

**VYSOKÁ ŠKOLA CHEMICKO – TECHNOLOGICKÁ
v Praze**

**FAKULTA CHEMICKO-INŽENÝRSKÁ
Ústav počítačové a řídicí techniky**



VŠCHT PRAHA

**On-line databáze provozních dat pro laboratorní
odparku**

DIPLOMOVÁ PRÁCE

Jan Polnický

**Praha
2001 / 2002**

Souhrn

V rešeršní části je zpracována problematika na téma real-time databází pro sběr a archivaci procesních dat a jsou představena známá řešení z průmyslu. Stručně jsou charakterizovány v této práci využité produkty firem ProjectSoft a Oracle, dále také nástroje pro získávání znalostí z archivovaných dat. Dále je popsán použitý technologicko-informační systém pro laboratorní odparku včetně struktury měřících a řídicích okruhů i komunikační architektury. Navržená struktura relační databáze pro archivaci dat a řízení provozu odparky byla otestována v provozních podmínkách. Byla také vytvořena komplexní aplikace pro on-line výpočet neznámé veličiny z veličin měřených a tabelovaných dat uložených v databázové struktuře.

Klíčová slova: odparka, počítačové řízení, archivace procesních dat, real-time databázový systém, relační model, on-line výpočty

Název práce: On-line databáze provozních dat pro laboratorní odparku

Vedoucí práce: Doc. Ing. Miloš Kmínek, CSc.

Konzultanti: Ing. Pavel Burian, CSc.

Ing. Zdeněk Kokoška

Vypracoval: **Jan Polnický**
(email: jan.polnicky@vscht.cz)

Prohlašuji, že jsem tuto práci vypracoval sám a všechny použité zdroje jsem uvedl v seznamu použité literatury.

V Praze dne

Podpis

Poděkování

Rád bych na tomto místě poděkoval vedoucímu práce doc. Kmínkovi, konzultantům Ing. Burianovi a Ing. Kokoškovi, ale také Ing. Šindelářovi, M. Končelovi a A. Pavelkovi za jejich cenné rady a připomínky. Dále také D. Vojtové, I. Polnické za korektury textu a své rodině za podporu nejen při studiu.

Obsah

1	Rešerše na téma komerční databázové systémy se zřetelem na jejich využití k řízení	6
1.1	Databázové systémy v průmyslové automatizaci	6
1.1.1	Porovnání klasických a real-time databází	8
1.1.2	Databázová rozhraní	9
1.1.2.1	Rozhraní API	10
1.1.2.2	Microsoft - ODBC	10
1.1.2.3	Microsoft - OLE DB	12
1.1.2.4	Sun Microsystems - JDBC	13
1.1.2.5	Borland / Inprice - BDE	14
1.1.3	Řešení z průmyslu	15
1.1.3.1	Řešení Wonderware - FactorySuite 2000 - IndustrialSQL Server	16
1.1.3.2	Řešení OSI Software – systém PI™	18
1.1.3.3	Řešení Intellution – iFIX SCADA a iHistorian	20
1.1.3.4	Řešení CI Technologies – systém Citect a Plant2Business	22
1.1.3.5	Řešení National Instruments – systém Lookout™ a databáze Citadel	23
1.1.3.6	Řešení Siemens – systém myAMC®	23
1.1.3.7	Řešení Coral s.r.o. – systém TIRS®	24
1.1.3.8	Řešení Amit s.r.o. - DB-Net	25
1.1.3.9	Řešení GeoVap s.r.o. – systém Reliance	25
1.1.3.10	Řešení Medium Soft a.s. - Vidium	26
1.1.3.11	Řešení Microsys – Promotic 2000	27
1.1.3.12	Řešení IpeSoft – D2000 Enterprise Production Systems	27
1.1.3.13	Řešení Schneider-electric – OFC (OPC Factory Server)	28
1.1.3.14	Závěr - porovnání	28
2	ProjectSoft – TomPack	29
3	Oracle®	31
3.1	Databázový systém Oracle9i	31
4	Úvod do jazyka SQL	33
5	Dolování dat (Data-mining)	34
6	Představení technologicko-informačního systému pro odparku	35
6.1	Technický popis vlastní odparky	35
6.2	Technický popis měřících a řídicích okruhů	36
6.3	Technický popis struktury komunikace	38
7	Volba, návrh a popis struktury databáze pro sběr procesních dat	39
7.1	Technický popis použitých serverů	39
7.2	Struktura databáze	40
8	Návrh základního programového vybavení	44
8.1	Nastavení systému pro přístup do databáze	44
8.1.1	Pro Oracle-nativní aplikace (SQL*Net)	44
8.1.2	Pro aplikace třetích stran	46
8.2	Programové vybavení pro administraci a obsluhu db	48
8.2.1	Oracle Enterprise Manager	48
8.2.2	Oracle SQL*Plus	49
8.2.3	SQL-Programmer	50
8.3	Programové vybavení pro operace s daty	51
8.3.1	Oracle Discoverer	51

8.3.2	Matlab - Database Toolbox.....	53
8.3.3	Program TomPack jako databázový klient	54
9	Aplikační programy	55
9.1	On-line odhad koncentrace roztoku sacharózy	55
9.1.1	Zadání a teoretické řešení	55
9.1.2	Praktické řešení	57
9.1.3	Neočekávané problémy při experimentálním ověřování funkčnosti.....	61
10	Závěr	65
	Seznam obrázků v textu.....	66
	Seznam tabulek v textu.....	67
	Význam použitých zkratk a pojmů	69
	Literatura.....	71
	Přílohy.....	73
❖	Příloha 1: Obsah vloženého CD-ROMu	73
	Použití:	73
	Příloha 2: Tabulka zvýšení bodu varu v závislosti na hmotnostní koncentraci roztoku sacharózy v roztoku ⁵	74
❖	Příloha 3: Tabulka závislosti teploty varu vody na tlaku ³⁴	75
❖	Příloha 4: Možné údaje pro archivaci s označením datového typu	76
❖	Příloha 5: Tabulka porovnání známých řešení z průmyslu a programu TomPack.....	77
❖	Příloha 6: Tabulka ODBC funkcí programu TomPack ³	78
❖	Příloha 7: Vývojové prostředí programu SQL-Programmer.....	79
❖	Příloha 8: Obrazovka operátora – záznam dat do databáze (TomPack)	80
❖	Příloha 9: Obrazovka operátora – komunikace s databází (TomPack).....	81

Teoretická část

1 Rešerše na téma komerční databázové systémy se zřetelem na jejich využití k řízení

Z důvodu vysokých pořizovacích nákladů se dříve informační systémy objevovaly pouze u větších podniků. Byly to převážně systémy PIMS (veškeré zkratky a pojmy použité v tomto dokumentu jsou vysvětleny v seznamu na konci – viz str. 67-70) od AspenTech, Biles&Associates, OSI Software nebo datové záznamníky kategorie SCADA/HMI např. InTouch, FactoryLink, Genesis a další systémy vytvářené na míru. Konfigurace a údržba těchto systémů byla velmi časově i finančně náročná, protože vyžadovala specialisty z dodavatelských firem. Důvodem bylo, že systémy nepoužívaly technologii RDBMS spolu s dotazovacím jazykem SQL, která je celosvětově akceptovaným a otevřeným standardem se zaručeným vývojem a podporou. Střední a menší podniky většinou neměly vůbec zajištěné ukládání důležitých dat o skutečném průběhu výroby.

V dnešním tržním světě je pro zajištění konkurenceschopnosti produktů nezbytným předpokladem analýza technologických neboli procesních dat. Na základě takových analýz je možné zefektivňovat výrobu pomocí řízení s využitím prvků umělé inteligence a tím zvyšovat kvalitu výrobku při optimálním využití všech prostředků vložených do výroby. Mnoho firem se snaží vytvořit komplexní softwarový produkt pokrývající potřeby průmyslové automatizace, jako jsou např. přímé řízení pomocí PC, sběr procesních dat, jejich vizualizace a archivace, asynchronní přístup k těmto datům z jiných klientských aplikací přes intra/internet či diagnostika chybových stavů. Srdcem těchto výrobně-informačních systémů se stávají tzv. **real-time databáze**. Ty ve spolupráci s informačními, plánovacími a administrativními systémy z kategorie ERP^{13, 17} (zajišťujícími zejména financování, lidské zdroje a obchodní logistiku) umožňují vertikální tok informací v podniku – tzv. **totální (globální) automatizaci**.

1.1 Databázové systémy v průmyslové automatizaci

Databázové systémy (DS) reprezentují specifický druh informačního systému skládajícího se ze **systému řízení báze dat (DBMS)** a **databáze (DB)**. *Systém řízení báze dat* určuje předpis/vazbu mezi daty uloženými v *databázi (datovém souboru - skladu)* a umožňuje také přístup a manipulaci s těmito daty.

Tabulka 1.1 Databázové systémy – obecná struktura

Databázový systém (DS)	systém řízení báze dat	relační / hybridní logický datový model
	databáze	číselná část
	(datový soubor - sklad)	multimediální část

Z pohledu vývojáře se databázové systémy rozdělují na **souborové** a **klient/server** systémy³⁰. V případě *souborových databází* aplikace manipuluje prostřednictvím databázových knihoven přímo se soubory v souborovém systému. Tímto způsobem je možno přistupovat nejen k databázovým souborům na lokálním systému, ale i k vzdáleným databázovým souborům na souborovém serveru. V takovýchto databázových systémech se však obtížněji realizuje souběžný přístup více uživatelů, neboť neexistuje žádný proces, který by koordinoval jednotlivé přistupující aplikace. Je přitom vcelku jedno, jestli je celá databáze

uložena v jednom souboru (jako v případě MS Access), anebo jestli je uložena každá tabulka separátně (jako v případě DBase/FoxPro nebo Paradox). Při přístupu ke *klient/serverové* databázi komunikuje program s *procesem*, který databázi obhospodařuje. Tento proces může běžet na stejném počítači (jako klientská aplikace), ale stejně tak i na počítači vzdáleném. Proces je schopen koordinovat přístup jednotlivých uživatelů, takže systém spolehlivěji zvládá víceuživatelský přístup a je výrazně bezpečnější.

Tabulka 1.2 Představitelé souborových a klient/server databázových systémů

Typ databázového systému	Typičtí představitelé
souborový	databázové tabulky MS Access, Paradox, DBase/FoxPro, atd.
klient/server	Oracle8i/9i, MS SQL Server, DB2/Informix atd.

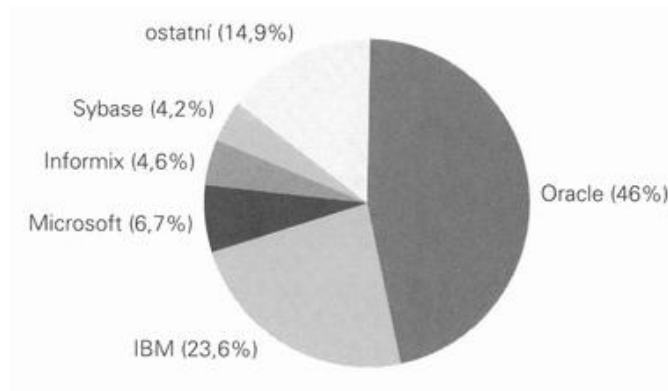
Vlastní **databáze neboli datový sklad** ¹⁹ je jednou z důležitých částí informačního systému pro technologické procesy. Z hlediska **typu dat** jej lze rozdělit na část **číselnou** a **multimediální**. *Číselná část* datového skladu obsahuje údaje z jednotlivých zdrojů. Ke každému údaji lze přiřadit alespoň jednu agregační funkci - předpis, který stanoví jakým způsobem bude údaj dále zpracován. Uspořádání údajů si můžeme představit jako multidimenzionální kostku. K požadovanému údaji se lze dostat specifikací jednotlivých souřadnic. Je tedy jednoznačně určen zdrojem dat, časovou značkou a názvem, přesněji však identifikačním číslem (id), neboť názvy mohou být duplicitní, ale identifikační číslo nikoliv. Vypovídací schopnost údaje je samozřejmě vyšší, než jeho samostatná presentace v čase. Další informace podávají metadata, což jsou data o datech. Funkční model datového skladu lze vytvořit pouze na základě dobré znalosti metadat. Tato část datového skladu bývá nejčastěji programována v jazyce C a Embedded SQL. *Multimediální část* datového skladu není přímou součástí databáze. Typickými údaji jsou textové zprávy (řetězce i pole), obrázky (schémata), audio či video záznamy. Slouží zejména pro potřeby vizualizace.

Z hlediska **funkce** databázový sklad dělíme na modul **vstupní** a **výstupní (dotazovací, presentační)**. *Část vstupní* komunikuje s datovými pumpami, ale zajišťuje také agregaci, archivaci dat apod. *Část výstupní* zajišťuje plnění klientských požadavků z jednotlivých uživatelských PC.

Nejčastěji používané databázové systémy k ukládání dat využívají tzv. **relační model dat** a nesou pojmenování **relační - SQL databáze**. Slouží především k uložení dat v relacích (tabulkách) a k definování vztahů mezi těmito tabulkami ¹⁶.

Tabulka 1.3 Typičtí představitelé relačních databázových systémů

Firma	Databázový systém	Informace na adrese
Oracle	Oracle8i/9i	www.oracle.cz
Microsoft	SQL Server 7/2000	www.microsoft.cz
Sybase	Adaptive Server Enterprise	www.sybase.cz
IBM	DB2, Informix	www.ibm.com



Obrázek 1.1 Celkový podíl na trhu relačních databázových licencí v roce 2000 (zdroj IDC 2001)

Obecně lze říci²², že se tyto klasické relační databáze pro real-time sběr a archivaci dat z technologie nehodí. Přesvědčily se již o tom mnohé firmy, které zkoušely použít relační databáze pro sběr dat v reálném čase. Ani nejvýkonnější počítače s dostatečnou kapacitou paměti nedokázaly zajistit dostatečnou pružnost reakcí databáze. Záleží však na množství dat a četnosti (periodě) archivace. Tzv. **hybridní** databáze jsou rozšířením relačních databází o podporu objektů. Jsou to tedy objektově-relační databáze.

K ukládání **real-time technologických dat** je proto nutné použít tzv. **real-time databázový systém (RTDBS)**, který realizujeme spojením databázového systému a real-time nadstavby. Je to vlastně velice úzce specializovaný software mající za úkol pouze jedinou věc – uložit hodnoty veličin (myšleno číselné, ale např. i logické) v čase tak, aby bylo možné zpětně rekonstruovat jejich průběh s danou přesností¹⁶. V real-time databázi se vše týká veličiny. Před ukládáním vlastních hodnot veličiny v čase je nutné nastavit několik parametrů (rozsah hodnot, jednotka, přesnost ukládání a zdroj veličiny).

1.1.1 Porovnání klasických a real-time databází

Tradiční real-time databázové systémy zpracovávají data ve strukturách závislých pouze na dané aplikaci¹⁵. Požadavek efektivního přístupu ke stále rostoucímu množství informací vyžaduje spravování a uchovávání dat v systematické a organizované podobě. Vyplývá z toho požadavek na sloučení databázové a real-time technologie. Tyto celky nazýváme RTDBS. Jsou to skladiště dat s algoritmy pro jejich ukládání a manipulaci, které musí zajistit určitý stupeň spolehlivosti právě v případě systémových požadavků na časovou odezvu.

Klasické konvenční databázové systémy upřednostňují logickou konzistenci dat jako jediné kritérium správné funkce. K jejich zajištění používají nejčastěji uzamykání dat pro čtení nebo změnu údajů a tím blokují jiné databázové transakce. V takových případech je velmi složité předvídat délku zpoždění, jelikož blokování může být kaskádní - aditivní. RTDBS mají zcela odlišné požadavky na výkonnost a kritéria přesnosti. Hlavním cílem konvenčních databázových systémů je „průměrně“ rychlá odezva, kdežto u real-time systémů musí být transakce vyhodnocovány podle toho, *jak často* promeškávají své kritické termíny, a nebo jaké jsou *náklady* způsobené tzv. **zpožděnými transakcemi** (transakce, které již promeškaly svůj kritický termín). Korektnost real-time databázových transakcí nezáleží pouze na logických výpočtech, ale také na čase, kdy jsou transakce vyřízeny.

Data v RTDBS musí tedy co nejpřesněji odrážet aktuální stav zkoumaného reálného prostředí. Tím, že jsou data ukládána v diskrétních časových intervalech, jsou už vlastně

aproximací reality, která se s ubíhajícím časem stává méně přesnou a až dosáhne určitého bodu (daného absolutním intervalem platnosti I_x), hodnota již neodráží skutečný stav. Charakteristickými vlastnostmi pro data jsou :

- **časová značka** (nejčastěji datum a čas měření)
- **absolutní časová konzistence**, která je splněna, platí-li

$$t_{akt} - t_x \leq I_x \quad (1.1)$$

kde I_x je absolutní interval platnosti (poslední měřenou hodnotu považujeme za platnou), t_x čas poslední měřené hodnoty a t_{akt} aktuální čas.

- **relativní časová konzistence**, spojená s relativním časovým intervalem vztáženým k množině dat

Zpožděná transakce může mít pro daný systém po jejím dokončení pozitivní, negativní či nulový přínos. RTDBS by měl zpožděné transakce ideálně zpracovávat následujícím způsobem. V případě negativního přínosu má systém za úkol podle volných zdrojů jako jsou procesorový čas a volná paměť rozhodnout o prioritách dokončení této a ostatních důležitých transakcí. Nemělo by se stát, že jedna zpožděná transakce zpozdí další „čekající v řadě“. V případě pozitivního přínosu dokončení zpožděné transakce, který by měl být menší než v případě splnění kritického termínu (jinak termínování konkrétních transakcí ztrácí smysl), by se měl systém, pokud je to možné, snažit o včasné dokončení. Může také naopak rozhodnout o snížení její priority, aby se uvolnily systémové zdroje pro dokončení jiných transakcí. Rozhodnutí je vždy závislé na konkrétní aplikaci, na významnosti jednotlivých procesů. Real-time systémy můžeme rozdělit na tzv. **HARD** a **SOFT real-time systémy**. V případě, že si nemůžeme dovolit zpožděné transakce, hovoříme o hard (nepoddajném) a v opačném případě, kdy jsou tolerovány, o soft (poddajném) real-time systému. Zajistit absolutní garanci splnění časových nároků je velmi obtížné, a proto je většina real-time systémů konstruována jako soft systémy.

1.1.2 Databázová rozhraní

Databázové rozhraní je softwarová vrstva mezi databázovým systémem a klientskou aplikací pro obousměrnou výměnu dat, ale i pro ovládání DS. Tedy zprostředkovává přístup k datům anebo umožňuje poskytnutá data zapisovat do databáze. Původním záměrem bylo unifikovat přístup k datům, v dnešní době máme však k dispozici již několik typů těchto rozhraní³⁰.

Tabulka 1.4 Typy databázových rozhraní

Označení rozhraní	Význam zkratky
API	Application Programming Interface
ODBC (Microsoft)	Open DataBase Connectivity
OLE DB (Microsoft)	Object Linking and Embedding for Databases
JDBC (Sun Microsystems)	Java DataBase Connectivity
BDE (Borland / Inprice)	Borland Data Engine

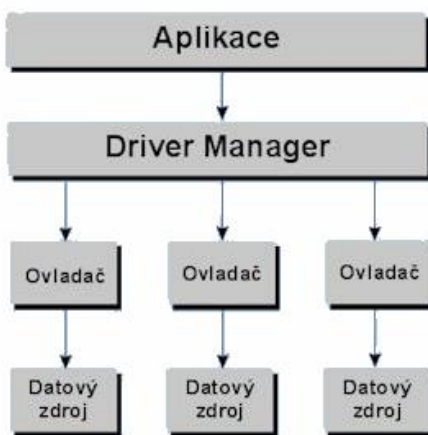
1.1.2.1 Rozhraní API

Aplikační programové rozhraní (API) je pravděpodobně historicky nejstarším způsobem přístupu k datům. Je to vlastně knihovna funkcí (obvykle určena pro překladače C++), která umožňuje pracovat s databází na poměrně nízké úrovni. Základním problémem tohoto rozhraní bylo, že nebylo nijak standardizované. Tedy každý výrobce databázového stroje si vymyslel také svou sadu příkazů pro jeho ovládání. Pokud někdo chtěl vytvořit aplikaci, která by pracovala s více datovými stroji, musel určitou část programu (která přístup k datům obstarávala) implementovat několikrát. I dnes má přístup přes specifické rozhraní API své místo. Je totiž ze všech způsobů nejrychlejší a dokáže využít všech (tedy i velmi specifických) vlastností příslušného databázového stroje. Z tohoto důvodu je k různým vývojovým prostředím (obvykle za velké peníze) možné dokoupit speciální "databázové ovladače", jež zajistí spojení mezi příslušným vývojovým prostředkem a příslušnou databází.

1.1.2.2 Microsoft - ODBC

Prvním a dnes nejpoužívanějším standardem pro přístup k datům je ve světě Windows rozhraní ODBC. Tento standard zavedla firma Microsoft a první implementace byla k dispozici ještě v době 16bitových Windows. Ačkoliv základní architektura zůstala stejná, ODBC bylo od té doby značně vylepšeno. Protože rozhraní ODBC mělo zajistit jednotnou komunikaci s různými datovými zdroji, bylo třeba knihovny ODBC rozdělit na dvě části - na část společnou pro všechny datové zdroje a na část určenou ke komunikaci s konkrétním databázovým zdrojem. Část společná pro všechny datové zdroje se nazývá **ODBC Driver Manager** a komunikaci s databázovým zdrojem zajišťuje **ODBC ovladač (driver)**.

Klientská aplikace komunikuje se společným *jádrem (Driver Managerem)*. Ten zajistí část logiky pro vykonání požadavku a zbytek předá ke zpracování příslušným ovladačem. Provádí také v rámci možností ošetření chyb a může zaznamenávat volání funkcí pro účely ladění. *ODBC ovladač* (neboli driver) je systémová (32-bitová) knihovna naprogramovaná v jazyce C/C++ umožňující klientským aplikacím přistupovat k datovým zdrojům (Data Sources). Řídí veškerou komunikaci se skutečným datovým zdrojem (databázovým strojem). Jeden ovladač může také komunikovat s více datovými zdroji stejného typu. Z toho popisu je také vidět, že ODBC je zcela nezávislé na platformě serveru a na síťovém prostředí.



Obrázek 1.2 Obecná architektura rozhraní ODBC ³⁰

Aby bylo možné jednoduše spravovat jednotlivá spojení na databáze, podporuje ODBC vytváření tzv. „aliasů“. Pod pojmem alias si můžeme představit pojmenované spojení na jeden konkrétní zdroj dat (např. na konkrétní databázi na konkrétním serveru). Z programového kódu se potom můžete odkazovat přímo na toto pojmenované spojení. Jelikož fyzické umístění je specifikováno v nastavení příslušného aliasu (a nikoliv "natvrdo" v programovém kódu), je možné změnit datový zdroj bez zásahů do vlastní aplikace. V omezené míře je také možné změnit typ datového zdroje (viz dále). V nastavení aliasu jsou také další parametry potřebné pro připojení k datovému zdroji - např. jméno a heslo, typ síťového spojení apod. ODBC rozeznává tři typy aliasů - **uživatelské, systémové a souborové**. *Uživatelské a systémové aliasy* jsou uloženy v registrační databázi Windows. Uživatelské aliasy se ukládají v uživatelské větvi registrační databáze a jsou tudíž přístupné jen příslušnému uživateli. Systémové aliasy se ukládají ve větvi "local machine" a jsou tudíž dostupné všem uživatelům (s dostatečným oprávněním) a systému jako takovému (např. službám ve Windows NT). *Souborové aliasy* jsou naproti tomu uloženy v souboru, což usnadňuje jejich přenos mezi počítači.

Jelikož se možnosti jednotlivých datových zdrojů liší, ODBC podporuje pouze jakýsi "průnik" funkcionalit, který je všem datovým zdrojům společný. Jelikož ani tento průnik by dostatečně nevyhovoval velkým i malým databázím, ODBC definovalo tři **stupně funkcionality**, do kterých se mohou ovladače zařadit. Tyto úrovně se nazývají **základní (core), úroveň 1 (level 1) a úroveň 2 (level 2)**. *Základní úroveň* je určena pro jednoduché desktopové databáze. Umožňuje připojování k datovým zdrojům, spouštění SQL příkazů, čtení výsledné množiny (result setu) směrem dopředu, potvrzování a rušení transakcí a základní obsluhu chyb. *Úroveň 1* navíc přidává práci se systémovým katalogem (např. zjišťování dostupných tabulek), zjišťování podporovaných datových typů a skalárních funkcí, průchod výslednou množinou oběma směry, práci s primárními klíči a s uloženými procedurami. *Úroveň 2* umožňuje navíc ještě nastavovat úroveň izolace mezi transakcemi, práci se záložkami anebo asynchronní zpracování jednotlivých příkazů.

Aby byla situace ještě o něco zamotanější, každý ovladač podporuje jinou **úroveň SQL gramatiky**. Opět jsou definovány tři úrovně podpory, tentokrát pojmenované **minimální (minimal), základní (core) a rozšířená (extended)**. *Minimální úroveň* vyžaduje příkazy pro vytváření a rušení tabulek, jednoduché varianty příkazů SELECT, UPDATE a INSERT. Pracuje pouze s datovými typy CHAR, VARCHAR a LONG VARCHAR. *Základní úroveň* přidává práci s indexy, restrukturalizaci tabulek, agregační funkce a práci s uživatelskými právy. Přidává také další numerické datové typy. *Rozšířená úroveň* SQL gramatiky přidává podporu skalárních funkcí, zamykání (SELECT FOR UPDATE) a podporuje datové typy datum a čas.

Ačkoliv bylo rozdělení funkcionality na několik úrovní nutné, přináší to programátorům opět problémy. Některé ovladače totiž nepodporují ani základní funkčnost - např. rušení transakcí u Paradoxu anebo dBase. Jakákoliv fiktivní aplikace, zcela nezávislá na datovém zdroji, bude umožňovat jen velmi primitivní manipulaci s daty. Pokud budete prostřednictvím ODBC ovladačů přistupovat k souborovým databázím (dBase, FoxPro, Paradox apod.), je též třeba si na klientských počítačích "ohlídat" implementaci ovladače, neboť jich existuje mnoho a každý se chová jinak. U velkých databázích našťastí nebývá tolik problémů, protože tyto databázové stroje podporují prakticky všechno.

ODBC Driver Manager, se kterým komunikuje uživatelská aplikace, podporuje všechny tyto funkce. Funkce, jež nejsou implementovány ovladačem, obvykle vracejí prázdné hodnoty. Pro tvůrce informačních systémů je důležitou zprávou, že informace o tom, jak napsat ODBC ovladač, jsou volně k dispozici, takže ODBC ovladače mohou vytvořit nezávislé databázové firmy pro přístup ke svým datům. V zásadě lze ODBC ovladače rozdělit do dvou skupin. Ovladače pro souborové databáze musejí zajistit vykonávání veškerých

příkazů ve vlastní režii. Naopak ovladače pro klient/server databáze pouze zajistí předání příkazu na server a převzetí výsledku ze serveru. Driver také musí zajistit zobrazení okna pro zadávání parametrů pro konfiguraci aliasu a případě okna pro přihlašování k databázi.

Pro manipulaci s daty používá ODBC standardní jazyk **SQL**. Veškerá manipulace s datovým zdrojem je ale realizována prostřednictvím sady funkcí ODBC API (např. pro přímé spuštění SQL příkazu se používá funkce `SQLExecDirect`). Tyto funkce pokrývají přihlašování k datovému zdroji, práci s SQL příkazy, získávání výsledků, práci s transakcemi, práci s kurzory, práci se systémovým katalogem, funkce pro ošetřování chyb a ještě další.

Jedním z nedostatků ODBC je, že neexistuje žádná standardní "objektová nadstavba", která by zajišťovala pohodlnější manipulaci s daty. Pokud jde o nabídku konkrétních ovladačů, je možné konstatovat, že jsou k dispozici prakticky pro každý databázový systém.

Tabulka 1.5 Přehled základních ODBC ovladačů ze sady MDAC od firmy Microsoft

Jméno ovladače	Datový zdroj
MS Access driver	soubory *.mdb
MS dBase driver	soubory *.dbf
MS Excel driver	soubory *.xls
MS FoxPro VFD driver	soubory *.dbf
MS Paradox driver	soubory *.db
MS Text driver	soubory *.txt
MS Visual FoxPro driver	-
MS ODBC for Oracle	RDBMS Oracle
SQL server	RDBMS

1.1.2.3 Microsoft - OLE DB

OLE DB je další rozhraní pro přístup k datům z dílny Microsoftu. Vývojově je výrazně novější než ODBC. Hlavním důvodem, proč bylo rozhraní OLE DB navrženo, byl přístup k **heterogenním datům** a přístup k jiným než relačním datům. OLE DB je také postaveno na komponentovém standardu **COM**, což jej činí flexibilnějším než běžné API. OLE DB je tedy definicí otevřené kolekce rozhraní, které zapouzdřují databázové funkce.

V OLE DB se pracuje s pojmy „**consumer**“ a „**provider**“. Consumer je aplikace využívající přístup k OLE DB komponentám, jako provider se označuje komponenta, která nabízí OLE DB interface a je tudíž schopna poskytovat data. Dále se ještě setkáváme s pojmem servisní komponenta (service component), která implementuje obecnou funkcionalitu OLE DB.

V OLE DB se setkáváme s několika **objekty**. Prvním z nich je objekt „*DataSource*“. Ten zapouzdřuje funkcionalitu OLE DB zdroje jako takového - např. informace o databázovém připojení. Další objekt označovaný jako „*Session*“ zajišťuje zpracování (i paralelní) v rámci jednoho klienta. Jinými slovy, tento objekt je třeba pro oddělení více transakcí v rámci jedné klientské aplikace. Objekt „*Command*“ je abstrakcí pro databázový příkaz - ať už jde o dotaz, příkaz k aktualizaci anebo uloženou proceduru. Výsledkem dotazu je objekt „*Rowset*“, který obsahuje nějaká data v tabulární podobě. Tento objekt se ale také používá pro popis schématu databáze. Poněkud stranou stojí další tři objekty, neboť ty reprezentují infrastrukturu OLE DB. Objekt „*Enumerator*“ slouží ke správě a případně i k vyhledávání datových zdrojů. Ačkoliv objekt „*Session*“ obsahuje částečně podporu transakcí, mohou někteří provideři podporovat také objekt „*Transaction*“, který poskytuje

některé pokročilé funkce. Posledním důležitým objektem je „*Error*“ sloužící pro ošetřování chybových stavů.

OLE DB také zjednodušuje tvorbu providerů (oproti ovladačům u ODBC). Tím, že OLE DB má bohatší strukturu, má tvůrce providerů více možností, kde zvolit hranici mezi obecnou implementací od Microsoftu a tvorbou vlastního kódu. Využití technologie komponent také umožňuje rozšiřování tohoto modelu. To umožní uživatelům přes toto obecné rozhraní využít maximální funkcionalitu datových zdrojů.

Pokud chcete pouze minimální implementaci, stačí implementovat pouze načítání „*Rowsetu*“. Veškerou logiku, která zajistí vyhledávání a třídění, využijete z knihoven OLE DB Simple Provider Toolkit.

Jelikož jde o poměrně nový standard, není ještě k dispozici tolik providerů jako u staršího ODBC. Standardní instalace obsahuje providery pro Microsoft SQL Server, Oracle a Jet (formáty Access, dBase/FoxPro, Paradox a Excel). Další speciální provider umožňuje komunikaci s libovolným ODBC zdrojem. Tento provider tedy zajišťuje, aby bylo možné komunikovat i s datovými zdroji, pro které nebyl zatím vytvořen OLE DB provider. Na rozšíření seznamu providerů se samozřejmě pracuje. Oproti ODBC je důležité, že OLE DB používá při zpracování dat standardně UNICODE. Na rozdíl od ODBC poskytuje OLE DB standardní objektovou nadstavbu **ADO**, která zapouzdřuje základní funkcionalitu pro pohodlnější přístup. ADO objekty pouze zprostředkovávají přístup k OLE DB, ale nenabízejí žádnou novou funkcionalitu (naopak některé "systémové" funkce nejsou přes ADO vůbec přístupné). Tato knihovna je pokračovatelem doposud používaných knihoven DAO a RDO (ty sice stále ještě žijí, ale spíše z důvodů kompatibility, a už se dále nevyvíjejí). Tato knihovna nabízí několik **objektů pro přístup k datům**. Jsou to objekty „*Connection*“, „*Command*“, „*RecordSet*“, „*Parametr*“, „*Field*“, „*Error*“ a „*Property*“. Z názvů objektů již vyplývá, jakou činnost budou zajišťovat. Tato knihovna je k dispozici ve všech vývojových nástrojích Microsoftu - např. Visual Basicu, Visual C++, ale i v ASP stránkách. Bude ale součástí i dalších vývojových produktů od jiných výrobců.

1.1.2.4 Sun Microsystems - JDBC

JDBC je rozhraní pro přístup k datům z prostředí Javy a svojí architekturou velice blízké rozhraní ODBC. Toto rozhraní je stejně jako Java platformově nezávislé. Umožňuje připojování k různým datovým zdrojům z prostředí Javy a prostřednictvím jazyka SQL. Podobně jako u ODBC, i zde existuje společný *JDBC driver manager* a samostatné JDBC ovladače.

JDBC rozeznává čtyři základní typy ovladačů. Prvním typem je tzv. **JDBC-ODBC Bridge**, který překládá volání JDBC na volání ODBC a ta pak zajišťuje příslušný ODBC ovladač. Toto řešení je sice asi nejpoužívanější, ale pro Javu není příliš elegantní. Druhým typem je ovladač, který převádí volání JDBC na volání nativního API příslušného datového zdroje. Opět i zde narážíme na problémy s nativním kódem, kterému se při programování v Javě snažíme vyhnout. Třetí možnost představuje ovladač, jenž volá vzdálenou komponentu na střední vrstvě, která převádí tato volání na volání pro určitý datový zdroj. Tento způsob připojení k datům je asi nejvíce flexibilní a v tomto případě není nutné na straně klienta instalovat žádný nativní kód.

Přístup k databázím z programů v Javě naráží na bezpečnostní problémy. Pokud spouštíte lokálně instalované aplikace v Javě, je vše v pořádku. Pokud ale chcete přistupovat k databázím z appletů, které si uživatel stahuje z Internetu, musíte tyto applety digitálně podepisovat (jinak **Java Virtual Machine** - JVM nedovolí přístup na jiné síťové zdroje nebo do lokálního systému). Pokud bude takový applet komunikovat s databází na vzdáleném serveru, je třeba zajistit také vhodný komunikační kanál na firewallech.

JDBC pracuje s několika rozhraními. Pro manipulaci s ovladači a pro správu databázových připojení se používá rozhraní *java.sql.DriverManager*. Konkrétní spojení na databázi reprezentuje rozhraní *java.sql.Connection*. Rozhraní *java.sql.Statement* se používá pro spuštění SQL příkazů přes vybrané spojení a pro reprezentaci výsledné množiny je použit *java.sql.ResultSet*.

1.1.2.5 Borland / Inprise - BDE

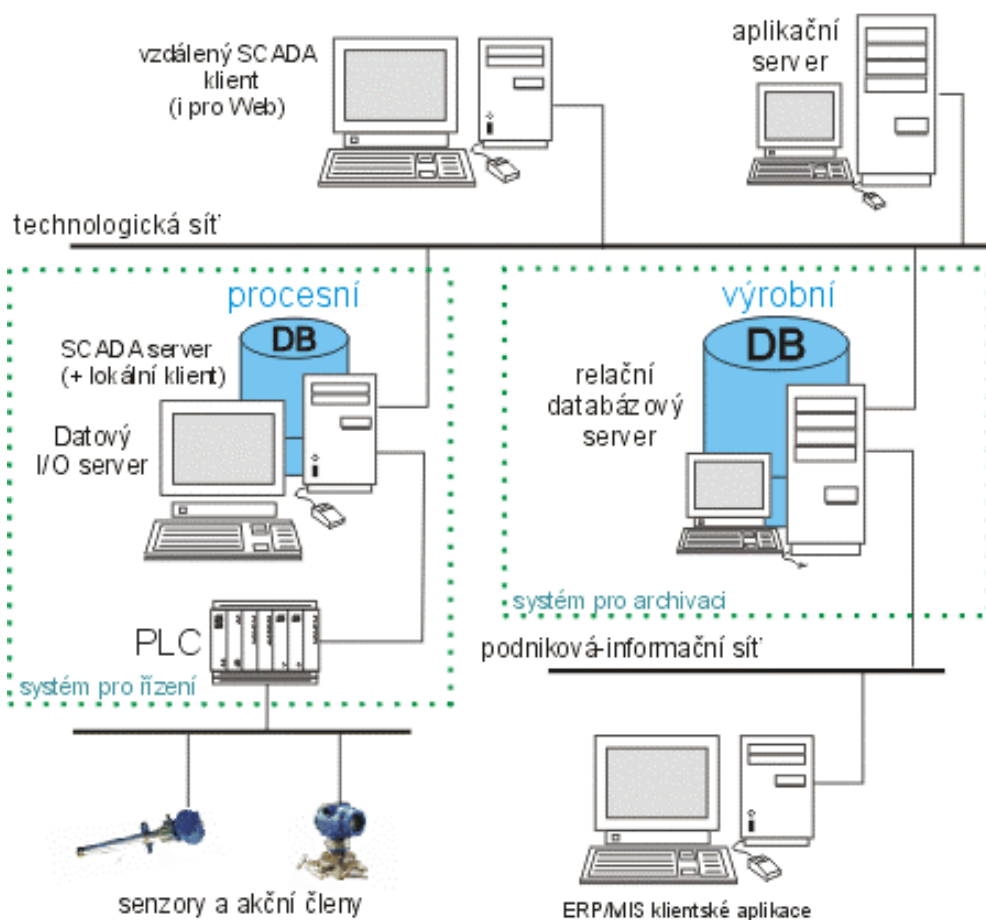
Databázové rozhraní BDE, vyvíjené jako IDAPI (Integrated Database Application Programming Interface), bylo původně určeno jako alternativa k rozhraní ODBC. Firma Borland (nyní Inprise), která toto rozhraní vyvíjela, však neměla dostatečnou sílu ho prosadit pro širší využití. Svůj díl na tom mělo i to, že jeho uvedení bylo dlouho oddalováno a mezeru na trhu rychle vyplnilo ODBC. V současnosti najdete toto rozhraní tedy jen ve vývojových nástrojích firmy Inprise (a v programech v nich vyvinutých) a je vnímáno spíše jako doplněk ODBC.

Myšlenka BDE je v podstatě velmi podobná myšlence ODBC. I zde existuje sdílené jádro BDE, ke kterému se připojují jednotlivé databázové ovladače. Firma Inprise dodává ovladače pro Paradox, dBase (omezeně i FoxPro), Access a SQL servery InterBase, Microsoft, Sybase, Informix a IBM DB2. Též je zde možnost přistupovat na ODBC ovladače.

Ke knihovnám BDE byla také vytvořena objektová nadstavba - Visual Component Library (ta se ale z větší části zabývá jinými než databázovými záležitostmi). Logika této knihovny je obdobná jako u knihovny ADO.

1.1.3 Řešení z průmyslu

Programová řešení pro sběr a archivaci velkých objemů dat z výrobních procesů jsou kritickým místem pro každého výrobce TIS podniku. V průmyslovém prostředí je kladen největší důraz na spolehlivost systému pro řízení a systém archivace dat je až druhořadý. Pokud nastane totiž problém se serverem pro archivaci dat (síťový výpadek, neočekávaná chyba na serveru, atd.), řízení nesmí být žádným způsobem ovlivněno. Proto nejtypičtější je kombinace odděleného **systému pro řízení** uchovávajícího krátkou historii hodnot procesních veličin v tzv. „procesní“ databázi (obvykle lokální souborový DS) se **systémem pro archivaci** pro shromažďování historických dat do tzv. „výrobní“ databáze (obvykle vzdálená relační DS typu klient/server) a jejich další zpracování (analýzy, apod.) klientskými aplikacemi. Lokální databáze totiž umožňuje několikanásobně vyšší rychlosti datových transakcí než externí databáze používané pro archivaci dat.

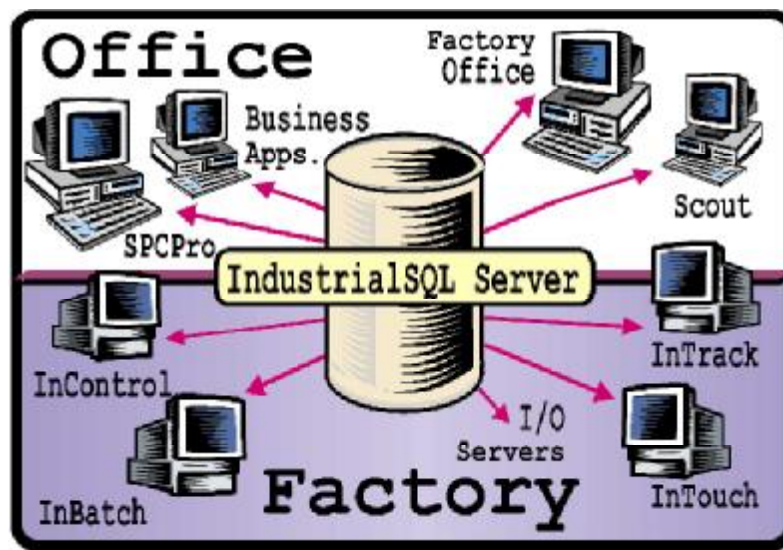


Obrázek 1.3 Zobecněná architektura TIS pro řízení a sběr výrobních dat

Konkrétní typy použitých DS pro systémy řízení (SCADA/HMI) a systémy archivace se svými specifickými parametry jsou obvykle firemním tajemstvím a je velice složité o nich získat jakékoliv informace. Přesto jsem některé zjistil a v dalších odstavcích Vás s nimi seznámím.

1.1.3.1 Řešení Wonderware - FactorySuite 2000 - IndustrialSQL Server

Softwarový produkt **FactorySuite 2000**^{2, 8} americké společnosti **Wonderware Corporation**⁹ komplexně pokrývá potřeby průmyslové automatizace. Tvoří ucelený výrobní informační systém, označovaný jako MMI nové generace, který zajišťuje v reálném čase sběr výrobních dat, jejich vizualizaci, ukládání a poskytování všem zainteresovaným pracovníkům. Doplňuje tak vhodně podnikové systémy z kategorie ERP. Skládá se z několika částí:

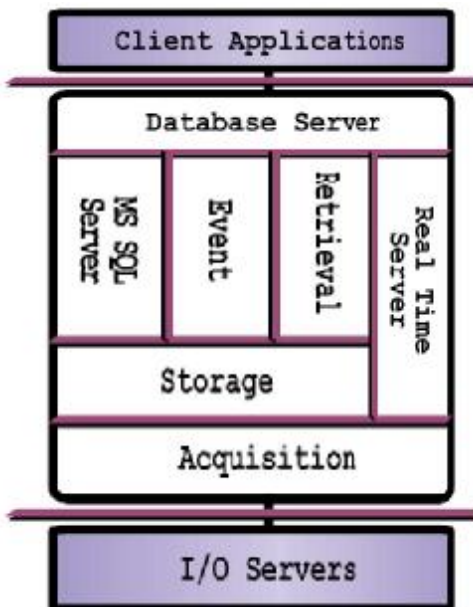


Obrázek 1.4 Architektura produktu FactorySuite, Wonderware⁹

InTouch je 32 bitový objektově orientovaný grafický software pracující na platformě PC pod OS Microsoft[®] Windows[®]. Umožňuje vytváření komplexních aplikací typu SCADA/HMI pro supervizní sledování a řízení procesů.

IndustrialSQL Server (InSQL)^{1, 2, 17, 18} je výkonná a plnohodnotná *databáze* pracující v reálném čase nad jádrem *relační databáze Microsoft[®] SQL Server*, která je součástí licence produktu. Zajišťuje sběr a ukládání procesních dat a poskytování požadovaných aktuálních či historických hodnot všem klientům v rámci podnikové klient-server architektury. Procesní data jsou integrována s údaji o výrobních událostech a se souhrnnými a konfiguračními daty. Pro řádová urychlení přístupu k datům (reálný režim) a k minimalizaci prostoru při bezztrátovém ukládání dat jsou použity vlastní nadstavbové rutiny/algoritmy a technologie OLE DB. Tyto speciální optimalizace jsou nutné pro rychlé zpracování extrémně vysokého počtu datových přenosů, typických pro oblast průmyslové automatizace. Data jsou v databázi uložena ve speciální **adresářové struktuře**. Při dotazu na tyto hodnoty vyvolá InSQL server knihovnu „OLEDB.dll“, která data načte a odešle je klientovi standardní cestou *jako by* byla data vrácena SQL serverem.

InSQL server umožňuje rovněž provést fyzické oddělení počítačové sítě, která doposud mnohdy zpřístupňovala jak technologická, tak obchodní nebo uživatelská data. Začleněním databázového serveru do topologie sítě tak vzniká užitečný mezičlánek propojující dvě nezávislé sítě – technologickou síť ve výrobě a administrativní síť ve zbytku podniku.



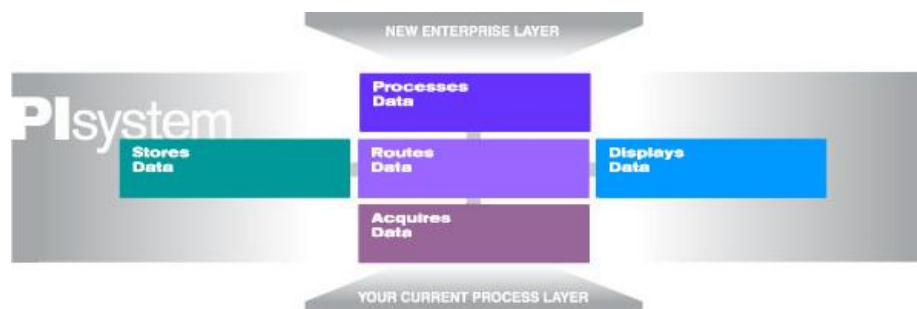
Obrázek 1.5 Architektura IndustrialSQL Serveru – FactorySuite, Wonderware⁹

Otevřenost systému jenž je dána použitím standardu relační databáze MS SQL Server, umožňuje snadnou správu i integraci procesních a administrativních dat v rámci výrobních organizací. K presentaci a dalšímu zpracování dat (analýzy, atd.) lze použít buď samostatné klientské prohlížeče dle vlastního výběru - ať už nativního klienta od firmy Wonderware (InTouch, FactoryOffice) nebo Microsoft (Office, Internet Explorer) nebo jakékoliv jiné aplikace přistupující k datům pomocí standardů SQL / ODBC / OLE DB, vytvořené nezávislymi dodavateli „na míru“ zákaznickým požadavkům (pomocí MS Visual Basic, C++ aj.). Zajímavou vlastností je tzv. rozdílový import konfiguračních databázových proměnných z aplikací InTouch do konfigurační databáze InSQL Server. IndustrialSQL Control ovládá InSQL Server a monitoruje jeho činnost a služby.

InControl je software pro přímé řízení v reálném čase pomocí PC s připojením na nejznámější I/O systémy a sběrnice. Využívá technologie Microsoft OLE Automation a OCX. Jeho programování je v souladu se specifikacemi IEC 1131-3 a RS 274D. **Scout** je sada internetovských nástrojů (server, prohlížeč, InternetDDE, InTouch konvertor) optimalizovaných pro použití s produkty FactorySuite. Manažeři a supervizoři mohou mít prostřednictvím běžného internetového připojení odkudkoliv přístup na vzdáleně provozované technologie. **InSQL Trend**, **InSQL QuickLook**, **InSQLVector** jsou aplikace pro vizualizaci/analýzu procesních dat, ať už historických či aktuálních.

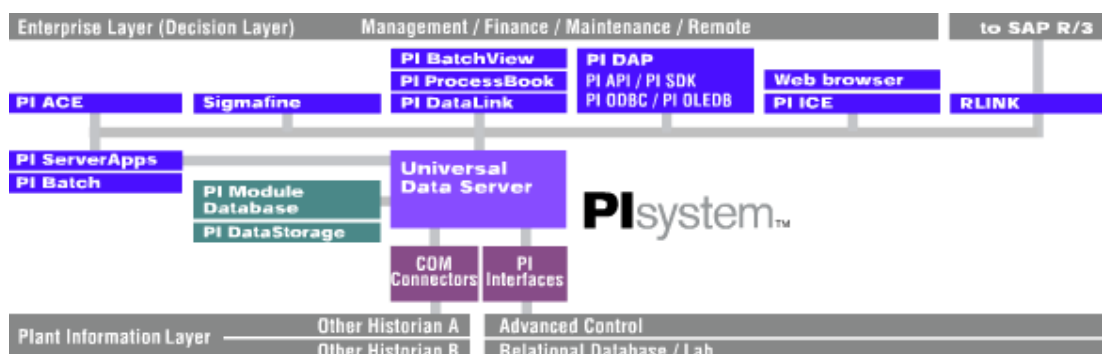
1.1.3.2 Řešení OSI Software – systém PITM

PITM System (Plant Information System) je systém pro správu informací v technologicko-informačních systémech od firmy OSI Software²⁹. Je článkem mezi procesní a podnikovou vrstvou jak znázorňuje následující obrázek.



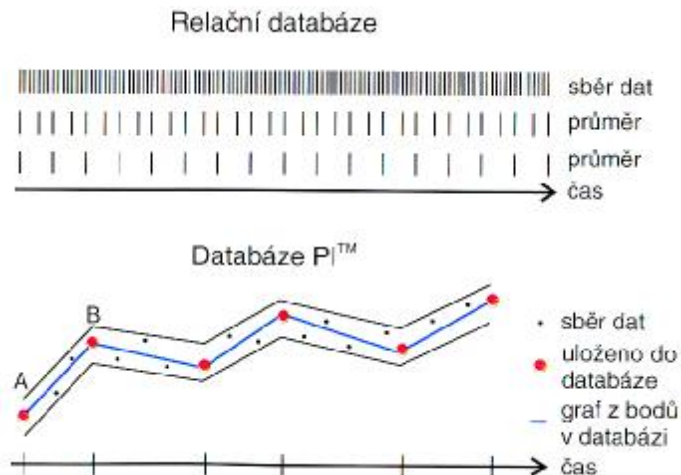
Obrázek 1.6 Základní koncepce PI Systemu - OSI Software²⁹

Skládá se z mnoha modulů pro různé operace s daty. Centrálním prvkem je zde **Universal Data Server** spolu s moduly **PI Database** a **PI DataStorage**, které slouží k real-time ukládání i k zpřístupnění technologických dat.



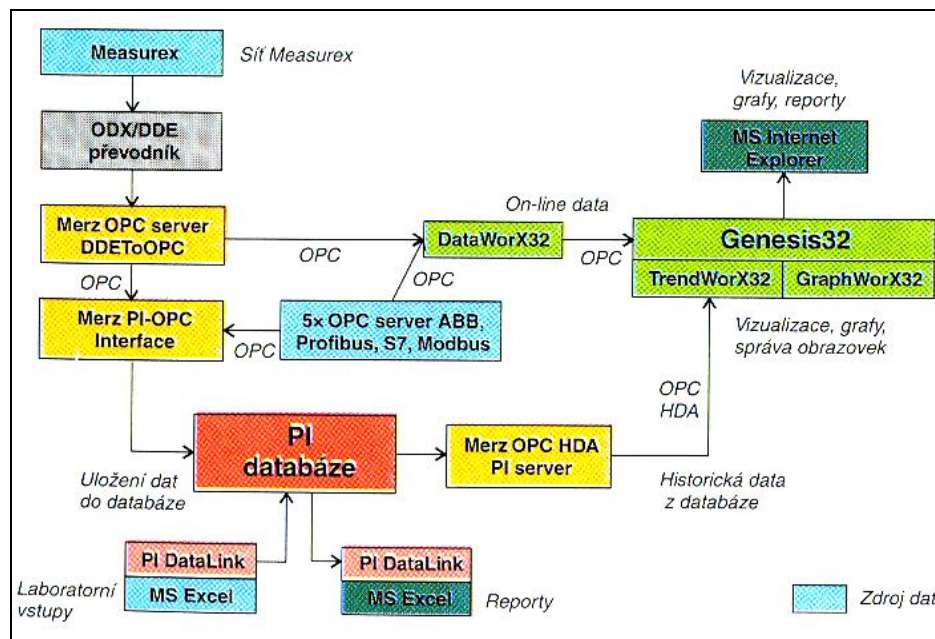
Obrázek 1.7 Architektura systému PI - OSI Software²⁹

Do PITM databáze však nejsou ukládány všechna naměřená data¹⁶, ale jen vybrané body – hodnoty v čase, při kterých došlo k významné změně veličiny. Tato metoda je známá pod názvem „metoda změnového ukládání“. Na základě poslední uložené hodnoty v databázi (A) a poslední získané hodnoty se kolem hodnot v paměti počítače průběžně tvoří pás hystereze (A-B). Teprve při vybočení některé z hodnot z tohoto pásu se do databáze запиše poslední hodnota, jež se nachází v pásmu hystereze (B). Pás hystereze vůbec nemusí být vodorovný, ale s určitou přesností kopíruje chování měřené veličiny. Obdobně se situace opakuje dál v čase. Graf sestavený z uložených hodnot, přibližující reálně naměřený průběh sledované veličiny, je tvořen spojnicí uložených bodů, přičemž odchylka každého naměřeného bodu od takto vzniklé křivky – lomené čáry – je vždy menší než nadefinovaná přesnost.



Obrázek 1.8 Rozdíl mezi ukládáním do relační a real-time databáze PI™

Tato real-time databáze byla například implementována v technologicko-informačním systému (TIS) pro papírny ve Štětí¹⁶. Byly zde tři různé zdroje informací – ze sítě řídicích systémů firmy Measurex, softwarové výstupy technologických serverů OPC a laboratorní vstupy (Excel). S ohledem na typovou i formátovou různorodost těchto vstupů byly pro zajištění jejich kompatibility vybrány datové formáty standardu **OPC**.

Obrázek 1.9 Schéma přenosu v TIS s databází PI™ pro Štětí¹⁶

Informace získávané z měřicích systémů Measurex byly transformovány prostřednictvím softwarových převodníků ODX/DDE a DDEToOPC. Takto sjednocená data byla pak prostřednictvím převodního programu *Merz PI-OPC Interface* ukládána z OPC serverů do databáze PI a zároveň poskytována vizualizačnímu (SCADA/HMI) systému *Genesis32*. Pro on-line přenos dat byla použita specifikace *OPC Data Access 2.0* a pro přenos historických dat z databáze PI *OPC Historical Data Access 1.0*.

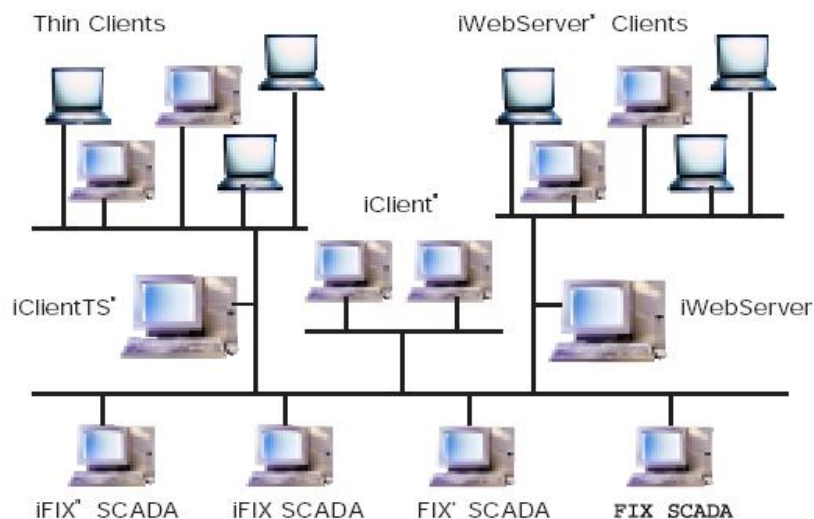
Modul *Merz PI-OPC Interface* má za úkol průběžně monitorovat “pohyb“ měřených veličin a získané hodnoty předávat na vstup databáze. Před možnou ztrátou dat při výpadku spojení mezi OPC serverem a databází PI chrání TIS tzv. **buffering** - měřená data jsou při přerušení komunikace s databází ukládána na lokální pevný disk počítače s OPC serverem a po obnovení spojení jsou do ní zpětně zanesena.

Modul *PI DataLink* zajišťuje propojení mezi PI databází a programem MS Excel. Tento doplněk pro program MS Excel umožňuje jednoduchou vizualizaci uložených údajů v rámci tzv. Uživatelských reportů informujících např. o aktuálním stavu výroby, spotřebě, bilanci, minimech a maximech, přehledu význačných bodů v databázi PI apod. Kromě těchto reportů má obsluha k dispozici široké spektrum dotazů přímo spojených s databází. Tento modul lze ale využít i ke komunikaci opačným směrem – pro zadávání laboratorních dat (testovacích vzorků) z různých částí technologie do centrální databáze. Díky tomuto uložení jsou tyto informace k dispozici na všech úrovních řízení technologie a výroby.

Vizualizační program *Genesis32* slouží ke sledování technologických dat, prohlížení protokolů a tvorbě grafů ze všech úrovní (dispečeri, technologie, topmanagement). Zajímavostí je plug-in *WebHMI* pro MS Internet Explorer (objekt ActiveX).

1.1.3.3 Řešení Intellution – *iFIX SCADA a iHistorian*

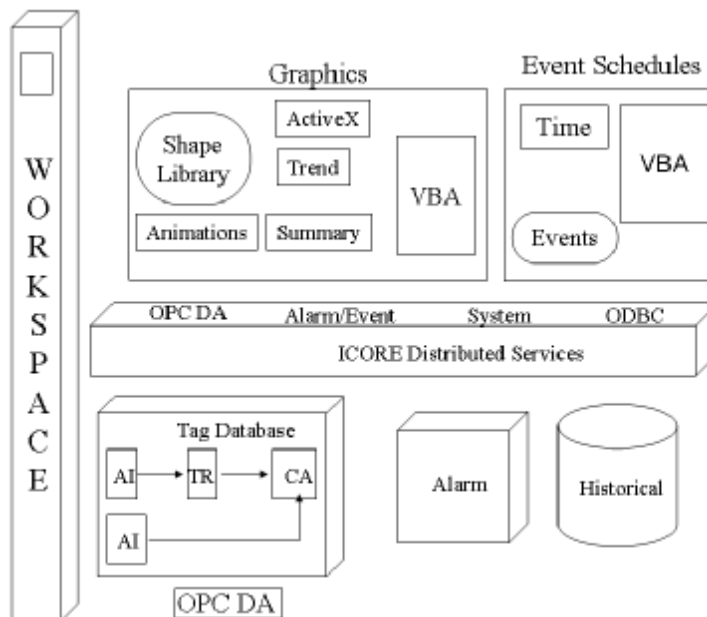
iFIX je SCADA/HMI komponent rodiny Intellution Dynamic od firmy Intellution^{24, 42, 47}. Realizuje vizualizaci, sběr údajů a supervizorské řízení výrobních procesů.



Obrázek 1.10 Architektura klient/server systému iFIX Dynamic – Intellution²⁴

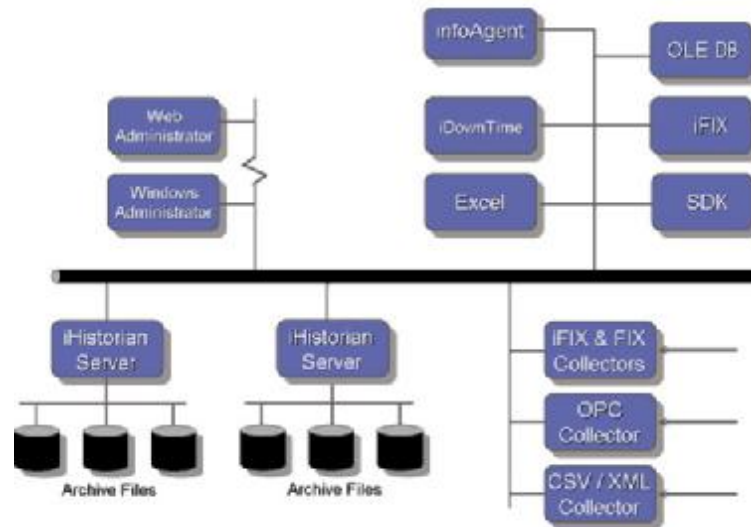
iFIX SCADA Server je připojen k fyzickým datovým I/O a udržuje **real-time procesní databázi**. K dispozici je rozsáhlá škála typů databázových bloků včetně analogových a digitálních vstupů a výstupů, výpočtových, alarmových, souhrnných a statistických funkcí k řízení. Na základě požadavků klientů poskytuje svoje údaje v reálném čase. Plně podporuje objekty ActiveX pro vytváření interaktivních aplikací a OPC klient/server architekturu současně na straně serveru i klientu. **SQL/ODBC API** umožňuje jednoduchou integraci s **relačními databázemi (zejména MS SQL Server)**, dále zahrnuje VBA jako výkonný nástroj na vytváření skriptů.

iClient™ je standardním klientem umožňujícím plný přístup k údajům ze všech síťových SCADA Serverů. *iClientTS™* je tenký klient a web řešení využívající Microsoft Terminal Services. *iWeb Client* je trochu jiný typ klienta spolupracující s *iWeb Server*, který publikuje HTML stránky (zkonvertované iFIX grafické objekty).



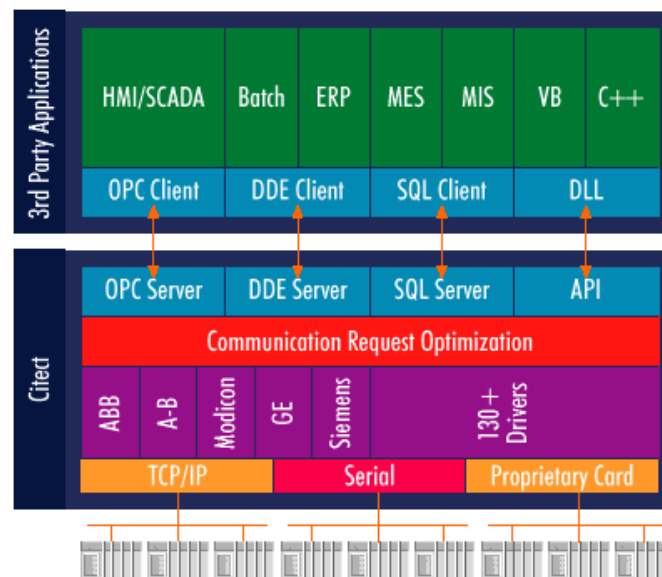
Obrázek 1.11 Vnitřní architektura programu iFIX - Intellution

iHistorian je novinka firmy Intellution. Je to distribuovaný systém skládající se ze tří logických částí ²⁷: *Nástroje pro sběr a archivaci dat* jsou distribuovaně rozmístěny v systému. Lokální sběrače dat tzv. **data collectors** mají za úkol připravovat data (komprimace, časové značky, apod.) pro centrální server iHistorian. Pokud dojde k přerušení komunikace, lokální sběrače pracují dál, aby po opakovaném navázání komunikace s centrálním serverem mohla být data zpětně do databáze zanesena. Servery systému umožňují archivovat 100-100 000 údajů z přípojných bodů. *Nástroje pro administraci* pracují jak na úrovni aplikace, tak i pomocí vzdáleného přístupu prostřednictvím sítě. Dalšími aplikacemi jsou *nástroje pro analýzu dat* prostřednictvím sítě Internet (NorthSmart), využitelnosti zařízení (iDowntime) a dále marka pro MS Excel.

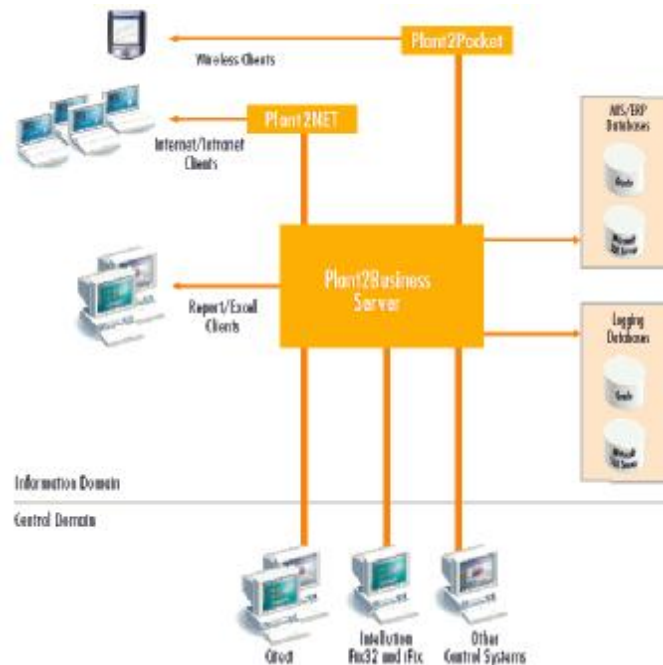
Obrázek 1.12 Architektura systému iHistorian – Intellution ²⁴

1.1.3.4 Řešení CI Technologies – systém Citect a Plant2Business

Firma **CI Technologies** ³¹ se zviditelnila svým universálním datovým serverem **Citect** spolupracujícím s klientskými aplikacemi s využitím standardů OPC, DDE, ODDC/SQL i na nejnižší úrovni API/DLL.

Obrázek 1.13 Komunikační architektura datového serveru Citect – CI Technologies ³¹

V roce 2000 tato firma uveřejnila softwarový produkt z kategorie SCADA/HMI pod názvem **Plant2Business** ³² skládající se ze tří modulů: **P2B Server** s integrovaným **Plant2SQL**, **Plant2Net** a **Plant2Pocket**.




Obrázek 1.14 Architektura systému Plant2Business – CI Technologies ³¹

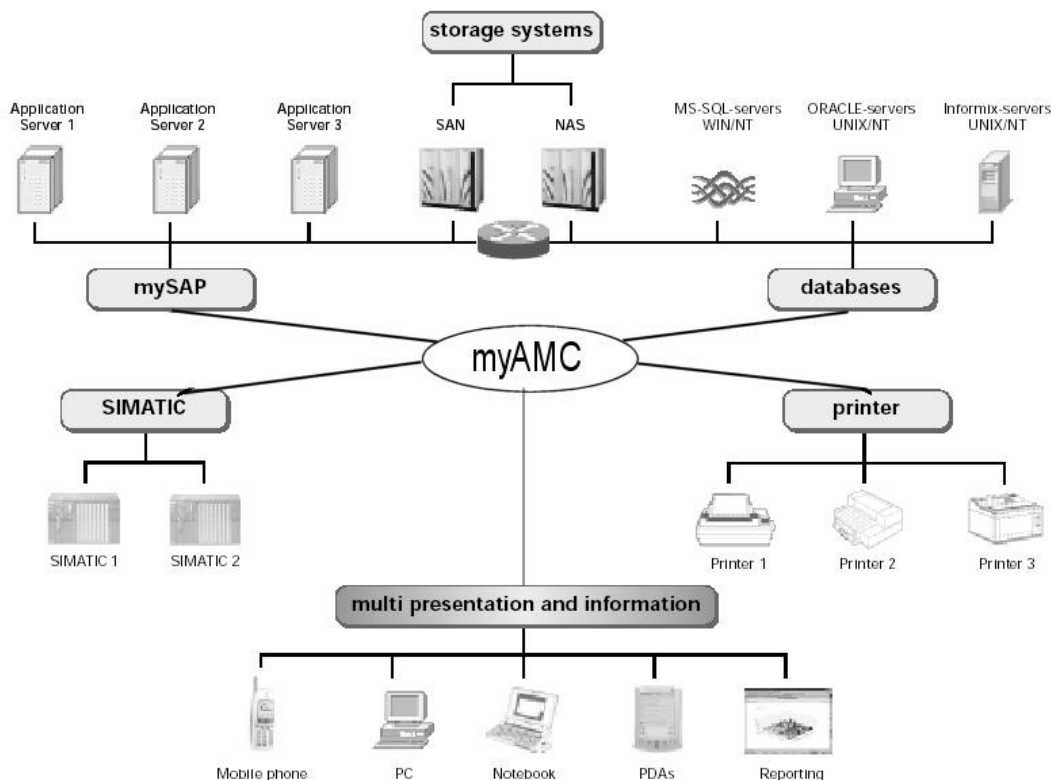
Modul **Plant2SQL** zpřístupňuje data z řízeného procesu dalším uživatelům z prostředí podnikové infrastruktury a tím odstraňuje pomyslnou bariéru mezi SCADA a ERP produkty. Přesto však Plant2SQL umožňuje fyzicky oddělit technologickou a informační síť, aby se zatížení informační sítě podniku nepřeneslo do technologie. Plant2SQL využívá k ukládání procesních dat databázi **MS SQL Server 7** nebo **MSDE** (Microsoft Data Engine), což je databázový stroj založený na architektuře MS SQL Serveru (omezená verze bez GUI rozhraní). Přes jednotné rozhraní **OLE DB** dokáže zpřístupňovat veškeré informace z programu Citect klientským aplikacím. Pro usnadnění tvorby klientských aplikací je možné využívat objektové knihovny Plant2SQL, jako např. **ADO**.

1.1.3.5 Řešení National Instruments – systém Lookout™ a databáze Citadel

Lookout™ je objektově orientovaný software typu SCADA/HMI ^{33, 46}. Ukládání událostí a historických trendů je vyřešeno buď do **lokálních ASCII** souborů (tabulek *.csv) pro rychlý přístup, nebo do **Lookout Citadel databáze** pomocí ODBC/SQL. Umožňuje také filtrování a kompresi ukládaných dat pro úsporu místa na paměťových médiích.

1.1.3.6 Řešení Siemens – systém myAMC®

 myAMC® (Application Managment Center) monitoruje, prezentuje, ale hlavně spravuje veškeré podnikové i technologické informace ²⁶. Pro ukládání dat jsou vytvořena úložiště tzv. „**SAN/NAS repository**“, ze kterých je možné data dále archivovat do relačních databázových serverů **MS SQL, Oracle nebo Informix**.

Obrázek 1.15 Architektura systému myAMC – Siemens ²⁶

1.1.3.7 Řešení Coral s.r.o. – systém TIRS[®]

TIRS[®] je technologicko-informační systém pro průmyslovou automatizaci kategorie MMI&SCADA pro tvorbu i provozování dispečerských pracovišť, informačních a řídicích systémů ²⁰. Je určen pro hierarchicky členěná dispečerská pracoviště, řídicí a monitorující nejen lokální systémy různých technologií, ale i distribuované systémy a technologické sítě propojené sítí LAN, WAN nebo Internet. Jeho velkou předností je modulárnost, otevřenost a snadná možnost integrace do nadřazených informačních systémů podniku, které mohou pracovat pod OS Windows, DOS i UNIX. Stavebnicovost a důsledná otevřenost systému umožňuje připojovat vlastní moduly pro sběr i zpracování dat. Rekonfiguraci systému lze provádět i za chodu aplikace. Data z různých typů PLC automatů lze zobrazovat společně ve vizualizačních panelech i zapisovat do databází historických dat. Typ databází je dán použitými ODBC ovladači **od lokální dBase** (standardně pro poruchy a události), **MS Access** nebo **FoxPro až po datové servery s SQL databázemi**. Mezi nejpoužívanější patří relační databáze **MS SQL Server** a **Oracle**. U každé proměnné je možné kromě aktuální hodnoty pracovat s limitami, hysterezí, časem přenosu, minulou hodnotou a hlavně s příznakem platnosti dat. Zajímavostí je integrace prvků GIS obsahující databázové informace o objektech a aktuální hodnoty objektů. Grafická data mohou být ve vektorovém tvaru i rastrovém tvaru.

Jádro systému **TIRS[®] Server** je koncipováno jako server celého systému a ostatní úlohy (moduly) systému **TIRS[®] 32** se k jádru připojují jako klienti. Jádro obsahuje **registrační databázi** všech přihlášených signálů, tzv. datových bodů, a zajišťuje vazby mezi všemi prvky systému **TIRS[®] 32**. Jádro dále zabezpečuje tvorbu, ukládání, tisk, zobrazení a

zvukový doprovod poruchových hlášení. **TIRSWeb** je obdobou systému TIRS[®]32 a je určen pro rozsáhlejší (distribuované) aplikace.

Tabulka 1.6 Přehled používaných SQL databází s vazbou na systém TIRSWeb – Coral s.r.o.

Klient – zákazník	Použitá SQL databáze
Elektrárna Opatovice a.s.	MS SQL Server
Tepelné hospodářství Pardubice	MS SQL Server
Informační systém povodí Labe	Oracle na Linuxu s replikací
BVV – Vizualizace a řízení tepla	MS SQL Server
Plzeňská distribuce tepla	MS SQL Server

1.1.3.8 Řešení Amit s.r.o. - DB-Net

DB-Net²¹ je průmyslový informační systém, který je typickým představitelem decentralizovaného řízení. Je to soubor programových, komunikačních a technických prostředků umožňující snadno a komplexně provázat veškeré měřené a regulované údaje a předávat je k dalšímu zpracování a archivování. Jsou zde blíže nepopsané lokální archivy, jež umožňují archivovat měřená a vypočtená data i v okamžiku, kdy není v činnosti dispečerské pracoviště anebo došlo k poruše komunikace. Automatický zpětný přenos dat zabezpečí, že ani v těchto případech uživatel nepřijde o důležité údaje.

Komunikační ovladače – knihovny **Atouch** - představují nejnižší programovou úroveň zajišťující vazbu (obousměrný přenos dat) mezi informačním systémem DB-Net a aplikacemi na PC prostřednictvím protokolu TCP/IP. Ovladače tedy mohou využívat některé **aplikace třetích stran** (vizualizační systémy, speciální archivační a programové aplikace, **SQL servery** nadřazených ERP implementací apod.).

1.1.3.9 Řešení GeoVap s.r.o. – systém Reliance

Reliance 3 patří do kategorie průmyslových vizualizačních SCADA/HMI systémů. Systém je tvořen programovými moduly *Reliance design/runtime*, *Reliance server/runtime server*, *Reliance J* a komunikačními ovladači. **Reliance design** je vývojové prostředí, kde se vytváří vizualizační projekt spustitelný v modulu runtime, zajišťující běh vizualizačního programu u cílového uživatele.

Reliance server²¹ je komunikační koncentrátor, který v reálném čase prostřednictvím komunikačních ovladačů zajišťuje sběr dat z technologických stanic a jiných runtime modulů. Plní funkci centrálního datového skladu a komunikačního centra rozsáhlejších vizualizačních projektů. Získaná data archivuje a poskytuje je jiným runtime modulům či serverům po síti protokolem TCP/IP. Zároveň procesní data (aktuální i historické hodnoty) poskytuje webovým klientům Reliance J. Zajišťuje také předávání a vykonávání povelů od autorizovaných klientů. Reliance server nemá možnost grafického zobrazení technologických dat. Zajišťuje tedy sběr, distribuci a archivaci dat i poruchových hlášení a příjem a vykonávání povelů od klientů. Archivace dat je buď realizována pomocí **BDE** (dotazy) do lokálních **Paradoxových** či **DBase tabulek**, nebo přímo **SQL příkazy** klasických relačních serverů. Pro náročnější průmyslové použití byl systém testován s **InSQL Serverem** firmy Wonderware.

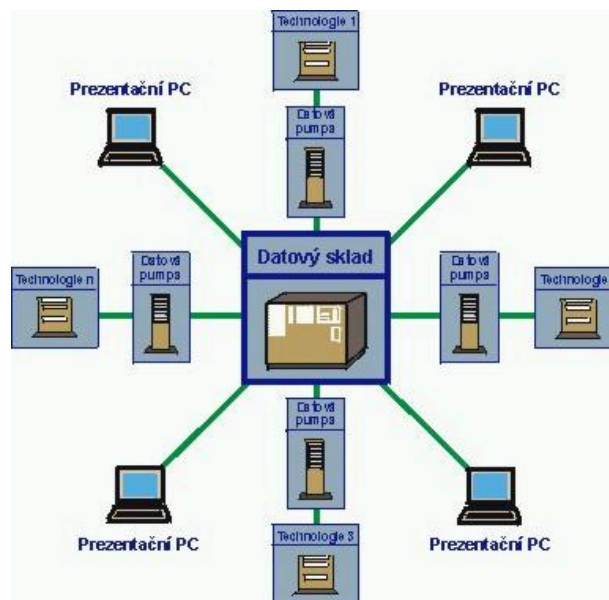
Reliance J je Java applet určený pro zobrazení vizualizačních schémat s aktuálními hodnotami klientům sítě Internet/intranet. Umožňuje sledování technologie pomocí internetového prohlížeče podporujícího jazyk Java (např. Microsoft Internet Explorer 4+, Netscape Communicator 4+). Dokáže zobrazit vizualizaci vytvořenou ve vývojovém prostředí

a není tedy třeba tvořit zvláštní webovou verzi. Jako datový zdroj využívá Reliance server nebo Reliance runtime server, od kterého získává aktuální hodnoty i archivní data včetně poruchových hlášení protokolem TCP/IP.

1.1.3.10 Řešení Medium Soft a.s. - Vidium

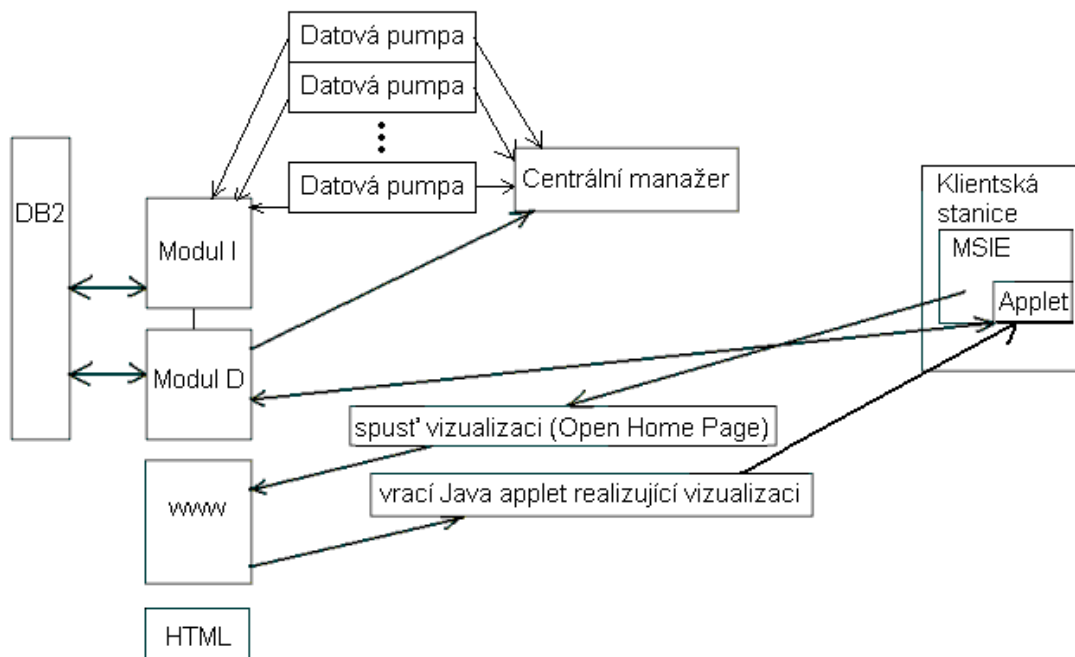
Vidium je informační systém pro sběr, uchování a zpracování dat z více heterogenních zdrojů na základě datového skladu a internetových technologií²³. Základní myšlenkou je centralizace dat z různých zdrojů, jejich zpracování na jednom místě a distribuování v požadované formě k uživateli. Zdroje dat mohou být historicky i technicky různého původu a uloženy na různých datových serverech, které archivují data příslušné technologie nebo provozu. Nutné ale je, aby byly přístupné počítačovou sítí.

Pomocí specializovaného počítače zvaného "**datová pumpa**" je zajištěno spojení s provozními počítači (zdroji heterogenních dat). Data jsou pumpou cyklicky načítána, kontrolována (pokud je to možné) a v jednotném formátu dočasně archivována v datovém skladu pro případ nečekané poruchy nebo výpadku. Důležité je, že do stávajících výpočetních systémů v technologiích se nezasahuje, resp. úpravy jsou minimální a pouze na straně výstupu dat.



Obrázek 1.16 Ideové schéma systému Vidium - MediumSoft

Na klientské stanici se spustí vlastní vizualizace (java applet) z prostředí internetovského prohlížeče. Tento applet naváže komunikaci s datovým skladem a předává jak statická (agregované údaje, bitmapy apod.), tak i dynamická data. S datovým skladem applet spolupracuje pomocí tzv. **modulu D** (D = dotaz). Tento modul má za úkol zásobovat vizualizační část systému požadovanými informacemi. Druhý, vstupní **modul I** (Input), je na něm nezávislý a nepřetržitě zpracovává údaje z datových pump, ukládá je do databáze a vypočítává zadané agregace. Nezávisle na propojení datová pumpa – datový sklad – klientská stanice běží spuštěný program centrálního manažera, který umožňuje řízení a kontrolu. Protože všechny části systému jsou co nejvíce autonomní, není jeho činnost k chodu systému nutná, v případě instalace jednotlivých fyzických komponent systému na odlišných stanovištích je však výhodou. Datový sklad je tvořen relační databází **IBM DB2**.

Obrázek 1.17 Popis funkce systému Vidium – MediumSoft ²³

1.1.3.11 Řešení Microsys – Promotic 2000

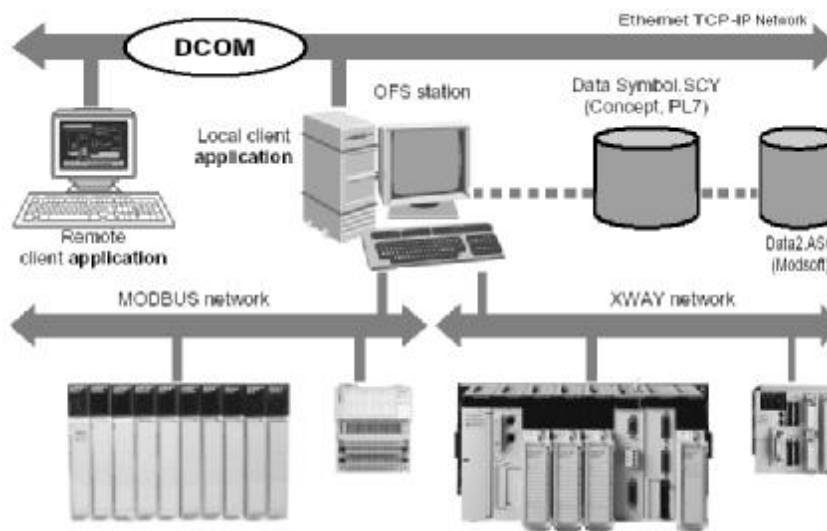
Promotic 2000 pro Windows 9x/ME/NT/2000 je komplexní softwarový nástroj pro tvorbu vizualizačních a řídicích systémů ^{25, 43}. Umožňuje efektivně vytvářet distribuované, flexibilní a otevřené aplikace v nejrůznějších odvětvích průmyslu. Díky zabudované podpoře standardních rozhraní **ODBC**, **DAO**, **SQL** v modulu **PmDatabase** je umožněn přístup k externím podnikovým databázím (např.: dBase, Access, FoxPro, Paradox, Informix, Oracle).

1.1.3.12 Řešení IpeSoft – D2000 Enterprise Production Systems

Balíček **D2000 Enterprise Production Systems** firmy IpeSoft obsahuje výkonný SCADA/HMI systém (architektury klient/server) **D2000 Actis** a systém komplexních výrobních informací **D2000 Entis** ³⁵. K zpracování velkého množství informací v reálném čase D2000 Actis využívá vlastní technologii, tzv. dynamický objektový datový model (DODM), a translačně pracující databázové systémy na bázi SQL. Výkonný systém archivaci **D2000 Industrial SQL Archiv** zabezpečuje tři úrovně archivace (primární, statistický a dlouhodobý archiv), dodatečný zápis a zpracování zpožděných údajů.

1.1.3.13 Řešení Schneider-electric – OFC (OPC Factory Server)

OFC (OPC Factory Server) je software pro průmyslové použití firmy Schneider-electric^{28, 44}. **OPC Server** je modul pro čtení dat z OPC severů, jejich archivaci a distribuci. Pro ukládání dat jsou v systému začleněny databáze **MS Access (mdb)** a soubory formátu **ASCII (asc)**.



Obrázek 1.18 Architektura OFC (OPC Factory Server) – Schneider-electric²⁸

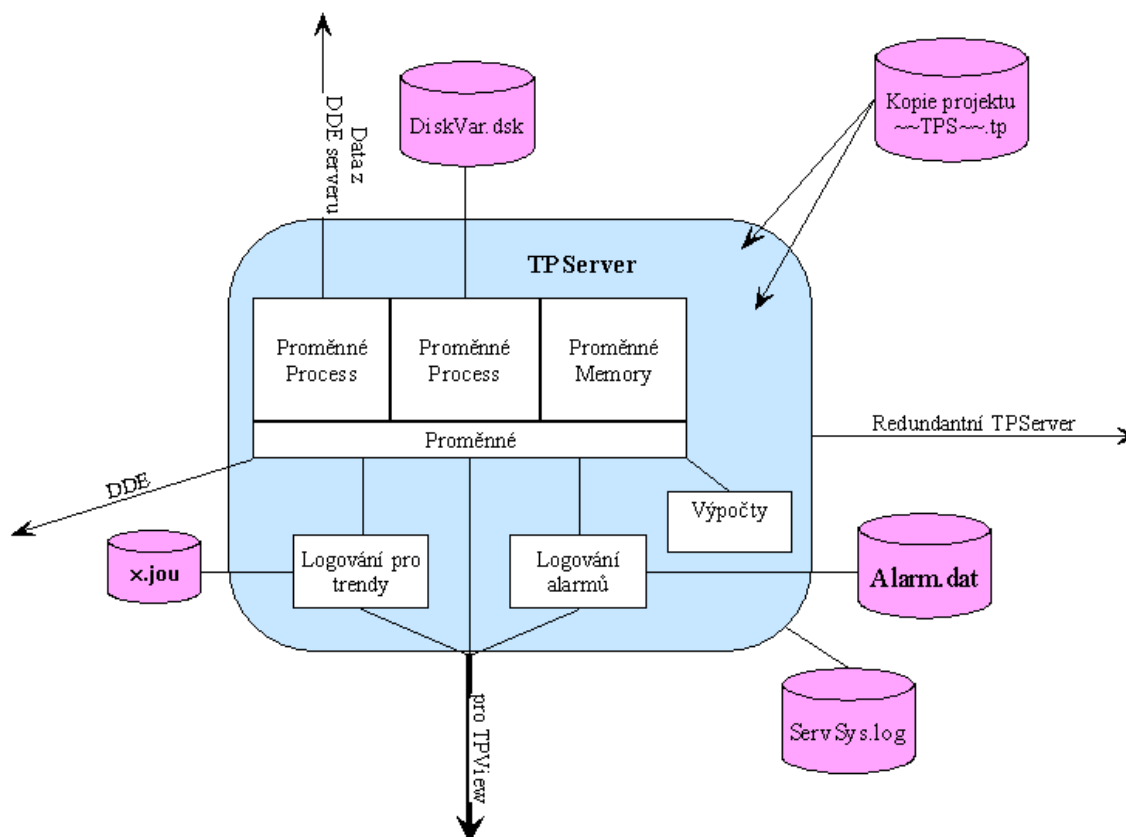
1.1.3.14 Závěr - porovnání

Na závěr jsem výše popsaná řešení z průmyslu přehledně porovnal v tabulce s programem TomPack (viz. Příloha 5), podrobněji popsaným v následující kapitole. Veškeré použité zdroje v elektronické podobě naleznete na CD-ROM (viz. Příloha 1).

2 ProjectSoft – TomPack

Softwarový produkt **TomPack** firmy *ProjectSoft*^{3, 14} (Hradec Králové, CZ) slouží pro vizualizaci a řízení technologických procesů typu SCADA/HMI na podporované platformě PC s operačními systémy Windows NT/2000. Je naprogramován v jazyce C a je chráněn HW klíčem pro LPT port. Má filosofii podobnou jako program InTouch z balíku FactorySuite (kapitola 1.1.3.1). Jeho architektura typu klient/server umožňuje redundanci jednotlivých částí systému, tj. při výpadku jednoho z operátorských PC lze technologii řídit pomocí záložního počítače. Je složen ze čtyř programů - **TPServer**, **TPView**, **TPConfig** a **UniServer**.

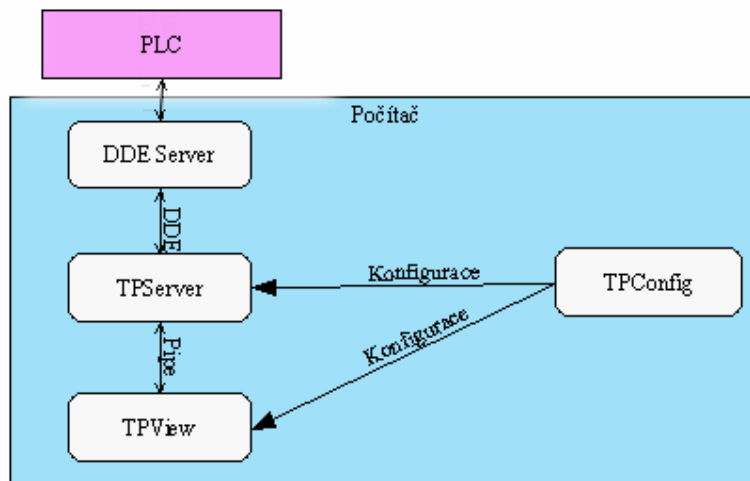
TPServer je ústředním členem pro komunikaci uvnitř PC i po síti mezi DDE serverem a klientem TPViewer. Dle jejich požadavků zasílá či modifikuje data uložená v DDE. Dále spravuje vnitřní proměnné, zaznamenává trendy a alarmy do souborů (**.jou* a *alarm.dat*).



Obrázek 2.1 Architektura systému TomPack - TP Server³

TPView je vlastní vizualizační program, který pomocí displejů zobrazuje data přijatá od TPServeru a jejich změny mu zpět posílá.

TPConfig slouží pro tvorbu nového projektu a jeho konfiguraci (displeje propojené s procesem). Vytvořený projekt ukládá do jediného souboru a posílá ho programům TPServer a TPView.



Obrázek 2.2 Nejjednodušší konfigurace systému TomPack³ (Odparka)

Pro rozšíření možností programátora, TomPack obsahuje skriptovací jazyk (podmnožina jazyka C), který umožňuje definovat výpočty pomocí základních matematických funkcí, práci s okny a alarmy pomocí systémových funkcí, práci se znakovými soubory a další speciální funkce např. časovače, regulátory a ODBC funkce pro komunikaci s databázemi.

UniServer je vlastní DDEServer, který realizuje DDE komunikaci pomocí ovladače (např. SattBus) řídicího automatu PLC s TPServerem. Je možné využívat i DDEServery třetích stran (např. od firmy Wonderware). V poslední verzi (TomPack v1.10) jsou podporovány tyto komunikační protokoly:

Tabulka 2.1 Podporované komunikační protokoly pro UniServer pro TomPack (v1.10)³

Firma	Komunikační protokol	HW pro PC
ABB(SattControl, Alfa-Laval)	SattBus	SattBus ISA karta
ABB(SattControl, Alfa-Laval)	COMLI	sériový port
ABB	PROCONTIC	sériový port
SIEMENS	AS511	sériový port, převodník na proudovou smyčku
SIEMENS	PPI	sériový port, převodník na RS485
SIEMENS	MPI	sériový port, převodník MPI/DP
SIEMENS	S7	Profibus nebo Ethernet karta
SAIA	S-BUS	sériový port
Rockwell	DF-1	sériový port
ostatní...	universální	sériový port

Více informací o programu TomPack se dozvíte z literatury³.

3 Oracle®

Oracle® je celosvětově druhá největší softwarová firma a dodavatel nejmodernějších globálních e-business řešení pro komerční internet a webové aplikace. Nabízí svou internetovou platformu, nástroje a internetové aplikace, k těm pak náležitě konzultace, výuku a podporu ve více než 145 zemích světa¹⁰. Do následující tabulky uvádím hlavní produkty z jejich širokého sortimentu a jejich využití.

Tabulka 3.1 Přehled produktů firmy Oracle a jejich využití

Produkt	Využití
Oracle 9i Database (Enterprise manager, IFS, ...)	vlastní databázový systém (RDBMS)
Oracle 9i Application Server (Portal, Web services, ...)	internetová platforma pro vývoj presentací a aplikací (podnikové portály, portlety, podpora XML)
Internet Developer Suite <ul style="list-style-type: none"> • JDeveloper • Discoverer • Forms & Reports Developer 	<ul style="list-style-type: none"> • produktivní rozhraní pro vývoj JAVA aplikací (servlety) • tvorba sestav a analýz • design a generování formulářů a reportů (sestav)
Nástroje typu CASE <ul style="list-style-type: none"> • Designer 2000/v. 2.1 	<ul style="list-style-type: none"> • návrh a modelování struktury systémů (z ER diagramů) návrh a generování databázových objektů
Data Warehousing and Business Intelligence <ul style="list-style-type: none"> • Data Mining Suite • Warehouse Builder 	<ul style="list-style-type: none"> • prostředek pro dolování dat (datové sklady) • prostředek pro vytváření virtuálních obchodů
Produkty ERP <ul style="list-style-type: none"> • Application R11/R11i 	<ul style="list-style-type: none"> • ERP aplikace (kniha závazků a pohledávek, majetek, zásoby, řetězce)

3.1 Databázový systém Oracle9i

Oracle RDBMS (od verze 8 a výše) je víceuživatelský hybridní neboli objektově-relační databázový systém, který využívá třívrstvou architekturu typu klient-server (přesněji tenký klient – databázový server – aplikační server). Pracuje pod různorodými platformami, z čehož vyplývá i rozmanitost podporovaných operačních systémů. Mezi podporované sítě patří např. Microsoft LAN Manager, NetWare, VINES, DECnet a další sítě komunikující pomocí protokolu TCP/IP.

Tabulka 3.2 Svoboda a flexibilita při výběru provozní platformy a architektury pro systém Oracle¹⁰

Firma	Typ operačního systému
Compaq	Windows NT, Windows 2000, Tru64 UNIX, TruCluster Server, OpenVMS
Sun	Solaris (32/64 bit), Sun clustery
HP	Windows NT, Windows 2000, HP-UX (32/64 bit), HP ServiceGuard clustery
IBM	Windows NT, Windows 2000, OS-390, AIX (32/64-bit) Linux servery/clustery
Intel	Windows NT/2000, Linux

Od verze 8 Oracle podporuje velmi rozsáhlé databáze až do velikosti desítek terabajtů (1 TB = 10^{12} B) a umožňuje až několik desítek tisíc paralelně připojených uživatelů. Rozšířila se také podpora ukládání rozsáhlých **objektů** (např. pro obrazová a multimediální data) vytvořením nových datových typů. Více informací o datových typech v literatuře^{10, 11, 39}.

Tabulka 3.3 Základní datové typy systému Oracle

Datový typ	Anglický ekvivalent	Význam - použití
(N)VARCHAR2	VARiable CHARacter	řetězec znaků proměnné délky
(N)CHAR	CHARacter	řetězec znaků definované délky
DATE	-	datum a čas
INTEGER	-	celé číslo, maximální velikost je dána platformou, ale nejčastěji bývá omezena na 32 bitů
NUMBER	-	číslo s definovanou přesností
FLOAT	-	číslo s desetinnou čárkou
BLOB	Binary Large Objects	rozsáhlé binární objekty
(N)CLOB	Character Large Object	rozsáhlé znakové objekty
BFILE	Binary FILE	binární soubory
LONG	-	celé číslo o „neomezené“ velikosti
RAW	-	binární řetězce
Unicode	-	universální znakový formát

Existují též verze databáze typu Oracle nejen pro standardní PC, ale i pro zařízení typu PDA (tzv. OracleLite). Rozšířila se také sada nástrojů pro správu databázového systému Oracle zvaná **Enterprise manager** o následující komponenty:

Tabulka 3.4 Komponenty Enterprise manageru

Komponenta	Použití
Storage manager	sleduje využití disku a velikost datových souborů
Schema manager	spravuje všechny databázové objekty (tabulky, procedury, sekvence,...)
Security manager	vytváří uživatele, role, práva a uživatelské profily
SQL Worksheet	dotazovací prostředí (jazyk SQL)
Data manager	provádí import/export dat
Backup manager	provádí zálohování/zotavování databáze
Resource manager	řídí přidělování a využití systémových zdrojů (čas CPU, paralelní procesy, ..) pro dané uživatele/skupiny
Instance manager	ovládání databáze, konfigurace

Novinkou verze Oracle9i je tzv. **Real Application Cluster**, který umožňuje správu více serverů v clusteru (spojení serverů přes rychlé Ethernet sítě >100Mbit) z hlediska systému i koncového uživatele či aplikace jako jediného systému. To prakticky odstraňuje bariéry růstu systémových nároků zároveň s maximálním omezením možných výpadků. Dále eliminuje nutnost změn v aplikaci nebo datech při přidávání dalších serverů do clusteru. Zajišťuje vysokou dostupnost aplikace i v případě výpadku jednoho či více jednotek clusteru.

4 Úvod do jazyka SQL

Jazyk SQL patří mezi standardy skupiny databázových dotazovacích jazyků. V roce 1974-75 firma IBM zahájila výzkum možností relačních databází (prototyp relačního databázového systému **SEQUEL-XRM**) a byly vytvořeny základní příkazy pro jejich ovládání a dotazování – jazyk **SEQUEL** ³⁶. Následně vznikla řada relačních databázových systémů od IBM - System R, Oracle Corp. – systém Oracle a další (SQL/DS, DB2). Od roku 1984 se jazyk **SQL** používá v praxi jako standard, r. 1986 vyšla ještě verze SQL86 bez prvků integrity dat, ale v další verzi SQL92 – SQL2 (1992) byly tyto prvky doplněny (primární klíče a definice integritního omezení dat). Nový SQL3 zahrnuje navíc i objektový přístup. Příkazy jazyka SQL jsou členěny do několika skupin.

Tabulka 4.1 Typy, využití a základní příkazy jazyka SQL

Typ	Využití	Příkazy a jejich funkce
DML	manipulace s daty	SELECT (získání dat) INSERT (vkládání dat) UPDATE (modifikace dat) DELETE (mazání dat)
DDL	definice dat (datových struktur)	CREATE (tvorba) ALTER (změna) DROP (odstranění)
DCL	řízení a správa databází	GRANT (přidělení práv) REVOKE (odnětí práv) COMMIT (potvrzení transakcí) ROLLBACK (stornování transakcí)

Původně byl jazyk SQL navržen jako neprocedurální jazyk (příkazy popisují výsledek, nikoliv postup získání výsledku), jeho modifikací však vznikl i jazyk procedurální **PL/SQL** (popisující přesný postup vedoucí k výsledku) Jazyk **PL/SQL** je velmi podobný jazyku Pascal a využívá prvky SQL. Piší se v něm programové bloky tzv. **uložené procedury** a **spouště (triggers)**. *Sekvence (sequence)* je generátor unikátních čísel realizovaný ve sdílené paměti na serveru. Poskytuje řadu unikátních čísel pro PL/SQL programy, používaných jako primární klíče do tabulek. *Spouště (triggers)* jsou uložené procedury, které se aktivují při definované akci nad databázovou tabulkou.

O jazyku SQL i PL/SQL bylo napsáno již mnoho literatury a proto Vás v následující tabulce odkáží na některé vybrané tituly.

Tabulka 4.2 Doporučená literatury ke studiu jazyka SQL

ŠIMŮNEK M.: SQL Kompletní kapesní průvodce. GRADA Publishing, 1999.
HALAŠKA I.: Databázové systémy – Jazyk SQL a systém ORACLE. ČVUT Skriptum, 1995.
Internet: http://www.volny.cz/dbases/sql_dev.html , 2002.
Internet: http://odkazy.internetshopping.cz/internet/sql/ , 2002.
Internet: http://skrivan.hyperlink.cz/sql.html , 2002.
JANDA J.: Jemný úvod do SQL. Elektronický dokument (Sql.pdf), 1997. zdroj : Internet: http://home.zcu.cz/~janda50/SQL/sql.html (viz. Příloha 1)

5 Dolování dat (Data-mining)

Dolování dat je obecně definováno jako netriviální proces výběru, testování a modelování ve velkých objemech dat, jehož účelem je zjistit neznámé a potenciálně užitečné relace neboli vztahy mezi těmito daty^{37, 38}. Metody, se kterými data-mining pracuje, jsou skutečně netriviální a v mnoha případech absorbují poslední poznatky z umělé inteligence a strojového učení. Zároveň však podstatným znakem data-miningu je snaha zjištěné výsledky reprezentovat formou co možná nejpřístupnější uživateli. Hlavní prioritou data-miningu je především praktická použitelnost metod. Data-mining je zároveň schopen generalizace, tj. nalezené závislosti by měly být obecné a měly by platit i pro ostatní data stejného charakteru. Data-mining je iterativním procesem, kdy souvislosti nalezené v jedné fázi pomáhají datům lépe porozumět a obohatit tak vstupy pro další fázi. Data-mining nevznikl jako jednotný vědní obor a metody, které používá, patří do nejrůznějších oblastí.

Tabulka 5.1 Rozčlenění metod analýzy databází prostřednictvím nástrojů pro dolování dat

Kategorie metody	Název metody	Typ metody - stručný popis
prediktivní	<i>Regrese</i>	standardní statistická metoda schopná určit stupeň důležitosti vstupních proměnných
	<i>Neuronové sítě (NS)</i>	metoda pro hledání parametrů modelu založená na flexibilním systému vnořených funkcí
	<i>Rozhodovací stromy</i>	metoda popisující model použitím skupiny jednoduchých pravidel
„neprediktivní“	<i>Klasifikace (shluková analýza)</i>	rozčlenění dat do skupiny podle kritérií (kdy kritéria nejsou předem specifikována)
	<i>Analýza asociací (Shopping Bag Analysis)</i>	četnost kombinací produktů společně se vyskytujícími v „datovém koši“
	<i>Skrytá projekce - projektivní geometrie</i>	separace dat do shluků a generování polyhedrů pro jejich pokrytí

Tabulka 5.2 Přehled nástrojů pro dolování dat

Firma	Produkt	Stručný popis
Oracle www.oracle.com	Data Mining Suite (systém Darwin)	vytváření modelů pro dolování (Java API)
Zaptron www.zaptron.com	DataX TM	metody NS, Fuzzy logiky, analýzy čas. řad)
	MasterMiner TM	dolování v hyperprostoru (skrytá projekce)
SpeedWare www.speedware.com	Media Professional	software pro podporu manažerského rozhodování (technologie OLAP/MOLAP)
SAS Inst. www.sas.com	Enterprise Mining <ul style="list-style-type: none"> • SAS/Insight • SAS/Stat • SAS/ETS • SAS/IML 	dynamický nástroj pro analýzu neparametrické analytické nástroje časové analýzy a predikce numerické řešení
Aspen Technology www.aspen.com	NeuralSim	predikce, modelování a automatická klasifikace v reálném čase
ANGOSS Software www.angoss.com	KnowledgeSTUDIO	jednotné prostředí pro dolování dat a porovnání výsledky jednotlivých algoritmů

Praktická část

6 Představení technologicko-informačního systému pro odparku

Laboratorní odpařovací stanice vyrobená firmou **Armfield (UK)** byla původně vybavena pouze místním měřením a ručním ovládáním, proto byly pro její řízení navrženy nové okruhy měření a regulace a dále realizován⁶ počítačový řídicí systém Kmínek. Jako technologicko-informačního systému (TIS) bylo využito softwaru pro řízení a vizualizaci **TomPack** (kapitola 2). V použitém TIS se z důvodu absence vlastní databáze musela historie procesních dat ukládat v alfanumerické formě do souborů na lokálním disku. Z hlediska diagnostiky, vyššího řízení a optimalizace procesů to bylo nevhodné, a proto se začalo přemýšlet o připojení na on-line databázi pracující v reálném čase. S využitím intranetové sítě VŠCHT byl sběrný počítač propojen⁶ s databázovým serverem **Oracle** a tím se rozšířily možnosti pro řízení. Dalšími úkoly, které jsou součástí mé diplomové práce, bylo navržení struktury pro ukládání měřených dat v reálném čase, testování komunikace procesního počítače s databázovým serverem a vytvoření aplikačních programů využitelných k řízení.

6.1 Technický popis vlastní odparky

Technologická stanice se skládá z těla odparky (princip stoupajícího filmu), oddělovače kapalina-pára, zásobníku zahuštěného roztoku, kondenzátoru, dvou zásobníků pro kondenzovanou brýdovou páru, dávkovacího čerpadla pro surovinu a vakuové pumpy. Zdrojem tepelné energie v systému je tlaková pára. Do spodní části těla odparky je čerpadlem dávkován roztok na zahuštění, zde je zahříván na teplotu varu, a z horní části odchází směs zahuštěného roztoku spolu s brýdovou parou. V separátoru je směs rozdělena na kapalinu odváděnou do zásobníku zahuštěného roztoku a páru pokračující do kondenzátoru, kde předá teplo chladicí vodě a tím zkondenzuje. Tento kondenzát se odvádí do jednoho ze dvou zásobníků. Odpařování může být jednorůchodové, anebo je možné roztok po oddělení od brýdových par vést do odparky a znovu zahušťovat. V systému je možné snížit tlak a tím i teplotu varu, což je důležité zejména u roztoků obsahujících termolabilní složky.

Tabulka 6.1 Technické parametry odparky

Parametr	Hodnota
maximální výkon (dle recirkulace)	15 až 60 kg.h ⁻¹
maximální tlak topné páry	200 kPa
maximální spotřeba topné páry	15 kg.h ⁻¹
maximální spotřeba chladicí vody	500 kg.h ⁻¹
absolutní tlak páry v brýdovém prostoru	30 až 100 kPa
délka odpařovací trubky	1,5 m
vnitřní objem	0,003 m ³

6.2 Technický popis měřících a řídicích okruhů

Při modernizaci řídicího systému laboratorní stanice odparku došlo k výměně čidel a akčních členů. Při jeho návrhu bylo z pedagogických důvodů použito průmyslové verze polní instrumentace na základní hierarchické úrovni řízení průmyslového procesního počítače.

Tabulka 6.2 Základní informace o měřící a řídicí soustavě

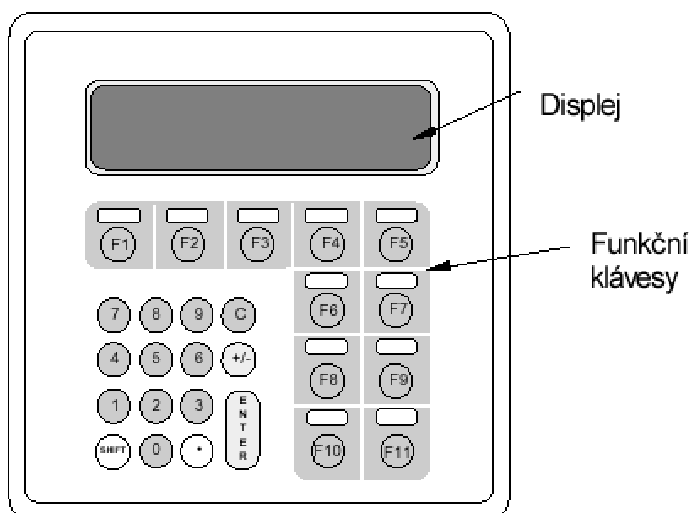
Typ konfigurace, počet úrovní	hierarchická, 2
Výrobce a typ procesního počítače	Alfa Laval, OP-45 SB
Počet vzdálených terminálů (RTUs)	prozatím žádný
Typ sběrnice, max. délka sběrnice	Sattbus, 1000 m

Využilo se programovatelného logického automatu **OP-45 SB** od firmy **Alfa Laval SattControl**, který umožňuje práci s analogovými signály a vestavěnými PSD regulátory s analogovým nebo binárním výstupem.

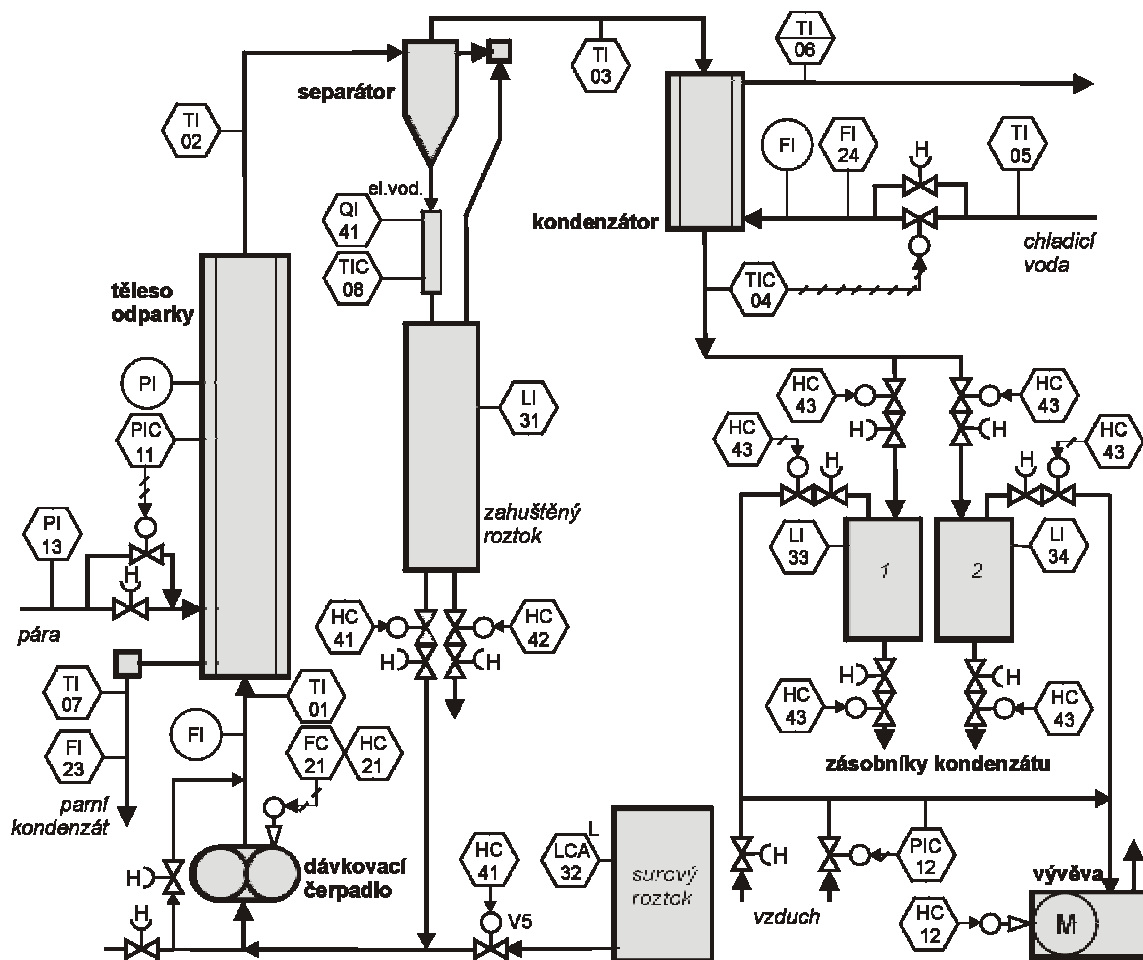
Tabulka 6.3 Typy I/O pro PLC OP-45 SB

Typ I/O	Počet	Napět'ová/proudová úroveň
digitální vstup (DI)	16	24 V=
digitální výstup (DO)	16	24 V=
analogový vstup (AI)	16	volitelně: 4-20 mA, 0-20 mA, 0-10 V, 0-5 V, 1-5 V
analogový výstup (AO)	6	volitelně: 4-20 mA, 0-20 mA, 0-10 V, 0-5 V, 1-5 V

Na centrální jednotce procesního počítače je panel se čtyřřádkovým alfanumerickým displejem a číselnou klávesnicí s dvanácti funkčními klávesami, ze kterých je možné v lokálním provozu ovládat některé základní funkce. V procesním počítači jsou naprogramovány veškeré algoritmy základní úrovně řízení (měření, logické řízení, regulace) a kontroly procesu (hlídání mezí veličin, diagnostika chybových stavů a blokace).



Obrázek 6.1 Vyobrazení řídicího panelu procesního počítače

Obrázek 6.2 Schéma stanice odparky s vyznačením okruhů měření a regulace ⁶

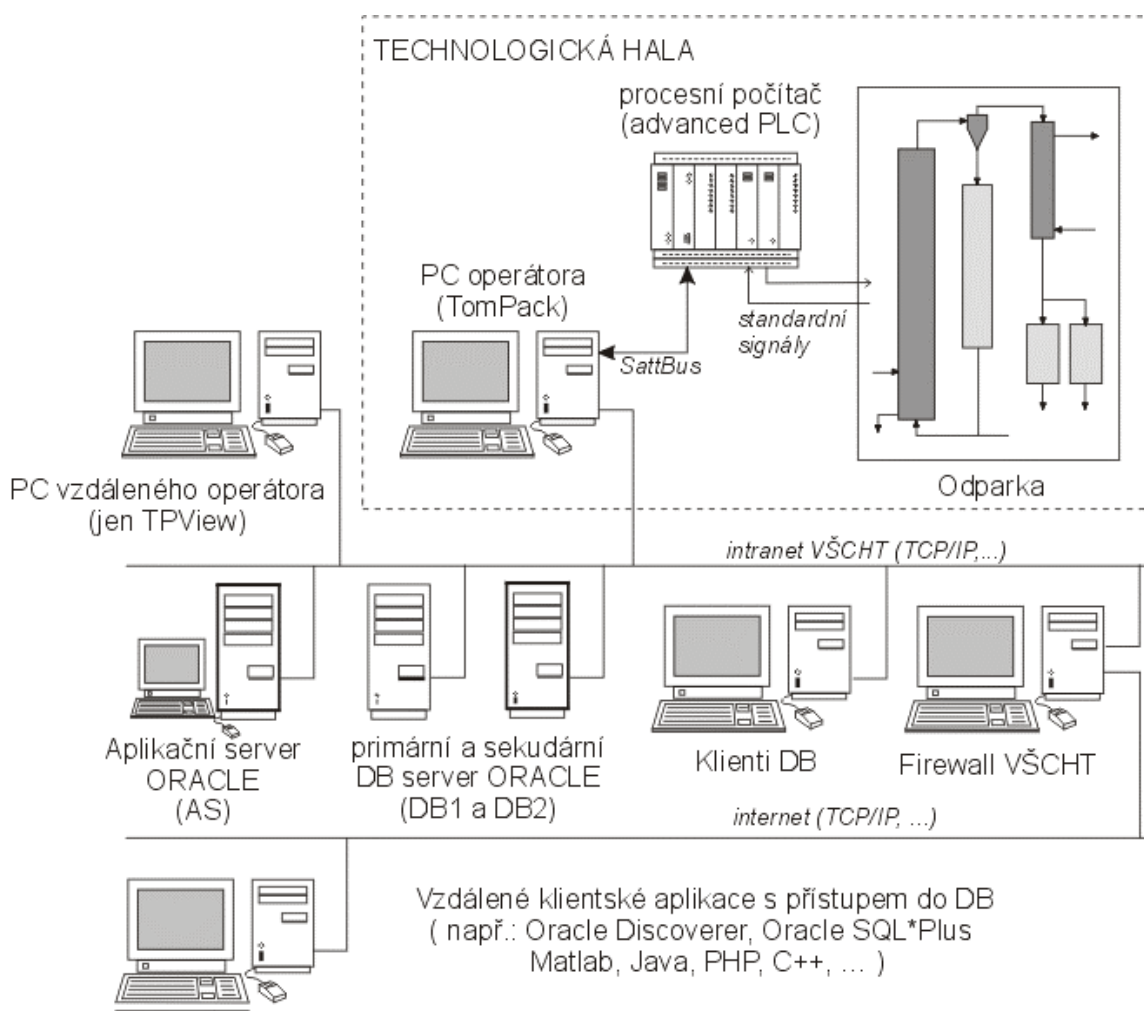
Tabulka 6.4 Popis použitých analogových (A) nebo digitálních (D) čidel (Č) a akčních členů (Ak)

Označení	Č/A	A/D	Použití	Výrobce	Signály
FI 23, 24	Č	A	čidla pro měření průtoku	Fischer-Rosemount	4-20 mA
PI 11, 12, 13	Č	A	čidla pro měření tlaku	Fischer-Rosemount	4-20 mA
TI 01 až 08	Č	A	čidla pro měření teploty	Senzit	0-10 V 4-20 mA
LI31	Č	A	čidlo výšky hladiny	Nivelco	4-20 mA
LI33, 34	Č	A	čidla výšky hladiny	Dinel	4-20 mA
QI 41	Č	A	koncentrační čidlo (el. vodivost)	Omega	4-20 mA
FC 21	Ak	A	dávkový čerpadlo	ProMinent	4-20 mA
okruhy PIC 11, 12 TIC 04	Ak	A	pneumatické regulační ventily	Baumann Fischer-Rosemount	4-20 mA
ostatní	Ak	D	regulační ventily (dvoupolohové)	Staspo	240 V~

6.3 Technický popis struktury komunikace

Procesní počítač PLC OP-45 SB je propojen přes průmyslovou sběrnici typu Sattbus (Alfa-Laval) s operátorským PC (OS Windows 2000). V tomto počítači je zabudována speciální komunikační karta (SattBus ISA) a nainstalován program pro vizualizaci a řízení TomPack (se všemi moduly TPServer, TPViewer, TPConfig i UniServer) (viz. kapitola 2).

Data z procesu jsou tedy přenášena standardními signály do programovatelného logického automatu PLC, dále pak přes dvojitvodičovou sběrnici SattBus do operátorského PC, odkud jsou pomocí ODBC driveru a implementovaných funkcí v programu TomPack (modul Server) ukládána přes školní TCP/IP síť do primární databáze. Z primární databáze mohou být data replikována do sekundárního serveru, ze kterého mohou být dále volně publikována pro vzdálené klientské aplikace. Schéma přenosu dat je znázorněno na následujícím obrázku.



Obrázek 6.3 Schéma komunikační struktury – síťová architektura

7 Volba, návrh a popis struktury databáze pro sběr procesních dat

Požadavky na databázový systém vycházejí z požadavků uživatele a nebo přímo z technologie⁸.

Ze strany technologie musí být databázový systém **dostatečně rychlý**, schopen ukládat velké množství informací v krátkých pravidelných časových intervalech pro sledování procesu, ale i pro vznik událostí (alarmy a jiná varování) i zcela nepravidelných. Přestože „náš“ systém pro měření, řízení odparky (včetně archivace) bude ze začátku ukládat procesní data s frekvencí pohybující se v řádu několika sekund, je zapotřebí vytvořit strukturu, která v budoucnosti bude schopna pojmout mnohonásobně větší množství údajů. Každý údaj musí nést přesnou informaci o čase, ale také o jeho spolehlivosti. Z toho vyplývá, že **nároky na paměť** databázového serveru jsou značné a je tedy snaha, je bez ztráty přesnosti ukládaných údajů minimalizovat. Jedním ze způsobů je možnost volby mezi **opakovaným** (cyklickým) ukládáním s pevně danou *periodou* nebo úspornějším **změnovým ukládáním**. V případě změnového ukládání mluvíme o tzv. *pásmu necitlivosti v hodnotové a časové oblasti* určujícím maximální změnu údaje/času, která se ještě v databázi neuloží. Při dotazování z databáze by mělo být možno specifikovat, kolik bude vrácených údajů a nebo jaký bude časový interval, ze kterého výběr (selekcí dat) vytváříme.

Z uživatelského hlediska musí být databázový systém spolu s nadřazeným, dostatečně rozšířeným operačním systémem **spolehlivý**, **otevřený** pro vývoj klientských aplikací a **kompatibilní** s různými komunikačními protokoly. Musí být schopen automatické **kontroly integrity**, **práce s objekty** různých datových typů včetně jejich importu, exportu a publikace v rámci intra / internetu. Samozřejmostí by měla být jednoduchá konfigurace a administrace systému (GUI) i kvalitní uživatelská dokumentace.

7.1 Technický popis použitých serverů

Na databázové a aplikační servery se kladou mnohem vyšší požadavky než na pracovní stanice a vyšší i než na běžný síťový server pro sdílení diskového prostoru a správy uživatelských kont. Každý server s důležitými daty (relativní pojem důležitosti) by měl mít na síti svůj sekundární (záložní) server.

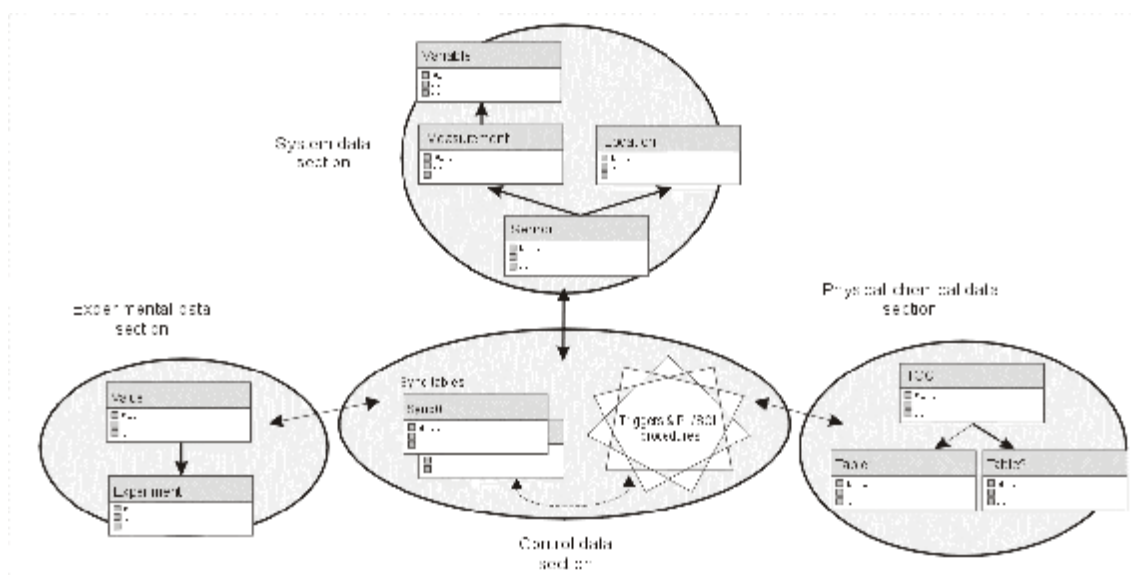
Pro náš systém měření, řízení a archivace jsou využity z pedagogických důvodů dva databázové servery **Oracle**, mezi nimiž bude probíhat průběžná replikace dat. Primární databázový server se systémem **Oracle8i** (DB1), který spravuje Ing. Z. Kokoška, slouží pouze pro komunikaci s procesním počítačem (transakcím pro sběr dat bude pomocí **Oracle Resource Manageru** přiřazena většina systémových zdrojů). V rámci diplomové práce jsem připravil po technické i programové stránce nový databázový a aplikační server Oracle. Sekundární server (DB2) s nejnovějším databázovým systémem **Oracle9i** bude publikovat kopii dat po intra / internetové síti pro off-line zpracování (např. identifikace systému s využitím matematického modelu) a dále je bude zprostředkovávat aplikačnímu serveru **Oracle9iAS** (AS) – internetové platformě pro vývoj prezentací a aplikací využitelných k řízení odparky.

Tabulka 7.1 Parametry aplikačního a databázových serverů

Parametr	Hodnota		
	DB Server 1 (DB1)	DB Server 2 (DB2)	Aplikační server (AS)
Platforma	PC	PC	PC
Procesor (CPU)	AMD K6 233 MHz	Intel Pentium II 266 MHz	Athlon 1700 MHz+
Operační paměť (RAM)	128 MB	256 MB	1024 MB
Disková paměť (HDD)	15 GB	60 GB	2 x 40 GB (RAID 0)
Ostatní HW	CD-ROM	U-IDE 100, CD-ROM	U-IDE 100 RAID, CD-ROM
Operační systém (OS)	Linux RedHat 6.2	MS Windows 2000 Server	MS Windows 2000 Server
RDBMS / AS	Oracle8i EE	Oracle9i EE s JServerem	Oracle9i AS

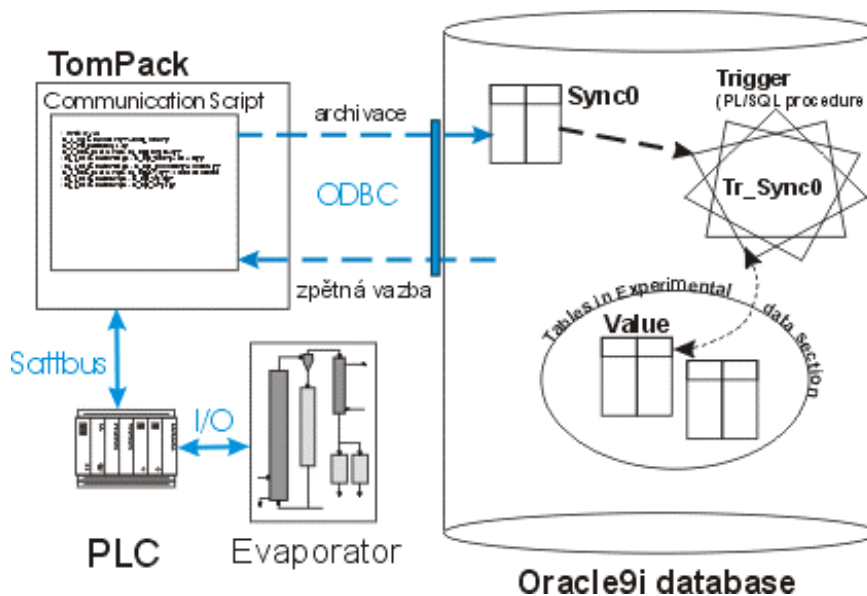
7.2 Struktura databáze

První verzi struktury databáze navrhl Ing. Z. Kokoška⁶. Předpokládala tři *sekcce* (v databázi Oracle označované jako schémata) pro možnost oddělení správy dat různými uživateli. Strukturu jsem modifikoval a rozšířil o synchronizační tabulky pro výpočty nad databází, trigger (spoušť) s výpočetními procedurami a o celou *Sekci fyzikálně-chemických dat (Physical-chemical data section)*, kde budou průběžně ukládány tabulky závislosti fyzikálně-chemických vlastností různých látek využitelných pro on-line řízení odparky. Pro sjednocení pojmenování objektů v databázi byly zvoleny anglické názvy a zavedena obvyklá konvence označení **primárních - PK** (id_XXX) a **cizích - FK klíčů** (XXX_id). Tzv. klíče jsou³⁸ pole nebo kombinace více polí, které jednoznačně určují záznam v databázové tabulce. Nejčastěji jsou primárními klíči indexy tabulek, přičemž hodnoty pro indexy jsou generované ze sekvencí.



Obrázek 7.1 Nová struktura databáze pro sběr procesních dat

Sekce dat pro řízení (Control data section) nyní obsahuje celkem tři synchronizační tabulky a dva triggerry s PL/SQL procedurami. Do synchronizační tabulky **Sync0** data ukládá pouze program **TomPack** pomocí databázového rozhraní typu ODBC podle následujícího schématu:



Obrázek 7.2 Schéma přenosu dat do databázové tabulky systému Oracle

Po vložení nového řádku do tabulky Sync0, reprezentujícího vybrané aktuální hodnoty veličin měřených na odparce, je aktivována spoušť **Tr_sync0** obsahující PL/SQL proceduru pro rozmístění všech informací do definované datové struktury (tabulka Value) a vymazání tohoto dočasného řádku ze synchronizační tabulky (prozatím není aktivní). V následující tabulce je uveden zkrácený výpis informací o sloupcích v tabulce Sync0:

Tabulka 7.2 Podrobné informace o sloupcích synchronizační tabulky Sync0 (zkrácený výpis)

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
id_sync0	NUMBER	PK, NOT NULL	index pro tabulku
tstamp	VARCHAR2(32)	NOT NULL	časová značka
experiment_id	NUMBER	FK	id experimentu
senzor_id_01	NUMBER		číselné „id“ pro první senzor
value_01	NUMBER		naměřená hodnota - údaj pro senzor01
senzor_id_02	NUMBER		číselné „id“ pro druhý senzor
value_02			naměřená hodnota - údaj pro senzor02
...	NUMBER		...další dvojice senzor_id a value

Dále jsou v této sekci další dvě synchronizační tabulky **Sync1** a **Sync2** stejné struktury a trigger **Tr_calc**. Tyto tři objekty slouží pro on-line výpočty nad databází a jejich funkce je podrobněji popsána v kapitole 9.1.2.

Tabulka 7.3 Podrobné informace o sloupcích synchronizačních tabulek Sync1 a Sync2

Název sloupce	Datový typ	Integritní omezení	Popis sloupce
tstamp	VARCHAR2(32)	PK, NOT NULL	časová značka výpočtu
flag	NUMBER	NOT NULL	stav výpočtu
procedure	VARCHAR2(32)		jméno procedury pro výpočet
tablename	VARCHAR2(32)		jméno zdrojové tabulky
x0	NUMBER		výsledek výpočtu
x1	NUMBER	NOT NULL	první proměnná
x2	NUMBER		druhá proměnná

Sekce experimentálních dat (Experimental data section) obsahuje dvě tabulky **Value** a **Experiment**. Tabulka Value obsahuje vlastní naměřená data (údaje zvolené pro archivaci – viz. Příloha 5) a tabulka Experiment všeobecné informace o pokusu.

Tabulka 7.4 Podrobné informace o sloupcích tabulky Value

Název sloupce	Datový typ	Integritní omezení	Popis sloupce
id_value	NUMBER	PK, NOT NULL	index pro tabulku
tstamp	DATE	PK	časová značka
senzor_id	NUMBER(5)	FK, NOT NULL	odkaz na id_senzor
value	NUMBER	NOT NULL	naměřená hodnota
experiment_id	NUMBER(5)	FK	odkaz na id_experiment

Tabulka 7.5 Podrobné informace o sloupcích tabulky Experiment

Název sloupce	Datový typ	Integritní omezení	Popis sloupce
id_experiment	NUMBER	PK, NOT NULL	index pro tabulku
username	VARCHAR(50)	NOT NULL	jméno osoby, která pokus prováděla
start_time	DATE	NOT NULL	datum a čas začátku experimentu
stop_time	DATE	NOT NULL	datum a čas konce experimentu
desc	VARCHAR(400)		popis experimentu

Sekce systémových dat (System data section) se skládá celkem ze čtyř tabulek, obecně o čidle a typu měření. Hlavní je zde tabulka **Senzor**, jejíž údaje vypovídají o použitém čidle, dále tabulka **Location** o fyzickém umístění čidla na odparce, tabulka **Measurement** o typu měření a nakonec tabulka **Variable** obsahuje informace o měřené veličině a její jednotce.

Tabulka 7.6 Podrobné informace o sloupcích tabulky Senzor

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
id_senzor	NUMBER	PK, NOT NULL	index pro tabulku
measurement_id	NUMBER	FK, NOT NULL	odkaz na id_measurement
location_id	NUMBER	FK, NOT NULL	odkaz na id_location
senzor_range	VARCHAR2(32)		rozsah senzoru
manufacurer	VARCHAR2(255)		výrobce
serial_num	VARCHAR2(255)		sériové číslo

Tabulka 7.7 Podrobné informace o sloupcích tabulky Location

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
id_location	NUMBER	PK, NOT NULL	index pro tabulku
location	VARCHAR2(32)		umístění čidla na odparce
senzor_sign	VARCHAR2(400)		označení senzoru na schématu

Tabulka 7.8 Podrobné informace o sloupcích tabulky Measurement

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
id_measurement	NUMBER	PK, NOT NULL	index pro tabulku
variable_id	NUMBER	FK, NOT NULL	odkaz na id_variable
measurement	VARCHAR2(255)		typ měření

Tabulka 7.9 Podrobné informace o sloupcích tabulky Variable

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
id_variable	NUMBER	PK, NOT NULL	index pro tabulku
variable	VARCHAR2(32)	NOT NULL	značka veličiny
var_desc	VARCHAR2(255)		popis veličiny
unit	VARCHAR2(32)		jednotka

Sekce fyzikálně-chemických dat (*Physical-chemical data section*) obsahuje tabulky závislostí fyz.-chem. vlastností různých látek **Table1**, **Table2** (podobné struktury jako Table1, jen chybí sloupec x2) a dále jedna souhrnná tabulka **TOC** s popiskami ostatních tabulek.

Tabulka 7.10 Podrobné informace o sloupcích tabulky Table1 (Table2)

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
id_table1	NUMBER	PK, NOT NULL	index pro tabulku
x0	NUMBER	NOT NULL	závislá proměnná
x1	NUMBER	NOT NULL	první nezávislá proměnná
x2	NUMBER		druhá nezávislá proměnná

Tabulka 7.11 Podrobné informace o sloupcích tabulky TOC

Název sloupce	Datový typ	Integritní omezení	Popis funkce sloupce
toc_id	NUMBER	PK, NOT NULL	index pro tabulku
desc_us	VARCHAR2(400)		popis závislosti (anglický jazyk)
desc_cz	VARCHAR2(400)		popis závislosti
fcn	VARCHAR2(32)		funkční závislost
var_cnt	NUMBER		počet proměnných
x0	VARCHAR2(255)		popis závisle proměnné
x1	VARCHAR2(255)		popis první nezávisle proměnné
x2	VARCHAR2(255)		popis druhé závisle proměnné

Celou strukturu vytvoří správce databáze (konkrétně databázový uživatel „system“) spuštěním inicializačních PL/SQL skriptů v programu **SQL*Plus**. (viz. Příloha 1)

8 Návrh základního programového vybavení

Základní rozdělení programového vybavení neboli klientských aplikací s přístupem do databáze Oracle je možné provést následovně :

Tabulka 8.1 Programové vybavení pro správu DS a operace s daty

Klientská aplikace dodavatel a typ	použítá pro	
	správu a administraci DS	operace s daty
Oracle – native, basic	<i>Oracle SQL*Plus</i>	<i>Oracle SQL*Plus</i>
Oracle – native, GUI	<i>Oracle Enterprise Manager</i>	<i>Oracle Discoverer</i>
jiný (third-side)	<i>SQL Programmer, ...</i>	<i>Matlab, (TomPack) ...</i>

Nativní aplikace firmy Oracle používají ke komunikaci s databází Oracle vlastní komunikační protokol **SQL*Net** (ve staré verzi Oracle8 to byl Net8). Naproti tomu klientské aplikace třetích stran musí použít ke komunikaci s datovými zdroji Oracle specifický typ databázového rozhraní (viz. kapitola 1.1.2). Nejčastěji využívané standardy rozhraní jsou **ODBC** ve Windows a **JDBC** pro jiné platformy (Linux, atd.). V dalších kapitolách Vás seznámím s konkrétními postupy nastavení systému MS Windows, které umožní aplikacím přístup do vzdáleného databázového serveru Oracle8i/9i.

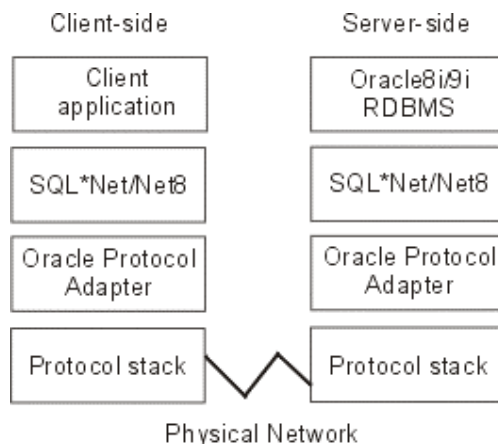
8.1 Nastavení systému pro přístup do databáze

8.1.1 Pro Oracle-nativní aplikace (SQL*Net)

SQL*Net je softwarová vrstva umožňující obousměrnou komunikaci mezi klientem a databází Oracle. Mezi podporované síťové protokoly patří :

- ❖ TCP/IP bez šifrování - definice *hostmane:port* (obvykle 1521)
- ❖ TCP/IP s SSL šifrováním - definice *hostmane:port* (obvykle 2484)
- ❖ IPC (InterProcess Communication) - definice *IPC hodnoty/klíče*
- ❖ NMP (NaMed Pipes) - definice tzv. *pipename* (datové „trubky“)

V našem případě využíváme síťového protokolu TCP/IP bez šifrování standardně adresovaného na portu 1521. Komponenta SQL*Net se instaluje s libovolným produktem firmy Oracle a ihned po instalaci vyžaduje nastavení pomocí konfiguračního průvodce **Net Configuration Assistant** nebo dodatečně pomocí **Net Manageru**.



Obrázek 8.1 Architektura komunikace mezi klientem a databází Oracle (SQL*Net)

Tito průvodci definují tzv. **TNS aliasy** (Transparent Network Substrate names), které se ukládají do lokálního konfiguračního souboru (ORA_HOME\$ je proměnná s názvem kořenového adresáře Oracle), konkrétně:

- ❖ ORA_HOME\$/network/admin/TNSnames.ora pro Oracle9iClient
- ❖ ORA_HOME\$/net8/admin/TNSnames.ora pro Oracle8iClient

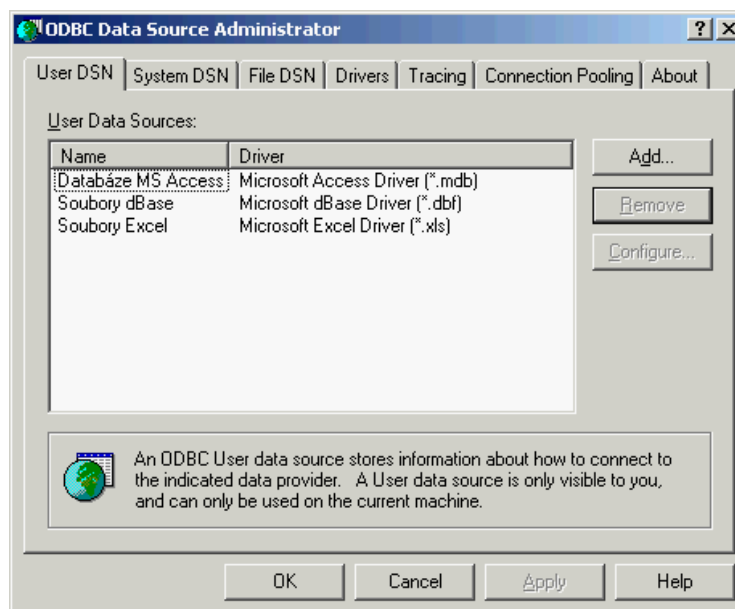
Tyto TNS aliasy používáme jako *přihlašovací řetězce* v klientských programech (pro anglické verze aplikací známé jako *connect string* nebo také *host string*).

Tabulka 8.2 Konkrétní konfigurační řetězce (TNS aliasy) pro databázové servery Oracle na VŠCHT

Konkrétní konfigurační řetězce (TNS aliasy)	
pro DB Server 1	pro DB Server 2
<pre> KOKY.VSCHT.CZ = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (Host = a332-1) (Port = 1521))) (CONNECT_DATA = (SID = odparka1))) </pre>	<pre> DB90.VSCHT.CZ = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (Host = adolf) (Port = 1521))) (CONNECT_DATA = (SID = DB90))) </pre>

8.1.2 Pro aplikace třetích stran

Klientské programy třetích stran se pro přístup k datovým zdrojům odkazují na pojmenování zdroje tzv. **DSN alias** (Data Source Name). DSN alias se konfiguruje pomocí **ODBC Administratora**, který vyvoláme dvojitým kliknutím na ikonu *Datový zdroj (ODBC)* z nabídky *START / Nastavení / Ovládací panely / Nástroje pro správu* (platné pro MS Windows 2000 Professional CZ).



Obrázek 8.2 Úvodní nabídka ODBC Administrátora

Tento průvodce nastavením ODBC přidává, odebrává a konfiguruje ovladače a zdroje rozhraní ODBC. Je možné uložit tři typy DSN aliasů (viz. kapitola 1.1.2.2) a vytvářet logovací soubory pro krokování/ladění činnosti ODBC ovladačů. Pro komunikaci s databází Oracle8i/9i můžeme potenciálně využít jeden z těchto ovladačů:

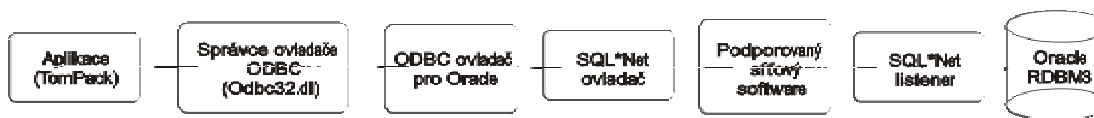
- ❖ Microsoft ODBC for Oracle
- ❖ Oracle ODBC for Oracle9i nebo Oracle8i

Ovladač **Microsoft ODBC for Oracle** je součástí základní instalace MS Windows (balíček MDAC). Umožňuje přistupovat jak k lokální, tak k vzdálené databázi přes intra/internetovou síť kompatibilní s SQL*Net. Podporuje např. i přístup k uloženým PL/SQL procedurám a integrace s XA/DTC. Tento ovladač je ODBC 2.5 kompatibilní, což znamená, že vyhovuje shodě s SQL Level Core, shodě s API Level 1, podporuje některé z funkcí API Level 2 a syntaxe s Extended SQL. V praxi to znamená, že plně podporuje komunikaci s databází Oracle 7.3.x a částečně i Oracle8 a Oracle9i. Týká se to hlavně práce s novými datovými typy obsaženými v Oracle8/9i.

Ovladače **Oracle ODBC for Oracle8i/9i** jsou nativní drivery od firmy Oracle. Přestože jsou inovované verze těchto ovladačů (www.oracle.com) distribuovány i separátně vyžadují vždy alespoň minimální instalaci klientského softwaru *Oracle Client 8i/9i*.

Pro síťovou komunikaci s datovými zdroji Oracle je tedy třeba mít :

- klientskou aplikaci (např. TomPack)
- Správce(manager) ovladače ODBC
- ODBC ovladač(driver) pro Oracle (jeden z výše uvedených)
- počítačovou síť podporující SQL*Net spojení
- SQL*Net Listener (součástí databázového systému Oracle8i/9i)
- Oracle RDBMS (Oracle8i/9i)



Obrázek 8.3 Architektura ODBC komunikace aplikace a databáze Oracle

Pozn.: Při instalaci staršího ODBC ovladače vedle novější verze DS Oracle9i (do jiného adresáře definovaného proměnnou ORA_HOME\$) se zobrazí chybové hlášení „*Správce pro nastavení ovladače: Rutiny pro nastavení ovladače ODBC Oracle in ORA_HOME\$ nelze načíst z důvodu systémové chyby s kódem 126*“ a následně varování „*Funkce ConfigDSN, ConfigDriver nebo ConfigTranslator selhala: Nelze načíst knihovnu pro nastavení nebo knihovnu pro převaděče*“.

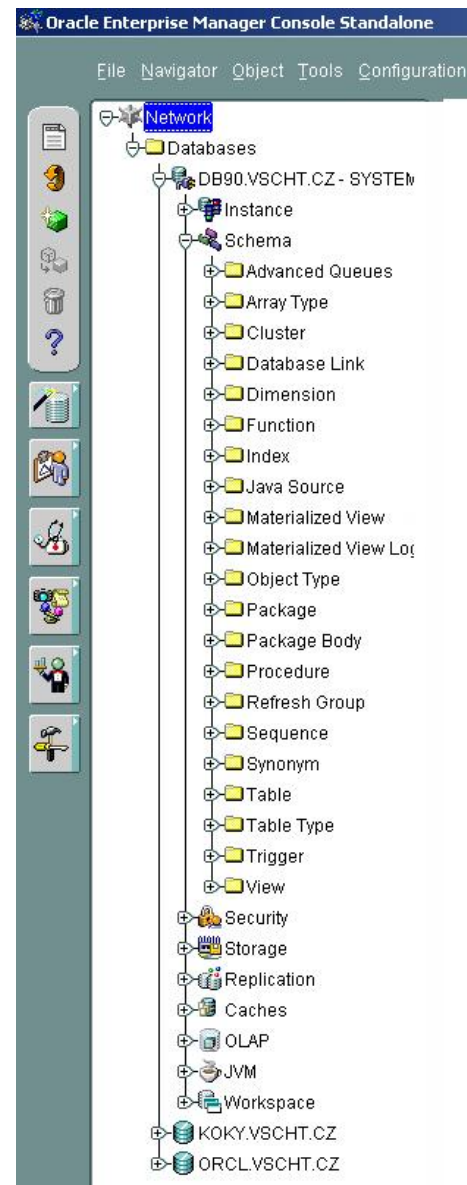
8.2 Programové vybavení pro administraci a obsluhu db

Pro administraci databáze je vytvořen **Oracle Enterprise Manager**, který je součástí instalaci **databáze Oracle 9i** (server-side) a nebo sady klientských aplikací **Oracle Client9i** (client-side). Databázi lze však řídit na nejnižší úrovni i pomocí příkazů jazyka SQL (typu DCL) s využitím programu **SQL*Plus**. Dalším programem pro správu databáze s velmi příjemným uživatelským prostředím je **SQL-Programmer**. Všechny tyto programy slouží k editaci, údržbě a testování validity objektů databáze.

8.2.1 Oracle Enterprise Manager

Oracle Enterprise Manager (OEM) je plně grafické, uživatelské rozhraní pro centralizovanou správu databází vytvořené v jazyce Java. Uživatel s dostatečnými přístupovými právy do databáze (většinou databázový administrátor) se po spuštění tohoto programu může přihlásit k RDBMS dvěma způsoby. Buď jako samostatný systém řízení (stand-alone) a nebo k řídicímu serveru **Oracle Management Server**, který spravuje několik databázových systémů najednou. Z důvodu absence Oracle Management Serveru na školní síti VŠCHT, je možná jen první varianta.

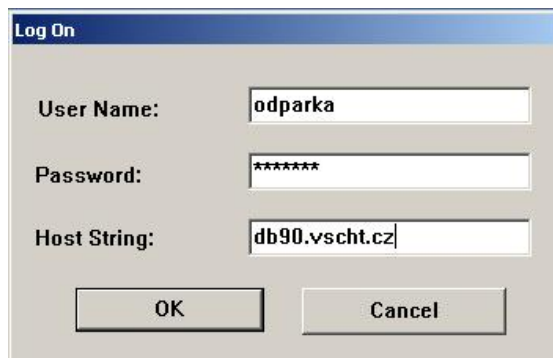
Zobrazí se stromovitá nabídka(menu) s hlavním kořenem *Network/Databases* a TNS aliasy všech dostupných databázových systémů. Po úspěšném přihlášení k některé z databází se nabídka rozvine o další úroveň, jejíž položky reprezentují jednotlivé komponenty OEM uvedené v tabulce 3.4. V ovládacím panelu *Navigator* jsou důležité řadící položky *By schema* a *By object*. V režimu řazení *By Object* jsou databázové objekty všech uživatelů (včetně systémových) pod jednou záložkou (např. *Procedure* či *Tables*) naopak v režimu *By schema* je výčet všech objektů pro každého uživatele zvlášť. Vše názorně vyobrazeno na vedlejším obrázku (režim *By Object*).



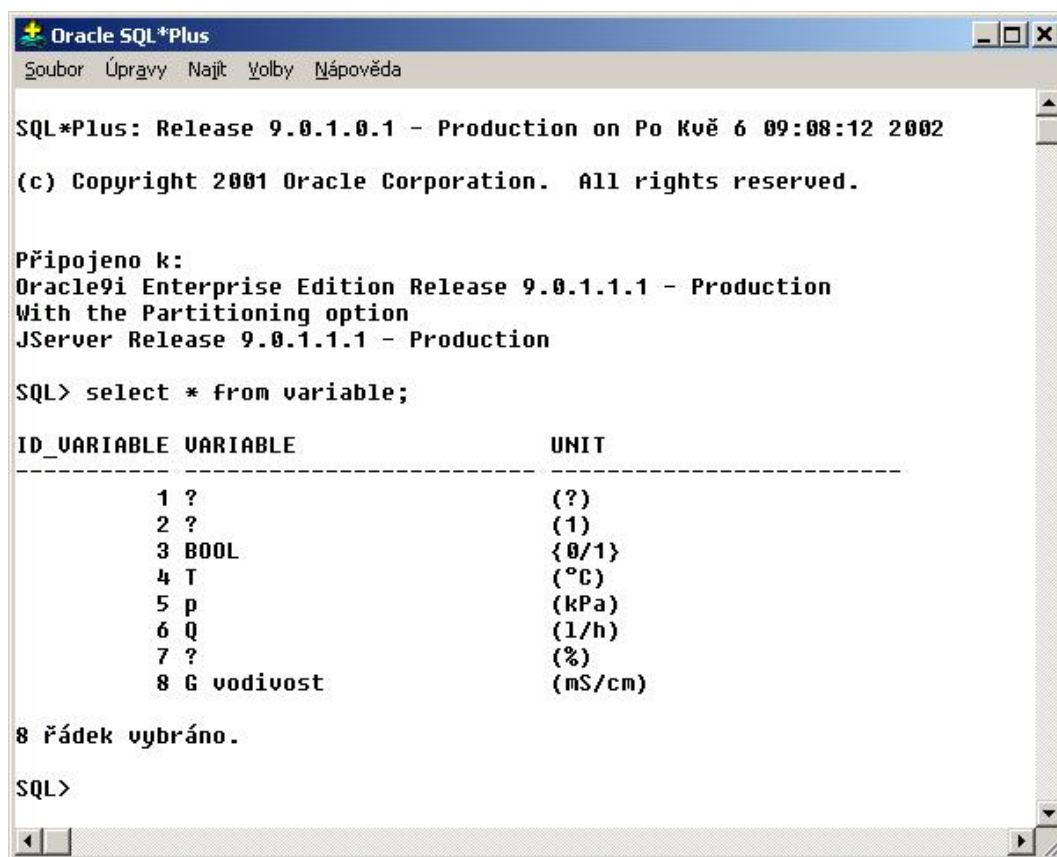
Obrázek 8.4 Stromová nabídka OEM

8.2.2 Oracle SQL*Plus

Oracle SQL*Plus je textově orientovaný interpret příkazů jazyka SQL. Po přihlášení k databázi Oracle se zobrazí okno s příkazovým řádkem začínajícím řetězcem „SQL>“ akceptující příkazy pro manipulace s daty (DML), pro definice datových struktur (DDL) i pro řízení a správu databází (DCL).



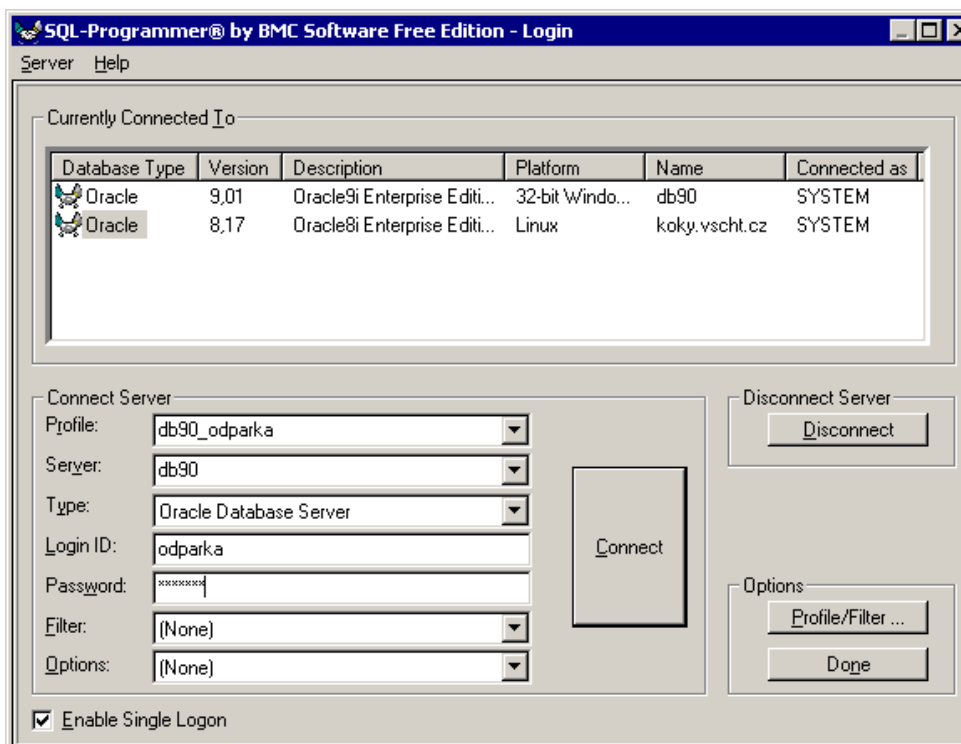
Obrázek 8.5 Přihlášení k databázi Oracle (SQL*Plus)



Obrázek 8.6 Textově orientovaný interpret příkazů Oracle SQL*Plus

8.2.3 SQL-Programmer

SQL-Programmer je vývojové prostředí od firmy **BMC Software Inc.**⁴¹ pro tvorbu objektů v prostředí relačních databází založených na standardu SQL. Je to velmi universální prostředí podporující komunikaci s *IBM DB2*, *Oracle*, *Microsoft* i *Sybase SQL Serverem*. Na rozdíl od SQL*Plus, přináší SQL-Programmer řadu užitečných vlastností jako jsou např. správa „login-profilů“ pro usnadnění přihlašování či možnost paralelního připojení k více databázovým serverům.



Obrázek 8.7 Přihlášení k databázi Oracle v programu SQL-Programmer

Uživatelům databázového serveru Oracle usnadňuje tvorbu funkcí a procedur včetně triggerů, ale také indexů, tabulek, pohledů i synonym. Umožňuje programové bloky kompilovat i pouštět a tím zjišťovat jejich validitu.

V Příloze 7 je vyobrazeno grafické vývojové prostředí programu SQL-Programmer a jeho rozdělení do několika částí. Základní nabídka pro správu souborů, tiskárny a operace se schránkou je umístěna standardně v horizontálním panelu nahoře. V pravé části obrazovky je stromovitá nabídka s databázovými objekty velice podobná jako v Oracle Enterprise Manageru. V dolní části jsou dvě okénka pro ladění programů (debugging) a přehled parametrů, lokálních a globálních proměnných (variable). Největší okno v pravé části obrazovky je vlastní pracovní plocha pro úpravu, nastavení a sledování chodu programů a jiné výstupy.

8.3 Programové vybavení pro operace s daty

8.3.1 Oracle Discoverer

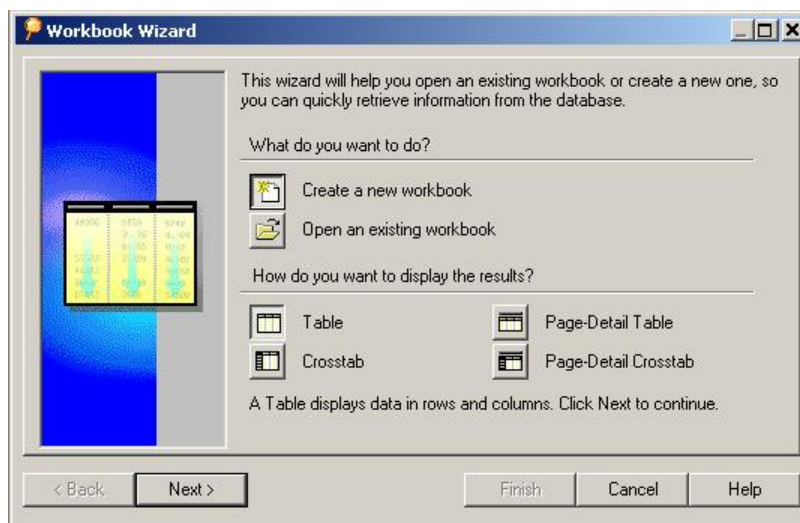
Pro přístup k datům firma Oracle vytvořila produkt **Oracle Discoverer** z balíčku **Oracle Internet Developer Suite** ¹⁰. Je to dotazovací, reportovací a analytický nástroj pro přístup k datům s jednoduchým grafickým rozhraním (GUI). Zjednodušuje práce s vytvářením grafů a sestav, s vyhledáváním dat i rozesíláním výsledků pomocí intranetu nebo Internetu. Nástroj je zaměřen na běžného uživatele a plně odpovídá jeho požadavkům a potřebám při zachování vysokého výkonu a jednoduché centrální administraci.

Je rozdělen na dvě části. **Administrativní** část (Administrative Edition) slouží k vytváření a odstraňování tzv. „**End User Layer (EUL)**“, což jsou specifické indexy, tabulky a pohledy sloužící pro vlastní potřebu programu Discoverer/Desktop Edition. Tvorba EUL se provádí jen jednou a to při prvním přihlášení do databáze Oracle.



Obrázek 8.8 Přihlášení do databáze Oracle v klientské aplikaci Discoverer

Průvodce poté oznámí, že nemáte přístup do žádné EUL, která je podmínkou pro práci Discovererem. Potvrdíme vytvoření EUL a vyčkáme než se vytvoří. Pak ukončíme Administratora a spustíme část **uživatelskou** (User/Desktop Edition), kde definujeme nový projekt vybráním konkrétních sloupců z konkrétních tabulek.



Obrázek 8.9 Vytvoření nového projektu v programu Discoverer

Následující obrázky - tabulka a znázornění průběhu veličin - jsou generovány ze skutečných dat [teplotní čidla TI01 až TI05 – viz. Obrázek 6.2] naměřených na odparce a uložených v databázi Oracle.

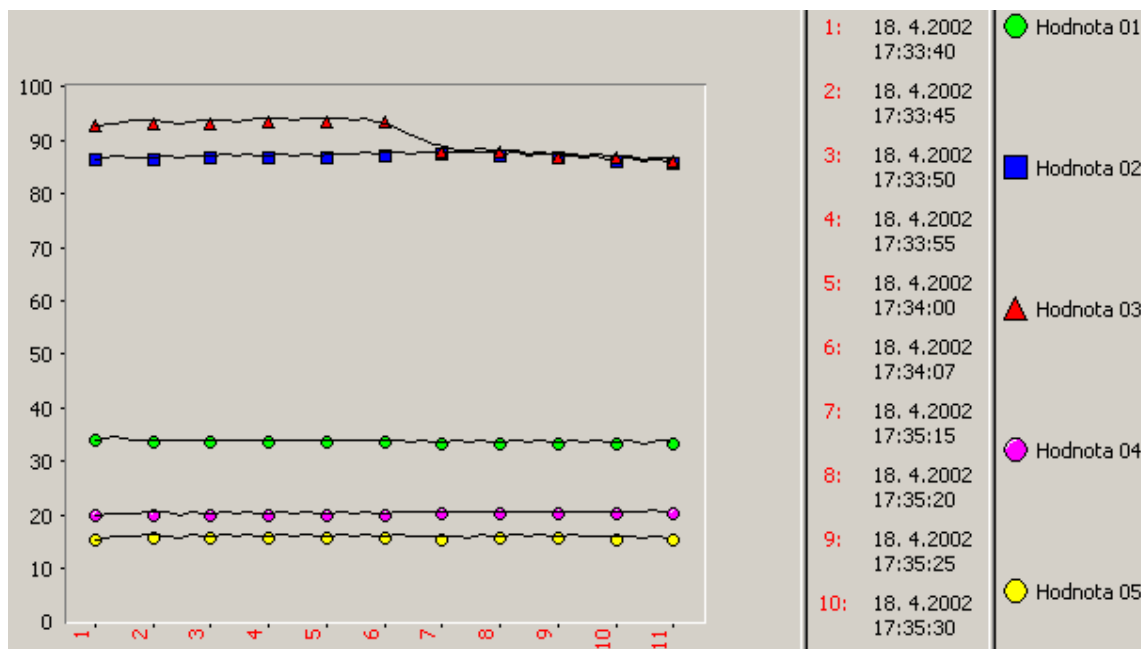
Oracle Discoverer - [Workbook1]

File Edit View Sheet Format Tools Graph Window Help

Tahoma 8 B U

	Tstamp	Hodnota 01	Hodnota 02	Hodnota 03	Hodnota 04	Hodnota 05
1	18. 4.2002 17:33:40	33.98	86.52	92.77	19.92	15.43
2	18. 4.2002 17:33:45	33.79	86.52	93.17	19.92	15.53
3	18. 4.2002 17:33:50	33.79	86.72	93.17	19.92	15.53
4	18. 4.2002 17:33:55	33.79	86.72	93.36	19.92	15.53
5	18. 4.2002 17:34:00	33.79	86.92	93.36	19.92	15.53
6	18. 4.2002 17:34:07	33.79	87.11	93.56	20.02	15.53
7	18. 4.2002 17:35:15	33.40	87.50	87.89	20.12	15.43
8	18. 4.2002 17:35:20	33.40	87.11	87.89	20.12	15.53
9	18. 4.2002 17:35:25	33.40	86.72	86.92	20.22	15.53
10	18. 4.2002 17:35:30	33.40	86.13	86.92	20.22	15.43
11	18. 4.2002 17:35:35	33.40	85.74	86.13	20.22	15.43

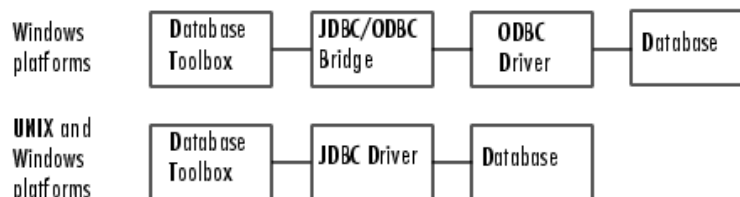
Obrázek 8.10 Část tabulky s reálnými daty (Oracle Discoverer)



Obrázek 8.11 Časový průběh reálných dat z odparky (Oracle Discoverer)

8.3.2 Matlab - Database Toolbox

Multiplatformní, komplexní, objektově orientovaný program **Matlab** firmy **MathWorks**⁴⁰ obsahuje modul nazvaný **Database Toolbox** umožňující import i export dat jakékoliv databáze podporující komunikaci s využitím databázových rozhraní typu ODBC a JDBC.



Obrázek 8.12 Přístup k databázím z Database Toolbox (Matlab) ve Windows a Linux

Sada nástrojů Database Toolbox ve verzi 2.1 obsahuje příkazy (funkce) rozdělené do několika skupin (např. pro nastavení ovladačů a správce ovladače nebo pro přístup k datům). K nejdůležitějším příkazům patří příkaz `database`, který vytváří databázový objekt s informacemi o spojení k datovému zdroji, dále příkaz `exec`, který vykoná vlastní SQL příkaz (typu `SELECT`) a otevírá kurzor a nakonec příkaz `fetch` pro přenesení - načtení dat do objektu Matlabu (matice).

Tabulka 8.3 Algoritmus načtení proměnných programem Matlab z databáze Oracle

```

conn = database ('ADOLF_ODBC', 'odparka', 'odparka');
curs = exec (conn, 'select * from table1');
curs = fetch (curs);
curs.Data
  
```

Tabulka 8.4 Další užitečné příkazy z Database Toolbox, Matlab

Příkaz	Funkce příkazu
<code>chkprops</code>	mění vlastnosti databázových objektů Matlabu
<code>ping</code>	vrací informace o stavu databáze
<code>logintimeout</code>	nastavuje max. čas na vytvoření spojení s databází
<code>insert</code>	exportuje pole proměnných (matici) Matlabu do databázové tabulky
<code>update</code>	přepíše data v databázové tabulce daty z Matlabu
<code>commit</code>	potvrdí předchozí datové transakce
<code>rollback</code>	zruší předchozí datové transakce
<code>rows</code>	zjistí počet řádků přenesené sady dat (ResultSet)

Pro více informací použijte příkaz Matlabu „`help database`“.

8.3.3 Program TomPack jako databázový klient

Program TomPack³ podporuje komunikaci s datovými zdroji v rámci svých ODBC funkcí (viz Příloha 6). Pro snazší ladění programů, všechny tyto funkce vrací logické návratové hodnoty (typ Boolean: úspěch „true“, neúspěch „false“). Po definici a otestování datového zdroje ODBC (viz. kapitola 7.1.2), příkazem `DBConnect` otevřeme spojení s databází. Dále připravíme tabulku pro přenos všech proměnných (`DBInitTable`) a podle datového typu, každou z proměnných připojíme pro přenos jedním z `DBBind%` příkazů (`DBBindBool`, `DBBindInt`, `DBBindReal`, `DBBindString`). Teprve nyní otevřeme tabulku (`DBOpenTable`) s daným filtrem a řadící podmínkou. Pro čtení dat posouváme ukazatelem (`DBMoveFirst`, `DBMoveLast`, `DBMoveNext`, `DBMovePrev`) po řádcích tabulky a tím aktualizujeme proměnné.

Tabulka 8.5 Algoritmus čtení proměnných programem TomPack z tabulky databáze Oracle

```
DB_B_conn = DBConnect(h, "ADOLF_ODBC");
DB_B_init = DBInitTable(h2, h);
DB_B_bind = DBBindReal(h2, var1, "[X1]");
DB_B_bind = DBBindReal(h2, var2, "[X2]");
DB_B_open = DBOpenTable(h2, "[TABLE1]", "", "");
DB_B_move = DBMoveFirst(h2);
DB_B_close = DBCloseTable(h2);
DB_B_disconn = DBDisconnect(h);
```

Pro kontrolu pozice ukazatele používáme `DBIsBOF` (vrací true, je-li ukazatel za poslední položkou) nebo `DBIsEOF` (vrací true, je-li ukazatel před poslední položkou). Pro zápis proměnných použijeme příkaz `DBAddNew`. Nakonec uzavřeme tabulku i spojení s databází (`DBCloseTable`, `DBDisconnect`).

Tabulka 8.6 Algoritmus zápisu proměnných programem TomPack do tabulky databáze Oracle

```
DB_B_conn = DBConnect(h, "ADOLF_ODBC");
DB_B_init = DBInitTable(h2, h);
DB_B_bind = DBBindReal(h2, var1, "[X1]");
DB_B_bind = DBBindReal(h2, var2, "[X2]");
DB_B_open = DBOpenTable(h2, "[TABLE1]", "", "");
DB_B_addnew = DBAddNew(h2);
DB_B_close = DBCloseTable(h2);
DB_B_disconn = DBDisconnect(h);
```

Pozn.: Vzhledem k tomu, že příkaz `DBConnect` (pro TomPack v1.10) neumožňuje definici uživatelského jména a hesla jako parametry, je nutné využít starší ODBC ovladač - **Oracle ODBC for Oracle verze 8.1.6.6** – který podporuje syntaxi „*user/pasword*“. Jakýkoliv jiný ODBC ovladač (<v8.1.6.6, v9.0.1.1 ani v9.0.1.2) totiž vyžaduje zadání hesla pro každé přihlášení (pro každý `DBConnect`).

9 Aplikační programy

Aplikační programy budou sloužit pro jednorázové či cyklické výpočty nad databází Oracle pro měření a řízení odparky programem TomPack.

Obecný návod k vytváření aplikačních programů

1. stanovit strukturu zdrojových a synchronizačních tabulek a jejich umístění
2. vytvořit trigger (spoušť) nad synchronizační tabulkou s výkonnou procedurou
3. vytvořit skript v TomPacku pro ukládání a čtení ze synchronizačních tabulek
4. zobrazování výsledků v programu TomPack

9.1 On-line odhad koncentrace roztoku sacharózy

Pro ověření funkčnosti navržené struktury databáze a komunikace programu TomPack s databázovým serverem Oracle bylo třeba vytvořit aplikaci, která by pracovala s aktuálními technologickými daty a ostatními informacemi uloženými v databázi a výsledky svých výpočtů předávala zpět do operátorského počítače v co nejkratším čase.

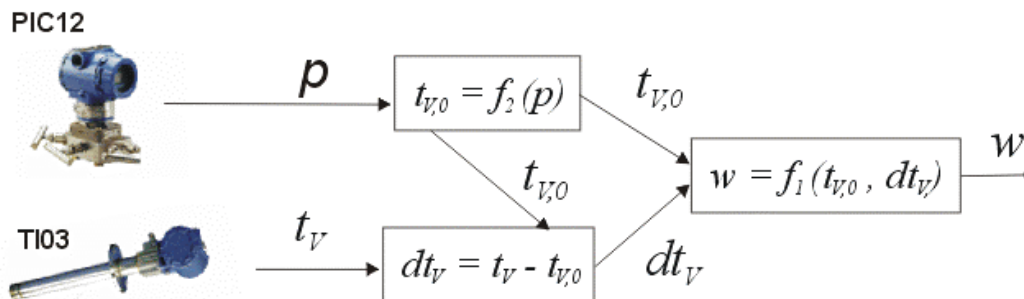
9.1.1 Zadání a teoretické řešení

Mým úkolem bylo navrhnout a vytvořit komplexní aplikaci pro on-line odhad koncentrace zahušťovaného roztoku sacharózy w (hm. %) na základě dvou měřených hodnot na odparce, konkrétně z hodnoty teploty varu zahušťovaného roztoku (resp. teploty par po separaci) t_V (°C) a tlaku v systému p (kPa) [čidla PIC12 a TI03 – viz. Obrázek 6.2]. Vypočtená hodnota koncentrace se má periodicky zobrazovat na panelu operátora (TPView).

K tomuto odhadu jsem v literatuře našel dvě tabulky (závislosti) fyzikálně-chemických vlastností (pro roztok sacharózy a pro vodu), převedl je do tabulek databáze Oracle a vytvořil program pro výběr hodnot a výpočty (s využitím interpolace) nad těmito tabulkami.

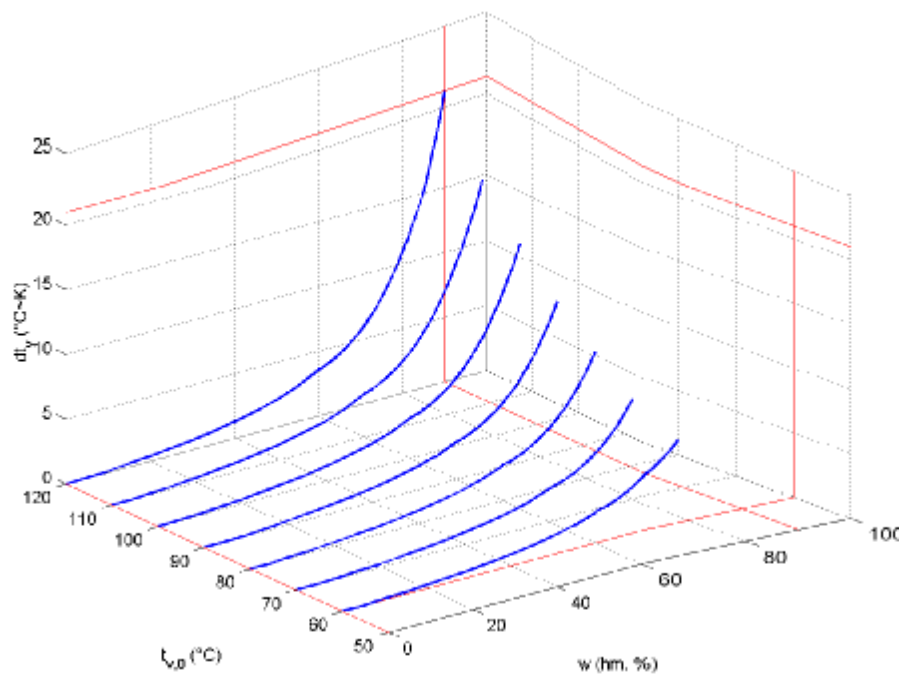
Tabulka 9.1 Fyzikálně-chemické závislosti, jejich tabulky a skripty

Závislost	Jméno tabulky	Skript pro vytvoření tabulky
$w = f_1(t_{V,0}, dt_V)$	table1	t_table1.sql
$t_{V,0} = f_2(p)$	table2	t_table2.sql

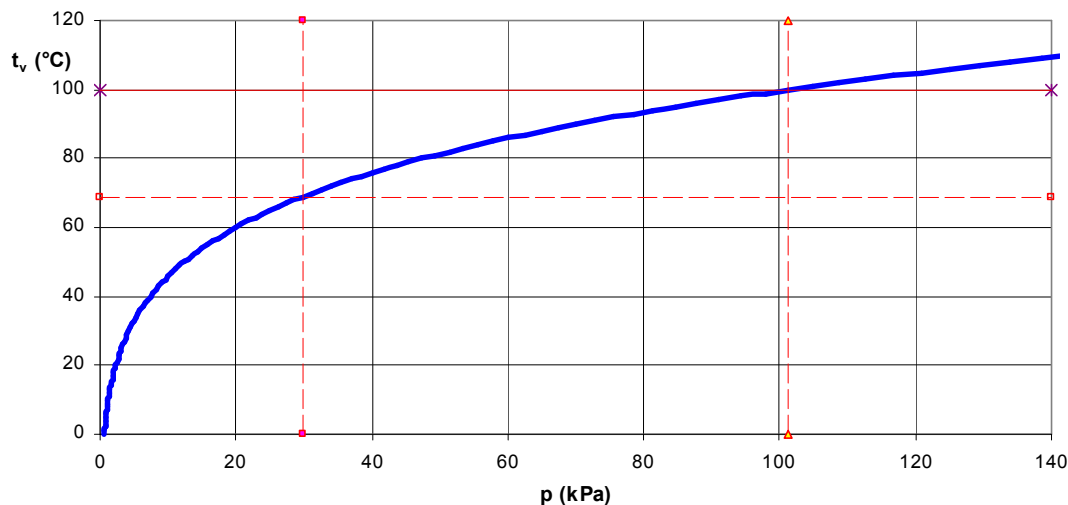


Obrázek 9.1 Schematické znázornění algoritmu výpočtu koncentrace zahušřovaného roztoku

Nelineární monotónně rostoucí závislosti změny teploty varu na hmotnostní koncentraci roztoku sacharózy⁵ $w = f_1 (t_{V,0}, dt_V)$ definovány pro sedm teplot varu čistého rozpouštědla jsou pro představu znázorněny na následujícím obrázku (tabulka viz. Příloha 2).

**Obrázek 9.2 Závislost zvýšení teploty varu na hmotnostní koncentraci roztoku sacharózy**

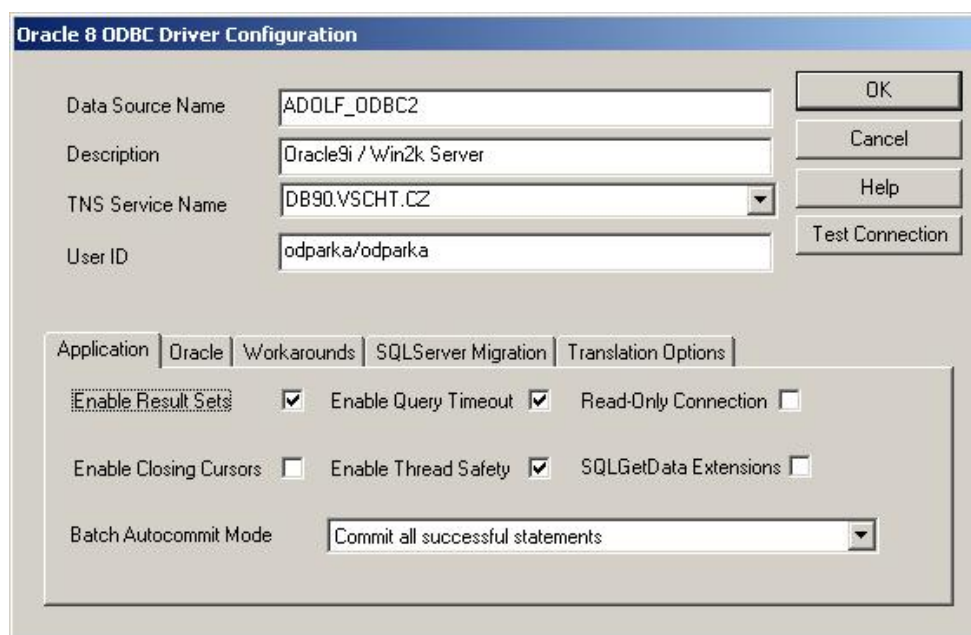
Druhá závislost teploty varu vody (jako rozpouštědla pro roztok sacharózy) na tlaku³⁴ $t_{V,0} = f_2 (p)$ je také nelineární (tabulka viz. Příloha 3).

**Obrázek 9.3 Závislost teploty varu vody na tlaku**

9.1.2 Praktické řešení

Bylo tedy nutné připravit datovou základnu a aplikační logiku na straně databáze Oracle, ale také operátorské obrazovky a komunikační skripty v aplikaci TomPack pro zápis a čtení informací uložených v databázi.

Nejdříve byla otestována ODBC komunikace programu TomPack se souborovou databází MS Access (soubory *.mdb) uloženou na lokálním disku počítače pomocí „neživých dat“ (náhodně generovaných programem). Byly tím ověřeny algoritmy pro čtení a zápis dat do souborové databáze (viz. kapitola 8.3.3). Při tomto pokusu mě překvapila zvýšená aktivita pevného disku, která byla pravděpodobně způsobena periodickým načítáním ODBC ovladačů a dalších informací nutných k přihlášení do databáze. Po nalezení vhodného ODBC ovladače pro komunikaci TomPacku s databázovým serverem Oracle se tento problém vyřešil. Přitom je zajímavé to, že pro danou konfiguraci (Tompack v1.10 a Oracle8i/9i) je jediným vhodným ovladačem **Oracle ODBC for Oracle ve verzi 8.1.6.6**, poněvadž jiné nativní ovladače Oracle (konkrétně v testovaných verzích 8.1.7.5, 9.0.1.1 a 9.0.1.2) ani ovladač třetí strany *Microsoft ODBC for Oracle (verze 2.573.6019)* nebyl schopen použít automaticky syntaxi „user/password“ uloženou do textového pole s názvem „User ID“, jak je vidět z následujícího obrázku.



Obrázek 9.4 Nastavení datového zdroje ODBC - pro databázový systém Oracle

Byl to velmi důležitý krok, protože jinak docházelo k neustálému zobrazování výzvy pro vložení hesla (autorizace) pro každý pokus otevření spojení s databází. Dále jsem vytvořil operátorské panely (obrazovky) v modulu TPConfig pro testování a ladění komunikace (viz. Příloha 8 a 9).

V rámci nové *Sekce fyzikálně-chemických dat* (Physical-chemical data section - viz. Obrázek 7.1) jsem pomocí skriptů v jazyce PL/SQL vytvořil celkem tři tabulky: TOC, Table1, Table2. (veškeré skripty na CD-ROM - viz. Příloha 1).

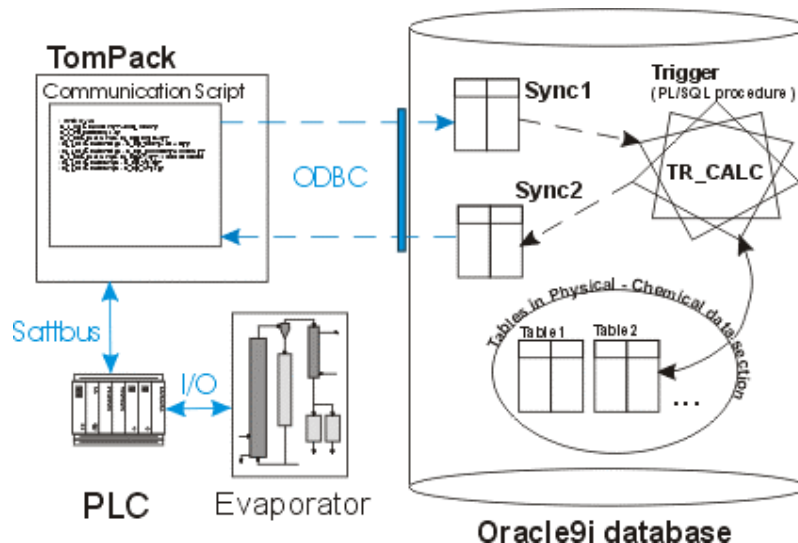
Tabulka **TOC** popisuje obsah všech ostatních tabulek v tomto tabulkovém prostoru. Tabulka **Table1** má čtyři sloupce označené *table1_id*, *x0*, *x1*, *x2*. Hodnoty sloupce označeného *table1_id* slouží pro identifikaci každého údaje v tabulce (indexace) a jsou generovány ze sekvence (uložená procedura generující s daným krokem číselnou posloupnost), *x0* je označení pro hmotnostní koncentraci (hm. %), *x1* pro teplotu varu čistého rozpouštědla (např. vody) ve (°C) a *x2* pro změnu teploty varu roztoku ve (°C ~ K). Do takto připravené tabulky jsem pomocí stejného skriptu (příkazy typu INSERT) vložil podle příložené „**Tabulky zvýšení bodu varu v závislosti na hmotnostní koncentraci roztoku sacharózy v roztoku**“ všechny známé trojice hodnot, doplněné o nulové hodnoty změny teploty varu pro čistá rozpouštědla. Pro představu je část tabulky uvedena na následujícím obrázku.

	Table1 Id	X0	X1	X2
1	1	5	60	0,02
2	2	5	70	0,02
3	3	5	80	0,02
4	4	5	90	0,03
5	5	5	100	0,03
6	6	5	110	0,03
7	7	5	120	0,03
8	8	10	60	0,08
9	9	10	70	0,08
10	10	10	80	0,09
11	11	10	90	0,09
12	12	10	100	0,10
13	13	10	110	0,11

Obrázek 9.5 Pohled na tabulku Table1 v nástroji Oracle Discoverer

Dále jsem vytvořil tabulku **Table2**, která má tři sloupce označené *table2_id*, *x0*, *x1*. Hodnoty sloupce označeného *table2_id* slouží pro identifikaci řádků, *x0* je označení pro teplotu varu vody ve (°C) a *x1* pro tlak systému v (kPa). Do takto připravené tabulky jsem vložil podle příložené „**Tabulky teploty varu vody na tlaku**“ všechny známe dvojice hodnot.

Do *Sekce dat pro řízení* (Control data section) uložil dvě synchronizační tabulky a spoušť (trigger). Přestože obě synchronizační tabulky **Sync1** a **Sync2** mají úplně stejnou strukturu slouží k jiným datovým transakcím. Do tabulky sync1 bude pouze zapisovat program TomPack. Každý zapisovaný řádek zahrnuje unikátní časovou značku výpočtu (sloupec *tstamp*), hodnoty pro výpočty (sloupce *x1*, *x2*) a sloupec *flag* informující o aktuálním stavu výpočtu. Tyto hodnoty ihned po vložení zpracuje databázový trigger **Tr_calc** „zavěšený“ nad synchronizační tabulkou sync1 a výsledky zapíše do tabulky sync2.



Obrázek 9.6 Schéma komunikace - datových transakcí pro on-line výpočty nad databází

Tělo triggeru **Tr_calc** obsahuje výkonné PL/SQL procedury pro výběr hodnot z tabulek ve *Fyzikálně-chemické datové sekci* a vlastní výpočty, reprezentující níže uvedené interpolační vzorce pro 2D a 3D problémy (lineární regrese).

Pro 2D interpolaci:

$$t_{V,0}^i = t_{V,0}^D + (t_{V,0}^H - t_{V,0}^D) \cdot \frac{p^{in} - p^D}{p^H - p^D} \quad (9.1)$$

Tabulka 9.2 Legenda pro vzorec 9.1

Proměnná	Jednotka	Význam
$t_{V,0}^i$	°C	interpolovaná teplota varu vody pro p^{in}
p^{in}	kPa	hodnota tlaku, pro kterou hledáme $t_{V,0}^i$
$t_{V,0}^H$	°C	nejbližší vyšší teplota varu roztoku, nalezená v tabulce, vůči $t_{V,0}^i$
$t_{V,0}^D$	°C	nejbližší nižší teplota varu roztoku, nalezená v tabulce, vůči $t_{V,0}^i$
p^H	kPa	nejbližší nižší hodnota tlaku, nalezená v tabulce, vůči t_v^{in}
p^D	kPa	nejbližší vyšší hodnota tlaku, nalezená v tabulce, vůči t_v^{in}

Pro 3D interpolaci:

$$w^j = (w^{i,a} - w^{i,b}) \cdot \frac{t_v^{in} - t_v^D}{t_v^H - t_v^D} + w^{i,b} \quad (9.2)$$

$$w^{i,a} = \left(w^{t_v^{in}=t_v^D, dt_v=dt_v^H} - w^{t_v^{in}=t_v^D, dt_v=dt_v^D} \right) \cdot \frac{dt_v^{in} - dt_v^D}{dt_v^H - dt_v^D} + w^{t_v^{in}=t_v^D, dt_v=dt_v^D} \quad (9.3)$$

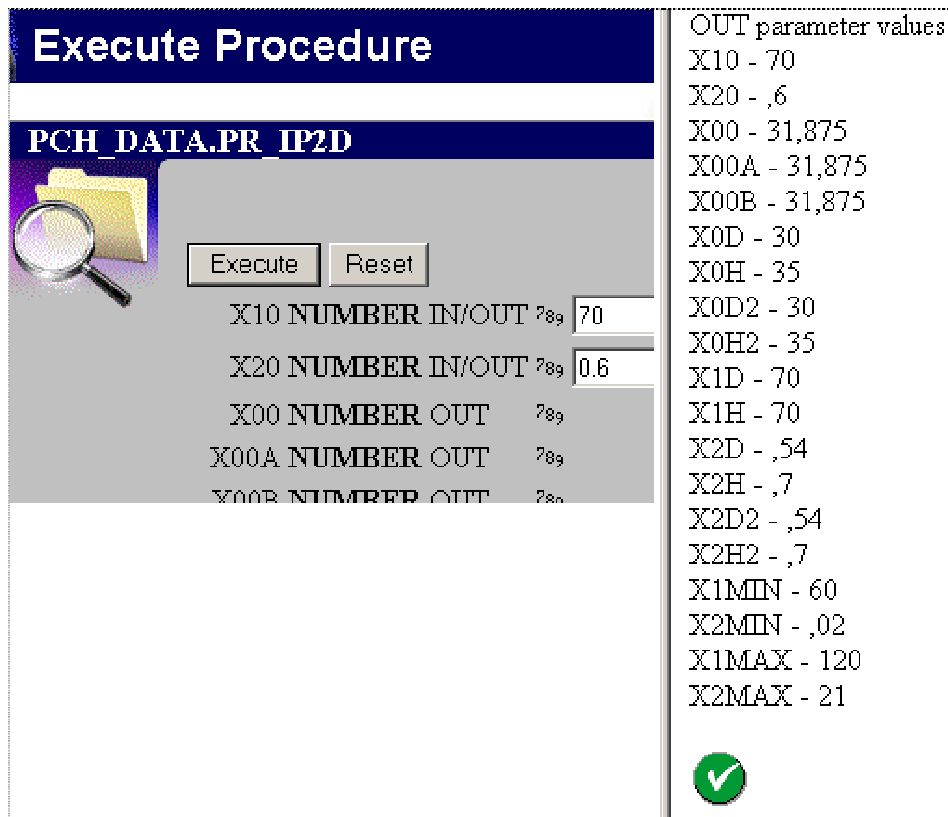
$$w^{i,b} = \left(w^{t_v=t_v^H, dt_v=dt_v^{H2}} - w^{t_v=t_v^H, dt_v=dt_v^{D2}} \right) \cdot \frac{dt_v^{in} - dt_v^{D2}}{dt_v^{H2} - dt_v^{D2}} + w^{t_v=t_v^H, dt_v=dt_v^{D2}} \quad (9.4)$$

$$dt_v = t_{v,w} - t_{v,0} \quad (9.5)$$

Tabulka 9.3 Legenda pro vzorce 9.2 až 9.5

Proměnná	Jednotka	Význam
t_v^{in}, dt_v^{in}	°C	hodnoty dt_v a t_v , pro které hledáme w^j
w^j	hm. %	interpolovaná koncentrace roztoku sacharózy pro t_v^{in}, dt_v^{in}
$w^{i,a}, w^{i,b}$	hm. %	hmotnostní koncentrace – mezivýsledky
dt_v	°C ~ K	změna teploty varu roztoku
$t_{v,w}$	°C	teplota varu roztoku sacharózy o hmotnostní koncentraci w
$t_{v,0}$	°C	teplota varu čistého rozpouštědla (voda)
t_v^D	°C	nejbližší nižší teplota varu roztoku, nalezená v tabulce, vůči t_v^{in}
t_v^H	°C	nejbližší vyšší teplota varu roztoku, nalezená v tabulce, vůči t_v^{in}
dt_v^D	°C ~ K	nejbližší nižší dt_v , nalezená v tabulce, vůči dt_v^{in}
dt_v^H	°C ~ K	nejbližší vyšší dt_v , nalezená v tabulce, vůči dt_v^{in}
$w^{t_v^{in}=t_v^D, dt_v=dt_v^H}$	hm. %	koncentrace roztoku sacharózy pro $t_v^{in}=t_v^D$ a $dt_v=dt_v^H$

Výsledky jsem ověřil v rámci **WebDB** (internetové GUI pro Oracle8i), kde jsem uloženou proceduru spustil a otestoval několika kombinacemi různých vstupních hodnot.



Obrázek 9.7 Ukázka spuštění procedury(3D) v nástroji WebDB pro Oracle 8i

9.1.3 Neočekávané problémy při experimentálním ověřování funkčnosti

Při prvním testu on-line odhadu koncentrace uskutečňovaném přímo na odparce bylo jako vstupního roztoku pro odpařování použito pitné vody z vodovodní sítě. Byla tedy očekávána nulová změna teploty varu dt_v mezi teplotou varu vody měřenou [čidlo TI03] a teplotou varu rozpouštědla vypočítanou z měřeného tlaku v systému [čidlo PIC12] pomocí závislosti $t_{v,0} = f_2(p)$. Získané hodnoty byly ale překvapivé:

Tabulka 9.4 Hodnoty měřených a vypočtených veličin při prvním testování odhadu koncentrace

t_v (°C)	p (kPa)	$t_{v,0}$ (°C)	dt_v (°C)	w (hm. %)
78,71	60,06	85,9	-7,2	-
78,52	59,77	85,8	-7,3	-
78,13	60,21	86,0	-7,9	-
83,01	82,91	94,4	-11,4	-
84,77	83,35	94,6	-9,8	-
86,52	91,26	97,0	-10,5	-

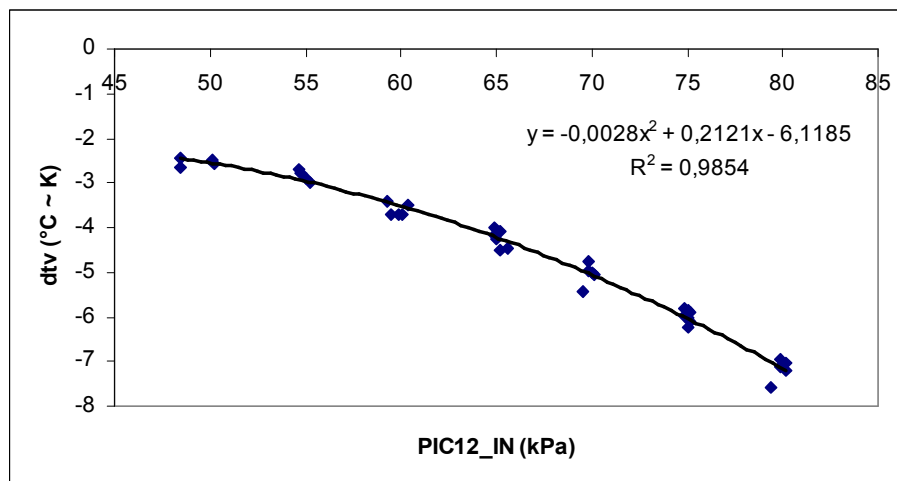
Jelikož vypočítaná teplota varu byla o několik stupňů vyšší než teplota měřená, dospěli jsme k jedné z možných příčin: teplota a tlak se měřila v místech příliš od sebe vzdálených. Proto byly experimenty přerušeny a tlakové čidlo přemístěno blíže vůči teplotnímu senzoru. Přestože při druhém testu došlo k relativnímu zlepšení diference mezi teplotami varu vůči předchozímu pokusu, byl rozdíl stále záporný.

Tabulka 9.5 Hodnoty měřených a vypočtených veličin při druhém testování odhadu koncentrace

t_v (°C)	p (kPa)	$t_{v,0}$ (°C)	dt_v (°C)	w (hm. %)
78,13	48,49	80,5	-2,4	-
82,03	59,47	85,7	-3,6	-
83,79	65,63	88,2	-4,4	-
83,6	65,19	88,0	-4,5	-
84,38	69,58	89,8	-5,4	-
85,74	75,00	91,7	-6,0	-
84,74	74,85	91,7	-5,9	-

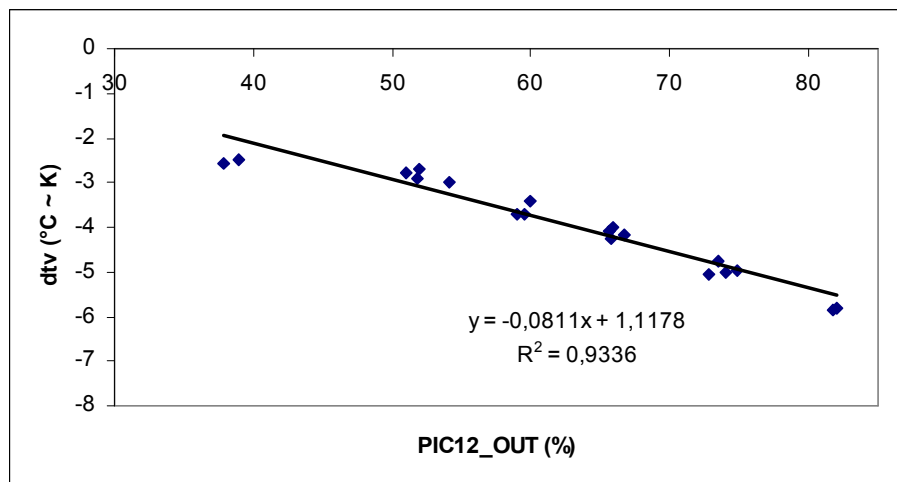
Z technických důvodů již nebylo možné přemístit čidlo tlaku do téhož místa jako teploty. Proto jsme začali hledat faktory ovlivňující tuto diferenci. Jedním z těchto faktorů by mohla být **dynamika proudění brýdových par potrubím** s odlišným vnitřním průměrem a s tím související tlaková ztráta. Změna teploty varu by tedy mohla být závislá na tlaku v brýdovém prostoru [hodnota $PIC12_IN$] anebo na otevření pneumatického regulačního ventilu [hodnota $PIC12_OUT$].

Ze série měření pro žádanou hodnotu tlaku páry v topné komoře odparky 25 (kPa) [hodnota $PIC11_IN$] byla zjištěna tato závislost $dt_v = f(PIC12_IN)$:



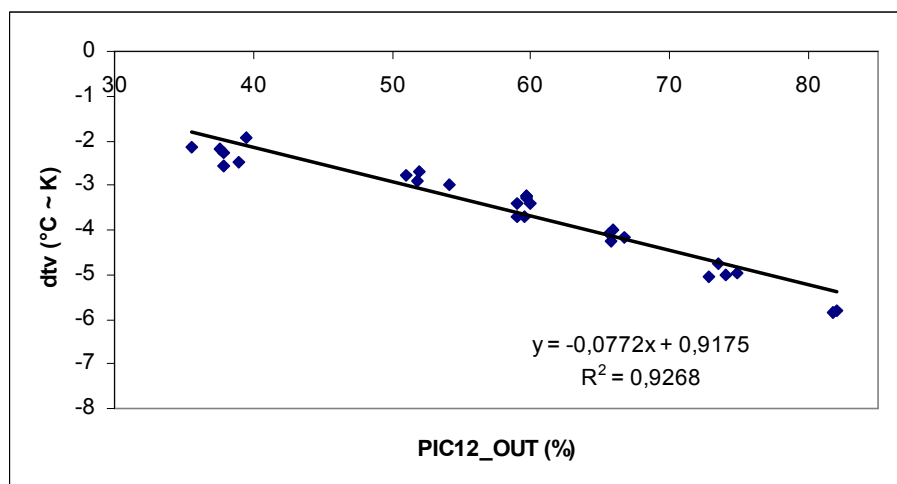
Obrázek 9.8 Závislost změny teploty varu na tlaku v brýdovém prostoru (pro tlak páry 25kPa)

Zjistili jsme však, že pro tlak páry v topné komoře pohybujícího se kolem žádané hodnoty 50kPa je tato závislost odlišná a že tedy $dt_v = f(PIC12_IN, PIC11_IN)$. V další sérii měření jsme sledovali vliv otevření pneumatického regulačního ventilu - závislost $dt_v = f(PIC12_OUT)$:



Obrázek 9.9 Závislost změny teploty varu na otevření regulačního ventilu (pro tlak páry 25kPa)

I pro vyšší tlak páry v topné komoře (*PIC11_IN*) regulovaný na 50kPa byly výsledky porovnatelné, a proto jsem je vynesl do stejného grafu:



Obrázek 9.10 Závislost změny teploty varu na otevření regulačního ventilu (pro tlak páry 25 a 50kPa)

Pomocí zjištěných funkčních závislostí byla měření opakována s kompenzací vlivu otevření regulačního ventilu na změnu teploty varu roztoku sacharózy. Očekávané hodnoty dt_v se dostaly do kladných čísel, a proto mohl být uskutečněn i odhad hmotnostní koncentrace cukerného roztoku.

Tabulka 9.6 Hodnoty vypočtených veličin při testování odhadu koncentrace s kompenzací

$t_{v,0}$ (°C)	dt_v (°C)	w (hm. %)
86,10	0,04	5,8
86,04	0,08	9,2
86,28	0,10	10,7
85,97	0,12	11,8
85,91	0,17	14,2
85,85	0,22	16,3
86,10	0,23	17,0

Zadaný úkol – vytvoření aplikace pro on-line odhad koncentrace pracující nad databází – byl tímto splněn a výsledky ověřeny podle tabulky (viz. Příloha 2), ale z časových důvodů již nemohly být vypočtené hodnoty koncentrace sacharózy potvrzeny analytickou metodou (např. refraktometricky). V souvislosti s měřeními a z nich vypočtenými hodnotami veličin je na místě vyjádřit se alespoň obecně k **chybám měření** - nebo spíše podle nové koncepce k **nejistotám v měření**^{48, 49, 50} (kvantitativními ukazateli kvality výsledků).

Předpokládané zdroje nejistot při tomto *nepřímém měření* jsou následující:

- chyby přístrojů (např. rozlišovací schopnost daná počtem bitů A/D převodníku)
- chyby plynoucí z neznámých nebo nekompenzovatelných vlivů prostředí
- nepřesnost referenčních materiálů (experimentální tabulky)
- náhodné chyby při měření
- chyby při matematickém zpracování dat (např. při interpolaci)

Ke stanovení velikosti nejistot existují dvě metody:

- statistické zpracování naměřených údajů (metoda A – nejistota typu A)
- jiné než statistické zpracování naměřených údajů (metoda B – nejistota typu B)

Zamyslíme-li se nad konkrétním výpočtem nejistoty tohoto nepřímého měření koncentrace, je zřejmé, že metodu B nebudeme moci použít z důvodu neekvidistantního rozložení hodnot v experimentálních tabulkách (viz. Příloha 2 a 3). Pro vyhodnocení standardních nejistot metodou typu A, vycházející ze statistického zpracování naměřených údajů, bude nutné provést rozsáhlejší měření.

10 Závěr

Na závěr bych chtěl říci, že tato práce měla výrazně technický charakter. Kromě návrhu a sestavení počítačových komponent musel být databázový a aplikační server kompletně vybaven i po programové stránce. Přes snahu aplikovat maximum poznatků z teorie, ale také i z praxe u firmy Oracle, byl tento úkol velmi náročný. Při testování komunikace programu TomPack s datovými zdroji se vyskytly problémy, které vyústily v konzultaci s autorem tohoto programu a podařilo se je vyřešit. Vytvořená databázová struktura, založená na relačním modelu dat, byla úspěšně otestována pro sběr procesních dat v reálném čase a zdá se být vyhovující. Byla také ověřena možnost on-line výpočtů neznámých veličin na základě veličin měřených a (tabelovaných) dat uložených v databázi.

Byl realizován a ověřen i zpětný tok informací, kdy vypočtené hodnoty neznámých veličin mohou sloužit pro on-line řízení laboratorní odpařovací stanice. Byla rovněž ověřena možnost komunikace programu Matlab s databází, takže je možné jakékoliv matematické zpracování dat. Data uložená v databázi mohou být také využita jako vstupy např. pro experimentální identifikaci tohoto systému (nalezení parametrů jeho statických a dynamických modelů).

Seznam obrázků v textu

Obrázek 1.1 Celkový podíl na trhu relačních databázových licencí v roce 2000 (zdroj IDC 2001)	8
Obrázek 1.2 Obecná architektura rozhraní ODBC ³⁰	10
Obrázek 1.3 Zobecněná architektura TIS pro řízení a sběr výrobních dat	15
Obrázek 1.4 Architektura produktu FactorySuite, Wonderware ⁹	16
Obrázek 1.5 Architektura IndustrialSQL Serveru – FactorySuite, Wonderware ⁹	17
Obrázek 1.6 Základní koncepce PI Systemu - OSI Software ²⁹	18
Obrázek 1.7 Architektura systému PI - OSI Software ²⁹	18
Obrázek 1.8 Rozdíl mezi ukládáním do relační a real-time databáze PI TM	19
Obrázek 1.9 Schéma přenosu v TIS s databází PI TM pro Štětí ¹⁶	19
Obrázek 1.10 Architektura klient/server systému iFIX Dynamic – Intellution ²⁴	20
Obrázek 1.11 Vnitřní architektura programu iFIX - Intellution.....	21
Obrázek 1.12 Architektura systému iHistorian – Intellution ²⁴	22
Obrázek 1.13 Komunikační architektura datového serveru Citect – CI Technologies ³¹	22
Obrázek 1.14 Architektura systému Plant2Business – CI Technologies ³¹	23
Obrázek 1.15 Architektura systému myAMC – Siemens ²⁶	24
Obrázek 1.16 Ideové schéma systému Vidium - MediumSoft	26
Obrázek 1.17 Popis funkce systému Vidium – MediumSoft ²³	27
Obrázek 1.18 Architektura OFC (OPC Factory Server) – Schneider-electric ²⁸	28
Obrázek 2.1 Architektura systému TomPack - TP Server ³	29
Obrázek 2.2 Nejjednodušší konfigurace systému TomPack ³ (Odparka).....	30
Obrázek 6.1 Vyobrazení řídicího panelu procesního počítače.....	36
Obrázek 6.2 Schéma stanice odparka s vyznačením okruhů měření a regulace (Kmínek ⁶)... 37	
Obrázek 6.3 Schéma komunikační struktury – síťová architektura	38
Obrázek 7.1 Nová struktura databáze pro sběr procesních dat	40
Obrázek 7.2 Schéma přenosu dat do databázové tabulky systému Oracle	41
Obrázek 8.1 Architektura komunikace mezi klientem a databází Oracle (SQL*Net)	45
Obrázek 8.2 Úvodní nabídka ODBC Administrátora.....	46
Obrázek 8.3 Architektura ODBC komunikace aplikace a databáze Oracle	47
Obrázek 8.5 Přihlášení k databázi Oracle (SQL*Plus).....	49
Obrázek 8.6 Textově orientovaný interpret příkazů Oracle SQL*Plus.....	49
Obrázek 8.7 Přihlášení k databázi Oracle v programu SQL-Programmer	50
Obrázek 8.8 Přihlášení do databáze Oracle v systému Discoverer	51
Obrázek 8.9 Vytvoření nového projektu v Discovereru	51
Obrázek 8.10 Část tabulky s reálnými daty (Oracle Discoverer).....	52
Obrázek 8.11 Časový průběh reálných dat z odparky (Oracle Discoverer)	52
Obrázek 8.12 Přístup k databázím z Database Toolbox (Matlab) ve Windows a Linux.....	53
Obrázek 9.1 Schematické znázornění algoritmu výpočtu koncentrace zahušťovaného roztoku	56
Obrázek 9.2 Závislost zvýšení teploty varu na hmotnostní koncentraci roztoku sacharózy ...	56
Obrázek 9.3 Závislost teploty varu vody na tlaku.....	56
Obrázek 9.4 Nastavení datového zdroje ODBC - pro databázový systém Oracle	57
Obrázek 9.5 Pohled na tabulku Table1 v nástroji Oracle Discoverer	58
Obrázek 9.6 Schéma komunikace - datových transakcí pro on-line výpočty nad databází....	59
Obrázek 9.7 Ukázka spuštění procedury(3D) v nástroji WebDB pro Oracle 8i.....	61
Obrázek 9.8 Závislost změny teploty varu na tlaku v brýdovém prostoru (pro tlak páry 25kPa)	62

Obrázek 9.9 Závislost změny teploty varu na otevření regulačního ventilu (pro tlak páry 25kPa)	63
Obrázek 9.10 Závislost změny teploty varu na otevření regulačního ventilu (pro tlak páry 25 a 50kPa).....	63

Seznam tabulek v textu

Tabulka 1.1 Databázové systémy – obecná struktura.....	6
Tabulka 1.2 Představitelé souborových a klient/server databázových systémů	7
Tabulka 1.3 Typičtí představitelé relačních databázových systémů	7
Tabulka 1.4 Typy databázových rozhraní.....	9
Tabulka 1.5 Přehled základních ODBC ovladačů ze sady MDAC od firmy Microsoft.....	12
Tabulka 1.6 Přehled používaných SQL databází s vazbou na systém TIRWeb – Coral s.r.o.	25
Tabulka 2.1 Podporované komunikační protokoly pro UniServer pro TomPack (v1.10) ³	30
Tabulka 3.1 Přehled produktů firmy Oracle a jejich využití.....	31
Tabulka 3.2 Svoboda a flexibilita při výběru provozní platformy a architektury pro systém Oracle	31
Tabulka 3.3 Základní datové typy systému Oracle	32
Tabulka 3.4 Komponenty Enterprise manager.....	32
Tabulka 4.1 Typy, využití a základní příkazy jazyka SQL.....	33
Tabulka 4.2 Doporučená literatury ke studiu jazyka SQL.....	33
Tabulka 5.1 Rozčlenění metod analýzy databází prostřednictvím nástrojů pro dolování dat .	34
Tabulka 5.2 Přehled nástrojů pro dolování dat	34
Tabulka 6.1 Technické parametry odparky.....	35
Tabulka 6.2 Základní informace o měřicí a řídicí soustavě.....	36
Tabulka 6.3 Typy I/O pro PLC OP-45 SB.....	36
Tabulka 6.4 Popis použitých analogových (A) nebo digitálních (D) čidel (Č) a akčních členů (Ak).....	37
Tabulka 7.1 Parametry aplikačního a databázových serverů.....	40
Tabulka 7.2 Podrobné informace o sloupcích synchronizační tabulky Sync0 (zkrácený výpis)	41
Tabulka 7.3 Podrobné informace o sloupcích synchronizačních tabulek Sync1 a Sync2.....	42
Tabulka 7.4 Podrobné informace o sloupcích tabulky Value	42
Tabulka 7.5 Podrobné informace o sloupcích tabulky Experiment	42
Tabulka 7.6 Podrobné informace o sloupcích tabulky Senzor.....	42
Tabulka 7.7 Podrobné informace o sloupcích tabulky Location.....	43
Tabulka 7.8 Podrobné informace o sloupcích tabulky Measurement	43
Tabulka 7.9 Podrobné informace o sloupcích tabulky Variable	43
Tabulka 7.10 Podrobné informace o sloupcích tabulky Table1 (Table2).....	43
Tabulka 7.11 Podrobné informace o sloupcích tabulky TOC.....	43
Tabulka 8.1 Programové vybavení pro správu DS a operace s daty	44
Tabulka 8.2 Konkrétní konfigurační řetězce (TNS aliasy) pro databázové servery Oracle na VŠCHT.....	45
Tabulka 8.3 Algoritmus načtení proměnných programem Matlab z databáze Oracle	53
Tabulka 8.4 Další užitečné příkazy z Database Toolbox, Matlab.....	53
Tabulka 8.5 Algoritmus čtení proměnných programem TomPack z tabulky databáze Oracle	54
Tabulka 8.6 Algoritmus zápisu proměnných programem TomPack do tabulky databáze Oracle	54

Tabulka 9.1 Fyzikálně-chemické závislosti, jejich tabulky a skripty.....	55
Tabulka 9.2 Legenda pro vzorec 9.1	59
Tabulka 9.3 Legenda pro vzorce 9.2 až 9.5	60
Tabulka 9.4 Hodnoty měřených a vypočtených veličin při prvním testování odhadu koncentrace.....	61
Tabulka 9.5 Hodnoty měřených a vypočtených veličin při druhém testování odhadu koncentrace.....	62
Tabulka 9.6 Hodnoty vypočtených veličin při testování odhadu koncentrace s kompenzací .	64

Význam použitých zkratk a pojmů

Zkratka	Význam anglické zkratky	Význam - použití
ActiveX	-	protokol pro integraci komponent třídy COM
Ad-hoc query	-	náhodný dotaz
ADO (ActiveX DO)	ActiveX Data Objects	knihovny VB pro přístup k datovým zdrojům
API	Application Programming Interface	programátorské prostředí (rozhraní)
C/C++ language	-	programovací jazyk C/C++
CASE	Computer Aided System Engineering	nástroje pro návrh a modelování struktury systémů
COM	Component Object Model	specifikace tvorby komponent využívaných v různých progr. jazycích
CPU	Central Processor Unit	centrální procesorová jednotka
DAO	Data Access Objects	knihovny VB pro přístup k datům v souborech MDB (MS Access)
DB	DataBase	databáze
DBMS	DataBase Management System	systém řízení báze dat
DCL	Data Control Language	typ SQL dotazu – řízení a správa DBMS
DCS	Distributed Control System	distribuovaný řídicí systém
DDE (NetDDE)	Dynamic Data Exchange	dynamická výměna dat (po síti)
DDL	Data Definition Language	typ SQL dotazu – definice dat (dat. struktur)
DML	Data Manipulation Language	typ SQL dotazu pro manipulace s daty
DS	Data Source	datový zdroj
DSC	Digital Supervisory Control	číslicové dohlížecí řízení
DSN alias	Data Source Name	pojmenování datového zdroje
EIS	Executive Information Systems	manažerské informační systémy (OLAP)
ERP	Enterprise Resource Planning	plánování zdrojů
Extended SQL	-	rozšířený SQL
GUI	Graphical User Interface	grafické uživatelské prostředí (rozhraní)
HDD	Hard Disk	pevný disk
HTML	HyperText Markup Language	jazyk pro tvorbu WWW dokumentů
HW	Hardware	počítačové komponenty (hmotné části)
IEC	International Electro-technical Commission	IEC 1131-3 : norma pro programovací jazyky / programování
JAVA language	-	programovací jazyk JAVA
JSP	JAVA Server Pages	dynamické WWW dokumenty psané v Javě
MDAC	Microsoft Data Access Components	balíček ODBC ovladačů pro Windows
MMI	Manufacturing and Management Information (system)	informační systémy pro výrobu a řízení
ODBC driver	Open Database Connectivity	ovladač pro komunikaci s datovými zdroji
OLAP	On-Line Analytical Processing	analýza dat v multidimenzionální kostce
OLE DB	Object Linking and Embedding for DB	objektové rozhraní pro přístup k datům
OPC	OLE for Process Control	OLE optimalizované pro řízení procesů
Oracle Cluster	-	blok spolupracujících serverů
Oracle EE	Oracle Enterprise Edition	rozšířená verze databáze Oracle
Oracle IFS	Internet File System	internetový souborový systém
OS	Operating System	operační systém (nehmotná část)
PC	Personal Computer	osobní počítač
PDA	Personal/Pocket Digital Assistant	osobní kapesní počítač
PIMS	Plant Information Management System	podnikový informační systém
PL/SQL	Procedural Language / Structured Query Language	strukturovaný dotazovací jazyk pro procedury
plug-in	-	programová nadstavba
procedura, script	-	programová sekvence příkazů

Zkratka	Význam anglické zkratky	Význam - použití
RAM	Random Access Memory	operační paměť (čtení-zápis)
RDBMS	Relational DataBase Management System	relační databázový systém
real-time	-	pracující v reálném čase
redundance	-	nadbytečnost (účelná)
RTDBS	Real-Time Database System	databázový systém pracující v reálném čase
RTUs	Remote Terminal Units	vzdálené terminály
SCADA/HMI	Supervisory Control and Data Acquisition / Human Machine Interface	kategorie softwaru pro sběr provozních dat a řízení
SEQUEL	Structured English Query Language	předchůdce jazyka SQL
SQL	Structured Query Language	strukturovaný dotazovací jazyk
SQL MDX	SQL Multidimensional Extension	rozšíření jazyka SQL o konstrukty pro práci s multidimenzionálními daty
SSL	Secure Socket Layer	šifrování paketů, pro zvýšení bezpečnosti
TIS	-	technologicko-informační systém
TNS alias	Transparent Network Substrate	pojmenování datového zdroje Oracle
trigger	-	databázový objekt (spouštěč)
upgrade	-	inovace, aktualizace
VB	Visual Basic	objektově orientovaný programovací jazyk
WWW	World Wide Web	celosvětová internetová síť
XML	eXtended Markup Language	rozšířený jazyk HTML

Literatura

1. NIKL J.: Wonderware IndustrialSQL Server 7.1 a příklady aplikací v ČR. *Automa*, 1998, číslo 3-4, str. 49-55.
2. ČERVENKA Z.: Wonderware FactorySuite 2000 – v. 7.1 a Wonderware InTouch 7.1. *Automatizace*, ročník 43, číslo 9, září 2000, str. 256-260.
3. *Manuály k programu TomPack*. (interní materiál firmy ProjectSoft, Hradec Králové), 2000-2001.
4. HALAŠKA I.: *Databázové systémy – Jazyk SQL a systém ORACLE*. ČVUT Skriptum, 1995.
5. BUBNÍK Z., aj.: *Sugar Technologists Manual Chemical and Physical Data for Sugar Manufacturers and Users*. 1995, Bartnes /8th edition/ s.160, 170.
6. KMÍNEK M., BURIAN P., HRNČIŘÍK P., JAKEŠ B., KOKOŠKA Z., POLNICKÝ J.: Laboratorní systém pro výzkum a výuku řízení a IT. In *Proc. of IEPC 2001*.
7. Internet: www.microsoft.com, msdn.microsoft.com, 2001. (firemní stránky Microsoft)
8. HANTA V.: *Procesní data v reálném čase*. 48. CHISA, říjen 2001, Smí – Šumava, CD.
9. Internet: www.wonderare.com, 2001. (firemní stránky Wonderware Corp.)
10. Internet: www.oracle.com, 2001. (firemní stránky Oracle)
11. MCCULLOUGH-DIETER C.: *Mistrovství v Oracle8i*. Computer Press, Praha, 1999.
12. TAUFER I.: Jak psát habilitační, disertační a diplomové práce. In *4th International Scientific-Technical Conference Process Control 2000*, Říp 2000, Kouty nad Děsou.
13. BURIAN P.: *Informační systémy podniku (i-Development*e-Management*i-Process)*. VŠCHT Skriptum, 2000.
14. Internet: www.projectsoft.cz, 2001. (firemní stránky ProjectSoft s.r.o.)
15. KRÓL V.: Základní principy databází reálného času. *Automa*, 2001, číslo 12, str. 36-39.
16. STIANKO M.: Technologicko informační systém pro papírny ve Štětí. *Automatizace*, ročník 44 (2001), číslo 2, str. 188-190.
17. NIKL J.: Wonderware IndustrialSQL Server 7.0. *Automatizace*, ročník 41 (1998), číslo 2, str. 79-84.
18. NIKL J.: Wonderware IndustrialSQL Server 7.1 a příklady aplikací v ČR. *Automatizace*, ročník 43 (2000), číslo 9, str. 630-634.
19. ŠTEFANÍK J., *Využití datových skladů ve vazbě na technologické procesy*. Medium Soft a.s., Divize IZS a speciálních systémů, Ostrava. str. 283-290.
20. Internet: www.coral.cz, 2002. (firemní stránky Coral s.r.o.)
21. Internet: www.amit.cz, 2002. (firemní stránky Amit s.r.o.)
22. Internet: www.geovap.cz, www.reliance.cz 2002. (firemní stránky GeoVap s.r.o.)
23. Internet: www.mediumsoft.cz, 2002. (firemní stránky MediumSoft s.r.o.)
24. Internet: www.intellution.com, 2002. (firemní stránky Intellution)
25. Internet: www.microsys.cz, 2002. (firemní stránky Microsys s.r.o.)
26. Internet: www.siemens.de, 2002. (firemní stránky Siemens)
27. ŘÍHA P.: Rozšíření systému e-Manufacturing o platformu iHistorian. *Automatizace*, ročník 45 (2002), číslo 2, str. 108-110
28. Internet: www.schneider-electric.cz, 2002. (firemní stránky Schneider-electric)

29. Internet: www.osisoft.com, 2002. (firemní stránky OsiSoft)
30. BERAN M.: Slova ODBC a JDBC nejsou mumláním zaklánače. *Computerworld*, číslo 43, 1999.
31. Internet: www.citect.com, www.ct.com.au 2002. (firemní stránky Citect)
32. LÍN T.: Plant2Business – cesta k monitoringu. *Automatizace*, ročník 44 (2001), číslo 4, str. 232-234.
33. Internet: www.ni.com, 2002. (firemní stránky National Instruments)
34. HOLEČEK O.: *Chemicko-inženýrské tabulky*. Skriptum VŠCHT, 1997. str. II-1 až II-3 tab. II-A.
35. Internet: www.ipesoft.sk, 2002. (firemní stránky IpeSoft s.r.o.)
36. HANTA V.: *Tvorba dotazů na databáze pomocí SQL příkazů*. Seminář ÚPŘT VŠCHT ze dne 7.12.2001.
37. BURIAN P.: Induktivní získávání znalostí pro řízení a diagnostiku chemicko-technologických procesů. *Automatizace*, ročník 44 (2001), číslo 6, str. 369-373.
38. Internet: www.adastra.cz, 2002. (firemní stránky Adasrta s.r.o.)
39. Internet: <http://atrey.karlin.mff.cuni.cz/~doug/study>, 2002. (autor: MACHALÍK T.)
40. Internet: www.mathworks.com, 2002. (firemní stránky MathWorks)
41. Internet: www.bmc.com, 2002. (firemní stránky BMC Software Inc.)
42. BEHÚŇ M.: Intellution – od systému SCADA k objektově orientované automatizaci. *Automatizace*, ročník 42 (1999), číslo 3, str. 166-168.
43. MALINA Z.: Promotic 2000 – Moderní a výkonné řešení pro Vaše aplikace. *Automatizace*, ročník 44 (2001), číslo 10, str. 643-644.
44. ČERVINKA M.: Prostředky Modicon pro vizualizaci a řízení procesů. *Automatizace*, ročník 44 (2001), číslo 10, str. 643-644.
45. Internet: www.pantek.cz, 2001. (firemní stránky Pantek s.r.o.)
46. Internet: www.teconsro.cz, 2002. (firemní stránky Tecon s.r.o.)
47. Internet: www.unis.cz, 2002. (firemní stránky Unis s.r.o.)
48. VDOLEČEK F.: Nejistoty v měření I: vyjadřování nejistot. *Automa*, 2001, číslo 7-8, str. 50-54.
49. VDOLEČEK F.: Nejistoty v měření II: nejistoty přímých měření. *Automa*, 2001, číslo 10, str. 52-56.
50. VDOLEČEK F.: Nejistoty v měření III: nejistoty nepřímých měření. *Automa*, 2001, číslo 12, str. 28-33.

Přílohy

❖ Příloha 1: Obsah vloženého CD-ROMu

Adresářový strom	Popis obsahu jednotlivých adresářů
<pre> . ├── Literatura │ ├── Web │ ├── Scan │ └── Jine ├── TomPack │ ├── Doc │ └── Skripty ├── Oracle │ ├── Doc │ ├── Skripty │ └── Nastaveni ├── SQL ├── Diplomova_prace └── Shareware </pre>	<p>kořenový adresář CD (root)</p> <p>použitá literatura z Internetu skenované materiály neražazená literatura</p> <p>dokumenty k programu TomPack skripty k programu TomPack</p> <p>dokumenty k programům Oracle skripty k programům Oracle</p> <p>informace o jazyku SQL vlastní dokument volně šiřitelné programy</p>

Použití:

Tento disk obsahuje dokumenty a další typy souborů, které lze otevřít v následujících doporučených programech:

*.doc	dokument MS Word (ve verzích 2000, XP a vyšší)
*.txt	textový dokument (Notepad, Write)
*.pdf	dokument Adobe Acrobat Reader (verze 4.05 a vyšší)
*.htm, *.mht	HTML dokumenty Internet Explorer (verze 5 a vyšší)
.sql	skripty v jazyce PL/SQL (Oracle SQL Plus, Oracle JDeveloper)
*.png, *.bmp	obrázky (libovolný prohlížeč podporující tyto formáty)

Byl vytvořen jeden exemplář datového disku (CD-ROM), který bude k dispozici u vedoucího této diplomové práce Doc. M. Kmínka.

Kontaktní informace:

Adresa pracoviště: VŠCHT Praha, Technická 5, 166 28, Praha 6. (FCHI – ÚPŘT)
Telefon: +420 2 24354296
Email: Milos.Kminek@vscht.cz

Příloha 2: Tabulka zvýšení bodu varu v závislosti na hmotnostní koncentraci roztoku sacharózy v roztoku ⁵

w (hm. %)	t _{v,0} (°C)						
	60	70	80	90	100	110	120
5	0.02	0.02	0.02	0.03	0.03	0.03	0.03
10	0.08	0.08	0.09	0.09	0.10	0.11	0.12
15	0.16	0.17	0.18	0.19	0.21	0.22	0.24
20	0.25	0.27	0.29	0.31	0.33	0.36	0.38
25	0.37	0.39	0.42	0.45	0.48	0.52	0.55
30	0.50	0.54	0.57	0.62	0.66	0.71	0.75
35	0.66	0.70	0.75	0.81	0.87	0.93	0.99
40	0.85	0.91	0.97	1.04	1.12	1.19	1.28
45	1.08	1.16	1.25	1.34	1.43	1.53	1.63
50	1.39	1.49	1.60	1.71	1.83	1.96	2.09
55	1.78	1.91	2.05	2.19	2.35	2.51	2.68
60	2.34	2.52	2.70	2.89	3.09	3.31	3.54
61	2.43	2.61	2.79	2.99	3.20	3.43	3.66
62	2.52	2.71	2.90	3.11	3.33	3.56	3.80
63	2.62	2.82	3.02	3.24	3.47	3.71	3.96
64	2.74	2.95	3.16	3.39	3.62	3.88	4.14
65	2.87	3.09	3.31	3.55	3.80	4.06	4.34
66	3.02	3.24	3.48	3.73	3.99	4.27	4.56
67	3.18	3.42	3.67	3.93	4.20	4.50	4.80
68	3.36	3.61	3.87	4.15	4.44	4.75	5.07
69	3.56	3.82	4.10	4.39	4.70	5.02	5.37
70	3.77	4.05	4.34	4.65	4.98	5.33	5.69
71	4.00	4.30	4.61	4.94	5.29	5.66	6.04
72	4.26	4.57	4.91	5.26	5.63	6.02	6.43
73	4.54	4.87	5.23	5.60	5.99	6.41	6.85
74	4.84	5.19	5.57	5.97	6.39	6.83	7.30
75	5.16	5.54	5.95	6.37	6.82	7.29	7.79
76	5.51	5.92	6.35	6.81	7.28	7.79	8.32
77	5.89	6.33	6.79	7.27	7.78	8.32	8.89
78	6.30	6.76	7.26	7.77	8.32	8.90	9.51
79		7.23	7.76	8.31	8.90	9.52	10.20
80		7.74	8.30	8.89	9.52	10.20	10.90
81			8.87	9.51	10.20	10.90	11.60
82			9.49	10.20	10.90	11.60	12.50
83				10.90	11.60	12.40	13.30
84				11.60	12.40	13.30	14.20
85					13.30	14.20	15.20
86					14.20	15.20	16.20
87						16.20	17.30
88						17.30	18.50
89							19.70
90							21.00

❖ Příloha 3: Tabulka závislosti teploty varu vody na tlaku ³⁴

p (kPa)	t (°C)	p (kPa)	t (°C)	p (kPa)	t (°C)
0.61173	0.01	6.27950	37	36.97800	74
0.65716	1	6.62980	38	38.56300	75
0.70605	2	6.99690	39	40.20500	76
0.75813	3	7.38140	40	41.90500	77
0.81359	4	7.78400	41	43.66500	78
0.87260	5	8.20540	42	45.48700	79
0.93537	6	8.64640	43	47.37300	80
1.00209	7	9.10760	44	49.32400	81
1.07297	8	9.58980	45	51.34200	82
1.14825	9	10.09380	46	53.42800	83
1.22810	10	10.62050	47	55.58500	84
1.31290	11	11.17060	48	57.81500	85
1.40270	12	11.74490	49	60.11900	86
1.49790	13	12.34400	50	62.49900	87
1.59880	14	12.97000	51	64.95800	88
1.70560	15	13.62300	52	67.49600	89
1.81850	16	14.30300	53	70.11700	90
1.93800	17	15.01200	54	72.82300	91
2.06440	18	15.75200	55	75.61400	92
2.19790	19	16.52200	56	78.49500	93
2.33880	20	17.32400	57	81.46500	94
2.48770	21	18.15900	58	84.52900	95
2.64470	22	19.02800	59	87.68800	96
2.81040	23	19.93200	60	90.94500	97
2.98500	24	20.87300	61	94.30100	98
3.16910	25	21.82100	62	97.95900	99
3.36290	26	22.86800	63	101.32000	100
3.56700	27	23.92500	64	104.99000	101
3.78180	28	25.02200	65	108.77000	102
4.00780	29	26.16300	66	112.66000	103
4.24550	30	27.34700	67	116.67000	104
4.49530	31	28.57600	68	120.79000	105
4.57780	32	29.85200	69	125.03000	106
5.03350	33	31.17600	70	129.39000	107
5.32290	34	32.54900	71	133.88000	108
5.62670	35	33.97200	72	138.50000	109
5.94540	36	35.44800	73	143.24000	110

❖ Příloha 4: Možné údaje pro archivaci s označením datového typu

Označení obvodu*	IN	OUT	W	ON	R0	TI	TD	KLPR
FI23	real	x	x	x	x	x	x	real
FI24	real	x	x	x	x	x	x	x
FIC21	real	real	real		real	real	real	x
LI31	real	x	x	x	x	x	x	x
LI33	real	x	x	x	x	x	x	x
LI34	real	x	x	x	x	x	x	x
PIC11	real	real	real	bool	real	real	real	x
PIC12	real	real	real	bool	real	real	real	x
PI13	real	x	x	x	x	x	x	x
QI41	real	x	x	x	x	x	x	x
TI01	real	x	x	x	x	x	x	x
TI02	real	x	x	x	x	x	x	x
TI03	real	x	x	x	x	x	x	x
TIC04	real	real	real	bool	real	real	real	x
TI05	real	x	x	x	x	x	x	x
TI06	real	x	x	x	x	x	x	x
TI07	real	x	x	x	x	x	x	x
TIC08	real	real	real	bool	real	real	real	x

Legenda pro názvy sloupců předchozí tabulky:

IN	veličina regulovaná (měřená veličina)
OUT	akční veličina (výstupní veličina regulátoru)
W	žadaná hodnota
ON	stav regulátoru
R0	zesílení regulátoru
TI	integrační konstanta regulátoru
TD	derivační konstanta regulátoru
KLPR	klouzavý průměr

Pozn. *: Písmenný kód s identifikačním číslem pro označení měřicích a řídicích obvodů (podle normy ČSN ISO 3511-1)

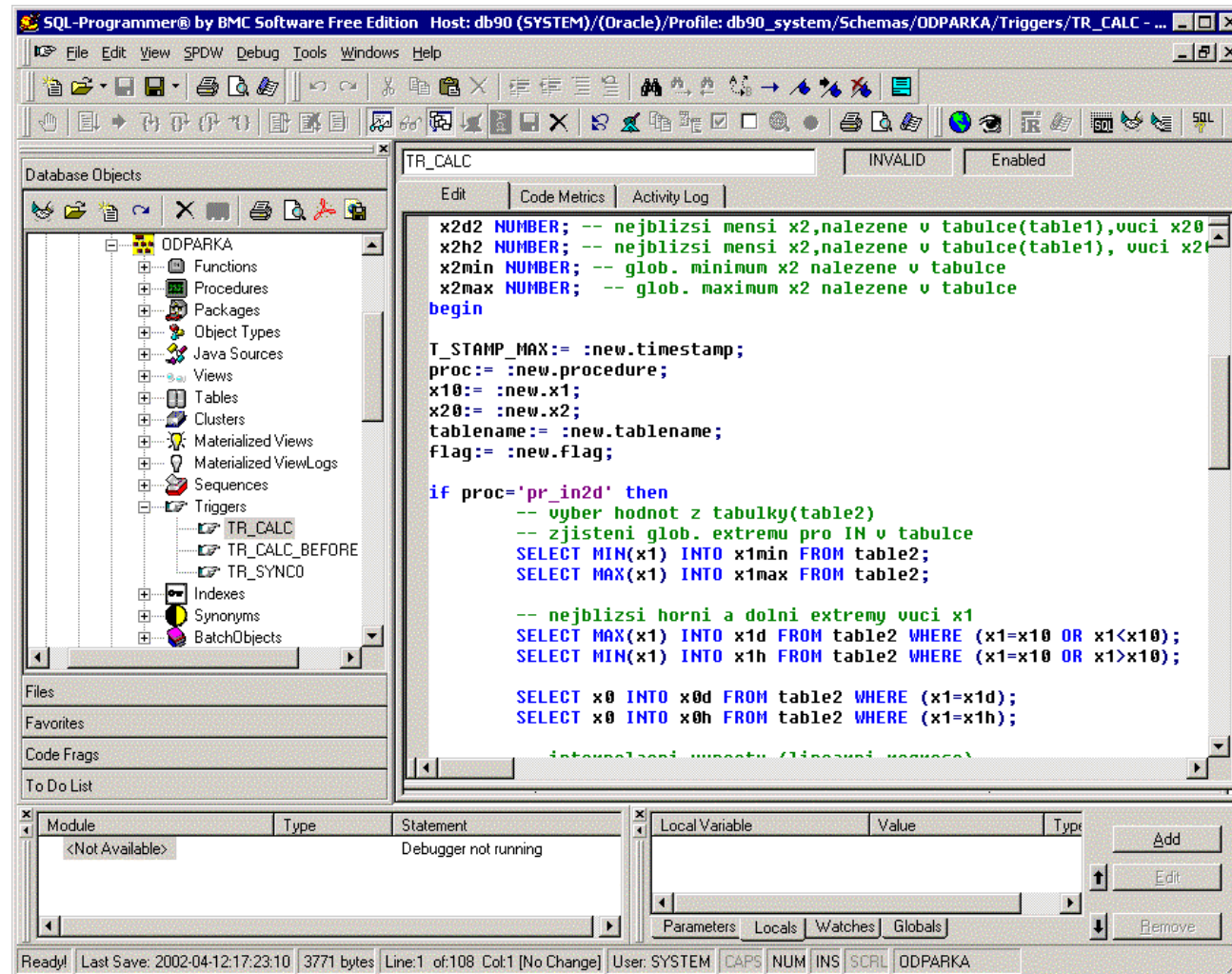
❖ Příloha 5: Tabulka porovnání známých řešení z průmyslu a programu TomPack

Firma	Produkt	SW typu SCADA/HMI	RT Databázový systém		Otevřenost systému pro výměnu dat
			procesní data	historická data	
Wonderware	FactorySuite 2000	InTouch7.0	InSQL (RT procesní db - MS SQL)		InSQL: ODBC/SQL (MS SQL, Oracle)
OSI Software	-	-	PI™ Database /DataStorage		PI Interface: ODBC/SQL + export do Excelu
Intellution	Dynamic	iFIX SCADA	iFIX SCADA Server (RT procesní db)	iHistorian Server (Data Collectors)	iHistorian: ODBC/SQL: MS SQL
CI Technologies	Citect (Data Server)	Plant2Business	Plant2Business Server (RT procesní db)	Plant2SQL (MS SQL7, MSDE)	Plant2SQL: SQL/ODBC: MS SQL
National Instruments { Tecon s.r.o., CZ }	Lookout™	Lookout™ Client/Web	Lookout Server local db (ASCII – CVS)	LookOut Citadel	Lookout Citadel: SQL/ODBC
Siemens	myAMC®	Simantic	myAMC - repository SAN/NAS		ODBC/SQL: MS SQL, Oracle, Informix
Coral s.r.o., CZ	TIRS	TIRS32 / Web	TIRS Server registrační local db (dBase, FoxPro)	-	ODBC/SQL: MS SQL, Oracle
Amit s.r.o., CZ	DB-Net	DB-Net	lokální archivy	-	ODBC/SQL
GeoVap s.r.o., CZ	Reliance	Reliance Design/Runtime	Reliance Server local db (Paradox, dBase)	-	BDE, ODBC/SQL: MS SQL
Medium Soft a.s., CZ	Vidium	-	-	IBM DB2 V5.0	ODBC/SQL
Microsys, CZ	Promotic 2000	Promotic 2000	PmDatabase (local db?)	-	ODBC/SQL: dBase, MS Access, FoxPro, Paradox, Informix, Oracle + ADO + ActiveX
IpeSoft, SK	D2000 Enterprise Production Systems	D2000 Actis	D2000 Actis (DODM + db)	D2000 Industrial SQL Archiv	ODBC/SQL, OLE + export Excel, Crystal Reports
Schneider-electric	OFS (OPC Factory Server)	-	interní MBD (Access), ASCII soubory	-	DCOM
ProjectSoft, CZ	TomPack	TomPack Config/Viewer	TomPack Server (local .jou file ~ MS Access MDB)	-	skripty TomPack s funkcemi ODBC/SQL: Oracle, MS SQL

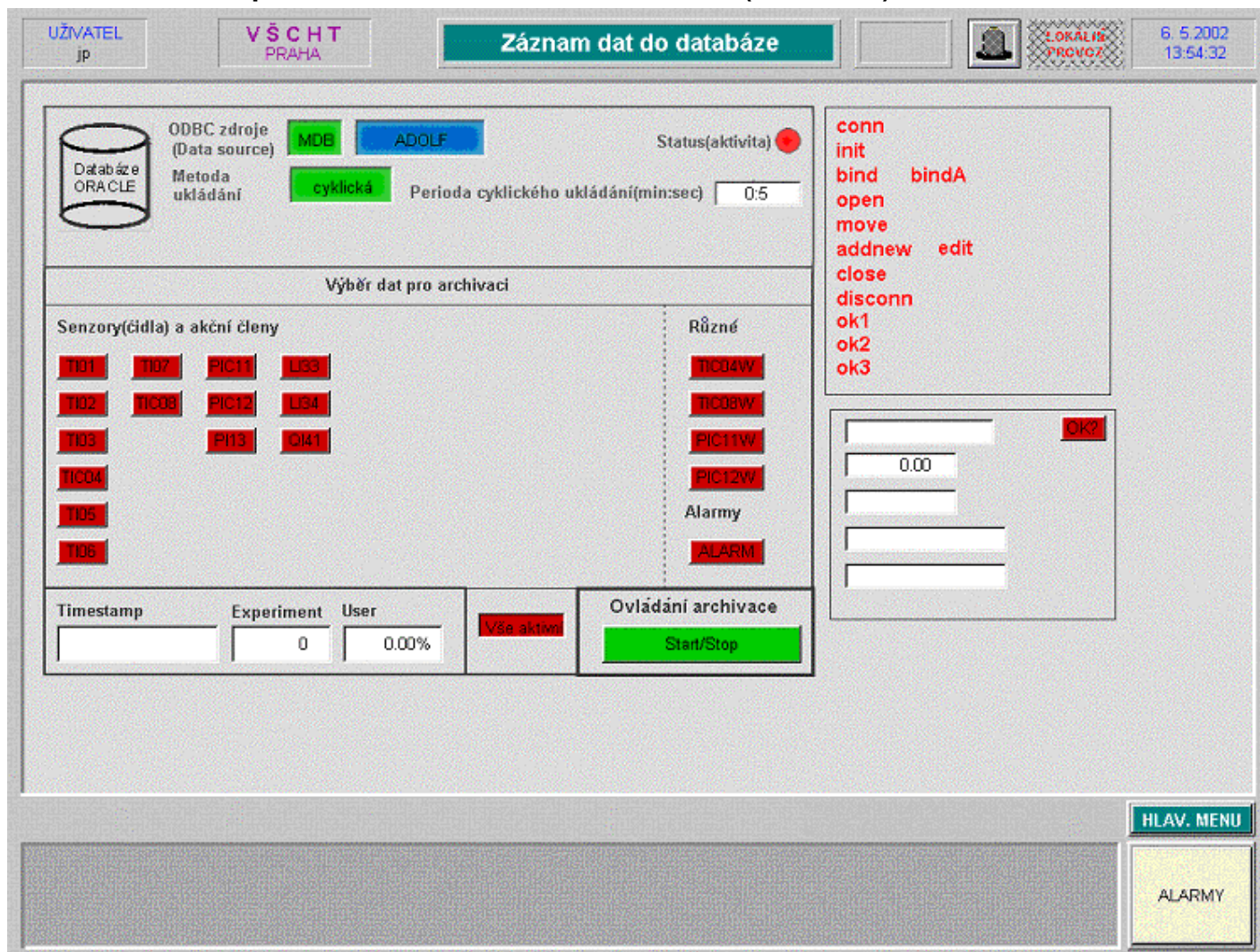
❖ Příloha 6: Tabulka ODBC funkcí programu TomPack ³

<i>Funkce</i>	<i>TPS</i>	<i>TPV</i>	<i>Typ fce</i>	<i>Typ parametrů</i>					<i>Popis</i>
DBConnect	x	x	bool	integer&	string				Otevře spojení na ODBC zdroj. První parametr je <i>integer</i> proměnná, do které se uloží handle na spojení. Druhý parametr je přípojovací řetězec (DSN jméno).
DBOpenTable	x	x	bool	integer&	int	string	string	string	Otevře tabulku. První parametr je <i>integer</i> proměnná, do které se uloží handle na tabulku. Druhý je handle spojení (z DBConnect), třetí je jméno tabulky (musí být v hranatých závorkách). Čtvrtý parametr je filtr a pátý řádicí podmínka (pro ODBC WHERE a ORDER BY).
DBBindBool	x	x	bool	integer	bool&	string			Připojí <i>bool</i> proměnnou pro přenos hodnot. Druhý parametr je jméno sloupce, opět v hranatých závorkách.
DBBindInt	x	x	bool	integer	integer&	string			Připojí <i>integer</i> proměnnou pro přenos hodnot. Druhý parametr je jméno sloupce, opět v hranatých závorkách.
DBBindReal	x	x	bool	integer	real&	string			Připojí <i>real</i> proměnnou pro přenos hodnot. Druhý parametr je jméno sloupce, opět v hranatých závorkách.
DBBindString	x	x	bool	integer	string&	string			Připojí <i>string</i> proměnnou pro přenos hodnot. Druhý parametr je jméno sloupce, opět v hranatých závorkách.
DBMoveFirst	x	x	bool	integer					Přesune ukazatel v tabulce na první záznam a aktualizuje proměnné.
DBMoveLast	x	x	bool	integer					Přesune ukazatel v tabulce na poslední záznam a aktualizuje proměnné.
DBMoveNext	x	x	bool	integer					Přesune ukazatel v tabulce na další záznam a aktualizuje proměnné.
DBMovePrev	x	x	bool	integer					Přesune ukazatel v tabulce na předchozí záznam a aktualizuje proměnné.
DBIsBOF	x	x	bool	integer	bool&				Do proměnné ve druhém parametru vloží <i>true</i> , je-li ukazatel za poslední položkou. Vrací-li funkce <i>false</i> , je hodnota proměnné nedefinována.
DBIsEOF	x	x	bool	integer	bool&				do proměnné ve druhém parametru vloží <i>true</i> , je-li ukazatel před první položkou. Vrací-li funkce <i>false</i> , je hodnota proměnné nedefinována.
DBCcloseTable	x	x	bool	integer					Uzavře tabulku
DBDisconnect	x	x	bool	integer					Uzavře spojení.
DBEdit	x	x	bool	integer					Přepíše aktuální řádek proměnnými
DBAddNew	x	x	bool	integer					Přidá nový řádek s hodnotami proměnných

❖ Příloha 7: Vývojové prostředí programu SQL-Programmer



❖ Příloha 8: Obrazovka operátora – záznam dat do databáze (TomPack)



❖ Příloha 9: Obrazovka operátora – komunikace s databází (TomPack)

