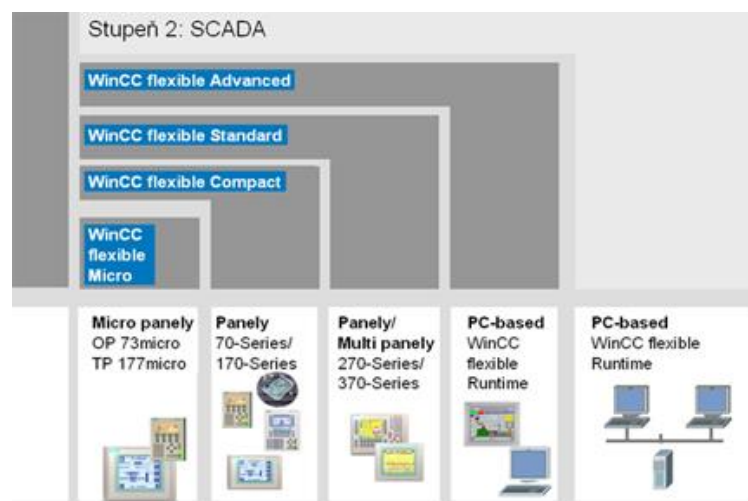


Obr. 12 Domácí použití systému Promotic

3.7 WinCC flexible

Jedním z významných systémů pro tvorbu SCADA aplikací je WinCC flexible od společnosti Siemens. Tento software je poměrně dost univerzální jelikož jeho použití spočívá hlavně ve vývoji animací pro panely HMI. V základu je možné tento systém pořídit ve čtyřech verzích jak je naznačeno na Obr. 13. Výhodou tohoto softwaru je zajištění snadné konektivity s výrobky od společnosti Siemens, kde dochází k propojení s programem v PLC téměř zcela automaticky, bez zbytečného nastavování a ladění. Avšak díky jeho univerzální konektivitě přes standardy OPC, DDE a OLE, jej lze propojit se zařízeními od jiných výrobců. WinCC je postaveno na databázovém systému MS SQL server, ten obsahuje všechna relevantní data a umožňuje dlouhodobé archivování.

Samozřejmě zde nechybí podpora technologie intranet/internet a tím i podpora ovládání na dálku, nově je i zde zřízena funkce pro smartphony. Za zmínku zřejmě stojí tzv. „Faceplate“, což jsou speciální objekty, které programátor může vytvářet sám. Těmito objekty mohou být např. regulátory, měřicí systémy, grafy s tlačítky nebo cokoli jiného. Jejich hlavní výhodou je možnost centrální změny parametrů, i když jsou již umístěny někde v projektu na jednotlivé obrazovky.



Obr. 13 Přehled verzí prostředí WinCC [12]

3.8 Shrnutí

Všechny výše zmíněné systémy mají takřka stejné možnosti konektivity i návrhu vlastní vizualizace. Je to zapříčiněno konkurenčním bojem těchto firem, aby byly schopné konkurovat, musí mít stejné možnosti jako ostatní. Jediné rozdíly na první pohled patrné jsou vývojová prostředí a jejich různý přístup k zpracování a vizualizaci dat, je tedy už jen na vývojářích jaký systém se rozhodnou používat a se kterým se naučí pracovat. Jako jediný logický faktor, podle kterého se dá orientovat je stabilita celého systému a pořizovací cena daného systému. Stabilitu systému se bez hlubší znalosti a zkušeností s daným softwarem, posuzovat nedá proto bych následující srovnání provedl podle pořizovací ceny. Tato cena bude tedy odhadnuta na zadanou aplikaci pro stanice měření netěsností. Následující tabulka uvádí ceny dostupných systému, u nevyplněných řádků nebylo možno zjistit cenu u výrobce.

Systém	Cena
Wonderware InTouch	-
Control WEB	24 100 ,-Kč
Reliance	Reliance 4 Design Enterprise (3000 bodů) 21 900 ,-Kč
Promotic	PmDev 9 000 PmRtStandard 20 000 Celkem: 29 000 ,-Kč
WinCC	-

Řešení této práce se však skýtá jinde než u výše popsaných HMI/SCADA systémů. Výsledný program bude napsán ve vývojovém prostředí Visual Studio 2015, jehož výstupem bude „.exe“ soubor který bude spuštěn na počítači umístěném ve stanici na měření netěsností. Toto řešení bylo vybráno, jelikož není vázáno žádným systémem a jeho omezeními. Jelikož se jedná o vývojové prostředí možnosti, které poskytuje, jsou mnohem rozsáhlejší než u výše uvedených systémů. Navíc je zde požadavek, aby aplikace používala více vláken na zpracování dat, tohoto cíle bude nejspíše dosaženo právě zde.

4 Databázové systémy

Pod pojmem databáze se rozumí kolekce dat uložených v jednom nebo více souborech na pevném disku. Za tímto účelem se používají tzv. systémy řízení báze dat (SŘBD, popř. DBMS – Database Management System), tento systém obstarává přístup k údajům uloženým v databázi, tzn. databázový systém = DBMS + databáze. V databázích existují tři základní způsoby pro jejich tvorbu. Prvním z nich je hierarchický model, zde jsou data uložena v hierarchické struktuře, která připomíná „obrácený strom“. První z tabulek slouží jako kořen a ostatní tabulky jsou vázány na ni, jako větve vycházející z kořene nebo z ostatních větví. Vztahy jsou zde prezentovány pojmy rodič a potomek. Mezi hlavní nevýhody tohoto modelu jsou vztahy M:N, které nejsou v této struktuře podporovány. Druhým z modelů databází je síťový model, jehož struktura je definována pojmy uzel a množinová struktura, kde uzly odpovídají entitám a množinové struktury prezentují vztahy mezi jednotlivými entitami. Hlavní nevýhodou této struktury jsou její obtížné modifikace.

Třetím modelem je relační databázový model, který je v současné době nerozšířenější, jeho zakladatelem je Dr. Edgar F. Codd, který tento model uvedl roku 1969. Model založil na dvou hlavních disciplínách, na predikátové logice a na teorii množin. K samotným fyzickým datům přistupujeme přes logickou strukturu, jejímž základním prvkem je tabulka. Tyto tabulky neboli také entity, se skládají ze sloupců tzv. atributy (ID, jméno, příjmení,...) a řádků, které tvoří záznam samotných dat. Každý řádek zde obsahuje unikátní klíč pomoci, kterého lze snadno dohledávat konkrétní záznam uložený v databázi. Jeho hlavní výhodou je jednoduché vyjádření všech typů vazeb mezi jednotlivými záznamy tabulky, tzv. kardinalita vztahu. Ta nám vyjadřuje kolik řádků z jedné tabulky může vstupovat do vztahů druhé tabulky.

Pro samotnou práci s databází je v dnešní době nejpoužívanější jazyk SQL, který slouží nejen pro získávání a ukládání dat, ale také pro jejich modifikaci a definici datové struktury. [14]

4.1 Oracle Database 12c

Jedním ze světové špičky databázových řešení je společnost Oracle, která sama tvoří normy a udává směr vývoje tohoto odvětví. Ve velké míře je jejich systém využíván ve státní správě a ve velkých firmách, kde se požaduje co nejrobustnější řešení. Verze 12c je nejnovější verzí systému, která byla uvedena na trh v roce 2014. Architektura této databáze má jeden hlavní rozdíl oproti drtivě většině ostatních systémů a to ten, že Oracle má vždy k jedné instanci serveru pouze jednu databázi jako takovou. U ostatních systémů je možné v jedné instanci obsluhovat současně několik oddělených databází.

Oracle 12c je relační databáze, která podporuje konstrukci objektů, ty se skládají z tabulek jako u normální databáze. Výhodou tohoto řešení je ulehčení práce při návrhu databáze. Další výhodou je podpora procedurálního jazyka PL/SQL, třeba i s XML datovými typy. Unikátním způsobem je zde také řešena výkonost, bezpečnost a ukládání dat, což v kombinaci s mnoha ostatními vlastnostmi přináší jeden z nejspolehlivějších systémů na trhu. Tento fakt je však vykoupen vysokou cenou systému, sice existuje více variant, ale pro potřeby této práce se nabízejí lepší kandidáti. Jelikož i přes jeho robustnost a bezpečnost samotný systém není příliš rychlý, co se týče vyhledávání dat, a také je náročný na jeho správu.

Hardwarové požadavky systémů se liší podle použitého operačního systému, zdali je použita 32-bitová nebo 64-bitová verze. Obecně se však parametry dají shrnout jako, 5GB volného místa na pevném disku a minimálně 1GB operační paměti. Mezi podporované operační systémy patří nejen Windows a Linux ale také méně používané systémy jako je Solaris a HP-Unix. [16]

4.2 MySQL

Jedná se o nejrozšířenější Open Source databázový systém na světě. Původně byl vytvořen firmou MySQL AB, od roku 2008 je však vlastněn společností Sun Microsystems, dceřinou společností Oracle Corporation. Jak už název napovídá komunikace s databází je založena na dotazovacím jazyku SQL. MySQL se hojně využívá v kombinaci s Apatchem, operačním systémem Linux a s jazykem PHP. Společně se nazývají technologií LAMP a využívají se jako softwarový základ webových serverů. Z tohoto faktu vyplývá že MySQL je nejvíce nasazeno na webových stránkách (např. Facebook, YouTube, Ebay).

Jelikož existuje více verzí MySQL odvíjejí se od toho různé systémové nároky. Podpora se však týká obou verzí operačního systému tedy pro 32-bitové i pro 64-bitové systémy. K podporovaným operačním systémům patří snad všechny známé systémy jako např. Linux, MacOS, Solaris, HP-Unix a samozřejmě Windows ale až od verze 2000. Dalším nárokem je podpora protokolu TCP/IP a minimálně 200MB volného místa na pevném disku. [15]

4.3 Microsoft SQL server

MS SQL je relační systém řízené báze dat, a obdobně jako u společnosti Oracle je i firma Microsoft jedním z klíčových lídrů na trhu databázových systémů. Jelikož se jedná o databázový systém od firmy Microsoft je zde zahrnuta podpora .NET, tím se zde nabízí komplexní služby pro vývojáře, které umožňují zkrácení času na vývoj a tím i celkové snížení nákladů. Jsou zde však podporovány pouze operační systémy od firmy Microsoft a tím se náklady o něco zvedají. Podobně jako u společnosti Oracle je zde zahrnuta podpora složitých datových typů, avšak jejich použití se příliš nedoporučuje zejména pro nadměrné zatížení systému.

Obdobně tak jako u MySQL se i zde liší systémové nároky podle edice. Avšak pro praxi to znamená, že pro 32-bitovou architekturu je zapotřebí 512 MB operační paměti a procesor minimálně řady Pentium III. U 64-bitové architektury se nároky zdvojnásobují což znamená 1GB RAM a procesor o minimální taktovací frekvenci 1GHz, což v dnešní době představuje velice malé nároky, které dokáže splnit každý osobní počítač.

4.4 PostgreSQL

PostgreSQL je následovníkem systému Ingres. Jeho vývoj byl započat roku 1986 v kalifornské univerzitě Berkley. Projekt Progres byl vyvinut za účelem řešení tehdejších problémů s databázemi jako např. podpora vlastních datových typů, alertů nebo triggerů. V roce 1995 s nástupem jazyka SQL se vývoj přesunul mimo univerzitu a o dva roky později došlo ke změně názvu na PostgreSQL, první vydanou verzí byla verze 6.0. PostgreSQL je podle slov výrobce nejpokrokovější Open Source relační systém databází současnosti. Jeho podpora zahrnuje transakční zpracování dotazů, i podporu cizích klíčů. K nejznámějším firmám používající tento DBMS patří Yahoo a Skype. [17]

Systémové požadavky tohoto DBMS jsou 64-bitový operační systém, 2GB volného místa na pevném disku a je doporučen alespoň dvoujádrový procesor. K dalším doporučením patří metoda RAID 1 a vyšší, je to hlavně kvůli ochraně dat na disku. Mezi

podporované operační systémy patří snad všechny známé systémy jako např. Windows, Linux, MacOS, Solaris a další. Výhody tohoto systému jsou rozšiřitelnost jazyka SQL, podpora dotazů a neustálé dodržování nejnovějších standardů jazyka SQL. K nevýhodám se zde však řadí menší podpora mezi Open Source aplikacemi (spíše MySQL) a menší rychlost systému.

4.5 SQLite

SQLite je velice malá knihovna napsána v programovacím jazyce C, jejímž primárním úkolem je implementace jazyka SQL do souboru dbm. Celá databáze je tedy pak umístěna do jediného souboru, který je uložen na pevném disku. Toto je největší rozdíl oproti ostatním dříve popsaným systémům, které pracují na principu server-klient, kde databázový server je spuštěn jako samostatný proces. Díky její jednoduchosti jsou systémové nároky takřka minimální, zajímavostí také je že celková velikost činí přibližně 350kB. Na druhou stranu ze standardu SQL92 nejsou implementovány všechny funkce, i přes to se jedná o velice nasazovaný systém, který lze použít u velké škály programovacích jazyků jako např. C, C#, Java, PHP a další. Díky tomu lze tuto knihovnu použít snad na všech operačních systémech. Mezi největší firmy používající tento systém patří Netscape Communication Corporation, která jej používá v ochranné známce Mozilla.

Tento systém byl vybrán pro řešení této práce, díky jeho jednoduchosti, rychlosti a také proto že se jedná o volné dílo a tím i odpadají pořizovací náklady. Dalším důvodem je uložení databáze v jednom souboru, tento fakt snadno umožní zálohování celé databáze, tím že se jednoduše vytvoří kopie daného souboru.

4.6 Porovnání DBMS

Na základě více popsaných systémů byl vytvořen stručný přehled základních parametrů pro možnost rychlého porovnání.

4.6.1 Základní údaje

V následujícím přehledu jsou vypsány typy licencí. Proprietární software neboli také software s uzavřeným kódem je software kde autor upravuje licenci, v praxi to znamená, že zdrojové kódy nejsou volně přístupné a je třeba jej zakoupit u autora. General public licence (GPL) je označení pro všeobecně veřejnou licenci s požadavkem na to že upravené dílo musí být pod stejnou či kompatibilní licenci. BSD licence je pak označení pro svobodný software kde jsou uvedeny informace o autorovi a zřeknutí se jakékoli zodpovědnosti co se s dílem stane. S takto označeným dílem se může pak dělat cokoliv pokud se zachovají informace o autorovi a zřeknutí se odpovědnosti. [19]

DBMS	Výrobce	První verze (rok)	Poslední verze	Licence
Oracle	Oracle Corporation	1979	12c	Proprietární SW
MySQL	Sun Microsystem	1996	5.7	GPL
MS SQL	Microsoft	1989	13.0.1601.5	Proprietární SW
PostgreSQL	PostgreSQL Global Development Group	1989	9.6.1	BSD
SQLite	D. Richard Hipp	2000	3.16.1	Public domain

Tabulka 1: Přehled základních údajů databázových systémů

4.6.2 Podpora operačních systémů

Při výběru databázového systému je dobré mít přehled o možné použitelnosti na jiných operačních systémech. Značnou výhodu mají systémy, u kterých závisí dostupnost na existenci Java Virtual Machine a ne na operačním systému.

DBMS	Windows	Linux	Unix	MacOS
Oracle	Ano	Ano	Ano	Ano
MySQL	Ano	Ano	Ano	Ano
MS SQL	Ano	Ne	Ne	Ne
PostgreSQL	Ano	Ano	Ano	Ano
SQLite	Ano	Ano	Ano	Ano

Tabulka 2: Podpora operačních systému databázových systémů

5 Datový model

Cílem relačního modelu dat je vytvořit popis modelované reality pomocí entit (tabulek), atributů (sloupců) a relací mezi nimi. Entitou tedy bude každý významný, rozlišitelný objekt reality. Atributem je pak každá myšlená vlastnost nebo charakteristika entity. Relace jsou pak vztahy mezi jednotlivými entitami.

5.1 Stanovení entit

U stanic na měření netěsností je vhodné uchovávat hned několik okruhů dat a událostí. Hlavním okruhem dat, který je potřeba uchovávat jsou výsledky měření. Znalost těchto výsledků je stěžejním prvkem při zpětném hledání chyby ve výrobě, v případě kdy zákazník vrátí vadný kus lze zpětně dohledat, jak daný kus byl ověřen, použitá metoda nebo kdo kontrolu prováděl, kdy k tomu došlo a jaké byly výsledné parametry zkoušky. Dalším okruhem dat je záznam pracovníků provádějících zkoušky, případně údržbu na dané stanici. K tomu se také váže záznam příchodů a odchodů ke stanici (pracoviště), který využije hlavně management podniku.

Dalším významným okruhem dat je záznam o kalibraci stanice, eventuálně ověření této kalibrace. Kalibraci stanice je vhodné provádět v pravidelných intervalech stejně tak i ověření těchto hodnot, tyto parametry jsou stěžejní pro výsledné měření. Tento okruh však z časových důvodů nebyl zahrnut do finálního řešení aplikace. Nepostradatelnou složkou je také záznam chyb, stavů a varování na stanici. Díky těmto datům je zpětné hledání chyby na stanici značně ulehčeno a urychleno. V pravidelných intervalech by také měly být skladovány stavy čítačů. Jedná se hlavně o čítače otestovaných výrobků, zdali byla zkouška úspěšná/neúspěšná nebo měření skončilo chybou. Tyto údaje jsou nezbytné pro statistické zhodnocení výroby.

Pro potřeby údržby stanice je také nezbytné skladovat data od čítačů jednotlivých zařízení (pumpy, ventily, ...) ty se hlavně využijí v hlášení pro údržbu, kdy je potřeba vyměnit olej nebo ventil. Posledním okruhem dat jsou parametry stanice. Tyto parametry se dělí do dvou skupin, první skupinou jsou metody měření. Metody jsou parametry, které jsou proměnné od měřeného výrobku, např. tlak helia ve výrobku, tlak pevnostní zkoušky a další. Těchto metod může být tolik kolik je testovaných výrobků na stanici. Druhou skupinou jsou parametry stanice, ty stanovují základní nastavení jako např. max/min tolerance atmosféry v komoře, čas zavzdušnění komory a jiné.

Hlavními entitami v tomto datovém souboru tedy jsou:

- Výsledky měření
- Operátor provádějící měření, popřípadě údržbu
- Časy přihlášení a odhlášení operátora
- Výsledky kalibrace stanice, nebo jejich ověření
- Metody měření, datum jejich poslední editace/vytvoření a název
- Parametry stanice
- Čítače pro jednotlivá zařízení (pumpy, ventily)
- Čítače výsledků měření (dobrý kus, špatný kus, chyba)
- Záznam alarmu, stavů

5.1.1 Metody měření

Tyto parametry jsou jedny z nejdůležitějších k nastavení stanice. Jsou specifické ke každému testovanému výrobku a proto je důležité mít je řádně uloženy spolu s relevantními informacemi o nich. Obtíž těchto parametrů spočívá v tom, že každá stanice má své vlastní a nelze tedy vytvořit jejich společný formát. Proto jejich ukládací mechanismus musí být univerzální a s tímto faktorem počítat. Parametry související k metodám jsou kromě jejich názvu a hodnoty také popisy a offset, který určuje místo v paměťovém bloku PLC na kterém se daná proměnná nachází. Jelikož je metod více tak nejjednodušší formou záznamu by bylo dvourozměrné pole, to se však neslučuje s koncepcí databází a tak je zapotřebí vytvořit dvě tabulky, jednu s více uvedenými parametry a druhou pouze pro hodnoty. Sloupce tabulky s hodnotami jsou tedy názvy jednotlivých parametrů metody, a proto tato tabulka bude pro každou stanici jiná.

Relevantními informacemi které se vztahují k této entitě jsou:

- Datum a čas vytvoření
- Datum a čas poslední modifikace
- Operátor, který prováděl jednu z předešlých věcí

Tyto informace postupem času mohou nabývat na objemu, proto je vhodné pro ně vytvořit samostatné tabulky. Každá z těchto dvou tabulek obsahuje dva cizí klíče jeden od metody, ke které se záznam vztahuje a druhá od operátora provádějícího změny. Tímto způsobem jsou uloženy časové záznamy.

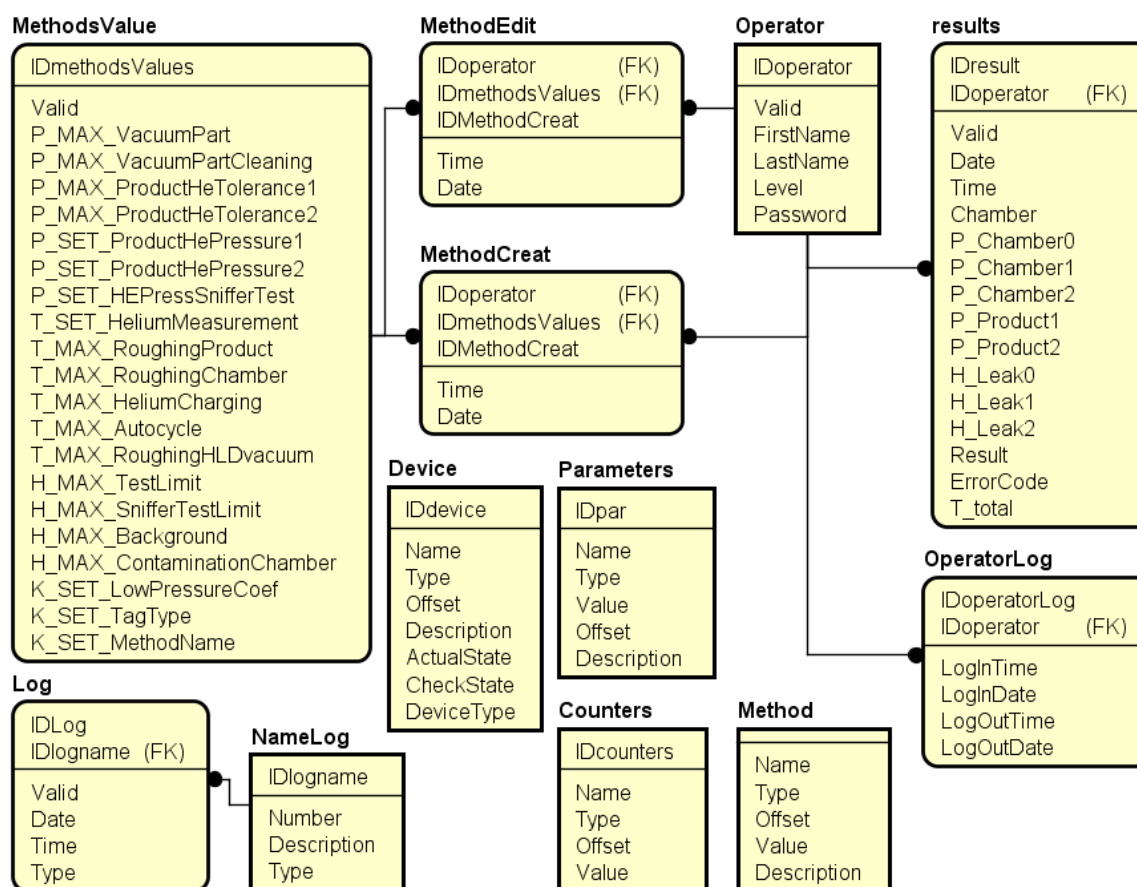
5.2 Relace

Ze stanovených entit je možno vycítit potřebné relace mezi nimi. Tyto vzájemné vztahy jsou v datovém modelu vyjádřeny pomocí cizích klíčů (FK – foreign key), ty předávají spojující informace mezi tabulkami.

Relace v modelu tedy jsou vztahy:

- Každý výsledek vytvořil právě jeden zaměstnanec
- Každý operátor přišel a odešel ke stanici
- Kalibraci provádí právě jeden operátor
- Metodu vytvořil operátor, v určený čas
- Naposledy upravoval metodu operátor
- Metody mají svůj výčet parametrů
- Záznam o stavu stanice má právě jeden název

Na následujícím obrázku je znázorněn datový model. U žádné z tabulek neexistuje přirozený kandidátní klíč, proto je u všech ostatních tabulek zaveden primární klíč (PK) umělý, tabulka „Method“ primární klíč neobsahuje, protože není relevantní k jejímu obsahu.



Obr. 14 Datový model pro stanici měření těsnosti

5.3 Atributy

Jelikož na první pohled není z modelu dat možné pochopit význam jednotlivých atributů, je zde nastíněn jejich účel. Atribut valid, jenž se vyskytuje téměř v každé tabulce značí, zdali daný záznam je viditelný v aplikaci. Smaže-li operátor daný záznam hodnota valid se nastaví na nulu a nebude jej možné již zobrazit, až po odblokování administrátora, tyto záznamy však zůstanou v databázi pro případ, že by se jednalo o omyl při výběru, sice celková velikost databáze může narůst, avšak velikost jednotlivých záznamu není tak markantní, aby si s tímto faktorem dnešní harddisky nijak neporadily.

Tabulka výsledků (results)

- Chambre – číslo komory na které byl výrobek otestován
- H_Leak0– pozadí komory, jedná je hodnotu helia v komoře na začátku testu
- P_Product1 – zkouška heliem (helium test), tlak zkoušky
- P_Chamber0 – hodnota vakua v komoře
- Result – výsledek OK/NOK kus
- T_total – výsledný čas testu

Čítače pro zařízení (device), každý záznam v tabulce představuje jeden čítač výsledků nebo jiné události, proto sloupec „Name“

- Name – název čítače
- Offset – pozice uložení v datovém bloku PLC
- Type – typ zařízení, pumpa, ventil, héliový detektor

6 Technologie OPC

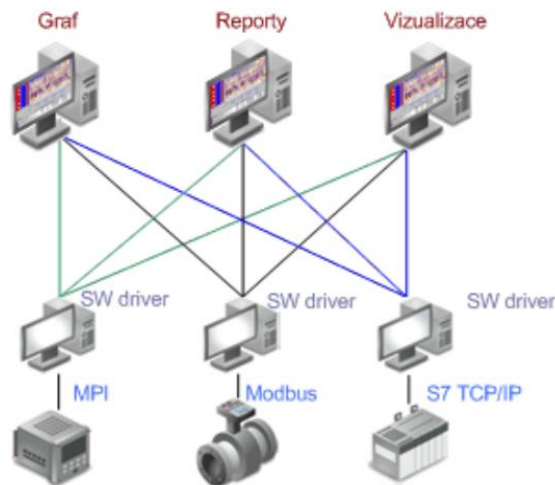
Řízení více stanic se nejnáze docílí pomocí společné komunikační technologie, která je přístupná u všech běžných výrobců na trhu a tou je právě OPC. Tato kapitola se věnuje právě tomuto standardu a jeho specifikacím.

V současné době se pro přenos dat v průmyslových řídicích systémech stále častěji používá standart OPC (OLE for Process Control). Tento standart průmyslové automatizace vznikl ve spolupráci mnoha světových dodavatelů automatizačních prostředků hardwaru a softwaru, spolu s firmou Microsoft. Jedná se o společné komunikační rozhraní určené pro vzájemnou komunikaci mezi různými zařízeními určenými pro monitorování a řízení technologických procesů. Jeho hlavní ideou je zabránit závislosti daného monitorovacího nebo řídicího softwaru na výrobcu hardwaru. Díky této vlastnosti si koncový uživatel může vybrat libovolný software a hardware nehledě na výrobce dané technologie, standart OPC zajistí správnou komunikaci mezi těmito zařízeními. Nezajímá se tedy tak o dostupnost komunikačních driverů pro jednotlivá zařízení. OPC standart je založen na metodách OLE, COM a DCOM (Object Linking and Embedding, Common Object Model, Distributed COM).

Tento standard vyvíjí a udržuje nezisková organizace OPC Foundation. Tato organizace sídlí ve Scottsdale, Arizona USA. Společnost OPC Foundation sdružuje více než 300 členů z řad celosvětových firem mající významný podíl na průmyslové automatizaci. Mezi tyto firmy lze např. zařadit Honeywell, Rockwell Software, Siemens atd.. Jedním z nejpodstatnějších členů je samozřejmě společnost Microsoft, která se neustále aktivně podílí na tvorbě nových specifikací. Z českých firem jsou zde členy Merz (Liberec), OPC Labs (Plzeň) a Geovap (Pardubice.) [7]

6.1 Přenos dat

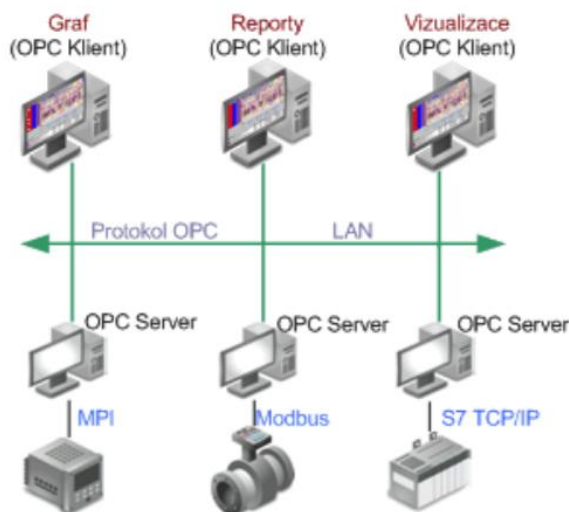
Použijeme-li hardware od různých výrobců a jejich klientský aplikační program, jsme obvykle nuceni pro získávání dat použít ovladače daného zařízení. Tyto aplikační programy byly vyvinuty pro získání dat z více zdrojů, jsou však závislé na konkrétních ovladačích. Při použití této architektury může docházet k vzájemnému ovlivňování komunikace, nekompatibilitě s daným operačním systémem atd.. K další komplikaci může dojít při rozšíření nového zařízení, kdy je zapotřebí úprava řídicího systému (úprava na nový ovladač). Dále zde může nastat problém s ovladači od různých výrobců, ne každý dodavatel podporuje všechny vlastnosti daného hardwaru. Toho se týká i změny vlastností daného hardwaru, ovladače se stávají nefunkční a je zapotřebí upgrade. Jedna z možností topologie přenosu bez použití standardu OPC je znázorněna na Obr. 15.



Obr. 15 Komunikace bez OPC [7]

Výrobci hardwaru tento problém řeší vývojem dalších ovladačů, avšak brání jim odlišnost v klientských protokolech. Tyto rozdíly, jsou zapříčiněné konkurenčním bojem mezi vývojáři.

Přenos dat využívající technologii OPC je založen na architektuře Klient – Server. OPC je tedy primárně rozděleno do dvou typů programů a to na OPC Server a OPC Klient. Rozhraní pro komunikaci mezi veškerým hardwarem a softwarem tvoří komunikační kanály zejména se jedná o Ethernet, LAN apod.. Topologie je znázorněna na Obr. 16. Výměna dat je zde realizována metodami OLE, COM a DCOM.



Obr. 16 Technologie OPC [7]

Budeme-li dále mluvit o OPC, je vhodné uvést některé základní pojmy.

CACHE – je to pomocná paměť pro OPC server, ta využívá průběžné ukládání dat z jiných zdrojů (PLC). Potřebuje-li OPC klient data, není nutné je získávat přímo z PLC, ale mohou být snadno a rychle získána z této paměti. [9]

DEVICE – jedná se o zdroj dat, ze kterého OPC Server získává data např. PLC.

OPC Klient – jde o program, který přijímá data z OPC Serveru ve formátu OPC. Tyto data jsou zprostředkována v podobě vizualizace viz kapitola 3.1.

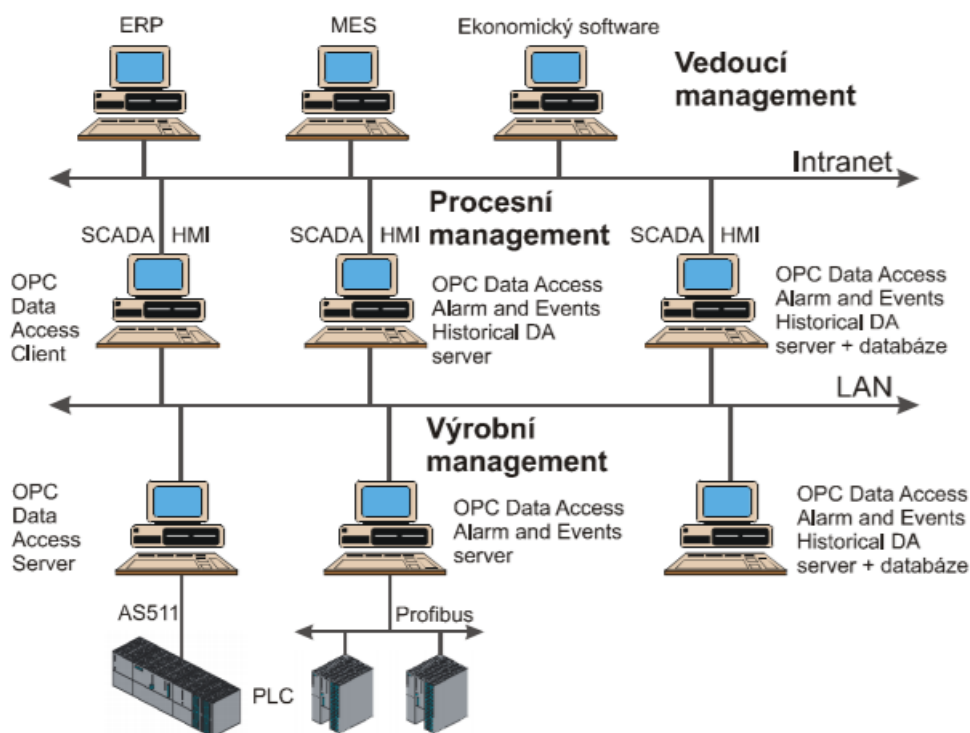
OPC Server – tento program komunikuje s připojeným zařízením a jeho komunikačním protokolem (např. Modbus, MPI, PPI aj.). Tato data se dále převádějí do formátu OPC a jsou dále poskytována ostatním aplikacím.

Asynchronní přenos – probíhá v případě, kdy klient pošle serveru požadavek na čtení/zápis dat a dále již nečeká na výsledek požadavku, ale provádí jiné operace. Je patrné, že tento typ přenosu je vhodný u déle trvajících operací protože server zpracovává požadavek během určité doby a výsledek je poslán až poté.

Synchronní přenos – zde se jedná o opačný typ přenosu server – klient. Kdy klient pošle požadavek na OPC server a čeká na odpověď. Tento způsob komunikace je vhodný pro rychlé operace. U většiny příkladu je zde OPC klient na stejném PC jako OPC server.

6.2 Architektura OPC

Při výměně dat podle standardu OPC jak už bylo dříve zmíněno je využíváno přijaté a osvědčené schéma klient/server. K jednomu serveru se může připojit více klientů od různých výrobců, stejně tak se i jeden klient může připojit k serverům od různých dodavatelů. Na následujícím obrázku je znázorněna architektura OPC serveru.



Obr. 17 Architektura OPC [8]

Jak jde vidět z Obr. 17 tato architektura má tři úrovně řízení. Nejnižší úroveň označena jako výrobní management, obsahuje komunikační a řídicí počítače připojené k logické vrstvě systému (PLC) na jedné straně a na druhé do místní podnikové sítě (LAN). Vedle těchto počítačů je ještě znázorněn komunikační a datový server, který může uchovávat technologická data v databázi. V dnešní době je však zvykem tyto počítače slučovat do jednoho.

Střední úroveň řízení je také nazývána procesní (Process Management), zde se obvykle nacházejí počítače s vizualizačními a monitorovacími aplikacemi, které prezentují jednotlivé výrobní procesy operátorů v grafické podobě. V dnešní době bývá zvykem první a druhou úroveň slučovat do jedné, avšak tento krok je vhodný především pro menší

výrobní technologické procesy. U rozsáhlých a komplexních výroben je vhodné dodržovat tuto architekturu.

Na nejvyšší úrovni, označené jako vedoucí management, jsou především nadřazené podnikové informační systémy typu MES (Manufacturing Execution System), ERP (Enterprise Resources Planning) a různý ekonomický software. Zde je OPC standart využit jako společné komunikační rozhraní pro systémy typu SCADA. [8]

6.3 Specifikace OPC

Pomocí OPC specifikací je vytvářen OPC standard. Tyto specifikace jsou volně přístupné technické dokumentace, které definují chování a nastavení standardu OPC. Následující přehled popisuje jednotlivé specifikace, které jsou dostupné na stránkách organizace OPC Foundation.

OPC Data Access (DA) – jedná se o jednu z nejstarších a nejpoužívanějších specifikací, která určuje přístup k datům v reálném čase. Jejím záměrem je především spojitý charakter datové komunikace. Standart podrobně popisuje datovou komunikaci mezi logickou vrstvou (PLC, DCS) a klientskými aplikacemi jako jsou uživatelská a datová rozhraní (HMI).

OPC Alarm and Events – je zde definováno poskytování informací OPC klientům o výskytech alarmů a specifických událostí. Především se jedná o problematiku sdílení a výstrah událostí mezi koncovými zařízeními a klientskými aplikacemi. Upozornění na alarm nebo specifikovanou událost je na rozdíl od DA poskytováno na vyžádání.

OPC Batch – Obdobně jako u OPC Data Acces, používá se u spojitých provozů a je především určena pro technologie s dávkovou výrobou.

OPC Historical Data Access – Defínuje problematiku klientských aplikací přístupu k procesním datům z databázových a datových zařízení. Tento protokol se zabývá pouze problematikou výměny již archivovaných dat. Stejně tak jako u DA je možné využít technologii DCOM, to umožňuje přenášet archivovaná data po síti LAN.

OPC Security – Je určen pro důkladné zabezpečení přístupu obsluhy při ovládání technologie přes OPC klient prostřednictvím OPC serveru. Využívá se zde zabezpečení systému Windows.

OPC Data Exchange - Slouží pro horizontální komunikaci mezi řídicími jednotkami s různými komunikačními protokoly. (např. EtherNet/IP, PROFINet) prostřednictvím sítě Ethernet. [7]

OPC Complex Data – jsou zde definovány možnosti popisu komplexních dat a jak s těmito daty zacházet a zpřístupňovat je.

OPC Universal Access – sjednocují se zde standardy jako OPC Data Access, OPC Alarms and Events a OPC Historical Data

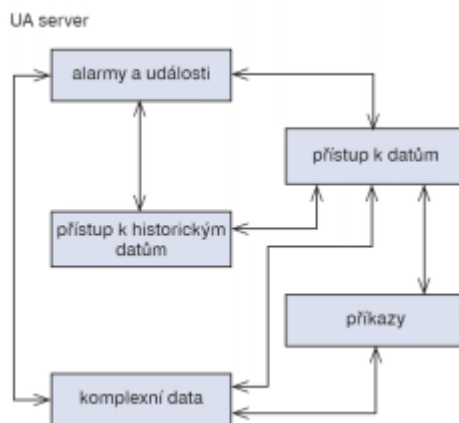
6.4 OPC UA

Během vývoje OPC bylo snahou vyvinout standard pro komunikaci nejen mezi snímači, akčními členy s PLC a SCADA systémy ale i vyššími úrovněmi řízení jako jsou systémy MES a podnikové rozhraní ERP. Postupně tedy vznikali specifikace jako OPC DA a OPC AE atd.. Problémem těchto specifikací je jejich koexistence, která není vždy bezproblémová. Využívání mechanismů COM a DCOM se časem ukázalo jako omezení, vzhledem k tomu že se v průmyslu používají i jiné operační systémy než Windows. V dnešní době jsou standardy COM a DCOM zastaralé a patří již tedy minulosti. Proto

vznikl tento standart OPC UA, který je založen na webových službách používající jazyk XML.

Společnost Microsoft tedy opustila řešení COM, DCOM a staví tento standart na koncepci „.net“ (dot net), využívající ověřené webové a internetové standardy a architekturu orientovanou na služby. SOA (Service Oriented Architecture) tedy architektura orientovaná na služby má základní myšlenku postavenou na tom, že jednotlivé aplikace jsou složeny ze služeb, k nimž aplikace přistupují prostřednictvím standardizovaných rozhraní. Architektura stanovuje, jak jsou jednotlivé služby v síti lokalizovány, vykonávány, spravovány, sledovány a zabezpečovány. Architektura SOA se skládá ze tří základních prvků, těmi jsou odběratel, registr a poskytovatel. Poskytovatel posílá do registru informace o síťových službách, které je schopen zajistit a o tom, jak vypadá rozhraní pro tyto služby, zatím co odběratel hledá v tomtéž registru vhodného poskytovatele dané služby. Výhodou tohoto systému oproti COM a DCOM je kromě nezávislosti na platformách od společnosti Microsoft, také to, že odběratelé a poskytovatelé, resp. jejich služby, jsou spojeni ad hoc, a mohou se tedy na sobě nezávisle měnit. U COM a DCOM zatím byly jednotlivé služby pevně vázány na dané klienty. Tato vlastnost SOA tedy sebou přináší značnou flexibilitu řešení. [13]

Jak již bylo řečeno základním prvkem OPC UA je server, který v sobě spojuje všechny dřívější funkce klasických OPC serverů, viz Obr. 18. Spojuje tedy přístup k datům, k historickým datům i k událostem a alarmům. Pracuje s jednou sadou služeb pro zápis/čtení dat, zobrazení a procházení datového prostoru. Jelikož je adresový prostor jednotný, nemůže dojít k nekonzistenci dat mezi servery pro data a události. Je zde i zajištěna odolnost vůči chybám komunikace, kdy klient může požadovat pravidelnou zprávu potvrzující jejich komunikaci. Bezpečnost komunikace mezi klientem a serverem zde zajišťují funkce pro řízení přístupu, kterými jsou standardně UA servery vybaveny. Klient zde předkládá své oprávnění a server kontroluje, zdali klient má oprávnění na danou službu, tyto zprávy je možno šifrovat pro zvýšení bezpečnosti. [13]



Obr. 18 OPC Universal Access [13]

7 Realizace aplikace

Pro realizaci výsledné SCADA aplikace bylo zvoleno vývojové prostředí Visual Studio, proto bych rád začátkem této kapitoly rozebral toto vývojové prostředí a jeho koncept. Další část textu je věnována kompozici aplikace, na co je zaměřena, a jaké jsou na ni kladené nároky a požadavky.

7.1 Visual Studio

Pro realizaci aplikace bylo zvoleno vývojové prostředí Visual studio, jedná se o univerzální platformu vytvořenou společností Microsoft pro tvorbu široké škály aplikací na různých platformách Microsoft. Visual studio samozřejmě obsahuje editor kódu se zvýrazňováním syntaxe, kompilátor integrovaných jazyků, debugger a v neposlední součásti Solution explorer a Team explorer. Solution explorer umožňuje procházet strukturu projektu a editace jeho komponent. Team explorer představuje zálohování projektu na serverové uložení. Tato služba není nijak zpoplatněna pro menší tým programátorů (do 5 členného týmu), stejně tak i licence potřebná pro užívání. Visual studio je tedy „freeware“ vývojové prostředí. Další součástí dostupnou pod touto licencí je nástroj IntelliSense, který automaticky doplňuje názvy tříd, metod, proměnných atd. čímž usnadňuje a urychluje práci programátorů. Mezi implicitně podporované jazyky patří C/C++, Visual Basic .NET a C#. Podpora dalších jazyků je možná, ale musí být přidána.

Visual studio dále obsahuje vizuální designery, které zobrazují grafickou podobu vytvářených komponent a aplikací. Mezi tyto designery patří Windows Form a WPF (Windows Presentation Foundation). WPF zobrazuje aktuální stav uživatelského rozhraní pomocí XAML (Extensible Application Markup Language). Nejnovější verzí Visual studia je Visual studio 2017.

7.2 .NET Framework

Společnost Microsoft vytvořila platformu .NET především pro své operační systémy Windows, které ji také nejvíce používají. Není však vyloučeno tuto platformu použít i na jiných operačních systémech. Nabízí prostředí pro vývoj a běh aplikací, a to jak spouštění aplikací, tak i potřebné knihovny pro ně. Programátor tak není při použití jedné z technologií přímo závislý na použití konkrétního programovacího jazyka. Nejpoužívanějšími jazyky pro vývoj těchto aplikací právě jsou C# a Visual Basic. Avšak je možné použít i jiné například C++, F# (funkcionální jazyk) nebo J# (.NET jazyk podobný Javě). Všechny tyto jazyky jsou schopny využívat společné knihovny dostupné ve Frameworku. Dále uvedený seznam některých součástí platformy .NET představuje její nejpoužívanější část:

- ASP.NET — slouží k tvorbě webových aplikací.
- ADO.NET — knihovna sloužící pro práci s databázemi.
- Windows Forms - tvorba grafických uživatelských rozhraní, v dnešní době už je využití Windows Forms spíše na ústupu.
- WPF (Windows Presentation Foundation) — tvorba grafických uživatelských rozhraní, postupně nahrazuje starší Windows Forms.
- WCF (Windows Communication Foundation) — vývoj webových služeb, případně komunikační infrastruktury mezi aplikacemi.

- LINQ (Language Integrated Query) — dotazovací jazyk, určený pro přístup k datům bez ohledu na jejich zdroj.

Velmi důležitou součástí Frameworku je CLR (Common Language Runtime). Jde o virtuální stroj, který spouští .NET aplikace. Jelikož ve Frameworku lze použít více programovacích jazyků, tak nezáleží na tom, jaký jazyk je použit při tvorbě aplikace. Zdrojový kód je tak či tak přeložen překladačem do tzv. společného mezi jazyka CIL (Common Intermediate Language). Ten až těsně před spuštěním aplikace je přeložen do nativního kódu odpovídajícího konkrétní architektuře, na které aplikace běží. Tento proces je nazýván Just-In-Time (JIT) překlad a jde o podobný proces, který je využit pro aplikace vyvíjené v Javě. Právě díky tomuto způsobu překladu není tvorba aplikací omezena na použití konkrétního jazyka a programátor má možnost použít kterékoli z podporovaných jazyků. Díky CIL je možné používat různé části kódu psané v jiných jazycích a objekty deklarované v různých jazycích mohou spolu komunikovat. Každá z částí bude přeložena zvlášť, právě do CIL, a poté již nic nebrání překladu do nativního kódu stroje. CLR se za běhu aplikace také stará například o ošetření výjimek, typovou bezpečnost, správu vláken a správu paměti. [22]

7.3 WPF a C#

WPF je součástí .NET Frameworku, určená pro tvorbu grafických uživatelských rozhraní pro aplikace na systémech Windows. WPF nahrazuje postupně starší technologii Windows Forms, která se ovšem stále hojně používá.

7.3.1 WPF

WPF technologie používá k vykreslování grafických objektů DirectX, který nejčastěji přímo přistupuje k hardwaru grafické karty. Dříve používané GDI (Graphics Device Interface) touto funkcí nedisponovalo a tím se tak část práce přenechávala procesoru. Nespornou výhodou WPF je, že veškeré grafické prvky jsou popisovány ve vektorové grafice, což umožňuje použití výsledné aplikace na zařízeních s různým grafickým rozlišením a velikostí obrazovky bez výrazného problému se zobrazením grafického rozhraní uživatele.

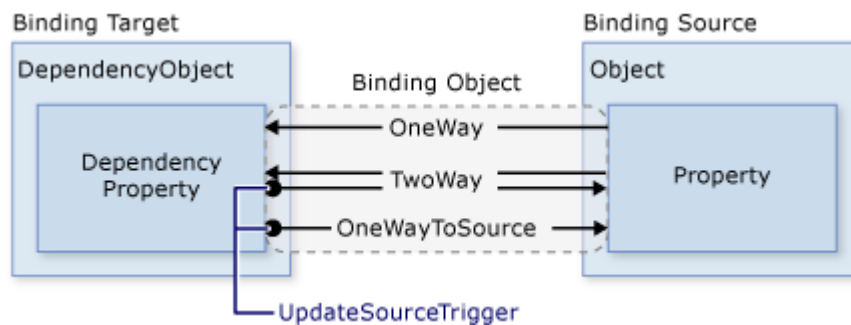
WPF odděluje vývoj grafického rozhraní a vnitřní logiky aplikace. Pro vytváření grafického rozhraní je použit jazyk XAML. K programování pozadí (logiky) aplikace je možno využít více .NET jazyků, jak již bylo zmíněno dříve, ale nejčastěji je však používán jazyk C#, stejně tak jako v této práci. Oba jazyky jsou dále podrobně rozebrány níže. Důležitým prvkem v technologii WPF je propojování dat tzv. Data binding.[20] Data binding umožňuje načítat data do aplikace z různých zdrojů a poté nezávisle na zdroji tato data zobrazovat uživateli. Pomocí data bindingu je zajištěno čtení dat ze zdroje do aplikačních objektů, odkud mohou být zobrazena a uživatelem upravena, tak také opětovně uložena zpátky do zdroje.

V praktickém použití to tedy znamená, že si programátor vytvoří veřejnou třídu, která je potomkem třídy `INotifyPropertyChanged`, implementuje metodu `OnPropertyChanged` a každému prvku grafického rozhraní vytvoří objekt `private` a `public`. Dále pak stačí z konstruktoru `mainwindow` vytvořit objekt této třídy a přiřadit ho do `DataContext` což je implementovaný objekt Frameworku. Tímto postupem se snadno docílí požadovaného propojení. V rámci WPF aplikace je možné rozlišit celkem čtyři druhy data bindingu:

- One way – data jsou pouze čtena do aplikace, pokud jsou data ze zdroje změněna, změna se projeví i v aplikaci

- Two way – předchozí princip zůstává stále stejný a platný i zde, ale jsou-li data upravena v aplikaci, změny se zapíší i do zdroje
- One way to source – aplikace má pouze zápisová práva ke zdroji
- One time - data jsou jednou načtena do aplikace, pozdější změny se nijak neprojeví

Na následujícím obrázku je znázorněn princip data bindingu.



Obr. 19 Data binding princip [21]

7.3.2 XAML

Jedná se o jazyk založený na XML, taktéž vytvořený formou Microsoft. U WPF slouží XAML k definici grafických prvků uživatelského rozhraní, jejich vlastností, zdrojů, stylů atd. Jedná se o kompletně deklarativní jazyk, který odděluje návrh uživatelského rozhraní od aplikační logiky programu.

Při deklarování jednotlivých prvků je využito stromové hierarchie, kde vždy existuje jeden kořenový prvek. Jednotlivé objekty popsané pomocí XAML kódu jsou propojeny přímo na instance objektů CLR. Níže uvedená ukázka XAML kódu vytvoří okno s tlačítkem.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Button Content="Button"/>
    </Grid>
</Window>
```

7.3.3 C#

C# je jeden z objektově orientovaných jazyků na vysoké úrovni, který byl vytvořen společností Microsoft pro použití v rámci Frameworku. Tento jazyk především vyšel z jazyků C++ a Java. CTS (Common Type System), všechny typy jsou tedy odvozené od třídy System Object. Dobrým příkladem je že každý typ dědí metodu ToString(), která vrací textovou podobu obsahu daného typu. Při použití jazyka C# při tvorbě WPF aplikace je důležitá existence klíčového slova partial, pomocí kterého se definují různé části jedné třídy ve více zdrojových souborech. Tato vlastnost je velice užitečná, když některá část třídy je generována automaticky a další část je psaná programátorem, což je přesně případ tvorby WPF aplikace pomocí Visual Studio. V jazyce C# není možné vytvářet globální proměnné a funkce, všechny prvky musí být obsaženy ve třídách. Náhradou mohou být statické metody a atributy. Vícenásobná dědičnost zde také není k dispozici, ale každá třída může implementovat více rozhraní. C# je case sensitive, v názvech se rozlišují velká a malá písmena, na rozdíl od jazyka Visual Basic, který je rozšířenější a také patří do rodiny .NET. Další zajímavostí je použití tzv. nullovatelných (nullable) typů. Nullable může být proměnná kteréhokoliv datového typu, pokud je označena právě takto. Výsledkem je, že hodnota daného typu může nabývat hodnoty null a s tím souvisí operátor koalescence,

který při přiřazení testuje zdrojovou proměnnou právě na tuto hodnotu. Ten pak případně do cílové proměnné přiřazuje hodnotu z alternativního zdroje. Od verze 3.0 je k dispozici programátorovi také klíčové slovo `var`. To lze využít při deklarování proměnných jako výsledku nějaké operace, není tak třeba rozepisovat cílový datový typ. Místo něj lze použít právě toto klíčové slovo a kompilátor poté provede operace na pravé straně, přiřazení automaticky určí datový typ dané operace a deklaruje výchozí proměnnou. Využití tohoto klíčového slova může výrazně zrychlit psaní kódu, nicméně se také může stát že výsledný kód se stane méně čitelným.

7.4 Koncept aplikace

Výsledná aplikace jak již bylo zmíněno, je určena pro stanice na měření netěsností a to zejména v průmyslovém použití. Proto jsou zde kladeny nároky na přehlednost údajů, jejich snadnou dostupnost. Další faktorem, na který se musí nahlížet je výsledný hardware, na kterých tato aplikace bude běžet. V tomto případě se jedná o dotykový počítač, umístěný v úrovni očí operátora. Je tedy zřejmé, že jednotlivé ovládací prvky musí mít dostatečnou velikost, aby s nimi bylo snadné zacházet třeba i v pracovních rukavicích apod.

Samotnou aplikaci je pak vhodné rozdělit na jednotlivé dílčí úkony stroje, nebo na operace které chceme právě provádět. Za tímto účelem je aplikace rozdělena do několika obrazovek, mezi kterými se přepíná pomocí TabControl, každá z obrazovek je pak tedy jakousi záložkou hlavního menu. Toto vše se hlavně týká grafického pojetí aplikace. Z funkcionálního hlediska by aplikace měla být schopna komunikovat s PLC, který je vždy použit pro řízení takto komplexního stroje. Při spuštění se automaticky připojit nebo v případě nezdaru ohlásit tuto skutečnost operátorovi a pokud možno zkoušet se neustále připojit v případě drobné závady. Další podstatnou funkcí je schopnost si autonomně stahovat a ukládat relevantní informace k procesu jako jsou výsledky měření, výsledky kalibrace, stavy jednotlivých snímačů apod., nejlépe k těmto skutečnostem vytvářet grafy a tabulky s těmito údaji.

7.5 Obrazovka měření

Úvodní a také hlavní obrazovkou aplikace je obrazovka měření. Na této stránce jsou k dispozici všechny nejdůležitější informace týkající se procesu měření. Informuje operátora o stavu stanice, průběhu a výsledcích heliového testu, hodnotách tlakových/vakuových čidel a heliového detektoru, dále jsou zde také umístěny tlačítka pro klasickou správu aplikace jako je přihlásit uživatele nebo vypnout aplikaci. Nechybí zde ani základní ovládací tlačítka jako je reset chyb a čítačů. Panel měření obsahuje několik přepínatelných polí, která umožňují sledovat aktuální stav stanice a logové chyby. Její rozložení je znázorněno na Obr. 20.

The screenshot shows the measurement application interface. At the top, there are tabs for 'MĚŘENÍ', 'SERVIS', 'RECEPT', 'PARAMETRY', and 'VÝSLEDKY'. The user is logged in as 'Uživatel: LT DefaultOP' and there are buttons for 'PŘIHLÁŠIT' and 'Vypnout'. The main area is divided into three columns for 'KOMORA 1', 'KOMORA 2', and 'KOMORA 3'. Each chamber has input fields for 'KÓD VÝROBKU', 'HELIUM KOMORA', 'TLAK VÝROBEK', and 'VÝSLEDEK'. Below this is the 'AKTUÁLNÍ DATA' section with fields for 'LIMIT NĚTĚSNOTI', 'HELIUM HLD', 'ČERPACÍ VĚTEV', 'MĚŘÍCÍ VĚTEV', 'TLAK V ZÁSOBNÍKU', and 'KONCENTRACE HE'. To the right is the 'LOG' section with buttons for 'Počítadla' and 'LOG History', and a table showing status events.

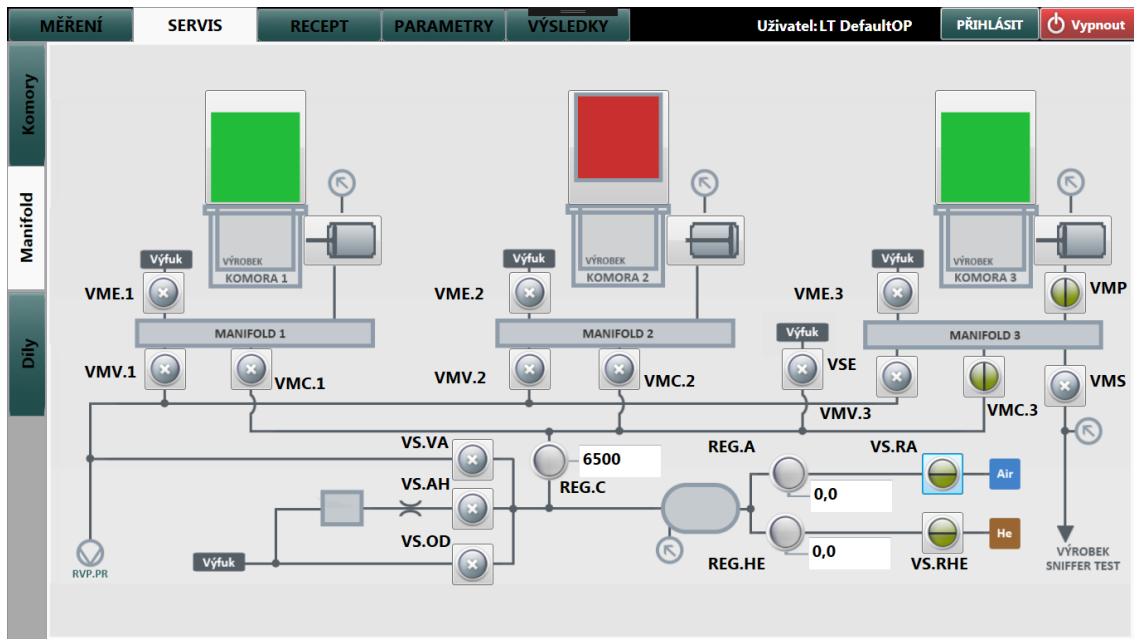
Čas	Číslo Stavů	Popis stavu
14:29:05	C-1	Manuální ovládání stanice
14:29:05	CX0-1	Výchozí poloha komory

Obr. 20 Základní obrazovka aplikace

7.6 Obrazovka servis

Panel servis umožňuje obsluhu sledovat aktuální stav věch ventilů, měrek a čidel. Po přihlášení údržby dovoluje přechod do manuálního režimu řízení stanice, tento režim je užitečný při diagnostice závady nebo údržby stanice, a proto jsou v něm povoleny některé zakázané stavy ventilu, při kterých může dojít k poškození stanice, a proto by tyto operace měl provádět jen kvalifikovaný personál. Systém sice obsahuje ochranu proti nesprávné kombinaci ventilů, i přes to některé kombinace jsou povoleny.

Panel dále obsahuje záložky na levé straně, tyto záložky představují rozdělení zapojení vakuo-pneumatického schématu odpovídající reálnému zapojení jednotlivých komponent. Třetí záložkou jsou díly, na této záložce můžeme sledovat jednotlivé komponenty stanice a jejich aktuální stav (počet cvaknutí, motohodiny) před potřebným servisem nebo výměnou.



Obr. 21 Obrazovka manuálního ovládání stanice

7.7 Obrazovka recept

Panel recept Obr. 22, slouží k nastavování parametrů pro automatický heliový test. V levé části si můžeme povšimnout tabulky s jednotlivými parametry, které jsou uloženy v PLC. Jednotlivé parametry je možné editovat jednoduchým kliknutím na požadovaný parametr a v pravé části je jej lze přepsat a pro jeho potvrzení zmáčknout tlačítko „Zapiš novou hodnotu“. Tímto se provede editace parametru v paměti PLC, pro jeho uložení do databáze je třeba zmáčknout tlačítko „Ulož recept“. Pro vytvoření nové metody upravíme parametry v tabulce, tak jak potřebujeme a potvrdíme tlačítkem „Ulož jako nový recept“. Nová metoda se pak zobrazí v comboboxu vpravo dole, kde lze metody měnit.

Název	Hodnota
P_MAX_VacuumPart	40.0
P_MAX_VacuumPartCleaning	100.0
P_MAX_ProductHeTolerance1	50.0
P_MAX_ProductHeTolerance2	200.0
P_SET_ProductHePressure1	500.0
P_SET_ProductHePressure2	6000.0
P_SET_HEPressSnifferTest	3000.0
T_SET_HeliumMeasurement	7S
T_MAX_RoughingProduct	16S
T_MAX_RoughingChamber	2M
T_MAX_HeliumCharging	5S
T_MAX_Autocycle	3M
T_MAX_RoughingHLDvacuum	2M
H_MAX_TestLimit	8.4e-4
H_MAX_SnifferTestLimit	1.0e-4
H_MAX_Background	5.0e-5
H_MAX_ContaminationChamber	1.0e-4
K_SET_MethodName	'DEFAULT-LT'
K_SET_LowPressureCoef	8.6
K_SET_TagType	0

'DEFAULT-LT'

Maximální doba čerpání výrobku

16S

Zapiš novou hodnotu

Ulož recept

Ulož jako nový recept

Vybrat Metodu

Obr. 22 Obrazovka správy metod

7.8 Obrazovka parametry stanice

Panel parametry slouží k editaci parametrů stanice. Jeho nastavení probíhá analogicky jako u panelu recept.

Název	Hodnota
P_MAX_ATM	1350.0
P_MAX_RoughingChamber	10.0
P_MAX_HLD	1.0
P_MAX_Sniffer	5.0
P_MAX_VacuumMixture	100.0
P_MIN_ATM	700.0
P_MIN_MixVessel	8000.0
P_SET_RoughingPurge	500.0
P_SET_MixtureFinalPressure	9000.0
T_MAX_BackgroundMeasurement	7s
T_MAX_InitializationHLD	16m
T_MAX_StationInitialization	3h18m20s125ms
T_MAX_GoToStandby	60s
T_MAX_DoorMovement	8s
T_MAX_Tooling	5s
T_MAX_VentingProduct	30s
T_MAX_VentChamber	30s
T_MAX_HLDStaticCalibration	7m
T_MAX_GasNotPresent	10s
T_MAX_SnifferTest	10m
T_MAX_TagRead	10s
T_MAX_SnifferPreparation	15s
T_MAX_MixturePreparationInit	4M

Obr. 23 Obrazovka parametrů stanice

7.9 Obrazovka výsledky

Obrazovka výsledky zprostředkovává, výsledky měření uložené v databázi. Každý z výsledků obsahuje časový záznam a ID operátora, který daný výsledek změřil. Ve výsledcích lze snadno vyhledávat, buď volbou data v pravé části obrazovky, nebo podle uživatele.

Datum	Čas	Celkový čas[s]	Komora	Vakuum#0[mbar]	Vakuum#1	Vakuum#2	Tlak#1[mbar]	Tlak#2	Pozadí He[mbar.l/s]	Leak He#1	Leak He#2	Výsledek	Kód chyby
14.05.2017	14:42:55	45	1	0.010	0.012	0.500	486	5870	0.003	0.001	0.003	OK	0
14.05.2017	14:42:52	45	1	0.010	0.012	0.500	486	5870	0.003	0.001	0.003	OK	0
14.05.2017	14:42:50	45	1	0.010	0.012	0.500	486	5870	0.003	0.001	0.003	OK	0
14.05.2017	14:42:48	45	1	0.010	0.012	0.500	486	5870	0.003	0.001	0.003	OK	0
14.05.2017	14:39:22	40	1	0.010	0.012	0.500	500	5998	0.003	0.003	0.003	OK	0
14.05.2017	14:39:18	40	1	0.010	0.012	0.500	500	5998	0.003	0.003	0.003	OK	0
14.05.2017	14:39:16	40	1	0.010	0.012	0.500	500	5998	0.003	0.003	0.003	OK	0
14.05.2017	14:39:13	40	1	0.010	0.012	0.500	500	5998	0.003	0.003	0.003	OK	0
14.05.2017	13:17:59	40	1	0.010	0.012	0.500	500	5998	0.003	0.003	0.003	OK	0
14.05.2017	04:10:47	40	1	0.010	0.012	0.500	500	6120	0.003	0.003	0.004	OK	0

Obr. 24 Obrazovka výsledků

7.10 Správa a přihlášení uživatelů

Pop-up okno přihlásit, je vždy přizpůsobeno podle aktuálního uživatele, který je přihlášen. Je-li přihlášen operátor, jsou funkce omezeny pouze na přihlášení a odhlášení uživatele. Po přihlášení administrátora nebo nějakého uživatele s prověrkou větší než dva je možné jednotlivé uživatele editovat, vytvářet anebo zobrazit okno s údaji o jejich přihlášení a odhlášení k systému.

The screenshot shows a software interface with a main control panel on the left and a login dialog box in the center. The main panel has tabs for 'MĚŘENÍ', 'SERVIS', 'RECEPT', 'PARAMETRY', 'VÝSLEDKY', and 'Vývoj'. It displays data for three chambers (KOMORA 1, 2, 3) and 'AKTUÁLNÍ DATA'. The login dialog box is titled 'Přihlášení uživatelů' and contains a table of user login records and input fields for user selection and password.

Uživatel:	LogInTime	LogInDate	LogOutTime	LogOutDate	FirstName	LastName
03:44:33	2017-05-14	03:45:22	2017-05-14	LT	DefaultOP	
03:46:30	2017-05-14	03:47:34	2017-05-14	LT	DefaultOP	
03:54:09	2017-05-14	03:56:11	2017-05-14	LT	DefaultOP	
04:10:45	2017-05-14	04:11:04	2017-05-14	LT	DefaultOP	
13:15:59	2017-05-14	13:19:06	2017-05-14	LT	DefaultOP	
13:25:38	2017-05-14	13:25:55	2017-05-14	LT	DefaultOP	
13:27:02	2017-05-14	13:27:22	2017-05-14	LT	DefaultOP	
13:34:01	2017-05-14	13:34:08	2017-05-14	LT	DefaultOP	
13:36:01	2017-05-14	13:36:09	2017-05-14	LT	DefaultOP	
13:38:05	2017-05-14	13:38:41	2017-05-14	LT	DefaultOP	
13:45:48	2017-05-14	13:46:24	2017-05-14	LT	DefaultOP	
13:51:07	2017-05-14	13:51:27	2017-05-14	LT	DefaultOP	
13:52:19	2017-05-14	13:52:44	2017-05-14	LT	DefaultOP	
13:53:20	2017-05-14	13:53:34	2017-05-14	LT	DefaultOP	
13:54:01	2017-05-14	13:54:17	2017-05-14	LT	DefaultOP	
14:14:18	2017-05-14	14:14:39	2017-05-14	LT	DefaultOP	
14:15:09	2017-05-14	14:15:43	2017-05-14	LT	DefaultOP	
14:16:30	2017-05-14	14:16:44	2017-05-14	LT	DefaultOP	
14:28:15	2017-05-14	14:28:21	2017-05-14	LT	DefaultOP	
14:29:03	2017-05-14	14:41:42	2017-05-14	LT	DefaultOP	

Obr. 25 Obrazovka přihlášení uživatelů

7.11 Komunikace s PLC

Pro komunikaci s PLC byla zvolena, z časových důvodů, knihovna S7 .NET. Její provedení ovšem neodpovídá průmyslovému standardu, totéž by se dalo říci i o bezpečnosti a poruchovosti. Avšak pro komunikaci s PLC od společnosti Siemens je plně funkční a pro potřeby této práce zcela postačuje. Její hlavní výhodou je snadné použití a implementace do samotného kódu aplikace. Fakt, že tato knihovna je napsána zcela v jazyku C#, umožňuje debugging kódu a tak velice snadné odhalení chyby při její implementaci.

Mezi podporované typy PLC patří např.: S7-300, S7-1200 a S7-1500. Využívá standardu Ethernet, takže je i velice snadné se k danému PLC připojit bez jakéhokoliv nastavování OPC serveru apod. Jediné nastavení, které je třeba provést je v projektu pro PLC a to je povolení plného přístupu k PLC (no protection), a u jednotlivých data bloků mít zakázanou optimalizaci přístupu. Jsou-li tyto dvě věci splněny, stačí pro připojení k PLC použít jeden příkaz a to:

```
var plc = new Plc(CpuType.S71200, "192.168.63.78", 0, 1);
```

Prvním parametrem v konstrukturu vybíráme typ PLC ke kterému se chceme připojit, druhým parametrem je IP adresa na síti pod kterou dané PLC lze najít a poslední dva parametry představují „rack“ a „slot“. Tímto příkazem vytvoříme nový objekt pro komunikaci s naším PLC. Pro navázání komunikace je potřeba použít příkaz:

```
ErrorCode er = plc.Open();
```

Tento příkaz se pokusí navázat komunikaci s PLC a v případě neúspěchu vrátí chybovou hlášku. Pro celkové čtení data bloků v PLC nám knihovna S7 .NET poskytuje příkaz:

```
plc.ReadClass(sen, 2);
```

Tímto způsobem můžeme přečíst celý data blok jedním příkazem, prvním parametrem je objekt veřejné třídy, která obsahuje proměnné přesně poskládaný tak jak jsou uloženy v paměti PLC. Analogicky k tomuto příkazu existuje příkaz pro zápis do paměti a tak touto jednou třídou můžeme data z PLC číst i zapisovat.

Komunikace sama o sobě představuje poměrně náročnou činnost pro jedno vlákno a tak za použití třídy Thread, bylo pro komunikaci vytvořeno nové vlákno, které obstarává pouze tento proces. Docílilo se tak plynulosti aplikace i komunikace s PLC. Pro práci s tímto vláknem je zapotřebí specifikovat metodu, která bude v tomto vlákně vykonávána. Za tímto účelem byla vytvořena metoda S7communication, jenž obsahuje všechny výše zmíněné příkazy, ošetření výpadku a navázání komunikace s PLC. Metoda dále pro neustálou výměnu dat obsahuje nekonečnou smyčku, kterou lze přerušit ukončením aplikace a výpadkem komunikace. Příkaz pro vytvoření této komunikace vypadá následovně:

```
var com = new Thread(new ThreadStart(S7communication));
```

Ve smyčce se provádí standardní výměna dat, na začátku se přečtou všechny požadované data bloky a na konci se zapíše do potřebných bloků. V mezičase dochází k získávání dat od aplikace, ty které se mají zapsat do PLC a konverze těch, jenž se mají zobrazit v uživatelském prostředí, popřípadě uložit do databáze.

8 Realizace databázového systému

Pro realizaci databázového systému byl zvolen objektový koncept. Tím je myšleno, že celou databázi pro výslednou aplikaci lze chápat jako celek, který se rozdělí na menší části, v tomto případě třídy. Musíme si na začátku uvědomit, co vlastně chceme s databází dělat, jaké úkony s ní potřebujeme vykonávat. Hlavním úkolem databáze je samozřejmě ukládat data, číst je a také je čas od času upravit. Tyto úkony potřebujeme vykonávat nad všemi tabulkami bez výjimky, bylo by poměrně obtížné napsat dostatečně univerzální funkce pro tuto práci se všemi tabulkami. Proto je snazší vytvořit pro každou tabulku její vlastní třídu, do které implementujeme tyto funkce přímo na míru. Tyto funkce se sice budou analogicky opakovat, avšak každá tabulka potřebuje svůj vlastní unikátní přístup a tímto docílíme požadovaného výsledku.

Jak již bylo patrné z databázového modelu, celá databáze obsahuje dvanáct tabulek které je zapotřebí editovat. Podíváme-li se pozorněji a zamyslíme se nad relacemi mezi jednotlivými tabulkami, zjistíme, že práce s některými tabulkami jsou zcela závislé na práci s jinými. Není tedy zapotřebí vytvářet stejný počet tříd jako je tabulek, vystačíme si s menším počtem. Pro vytváření vlastní třídy pro tabulku je jasně patrné, že musí být bez relací anebo musí mít stěžejní funkci pro databázi.

8.1 Implementace SQLite

SQLite stejně tak jako každý databázový systém, potřebuje fyzický soubor, se kterým bude pracovat. Proto při spuštění aplikace je nutné tento soubor vytvořit nebo zkontrolovat zdali není již vytvořen. Jakmile je tento soubor vytvořen je nutné s ním navázat spojení. V C# to znamená, že je vytvořen nový objekt, který je stěžejním parametrem pro práci s databází a je vlastně obsažen v každém příkazu pro čtení, zápis atd. Jelikož jsme se rozhodli pro každou tabulku vytvořit vlastní třídu, úplně se samo nabízí vytvořit další třídu jenom pro připojení databáze, jejímiž potomky budou třídy právě pro práci s těmito tabulkami. Nová třída tedy bude obsahovat konstruktor s případným vytvořením nového souboru „db“ a navázáním komunikace. Jelikož se jedná o rodičovskou třídu všech tabulek, je zde vhodné i umístit společné univerzální funkce pro všechny tabulky.

Jak již bylo nastíněno dříve, jedna z tabulek bude mít proměnný počet sloupců. Také z pohledu celkové funkcionality programu by bylo vhodné, aby se program sám inicializoval na požadovanou stanici a byl tak univerzální pro více stanic. Proto do této třídy byly zahrnuty metody sloužící k čtení textových souborů s parametry stanice. Tyto metody předávají univerzální datový typ, ke kterému jsou v následných odvozených třídách vytvořeny konverzní metody.

Tento textový soubor obsahuje potřebná relevantní data získaná z projektu vytvořeného v prostředí TiaPortal. Mezi tyto informace patří název proměnné, datový typ a offset. Je pravdou, že tyto informace v tomto provedení nemají žádný hlubší význam, ale předpokládá se další rozšiřování programu a pro komunikaci s OPC jsou tato data zapotřebí a tak je tedy dobré již s nimi pracovat a ukládat je.

8.2 Vytváření tabulek

Jak již bylo dříve zmíněno, konstruktor je vhodný kandidát pro vytváření tabulky a její inicializaci. Každá tabulka v databázi je vytvářena stejným, příkazem, který zároveň kontroluje její existenci. Jedná se o příkaz:

```
CREATE TABLE IF NOT EXISTS MethodEdit (  
  IDMethodEdit INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
  Time TIME,  
  Date DATE,  
  IdOp INTEGER REFERENCES Operator (IDOperator) NOT NULL,  
  IdMethod INTEGER REFERENCES MethodsValues (IDmethodsValues) NOT NULL)
```

Uvedený příklad ukazuje, jak je vytvářena tabulka MethodEdit jenž se skvělým příkladem použití všech podstatných parametrů. Prvním sloupcem je unikátní symbol pro daný záznam, druhý a třetí je časový záznam, a poslední dva jsou cizí klíče. Tyto parametry odkazují na tabulky Operator a MethodValue.

9 Závěr

Cílem práce bylo vypracování několika dílčích bodů pro vizualizační software užitý ve stanicích pro kontrolu těsnosti. Prvním bodem bylo popsat základní principy a požadavky finální kontroly kvality těsnosti ve výrobě. Tyto principy byly vztaženy ke zkoušce heliem, jenž je řešením této práce. Druhým bodem bylo provést rešerši SCADA systémů pro tvorbu vizualizačního softwaru, mezi tento software se řadí světový výrobci, jako jsou Wonderware, Siemens a další, tento bod byl podrobně rozebrán a jednotlivé softwary byly porovnány.

Dalším bodem zadání bylo navržení databázového modelu pro skladování naměřených dat, spolu s dalšími klíčovými informacemi aplikace, jako řešení toho problému byl představen ER diagram s relacemi mezi entitami. Tento model představuje větší část finálního databázového systému, avšak pro reálné použití mu chybí mírné úpravy, především tabulka o záznamu s kalibracemi. Dalším úkolem bylo porovnat několik databázových systémů a vybrat neoptimálnější řešení, vybraným systémem je knihovna SQLite jenž je zadarmo a díky své jednoduchosti a rychlosti se jeví jako neoptimálnější kandidát. Tato knihovna se v praxi také skvěle osvědčila. V neposlední řadě byl načrtnut návrh realizace této databáze, bohužel z časových důvodů tento bod nebyl zcela dokončen a mé své mezery. Nicméně její praktická realizace se povedla a je uvedena v příloze. Představený koncept, rozdělení do tříd, funguje a je i dále rozšiřovatelný, ale jistě se časem ukáží jisté mouchy, které bude potřeba vyladit. Předposledním bodem zadání bylo celou aplikaci realizovat, tento bod byl splněn, i když výsledná grafika není konečným produktem, který byl nasazen v reálném použití, její funkčnost je však stálá. Posledním bodem zadání bylo ověřit a vyhodnotit výsledná řešení.

Ověření funkčnosti proběhlo a aplikace plní všechny popsané funkce, je ovšem pravdou že při delším používání se občas objeví nějaká neošetřená výjimka a program se zasekne. Nicméně dílčí části jsou vždy funkční. Celkové řešení má své mouchy a na programu je třeba stále pracovat, dalo by se říct, že je teprve v plenkách své finální podoby.

10 Seznam literatury

- [1] Klientské rozhraní aplikací SCADA [online]. České vysoké učení technické v Praze Fakulta elektrotechnická, 2006 [cit. 2016-11-26]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/balasi2_2007dipl.pdf. Diplomová práce. ČVUT. Vedoucí práce Jaroslav Baláš.
- [2] Wonderware InTouch [online]. s. 16 [cit. 2016-11-27]. Dostupné z: http://www.pantek.cz/pdf/produkty/intouch/intouch_dtsh.pdf
- [3] Moravské přístroje [online]. [cit. 2016-11-27]. Dostupné z: <http://www.mii.cz/art?id=380&cat=146&lang=405>
- [4] Pantek [online]. [cit. 2016-11-27]. Dostupné z: <http://www.pantek.cz/produkty/intouch/>
- [5] Control Web 7. Automa [online]. 2015(3), 3 [cit. 2016-11-29]. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/53552.pdf
- [6] Reliance [online]. [cit. 2016-11-30]. Dostupné z: <https://www.reliance.cz/cs/products/reliance4-scada-hmi-system#page=structure>
- [7] HROMEK, JIŘÍ. KOMUNIKACE OPC SERVERŮ SE SYSTÉMEM MES (COMES). 2013. DIPLOMOVÁ PRÁCE. VUT Brno. Vedoucí práce Ing. JAN PÁSEK, CSc. [cit. 2016-11-29]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=64236
- [8] GAJZLER, Jaroslav. KOMUNIKACE OPC SERVERŮ SE SYSTÉMEM MES (COMES). 2006. DIPLOMOVÁ PRÁCE. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Doc. Ing. František Hruška, Ph.D. [cit. 2016-11-29]. Dostupné z: http://digilib.k.utb.cz/bitstream/handle/10563/2300/gajzler_2006_dp.pdf?sequence=1
- [9] Komunikace přes standardní rozhraní OPC. MICROSYS, spol. s r.o. PROMOTIC [online]. [cit. 2016-12-04]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPC.htm>
- [10] Model 979C Helium Mass Spectrometer: OPERATIONS MANUAL [online]. In: . s. 102 [cit. 2016-12-12]. Dostupné z: <http://www.agilent.com/cs/library/usermanuals/public/6999-09-980A%20Model%20979C%20Helium%20Mass%20Spectrometer%20Leak%20Detector%20Operations%20Manual.pdf>
- [11] PROMOTIC SCADA visualization software [online]. [cit. 2016-12-13]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsPromotic.htm>
- [12] Siemens: WinCC flexible [online]. [cit. 2016-12-13]. Dostupné z: <http://stest1.etnetera.cz/ad/current/index.php?ctxnh=b797d89395&ctxp>
- [13] OPC Unified Architecture pro vzájemnou součinnost zařízení [online]. 2006(12) [cit. 2017-01-03]. Dostupné z: http://automa.cz/cz/casopis-clanky/opc-unified-architecture-pro-vzajemnou-soucinnost-zarizeni-2006_12_31518_506/

- [14] HERNANDEZ, Michael J. *Návrh databází*. Přeložil Jan Bouda. 1. vyd. Praha: Grada, 2006. 408s. ISBN 80-247-0900-7
- [15] MySQL [online]. [cit. 2017-01-05]. Dostupné z: <http://dev.mysql.com/doc/refman/5.7/en/>
- [16] Oracle [online]. [cit. 2017-01-05]. Dostupné z: <http://docs.oracle.com/database/122/index.htm>
- [17] PostgreSQL [online]. [cit. 2017-01-05]. Dostupné z: <https://www.postgresql.org/about/history/>
- [18] SQLite [online]. [cit. 2017-01-05]. Dostupné z: <https://sqlite.org/about.html>
- [19] MALÝ, Martin. Softwarové licence: úvod pro obyčejné lidi. In: Zdrojak.cz [online]. 2011 [cit. 2017-01-05]. Dostupné z: <https://www.zdrojak.cz/clanky/softwarove-licence-uvod-pro-obycejne-lidi/>
- [20] MICROSOFT: *Introduction to WPF* [online]. [cit. 2017-04-22]. Dostupné z: <https://msdn.microsoft.com/en-us/library/aa970268%28v=vs.100%29.aspx>
- [21] MICROSOFT: *Data Binding Overview* [online]. [cit. 2017-04-22]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms752347\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms752347(v=vs.110).aspx)
- [22] MICROSOFT: *Overview of the .NET Framework* [online]. [cit. 2017-04-22]. Dostupné z: <https://msdn.microsoft.com/en-us/library/zw4w595w%28v=vs.110%29.aspx>
- [23]

11 Seznam použitých zkratek

COM – Common Object Model
DCOM – Distributed COM
HMI – Human Machine Interface
HW – Hardware
OPC – OLE for Process Control
OLE – Object Linking and Embedding
PC – Personal Computer
PLC – Programmable Logic Controller
SCADA – Supervisory Control and Data Acquisition
DCS - Distributed Control System
LAN - Local Area network
MSLD - Mass Spectrometer Leak Detector
RTU - Remote Terminal Unit
MTU - Main Terminal Unit
I/O – Input/Output
ODBC - Open Database Connectivity
ADO - Active Data Objects
LPT - Line Printer Terminal (paralelní port)
USB - Universal Serial Bus
SVG – Scalable Vector Graphics
PK – Primary Key (Primární klíč)
FK - Foreign Key (Cizí Klíč)
DBMS – Database Management System
XAML - Extensible Application Markup Language
GDI - Graphics Device Interface
CLR - Common Language Runtime
CIL - Common Intermediate Language
CTS - Common Type System