

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra kybernetiky

# BAKALÁŘSKÁ PRÁCE

Plzeň, 2018

Lukáš Soukup

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra kybernetiky

## Bakalářská práce

Modelování vybraných energetických zařízení

Plzeň, 2018

Vedoucí: Ing. Martin Střelec, Ph.D.  
Vypracoval: Lukáš Soukup - A15B0552P

## PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne *18.5.2018*

.....  
*vlastnoruční podpis*

## **Poděkování**

Chtěl bych upřímně poděkovat zejména vedoucímu mé bakalářské práce Ing. Martinu Střelcovi, Ph.D. za jeho vedení, ochotu a trpělivost při konzultacích i mimo ně. Dále bych rád poděkoval mému kolegovi Pavlu Novotnému za spolupráci na společných částech našich bakalářských prací a za cenné rady v ostatních částech. V neposlední řadě bych chtěl poděkovat mým rodičům za jejich psychickou i finanční podporu po celou dobu mého studia.

## Abstrakt

Cílem bakalářské práce je vytvoření společného systému pro zpracování a uložení energetických dat měřených v areálu Fakulty aplikovaných věd Západočeské univerzity v Plzni a vytvoření energetických analytics pracujících nad tímto systémem.

V první části jsou analyzovány a zpřístupněny v současnosti používané datové zdroje (Energy management system a Telemetrická soustava). Dále byly tyto zdroje spojeny s nově vytvořenou konsolidovanou databází a byly vytvořeny softwarové nástroje pro získání dat z původních datových zdrojů. Pro potřeby navazujících aplikací byl vytvořen softwarový nástroj pro standardizaci naměřených dat, který pracuje nad společným datovým úložištěm.

V poslední části bylo zvoleno vhodné energetické zařízení pro tvorbu energetické analytické služby (analytics) - absorpční chladicí stroj. Vytvořené analytics slouží k ukázce možností a využitelnosti vytvořených softwarových nástrojů a pro analýzu energetické náročnosti chladicího zařízení, která umožní snížení provozních nákladů nebo diagnostiku daného energetického zařízení.

## Klíčová slova

databáze, energetická data, Telemetrická soustava, Energy Management System, softwarové technologie - Java, SQL, Matlab, standardizace dat, agregace, interpolace, energetická síť, energetický model

## Abstract

The bachelor's thesis focuses on the creation of system for processing and storing energy data measured in the campus of Faculty of Applied Science of University of West Bohemia in Pilsen and the development of energy analytics working above this system. In the first part, currently used data sources (Energy management system and Telemetry system) were analysed and opened up. Further the connection between these data sources and newly developed consolidated database were established by development of specialized software tools for data transmission. For need of foreseen applications, the software tool for standardization of measured data was created, which operates above common data source. In the last part, appropriate energy device (absorption chiller ) was chosen for making analytical services. Developed analytics serve to show possibilities and utilization possibilities of developed software tools for analyzing energy of absorption chiller, which enables optimization of operating costs or diagnostic of given energy device.

## **Key words**

database, energy data, Telemetry system, Energy Managment system, software technologies - Java, SQL, Matlab, standardization of data, aggregation, interpolation, energy grid, energy model

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
1.1	Motivace . . . . .	7
1.2	Vymezení obsahu práce . . . . .	8
1.3	Použité technologie . . . . .	9
1.3.1	Java . . . . .	9
1.3.2	SQL . . . . .	9
1.3.3	Matlab . . . . .	10
<b>2</b>	<b>Systémy pro sběr a uložení dat nasazené na ZČU</b>	<b>11</b>
2.1	Přehled nasazených systémů . . . . .	11
2.2	Detailní popis telemetrické soustavy . . . . .	11
2.3	Zpřístupnění datového zdroje . . . . .	12
2.3.1	Konsolidovaná databáze . . . . .	12
2.3.2	Analýza datového zdroje . . . . .	14
2.3.3	Anotace datových bodů . . . . .	14
2.3.4	Zpřístupnění datového zdroje . . . . .	16
2.4	Rozhraní pro získání energetických dat . . . . .	20
2.4.1	Popis struktury data acces layeru . . . . .	20
2.4.2	Popis funkčnosti data acces layeru . . . . .	24
2.4.3	Ukázka použití data acces layeru z Matlabu . . . . .	27
<b>3</b>	<b>Model vybraného energetického zařízení</b>	<b>29</b>
3.1	Princip absorpčního chlazení . . . . .	30
3.2	Adapted Gordon - Ng model (GN) . . . . .	32

3.3	Multivariate polynomial regression model (MPR) . . . . .	33
3.4	Porovnání modelů . . . . .	33
3.4.1	RMSE kritérium . . . . .	34
3.4.2	MAE kritérium . . . . .	34
3.4.3	Vlastnosti RMSE a MAE kritéria . . . . .	34
3.4.4	Porovnání přesnosti modelů . . . . .	35
<b>4</b>	<b>Závěr</b>	<b>37</b>



# Kapitola 1

## Úvod

### 1.1 Motivace

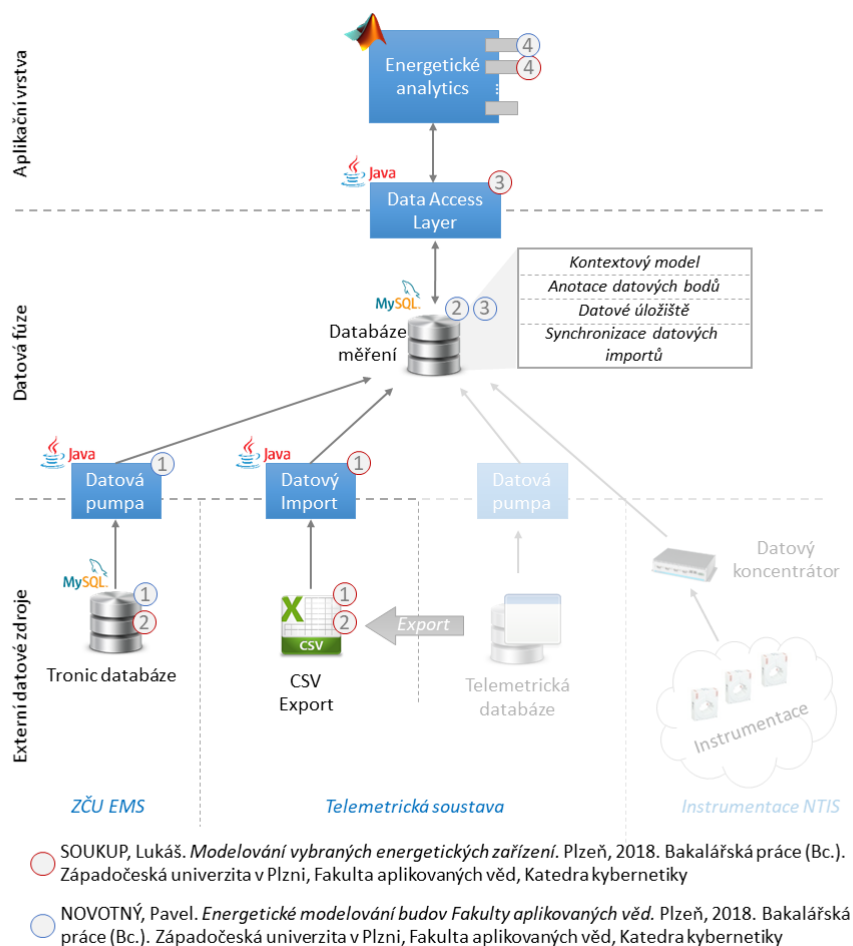
Tato bakalářská práce se zabývá analýzou energetických dat měřených v areálu Fakulty aplikovaných věd Západočeské univerzity v Plzni a následným vývojem nástrojů pro práci s těmito daty. Hlavní motivací pro tuto práci je potřeba monitorování energií, spotřeby a efektivnosti energetických zařízení pro provozní účely. Pomocí vhodného monitoringu energetických zařízení je možné zkoumat energetickou náročnost a účinnost těchto zařízení pro studijní, výzkumné i provozní účely. Cílem bakalářské práce je vytvoření speciálních energetických analytics (tj. modelů vybraných energetických zařízení) založených na nově vyvinutém systému pro sběr a zpracování dat v areálu Fakulty aplikovaných věd ZČU. Výsledný monitorovací systém usnadní práci technických pracovníků oddělení provozu a služeb ZČU a bude poskytnut pro výzkumné a studijní účely.

Cílem této bakalářské práce je zlepšit stávající stav, kdy jsou paralelně provozovány dva oddělené systémy pro sběr a práci s energetickými daty. Telemetrická soustava a EMS (energy management system) obsahují komplementární datové sady z areálu kampusu ZČU. Separátní systémy poskytují omezené možnosti pro výzkumné i technické účely a limitují jejich použití pro praktické využití při efektivním monitoringu a výzkumu zefektivnění práce energetických zařízení.

Činnosti spojené s realizací bakalářské práce byly rozděleny do ucelených oblastí, které pokrývaly (i) analýzu a zpřístupnění dostupných datových zdrojů ZČU obsahující měřená data energetických zařízení, (ii) analýzu a zvolení vhodných energetických zařízení pro modelování, (iii) návrh a implementaci vhodného rozhraní pro získání dat potřebných pro modelování vybraných energetických zařízení a (iv) vytvoření a ověření modelů vybraných energetických zařízení.

## 1.2 Vymezení obsahu práce

Vytvoření uceleného systému pro sběr energetických dat, jejich zpracování a následné využití je velice rozsáhlé téma, které by nebylo možno pokrýt jednou bakalářskou prací. Z tohoto důvodu byl celý problém rozdělen na dvě bakalářské práce. Rozdělení jednotlivých aktivit je vidět na schématu systému sběru dat 1.1. Tato bakalářská práce zahrnuje činnosti označené červenou barvou.



Obrázek 1.1: Schéma systému pro sběr energetických dat

Systém pro sběr energetických dat a jejich zpracování lze rozdělit do tří úrovní - *Externí datové zdroje, datová fúze a aplikační vrstva*.

V nejnižší vrstvě nazvané *Externí datové zdroje* je třeba vyřešit přístup k datovým zdrojům třetích stran, které slouží k primárnímu sběru dat z instalovaných měřících systémů (kapitola 2). V této bakalářské práci je řešena problematika zisku naměřených dat z telemetrické soustavy. Protože nebylo možno technicky realizovat přímý přístup do telemetrické databáze, byla zpracovávána offline data v podobě exportovaných csv souborů. Z nejnižší datové vrstvy jsou naměřená data přenášena do konsolidovaného datového úložiště pomocí datové pumpy. Řešení přenosu dat je více rozvedeno v kapitole 2.

Prostřední vrstva navazující na externí datové zdroje se zabývá *Datovou fúzí*. V této části jsou data z externích datových zdrojů zpracovávána do konsolidované databáze měření.

Nejvyšší oblastí je *Aplikační vrstva*, která obsahuje koncové aplikace pracující nad daty z databáze měření. Přechod z vrstvy datové fúze je řešen prostřednictvím *Data acces layeru*, pomocí kterého jsou získávána očištěná data pro potřeby energetických analytics. Problematika řešení data acces layeru je blíže popsána v sekci 2.4. Energetický model vybraného energetického zařízení představuje ukázkové energetické analytics, které bylo vytvořeno v rámci této bakalářské práce (kapitola 3).

## 1.3 Použité technologie

V této kapitole budou popsány softwarové technologie využité při řešení bakalářské práce. Vzhledem k rozsahu a heterogenitě jednotlivých činností bylo potřeba využít různých softwarových technologií. V každé části bakalářské práce byla využita technologie vhodná pro daný účel.

### 1.3.1 Java

Java je objektově orientovaný, interpretovaný jazyk, který byl vyvinut firmou Sun Microsystems roku 1995. Jedná se o jeden z nejpoužívanějších programovacích jazyků na světě. Velká popularita je způsobena zejména snadnou přenositelností, nezávislostí na architektuře a velkým počtem volně přístupných knihoven [3].

Výhodami jazyka Java oproti nižším programovacím jazykům, jako je například C/C++, jsou jeho jednoduchost a správa paměti (Java obsahuje tzv. garbage collector). Výhody pro použití v této bakalářské práci:

- Pokročilé nástroje pro získání a zpracování dat
- Jednoduchá práce s databází
- Přenositelnost výsledného kódu

### 1.3.2 SQL

SQL (Structured Query Language - standardizovaný strukturovaný dotazovací jazyk) je dotazovací jazyk používaný pro práci s relačními databázemi. Vznikl na základě potřeby vytvořit sadu příkazů pro popis a práci s relačními databázemi s důrazem na podobnost přirozenému jazyku (angličtině). SQL se dělí na čtyři základní části [2]:

- **DML - Data manipulation language** - příkazy pro manipulaci s daty. Příklady příkazů:

- SELECT - výběr dat
  - INSERT - vkládání dat
  - UPDATE - změna dat
  - ALTER - odstranění dat
- **DDL - Data definition language** - příkazy pro definici dat. Příklady příkazů:
    - CREATE - vytvoření nových objektů (databáze, tabulky,..)
    - ALTER - změna objektů
    - DROP - odstranění objektů
  - **DCL - Data control language** - příkazy pro správu přístupových práv a rolí. Příklady příkazů:
    - GRANT - přidělení přístupových práv
    - REVOKE - odebrání přístupových práv
  - **TDL - transactional control language** - příkazy pro správu databázových transakcí. Příklady příkazů:
    - BEGIN (START TRANSACTION) - zahájení databázové transakce
    - COMMIT - ukončení databázové transakce a uložení jejích výsledků
    - ROLLBACK - operace, která vrací databázi do předchozího stavu před transakcí.

### 1.3.3 Matlab

Matlab (**M**atrix **l**aboratory) je interaktivní programové prostředí a multiplatformní skriptovací programovací jazyk. Program je vyvíjen společností MathWorks. V září 2017 vyšla zatím poslední verze R2017b. Tento programovací jazyk byl původně vyvinut pro matematické výpočty a operace. Postupem času se však rozrostl a nyní obsahuje moduly a rozšíření například pro simulace mechanických soustav, biologických procesů, statistická měření, 3D grafy a mnoho dalších. Pro potřeby kybernetiky a simulace dynamických systémů je hojně využívána knihovna Simulink [6].

# Kapitola 2

## Systemy pro sběr a uložení dat nasazené na ZČU

### 2.1 Přehled nasazených systémů

Areál kampusu Západočeské univerzity v Plzni se skládá ze čtyř fakult, dvou výzkumných center, administrativních budov a knihovny. Denní zatížení elektrické sítě je velice heterogenní, což je způsobeno zejména specializovanými laboratořemi s velice nárazovým odběrem elektřiny. Průměrné zatížení celého komplexu se v běžný pracovní den pohybuje okolo 1 MW. Zvláště odběr vzduchotechnického systému (resp. HVAC), který zahrnuje například topení, ventilátory nebo provoz klimatizace, se může pohybovat až okolo 680 kW.

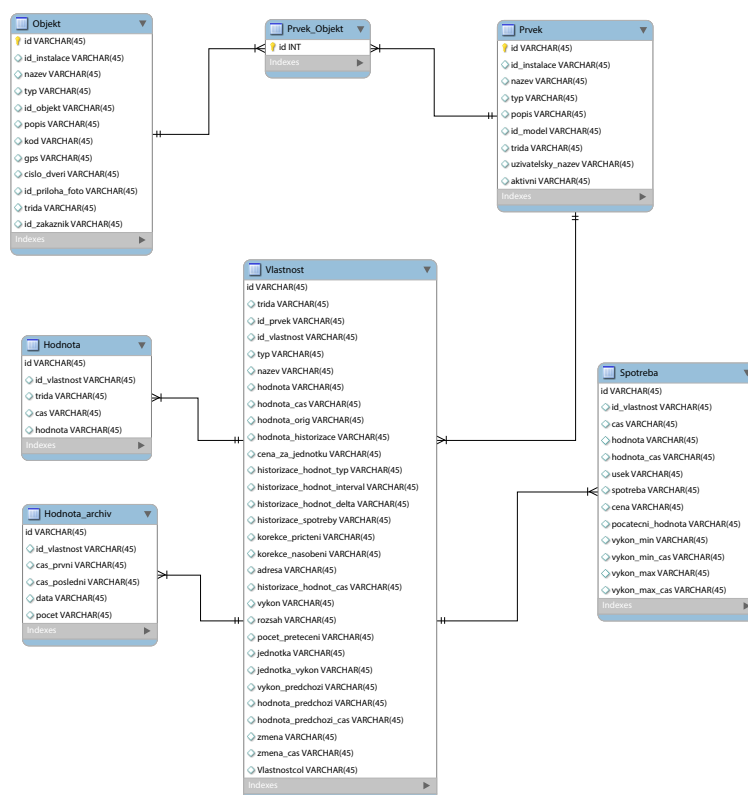
Pro sběr a uložení energetických dat v celém areálu Západočeské univerzity jsou využívány dva oddělené systémy. EMS (Energy Management System) se používá pro monitoring transformace a distribuce toků energie v rámci areálu ZČU. Systém telemetrie se používá pro fakturaci, vyúčtování energií a monitorování poruch. Rozdělení a nemožnost spolupráce těchto dvou oddělených systémů způsobuje komplikace při monitorování a řízení energetických toků na Západočeské univerzitě. Pro celkový přehled o spotřebě energií je potřeba tyto dva systémy sjednotit do jednoho, nad kterým bude mít univerzita plnou kontrolu.

### 2.2 Detailní popis telemetrické soustavy

Telemetrická soustava byla vybudována pro potřeby monitorování spotřeb energií a detekci poruchových stavů. Systém se skládá ze sensorové sítě, která je určena pro primární sběr dat z technologických procesů. Tato data jsou bezdrátově přenášena do datových koncentrátorů a dále zasílána na cloudové úložiště spravované externí společností. Na cloudovém úložišti je umístěna databáze pro uchování naměřených hodnot a informací o všech měřených signálech v areálu ZČU.

V rámci smluvních vztahů jsou periodicky z databáze exportována naměřená data

v podobě CSV souborů. Následující obrázek 2.1 graficky znázorňuje strukturu exportovaných dat.



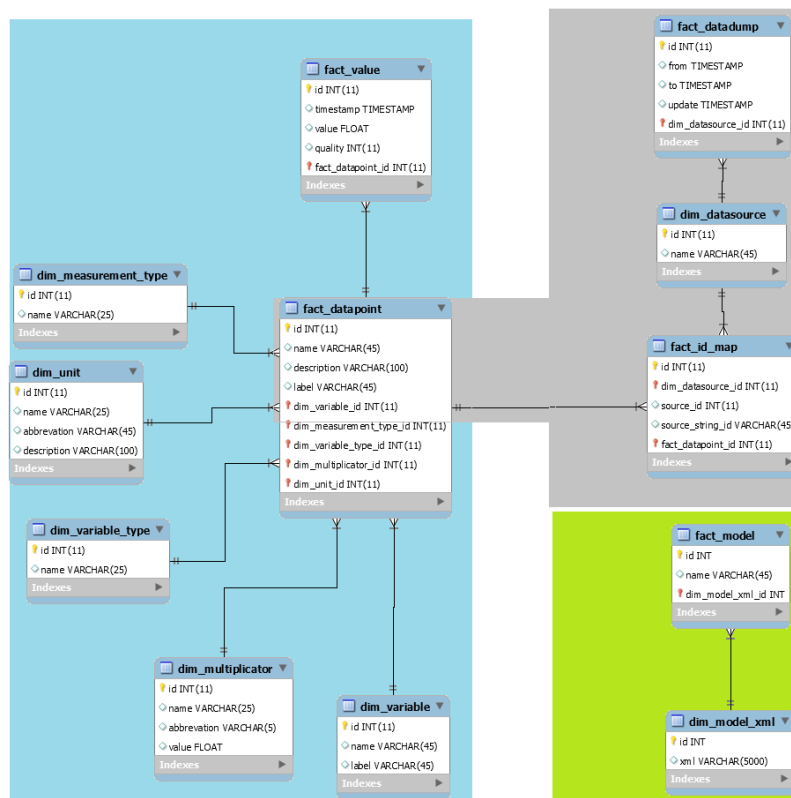
Obrázek 2.1: Struktura exportovaných CSV

Informace o jednotlivých datových bodech jsou uloženy v tabulce *Vlastnost*. Na obsáhlejší vlastnosti datového bodu, které jsou realizovány vlastními tabulkami, je zde cizí klíč s odkazem na tento záznam. Jedna ze speciálních vlastností datového bodu je v tabulce *Spotreba*, kam je se ukládají samotná naměřená data. Ke cloudovému úložišti je možný online přístup pomocí aplikace Flexim.

## 2.3 Zpřístupnění datového zdroje

### 2.3.1 Konsolidovaná databáze

Při realizaci bakalářské práce [7] byla vytvořena jednotná databáze konsolidující měřená energetická data z různých datových zdrojů. Na následujícím obrázku 2.2 můžeme vidět strukturu jednotné databáze.



Obrázek 2.2: Struktura konsolidované databáze

Strukturu databáze je možno rozdělit na tři části podle jejich účelu a vzájemných vztahů mezi tabulkami.

Modrá část na levé straně je zaměřena na uchování měřených hodnot a sestává se z dimenzionálních tabulek, které obsahují informace například o jednotkách, typu veličiny, proměnné a další. Tyto informace se využívají při vytváření a zapisování konkrétních datových bodů. Tato část dále obsahuje faktické tabulky s jednotlivými datovými body (tabulka *fact\_datapoint*) a všemi záznamy o naměřených hodnotách (tabulka *fact\_value*). Mezi tabulkami *fact\_datapoint* a *fact\_value* je relační vazba typu 1:N, kdy k jednomu datovému bodu existuje N naměřených hodnot.

Šedá část schématu označuje takzvanou dumpovou část databáze. V této části jsou uloženy informace o vkládání dat do databáze. Z informací v tabulce *fact\_datadump* lze určit aktuálnost dat uložených v databázi a je nezbytná pro vkládání dalších dat. V této části se také nachází tabulka *fact\_id\_map*, která obsahuje namapování datových bodů na své původní zdroje. Poslední tabulka této části je *dim\_datasource*. V této tabulce jsou informace o původu datového bodu v databázi.

Zelená část je částečně oddělena od zbytku a je věnována segmentu databáze věnující se kontextovým modelům.

### 2.3.2 Analýza datového zdroje

Vzhledem k omezeným možnostem přístupu na cloudové úložiště nebylo možné zpracovávat měřená data přímo, ale při řešení této bakalářské práce byly zpracovávány příslušné datové exporty. Pro získání dat z původních datových úložišť (resp. CSV souborů) a uložení do konsolidované databáze (viz 2.3.1) měření bylo nutno provést následující kroky.

Nejprve bylo potřeba analyzovat datové body obsažené v telemetrické soustavě a provést jejich anotaci pro potřeby konsolidovaného datového úložiště.

V budově Fakulty aplikovaných věd bylo manuálně vybráno přibližně 40 datových bodů, které byly následně anotovány a uloženy do excelovské tabulky. Anotované datové body v cílovém úložišti mohou být následně využity například pro potřeby kontextového modelování či data miningu.

Externím dodavatelem byla poskytnuta naměřená data z posledních několika let ve formě csv souborů. Jednotlivé csv soubory odpovídají dumpům jednotlivých tabulek telemetrické databáze, kterými jsou:

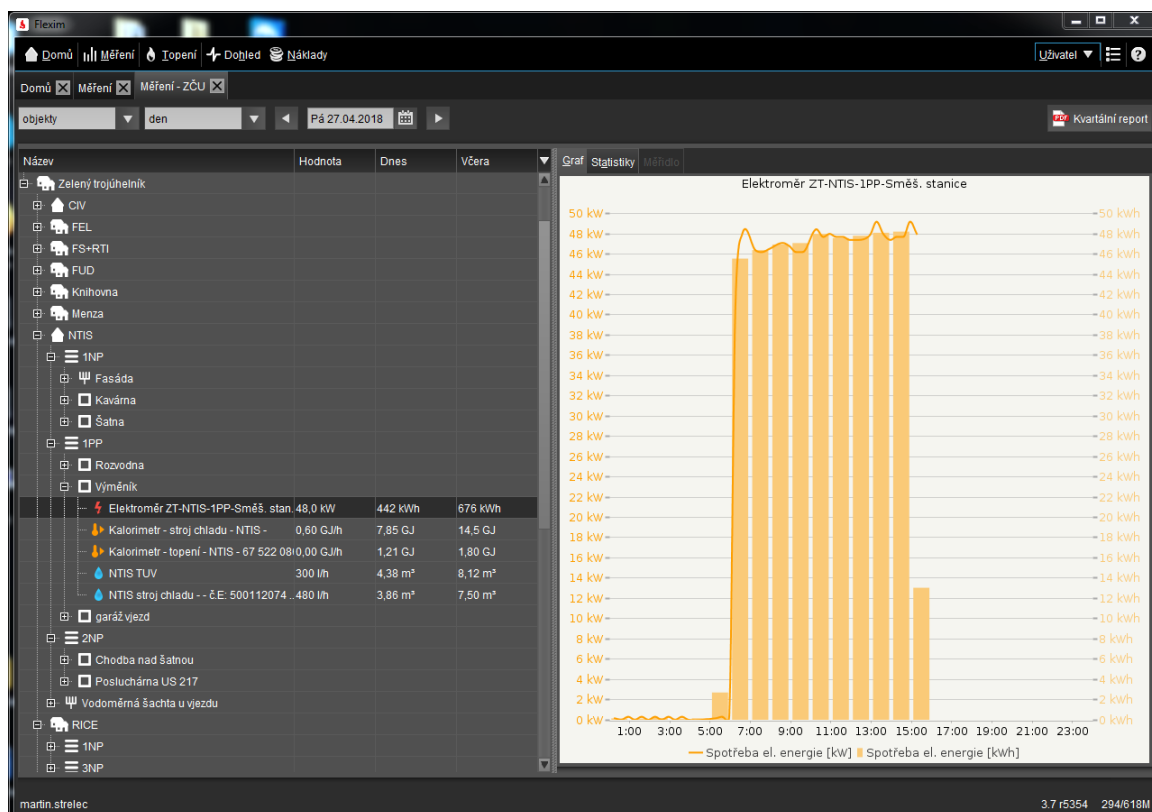
- **Objekt** - obsahuje informace o umístění měřicího zařízení jako například budova, patro, číslo dveří apod.
- **Prvek** - obsahuje informace o měřícím senzoru (např. elektroměr-chodba, teplo- měř-fasáda apod.)
- **Prvek\_objekt** - vazební tabulka mezi tabulkami Prvek a Objekt
- **Vlastnost** - obsahuje velké množství informací o jednotlivých vlastnostech měřeného signálu
- **Spotřeba** - obsahuje informace o spotřebě za daný časový úsek
- **Hodnota** - obsahuje informace o každé naměřené hodnotě ze senzoru
- **Hodnota\_archiv** - obsahuje informace o archivních záznamech (např. začátek a konec měřicího období, počet záznamů za toto období apod.)

Vzájemné vztahy jednotlivých tabulek jsou zobrazeny na obrázku 2.1.

### 2.3.3 Anotace datových bodů

Ze strany dodavatele systému byla ZČU poskytnuta aplikace Flexim, která zpřístupňuje datové úložiště uživatelům (tj. pracovníkům oddělení provozu a služeb ZČU). Na obrázku 2.3 lze vidět ukázkou obrazovky z dodané aplikace Flexim.





Obrázek 2.3: Ukázka aplikace Flexim

V levé části okna je pomocí stromové struktury zachyceno umístění datového bodu v hierarchické struktuře. U každého datového bodu je zde informace o jeho pojmenování, měřené veličině a jednotce měřené veličiny. Po výběru datového bodu a časového období se v pravé části okna zobrazí graf naměřených dat za dané období.

Pro potřeby pozdějšího využití v konsolidované databázi bylo třeba analyzovat datové body z telemetrické soustavy. Na základě informací získaných z aplikace Flexim byla vytvořena excelovská tabulka popisující datové body v areálu budovy Fakulty aplikovaných věd ZČU. Atributy jednotlivých datových zdrojů jsou:

- **id** - numerické identifikační číslo
- **name** - intuitivní pojmenování datového bodu
- **description** - obecný popis datového zdroje
- **path** - umístění datového zdroje
- **sampling period** - vzorkovací perioda měření
- **unit** - jednotka měřící hodnoty
- **multiplicator** - násobící konstanta
- **variable** - měřená veličina (výkon, teplota, průtok apod.)

- **variable type** - typ měřené proměnné (měřená, řídicí, virtuální)
- **medium** - médium měřené veličiny (voda, vzduch apod.)
- **sampling type** - typ měření veličiny (průběhově, rozdílově)

Anotace datových bodů byly uloženy do excelového souboru. Na obrázku 2.4 je ukázka anotovaného souboru *registr-telemetry*.

ID	name	description	Path	sampling period [min]	unit	multiplicator	variable	variable type	unit	medium	sampling type
1	Elektroměr ZI-NTIS-Garáž-rozvodna-NTIS		Zelený trojúhelník\NTIS\1PP\Rožvodna	-	kW	1	výkon	merena	kW	elektřina	prubezne
2	Venkovní teploměr		Zelený trojúhelník\NTIS\1NP\Fasáda	-	°C	1	teplota	merena	°C	vzduch	prubezne
3	Kavárna		Zelený trojúhelník\NTIS\1NP\Kavárna	-	kW	1	výkon	merena	kW	elektřina	prubezne
4	Podružný vodoměr - SV		Zelený trojúhelník\NTIS\1NP\Kavárna	-	m <sup>3</sup> /h	1	průtok	merena	m <sup>3</sup> /h	voda	prubezne
5	Podružný vodoměr - TV		Zelený trojúhelník\NTIS\1NP\Kavárna	-	m <sup>3</sup> /h	1	průtok	merena	m <sup>3</sup> /h	voda	prubezne
6	T12-ZI-NTIS-Prstor kavárny		Zelený trojúhelník\NTIS\1NP\Kavárna	-	°C	1	teplota	merena	°C	vzduch	prubezne
7	Pokojev teploměr		Zelený trojúhelník\NTIS\1NP\Šatna	-	°C	1	teplota	merena	°C	vzduch	prubezne
8	Elektroměr ZI-NTIS-Garáž-rozvodna-NTIS		Zelený trojúhelník\NTIS\1PP\Rožvodna	-	kW	1	výkon	merena	kW	elektřina	prubezne
9	Elektroměr ZI-NTIS-1PP-Směš. Stanice		Zelený trojúhelník\NTIS\1PP\Výměník	-	kW	1	výkon	merena	kW	elektřina	prubezne
10	Kalorimetr - stroj chladu - NTIS -		Zelený trojúhelník\NTIS\1PP\Výměník	-	GJ/h	1	energie	merena	GJ/h	teplo	prubezne
11	Kalorimetr - topení - NTIS - 67 522 086		Zelený trojúhelník\NTIS\1PP\Výměník	-	GJ/h	1	energie	merena	GJ/h	teplo	prubezne
12	NTIS TV		Zelený trojúhelník\NTIS\1PP\Výměník	-	m <sup>3</sup> /h	1	průtok	merena	m <sup>3</sup> /h	voda	prubezne
13	NTIS stroj chladu - c.E. 500015-21		Zelený trojúhelník\NTIS\1PP\Výměník	-	m <sup>3</sup> /h	1	průtok	merena	m <sup>3</sup> /h	voda	prubezne
14	Elektroměr ZI-NTIS-Garáž		Zelený trojúhelník\NTIS\1NP\Garáž vjezd	-	kW	1	výkon	merena	kW	elektřina	prubezne
15	Pokojev teploměr		Zelený trojúhelník\NTIS\2NP\Chodba nad řátnou	-	°C	1	teplota	merena	°C	vzduch	prubezne
16	Pokojev teploměr		Zelený trojúhelník\NTIS\2NP\Posluchárna US 207	-	°C	1	teplota	merena	°C	vzduch	prubezne

Obrázek 2.4: Ukázka anotace datových bodů

V levé části (zeleně označená hlavička) jsou obsaženy informace z telemetrické soustavy. V pravé části (červeně označená hlavička) se nachází manuálně anotované informace o jednotlivých datových bodech připravené pro následující automatický převod do nově vyvinuté konsolidované databáze a pro využití při navazující tvorbě energetických analytics (např. data mining).

### 2.3.4 Zpřístupnění datového zdroje

Nově vyvinutý systém sběru a zpracování energetických dat byl navržen tak, aby mohl být přímo nasazen místo současného systému sběru energetických dat na ZČU. Pro potřeby testování a pozdějšího využití historických dat bylo potřeba získat od dodavatelů historická měřená data z obou používaných systémů (EMS a telemetrická soustava).

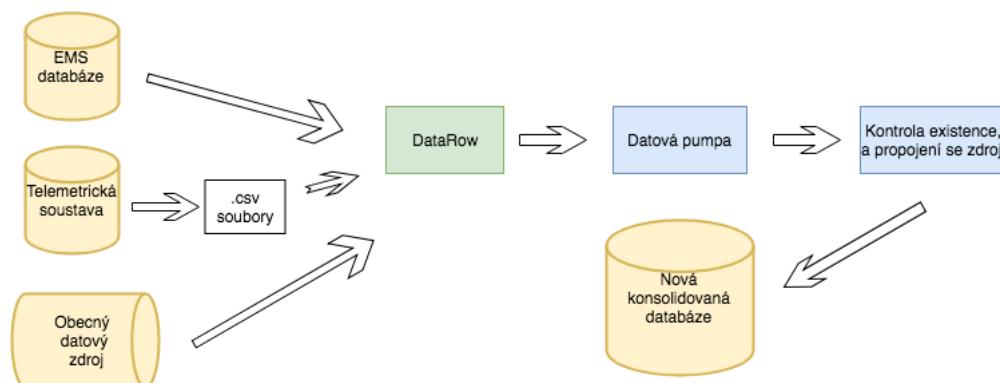
Historická data z telemetrické soustavy byla dodávána dodavatelem periodicky v podobě csv souborů. Tyto csv soubory odpovídají dumpům jednotlivých tabulek telemetrického datového úložiště a jejich bližší popis lze najít v kapitole 2.2. Pro export dat z jednotlivých csv souborů a EMS databáze byl v jazyce *Java* vytvořen nástroj zvaný *Datová pumpa*.

### Datová pumpa

Datová pumpa je nástroj pro zpracování a přenesení energetických dat z externí datové vrstvy do vrstvy datové fúze (viz obrázek 1.1). Systém datové pumpy byl navržen pro účely zpracování dat z externích datových úložišť a jejich přeřazení do nově vytvořené konsolidované databáze, která je kompletně v režii ZČU. Rozhraní datové pumpy bylo navrženo univerzálně tak, aby fungovala pro import z obou

stávajících systémů a bylo možné jednoduché rozšíření o případný další datový zdroj. Návrh a implementace byla provedena ve spolupráci s Pavlem Novotným [7]. Tato bakalářská práce se specializuje na zpracování csv souborů z telemetrické soustavy a některé části datové pumpy.

Obecný systém rozhraní datové pumpy, napojení na libovolný datový zdroj a nahrání do konsolidované databáze je zobrazen na obrázku 2.5.



Obrázek 2.5: Obecné schéma datové pumpy

V levé části obrázku se nachází externí zdroje energetických dat z externích firem, pro které je definováno jednotné datové rozhraní. Rozhraní datové pumpy je realizováno třídou *DataRow*, která zobecňuje specifické datové zdroje jako jsou například telemetrická soustava, EMS a případně další. Po převedení na společný formát jsou data z jednotlivých datových zdrojů posílána do společné části datové pumpy. Ve společné části se data dále zpracovávají a upravují do standardizované podoby pro konsolidovanou databázi. V poslední fázi dojde k vložení do nové databáze.

Rozhraní datové pumpy pracuje s objektem třídy *DataRow*, jehož atributy jsou kolekce *timestamp*, *quality*, *value* a *id\_datapoint*. Data v podobě tohoto objektu lze vkládat do datové pumpy z libovolného externího datového zdroje. Třída *DataRow*:

```

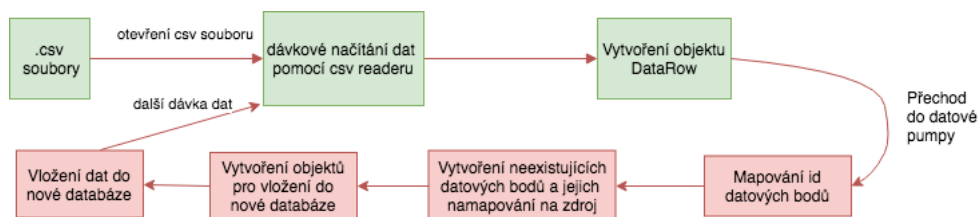
Object:DataRow
public ArrayList<Timestamp> timestamp = new ArrayList <>();
public ArrayList<Integer> quality = new ArrayList <>();
public ArrayList<Float> value = new ArrayList <>();
public ArrayList<String> id_datapoint = new ArrayList <>();

```

Obrázek 2.6: Třída DataRow

Jednotlivé kolekce obsahují sloupce csv souboru *Spotreba* rozložené do příslušných proměnných odpovídajících jednotlivým informacím o měřeném záznamu.

Schéma přenosu dat z telemetrické soustavy (prostřednictvím csv souborů) je zobrazeno na obrázku 2.7.

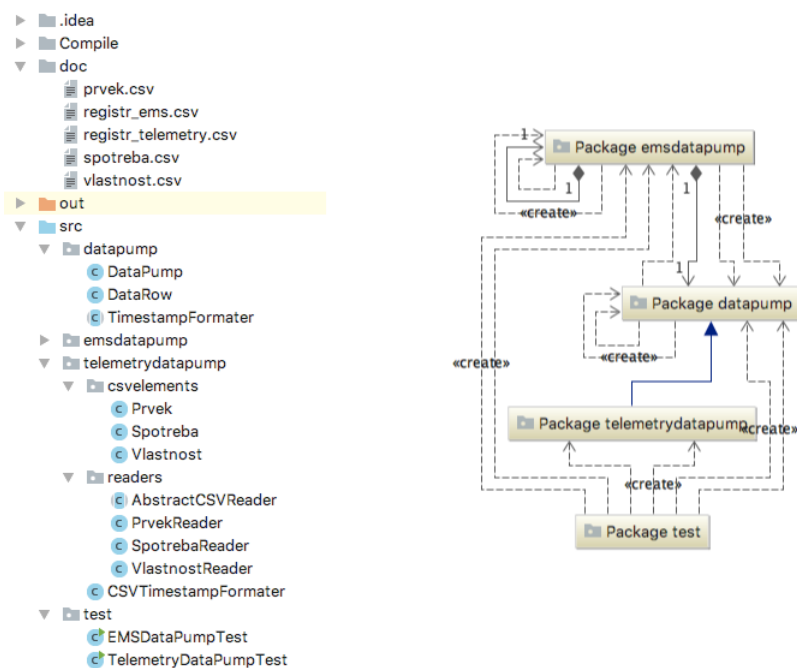


Obrázek 2.7: Schéma přenosu dat z telemetrické soustavy datovou pumpou

Na obrázku 2.7 můžeme vidět jednoduché grafické znázornění přenosu dat z telemetrické soustavy do nově vytvořené konsolidované databáze. Zeleně znázorněné objekty jsou obsaženy v komponentě vytvořené pro zpracování historických dat telemetrické soustavy (csv soubory). Červené označené objekty jsou společné pro libovolný externí datový zdroj, ze kterého byl vytvořen objekt *DataRow*.

## Softwarový design datové pumpy

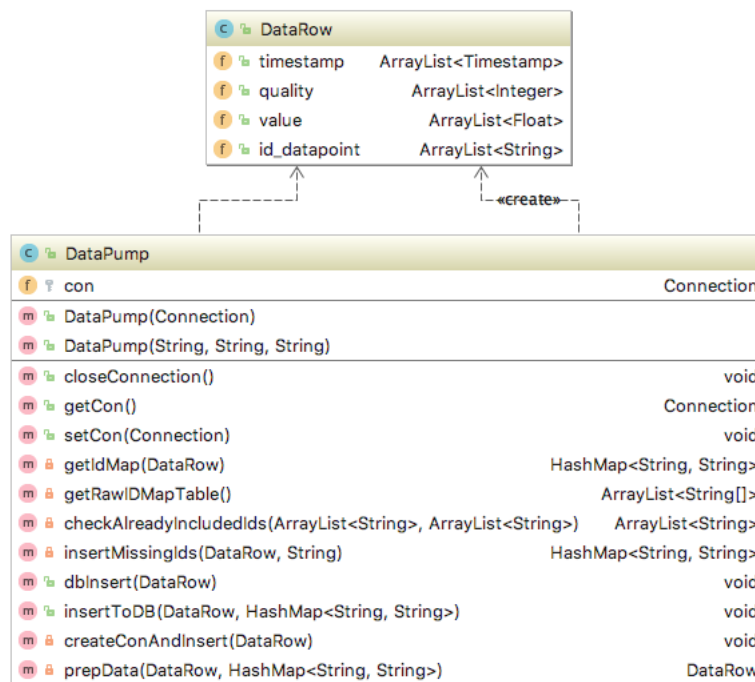
Na obrázku 2.8 je ukázka struktury datové pumpy. V levé části můžeme vidět strukturu softwaru pomocí stromové struktury. V pravé části je znázorněno UML schéma jednotlivých balíčků.



Obrázek 2.8: UML schéma balíčků datové pumpy

V současné době byly implementovány komponenty pro dva datové zdroje (EMS databáze a Telemetrická soustava resp. csv exporty). V balíčku *datapump* je společné rozhraní pro libovolný datový zdroj. Balíčky *emsdatapump* a *telemetrydatapump* obsahují jednotlivé implementované komponenty, které zpracovávají data z externích

zdrojů a převádějí je do společného formátu pro datovou pumpu. Balíček *telemetrydatapump* v sobě obsahuje další balíčky *csvelements* a *readers*. V balíčku *csvelements* jsou třídy, do jejichž instancí se ukládají načtená data z csv souborů pomocí příslušných čteček z balíčku *readers*. V balíčku *test* jsou vytvořeny spouštěcí třídy pro jednotlivé datové zdroje, které obsahují metodu *main*.



Obrázek 2.9: UML schéma balíčku datapump

Na obrázku 2.9 můžeme vidět podrobné UML schéma balíčku *datapump*. Rozhraní *DataRow* a třída *DataPump* jsou společné pro libovolný datový zdroj a dohromady tvoří spolehlivý nástroj pro zajištění konzistence dat při jejich vkládání do nově vyvinuté konsolidované databáze.

### Funkčnost datové pumpy pro přenos dat z telemetrické soustavy

Nejprve je potřeba otevřít a načíst csv soubor *Spotreba*, který byl dodán od dodavatele a obsahuje údaje o měření energetických dat v areálu Fakulty aplikovaných věd ZČU. Vzhledem k velikosti souboru *Spotreba* ( 1.3 GB) je potřeba provádět načítání a zpracování dat po dávkách, aby nedocházelo k zahlcení paměti počítače. Experimentálně bylo zjištěno, že vhodnou dávkou pro vytvoření objektu *DataRow* je 5000 záznamů. Ke čtení csv souboru se využívá externí knihovna *JavaCSVReader*. Pro připojení ke konsolidované databázi se využívá knihovna *JDBC driver*.

Načítání ze souboru probíhá v cyklu *while*, v každém kroku tohoto cyklu dojde k načtení jednoho řádku csv souboru (jednoho záznamu měření) a tento řádek se uloží

do pole typu *String*, kdy každá nová položka je oddělena středníkem. Z tohoto pole se odpovídající hodnoty ukládají na další pozici odpovídajících kolekcí v objektu *DataRow*.

Po příchodu objektu *DataRow* do společné části datové pumpy dochází k namapování příchozích datových bodů ze zdrojů na nově vytvořené datové body v konsolidované databázi. Algoritmus nejprve načte všechna data z tabulky *fact\_id\_map* (viz 2.3.1), která obsahuje informace o namapování nových datových bodů na jejich zdroj z externího datového úložiště. Pokud algoritmus objeví *id* zdroje, které dosud v konsolidované databázi nebylo, vytvoří nový datový bod v tabulce *fact\_datapoint*. Tento nově vytvořený datový bod je následně namapován na svůj zdroj v tabulce *fact\_id\_map*. Tento proces je potřeba, aby nedocházelo k vytváření duplikátních datových bodů a aby bylo možné dodatečně zjistit zdroj daného datového bodu v konsolidované databázi.

Z tabulky *fact\_id\_map* se vytváří objekt *HashMap < String, String >*, kde klíčem je *id* datového bodu ve svém zdrojovém úložišti a hodnotou je *id* datového bodu v nově vyvinuté databázi. Tato *HashMap* se dále využívá při vkládání samotných naměřených historických hodnot ze zdrojového úložiště.

V dalším kroce dochází k vkládání naměřených historických hodnot do tabulky *fact\_value*. Do tabulky se ukládají záznamy o hodnotě měření, čase měření, kvalitě měřeného signálu a cizí klíč, který odkazuje na *id* datového bodu v tabulce *fact\_datapoint*.

## 2.4 Rozhraní pro získání energetických dat

Pro potřeby následného využití konsolidovaných dat v navazujících aplikacích (viz aplikační vrstva) je potřeba zajistit konzistenci a kvalitu poskytovaných dat. Ke smysluplném využití těchto dat je potřeba standardizace, tj. aby byla ekvidistantní, čili aby jednotlivé záznamy byly po stejném časovém úseku (například přesně po 10 minutách). Avšak při analýze naměřených dat bylo zjištěno, že datové sady jsou mnohdy neekvidistantní a periody mezi měřeními jsou velmi různorodé. Proto byl vytvořen přístupový modul *DAL (Data Acces Layer)*, který ošetřuje poskytovaná data. Přesněji ekvidistantně vzorkuje a případně doplňuje datové sady. Tento přístupový modul byl následně využit pro potřeby modelování vybraného energetického zařízení v jazyce *Matlab*.

### 2.4.1 Popis struktury data acces layeru

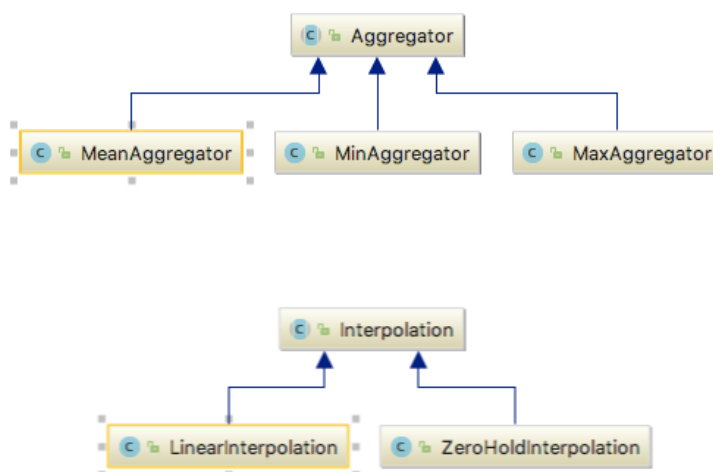
Přístupový modul pro získání standardizovaných energetických dat byl vytvořen v programovacím jazyce Java. Programovací jazyk java byl využit z následujících důvodů:

- Rozvinuté knihovny umožňující jednoduchou práci s databází
- Pokročilé nástroje pro zpracování a úpravu dat

- Možnost použití modulů v jazyce Java v Matlabu při navazující tvorbě energetických analytics

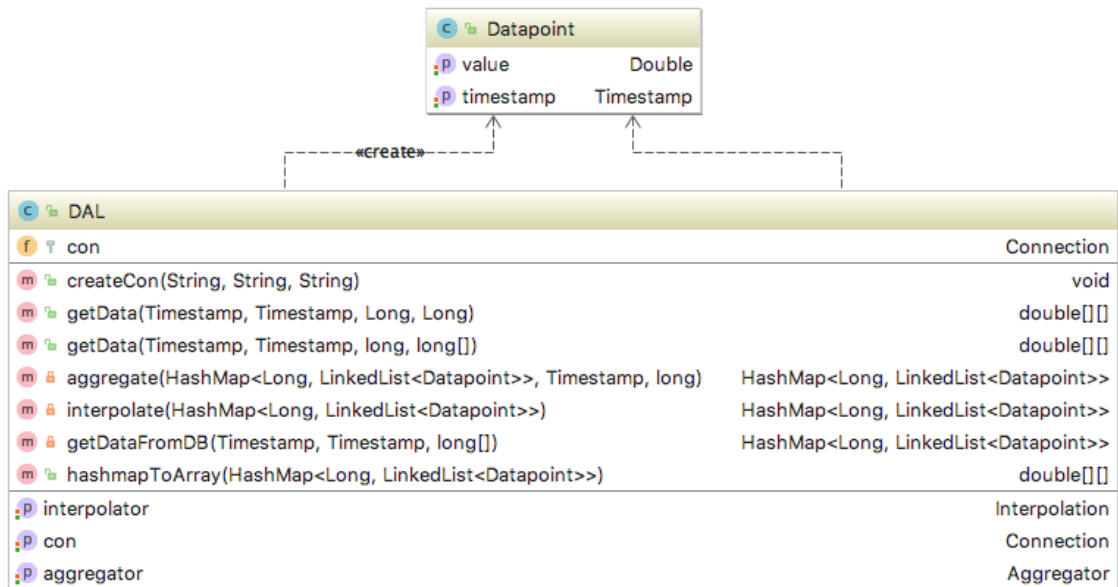
### Softwarový design data access layer

Základními balíčky data access layer jsou *aggregation* a *interpolation*, jejichž class diagramy jsou zobrazeny na obrázku 2.10 .



Obrázek 2.10: Class diagramy balíčků aggregation a interpolation

Ze schematu je patrné, že v obou balíčcích byly vytvořeny hlavní abstraktní třídy, které částečně určují strukturu svých potomků. Tato implementace zajišťuje jistotu, že s libovolným potomkem implementujícím abstraktní třídu, lze pracovat částečně shodným způsobem (tj. implementuje nutně abstraktní metody definované v předchůdci).



Obrázek 2.11: Podrobný diagram tříd balíčku dataaccesslayer

Na obrázku 2.11 můžeme vidět podrobný class diagram balíčku *dataaccesslayer*. V horní části se nachází třída *Datapoint*. Objekty této třídy v programu reprezentují jednotlivé záznamy měření. S touto třídou pracuje hlavní třída *DAL*, která zajišťuje veškerou funkčnost tohoto rozhraní pro získání standardizovaných energetických dat.

## Data access layer

Objekt *Datapoint* (viz 2.11) obsahuje atributy *value* (hodnota měření) a *timestamp* (časová značka měření). Tento objekt v programu reprezentuje konkrétní datový bod, který byl naměřen senzorem a uložen do databáze i se svou časovou značkou.

Pro standardizaci dat se využívají dvě základní funkcionality - agregace a interpolace dat. Agregace se používá, pokud se v jedné požadované periodě nachází více měření. Interpolace se používá, pokud se v dané periodě nenachází žádné měření a hodnotu musíme dopočítávat pomocí nejbližšího předchozího a nejbližšího dalšího měření.

Hlavní třídou *Data Acces Layeru* je *DAL*. Instanci této třídy je možné vytvořit několika různými konstruktory, které specifikují atributy této třídy. Mezi tyto atributy patří objekt *aggregator* specifikující instanci implementační třídy pro agregaci, objekt *interpolator* specifikující instanci implementační třídy pro interpolaci a objekt typu *Connection* specifikující připojení k databázi. Pro připojení k databázi se používá *JDBC Driver*.



## Agregace

Agregace je matematická operace používaná v případech, kdy v daném časovém období existuje více naměřených dat, ze kterých je potřeba vytvořit pouze jednu hodnotu, se kterou se bude dále pracovat.

Agregace je v programu reprezentována abstraktní třídou *Aggregator*. Implementace konkrétních agregačních funkcí je realizována v potomcích abstraktní třídy *Aggregator*. Každá oddělená třída obsahuje metodu *aggregate(double newValue)*, která podle matematického vzorce vypočte novou hodnotu, metodu *getValue()*, která vrací agregovanou hodnotu a metodu *reset()*, která nastaví zpět původní hodnoty pro další použití. Byly implementovány tři stavové agregátory, které udržují stavové informace například o aktuální hodnotě a počtu agregovaných hodnot. Implementované agregační funkce jsou:

- **MinAggregator** - vybírá vždy nejnižší naměřenou hodnotu za danou periodu měření

$$value = \min(newValue, value)$$

- **MaxAggregator** - vybírá vždy nejvyšší naměřenou hodnotu za danou periodu měření

$$value = \max(newValue, value)$$

- **MeanAggregator** - vypočítává průměrnou hodnotu naměřených hodnot za danou periodu měření. Průměrná hodnota pro  $n$ -tou hodnotu se počítá podle vzorce:

$$value = \frac{1}{n} \cdot newValue + \frac{n-1}{n} \cdot value ,$$

kde *value* je stavová hodnota agregátoru, *newValue* je nově agregovaná hodnota a  $n$  je počet dosud agregovaných hodnot v případě *MeanAggregator*.

## Interpolace

Interpolace v numerické matematice znamená nalezení přibližné hodnoty v určitém intervalu, je-li její hodnota známá jen v některých jiných bodech tohoto intervalu. Jedná se o určitou aproximaci dané funkce za podmínky, že hledaná křivka přesně prochází všemi známými body. Využívá se v případech, kdy jsou hodnoty funkce získány měřením a v některých intervalech měření chybí nebo není dostatečně kvalitní [10].

Pro potřeby interpolace je opět vytvořena abstraktní třída *Interpolation*, která je následně specifikována v potomcích. Interpolační funkce iterativně prochází ekvidistantně vypočtené hodnoty po agregaci. Pokud metoda najde *Datapoint*, do kterého nebylo nic agregováno (nemá vyčíslitelnou hodnotu - *Double.NaN (Not a Number)*), provede v tomto bodě interpolaci. Interpolační funkce jsou implementovány dvě:

- **ZeroHoldInterpolation** - prohledá předchozí naměřené hodnoty a danému *Datapointu* přiřadí stejnou hodnotu, jako měl jeho nejbližší předchůdce
- **LinearInterpolation** - funkce najde nejbližšího předchůdce a následovníka daného *Datapointu* a provede lineární interpolaci tak, že tyto dva body propojí pomyslnou přímkou a hodnotu *Datapointu* odečte z této přímky v čase, který *Datapointu* náleží. Vzorec pro výpočet lineární interpolace:

$$x = x_1 + \frac{x_2 - x_1}{t_2 - t_1} \cdot (t - t_1) ,$$

kde  $x_1$  a  $t_1$  je hodnota a časová značka nejbližšího předcházejícího *Datapointu*,  $x_2$  a  $t_2$  je hodnota a časová značka nejbližšího následujícího *Datapointu* a  $x$  a  $t$  je hodnota a časová značka *Datapointu*, jehož hodnota je vypočítávána.

## 2.4.2 Popis funkčnosti data acces layeru

Hlavní metoda třídy *DAL*, která bude mimo jiné využívána pro volání z *Matlabu*, je metoda *getData*. Vstupní parametry této metody jsou *Timestamp* objekty určující počáteční a koncové hodnoty časového intervalu, ze kterého se mají načíst naměřená data z databáze, požadovaná peridoda vzorkování v minutách a pole typu *Long*, ve kterém jsou *id* datových bodů, ze kterých chceme vytvářet model. Návrátová hodnota této metody je dvourozměrné pole typu *double*, kde jsou v první řádce údaje o čase měření a na dalších řádcích jsou jednotlivé naměřené hodnoty datových bodů odpovídající časům z prvního řádku. Ukázka kódu metody *getData*:

```
public double [][] getData(Timestamp start , Timestamp end , long
    period , long [] ids) {
    HashMap<Long, LinkedList<Datapoint>> nonEqidDatapoints
        = getDataFromDB(start , end , ids);
    HashMap<Long, LinkedList<Datapoint>> eqidDatapoints =
        new HashMap<Long, LinkedList<Datapoint>>();

    eqidDatapoints = aggregate(nonEqidDatapoints , start ,
        period);

    eqidDatapoints = interpolate(eqidDatapoints);

    double [][] equidDatapointsInArray =
        hashmapToArray(eqidDatapoints);

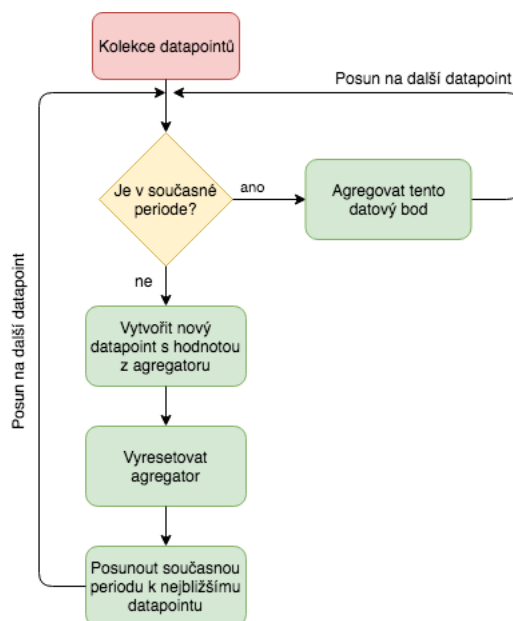
    return equidDatapointsInArray;
}
```

Z metody *getData* se nejprve se volá metoda pro načtení naměřených dat z databáze. Metoda *getDataFromDB* načte z databáze data za určené časové období s danými *id*. Z načtených dat se vytvoří objekt *HashMap < Long, LinkedList < Datapoint >>*,

kde klíčem je *id* datového bodu a v kolekci *LinkedList* jsou jednotlivé *Datapointy* seřazené vzestupně podle času. Takto vytvořená *HashMap* se vrací zpět do metody *getData*, odkud se naměřená data dále upravují do požadované podoby.

V dalším kroce se vytváří nový objekt *HashMap < Long, LinkedList < Datapoint >>*, kde klíčem jsou opět *id* datových bodů, ale v kolekci *LinkedList* se vytváří *Datapointy* ekvidistantně, čili časové údaje jednotlivých *Datapointů* jsou s rozdílem dané periody vzorkování. Tento proces probíhá zároveň s procesem agregace. Data v objektu *HashMap* se prochází dvojitým for cyklem, kdy první for cyklus prochází jednotlivé klíče *HashMapy* a druhý for cyklus prochází jednotlivé *Datapointy* v kolekci *LinkedList*. K ohraničení dané periody se využívá proměnná typu *Timestamp* pojmenovaná *pointer*, která ukazuje vždy na konec periody. Při postupném procházení *Datapointů* se kontroluje, jestli je časový údaj *Datapointu* před koncem periody. Pokud se *Datapoint* nachází v dané periodě, pošle se jeho hodnota do *agregatoru* metodou *aggregate*. Pokud se *Datapoint* nachází za koncem dané periody, přečte se z *agregatoru* aktuální hodnota a v nově vytvářeném objektu *HashMap* se vytvoří nový *Datapoint* s časovou značkou *pointeru* a hodnotou z *agregatoru*. Pokud v dané periodě nebyl žádný *Datapoint* načtený z databáze, do nově vytvořeného se jako hodnota uloží nečíselná hodnota (*Double.NaN*). Po vytvoření nového *Datapointu* v nové *HashMap* se vyresetuje *agregator* a pokračuje se dále.

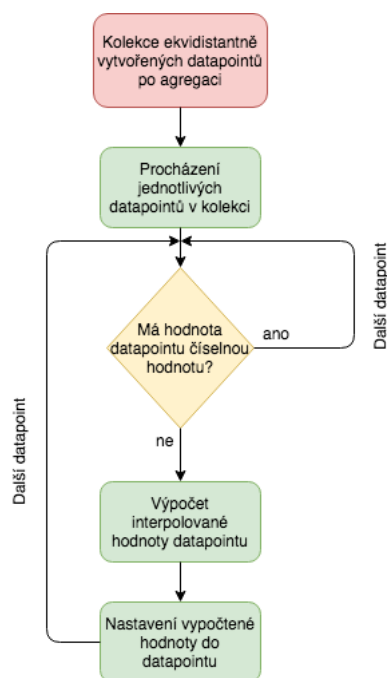
Proces agregace a standardizace dat do ekvidistantní podoby je znázorněn následujícím vývojovým diagramem:



Obrázek 2.12: Vývojový diagram procesu agregace

Po agregaci se již ekvidistantní data v objektu *HashMap* pošlou do *interpolatoru*. *Interpolator* opět prochází data ve dvojitým for cyklu přes klíče *HashMapy* a jednotlivé *Datapointy* v kolekci *LinkedList*. Pokud je nalezen *Datapoint* se nečíselnou hodnotou (*Double.Nan*), provede v tomto bodě interpolaci podle zvolené interpolační funkce.

Proces interpolace lze graficky znázornit následujícím vývojovým diagramem:



Obrázek 2.13: Vývojový diagram procesu interpolace

Nakonec se objekt *HashMap* s již standardizovanými daty předá do metody *hasmapToArray*. Tato metoda prochází objekt *HashMap* a vytváří z něho dvourozměrné pole typu *double*, které obsahuje v prvním řádku časové značky *Datapointů* a na dalších řádcích hodnoty *Datapointů* odpovídající časovým značkám z prvního řádku. Na každé další řádce jsou hodnoty *Datapointů* odpovídající jednomu datovému bodu z databáze.

Po převedení *HashMap* na dvourozměrné pole typu *double* se toto pole vrací jako návratová hodnota zpět do *Matlabu*, kde se s ním dále pracuje a vytváří model daného energetického zařízení.

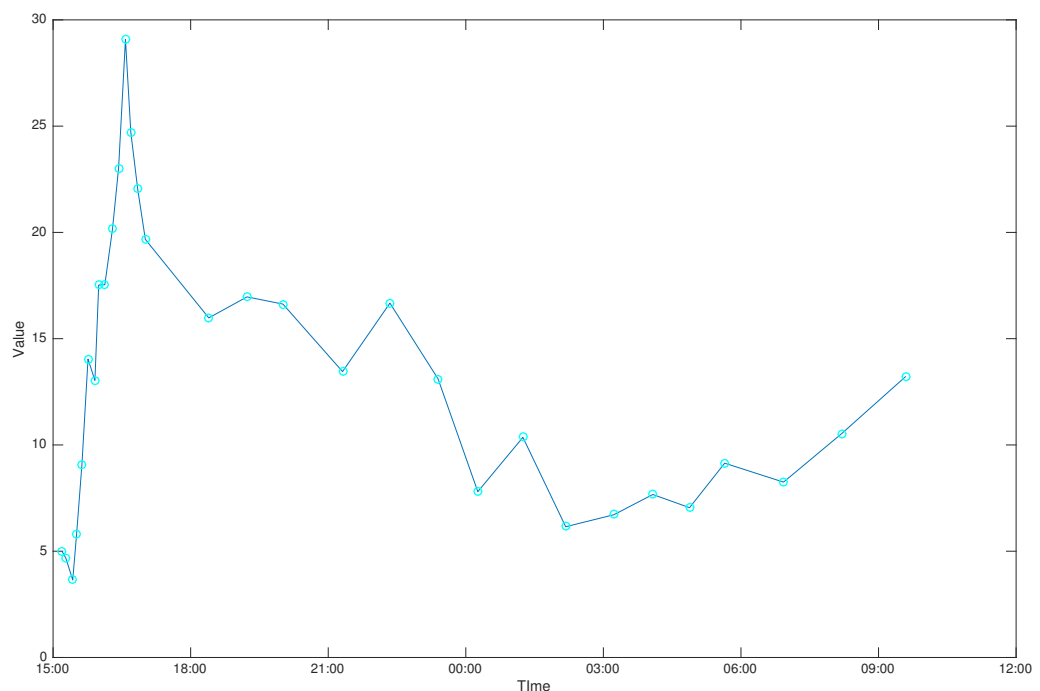
### 2.4.3 Ukázka použití data acces layeru z Matlabu

Pro potřeby testování a ukázkou funkčnosti data acces layeru byl v prostředí Matlab vytvořen ukázkový signál.

Ukázka tvorby ukázkového signálu v Matlabu:

```
for i = 1:countSamples
    if (i < countSamples/2)
        tDiff = randi([5,10]);
    else
        tDiff = randi([45,90]);
    end
    timestamp(end+1) = timestamp(end) + tDiff * 1/1440;
    signal(end+1) = (signal(end) + 3 * sin(0.75 * i/pi) + 3 *
        randn);
end
```

Ná následujícím obrázku 2.14 je vykreslení ukázkového signálu.



Obrázek 2.14: Uměle vytvořený ukázkový signál

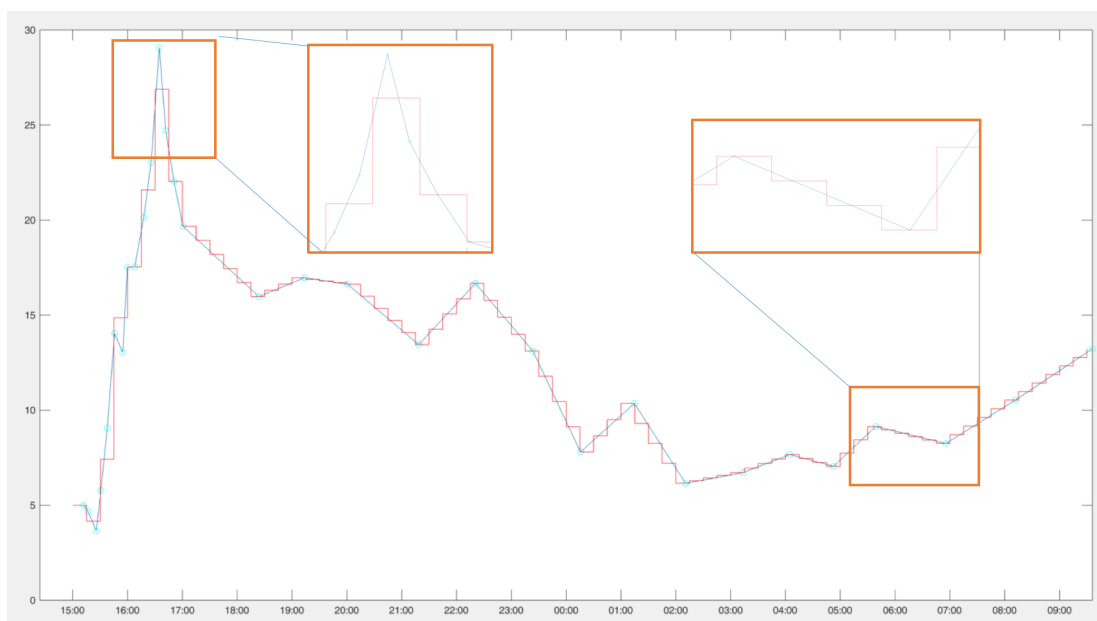
Z obrázku je patrné, že první polovina dat byla vygenerována s relativně krátkou periodou, konkrétně je perioda náhodné celé číslo v rozmezí od 5 do 10 minut. Za předpokladu požadované periody 15 minut je tato část určena pro ukázkou a testování funkčnosti agregace data acces layeru.

V pravé části grafu je zřejmé, že se perioda mezi měřeními prodloužila, konkrétně

se jedná o náhodné celé číslo v rozmezí od 45 do 90 minut. Opět za předpokladu požadované periody 15 minut slouží tato část pro testování a ukázkou interpolace data acces layeru.

Takto vytvořený umělý signál byl vložen do konsolidované databáze. Pro ukázkou funkčnosti data acces layeru byl v Matlabu vytvořen ukázkový skript. Tento ukázkový skript přistupuje k uměle vytvořenému ukázkovému signálu prostřednictvím data acces layeru. Požadovaná perioda pro ukázkou funkčnosti agregace a interpolace byla zvolena 15 minut.

Na následujícím obrázku 2.15 je vidět porovnání ukázkového signálu se zpracovaným ukázkovým signálem prostřednictvím data acces layeru.



Obrázek 2.15: Porovnání uměle vytvořeného signálu před a po zpracování data acces layerem

Na obrázku 2.15 vidíme modře vykreslený ukázkový signál a červeně vykreslený zpracovaný signál data acces layerem. Nad křivkami jsou vidět dva přiblížené výřezy, které podrobně ukazují, jak proběhly procesy agregace a interpolace.

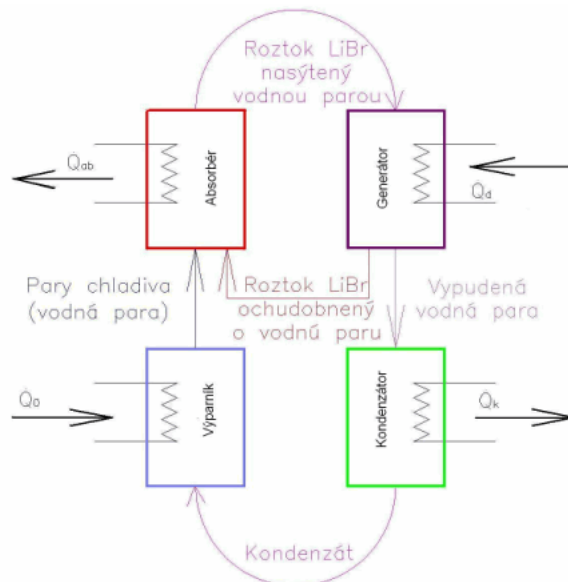
## Kapitola 3

# Model vybraného energetického zařízení

V areálu Západočeské univerzity je instalována celá řada energetických zařízení, která jsou určena k transformaci, distribuci či uložení různých druhů energií. Absorpční chladicí zařízení představuje jednu z největších transformačních jednotek poskytující chlad pro klimatizační jednotky umístěné ve Fakultě aplikovaných věd. S ohledem na vysokou energetickou náročnost bylo vybráno jako vhodné zařízení pro vytvoření energetických analytics zaměřených na monitorování a řízení spotřeby tepelné a elektrické energie.

### 3.1 Princip absorpčního chlazení

Absorpce je fyzikální děj, při němž se rozpouští plynná fáze v kapalině. Kapalina se nazývá absorbent a plyn absorbát. Jako pracovní dvojice se nejčastěji používají amoniak ( $NH_3$ ) - voda, voda - vodný roztok bromidu litného ( $LiBr$ ) [5].



Obrázek 3.1: Schéma jednoduššího kontinuálního absorpčního chladičského zařízení [5]

Princip absorpčního chlazení (viz obrázek 3.1) je založen na dobré rozpustnosti absorbentu, přičemž výrazný vliv na dynamiku tohoto děje mají teplotní úrovně v absorbéru a generátoru systému. Na vypuštění absorbátu se roztok z absorbéru přečerpá do generátoru, kde se mu dodá teplo potřebné k jeho vypaření. Z generátoru proudí chladivo do kondenzátoru a odtud zpět do vyparníku. Roztok absorbentu ochuzený o chladivo se po ochlazení vrátí do absorbéru. Na desorpci je třeba přivést tepelný tok s relativně vysokou teplotní úrovní (80 až 120 °C).

Při zanedbání příkonu čerpadla na dopravu bohatého roztoku absorbentu a chladiva absorbéru do generátoru můžeme základní energetickou bilanci absorpčního oběhu napsat ve tvaru 3.1:

$$\dot{Q}_0 + \dot{Q}_d = \dot{Q}_{ab} + \dot{Q}_K, \quad (3.1)$$

kde  $\dot{Q}_0$  je chladičí výkon,  $\dot{Q}_d$  je tepelný příkon potřebný k desorpci,  $\dot{Q}_{ab}$  je tepelný výkon absorbéru a  $\dot{Q}_K$  je tepelný výkon kondenzátoru.

Energetická efektivita absorpčního chladičského systému je potom dána jeho výkonovým číslem ( $COP$  - Coefficient of performance), které se určí ze vztahu 3.2:

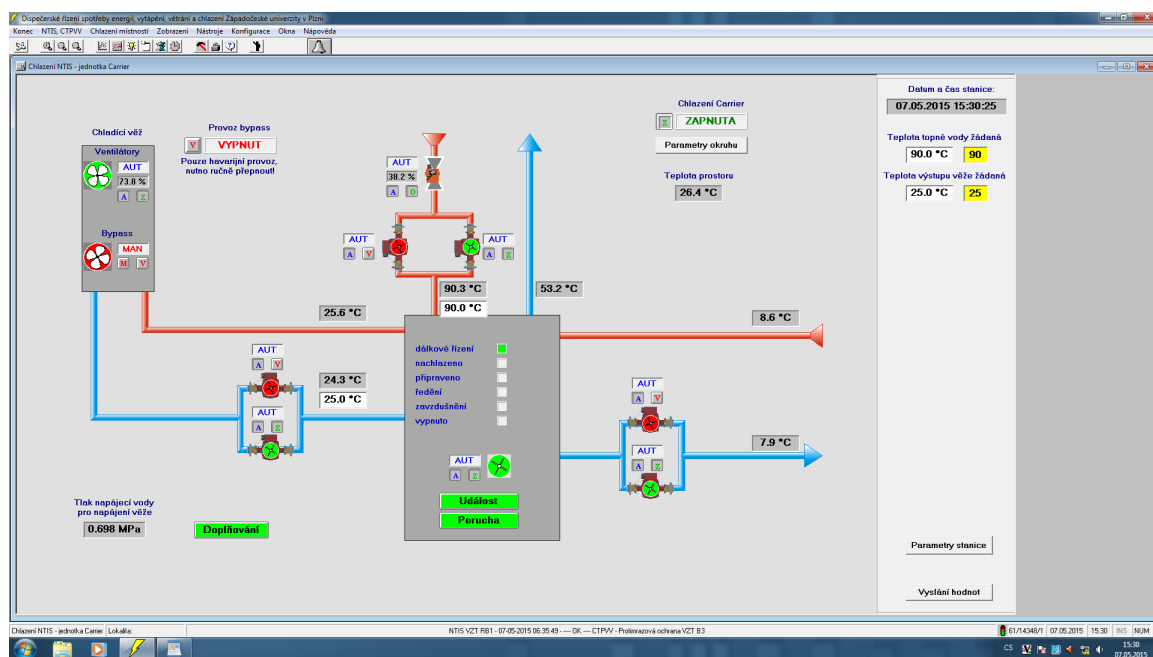
$$COP = \frac{\dot{Q}_0}{\dot{Q}_d} \quad (3.2)$$



V případě absorpčního chladicího zařízení pro budovu Fakulty aplikovaných věd se příkon čerpadla na dopravu pohybuje okolo 60 kW, což je nezanedbatelná hodnota. Energetická efektivita tohoto absorpčního chladicího systému je tedy dána vztahem 3.3:

$$COP = \frac{\dot{Q}_0}{\dot{Q}_d + P_{chill}}, \quad (3.3)$$

kde  $P_{chill}$  je výkon čerpadla pro dopravu absorbentu.



Obrázek 3.2: Screenshot schématu chladicího zařízení pro řízení z velína

Na obrázku 3.2 vidíme ukázkou schématu absorpčního chladicího zařízení ze softwaru pro dálkové řízení. Dálkové řízení probíhá nastavením řízených parametrů z velína. Řízené parametry systému jsou *Teplota topné vody* a *Teplota výstupu věže*.

Při výpočtu parametrů modelů budeme pracovat s měřenými hodnotami z absorpčního chladicího zařízení:

- $T_{eva}^{in}$  [°C] - vstupní teplo do výparníku
- $T_{eva}^{out}$  [°C] - výstupní teplo z výparníku
- $T_{ac}^{in}$  [°C] - vstupní teplo do absorber/condenser
- $T_{ac}^{out}$  [°C] - výstupní teplo z absorber/condenser
- $T_{gen}^{in}$  [°C] - vstupní teplo do generatoru
- $T_{gen}^{out}$  [°C] - výstupní teplo z generatoru

- $Q_{eva}$  [kW] - výkon výparníku

Výkon výparníku není přímo měřená veličina, její hodnota je tedy vypočítávána podle vztahu 3.4:

$$Q_{eva} = c_w \cdot \dot{m}_w \cdot (T_{eva}^{in} - T_{eva}^{out}) , \quad (3.4)$$

kde  $c_w$  je měrná tepelná kapacita vody,  $\dot{m}_w$  je průtok vody a  $(T_{eva}^{in} - T_{eva}^{out})$  je rozdíl vstupní a výstupní teploty výparníku. Protože průtok chlazené vody  $\dot{m}$  je neměřený, bude výkon výparníku uvažován relativní. Pro jednoduchost průtok chlazené vody  $\dot{m}$  bude uvažován konstantní, přičemž jeho hodnota bude zohledněna v koeficientech uvažovaných statických modelů absorpčního chladicího zařízení.

## 3.2 Adapted Gordon - Ng model (GN)

Obecný termodynamický model absorpčního chladicího stroje vytvořený Gordonem a Ng je kombinací fyzikálního a empirického přístupu. Vychází z předpokladu, že dominantní nevratnost absorpčního chladicího stroje je přenos hmotnosti s konečnou rychlostí. Ztráty způsobené přenosem hmotnosti konečnou rychlostí mohou být aproximovány jako teplotní nezávislost. Z [4] vychází, že ztráty na výparníku můžeme zanedbat a na ztráty na dvou tepelných výměnících (generator, absorber/condenser) můžeme pohlížet jako na konstanty charakteristické pro jednotlivý chladicí stroj. Obecnou rovnicí pro GN model lze získat po sérii transformací, začínajících prvním termodynamickým zákonem a využívajících entropické rovnováhy. GN model vypočítává převrácenou hodnotu COP podle následující rovnice (3.5):

$$\frac{1}{COP} = \left[ \frac{T_{ac}^{in} - T_{eva}^{out}}{T_{eva}^{out}} \right] \cdot \left[ \frac{T_{gen}^{in}}{T_{gen}^{in} - T_{ac}^{in}} \right] + \left[ \frac{1}{Q_{eva}} \right] \cdot \left[ \frac{T_{gen}^{in}}{T_{gen}^{in} - T_{ac}^{in}} \right] \cdot \left[ \alpha_1 - \alpha_2 \cdot \frac{T_{ac}^{in}}{T_{gen}^{in}} \right] \quad (3.5)$$

kde  $\alpha_1$  a  $\alpha_2$  jsou regresní parametry získané z experimentálních dat a zároveň konstanty charakterizující entropii konkrétního chladicího stroje.

Vzhledem k tomu, že poměr  $\frac{T_{ac}^{in}}{T_{gen}^{in}}$  oproti  $\left[ \frac{T_{gen}^{in} - T_{ac}^{in}}{T_{gen}^{in} \cdot COP} - \frac{T_{ac}^{in} - T_{eva}^{out}}{T_{eva}^{out}} \right] \cdot Q_{eva}$  vede na polynom, je možné vypočítat parametry  $\alpha_1$  a  $\alpha_2$  použitím lineární regrese.

S ohledem na to, že účel této analýzy je porovnání rozdílných přístupů k modelování pomocí odchylek mezi modelem a reálnými hodnotami z chladicího stroje, byly výsledné rovnice GN modelu přizpůsobeny k výpočtu kapacity chladicího stroje. Tepelnou energii dodanou do chladicího systému lze vypočítat pomocí COP následovně

$$\hat{Q}_{eva} = \frac{B}{1/COP - A} \quad (3.6)$$

a tepelnou energii odebranou z externího tepelného zdroje

$$\hat{Q}_{gen} = \frac{B}{1 - A \cdot COP} , \quad (3.7)$$

kde koeficienty  $A$  a  $B$  vypadají takto

$$A = \left[ \frac{T_{ac}^{in} - T_{eva}^{out}}{T_{eva}^{out}} \right] \cdot \left[ \frac{T_{gen}^{in}}{T_{gen}^{in} - T_{ac}^{in}} \right] \quad (3.8)$$

$$B = \left[ \frac{T_{gen}^{in}}{T_{gen}^{in} - T_{ac}^{in}} \right] \cdot \left[ \alpha_1 - \alpha_2 \cdot \frac{T_{ac}^{in}}{T_{gen}^{in}} \right]$$

### 3.3 Multivariate polynomial regression model (MPR)

Multivariable polynomial model (MPR) je model typu black-box. Z [4] vychází, že, tento model velice efektivně popisuje komplexní nelineární vztahy mezi vstupními a výstupními proměnnými systému bez nutnosti integrace popisu fyzikálních procesů uvnitř systému. Parametry MPR modelu se vypočítávají tak, aby model odpovídal experimentálním datům. Vypočtené parametry minimalizují kritérium nejmenších čtverců residuí polynomiální funkcí. Vzhledem ke své jednoduché struktuře jsou MPR modely využívány v mnoha oblastech jako například řízení, detekce poruch, optimalizace, diagnostika a další. Model druhého řádu absorpčního chladicího stroje lze vyjádřit následující rovnicí (3.9):

$$Q_{eva} = \beta_{0,k} + \beta_{1,k}T_{gen}^{in} + \beta_{2,k}T_{ac}^{out} + \beta_{3,k}T_{eva}^{in} + \beta_{4,k}T_{gen}^{in}T_{ac}^{out} + \beta_{5,k}T_{gen}^{in}T_{eva}^{in} + \beta_{6,k}T_{ac}^{out}T_{eva}^{in} + \beta_{7,k}(T_{gen}^{in})^2 + \beta_{8,k}(T_{ac}^{out})^2 + \beta_{9,k}(T_{eva}^{in})^2, \quad (3.9)$$

kde  $\beta_{i \in \{0, \dots, 9\}}$  jsou regresní parametry systému, které se identifikují pomocí lineární regrese minimalizující kritérium nejmenších čtverců na základě experimentálních dat.

Predikovaná hodnota  $\hat{Q}_{eva}$  se vypočítá následovně

$$\hat{Q}_{eva} = \varphi \cdot reg^T, \quad (3.10)$$

kde  $\varphi = [\beta_0, \dots, \beta_9]$  je vektor regresních koeficientů a  $reg = [I, T_{gen}^{in}, T_{ac}^{out}, \dots, (T_{eva}^{in})^2]^T$  je matice regresorů.

### 3.4 Porovnání modelů

Tato sekce se zabývá porovnáním dvou termodynamických modelů (GN a MPR), které byly navrženy a implementovány v Matlabu na základě matematických vztahů popsaných v předchozích sekcích. Mean Absolute Error (MAE) a Root Mean Squared Error (RMSE) jsou dvě často používané metody pro vyjádření přesnosti odhadovaných hodnot.

### 3.4.1 RMSE kritérium

RMSE (Root Mean Square Error) je standardní test odchylky residuí. Residua vyjadřují vzdálenost vypočtených dat z modelu od reálných naměřených dat z absorpčního chladicího stroje. Jinými slovy tento test vyjadřuje jak jsou údaje rozmístěny okolo linearizační přímky podle kritéria nejmenších čtverců. RMSE se běžně používá v klimatologii, předpovědích a regresní analýze pro ověření výsledků. Výpočet residuí 3.11 [8]

$$RMSE = \sqrt{(Q_{eva} - \hat{Q}_{eva})^2} \quad (3.11)$$

kde  $\hat{Q}_{eva}$  jsou odhadovaná data z modelu a  $Q_{eva}$  jsou naměřená experimentální data z absorpčního chladicího stroje. Čára nad rozdílem v kvadrátu vyjadřuje průměrnou hodnotu. Stejný vzorec lze zapsat také následujícím vztahem 3.12

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Q_{eva}(i) - \hat{Q}_{eva}(i))^2}{N}}, \quad (3.12)$$

kde  $N$  je počet diskrétně naměřených dat,  $Q_{eva}$  je vektor naměřených dat a  $\hat{Q}_{eva}$  je vektor predikovaných hodnot modelem.

### 3.4.2 MAE kritérium

Ve statistice MAE (Mean Absolute Error) kritérium vyjadřuje průměrnou velikost chyb v sadě předpovědí bez ohledu na jejich orientaci. Při porovnání s testovacím vzorkem se jedná o průměrný absolutní rozdíl mezi odhadem a skutečným měřením. Absolutní rozdíly v každém časovém okamžiku mají stejnou váhu [9].

$$MAE = \frac{\sum_{i=1}^N |Q_{eva}(i) - \hat{Q}_{eva}(i)|}{N} \quad (3.13)$$

### 3.4.3 Vlastnosti RMSE a MAE kritéria

#### Podobnosti

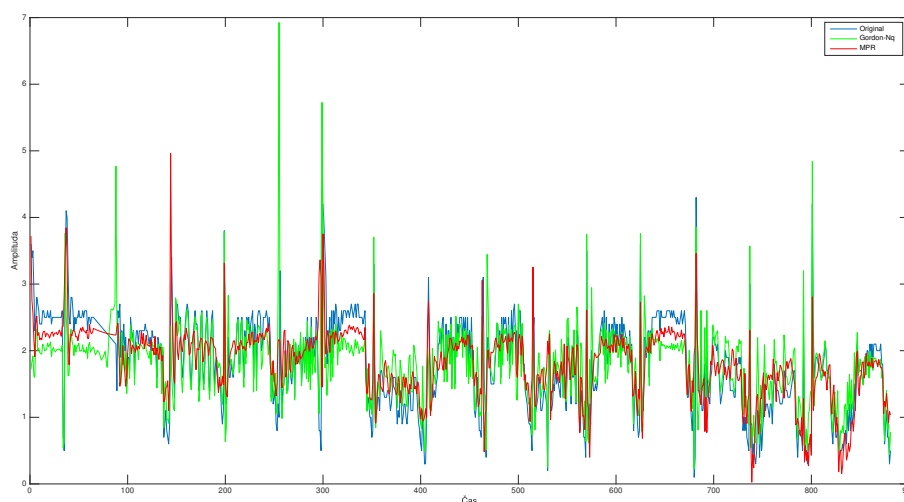
- Vyjadřují průměrnou chybu predikce modelu
- Dosahují hodnot od 0 do  $\infty$
- Jsou nezávislé na orientaci chyby
- Jsou takzvaně negativně orientované, což znamená, že čím nižší hodnota kritéria, tím lepší predikce chování systému

## Rozdíly

- Z důvodu umocnění před zprůměrováním přikládá RMSE větší váhu relativně velkým chybám
- S rostoucí variancí chyb nemusí být RMSE kritérium nutně vyšší

### 3.4.4 Porovnání přesnosti modelů

Tato sekce se zabývá validací implementovaných energetických modelů (GN a MPR). Validace ověřuje adekvátnost (platnost, použitelnost) simulačního modelu s modelovaným systémem z hlediska stanoveného účelu [1].



Obrázek 3.3: Porovnání obou modelů s naměřenými hodnotami

Na obrázku 3.3 vidíme porovnání naměřených dat (modrou barvou) s GN modelem (zelenou barvou) a MPR modelem (červenou barvou). Z obrázku je patrné, že GN model lépe sleduje absolutní chyby a dokáže lépe zachytit relativně velké výkyvy hodnot. Naopak MPR model je méně citlivý na změny ve vstupních datech a tím pádem ukazuje "hladší" průběh. GN model vykazuje lepší výsledky podle kritéria MAE, MPR model se ukazuje být přesnější podle kritéria RMSE.

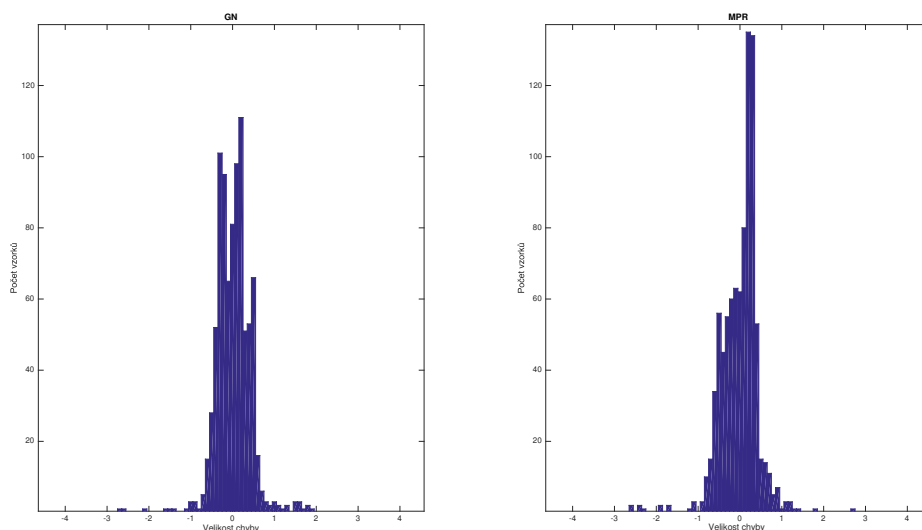
Tabulka 3.1: Tabulka porovnání srovnávacích kritérií pro modely GN a MPR

	RMSE	MAE
GN	0.4629	0.3119
MPR	0.4506	0.3342

V tabulce 3.1 vidíme průměrné odchylky modelů od naměřených hodnot podle kritérií RMSE a MAE. Jelikož kritérium RMSE přikládá vyšší váhu větším chybám,

vychází podle tohoto kritéria hůře model GN, který je schopen zachytávat velké výkyvy vstupních hodnot, ale nedokáže tak přesně kopírovat neustálé oscilace. Model MPR vykazuje lepší výsledky podle kritéria RMSE, ale horší podle kritéria MAE. To je způsobeno tím, že lépe sleduje malé oscilace, ale nedokáže tak přesně zachytit velkou změnu vstupních hodnot.

Oba modely by bylo možné vylepšit přidáním dalších regresních parametrů na úkor jednoduché struktury modelu.



Obrázek 3.4: Histogramy chyb obou modelů

Na obrázku 3.4 jsou vykreslené histogramy chyb jednotlivých modelů. Na ose  $x$  je velikost chyby a na ose  $y$  je počet vzorků s danou chybou. Z obrázku je patrné, že GN model má větší rozptyl chyb, ale menší počet extrémních hodnot. Naopak MPR model má menší rozptyl, ale větší množství extrémně velkých odchylek.

# Kapitola 4

## Závěr

Cílem této bakalářské práce bylo vytvoření jednotného systému pro sběr a zpracování energetických dat v areálu Fakulty aplikovaných věd Západočeské univerzity v Plzni a následné vytvoření energetických analytics vybraného energetického zařízení pracující nad tímto datovým úložištěm. Práce měla čtyři hlavní části - analýzu a zpřístupnění datových zdrojů ZČU obsahující měřená data energetických zařízení, analýzu a zvolení vhodných energetických zařízení pro modelování, návrh a implementaci vhodného rozhraní pro získání dat potřebných pro modelování, vytvoření a ověření modelů vybraných energetických zařízení.

V první části byla analyzována současná situace sběru energetických dat v areálu FAV. Energetická data jsou rozdělena ve dvou separovaných systémech, které jsou v režii dvou externích firem. Separování dat je pro praktické využití velice nevhodné a vznikla tak potřeba vytvořit konsolidované datové úložiště a tyto systémy sjednotit.

V další části po vytvoření společného datového úložiště vznikla potřeba naplnění tohoto úložiště historickými daty. Pro tyto účely byl vytvořen softwarový nástroj datová pumpa. Datová pumpa pracuje s univerzálním datovým rozhraním *Data-Row*. K softwaru lze jednoduše přidat komponenty, které zpracují a převedou data z libovolného externího datového zdroje na standardizovaný formát.

Pro práci s naměřenými daty v konsolidované databázi bylo potřeba vytvořit nástroj pracující nad touto databází, který převede data do standardizované podoby. Pro tyto účely byl v programovacím jazyce Java vytvořen nástroj *DAL - Data Access Layer*. DAL pracuje nad databází měření a pomocí matematických funkcí agregace a interpolace převádí data do standardizované ekvidistantní formy.

V návaznosti na všechny předchozí práce bylo možné v poslední části vytvořit energetické modely absorpčního chladicího stroje. V Matlabu byly implementovány dva energetické modely - *GN* a *MPR*. Identifikace parametrů obou modelů byla provedena lineární regresí minimalizující kritérium nejmenších čtverců. Pro porovnání odhadovaných hodnot z modelů s reálnými naměřenými hodnotami byla zavedena kritéria chyb RMSE a MAE.

Na základě implementovaných modelů je možné pozorovat vlivy vstupních veličin na chování systému. Toto pozorování je možné využít například pro optimalizaci využití elektrické energie a snížení nákladů na provoz absorpčního chladícího zařízení.



# Seznam obrázků

1.1	Schéma systému pro sběr energetických dat . . . . .	8
2.1	Struktura exportovaných CSV . . . . .	12
2.2	Struktura konsolidované databáze . . . . .	13
2.3	Ukázka aplikace Flexim . . . . .	15
2.4	Ukázka anotace datových bodů . . . . .	16
2.5	Obecné schéma datové pumpy . . . . .	17
2.6	Třída DataRow . . . . .	17
2.7	Schéma přenosu dat z telemetrické soustavy datovou pumpou . . . . .	18
2.8	UML schéma balíčků datové pumpy . . . . .	18
2.9	UML schéma balíčku datapump . . . . .	19
2.10	Class diagramy balíčků aggregation a interpolation . . . . .	21
2.11	Podrobný diagram tříd balíčku dataaccesslayer . . . . .	22
2.12	Vývojový diagram procesu agregace . . . . .	25
2.13	Vývojový diagram procesu interpolace . . . . .	26
2.14	Uměle vytvořený ukázkový signál . . . . .	27
2.15	Porovnání uměle vytvořeného signálu před a po zpracování data access layerem . . . . .	28
3.1	Schéma jednostupňového kontinuálního absorpčního chladicího zařízení [5] . . . . .	30
3.2	Screenshot schematu chladicího zařízení pro řízení z velína . . . . .	31
3.3	Porovnání obou modelů s naměřenými hodnotami . . . . .	35
3.4	Histogramy chyb obou modelů . . . . .	36

# Literatura

- [1] *Přednášky k předmětu Modelování a simulace 1.*
- [2] W3 Consortium. Sql tutorial. <https://www.w3schools.com/sql/>.
- [3] Pavel Herout. *Učebnice jazyka Java*. Kopp, 2010.
- [4] Alberto Coronas Jerko Labus, Joan Carles Bruno. Performance analysis of small capacity absorption chillers using defferent modeling methods. *Applied Thermal Engineering*, 2013.
- [5] Kristína Kazmárová. Ing., November 2011. <https://vetrani.tzb-info.cz/klimatizace-a-chlazení/7712-sorpcni-chladici-zarizeni>.
- [6] MathWorks. Explore matlab.
- [7] Pavel Novotný. Energetické modelování budov fakulty aplikovaných věd. mathesis, Západočeská univerzita v Plzni, May 2018.
- [8] Statistics How To. Rmse: Root mean square error, 2016. <http://www.statisticshowto.com/rmse/>.
- [9] Janet Wesner. Mae and rmse - which is better? *Human in Machine World*, 2016.
- [10] ZČU. *Učební text k předmětu Numerické metody*. ZČU.