



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

SBĚR MĚŘENÍ A OVLÁDÁNÍ

COLLECTION OF MEASUREMENT AND CONTROL DATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

David Želinský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2020



Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: David Želinský

ID: 203429

Ročník: 3

Akademický rok: 2019/20

NÁZEV TÉMATU:

Sběr MaR dat

POKYNY PRO VYPRACOVÁNÍ:

1. Proveďte literární rešerši o sběru dat z inteligentních budov.
2. Na základě databázového modelu vytvořte databázi a rozhraní v softwaru COACH AX.
3. Podle technologických schémat a datových bodů realizujte simulační model vybrané části technologie budovy.
4. Realizujte příslušnou vizualizaci MaR dat.
5. Ověřte funkčnost vlastního řešení.

DOPORUČENÁ LITERATURA:

VALEŠ, M., Inteligentní dům, 2. vyd., ERA, 2008, ISBN: 978-80-7366-137-3

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání: 3.2.2020

Termín odevzdání: 8.6.2020

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

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

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

Abstrakt

Bakalářská práce představuje pojmy jako inteligentní budova, chytré elektroinstalace a také sběr MaR dat, který s touto problematikou souvisí. V úvodní části bude vysvětleno, co je to inteligentní budova, jaká zařízení obsahuje, jak spolu taková zařízení komunikují, poté bude vysvětlen komunikační protokol BACnet, který se pro takovou komunikaci využívá. Dále bude vysvětleno, co jsou to MaR data a k čemu se využívají. Další část se věnuje vysvětlení pojmu databáze, k čemu slouží a jaké existují typy databází. Představen bude i jazyk SQL, který se v databázových systémech používá. Bude také předveden databázový model části budovy fakulty elektrotechniky a komunikačních technologií T12. Část bude věnována i softwaru Coach^{AX} společnosti Honeywell, jenž je možné použít pro zpracování MaR dat. Na závěr bude vytvořena simulace MaR dat. Tyto data budou ukládána do databáze programu Microsoft SQL server a pak zpětně zobrazovány uživateli. Stejná data budou vhodně zobrazena ve vizualizaci realizované v programu Coach^{AX}.

Klíčová slova

Inteligentní budova, MaR data, jednotka AHU, protokol BACnet, Coach^{AX}, databáze, Microsoft SQL Server, jazyk SQL, generování dat

Abstract

This bachelor thesis introduces us to terms such as a smart building, smart electrical installation and measurement and control data collection, which relates to this topic. In the introductory part, it will be explained, what a smart building is, how the devices in said building communicate and the BACnet communication protocol used in these devices. Next it will be explained, what the measurement and control data are and what they are used for. The next part defines the term database, what it is for and all its types. SQL language, which is used in almost every database will be introduced next. After, a database model of a part of the Faculty of Electrical Engineering and Communication T12's building will be presented. A part will be about the Honeywell company's Coach^{AX} software, which is used for measurement and control data processing. In the end a measurement and control data simulation will be created. These data will be appropriately displayed in a visualization realized in Coach^{AX} program. At the end, a measurement and control data simulation will be created. This data will be saved into the Microsoft SQL server database and then processed by the user. This same data will then be displayed in visualization in Coach^{AX} program.

Keywords

Smart building, measurement and control data, AHU unit, BACnet protocol, Coach^{AX}, database, Microsoft SQL Server, SQL language, data generation

Bibliografická citace:

Citace tištěné práce:

ŽELINSKÝ, David. *Sběr MaR dat*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/126876>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Radek Štohl.

Citace elektronického zdroje:

ŽELINSKÝ, David. *Sběr MaR dat* [online]. Brno, 2020 [cit. 2020-06-03]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/126876>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Radek Štohl.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Sběr MaR dat jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: **3. června 2020**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **3. června 2020**

.....
podpis autora

Obsah

1	ÚVOD	9
2	INTELIGENTNÍ BUDOVY.....	10
2.1	Vzduchotechnika.....	10
2.2	Komunikační protokoly.....	13
2.3	Sběr MaR dat.....	17
3	DATABÁZE A DATABÁZOVÝ MODEL ŠKOLY	19
3.1	Databáze.....	19
3.2	Typy databází	20
3.3	Relační databáze	20
3.4	DBMS systémy.....	21
3.5	Jazyk SQL.....	22
3.6	Databázový model školy	26
4	COACH ^{AX}	29
4.1	Platformy	29
4.2	Stanice.....	29
4.3	Práce v programu Coach ^{AX}	30
5	SIMULACE DAT BUDOVY E.....	32
5.1	Vybraná data.....	32
5.2	Zpracované proměnné.....	32
5.3	Simulace	34
5.4	Připravení proměnných pro zápis do databáze.....	35
5.5	Nastavení komunikace s databází.....	36
5.6	Zápis dat do databáze.....	38
5.7	Vypisování dat z databáze	39
6	VIZUALIZACE MAR DAT.....	42
6.1	Úvodní strana	42
6.2	Hlavní menu.....	43
6.3	Výpis proměnných	44
6.4	Ovládání simulace.....	45
6.5	Teplota a vlhkost v místnostech	47
6.6	Vzduchotechnická jednotka.....	48
6.7	Alarmy.....	49
6.8	Výpis z databáze.....	50
6.9	Zobrazení proměnných a jejich časových průběhů	51
7	ZÁVĚR.....	52
	Literatura	53
	Seznam příloh.....	55

Seznam obrázků

Obrázek 2.1: Model vzduchotechnické jednotky	12
Obrázek 2.2: Architektura protokolu BACnet [10]	14
Obrázek 2.3: Příklad BACnet objektu [9]	15
Obrázek 2.4: Typy objektů protokolu BACnet [9]	15
Obrázek 2.5: Příklad služby přístupu k objektům [9]	16
Obrázek 3.1: Řídící databázový systém [13]	19
Obrázek 3.2: Databázový model budovy T12 [16]	27
Obrázek 5.1: Model budovy T12 [19]	32
Obrázek 5.2: Nabídka výběru proměnných pro databázi	38
Obrázek 5.3: Vyplněná tabulka <i>Budova</i>	38
Obrázek 5.4: Vyplněná tabulka <i>Místnost</i>	39
Obrázek 5.5: Vyplněná tabulka <i>Skupina</i>	39
Obrázek 5.6: Vyplněná tabulka <i>VZT</i>	39
Obrázek 5.7: Vyplněná tabulka <i>Typ_Parametr</i>	39
Obrázek 5.8: Výstupní tabulka z databáze	40
Obrázek 5.9: Nabídka pro vypisování dat z databáze	41
Obrázek 5.10: Výstupní tabulka z databáze v programu Coach ^{AX}	41
Obrázek 6.1: Úvodní strana vizualizace	42
Obrázek 6.2: Hlavní menu vizualizace	43
Obrázek 6.3: Soupis použitých proměnných	44
Obrázek 6.4: Vizualizace simulovaných dat	45
Obrázek 6.5: Ovládání simulace – místní ovládání 3.150 a doporučené hodnoty ...	46
Obrázek 6.6: Vizualizace snímání teploty a vlhkosti v místnostech	47
Obrázek 6.7: Vizualizace spuštěné vzduchotechnické jednotky	48
Obrázek 6.8: Vizualizace výpisu alarmů	49
Obrázek 6.9: Náhled proměnných vypsaných z databáze	50
Obrázek 6.10: Soupis proměnných s rozšířením historie	51

1 ÚVOD

Technika se stejně jako mnohé další obory rozvíjí mílovými kroky. Doby, kdy byly využívány pouze základní poznatky z oblasti elektrotechniky jsou nenávratně pryč. V dnešním světě lidé dokážou těchto znalostí lépe využívat a ulehčit si každodenní práci. Ideálním příkladem jsou zařízení označovaná jako „chytrá“, která lidem nabízejí spoustu užitečných funkcí šetřící čas i energii. Mezi ně patří například mobilní telefony, hodinky, navigace, náramky, kamery, drony nebo i chytré budovy.

V této bakalářské práci budou čtenáři obeznámeni s možností sběru dat pro měření a regulaci těchto inteligentních budov a jejich dalšího využití. Jako příklad bude použit model budovy Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně, konkrétně budovy Technická 12 (T12), části E. Z dostupných proměnných, které jsou touto budovou sbírány, bude vytvořena simulace těchto dat. Tu bude možno ovládat v dostupném programu Coach^{AX}, určeném pro tuto práci. Simulované data budou ukládány do databáze založené na modelu, který vytvořil pan Bc. Matouš Rathouzský ve stejnojmenné práci z roku 2018. Bude realizován i zpětný výpis dat z databáze do programu Coach^{AX}. Ve stejném programu bude vytvořena i vizualizace simulovaných dat.

2 INTELIGENTNÍ BUDOVY

Inteligentní (nebo také chytré) budovy se vyznačují tím, že se ke svým obyvatelům chovají přívětivě a snaží se jim zjednodušit každodenní život. Při výstavbě takové budovy se v současné době často dbá na její tepelnou úspornost. Ta ale s pojmem inteligentní budova nesouvisí tolik, co chytré elektroinstalace. Ty dělají inteligentní budovu “inteligentní”. [1]

Budova školy T12 patří mezi takzvané účelové budovy (budovy plnící určité funkční zaměření). Kromě škol to mohou být například nákupní střediska, kancelářské budovy, nemocnice nebo letiště. [2] Takové budovy se vybavují systémy pro řízení osvětlení (světla, žaluzie), vytápění, ovzduší, systémy zabezpečení (kamerové systémy, vstupní systémy, pohybová čidla, ...) – vším potřebným pro správný chod dané budovy. [3] Tyto systémy jsou propojeny skrz chytré elektroinstalace a můžou být buď plně automatizované nebo ovládané ručně například z jednotlivých pracovišť zaměstnanců v případě individuálních požadavků.

Z pohledu řízení budovy jsou tyto zařízení propojeny do jedné komunikační/vizualizační platformy. Řídící systém, který tuto komunikaci zajišťuje, se nazývá MaR (provozní soubor Měření a Regulace) nebo také BMS (Building Management System) a zajišťuje řízení vytápění, vzduchotechniky, klimatizace a chlazení. Takové zařízení se obecně označují jako TZB (Technické Zařízení Budov) nebo také HVAC (Heating, Ventilation, Air Conditioning). [4]

2.1 Vzduchotechnika

Do budov účelového typu bývá nezbytné přivádět velké množství čerstvého vzduchu. To pozitivně ovlivňuje výkony zaměstnanců (v případě škol studentů, akademických pracovníků a všech ostatních zaměstnanců), proto bývá většinou instalována vzduchotechnika, protože to bývá často mnohem lepší zdroj čerstvého vzduchu než běžná větrací okna. Ta například ve výškových budovách nejsou vůbec a taková budova tak zcela spoléhá na vzduchotechniku. [5]

V budově T12 není vzduchotechnika jediný zdroj čerstvého vzduchu, nicméně její řízení spadá pod chytré elektroinstalace.

Výhody vzduchotechniky oproti větrání okny patří:

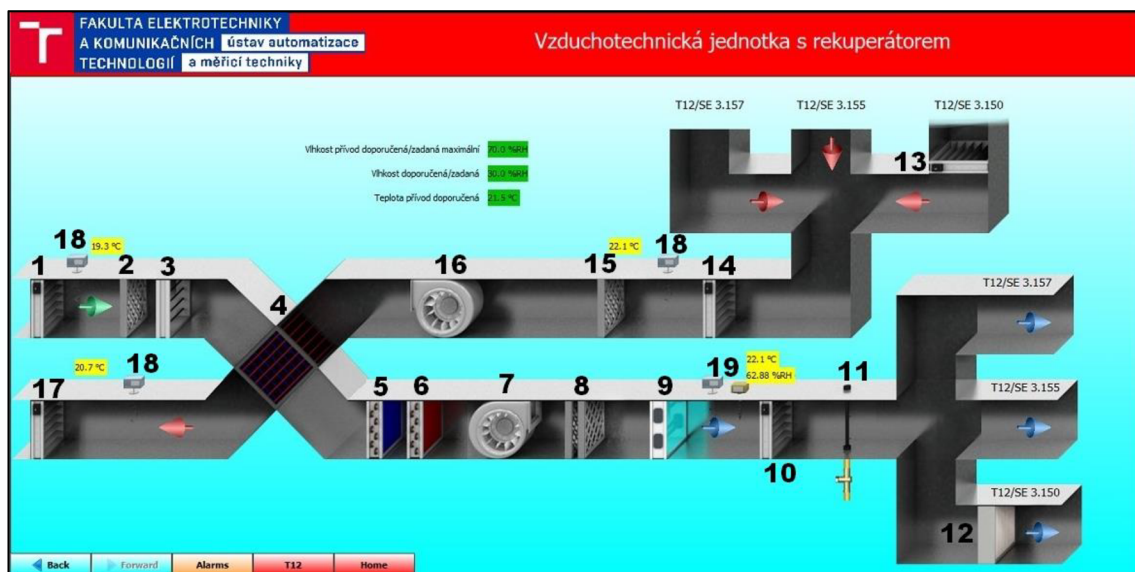
- Možnost automatizace
- Odstranění prachu z venkovního vzduchu
- Možnost tepelné regulace
- Možnost regulace vlhkosti
- Lze využít kdekoliv v budově včetně podzemí
- Nejlepší řešení pro místnosti, kde je nutná stálá teplota a vlhkost (laboratoře, serverovny, ...)
- V případě poruchy nebo nutné údržby dostane pověřená osoba informaci okamžitě bez průběžných kontrol vzduchotechniky
- Úspora nákladů na vytápění

Vzduchotechnika se skládá z potrubí rozvedeného po budově, ventilátorů a takzvaných vzduchotechnických jednotek, které vzduch „zpracovávají“. Tyto jednotky se umísťují nejčastěji na střechy nebo do podzemí budovy, ale můžou se nacházet i v jednotlivých patrech. Poslední možnost je realizována, pokud je potřeba přivádět vzduch s odlišnými vlastnostmi do různých místností.

2.1.1 Jednotka AHU

Obecnému typu vzduchotechnické jednotky se říká AHU (Air Handling Unit). Tuto jednotku tvoří část potrubí pro vzduch odsávaný z budovy a část potrubí pro přívod čerstvého vzduchu. Tyto části mohou být navzájem spojeny, což pomáhá rychlejší cirkulaci vzduchu v budově. Míchá se tedy čerstvý vzduch se vzduchem z budovy. Takový vzduch je čištěn v jednotce a poslán zpět do budovy. Do takových jednotek ale musí být nainstalován snímač CO₂, který v případě vysokého množství této sloučeniny dočasně uzavře průchod vzduchu zpětnou vazbou zpět do budovy. Do budovy tedy půjde pouze čerstvý venkovní vzduch.

Jinou možností je použít tzv. rekuperátor. V takové jednotce pak nemusí být instalovaný snímač CO₂, protože se vzduch v přívodu nemíchá se vzduchem v odtahu, ale pouze si předává teplo přes tepelně vodivé části rekuperátoru. Do budovy tak plyne pouze čerstvý vzduch, který je navíc ohříván teplem vzduchu, který plyne odtahovou částí, což šetří náklady na ohřev/chlazení vzduchu. Tato jednotka je použita i pro regulaci vzduchu vybraných místností budovy T12 bloku E. [6]



Obrázek 2.1: Model vzduchotechnické jednotky

V následující části budou vysvětleny jednotlivé části jednotky AHU a jejich význam.

1. **Vstupní klapky** – důležité pro regulaci procházejícího vzduchu. Svou funkci plní i při poruše, kdy zabraňují přenosu škodlivých látek nebo naopak přístupu vzduchu. Před nimi bývá mřížka, která zabraňuje průchodu větších těles.
2. **Hrubý filtr** – slouží k zabránění průchodu větších částic, které se dostaly do vzduchotechniky. V případě nutnosti ho lze snadno vyměnit.
3. **Elektrické vytápění** – to bývá instalováno v oblastech, kde venkovní vzduch bývá nižší než 0°C. Slouží k ohřevu studeného vzduchu, který by mohl způsobit zamrznání některých částí jednotky.
4. **Rekuperátor** – slouží pro předání energie vzduchu přívodu a odtahu
5. **Chladič** – pokud se vzduch nachází nad maximem teplotního rozsahu určeného pro budovu/místnost, začne chladičem proudit například studená voda, která přebírá teplo proudícího vzduchu.
6. **Ohřivač** – pokud se vzduch nachází pod minimem teplotního rozsahu určeného pro budovu/místnost, začne ohřivačem proudit například teplá voda, která bude předávat teplo proudícího vzduchu.
7. **Ventilátor přívodu** – pohání vzduch potrubím do budovy
8. **Jemný filtr** – určen pro zachycení prachových částic. Stejně jako u hrubého filtru, můžeme i tento filtr v případě nutnosti vyměnit.
9. **UV filtr** – zabraňuje šíření mikroorganismů do prostor budovy
10. **Protipožární klapky** – zabraňují šíření požáru a zplodin hoření
11. **Zvlhčovač** – v případě, že vlhkost vzduchu se nachází pod minimem doporučeného rozsahu, začne zvlhčovač vstříkovat do potrubí kapky

vody, které následně zvýší vlhkost na doporučenou hodnotu. V opačném případě je nutné zapnout ohřívač i chladič zároveň a na chladiči se začne srážet voda. Ohřívač potom ochlazený vzduch znovu ohřeje na doporučenou hodnotu

12. **HEPA filtr** – filtr určen pro zachycení nejmenších částic ve vzduchu. Používá se pro místnosti s vysokými nároky na čistotu.
13. **Klapky** – ovládají odtah vzduchu z laboratoře 3.150
14. Protipožární klapky
15. Hrubý filtr odtahu
16. **Ventilátor odtahu** – pohání vzduch potrubím ven z budovy
17. Výstupní klapky
18. **Snímače teploty**
19. **Snímač teploty a vlhkosti v přívodu jednotky**

2.2 Komunikační protokoly

Pro propojení zařízení a umožnění jejich vzájemné komunikace je třeba vyřešit problém, že každé zařízení může komunikovat po různých komunikačních sběrnících a také různými protokoly. Pojmy sběrnice a protokol jsou vysvětleny následujícími definicemi.

Sběrnice je „*skupina signálových vodičů, kterou lze rozdělit na skupiny řídicích, adresových a datových vodičů v případě paralelní sběrnice nebo sdílení dat a řízení na společném vodiči (nebo vodičích) u sériových sběrnic. Sběrnice má za účel zajistit přenos dat a řídicích povelů mezi dvěma a více elektronickými zařízeními. Přenos dat na sběrnici se řídí stanoveným protokolem.*“ [7]

Protokol je „*soubor pravidel pro komunikaci mezi dvěma nebo více uzly (systémy, regulátory)*“ [7] Komunikačních protokolů je spousta, ale pro automatizaci budov se používá především protokol BACnet.

2.2.1 Komunikační protokol BACnet

BACnet (Building Automation and Control Network) je standardním komunikačním protokolem pro automatizaci a řízení budov. Je to také celosvětová norma. Od roku 2003 je evropským standardem v rámci CEN (Committee for European Standardization) s označením ISO 16484-5. Jako protokol se může používat bez licenčních poplatků. Byl vyvinut americkým sdružením ASHRAE (American Society of Heating, Refrigeration and Air conditioning Engineers) za cílem vytvořit protokol umožňující spojení systémů různých výrobců do jednoho kooperujícího celku. [7], [8], [9]

Tento protokol umožňuje specifikovat většinu využívaných funkcí různých zařízení jako jsou analogové a dvoustavové (binární) vstupy a výstupy, ovládání,

plánování a další. Jedná se tedy o monitorování a řízení systémů, což má využití hlavně v automatizaci, řízení a komunikaci v budovách. Proto je založen na objektové reprezentaci a organizaci fyzických vstupů a výstupů. Objekty mohou reprezentovat jeden nebo více fyzických bodů, které dohromady vytváří určitou funkci.

Protokol BACnet lze využít pro:

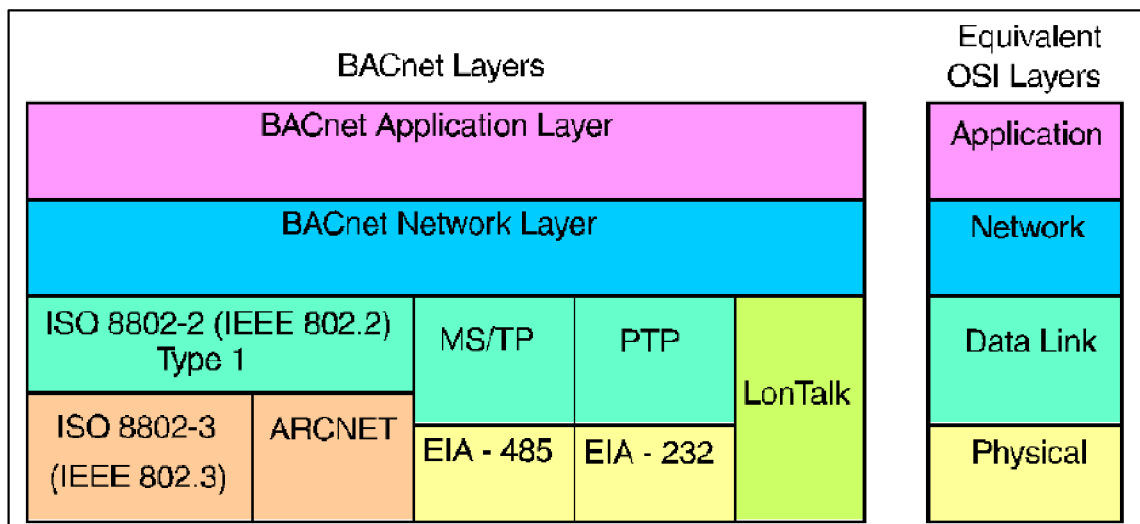
- Řízení TZB
- Detekci a hlášení požárů
- Řízení osvětlení
- Zabezpečení budovy
- Řídící rozhraní zařízení

Architektura BACnet

Specifikace protokolu BACnet je složena ze tří částí:

1. **Objekty** – metody, jak reprezentovat různá zařízení standardním způsobem
2. **Služby** – definuje zprávy zasílané počítačovou sítí pro sledování a řízení daného zařízení
3. **Komunikace** – definuje přístupné lokální sítě LAN, které se dají využít pro přenos zpráv

Samotná architektura je pak založena na modelu ISO/OSI.



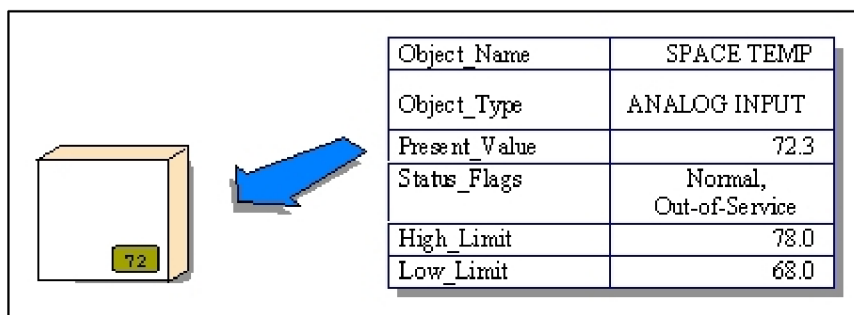
Obrázek 2.2: Architektura protokolu BACnet [10]

Protokol také umožňuje směřovat zprávy skrze existující IP sítě a Novell IPX sítě. Tyto protokoly jsou schopné zapouzdřit BACnet zprávy a přenést je pomocí tzv. tunelování (BACnet/IP Packet-Assembler-Disassembler: B/IP PAD). [8], [9]

BACnet objekty

Zařízení, přístroje, jednotky a snímače jsou v BACnet síti vždy reprezentovány jedním nebo skupinou BACnet objektů. Každý z nich má určitým způsobem nastaveny vlastnosti, které popisují jeho chování a řídí jeho provoz.

Každý objekt sítě BACnet poskytuje sadu vlastností, které se využívají pro organizaci, poskytování a získávání informací o daném objektu a řízení objektu. Můžeme si ho tedy představit jako tabulku tvořenou dvěma sloupci. Nalevo jsou jména informačních položek a napravo jejich údaj o hodnotě nebo vlastnosti.



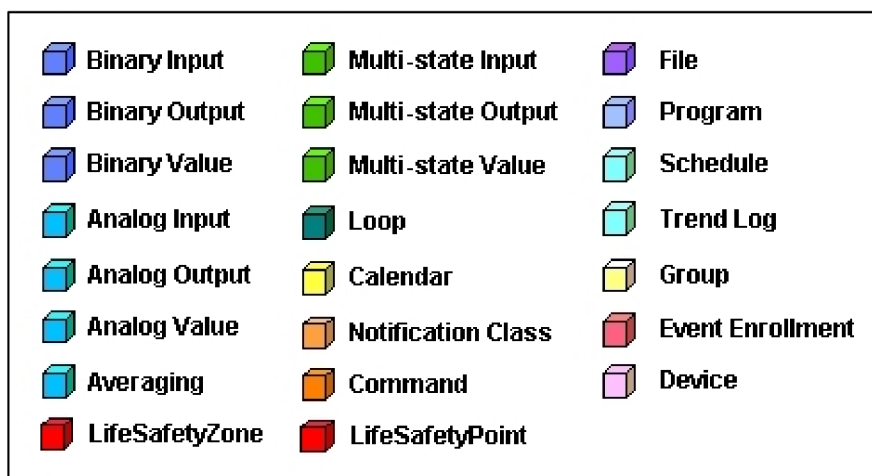
Object_Name	SPACE TEMP
Object_Type	ANALOG INPUT
Present_Value	72.3
Status_Flags	Normal, Out-of-Service
High_Limit	78.0
Low_Limit	68.0
























Obrázek 2.3: Příklad BACnet objektu [9]

Některé položky z vlastností objektu lze měnit (zapsat nebo změnit hodnotu či stav), jiné jsou určeny pouze pro čtení. Jako příklad bude sloužit snímač teploty. Tabulka vlastností tohoto objektu je uvedena na obrázku č. 2.3. Tento objekt je typu *ANALOG INPUT* neboli analogový vstup. *Present_Value* nám udává aktuální stav teploty ve °C. Další položka *Status_Flags* určuje, v jakém stavu se objekt nachází. Další položky nám definují, v jaké oblasti se má teplota tohoto snímače nacházet.

Většina těchto položek závisí na typu objektu, například u objektu reprezentující spínač určitého zařízení budeme hledat jiné položky, protože oproti snímači teploty bude hodnota dvoustavová a u té nemá smysl určovat její limity.

Protokol BACnet definuje celkem 23 různých standardních typů objektů:



 Binary Input	 Multi-state Input	 File
 Binary Output	 Multi-state Output	 Program
 Binary Value	 Multi-state Value	 Schedule
 Analog Input	 Loop	 Trend Log
 Analog Output	 Calendar	 Group
 Analog Value	 Notification Class	 Event Enrollment
 Averaging	 Command	 Device
 LifeSafetyZone	 LifeSafetyPoint	

Obrázek 2.4: Typy objektů protokolu BACnet [9]

Jsou to objekty s chováním přesně definovaným standardem a jsou tedy srozumitelné a snadno dekódovatelné pro ostatní aplikace. Každý objekt má takové parametry, které se v oblasti automatizace a řízení budov typicky vyžadují. [9]

BACnet služby

Jde o příkazy, které umožňují práci s objekty ve smyslu jejich obsluhy. Definují, jaké informace a úkony může jednotka BACnet chtít po jiné jednotce. Lze tedy říct, že jde o příkazy ve formě zpráv, které posílá jedno BACnet zařízení druhému s určitými požadavky. Zprávy jsou nazývány službami (*Services*), protože BACnet je založen na komunikačním modelu „Klient-Server“. Tyto služby jsou vysílány serverem k jeho příslušným klientům.

Jednotlivé aplikační služby se dělí do tříd podle jejich funkce:

- Služby událostí (Events) a hlášení (Alarms)
- Služby přístupu k souborům (File Access Services)
- Služby přístupu k objektům (Object Access Services)
- Služby poskytující vzdálenou správu zařízení (Remote Device Management Services)
- Služby virtuálního terminálu

Nejvyužívanější jsou služby přístupu k položkám objektů (třída služeb přístupu k objektům), které poskytují jednoduché nebo vícenásobné čtení a zápis dat (položek) objektů a spravovat jejich chování. [9]



Obrázek 2.5: Příklady služeb přístupu k objektům [9]

BACnet komunikace

Samotný systém vzájemného předávání zpráv mezi jednotlivými objekty je velmi důležitou částí každé komunikační technologie a BACnet není výjimkou. Ten většinou využívá již existujících/přenosových protokolů. Je to hlavně Ethernet,

ARCNET, LongTalk, což jsou zcela soběstačné LAN systémy, vyžadující jen minimální práci se specifikací způsobu přenášení BACnet zpráv. Tyto protokoly byly zvoleny z důvodu, že splňují celosvětové požadavky na řízení a automatizaci budov ve smyslu rychlosti, propustnosti, ceny, uživatelského přístupu atd. [9]

2.3 Sběr MaR dat

Pokud je již zajištěna komunikace mezi nainstalovanými zařízeními, je na čase využít jejich potenciál naplno a začít sbírat data, které poskytují. To do budoucna ušetří spoustu času, zpříjemní pobyt obyvatelům budovy, kteří nebudou muset nijak zasahovat do řízení těchto přístrojů. Bude možné je také efektivněji využívat a provádět včas jejich údržbu, případně lépe diagnostikovat jejich poruchu.

2.3.1 Skupiny měřených dat

Měřené hodnoty a snímané signály se dělí do tří skupin:

1. Přímé signály poruch
2. Kumulované hodnoty
3. Odvozené signály poruch

Přímé signály poruch

Většinou jde o dvoustavové (binární) vstupy, které hlásí, jestli porucha nastala nebo ne. V našem případě to budou poruchy aktivních prvků v AHU. Pokud je signál ve stavu 1 (*true*), znamená to, že nastala porucha. Tyto signály jsou běžně dostupné v systému MaR a v historických datech lze tedy dohledat, kdy a k jaké poruše došlo. K přímým signálům poruch také patří vybočení analogových hodnot mimo zadaný rozsah nebo binární proměnné, které informují o nutnosti údržby určitého zařízení (například výměna filtrů ve vzduchotechnice). [11]

Kumulované hodnoty

Do této skupiny patří hodnoty naměřené spotřeby energie, provozní hodiny a jiné. Jedná se o hodnoty, které s plynoucím časem stále rostou (kumulují se). U takových hodnot má smysl počítat denní, týdenní, měsíční nebo roční přírůstky. Nejčastěji se jedná o hodnoty vodoměrů, elektroměrů, plynoměrů, ale také například počet startů zařízení. [11]

Odvozené signály poruch

Jde o výsledky vyhodnocování naměřených i historických dat statistickými metodami pro detekci poruch. Hodnoty tedy jsou sbírány ve střední až dlouhé časové oblasti (například kumulované hodnoty), které jsou pak srovnávány s hodnotami sbíranými v reálném čase.

Sbírání těchto hodnot není v současné době moc rozšířené, nicméně se zvyšujícím se výpočetním výkonem a rostoucí záznamové kapacitě regulačních stanic se čeká nárůst sběru těchto dat. To může do budoucna poskytnout užitečné informace, které mohou odhalit možnou poruchu zařízení dříve, než dojde k jeho nenávratnému poškození. [11]

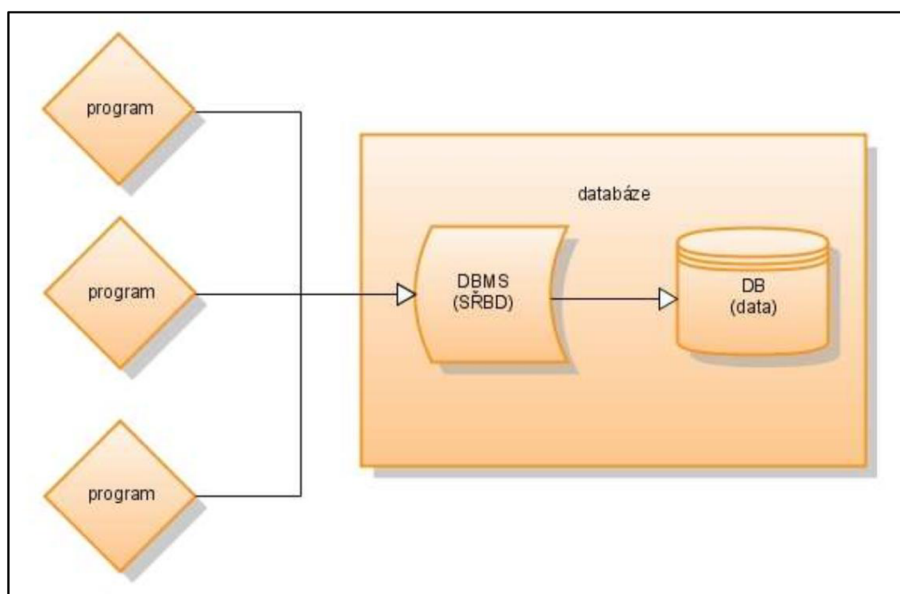
3 DATABÁZE A DATABÁZOVÝ MODEL ŠKOLY

Vyhodnocovat lze i aktuální data, ale pokud je potřeba vyhodnocovat a dohledávat historická data, bude nutné data někam zapisovat. K tomu slouží databáze. Okem neznalého člověka může vypadat databáze jako spousta tabulek s velkým množstvím dat. V následující kapitole bude tedy podrobněji vysvětleno, co to databáze je, k čemu slouží a jaké programy jsou pro práci s ní určeny.

3.1 Databáze

„Databáze je organizovaný soubor strukturovaných informací neboli dat, které se obvykle ukládají v elektronické podobě v počítačovém systému.“ [12]

Pro správu takové databáze je potřeba řídicí systém (DBMS – Database Management System). Jako databázový systém (databázi) pak označujeme data a systém DBMS.



Obrázek 3.1: Řídicí databázový systém [13]

Data jsou ukládána do množiny tabulek z řádků a sloupců, což nám umožňuje efektivně zpracovávat a filtrovat data pomocí takzvaných dotazů. Většina databází používá k zadávání dat a vytváření dotazů jazyk SQL. [12]

3.2 Typy databází

V závislosti na využití danou firmou/organizací se používají různé typy databází. [12] Mezi základní patří:

- **Relační databáze** – databázi tvoří sada; poskytuje nejefektivnější způsob přístupu ke strukturovaným informacím
- **Objektově orientované databáze** – informace jsou reprezentovány formou objektů, podobně jako v objektově orientovaném programování
- **Distribuované databáze** – skládá se z dvou nebo více souborů nacházejících se na různých umístěních (více počítačů, jedna nebo více sítí)
- **Datové sklady** – centrální úložiště dat; databáze určená pro rychlé zadávání dotazů a analýzu
- **Databáze noSQL** – také známá jako nerelační databáze; umožňuje ukládání a manipulaci s nestrukturovanými a částečně strukturovanými daty

3.3 Relační databáze

Relační databáze využívá tabulek, které jsou propojeny předem nastavenými vztahy. Sloupce tabulek se nazývají **atributy** a řádkům říkáme **záznamy**. Atributy určují vlastnosti objektů zapsaných to tabulek. Do tabulky se vkládají objekty stejného druhu, nicméně každý objekt je v tabulce pouze jednou. Při návrhu tabulky slouží atributy k definici vlastností, které budou mít vkládaná data. [13]

Atribut

Atributem je tedy sloupec v tabulce, který definuje vlastnost záznamů. Při návrhu tabulky se určuje **doména** atributu – rozsah hodnot, kterých může záznam nabývat. Z domény se při sestavení databáze určí datový typ hodnoty. Databáze poté nedovolí zapsat hodnotu v jiném formátu než je formát domény.

Existují speciální druhy atributů, které plní specifické funkce v databázi. Je to primární klíč a cizí klíč.

Primární klíč

Aby se dalo pevně určit pořadí záznamů a pak jednoznačně určit daný záznam, je nutné zvolit jeden atribut tabulky, který bude unikátní. Takovému atributu se říká **primární klíč**. Každý záznam v tabulce musí mít hodnotu tohoto atributu unikátní pro celou tabulku.

Cizí klíč

Pokud jsou záznamy v tabulkách propojeny vztahy, musí být tyto záznamy provázány. Toho lze docílit použitím **cizího klíče** v jedné z tabulek. Do tohoto atributu se poté bude kopírovat hodnota cizího klíče navázaného záznamu z druhé tabulky.

Propojení tabulek – relace

Při návrhu databáze se určují logické vztahy mezi tabulkami. Jako příklad bude sloužit databáze bloků budovy T12 a místností. V jedné tabulce máme vypsané bloky, v druhé místnosti daných bloků. Pro zjištění vztahu mezi tabulkami, je nutné zjistit, v jakém vztahu jsou záznamy. Tomuto vztahu se říká **multiplicita**. V příkladu to bude vztah **1:n**, protože blok může obsahovat více místností, ale určitá místnost nemůže být ve více blocích.

Dalším typem multiplicity je vztah **1:1**. V tomto případě odpovídá jednomu záznamu v jedné tabulce právě jeden záznam v druhé tabulce. Pokud tedy jedna tabulka bude seznam studentů, kteří dojíždí do školy autem a druhá tabulka bude seznam jejich aut, bude vztah mezi těmito tabulkami 1:1, protože právě jednomu studentovi patří právě jedno auto na parkovišti.

Poslední vztah je **m:n**, který je vždy nutný rozdělit na dva vztahy 1:n. Vytvoří se tedy další tabulka, ve které se budou objevovat jen cizí klíče ze zbylých dvou tabulek. [13]

3.4 DBMS systémy

Jak již bylo zmíněno, DBMS systémy jsou systémy pro řízení báze dat, které se starají o fyzické uložení záznamů do tabulek, správu a požadované operace. V současnosti mezi nejvyužívanější systémy patří MySQL, Microsoft Office Access, Microsoft SQL server nebo například Oracle Database. [12]

MySQL

MySQL je relační databázový systém s otevřeným zdrojovým kódem založený na SQL. Je navržen pro webové aplikace a může běžet na libovolné platformě. MySQL je díky své flexibilitě oblíbený mezi webovými vývojáři pro vytváření zejména elektronických obchodů.

Microsoft Office Access

DBMS systém Microsoft Office Access je program pro návrh a řízení menších databází. Program je součástí balíčku Office 365 a bývá často předinstalovaný na počítačích se systémem Windows. Slouží hlavně jednotlivcům a menším firmám.

Program má grafické uživatelské rozhraní a je proto použitelný i pro uživatele bez znalosti jazyka SQL. [14]

Microsoft SQL server

System Microsoft SQL server je profesionální nástroj pro vytváření a řízení databází libovolného rozsahu. Je využíván velkými firmami zejména pro lepší zpracování dat. Cena některých verzí se pro běžné uživatele může pohybovat v řádech desítek tisíc korun. Zde už je znalost jazyka SQL nezbytná. [14], [15]

Case Studio 2

Program Case Studio 2 nepatří mezi DBMS systémy, ale úzce s nimi souvisí. Slouží pro návrh databází bez použití jazyka SQL. Návrh se řeší graficky, pro každou tabulku lze připravit všechny atributy včetně primárních a cizích klíčů a také nastavit multiplicitu. Tento program byl použit pro návrh databázového modelu školy, který byl pak exportován do souboru, který lze otevřít v programu Microsoft SQL studio. Spuštěním tohoto souboru se vygeneruje databáze podle vytvořeného návrhu.

3.5 Jazyk SQL

SQL (Structured Query Language) je strukturovaný dotazovací jazyk určený pro práci s databázemi. Tento jazyk byl vyvinut společností IBM v 70. letech. Pro tento jazyk poté vznikly standardy ANSI (American National Standards Institute) v roce 1986 a ISO (International Organization for Standardization) v roce 1987. Přestože je SQL stále velmi používaný, vznikají současné době i nové jazyky pro databáze, případně se používají nerelační databáze. [12], [20], [21]

3.5.1 Datové typy

Každý údaj v databázi je určitého typu. Údaje jsou v tabulkách uspořádány tak, že každý sloupec obsahuje jen jeden datový typ. Mezi nejpoužívanější patří textový řetězec, číslo, datum nebo čas. Jediná hodnota, která lze napsat do jakéhokoli datového typu je hodnota *NULL*, která definuje nezapsanou hodnotu tedy neexistenci hodnoty nebo prázdné pole v tabulce. [20], [21]

Dále budou uvedeny nejčastěji používané datové typy. Ty je možné rozdělit do tří kategorií:

- Předdefinované datové typy
- Konstruované datové typy
- Uživatelem definované datové typy

Nejpoužívanější jsou datové typy předdefinované. Ty můžeme dále rozdělit na:

- Znakové datové typy
- Číselné datové typy
- Datové typy pro uložení hodnot data a času
- Datové typy *Large Object*

Znakové datové typy

Slouží pro ukládání převážně textových údajů. Skládají se z alfanumerických znaků.

- **CHAR(x)** – textový řetězec s pevnou délkou x
- **VARCHAR(x)** – textový řetězec s proměnnou délkou, maximální délka definovaná pomocí x
- **NCHAR(x)** – textový řetězec s pevnou délkou x ve vybrané národní znakové sadě
- **NVARCHAR(x)** – textový řetězec s proměnnou délkou ve vybrané národní sadě, maximální délka definovaná pomocí x

Číselné datové typy

Slouží pro ukládání číselných údajů. Můžou to být celá čísla, čísla s pevnou desetinnou čárkou, finanční částky s přesností na dvě nebo čtyři desetinná místa, nebo desetinných čísel používaných pro vědecko-technické výpočty.

- **BIT** – slouží pro vyjádření logické hodnoty ($0 = false$, $1 = true$)
- **INT** – celé číslo v rozsahu $\langle -2^{31}; 2^{31}-1 \rangle$
- **SMALLINT** – celé číslo v rozsahu
- **TINYINT** – celé číslo v rozsahu $\langle 0; 255 \rangle$
- **DECIMAL** – číslo s pevnou desetinnou čárkou v rozsahu $\langle -10^{38}-1; 10^{38}-1 \rangle$
- **REAL** – datový typ s pohyblivou desetinnou čárkou v rozsahu $\langle -3.40E+38; 3.40E+38 \rangle$
- **FLOAT** – datový typ s pohyblivou desetinnou čárkou v rozsahu $\langle -1.79E+308; 1.79E+308 \rangle$
- **MONEY** – číslo v rozsahu $\langle -922\,337\,203\,685\,477,5808; 922\,337\,203\,685\,477,5807 \rangle$, podporuje pouze Microsoft SQL server
- **SMALLMONEY** – číslo v rozsahu $\langle -214\,748,3648; 214\,748,3647 \rangle$, podporuje pouze Microsoft SQL server

Datové typy pro uložení hodnot data a času

Tyto datové typy mají poměrně velkou variabilitu v závislosti na národních formátech pro datum a čas. Patří sem typy DATA, TIME, TIMESTAMP. Těmi lze vyjádřit:

- hodnotu data a času (například 1. 1. 2020 12:04:20)
- hodnotu datového a časového intervalu

3.5.2 Struktura jazyka SQL

Pro orientaci v jazyce SQL bude použita systematická kategorizace. [20] Ta rozděluje příkazy do čtyř kategorií:

- Jazyk pro definici dat (DDL – Data Definition Language)
- Jazyk pro manipulaci s daty (DML – Data Manipulation Language)
- Příkazy pro řízení přístupu (DCL – Data Control Language)
- Příkazy pro řízení transakcí (TCC – Transaction Control Commands)

Ke každé kategorii budou uvedeny nejčastěji používané příkazy s jejich syntaxí. Ta se může od jiných zdrojů lišit vzhledem k možnosti vynechání některých klíčových slov, což některé příkazy umožňují.

Data Definition Language

Příkazy z této kategorie definují struktury a vytváří různé objekty v databázi. Mezi nejpoužívanější příkazy patří:

- **CREATE DATABASE** – vytváří databázi
- **CREATE TABLE** – vytváří tabulku, definuje sloupce a jejich datové typy. Také je možné přidat parametr jako je primární/cizí klíč nebo nenulovou hodnotu

```
CREATE DATABASE jmeno_databaze;  
  
CREATE TABLE jmeno_tabulky  
(  
    jmeno_sloupce datovy_typ parametr,  
    ...,  
    jmeno_sloupce datovy_typ parametr  
);
```

- **ALTER TABLE** – umožňuje provádět změny ve struktuře tabulky, jako je například přidání sloupce nebo definování výchozí hodnoty pro určitý sloupec

```
ALTER TABLE jmeno_tabulky  
(  
    prikaz jmeno_sloupce datovy_typ,  
    ...,  
    prikaz jmeno_sloupce datovy_typ  
);
```

- **DROP TABLE** – používá se ke smazání určité tabulky včetně všech dat v ní uložených

```
DROP TABLE jmeno_tabulky;
```

Data Manipulation Language

Příkazy umožňují manipulaci s údaji, to znamená výběr údajů, jejich vkládání a aktualizaci v tabulkách nebo jejich mazání. Patří sem příkazy jako:

- **SELECT** – tímto příkazem lze vybírat data z jedné nebo více tabulek

```
SELECT jmeno_sloupce, ..., jmeno_sloupce
FROM jmeno_tabulky;
```

- **INSERT INTO** – slouží pro manuální vkládání dat do tabulky

```
INSERT INTO jmeno_tabulky (sloupec_x, sloupec_y)
VALUES (hodnota_x, hodnota_y);
```

- **UPDATE** – používá se k úpravě již existujících hodnot v tabulkách. Pro přesnější úpravu se používá podmínka WHERE, jinak se přepíší všechny definované hodnoty.

```
UPDATE jmeno_tabulky (sloupec_x, sloupec_y)
SET sloupec_x = hodnota_x, sloupec_y = hodnota_y, ...
WHERE podminka;
```

- **DELETE FROM** – maže řádky z tabulky nebo tabulek za definované podmínky, bez podmínky maže všechny řádky, ale tabulka zůstává

```
DELETE FROM jmeno_tabulky
WHERE podminka;
```

Příkazy pro spojování tabulek

Pomocí příkazu **JOIN** lze spojovat různé tabulky databáze, což je užitečné při složitějším výběru pro zobrazení nebo mazání dat. Takto je možné vytvořit 5 různých spojení:

- **INNER JOIN** – po spojení bude tabulka obsahovat pouze záznamy, kde se hodnoty spojovacích atributů shodují
- **LEFT OUTER JOIN** – po spojení bude tabulka obsahovat záznamy obou tabulek, kde se hodnoty spojovacích atributů shodují, a také ostatní záznamy z tabulky napsané jako první do příkazu
- **RIGHT OUTER JOIN** – po spojení bude tabulka obsahovat záznamy obou tabulek, kde se hodnoty spojovacích atributů shodují, a také ostatní záznamy z tabulky napsané jako druhá do příkazu
- **FULL OUTER JOIN** – po spojení bude tabulka obsahovat všechny záznamy z obou tabulek
- **CROSS JOIN** – výsledná tabulka bude obsahovat všechny kombinace záznamů z obou tabulek přes spojovací atribut

```
SELECT sloupec1, sloupec2, sloupec3, ...
FROM jmeno_tabulky1
INNER JOIN jmeno_tabulky2 ON jmeno_tabulky1.tab2ID =
jmeno_tabulky2.ID;
```

Ostatní SQL příkazy

- **AS** – tzv. alias je příkaz pro vytvoření nových názvů tabulek nebo sloupců pro zkrácení a zjednodušení SQL dotazu

```
SELECT jmeno_sloupece AS jmeno_alias
FROM jmeno_tabulky;
```

- **GROUP BY** – slouží pro seskupení záznamů podle hodnot v určitém sloupci. Často se spojuje s dalšími funkcemi, jako je COUNT, MIN, MAX, SUM nebo AVG.

```
SELECT jmeno_sloupece, ..., jmeno_sloupece
FROM jmeno_tabulky
WHERE podmínka
GROUP BY jmeno_sloupece;
```

- **ORDER BY** – slouží pro seřazení záznamů podle hodnot v určitém sloupci vzestupně nebo sestupně

```
SELECT jmeno_sloupece, ..., jmeno_sloupece
FROM jmeno_tabulky
WHERE podmínka
ORDER BY jmeno_sloupece, ..., jmeno_sloupece ASC|DESC;
```

Jako ve většině jazyků, i v SQL je možné používat běžné příkazy, jako jsou logické operátory AND, OR, NOT, ..., matematické operátory +, -, *, /, =, <>, <=, >=, ..., nebo různé funkce, jako je ABS, MIN, MAX, SUM, AVG, COUNT a další.

Data Control Language

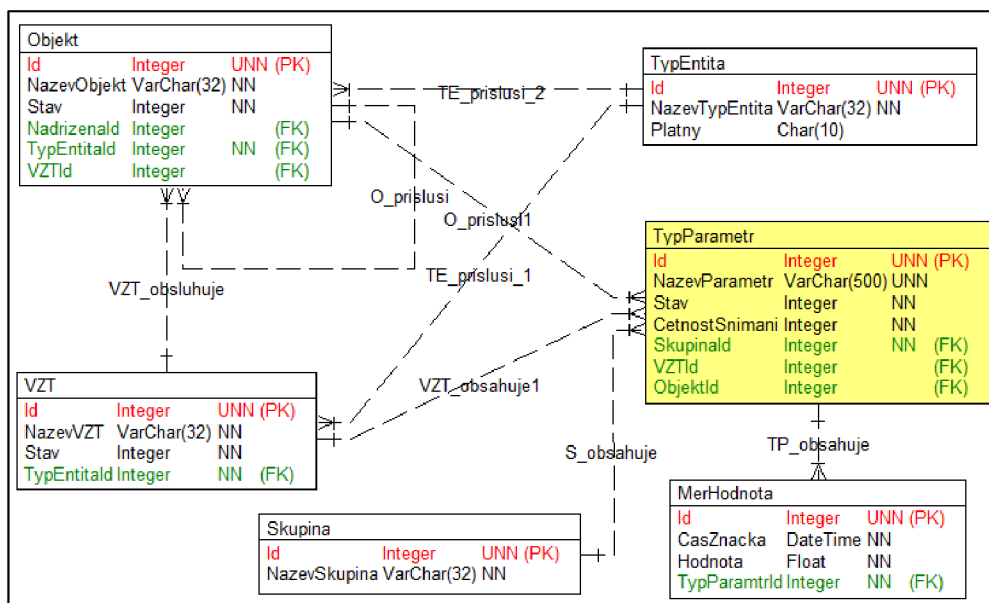
Tato kategorie obsahuje příkazy pro řízení provozu a údržbu databáze. Díky nim je možné přidávat nebo odebírat uživatelská práva pro přístup. Příkazy této kategorie jsou CREATE USER, ALTER USER, DROP USER, GRANT a REVOKE. [20]

Transaction Control Commands

Kategorie obsahuje příkazy pro řízení transakcí, jako jsou SET TRANSACTION, COMMIT, ROLLBACK a SAVEPOINT. [20]

3.6 Databázový model školy

Model byl vytvořen panem Bc. Matoušem Rathouzkým [16] v programu Case Studio 2 a součástí tohoto modelu je 6 tabulek, které jsou navzájem propojeny vztahy 1:n. Je to tedy model relační databáze. Pro úplnost bude vysvětleno, co jednotlivé tabulky v modelu reprezentují.



Obrázek 3.2: Databázový model budovy T12 [16]

Základní tabulka s názvem **TypEntita** obsahuje data o celku jako je například areál nebo budova. Obsahuje atributy *ID* (zároveň primárním klíčem tabulky), *NazevTypEntita* (název budovy nebo areálu) a *Platny*, což definuje stav dat v této tabulce.

Navazující tabulka nese jméno **VZT** a obsahuje názvy zařízení vzduchotechniky. Její atributy jsou *ID* (primární klíč), *NazevVZT* (název daného zařízení), *Stav* (stav dat v tabulce) a *TypEntitaID* (cizí klíč z tabulky *TypEntita*, realizuje vztah 1:n mezi těmito tabulkami).

Třetí tabulkou je **Objekt**, která sbírá data jako jsou názvy jednotlivých bloků nebo místností budovy. Obsahuje atributy *ID* (primární klíč), *TypEntitaID* a *VZTID* (cizí klíče, realizují vztahy mezi tabulkami), *Stav* a *Nadrizenald*, což je takzvaná selfrelace, která odkazuje zpět do tabulky na nadřizená data, a tak zlepšuje přehlednost takových dat.

MerHodnota je čtvrtou tabulkou modelu a slouží pro zápis MaR dat. Její atributy jsou *ID* (primární klíč), *CasZnacka* (datový typ DateTime; poslouží nám pro zobrazení dat v historii), *Hodnota*, a *TypParametrID* (cizí klíč).

Informace o skupině měřených dat nalezneme v tabulce **Skupina**. Slouží pro třídění dat, pokud budeme sbírat jiná data než MaR. Má atributy *ID* (primární klíč) a *NazevSkupina*.

Poslední tabulkou je **TypParametr** a ta obsahuje informace proměnných a o četnosti snímání jejich hodnot. Obsahuje atributy *ID* (primární klíč), *NazevParametr*, *Stav*, *CetnostSnimani*, *SkupinaID* (cizí klíč), *VZTID* (cizí klíč), *ObjektID* (cizí klíč).

Tento model byl mírně upraven (názvy tabulek vhodné pro tuto práci) a poté exportován do formátu čitelného programem Microsoft SQL server. Výsledkem exportu je SQL dotaz, ve kterém jsou SQL příkazy pro vytvoření identické databáze, podle již zmíněného modelu. Při spuštění se vygenerují jednotlivé tabulky a databáze je tak připravena pro plnění daty. [16]

4 COACH^{AX}

Program Coach^{AX} je nástroj pro zařízení HAWK nebo program Arena^{AX}, který souží jako monitorovací software pro inteligentní budovy. Jak Coach^{AX}, tak Arena^{AX} jsou založeny na Frameworku Niagara^{AX}. Všechny tyto nástroje jsou součástí systému CentraLine^{AX} od společnosti Tridium patřící pod Honeywell. Framework Niagara^{AX} je psaný v jazyce Java, proto lze přidat do programu Coach^{AX} jakékoliv nadstavby napsané v tomto jazyce. [17], [18]

4.1 Platformy

Platforma v programech AX může být považována za operační systém, který je nainstalovaný na počítači s programy AX (v našem případě Coach^{AX}). Platformy mají za úkol poskytovat prostředí a služby stanicím běžícím na této platformě.

V navigačním okně obsahuje platforma několik podprogramů sloužících k nastavení prostředí podle představ uživatele. Při prvním zapnutí programu Coach^{AX} je důležitý podprogram *License Manager*, který slouží k aktivaci zakoupené licence. *Application Director* slouží k zapínání a vypínání stanic. Počet stanic na platformě může být neomezený, ale pouze jedna může být aktivní. *Platform Administration* lze použít ke kompletní záloze stanic případně jejich restartu, pokud se ocitly v nefunkčním stavu. Je možné tu najít i soupis chybových hlášení. *Software Manager* poskytuje soupis dostupných softwarových modulů. Zde lze také nainstalované moduly nastavovat. [17], [18]

4.2 Stanice

Stanice tvoří prostory pro práci uživatele. Spouštění stanic se provádí přes podprogram *Application Director* dané platformy. Všechny stanice jsou zabezpečené přihlašovacím jménem a heslem, které si zvolí uživatel.

Práce ve stanicích může probíhat v různých zobrazeních. Základní je zobrazení *Wire sheet*, které zobrazuje funkční bloky nebo v případě složky s podsložkami budou zobrazeny ve *Wire sheet* jednotlivé podsložky.

Při poklikání na funkční bloky nebo složky se otevře zobrazení podrobností a nastavení.

Další zobrazení je editor Px souborů. Ty slouží pro vizualizaci.

Poslední zobrazení je náhled Px souborů, který je možné využít pro testování funkčnosti vizualizace.

4.3 Práce v programu Coach^{AX}

4.3.1 Navigation tree

Po připojení na platformu a vybranou stanici lze začít pracovat. Na levé straně je vidět objektový (navigační) strom (*Navigation Tree*), kde jsou všechny vytvořené složky a soubory. V rámci bakalářské práce jsou to složky se soubory, kde jsou zobrazeny proměnné, simulace dat těchto proměnných, soubory vizualizace, nastavení propojení mezi programy Coach^{AX} a Microsoft SQL server. V oddělení *Files* jsou pak pomocné soubory jako například hlavičky stránek vizualizace, obrázky a podobné.

4.3.2 Palette

V okně palety nástrojů (*Palette*) jsou veškeré pomocné objekty, které mohou generovat data, provádět různé operace, případně vytvořit vizualizaci těchto dat pomocí pevných i animovaných/pohyblivých objektů.

Pro práci s daty bude využita hlavně paleta *kitControl* a pro grafiku *bajoui*, *kitPx*, *kitPxHvac* a *kitPxGraphics*. Pro vizualizaci alarmů, která bude vytvořena, je určena paleta *alarm*.

4.3.3 Práce s daty

Hlavní práce s daty probíhá ve *Wire Sheet*. Proměnné lze vytvořit přidáním bloků proměnných typů *Numeric* (pro analogové hodnoty) nebo *Bool* (pro hodnoty binární). Jednoduchý způsob na jejich přidání je kliknutí pravé myši na pracovní plochu a vybrání dané proměnné. Pak stačí vybrat vstupy, výstupy a vhodný název.

Další prvky výsledné logiky se vkládají přesunutím z palety. Ty nejběžnější se nachází v paletě *kitControl*. Jsou to například konstanty, logické funkce, matematické funkce, časovače a podobné. I tyto bloky jdou přejmenovat a změnit jejich vstupy a výstupy.

Pak už nezbývá nic jiného než spojit tyto bloky dohromady podle potřeby.

4.3.4 PX grafika

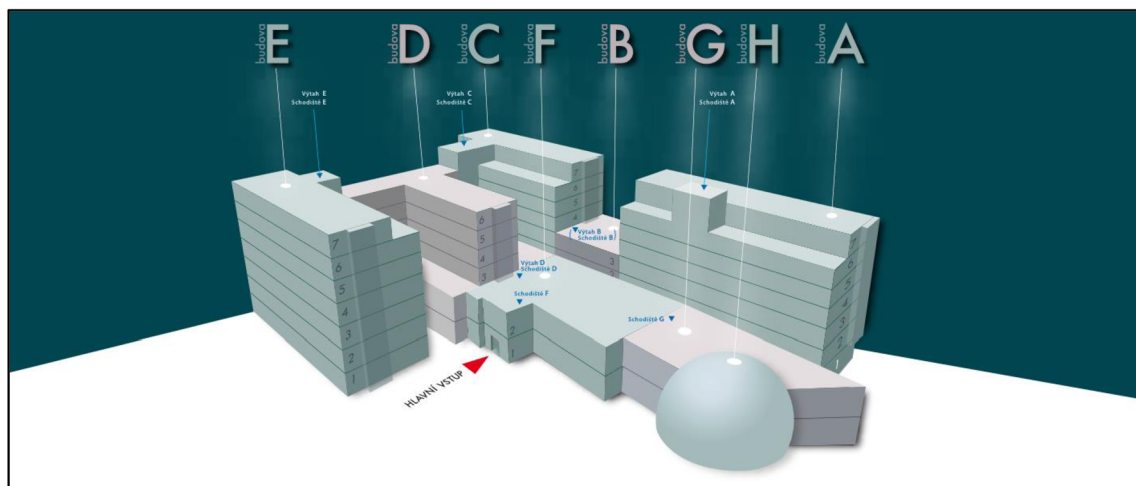
Pro vytvoření listu vizualizace je nutné vytvořit pracovní prostor. Ten se vytváří kliknutím pravým tlačítkem myši na příslušnou složku a následným kliknutím na *New View*. Zobrazí se prázdný list s pomyslnými okraji. Pozadí se vytvoří přidáním prvku *CanvasPane*. Poklikáním na něj ve *Widget Tree* se otevře nabídka, která slouží pro jeho nastavení.

Přidávání dalších prvků komplikuje změna jejich formátu při přetažení na původní prvek s pozadím. Proto se prvky nejprve musí přesunout na prostor mimo tento prvek a poté na něj, což už neovlivní jejich vzhled.

Pak se přidají další prvky jako hlavička, tlačítka pro přesun mezi stránkami vizualizace, grafy, případně okna se zobrazováním hodnot proměnných v reálném čase. Zde se nepřesouvá prvek z palety nástrojů, ale samotná proměnná z navigačního okna.

5 SIMULACE DAT BUDOVY E

Budova E, pro kterou byla tvořena simulace dat a vizualizace, je sedmipodlažní a vyhrazené prostory v ní má Ústav radioelektroniky (6. a 7. podlaží), Ústav telekomunikací (5. podlaží), Ústav biomedicínského inženýrství (3. a 4. podlaží) a Ústav automatizace a měřicí techniky (1. a 2. podlaží). [19]



Obrázek 5.1: Model budovy T12 [19]

5.1 Vybraná data

Proměnné byly vybrány z oblasti vzduchotechniky pod označením VZ13, což jsou proměnné, které zajišťují chod vzduchotechniky pro laboratoře genomiky a proteomiky SE 3.150 a SE 3.155 a chodbu SE 3.157. Tyto místnosti patří pod ústav biomedicínského inženýrství. Vzhledem k využití místností (práce s buněčnými kulturami a genetickým materiálem) je na regulaci vzduchu kladen velký důraz. V laboratoři SE 3.150 je udržována čistota dle třídy čistoty ISO 7 a 8 pomocí HEPA filtrů, germicidních lamp a germicidní filtrace vzduchu, konstantní teplota 21,5 °C a vlhkost vzduchu 30-70 %RH. [22]

5.2 Zpracované proměnné

Celkem bylo zpracováno 52 proměnných z celkových 183 pro budovu E. Proměnné jsou roztrženy do 6 tříd podle toho, zda jsou vstupní, výstupní nebo pouze pomocné a analogové nebo digitální (dvoustavové/binární).

Digitální vstupy

- Zanesení filtru odtah
- Zanesení filtru přívod 1
- Zanesení filtru přívod 2
- Chod zdroje chladu
- Chod (dP) ventilátor odtah
- Chod (dP) ventilátor přívod
- Chod zvlhčovače
- Protimrazová ochrana
- Porucha zdroje chladu
- Porucha zvlhčovače
- Přepínač místního ovládní
- Stav stykače čerpadla ohřev
- Zvlhčovač připraven k provozu
- Zvlhčovač potřebuje servisní kontrolu
- Chod filtru přívodu s UV
- Zanesení filtru přívodu s UV
- Požadován servis UV filtru přívodu
- Zanesení filtru přívodu Hepa

Digitální výstupy

- Klapka přívod a odtah
- Klapka přívod a odtah (3.150)
- Ventilátor odtah
- Ventilátor přívod
- Vlhčení zapnutí
- Zapnutí filtru přívodu s UV
- Režim chlazení
- Čerpadlo ohřev

Digitální proměnné

- Porucha ventilátor odtah
- Porucha ventilátor přívod
- Způsob regulace teploty
- Způsob regulace vlhkosti
- Povolení vlhčení
- Protimrazová ochrana kritická
- Porucha čerpadla ohřev

Analogové vstupy

- Teplota přívod měřená
- Teplota vnitřní měřená (3.150)
- Teplota vnitřní měřená (3.155)
- Teplota vnitřní měřená (3.157)
- Vlhkost přívod měřená
- Vlhkost vnitřní měřená (3.150)
- Vlhkost vnitřní měřená (3.155)
- Vlhkost vnitřní měřená (3.157)
- Teplota odtah měřená
- Teplota za rekuperátorem měřená
- Teplota vstup měřená

Analogové výstupy

- Povel pro chlazení
- Klapka rekuperace

- Vlhčení výkon

Analogové proměnné

- Teplota přívod doporučená
- Vlhkost doporučená
- Vlhkost zadaná
- Vlhkost přívod doporučená maximální
- Vlhkost přívod zadaná maximální

5.3 Simulace

Simulace dat je vytvořena v programu Coach^{AX} v zobrazení *Wire sheet*, které umožňuje vytvářet určitou celkovou logiku simulace a spojovat proměnné a ostatní bloky podle potřeb dohromady. Zobrazení je uvedeno v příloze č. 2.

Všechny bloky jsou pro alespoň částečnou přehlednost rozděleny (stejně jako proměnné v předchozí části) do tříd podle toho, jestli jsou analogové nebo digitální, vstupy, výstupy nebo pouze pomocné proměnné.

Nevýhodou tohoto prostředí je nemožnost měnit trasu spojů pro ještě větší přehlednost tak, jak to jde v prostředí programů LabView nebo rozšíření Simulink programu Matlab. Coach^{AX}, zato umožňuje velmi jednoduché propojování bloků napříč jednotlivými listy *Wire sheet*. Stačí vytvořit bloky na jednom listu, spojit je podle potřebné logiky a pak určité bloky přesunout na jiný list. Spojení zůstane, bude plně funkční. Toto spojení lze poznat podle ukončeného vedení z/do bloku. To se vytváří také v případě příliš komplikované cesty vedení mezi ostatními bloky.

Tato funkce byla využita pro připojení bloků simulace s bloky určenými pro zápis do databáze. Tyto bloky jsou pojmenovány stejně jako proměnné zapsané v souboru *Vybrané proměnné – BP* v příloze č. 3. Jejich názvy jsou přesné, ale často nelze poznat, o jakou proměnnou se jedná, a tak jsou nevhodné pro vytváření simulace. Ve složce *Simulace* jsou tedy bloky slovně pojmenované podle jejich funkce a jejich výstup je přímo spojen se vstupem databázových bloků ve složce *Data*.

5.3.1 Logika spojení bloků v simulaci

Vzhledem k tomu, že se nepoužívají reálná data, ale jedná se o simulaci, je nutné vymyslet podmínky pro spuštění jednotlivých proměnných. Nejjednodušší je použít dva bloky binárních proměnných, jeden bude určovat, jestli je nebo není pracovní den, a druhý bude rozhodovat o pracovní době. Proto se musí vložit logický blok *AND* před každý vstupní aktivní prvek vzduchotechniky (ohřívač, chladič, zvlhčovač, ventilátory, klapky a UV filtr). Do bloku *AND* se připojí oba binární bloky.

Ke všem těmto blokům patří i chybové bloky (znečištění, poruchy, nutné servisní kontroly). Tyto bloky jsou dvoustavové a stav 1 znamená, že je prvek znečištěn, má poruchu nebo potřebuje servisní kontrolu. Proto za každý poruchový blok je nutné vložit logický blok *NOT* pro obrácení logiky. Převertáčený výstup je možno zapojit do bloku *AND* odpovídající proměnné.

Pasivní prvky vzduchotechniky, jako jsou například filtry, se spouštět nemusí. Vizualizace by se tedy mohla řídit pouze tímto blokem. Bohužel některé bloky vizualizace jsou analogové a zobrazují například postupné znečištění filtru.

Nejjednodušší řešení tohoto problému je binární hodnoty vynásobit 100 za pomoci matematického bloku *MULTIPLY*. Bloky budou mít sice jen dva stavy, ale ty alespoň půjdou zobrazit. Tímto způsobem jsou převedeny i některé aktivní prvky vzduchotechniky, u kterých byly problémy se zobrazováním.

Binární výstupní proměnné jsou zapojeny stejně jako binární vstupní proměnné. Výstupy zapojených bloků *AND* zapojíme do odpovídajících výstupů, pokud takové výstupy jsou. To je možné z důvodu toho, že výstupní proměnné jsou přímo ovládané, vstupní pouze signalizují, v jakém stavu se prvky nachází. Pokud určité výstupy nemají svůj ekvivalentní vstupní protějšek, vytvoří se nová logika s blokem *AND* podle předchozího postupu.

Pak jsou tu analogové proměnné snímačů teplot a vlhkosti. Jejich hodnoty už se nedají ovládat binárně, a tak se musí simulovat. Ideální dostupný blok je generátor sinusového signálu. Jeho minimální a maximální hodnota je nastavena tak, aby prvky vzduchotechniky měnily své stavy a animovaná vizualizace tak měla smysl.

V simulaci se mohou vyskytnout i speciální případy, pro které je nutno vymyslet jedinečné zapojení. Jednoduchým příkladem je elektrický ohřívač, který slouží jako protimrazová ochrana. Při nízkých teplotách by mohly některé prvky vzduchotechniky zamrznat. Proto pokud je vstupní teplota nižší než bod mrazu a je pracovní den a pracovní doba, tak se tento ohřívač zapne. Chybovou proměnnou nemá, ale má jen omezený výkon, který je určen vzhledem k okolním podmínkám budovy. Nicméně může nastat situace, že venkovní teplota bude až příliš nízká a ohřívač by nebyl schopen vzduch ohřát natolik, aby k zamrznání ostatních částí vzduchotechniky nedocházelo. Proto je zde proměnná s názvem *Protimrazová ochrana kritická*. Ta se řídí vstupní teplotou.

Ideálním případem je chladič. Jeho podmínky pro zapnutí jsou: pracovní den, pracovní doba, žádná porucha a teplota větší než 21,5 °C nebo jakákoliv teplota, ale vlhkost větší než 30 %RH. Vlhkost se začne srážet na chladiči a vlhkosti bude tedy ubývat. Z toho plyne, že pokud bude chladič chladit vzduch kvůli vysoké teplotě, bude i ubírat vlhkosti, a tak se musí zapnout zvlhčovač. Naopak, pokud bude chladič zapnutý kvůli vysoké vlhkosti, sníží se teplota pod ideální hodnotu a musí se zapnout ohřívač, který by jinak neběžel.

Všechny potřebné bloky pro vytvoření takové simulace jsou pod paletou nástrojů *kitControl*.

5.4 Přípravení proměnných pro zápis do databáze

Pro správný zápis dat do databáze byla vytvořena ve stanici složka *Data*, kde jsou proměnné rozříděné podle datového typu. Ve složce je tedy dalších 6 podsložek (*Digital input*, *Digital output*, *Analog input*, *Analog output*, *Pseudo analog*, *Pseudo*

digital), ve kterých už jsou jednotlivé proměnné se svým pravým názvem uvedeným v souboru *Vybrané proměnné – BP* v příloze č. 3. Tyto proměnné jsou napojeny na výstupy všech proměnných použitých v simulaci. Stejně proměnné by byly použity, pokud by byl systém napojen na používání reálných dat z budovy T12.

Coach^{AX} je v této chvíli schopen uchovávat pouze aktuální hodnotu proměnné. Pro jednodušší zápis do databáze je tedy nutné uchovávat historická data proměnných už v programu Coach^{AX}. Toho lze docílit přidáním rozšiřujícího prvku *history* každé proměnné.

Globální přidání rozšíření proměnným je možné službou *ProgramService*, která se nachází v navigačním stromu stanice pod položkou *Config/Services*. Po otevření se zobrazí okno s objekty. Pro jistotu, že se vybírají správné objekty, je třeba nabídku vyčistit (*Clear All*) a následně vyhledat objekty (*Find Objects*). Následně se vybere složka s proměnnými a jejich typ. V tomto případě má více složek stejný typ (např. *Digital*), proto můžeme vyhledat proměnné stejného typu i v dalších složkách před přidáním rozšiřujícího prvku.

Jestliže se v nabídce *Object* zobrazují všechny proměnné daného typu, stačí stisknout tlačítko *Add Slot*, pojmenovat prvky, zvolit typ *history* a podle datového typu zvolit *NumericIntervalHistoryExt* pro analogové proměnné, případně *BooleanIntervalHistoryExt* pro binární proměnné. Řádek *Enabled* slouží pro zapnutí ukládání historie a v řádek *Interval* nastavuje časový interval, se kterým se budou hodnoty do historie ukládat. Po stisknutí tlačítka *OK* by se mělo po chvíli zobrazit okno s informací o úspěšném vytvoření globálního rozšíření pro zvolené proměnné. Proces je třeba zopakovat i pro proměnné jiného datového typu.

Za zmínku stojí i to, že se v navigačním stromu stanice zobrazila nová položka *History*, pod kterou je každé rozšíření historie všech proměnných všech datových typů. Po rozkliknutí libovolného prvku se zobrazí graf závislosti hodnoty dané proměnné na čase od prvního uložení hodnoty do rozšíření.

5.5 Nastavení komunikace s databází

I když program Coach^{AX} nabízí ukládání historických hodnot, se kterými lze poté pracovat a vyhodnocovat je, tak je stále výhodnější z pohledu bezpečnosti i lepší práce s daty ukládat data do databáze.

Pro komunikaci s databází je nutné vytvořit spojení, přes kterou bude program Coach^{AX} s databází komunikovat. Takové sítě je možné v nástroji *Drivers*. Po otevření tohoto nástroje a kliknutí na tlačítko *New* se otevře nabídka. Pro komunikaci s databází je nutné vybrat položku *Rdbms Network*. Po jejím přidání a otevření je nutné opět kliknout na *New* a zde si vybrat vhodnou databázi. V tomto

případě je to *SqlServerDatabase*, protože databáze bude v programu Microsoft SQL server.

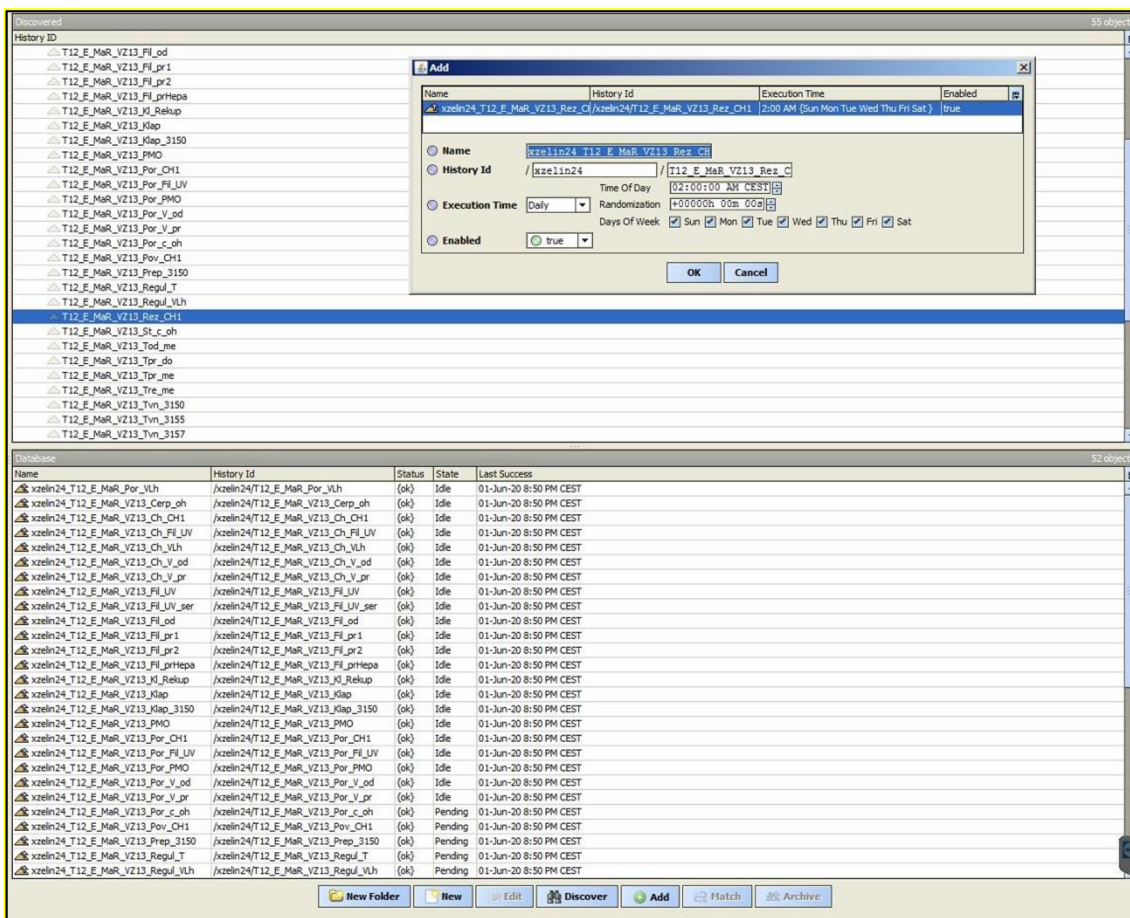
Pokud při vytváření je uživatel obeznámen se všemi databázemi a chce proces urychlit, může použít paletu nástrojů. Pro databázi v MS SQL server má paleta název *rdbSqlServer*. Do nástroje *Drivers* stačí přesunout prvek *RdbmsNetwork* z palety nástrojů a do tohoto prvku další prvek s názvem *SqlServerDatabase*.

Dále je třeba nastavit prvek *SqlServerDatabase*. Řádky *User Name* a *Password* slouží pro vyplnění přihlašovacích údajů do Microsoft SQL serveru. Ten je ve výchozím režimu nastaven tak, že se jako přihlašovací údaje používají údaje stejné jako pro přihlášení do systému Windows. To ale může být pro komunikaci problémové. Proto je tu možnost vytvořit vlastní účet pro MS SQL server a zadat tyto údaje.

Pole *Instance Name* slouží pro zadání názvu databáze. Dále je potřeba nastavit adresu, kde se SQL server s databází nachází. V této práci se používala databáze na školním serveru, na kterém je i program Coach^{AX}. Proto stačí zadat *rserver*. Na závěr už jen stačí prvek zapnout pomocí volby *true* v poli *Enabled*. Po uložení, aktualizaci a zjištění odezvy (*ping*) by se měl zobrazit v řádku Status stav *{ok}*.

V této chvíli tedy Coach^{AX} komunikuje s MS SQL serverem, ale nemá žádné data. Vybrání proměnných se realizuje přes položku *Histories* která je součástí nástroje *SqlServerDatabase*. Po jejím otevření se obrazovka rozdělí na dvě okna a kliknutím na tlačítko *Discover* se v horním okně objeví prvek se jménem podle stanice obsahující proměnné s rozšířením historie.

Po vybrání proměnné a kliknutí na *Add* se otevře další okno s parametry pro zaslání hodnot do databáze. V tomto případě se hodnoty zasílají s intervalem 10 s. Pro zapnutí stačí změnit hodnotu v řádku *Enabled* na *true* a odsouhlasit stiskem *OK*. Pokud vše proběhlo v pořádku, měl by se záznam objevit ve spodním okně a zde se bude po zadaném intervalu aktualizovat. Současně by se měla vytvořit tabulka pro každou zvolenou proměnnou ve vytvořené databázi.



Obrázek 5.2: Nabídka výběru proměnných pro databázi

V případě, že uživatel chce zapisovat více proměnných, lze je vybrat současně. Případně lze vybrat všechny. Každá proměnná má v databázi stejnojmennou tabulku, takže je možné postupně třídit data až v MS SQL serveru.

5.6 Zápis dat do databáze

Data se do databáze se již zapisují, ale do tabulek pojmenovaných podle proměnných. V modelu databáze data žádná nejsou. Jako první je vhodné ručně vyplnit všechny tabulky kromě tabulky *Měřená hodnota*, do které se budou pomocí SQL dotazu zapisovat data hromadně.

ID	NazevBudovy	platny
1	T12_E	1
➤➤	NULL	NULL

Obrázek 5.3: Vyplněná tabulka *Budova*

	ID	Mistnost_Nazev	Stav	T12_MaR_ID	VZT_ID
▶	2	SE_3.150	1	1	1
	3	SE_3.155	1	1	1
	4	SE_3.157	1	1	1
*	NULL	NULL	NULL	NULL	NULL

Obrázek 5.4: Vyplněná tabulka *Místnost*

	ID	Nazev_Skupina
▶	1	MaR
*	NULL	NULL

Obrázek 5.5: Vyplněná tabulka *Skupina*

	ID	VZT_Nazev	Stav	T12_MaR_ID
	1	VZ13	1	1
▶*	NULL	NULL	NULL	NULL

Obrázek 5.6: Vyplněná tabulka *VZT*

	ID	Nazev_Parametr	Stav	Cetnost_snimani	Skupina_ID	VZT_ID	Mistnost_ID
▶	2	XZELIN24_T12_E_MAR_VZ13_VLVN_3150	1	10	1	1	2
	4	XZELIN24_T12_E_MAR_VZ13_VLVN_3155	1	10	1	1	3
	5	XZELIN24_T12_E_MAR_VZ13_VLVN_3157	1	10	1	1	4
	6	XZELIN24_T12_E_MAR_VZ13_TVN_3150	1	10	1	1	2
	8	XZELIN24_T12_E_MAR_VZ13_TVN_3155	1	10	1	1	3
	10	XZELIN24_T12_E_MAR_VZ13_TVN_3157	1	10	1	1	4
	11	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	10	1	1	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Obrázek 5.7: Vyplněná tabulka *Typ_Parametr*

Zápis naměřených dat bude realizován pomocí SQL dotazu (*New Query*). Z informací obsažených v kapitole 3.5 lze vytvořit jednoduchý dotaz pro zápis hodnot do tabulky *Měřená data*. Dotaz porovnává název proměnné s již vyplněnými názvy v tabulce *Typ parametr*. Pokud bude název špatně zadaný, neměl by dotaz fungovat. Dotaz zapisuje celou uloženou historii každé proměnné. Proto je nutné skript před vyhodnocováním spustit. Přitom nemusí být program Coach^{AX} ani spuštěný, protože data se do databáze zapisovaly automaticky, SQL dotaz pouze přepíše data z jednotlivých tabulek do navrhnutého modelu.

```
insert into dbo.Merena_hodnota
(TypParametrID, Cas_Znacka, Hodnota)
```

```
select Typ_Parametr.ID, TAB.TIMESTAMP, TAB.VALUE
from dbo.XZELIN24_T12_E_MAR_VZ13_TVN_3155 as TAB, dbo.Type_Parametr
where Typ_Parametr.Nazev_Parametr='XZELIN24_T12_E_MAR_VZ13_TVN_3155';
```

5.7 Vypisování dat z databáze

Pro vypisování dat je SQL dotaz složitější vzhledem ke spojování tabulek do jedné, přehledné a pro koncového uživatele jednodušší pro pochopení. Dotaz tedy vytvoří jednu tabulku, kterou vytvoří z vybraných dat v různých tabulkách modelu.

```

select
Budova.NazevBudovy           as Budova,
Skupina.Nazev_Skupina       as 'Skupina dat',
Mistnost.Mistnost_Nazev    as Mistnost,
VZT.VZT_Nazev              as 'Podskupina dat',
Typ_Parametr.Nazev_Parametr  as 'Název proměnné',
Merena_hodnota.Hodnota     as Hodnota,
Merena_hodnota.Cas_Znacka  as Čas

from Merena_hodnota

left join Typ_Parametr on Typ_Parametr.ID=Merena_hodnota.TypParametrID
left join VZT on VZT.ID = Typ_Parametr.VZT_ID
left join Skupina on Skupina.ID = Typ_Parametr.Skupina_ID
left join Mistnost on Mistnost.ID = Typ_Parametr.Mistnost_ID
left join Budova on Budova.ID = Mistnost.T12_MaR_ID

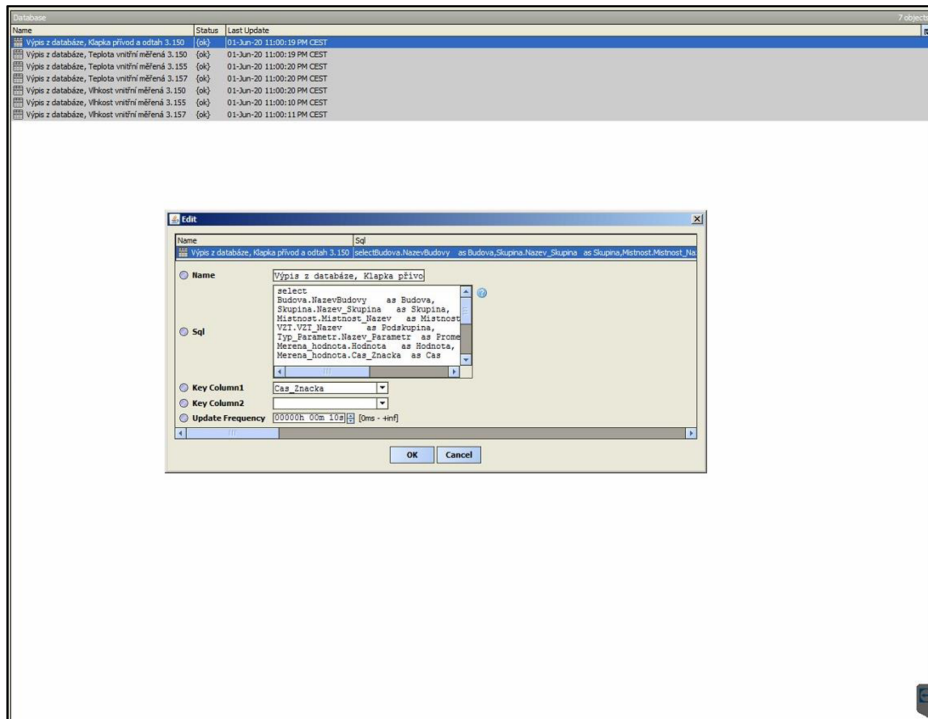
where Typ_Parametr.Nazev_Parametr='XZELIN24_T12_E_MAR_VZ13_KLAP_3150'
order by Merena_hodnota.Cas_Znacka

```

	Budova	Skupina dat	Mistnost	Podskupina dat	Název proměnné	Hodnota	Čas
1	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:30:50.037
2	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:31:00.010
3	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:31:10.020
4	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:31:20.010
5	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:31:30.007
6	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:31:40.007
7	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:31:50.027
8	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:32:00.007
9	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:32:10.017
10	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:32:20.013
11	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:32:30.010
12	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:32:40.037
13	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:32:50.007
14	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:33:00.017
15	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:33:10.020
16	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:33:20.013
17	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:33:30.007
18	T12_E	MaR	SE_3.150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1	2020-05-27 15:33:40.010

Obrázek 5.8: Výstupní tabulka z databáze

Pro přenos dat zpět do programu Coach^{AX} je nutné vytvořit body (*points*), které plní stejnou činnost jako již zmíněný SQL dotaz. Na rozdíl od něj se automaticky aktualizuje. Do nabídky datového bodu se napíše stejný SQL dotaz jako v SQL serveru. Jedinou změnou je víceslovné pojmenování sloupců, které program Coach^{AX} nepodporuje.



Obrázek 5.9: Nabídka pro vypisování dat z databáze

Pro zobrazení dat stačí rozkliknout datový bod a kliknout na tlačítko *Discover*. Výsledek pak vypadá následovně:

Budova	Skupina	Místnost	Podskupina	Promenna	Hodnota	Cas
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:30 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:31 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:31 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:31 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:31 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:31 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST
T12_E	MaR	SE_3_150	VZ13	XZELIN24_T12_E_MAR_VZ13_KLAP_3150	1.00	27-May-20 3:32 PM CEST

Obrázek 5.10: Výstupní tabulka z databáze v programu CoachAX

6 VIZUALIZACE MAR DAT

6.1 Úvodní strana

Na úvodní straně je k vidění plánec budovy T12. Po kliknutí na blok E se zobrazí náhled hlavního menu. Přesun na další stranu byl řešen jako *hyperlink polygon*. Ve vlastnostech tohoto objektu s definovatelnými okraji je nastaven odkaz na stránku s hlavním menu.



Obrázek 6.1: Úvodní strana vizualizace

List obsahuje navíc hlavičku s tlačítky zpět a dopředu (funkčně stejné jako tlačítka v internetovém prohlížeči), tlačítko s odkazem na úvodní stranu a na hlavní menu. Prostřední tlačítko přepíná na výpis alarmů. Tato hlavička je stejná pro všechny listy.

6.2 Hlavní menu




Zde máme na výběr z více možností přesunu. Jednotlivými tlačítky se dostaneme na odpovídající listy týkající se bloku E.



Obrázek 6.2: Hlavní menu vizualizace

6.3 Výpis proměnných

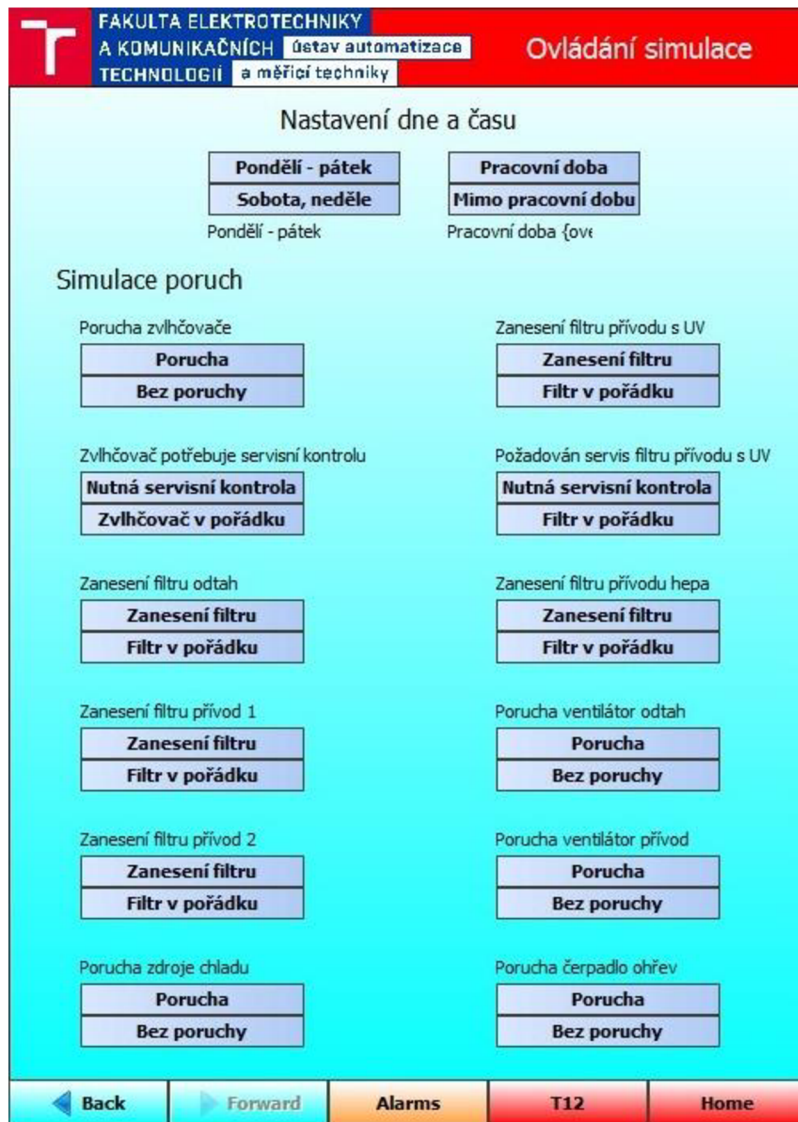
Na tomto listu jsou vypsané použité proměnné podle jejich typu. Byly použity analogové a digitální vstupy, výstupy i pomocné proměnné.

 FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ		Ústav automatizace a měřicí techniky		Proměnné	
Digitální vstupy	Zanesení filtru odtah, Zanesení filtru přívod 1, Zanesení filtru přívod 2, Chod zdroje chladu, Chod (dP) ventilátor odtah, Chod (dP) ventilátor přívod, Chod Zvlhčovače, Protimrazová ochrana, Porucha zdroje chladu, Porucha zvlhčovače, Stav stykače čerpadla ohřev, Zvlhčovač připraven k provozu, Zvlhčovač potřebuje servisní kontrolu, Chod filtru přívodu s UV, Zanesení filtru přívodu s UV, Požadován servis filtru přívodu s UV, Zanesení filtru přívodu Hepa, Přepínač místního ovládání				
Digitální výstupy	Klapka přívod a odtah, Klapka přívod a odtah (3.150), Ventilátor odtah, Ventilátor přívod, Vlhčení zapnutí, Zapnutí filtru přívodu s UV, Režim chlazení, Čerpadlo ohřev				
Digitální proměnné	Porucha ventilátor odtah, Porucha ventilátor přívod, Způsob regulace teploty, Způsob regulace vlhkosti, Povolení vlhčení, Protimrazová ochrana kritická, Porucha čerpadla ohřev				
Analogové vstupy	Teplota přívod měřená, Teplota vnitřní měřená (3.150), Teplota vnitřní měřená (3.155), Teplota vnitřní měřená (3.157), Vlhkost vnitřní měřená (3.150), Vlhkost vnitřní měřená (3.155), Vlhkost vnitřní měřená (3.157), Vlhkost přívod měřená, Teplota odtah měřená, Teplota za rekuperátorem měřená, Teplota vstup měřená				
Analogové výstupy	Povel pro chlazení, Klapka rekuperace, Vlhčení výkon				
Analogové proměnné	Teplota přívod doporučená, Vlhkost doporučená, Vlhkost zadaná, Vlhkost přívod doporučená, maximální, Vlhkost přívod zadaná maximální				
 Back		 Forward		Alarms	
				T12	
				Home	

Obrázek 6.3: Soupis použitých proměnných

6.4 Ovládání simulace

List „Ovládání simulace“ slouží k zapínání/vypínání vzduchotechniky pomocí nastavení pracovních dnů a pracovní doby. Všechny ostatní tlačítka jsou k simulaci poruch. Ty mohou být jak kritické nebo nekritické (servisní kontroly). Po správném stisku tlačítka by se měla změna projevit na funkčním modelu vzduchotechniky a také na výpisu alarmů.



Obrázek 6.4: Vizualizace simulovaných dat

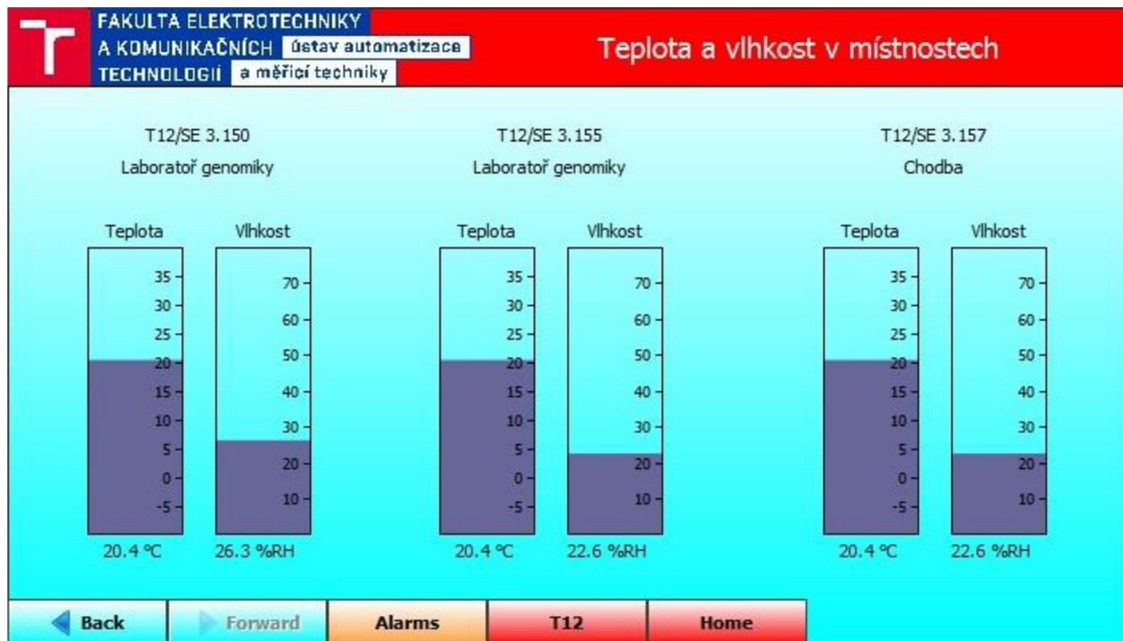
Další list ovládání simulace je určen pro přepínání místního ovládání v laboratoři SE 3.150. Lze také změnit přednastavené doporučené hodnoty. Změny se projeví na modelu vzduchotechniky.



Obrázek 6.5: Ovládání simulace – místní ovládání 3.150 a doporučené hodnoty

6.5 Teplota a vlhkost v místnostech

Na šesti ukazatelích je možné vidět aktuální stav teploty a vlhkosti v obou laboratořích a chodbě u laboratoří. Přesná hodnota se ukazuje pod každým ukazatelem.



Obrázek 6.6: Vizualizace snímání teploty a vlhkosti v místnostech

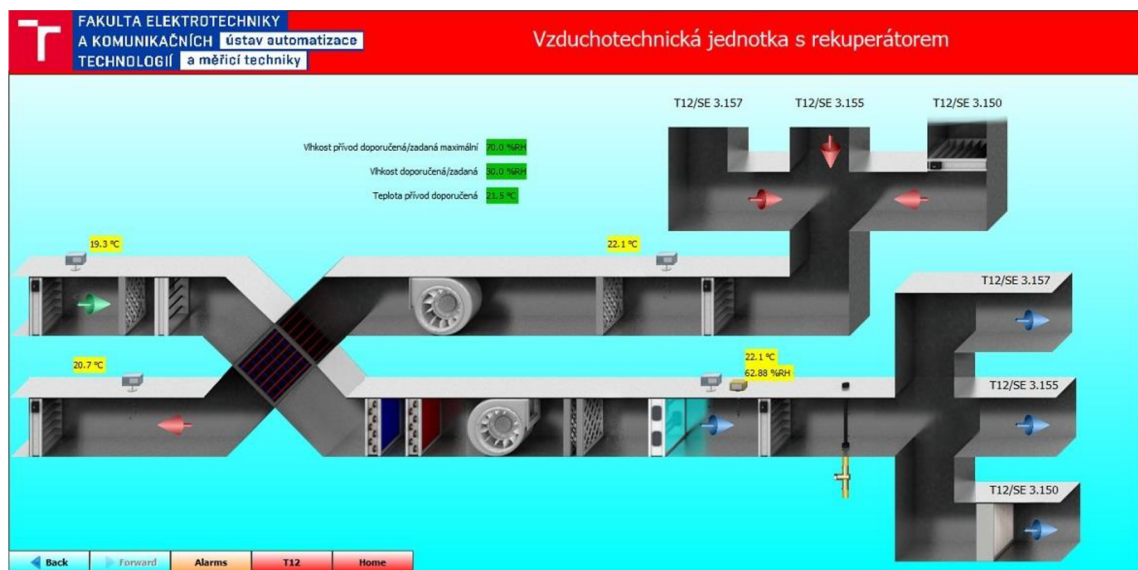
6.6 Vzduchotechnická jednotka

Funkčnost vzduchotechnické jednotky byla vysvětlena již v úvodu. Pro vytvoření této animace byly použity prvky palety *kitPxGraphics*. Případně můžou být použity i prvky palety *kitPxHvac*, které jsou názornější, ale není mezi nimi rekuperátor, což by znemožňovalo vytvořit správnou simulaci.

Tento list vizualizace je animovaný. Všechny prvky kromě snímačů jsou napojeny na simulaci, která se stará o jejich stavy. To znamená, že se simulací otvírají a zavírají klapky, zapínají a vypínají ventilátory, zvlhčovač, chladič a ohřívač a znečišťují filtry. U snímačů jsou zobrazeny aktuální naměřené hodnoty. Doporučené hodnoty jsou zobrazeny nad modelem.

Šipky ukazují proudění vzduchu. Na pravé straně se potrubí dělí do již zmiňovaných tří místností. V dělení je vidět, že laboratoř SE 3.150 má na svém vstupu HEPA filtr a na výstupu klapky, řízené při manuálním ovládnání.

Tento model je pouze názorný a neukazuje opravdové vedení potrubí vzduchotechniky.



Obrázek 6.7: Vizualizace spuštění vzduchotechnické jednotky

6.7 Alarmy

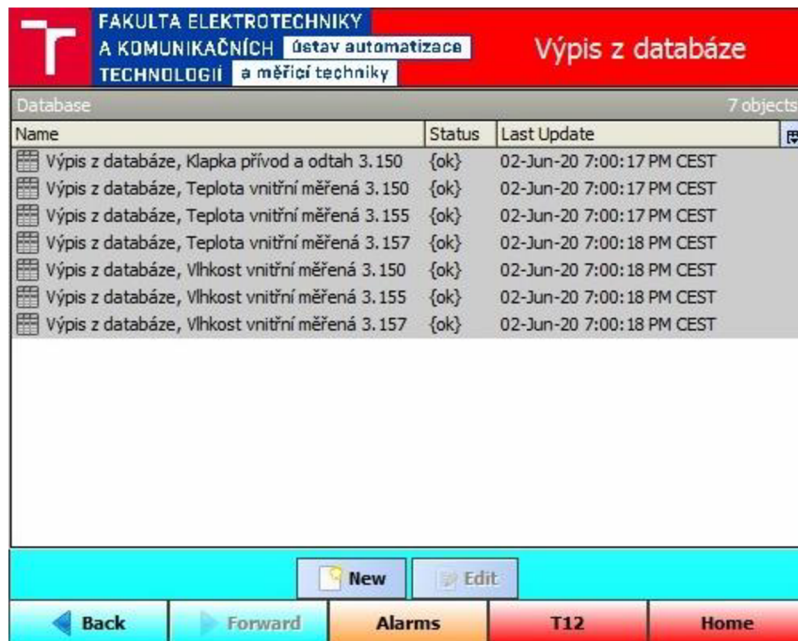
Po přidání rozšíření alarmů poruchovým proměnným lze přesunout prvek *ConsoleRecipient* na list vizualizace a tím přidat výpis alarmů do uživatelsky přívětivějšího prostředí. I tady je možné alarmy odsouhlasit a zobrazit si jejich detaily.

Open Alarm Sources				14 Sources / 16 Alarms	
Timestamp	Source State	Ack State	Source		
02-Jun-20 10:28:25 AM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
02-Jun-20 10:28:24 AM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
02-Jun-20 10:28:23 AM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
02-Jun-20 10:27:24 AM CEST	Normal	0 Aked / 3 Unacked	true {alarm,o		
26-May-20 9:23:06 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:23:04 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:23:03 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:23:02 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:23:00 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:22:59 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:22:58 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		
26-May-20 9:22:53 PM CEST	Normal	0 Aked / 1 Unacked	true {alarm,o		

Obrázek 6.8: Vizualizace výpisu alarmů

6.8 Výpis z databáze

Na tomto náhledu jsou vypsané všechny proměnné, které jsou vypisovány z databáze. Předem pro ně byly vytvořeny datové body. Postup je uveden v kapitole 5.7. Po rozkliknutí proměnné je možné si zobrazit i její data. Pro uživatele je tento náhled taky zjednodušenou cestou pro vytvoření nového datového bodu, který lze vytvořit kliknutím na tlačítko *New*.

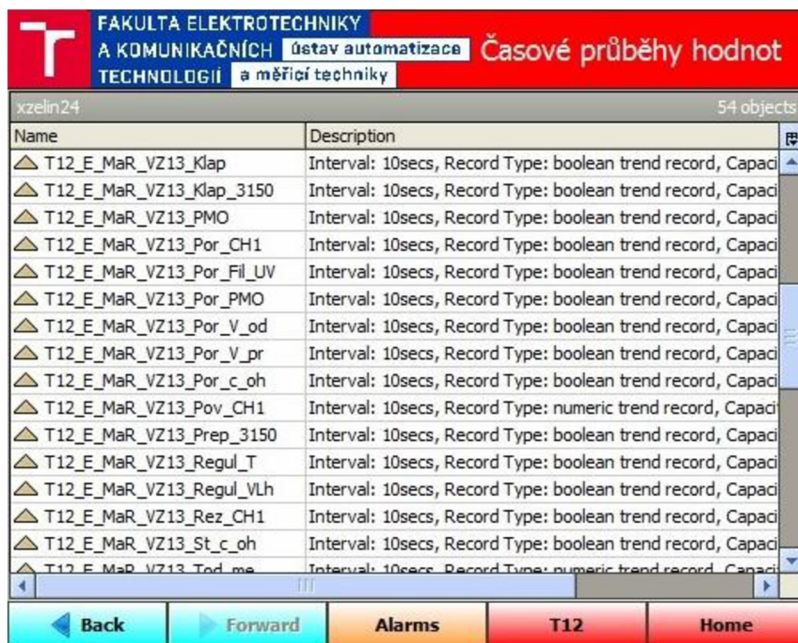


Name	Status	Last Update
Výpis z databáze, Klapka přívod a odtah 3.150	{ok}	02-Jun-20 7:00:17 PM CEST
Výpis z databáze, Teplota vnitřní měřená 3.150	{ok}	02-Jun-20 7:00:17 PM CEST
Výpis z databáze, Teplota vnitřní měřená 3.155	{ok}	02-Jun-20 7:00:17 PM CEST
Výpis z databáze, Teplota vnitřní měřená 3.157	{ok}	02-Jun-20 7:00:18 PM CEST
Výpis z databáze, Vlhkost vnitřní měřená 3.150	{ok}	02-Jun-20 7:00:18 PM CEST
Výpis z databáze, Vlhkost vnitřní měřená 3.155	{ok}	02-Jun-20 7:00:18 PM CEST
Výpis z databáze, Vlhkost vnitřní měřená 3.157	{ok}	02-Jun-20 7:00:18 PM CEST

Obrázek 6.9: Náhled proměnných vypsaných z databáze

6.9 Zobrazení proměnných a jejich časových průběhů

Na tomto listu vizualizace jsou vypsány všechny proměnné, které mají rozšíření historie. Po rozkliknutí libovolné z nich si lze zobrazit její časový průběh.



Name	Description
△ T12_E_MaR_VZ13_Klap	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Klap_3150	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_PMO	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Por_CH1	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Por_Fil_UV	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Por_PMO	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Por_V_od	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Por_V_pr	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Por_c_oh	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Pov_CH1	Interval: 10secs, Record Type: numeric trend record, Capaci
△ T12_E_MaR_VZ13_Prep_3150	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Regul_T	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Regul_VLh	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Rez_CH1	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_St_c_oh	Interval: 10secs, Record Type: boolean trend record, Capaci
△ T12_E_MaR_VZ13_Tod_me	Interval: 10secs, Record Type: numeric trend record, Capaci

Obrázek 6.10: Soupis proměnných s rozšířením historie

7 ZÁVĚR

V této bakalářské práci byl vysvětlen pojem inteligentní budova, komunikace zařízení v těchto budovách, data, které poskytují a jak je sbírat.

Část byla věnována i databázím, které mohou sloužit pro ukládání dat potřebných pro měření a regulaci v inteligentních budovách. Jedna kapitola byla věnována jazyku SQL, se kterým lze ve většině databází pracovat, datovým typům tohoto jazyku a nejpoužívanějším příkazům. Zmíněn byl také vytvořený databázový model školy vytvořený panem Bc. Matoušem Rathouzkým.

Bylo vysvětleno základní používání programu Coach^{AX} v režimu práce s jednotlivými bloky i v grafickém režimu.

V kapitole simulace dat budovy E byly představeny všechny vybrané proměnné vzduchotechniky budovy T12 bloku E. Proměnné byly připraveny na zápis do databáze a pomocí vytvořeného SQL dotazu byly data rozříděny do databázového modelu. Další SQL dotaz byl použit pro přehledné zobrazení zapsaných dat jak v programu Microsoft SQL server, tak v programu Coach^{AX}.

Nakonec byla vytvořena vizualizace těchto dat. Vizualizace je interaktivní, takže uživatel může zasáhnout do toho, co se ve výsledku zobrazí. Zprovozněn byl i soupis alarmů, který upozorňuje na kritické i nekritické situace, které mohou nastat. Byl vytvořen i grafický model části vzduchotechniky.

Při psaní této práce byla vynaložena snaha o její pochopitelnost i pro čtenáře nezabývající se touto problematikou. Naopak pro čtenáře s určitými znalostmi o elektrotechnice, inteligentních budovách nebo databázových systémech práce nabízí rozšíření jejich obzorů za možné hranice jejich zájmů. Objevují se i pasáže výukového charakteru, například základy jazyka SQL nebo práce v programu Coach^{AX} užitečné pro studenty, kteří hodlají v práci pokračovat nebo se v této oblasti vzdělat.

V tomto stavu práce splňuje zadání. Její zpracování nebylo příliš náročné, ale ani příliš jednoduché. To bylo způsobeno hlavně využíváním programu Coach^{AX}, který sice nabízí nespočet funkcí a zobrazení, ale bez kvalitní podpory je v něm těžké se orientovat. Oproti ostatním pracím bylo také nevýhodou, že byl program nainstalován pouze na jednom počítači v laboratoři univerzity a neexistovaly pro něj studentské licence pro používání doma. Ulehčení přinesl ironicky krizový stav v České republice, kdy nebylo možné navštěvovat školu, a tak byl zřízen vzdálený přístup k počítači z domova. Práce tak mohla být dokončena v zadaném termínu.

Zpracování práce mi umožnilo se naučit novým věcem, které bych se při studiu nejspíš nedozvěděl a pevně věřím, že tyto znalosti využiji při dalším studiu a při mém budoucím povolání.

Literatura

- [1] *Perspektivy bydlení: příloha časopisů ELEKTRO a AUTOMA* [online]. Praha: FCC Public, 2012, **2012**(10) [cit. 2020-01-04]. Dostupné z: http://www.odbornecasopisy.cz//flipviewer/Specialy/05/Special_05_output/web/Special_05_opf_files/WebSearch/page0001.html
- [2] MERZ, Hermann, Thomas HANSEMANN a Christof HÜBNER. *Automatizované systémy budov: sdělovací systémy KNX/EIB, LON a BACnet*. Praha: Grada, 2008. Stavitel. ISBN 978-80-247-2367-9.
- [3] Future office. *INELS – chytré elektroinstalace* [online]. Holešov: ELKO ep, c2017-2018 [cit. 2020-01-04]. Dostupné z: <https://www.inels.cz/future-office>
- [4] Automatizace budov: Automatizace budov – význam a funkce. *BUILDSYS* [online]. Brno: BUILDSYS, 2015 [cit. 2020-01-04]. Dostupné z: <http://www.buildsys.cz/buildsys-systemy-pro-rizeni-budov-automatizace-budov.html>
- [5] Vzduchotechnika: Kvalitní vzduchotechnika zajistí přísun čerstvého vzduchu. *TZB PRO s.r.o.* [online]. Praha: TZB PRO, 2020 [cit. 2020-01-04]. Dostupné z: <https://tzb-pro.cz/vzduchotechnika>
- [6] Air Handling Units Explained: How do Air Handling Units work? *The Engineering Mindset* [online]. The Engineering Mindset, 2018, 26. 9. 2018 [cit. 2020-01-04]. Dostupné z: <https://theengineeringmindset.com/air-handling-units-explained/>
- [7] MARTZ, Václav. Systémy používané v "inteligentních" budovách – přehled komunikačních protokolů. *Tzb-info.cz* [online]. Praha: Ing. Václav Matz, Ph.D., 2010, 25. 10. 2010 [cit. 2020-01-05]. Dostupné z: <https://vytapani.tzb-info.cz/mereni-a-regulace/6879-systemy-pouzivane-v-inteligentnich-budovach-prehled-komunikacnich-protokolu>
- [8] PIVOŇKOVÁ, Alena. *Optimalizační algoritmy řídicích systémů inteligentních budov* [online]. Praha, 2005 [cit. 2020-01-05]. Dostupné z: https://support.dce.felk.cvut.cz/mediawiki/images/4/47/Dp_2005_pivonkova_alena.pdf Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra řídicí techniky. Vedoucí práce Ing. Jaroslav Honců.
- [9] VOJÁČEK, Antonín. Úvod do BACnetu – Building Automation and Controls Network. *Automatizace.hw.cz: rady a poslední novinky z oboru* [online]. Praha: Antonín Vojáček, 2012, 26. dubna 2012 [cit. 2020-01-05]. Dostupné z: <https://automatizace.hw.cz/uvod-do-bacnetu-building-automation-and-controls-network>
- [10] BACnet™ – A standard communication infrastructure for intelligent buildings [online]. In.: [cit. 2018-01-02]. Dostupné z: <http://www.bacnet.org/Bibliography/AIC-97/AIC1997.htm>
- [11] Sběr dat pro vyhodnocování spotřeb energie klimatizačních jednotek. *Www.asb-portal.cz* [online]. Praha: ASB portál, 2010, 21. Květen 2010 [cit. 2020-01-05]. Dostupné z: <https://www.asb-portal.cz/stavebnictvi/technicka-zarizeni-budov/vetrani-a-klimatizace/sber-dat-pro-vyhodnocovani-spotreb-energie-klimatizacnich-jednotek>

- [12] Databáze: Co je relační databáze? *Oracle* [online]. Praha: Oracle, c2019 [cit. 2020-01-05]. Dostupné z: <https://www.oracle.com/cz/database/what-is-database.html#WhatIsDBMS>
- [13] Relační databáze. *Greendot.cz* [online]. Praha: Štěpánek, 2009, 14. 1. 2009 [cit. 2020-01-05]. Dostupné z: <http://vyuka.greendot.cz/materialy/material-4.pdf>
- [14] Microsoft Access Vs. SQL Server. *Techwalla* [online]. Techwalla, c2020 [cit. 2020-01-05]. Dostupné z: <https://www.techwalla.com/articles/microsoft-access-vs-sql-server>
- [15] SQL server. *www.microsoft.com* [online]. Praha: Microsoft, c2020 [cit. 2020-01-05]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2017-pricing>
- [16] RATHOUZSKÝ, M. *Sběr MaR dat*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 46 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.
- [17] HONEYWELL. *CentraLine^{AX}: Cap. 0 - Introduce^{AX}*. Richmond, 2015. Dostupné také z: Příloha č. 1/Cap0_IntroduceAX.pdf
- [18] HONEYWELL. *Arena Ax (Level 2) Training: Creating a Web Supervisor and Graphics Pages*. Richmond, 2013. Dostupné také z: Příloha č. 1/Cap4_3_7_ArenaAX_Lev_2_Training_WI_Version_1_00.pdf
- [19] Technická 12 - plánec. *Vutbr.cz* [online]. Brno: VUT, c2020, 12. 2. 2013 [cit. 2020-01-05]. Dostupné z: https://www.fekt.vut.cz/dokumenty/plany/technicka12_planec.pdf
- [20] LACKO, Ľuboslav. *SQL: kapesní přehled*. Brno: CP Books, 2005. ISBN 80-251-0788-4.
- [21] *W3schools: SQL tutorial* [online]. c1999-2020 [cit. 2020-06-03]. Dostupné z: <https://www.w3schools.com/sql/default.asp>
- [22] Laboratoř genomiky a proteomiky: biochemická laboratoř pro práci s buněčnými kulturami a genetickým materiálem. *Ústav biomedicínského inženýrství, FEKT VUT* [online]. Brno: VUT v Brně, c2020 [cit. 2020-06-03]. Dostupné z: <https://www.ubmi.feec.vutbr.cz/vybaveni/biochemicka-laborator-pro-praci-s-bunecnymi-kulturami-a-genetickym-materialem/>

Seznam příloh

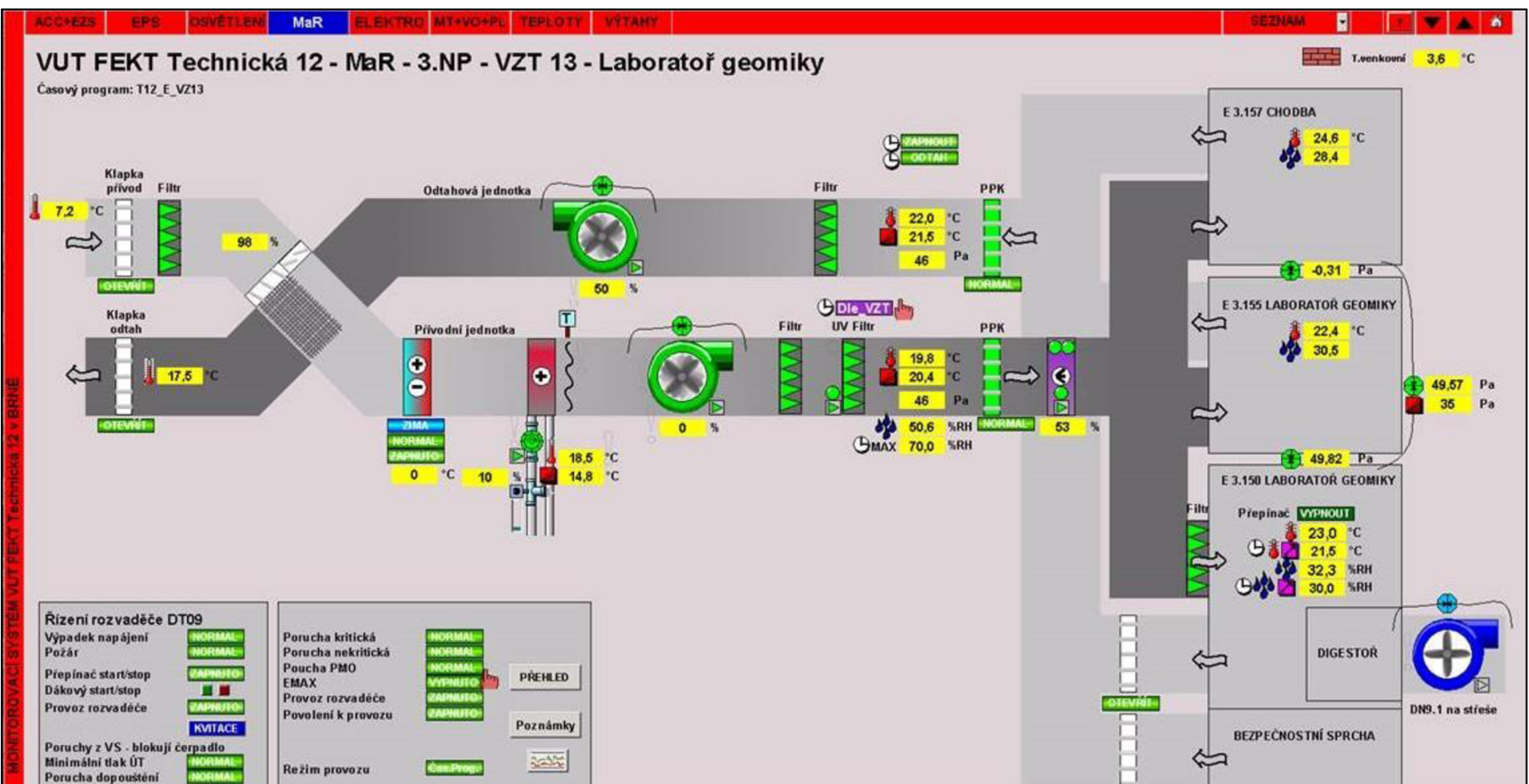
Příloha č. 1: Model vzduchotechniky VZ13 určený pro řízení z vrátnice budovy T12

Příloha č. 2: Zobrazení simulace ve *Wire Sheet*

Příloha č. 3: ZIP soubor, který obsahuje:

- Použité tréninkové materiály k programu Coach^{AX}
- T12 MaR – Soupis sbíraných MaR dat budovy T12
- Snímky obrazovky v programu CoachAX a MS SQL serveru
- Textový soubor s užitečnými odkazy na některý výuková videa
- Textový soubor se soupisem všech použitých proměnných včetně jejich definicí

Příloha č. 1 – Model vzduchotechniky VZ13 určený pro řízení z vrátnice budovy T12



Příloha č. 2 – Zobrazení simulace ve Wire Sheet

