

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

System pro sledování a zpracování dat ze solární elektrárny

Bc. Jan Štafa

Diplomová práce

2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan ŠTAFÁ**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Systém pro sledování a zpracování dat ze solární elektrárny**
Zadávací katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

V úvodní části je nutno provést rešerši stávajících řešení zpracovávání údajů poskytovaných měniči v solárních elektrárnách. Rešerši je nutné doplnit o porovnání s nově navrhovaným systémem, který bude předmětem této práce. Úvodní část musí obsahovat analýzu navrhovaného řešení, která bude obsahovat jak popis použitých hardwarových prostředků, tak návrh aplikačního řešení. Cílem této práce je vytvoření softwarového systému nad stávajícím hardwarovým řešením. Systém se bude skládat ze dvou aplikací. První bude zpracovávat data přicházející z měničů solární elektrárny prostřednictvím sériového rozhraní RS-485 a ukládat je do datového úložiště. Druhá aplikace bude zpracovávat uložená data a na základě zpracovaných dat bude předávat reporty o stavu a funkci elektrárny.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

GOOK M., Hardwarová rozhraní, Computer Press, 2006. NAGEL CH., EVJEN B., GLYNN J., WATSON K., SKINNER M., C 2008 Programujeme profesionálně, Computer Press, 2009. EVJEN B., HANSELMAN S., RADER D., ASP.NET 3.5 v jazycích C a Visual Basic, Computer Press, 2009. CONNOLLY T., BEGG C., HOLOWCZAK R., Mistrovství - Databáze, Profesionální průvodce tvorbou efektivních databází, Computer Press, 2009.

Vedoucí diplomové práce:

Ing. Miloslav Macháček
Katedra informačních technologií

Datum zadání diplomové práce: **30. října 2009**

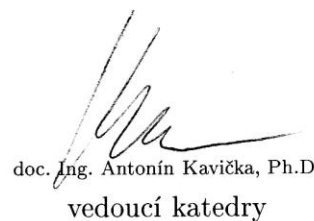
Termín odevzdání diplomové práce: **21. května 2010**



prof. Ing. Simeon Karamazov, Dr.

děkan

L.S.



doc. Ing. Antonín Kavička, Ph.D.

vedoucí katedry

V Pardubicích dne 10. listopadu 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 18. 5. 2010

Bc. Jan Štafa

SOUHRN

Práce se zabývá problematikou sledování a zpracování dat ze solárních elektráren se zaměřením na velké fotovoltaické systémy. Obsahuje ucelený koncept návrhu realizace monitorovacího systému pro solární elektrárnu. Dále se zabývá rozbohem komerčních řešení a jejich srovnání s navrhovaným systémem.

KLÍČOVÁ SLOVA

solární elektrárna, monitorování, fotovoltaika, sériová linka, protokol SMA, protokol SPINEL97

TITLE

A System for the Monitoring and Processing of Data from a Solar Power Plant

ABSTRACT

The work deals with the monitoring and processing of data from a solar power plant with a focus on big photovoltaic systems. It includes a comprehensive draft proposal for the realization of a monitoring system for a solar power plant. Further, it deals with the analysis of commercial solutions and their comparison with the proposed system.

KEYWORDS

Solar plant, monitoring, photovoltaic, serial port, protocol SMA, protocol SPINEL97

Obsah

1	Úvod	13
1.1	Úvod do fotovoltaiky	14
1.2	Princip fotovoltaického jevu	14
1.3	Fotovoltaická elektrárna	15
2	Rešerše stávajících řešení	17
2.1	SMA Sunny Webbox.....	17
2.2	AEG Solar Monitor.....	18
2.3	Závěr rešerše stávajících řešení.....	18
3	Úvodní studie	19
3.1	Odborný článek.....	19
3.2	Systémový kontextový diagram.....	21
3.2.1	Popis kontextového diagramu	21
3.3	Závěr úvodní studie	22
4	Analýza	23
4.1	Analýza požadavků na výsledný systém	23
4.1.1	Funkční požadavky klienta	23
4.1.2	Nefunkční požadavky klienta	25
4.1.3	Nefunkční požadavky serveru	29
4.2	Případy užití	30
4.2.1	Detailní pohled na diagram případu užití	31
4.2.2	Scénáře k vybraným případům užití.....	33
4.3	Datová analýza	35
4.3.1	Klientská část.....	35
4.3.2	Serverová část.....	37
4.3.3	Tabulka InventorTypes.....	40
4.4	Analýza dynamického chování	43
4.4.1	Klient – stavový diagram sběru dat	43
4.4.2	Klient – stavový diagram detekce zařízení.....	44
4.4.3	Klient – stavový diagram nastavení zřízení.....	44
4.4.4	Diagram spolupráce klient – server	45
5	Návrh	46
5.1	Návrh webových služeb	46

5.2	Komponenty klienta.....	47
5.2.1	Hlavní komponenta SolarMonitor	47
5.2.2	Komponenta SerialProtocols	48
5.2.3	Komponenta RestClient.....	48
5.2.4	Komponenta LocalDBLayer.....	49
5.2.5	Komponenta Deployment.....	49
5.3	Komponenty serveru	49
5.3.1	Komponenta Statistics	50
5.3.2	Komponenta Diagnostics.....	50
5.3.3	Komponenta Archive.....	50
5.4	Princip sběru dat ze zařízení.....	51
5.5	Rozbor sériového protokolu SMA	51
5.5.1	Příkaz pro start síťové konfigurace.....	52
5.5.2	Příkaz pro síťovou konfiguraci.....	53
5.5.3	Příkaz pro nastavení síťové adresy	53
5.5.4	Příkaz pro synchronizaci dat.....	53
5.5.5	Příkaz pro získání dat	54
5.5.6	Návrh principu sběru dat – protokol SMA	54
5.6	Rozbor sériového protokolu SPINEL97	55
5.6.1	Instrukce pro čtení parametrů	56
5.6.2	Instrukce pro čtení výrobních údajů	56
5.6.3	Instrukce pro nastavení vysílání	57
5.6.4	Instrukce pro čtení čítačů.....	57
5.6.5	Instrukce pro nastavení čítačů	57
5.6.6	Princip sběru dat – protokol SPINEL97	58
5.7	Návrh uživatelského rozhraní.....	59
5.7.1	Rozhraní klientské části.....	59
5.7.2	Rozhraní serverové části.....	59
5.8	Rozbor vybraných algoritmů a výpočtů	60
5.8.1	Výpočet aktuálního výkonu.....	60
5.8.2	Výpočet aktuálního zisku	61
5.8.3	Výpočet ztrát.....	61
5.8.4	Výpočet emisních povolenek.....	61
5.8.5	Algoritmus pro zjištění poruch	62

6	<i>Implementace</i>	63
6.1	Použité technologie	63
6.2	Použité nástroje	64
6.3	Popis implementace	64
6.3.1	Použité datové struktury	64
6.3.2	Komunikace pomocí sériové linky	65
6.3.3	Použité SQL dotazy	67
6.3.4	Pravidla použitá při implementaci	69
6.4	Vlastnosti klientské aplikace	70
6.4.1	Podpora různých sériových protokolů	70
6.4.2	Aktualizace	70
6.4.3	Licence.....	70
6.4.4	Kontrola spojení	71
6.4.5	Ochrana před dekompilací.....	71
6.5	Vlastnosti serverové části aplikace	71
6.5.1	Aktuální stavy.....	71
6.5.2	Statistické přehledy.....	72
6.5.3	Diagnostika.....	72
6.5.4	Archiv dat	72
7	<i>Závěr</i>	73
7.1	Doporučení pro další pokračování práce	73
7.2	Shrnutí dosažených výsledků	74
8	<i>Seznam použitých zdrojů</i>	76
8.1	Seznam literatury	76
8.2	Seznam internetových zdrojů	77
9	<i>Seznam příloh</i>	78

Seznam obrázků, tabulek a zkratk

Seznam obrázků

Obrázek 1 Kontextový diagram zobrazující zasazení systému do okolního prostředí.....	21
Obrázek 2 Obecný diagram případu užití.....	30
Obrázek 3 Detailní pohled na UC Sběr dat a UC Nastavení sběru dat	31
Obrázek 4 Detailní pohled na UC Diagnostika elektrárny a UC Práce s predikcemi	32
Obrázek 5 Detailní pohled na UC Statistické přehledy	32
Obrázek 6 Znárodnění vazby 1:N	37
Obrázek 7 ER diagram vztahů entit v databázi serveru.....	38
Obrázek 8 Stavový diagram zpracování datagramu	43
Obrázek 9 Stavový diagram detekce zařízení.....	44
Obrázek 10 Stavový diagram nastavení zařízení.....	44
Obrázek 11 Diagram spolupráce klient - server	45
Obrázek 12 Diagram komponent klientské části	48
Obrázek 13 Návaznost jednotlivých komponent v modelu MVC.....	50
Obrázek 14 Návrh uživatelského rozhraní klientské aplikace.....	59
Obrázek 15 Návrh GUI administračního rozhraní	60
Obrázek 16 Klient - zadání adresy serveru.....	80
Obrázek 17 Klient - zadání licenčního klíče	80
Obrázek 18 Klient - nastavení sériových portů	80
Obrázek 19 Server - úvodní strana administrace obsahující nejdůležitější informace.....	81
Obrázek 20 Server - Graf hodinového vývoje zisku v konkrétním dni.....	81
Obrázek 21 Server - úprava informací o měniči.....	82

Seznam tabulek

Tabulka 1 Detail atributů tabulky mon_sensor_box.....	36
Tabulka 2 Detail atributů tabulky mon_substation_items.....	36
Tabulka 3 Detail atributů tabulky mon_reporting.....	36
Tabulka 4 Detail atributů tabulky mon_monitoring.....	37
Tabulka 5 Detail atributů tabulky Licences.....	38
Tabulka 6 Detail atributů tabulky Sensorboxes.....	39
Tabulka 7 Detail atributů tabulky Reports	39
Tabulka 8 Detail atributů tabulky ReportTypes	39
Tabulka 9 Detail atributů tabulky Substations	40
Tabulka 10 Detail atributů tabulky Inventors.....	40
Tabulka 11 Detail atributů tabulky InventorTypes.....	41
Tabulka 12 Detail atributů tabulky PanelTypes	41
Tabulka 13 Detail atributů tabulky SensorboxDataRecords.	41
Tabulka 14 Detail atributů tabulky SubstationsDataRecords.....	42
Tabulka 15 Detail atributů tabulky InventorsDataRecords	42
Tabulka 16 Popis API serveru	47
Tabulka 17 Struktura SMA Net Telegramu	51
Tabulka 18 Popis struktury SMA Net Telegramu	52
Tabulka 19 Popis struktury SMA Data Telegramu vkládaného do SMA Net Telegramu	52
Tabulka 20 Příklad dotazu pro start síťové komunikace	52
Tabulka 21 Příklad odpovědi na dotaz pro start síťové dokumentace.....	52
Tabulka 22 Příklad dotazu pro síťovou konfiguraci.....	53
Tabulka 23 Příklad odpovědi na dotaz pro síťovou konfiguraci	53
Tabulka 24 Příklad dotazu pro nastavení síťové adresy.....	53
Tabulka 25 Příklad odpovědi na dotaz pro nastavení síťové adresy	53
Tabulka 26 Příklad dotazu pro synchronizaci dat	54

Tabulka 27 Příklad dotazu pro získání dat	54
Tabulka 28 Příklad odpovědi na dotaz pro získání dat.....	54
Tabulka 29 Popis formátu protokolu SPINEL97	56
Tabulka 30 Příklad instrukce Čtení komunikačních parametrů	56
Tabulka 31 Příklad odpovědi - adresa 04H, komunikační rychlost 9600Bd.....	56
Tabulka 32 Příklad instrukce Čtení výrobních údajů	56
Tabulka 33 Příklad odpovědi – číslo-výrobku 199 (=00C7H), sériové číslo 101 (=0065H) ...	56
Tabulka 34 Příklad aktivování samovolného vyslání zprávy; adresa 01H, podpis 02H:	57
Tabulka 35 Příklad odpovědi na aktivování samovolného vysílání	57
Tabulka 36 Automaticky odeslaná odpověď s daty	57
Tabulka 37 Příklad pro čtení čítačů.....	57
Tabulka 38 Odpověď na čtení čítačů (příklad z modulu Quido ETH 10/1).....	57
Tabulka 39 Příklad nastavení čítačů.....	58
Tabulka 40 Odpověď na nastavení čítačů – ok	58

Seznam zkratk

- ER – Entity-Relationship (vztah mezi entitami).
- PK – Primary Key (primární klíč).
- FK – Foreign Key (cizí klíč).
- HTTP – HyperText Transfer Protocol (hypertextový přenosový protokol).
- HTTPS – HyperText Transfer Protocol - Secure (bezpečná verze protokolu HTTP).
- IDE – Integrated Development Environment (integrované vývojové prostředí).
- MVC – Model View Controller (návrhový vzor).
- NN – Not Null (ne nulový).
- OOP – Object-Oriented Programming (objektově orientované programování).
- PDF – Portable Document Format (přenosný formát dokumentu).
- SQL – Structured Query Language (strukturovaný dotazovací jazyk).
- FTP – File Transfer Protokol (protokol pro přenos dat).
- Wp – Wattpeak, špičkový výkon.
- SMZD – systém pro monitorování a zpracování dat.
- GUI – Graphic User Interface (grafické uživatelské rozhraní).
- PV – PhotoVoltaic – fotovoltaiický.
- M2M – Machine2Machine – komunikace mezi zařízeními.
- EZS – elektronická zabezpečovací stanice.
- PAC – výstupní výkon [W] na AC straně měniče.
- PDC – výstupní výkon [W] na DC straně měniče.
- ETOTAL – celkový dodaný výkon měniče.
- AC – Alternating Current – střídavý proud.
- DC – Direct Current – stejnosměrný proud.

1 Úvod

Problematika výroby solární energie mě poprvé zaujala v době, kdy byla v místě mého bydliště postavena malá solární elektrárna. Zajímalo mne, jakým způsobem funguje výroba elektrické energie a jak je tato výroba sledována a vyhodnocována. Začal jsem si zjišťovat technologické postupy a metody monitorování. Bez patřičné dokumentace a fyzického přístupu k zařízení se mi můj záměr monitorovat solární elektrárnu však nezdařil. Proto jsem oslovil firmu zabývající se výstavbou solárních elektráren, se kterou jsem se domluvil na projektu monitorování fotovoltaických systémů. Díky této spolupráci jsem mohl nahlédnout pod „pokličku“ výroby solární energie a navrhnout a vyvinout systém monitorování a zpracování dat.

Mou další motivací byla možná práce na zajímavém projektu, který bude nasazen v provozu po dobu několika let. Chtěl jsem se začít seznamovat s praktickým vývojem softwaru, který bude opravdu používán v reálném prostředí a bude ho denně využívat několik desítek uživatelů. Zároveň jsem chtěl i zhodnotit své znalosti získané studiem na vysoké škole.

Cílem mé práce je návrh a implementace monitorovacího systému solární elektrárny, spolu s nastíněním možností jak získaná data zpracovat. Práce by měla sloužit jako základ pro vytvoření monitorovacího systému, který bude možno dále rozvíjet. Problematika monitorování solárních elektráren je velice rozsáhlá a zároveň stále ještě nedostatečně prozkoumaná část oboru informačních systémů, proto je zde velký prostor pro práce tohoto typu.

Ve své práci se zabývám analýzou sledování solárních elektráren, zejména pro koho je systém určen a jaké jsou funkční a nefunkční požadavky. Analyzuji také architekturu systému, datový model a dynamické chování některých částí systému.

Hlavní kapitoly jsou věnovány analýze, návrhu a vývoji vlastního řešení, kde se mimo jiné věnuji návrhu komponent systému, jejich spolupráci, rozboru klientské a serverové části s popisem vzájemné komunikace.

Jednou z kapitol této práce je návrh principu sběru dat ze zařízení umístěných na straně solární elektrárny, kde popisují jednotlivé postupy jak co efektivně a rychle získat potřebná data. Zabývám se také návrhem algoritmů pro vyhodnocování naměřených dat.

Práce kromě jiného obsahuje úvod do fotovoltaických systémů, dále pak srovnání komerčních řešení monitorování solární elektrárny s popisem jejich výhod a nevýhod.

Závěr je věnován zhodnocení dosažených výsledků, kde shrnuji přínosy své práce a následně uvádím další možnosti rozvoje systému pro monitorování a zpracování dat ze solární elektrárny, jimiž se budu zabývat v budoucnu.

1.1 Úvod do fotovoltaiky

Základním principem fotovoltaiky jsou sluneční paprsky dopadající na solární panely, ve kterých umístěné solární články přeměňují solární energii na stejnosměrný proud. Zařízení zvané měnič (někdy také inventar nebo střídač) poté stejnosměrný proud transformuje na střídavý, dále vedený do trafostanice, v níž je upraven na parametry rozvodné soustavy, do které je solární elektrárna připojena.

Použití fotovoltaických systémů je výhodné především tím, že sluneční záření je na celém světě zdarma. Fotovoltaické články mění v čase své vlastnosti, avšak výrobce obvykle garantuje minimální výkon 80 % po dobu 25 let. Panelům při jejich venkovním nainstalování nevadí déšť, sníh, kroupy ani hluboký mráz (provozní teplota je od $-30\text{ }^{\circ}\text{C}$ do $+90\text{ }^{\circ}\text{C}$, tuto teplotu však nevydrží měnič, jehož provozní teplota je od $-20\text{ }^{\circ}\text{C}$ do $+80\text{ }^{\circ}\text{C}$). Jejich provozu nepřekážejí ani vysoké teploty. Panely jsou otestovány v aerodynamickém tunelu pro rychlosti větru až $180\text{ km}\cdot\text{h}^{-1}$, teplotu $25\text{ }^{\circ}\text{C}$ a osvit $1000\text{ W}\cdot\text{m}^{-2}$.

Český Energetický regulační úřad k 19. lednu 2010 vykazuje celkový instalovaný výkon panelů 470 MW (na základě platných licencí na výrobu elektřiny ve fotovoltaických zdrojích). Do provozu bylo tedy uvedeno více jak 400 MW fotovoltaických zdrojů. Na konci roku 2010 se předpokládá nárůst instalovaného výkonu v solárních systémech minimálně na 1000 MW.

V České republice je řada firem, které se zabývají prodejem a montáží fotovoltaických systémů, výrobou solárních článků se zabývá například česká firma SOLARTEC s.r.o., nebo také původem japonská firma Kyocera, která má však u nás své obchodní zastoupení.

1.2 Princip fotovoltaického jevu

Slunečním zářením emitované fotony dopadají na křemíkové solární články, kde svojí energií vyvrážejí z krystalické mřížky elektrony. Ty se stávají volnými a jsou součástí elektrického proudu.

Fotovoltaický článek je obvykle tenká destička (0,5 mm) vyrobená z monokrystalu křemíku nejčastěji o velikosti 125x125 mm. Obě strany destičky jsou obohaceny o atomy vhodných prvků tak, aby jedna strana byla kladná a druhá záporná. Když na solární článek dopadají fotony, uvolňují se záporné elektrony, po kterých zůstávají kladně nabitá místa. Po přiložení elektrod k oběma stranám článku a spojení těchto elektrod začne vodičem protékat elektrický proud.

Existují monokrystalické články, které se skládají z jednoho krystalu křemíku o velikosti víc jak 10 cm vyráběné pomalým tažením roztaveného křemíku. Dále pak existují polykrystalické články, které se skládají z většího množství krystalů o velikosti 1 – 100 mm různě orientovaných. V neposlední řadě existují i články amorfni.

Běžný solární článek může při max. výkonu vytvořit napětí 0,5 V a elektrický proud až 3 A.

1.3 Fotovoltaická elektrárna

Solární elektrárna je souborem většího počtu solárních panelů, střídačů, ostatních konstrukčních a podpůrných prvků. Solární elektrárny se liší především svým výkonem, mají však obvykle stejný princip – energie vyrobená dopadem slunce na fotovoltaické panely se přemění v měničích na střídavé veličiny a poté je předána přes trafostanici do rozvodné elektrické sítě o kmitočtu 50 Hz.

Dle způsobu dodávky energie do elektrorozvodné sítě rozlišujeme dva základní typy fotovoltaických systémů, tedy:

- ostrovní systém (systém bez připojení na elektrorozvodnou síť),
- připojení na síť samostatnou přípojkou,

Základním stavebním kamenem fotovoltaické elektrárny jsou fotovoltaické panely, někdy také nazývané solární moduly. Tyto moduly jsou sestaveny ze vzájemně propojených solárních článků. Pokud je v jednom solárním modulu zasazeno například 36 článků, výstupní napětí je následně 18 V. Fotovoltaické články jsou v panelech určených pro velké solární systémy zapojeny sériově, což zvyšuje napětí. Výkon takto zapojených článků se zvyšuje paralelním zapojením stejných řetězců. Všechny články či panely dodávají stejnosměrné veličiny, tedy stejnosměrné napětí – stejnosměrný proud.

Na trhu je možno nalézt solární panel například s výkonem od 150 W až po 280 W. Výkon panelu je označován jako tzv. špičkový výkon – Wattpeak (Wp). Je to výkon naměřený za předem daných podmínek – ozáření $1000 \text{ W}\cdot\text{m}^{-2}$ a teplota $25 \text{ }^\circ\text{C}$. Účinnost solárních panelů se pohybuje v průměru mezi 14–17 %.

V praxi se využívají většinou monokrystalické fotovoltaické panely. Běžný solární panel je schopen vyrábět elektrickou energii i bez přímého osvětlení na základě difúzního záření, které je v ČR převládající. Dobře pohlcují difúzní záření amorfni panely, které mají životnost cca 12 let. K zajištění snadné manipulace s panelem při instalaci by jeho velikost neměla být větší než 2 m^2 . Orientace solárních panelů (zejména pak u stacionárních systémů) je jedním ze zá-

kladních parametrů potřebných ke stavbě slunečních elektráren. Přesné zaměření by mělo co nejvíce kopírovat slunce, lépe řečeno solární panel musí být nastaven jižním směrem – kolmo ke slunci a plus zhruba 5° na západní stranu.

V panelech je vyroben stejnosměrný proud, který se pro dodávku do sítě musí přeměnit na proud střídavý 230 V / 400 V, 50 Hz. Tuto konverzi provádí měnič, což je řídicí centrum solárního systému schopno podávat informace o vyrobené energii a provozu. Měnič je obvykle vybaven displejem zobrazujícím aktuální údaje o systému jako je například okamžitý výkon, napětí, vyprodukovanou energii za den, celkovou vyprodukovanou energii, dobu práce systému, případně poruchu a příčinu poruchy.

Nezbytnou součástí elektrárny je elektroměr měřící energii dodanou do distribuční sítě. Za každou kWh, která projde tímto elektroměrem, se účtuje distributorovi cena 12,89 Kč (cena v roce 2009 pro instalace s výkonem do 30 kWp – většina domácích instalací).

Největším problémem solárních systémů jsou tepelné ztráty vznikající v důsledku růstu teploty polovodiče, kdy se elektromagnetické záření přeměňuje s menší účinností (15–20 % při teplotě panelu 70°C), ztráty způsobeny ve vedení jsou 3–5 %, ztráty sklonem 3–5 %, ztráty při přeměně stejnosměrného napětí na střídavé napětí v měniči napětí jsou 2–8 %. S menší ztrátovostí se však počítá i v počátcích projektování (stejně jako u jiných druhů elektráren pracujících s obnovitelnými zdroji), odchylky je možné do jisté míry redukovat a jejich důsledky nemohou zásadně ovlivnit chod ani prosperitu fotovoltaických elektráren.

Výroba elektrické energie pomocí solárních elektráren je považována za ekologickou, protože při provozu elektrárny nevznikají škodlivé odpady ani emise oxidu uhličitého. Solární elektrárnu je však nutno z něčeho postavit, a při samotné výrobě solární elektrárny již vznikají vedlejší škodlivé produkty. Je potřeba před samotnou výstavbou solární elektrárny zvážit a rozhodnout, zda je tento obnovitelný zdroj energie pro nás opravdu ten pravý.

Po ukončení životnosti solárního systému je důležité všechny komponenty efektivně recyklovat a využít opět k nové výstavbě nebo v úplně jiném průmyslovém odvětví. Zejména recyklace solárních panelů a elektronických komponent elektrárny, jako jsou například měniče, je důležitá a je potřeba zjistit, zda firma dodávající tyto součásti odebírá případné recyklaci.

Samotná náročnost recyklace, například solárního panelu, se blíží náročnosti výroby panelu nového. Recyklace je však důležitá z jiného důvodu, již zmíněný panel obsahuje množství těžkých kovů, konkrétně podíl olova na hmotnosti panelů je přibližně 0,12 %, stříbra 0,14 %, cínu 0,12 % a mědi 0,37 %. Proto je nutné všechny komponenty solární elektrárny ekologicky zlikvidovat nebo opětovně recyklovat.

2 Rešerše stávajících řešení

V kapitole se věnuji popisu a shrnutí vlastností existujících komerčních řešení monitorování solárních systémů. Na trhu se pohybují zejména dvě firmy, které se zabývají komplexnějším monitorováním solárních elektráren. Jejich produkty, konkrétně vlastnosti a princip, jsou v této kapitole shrnuty a ohodnoceny.

2.1 SMA Sunny Webbox

SMA je přední světová firma, která vyvíjí, vyrábí a prodává měniče a monitorovací systémy pro fotovoltaická zařízení. Firma SMA má ve své nabídce několik typů monitorovacích zařízení, od jednoduchých zařízení určených pro domácnosti, až po komplexní řešení sledování velkých fotovoltaických parků. Pro účely rešerše je nejzajímavější zařízení, resp. systém s názvem SMA Sunny Webbox, který funguje na principu klient – server. Sunny Webbox na straně elektrárny sbírá data a v pravidelných časových intervalech je zabalí a zašle ke zpracování do centrály.

Při komunikaci s měniči je využíváno spojení pomocí sériového rozhraní RS485, kdy na jedné lince může být maximálně 50 těchto zařízení a délka vedení je maximálně 1219 metrů. Zařízení obsahuje síťovou kartu pro připojení do sítě internet. V momentě, kdy je klient připojen k internetu, můžeme jej nastavit tak, aby zasílal data na náš FTP server. Na serveru je již přítomen Sunny Portál, který data rozbalí a zpracuje. Tato aplikace je též dodávána od společnosti SMA. Je možno ji však nahradit vlastní aplikací.

Nevýhodou řešení od společnosti SMA je schopnost komunikovat pouze se zařízeními dané firmy, tedy s měniči, meteorologickými stanicemi a dalšími zařízeními. Zasílání dat na centrální server není šifrované, a proto může data kdokoli odposlechnout. Také interval zasílání dat je velice dlouhý (10 minut) a pro detailnější monitoring nevhodný.

Jako výhodu tohoto řešení vidím možnost vyvinout vlastní server pro zpracování dat a vcelku velký počet monitorovaných zařízení. Data nejsou uchovávána pouze na centrálním serveru, ale podle nastavení je může archivovat samotný klient. Velikost jeho paměťového prostoru je však omezena na 2GB. Webbox je jednoduše nastavitelný vzdáleně přes webové rozhraní. Systém od společnosti SMA je velice propracovaný a komplexní, jeho pořizovací cena tomu však odpovídá.

2.2 AEG Solar Monitor

Firma AEG je původem německá firma, její portfolio je velice rozsáhlé, protože je rozdělena na několik divizí. AEG však vyrábí i vybavení solárních elektráren, mezi tato zařízení patří také měniče, pro které AEG dodává vlastní řešení monitorovacího systému.

Monitorovací systém od společnosti AEG se jmenuje AEG Solar Monitor. Mezi jeho hlavní vlastnosti patří fakturace dle platné legislativy a v českém jazyce, dále ovládání internetovým prohlížečem, aktuální přehled o výrobě, upozornění na výpadek e-mailem nebo SMS, tvorba přehledných grafů, historie hodnot v distribuční síti. Solar Monitor nevyžaduje připojení k internetu, protože data jsou uložena lokálně.

Data ze Solar Monitoru se mohou stáhnout ve formátu CSV pro zpracování v uživatelské aplikaci, případně lze využít XML rozhraní pro M2M (Machine2Machine) aplikace. Díky tomu je možno si vytvořit vlastní centrální server, který bude sbírat data z těchto monitorů.

Na Solar Monitor lze dohlížet vzdáleně, pokud je připojen do sítě internet. Případně ho lze i přes vlastní webové rozhraní pohodlně nastavovat.

Monitorovány mohou být však jen maximálně tři střídače typu Protect PV. K zařízení lze napojit velké množství senzorů a dalších čidel, určených pro sledování například počasí. Komunikace s měniči probíhá pomocí sériového rozhraní RS485. Ke klientu je možno také připojit vstup S0 pro čtení dat z elektroměru.

Jako výhodu tohoto zařízení vidím právě zmiňovanou možnost napojení elektroměru, připojení senzorů pro sledování vlivů počasí a připravenost na českou legislativu. Velkou nevýhodou je možnost připojení pouze na měniče dané firmy a to ve velmi malém množství (velké PV systémy mají těchto měničů desítky). Cena celého systému se pohybuje v řádech desítek tisíc, proto by komplexní monitorování elektrárny bylo velice nákladné.

2.3 Závěr rešerše stávajících řešení

Analýza existujících řešení prokázala nedostatky na trhu v oblasti monitorování. Pro komplexní monitorování elektrárny jsou nabízené produkty dosti cenově nákladné, podporují jen omezený počet zařízení a jejich funkce nejsou dostačující. Díky relativně rychlému rozvoji solárních elektráren u nás je trh s monitorovacími systémy stále ještě nenasyčen, což potvrdila i úvodní rešerše komerčních produktů. Nabízená monitorovací řešení nejsou komplexní a nemusí vyhovovat potřebám investorů, takže je zde prostor pro vývoj vlastního systému, který bude splňovat náročné požadavky.

3 Úvodní studie

3.1 Odborný článek

System pro monitorování a zpracování dat ze solární elektrárny je komplexní systém pro sběr dat z různorodých zařízení a následné odesílání těchto dat k dalšímu zpracování do centrálního úložiště, kde jsou data analyzována a dlouhodobě uchována. Jedná se o distribuovaný systém využívaný velkým množstvím uživatelů, zejména investorů solární elektrárny.

Po úspěšné stavbě a zapojení elektrárny do provozu vzniká potřeba sledovat provozní stav elektrárny. Systém je orientován na sběr dat z fotovoltaických elektráren, komunikuje s různými měniči, elektroměry, meteorologickými stanicemi a s externími prvky zařízení. Přesněji řečeno, na straně elektrárny je instalováno hlavní zařízení nazvané SolarMonitor fungující jako klient. Komunikace mezi klientem a zařízeními elektrárny je koncipována pomocí sériové linky, která je pro svou výrobní cenu a technické vlastnosti jedno z nejideálnějších řešení sběru dat. Jednotka SolarMonitor komunikuje s celou řadou zařízení od různých výrobců, tyto zařízení jsou značky SMA, Fronius, Schüeco, Danfos, Kaco, Delta. Na SolarMonitor je napojena větev měničů, elektroměr, zabezpečovací stanice EZS, požární a pohybová čidla, meteorologická stanice včetně teploměru, termočlánu, čidla měření rychlosti větru, a měřič dopadající energie na metr plošný.

Z měničů získává SolarMonitor informace o celkové vyrobené energii, aktuálním výkonu, aktuálním provozním stavu, počtu nastalých událostí, výstupní frekvenci, celkovém součtu provozních hodin, síťovém napětí, výstupním výkonu a o izolačním odporu. V neposlední řadě je z měniče získáno sériové číslo a typ.

Meteorologická stanice poskytuje klientské jednotce informace o sériovém čísle, typu, aktuální intenzitě slunečního záření, rychlosti větru, teplotě okolí, teplotě termočlánu a době provozu. SolarMonitor je také napojen na trafostanici, ze které přijímá data o aktuálním množství dodané elektrické energie do rozvodné sítě a sériové číslo tohoto zařízení.

Výše uvedené informace SolarMonitor transformuje do požadovaného tvaru a ukládá do centrální databáze serveru SolarServer. Kromě sběru dat je klientská aplikace schopna se sama aktualizovat a obsahuje správu licencí, kdy po vypršení přiřazeného licenčního klíče je instalace znehodnocena. Zdrojový kód klienta je chráněn proti dekompilaci.

Aplikace SolarServer je schopna přijímat data i z jiných druhů sběrnic (zařízení třetích stran jako je WebBox, SunAnalyzer a jiné) a provádí hlubší analýzu a jednotlivou diagnostiku, tedy výpočet chyb, porovnání měničů, kontrola napětí u jednotlivých linek, výpočet účinností, ztrát na jednotlivých zařízeních, predikování výkonů, příjem reportů s chybovými hlášeními, analýza poruch klientských stanic, statistické vyhodnocení naměřených dat, archivace dat s detailním náhledem na jednotlivé naměřené informace.

Administrační rozhraní umožňuje definovat parametry elektrárny a jednotlivých zařízení, dále technický popis elektrárny, typ jednotlivých panelů, meteorologických stanice, měničů i trafostanic. Dají se zde přesně určit mezní hodnoty, teplotní koeficienty, vstupní údaje pro predikce výkonů, velikost činné plochy panelů, výkon panelů, tolerance panelů, výkonové charakteristiky daného druhu panelů, teplotního koeficientu, počty panelů na větev, počty větví na měnič a počty měničů.

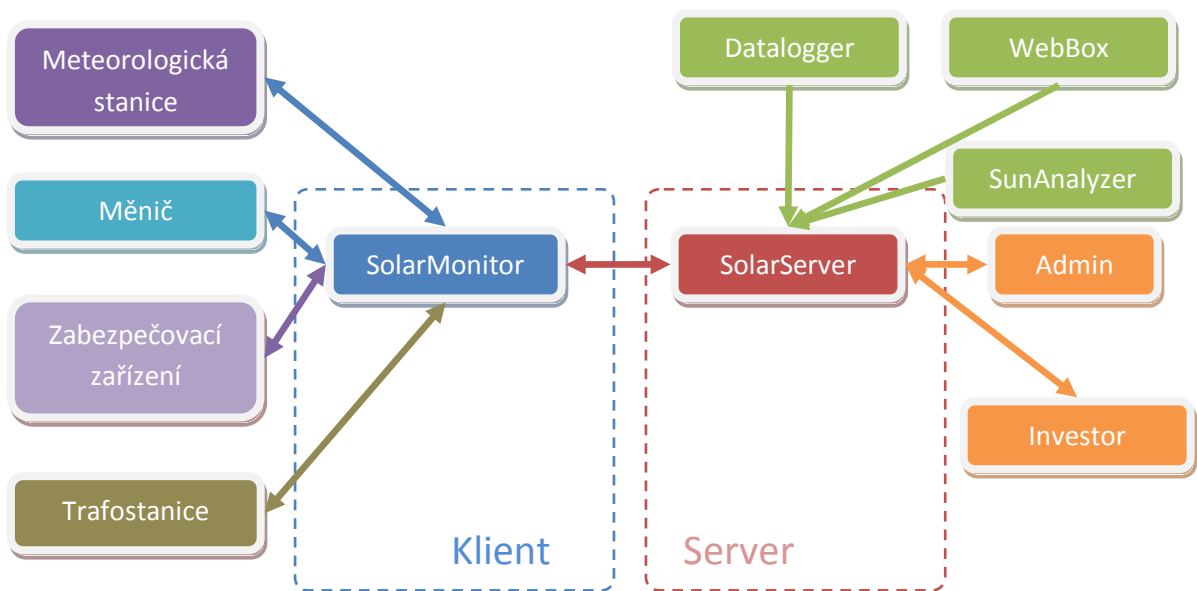
Diagnostická část systému SolarServer umožňuje sledovat aktuální stav jednotlivých měničů včetně možnosti popisu vzniklých chyb vztažených k projektovému číslu. SolarServer disponuje funkcí kontroly chybných měničů. Reporty vážných poruch se zaznamenají a odešlou prostřednictvím SMS a mailu dané zodpovědné osobě, případně investorovi.

Uživatelské rozhraní obsahuje podrobné zobrazení výkonu, zisku, ušetřených emisí, nejnovější reporty, připojení ilegálních měničů, výše zmíněná data z meteorologické stanice a předpověď počasí. Dále je v administraci obsažena šablona pro měsíční faktury.

Statistická analýza zobrazená uživateli na straně serveru obsahuje denní, měsíční a roční přehledy o vybraných veličinách, jako je výkon, zisk, množství ušetřených emisí. Tyto informace jsou zobrazeny formou přehledných sloupcových a líniových grafů s možností výpočtu minima, maxima a průměru případně celkové sumy. SolarServer se také stará o administraci licenčních klíčů a pravidelnou aktualizaci klientských jednotek.

Celkově je administrační rozhraní rychlé a přehledné. Každou tabulku je možno snadno řadit podle vybraného sloupce a záznamy v ní je možno procházet po stránkách. Pokud je potřeba změnit některý údaj o měniči, například sériové číslo, je toto provedeno přímo interaktivně v tabulce se záznamy. K administračnímu rozhraní se uživatel přihlašuje přes webové rozhraní s použitím protokolu HTTPS. Celá administrace je přístupná z běžného webového prohlížeče.

3.2 Systémový kontextový diagram



Obrázek 1 Kontextový diagram zobrazující zasazení systému do okolního prostředí

3.2.1 Popis kontextového diagramu

Systémový kontextový diagram je nejvíce abstraktní pohled na systém, ukazuje systém jako celek, jeho vstupy a výstupy od/k externím systémům.

Na obrázku 1 je vidět rozdělení systému na dvě části – klientskou (SolarMonitor) a serverovou část (SolarServer). Klientská část se zabývá sběrem dat a ovládáním jednotlivých zařízení na straně elektrárny. Serverová část přijímá i zpracovává data a může také zasílat požadavky klientské části. K serverové části mohou být dále připojeny systémy třetích stran (zobrazené zelenou barvou). Nejčastěji se však k SolarServeru připojují investoři, kteří mohou prostřednictvím SolarServeru kontrolovat funkčnost elektrárny a měnit případné parametry.

Množství datový kanálů, akcí a událostí je dosti velké, proto nemá smysl je nyní detailně popisovat. Obousměrné, resp. jednosměrné šipky znázorňují obousměrnou, resp. jednosměrnou komunikaci, datové toky nebo události, detailní popis je uveden v následující kapitole spolu s aktéry systému.

Aktéři Investor a Administrátor představují hlavního uživatele systému. Ostatní aktéři jsou spravovaná zařízení dodávající data nebo systémy třetích stran.

Entita SolarMonitor komunikuje jak s centrálním serverem tak i se zařízeními na straně elektrárny od kterých získává data a podle potřeby je nastavuje.

Klientem spravovaná zařízení na straně elektrárny:

- Meteorologická stanice – zařízení pro vyhodnocování povětrnostních podmínek, obsahuje senzor dopadající energie, senzor větru, termočlánek a další čidla.
- Měnič – zařízení převádějící stejnosměrný proud na střídavý. Obsahuje komunikační čip k zasílání provozních informací.
- Zabezpečovací zařízení – zařízení zajišťující bezpečnostní ochranu elektrárny před nedovoleným průnikem. Obsahuje čip pro zasílání informací o stavu ochrany.
- Trafostanice – spojuje vnitřní okruh elektrárny s rozvodnou sítí. Trafostanice zasílá impulzy, kdy jeden impulz je 100 W.

Aktér SolarServer přijímá i vyhodnocuje data a zasílá klientovi požadavky. Komunikuje s aktérem Investor a Administrátor. Může přijímat a vyhodnocovat data od systému třetích stran jako je například DataLogger, SunAnalyzer, WebBox a podobně.

Aktér Investor se připojuje k centrálnímu serveru za účelem zjištění stavu elektrárny, zobrazení statistických údajů a nastavení uživatelsky dostupných parametrů. Aktér Administrátor komunikuje s centrálním serverem za účelem nastavení administrátorsky dostupných parametrů, analýzy stavových hlášení, údržby systému, přípravy aktualizací systému.

3.3 Závěr úvodní studie

Po provedení úvodní studie jsou již všechny požadavky na systém známy a je tedy možno přistoupit k analýze systému.

Byl proveden popis požadavků formou odborného článku, který popsal výsledné funkce a vlastnosti systému. Dále byl vytvořen přehled o začlenění systému do okolního prostředí pomocí kontextového diagramu a jeho popisu.

Úvodní studie také definovala velice rozsáhlý systém, jehož analýza, návrh a implementace by svým rozsahem překonala možnosti této práce. Zejména implementace podpory zpracování dat ze zařízení třetích stran je velice náročná, a bylo by nutné všechna tato zařízení vlastnit, což je nejen finančně ale i technicky náročné. Proto budou následující kapitoly popisovat systém, tak aby výsledkem byl funkční základní systém pro monitorování a zpracování dat, na který je možno v budoucnu pohodlně rozšiřovat a vylepšovat.

V další kapitole se budu zabývat analýzou celého systému do takové úrovně, aby na jejím základě bylo možno vytvořit kompletní návrh aplikace, a poté přistoupím k její implementaci.

4 Analýza

V této kapitole se budu věnovat detailní analýze vlastního systému. Uvedu tedy seznam požadavků na výsledný systém spolu s analýzou případů užití, datovou analýzou a analýzou dynamického chování některých částí systému.

Předchozí kapitoly definovaly vlastnosti a funkce výsledného systému. V následujících kapitolách bude naopak popsáno, jak tyto funkce pracují.

4.1 Analýza požadavků na výsledný systém

V předchozí kapitole bylo popsáno, jaké vlastnosti bude systém mít a co přesně bude schopen provádět. Nyní se zaměřím na přesnou analýzu požadavků. Tedy na to, jak budou výše zmíněné funkce pracovat.

Nejdříve budou uvedeny funkční a nefunkční požadavky klientské části systému, a to formou katalogu, poté bude následovat analýza požadavků na serverovou část.

4.1.1 Funkční požadavky klienta

Funkční požadavky objasňují nutné úkony, aktivity, akce, události, které musí být vykonány při běhu klientské části systému. Následující přehled požadavků byl sestaven jak z úvodní studie, tak i v rámci komunikace se skutečnými uživateli budoucího systému.

4.1.1.1 Požadavky na design

- Klient umožňuje plnou konfiguraci sběru dat pomocí GUI.
- Klient ukládá nastavení do šifrovaného konfiguračního souboru.
- Při prvním spuštění je nutno zadat licenční klíč.
- Při dalším nastavení je nutno nastavit adresu serveru a sériové porty.
- Nastavení sériových portů obsahuje název portu, rychlost, protokol.
- Sériové porty je možno přidávat i odebírat a editovat.
- U protokolu trafostanice je možno nastavit počáteční hodnoty čítačů.
- Pokud je aplikace úspěšně spuštěna, tak je minimalizována do systémové části OS.
- Aplikaci je možno resetovat a nastavit znovu.

4.1.1.2 Požadavky na proces sběru dat

- Proces sběru dat je spuštěn automaticky po spuštění aplikace, pokud je vše v pořádku nastaveno.
- Sběr dat běží paralelně na všech nastavených sériových portech.
- Pokud se při sběru dat vyskytne chyba, musí být oznámena a zapsána do logovacího souboru. Chyba nesmí ovlivnit sběr dat na ostatních sériových portech ani nesmí způsobit pád programu.
- Při sběru dat musí být v poledne provedena přeregistrace všech zařízení (zejména měničů), aby byla zajištěna detekce nových připojených zařízení.
- Ze zařízení jsou sbírány informace o všech kanálech dle nastavení.
- Při zpracování příchozí odpovědi je kontrolováno pomocí sériového čísla, zda odpověď opravdu přišla od správného zařízení (pokud odpověď sériové číslo obsahuje).
- Při detekci neznámého typu zařízení je zaslán report.
- Pokud je aplikace spuštěna v noci (měniče nejsou online) cyklicky posílá žádost o detekci zařízení, dokud neobdrží odpověď. Po obdržení odpovědi počká 30 minut a následně začne samotný proces sběru dat.

4.1.1.3 Požadavky na ukládání dat

- Data jsou zasílána na centrální server pomocí webových služeb.
- Před uložením jsou data podle potřeby transformována do předem dohodnuté podoby (výpočet některých hodnot a podobně).
- Při výpadku spojení nebo nemožnosti uložit data na centrální server jsou data uložena do lokální databáze.
- Při nemožnosti uložit data na centrální server je vygenerován report, který je při nejbližší možné příležitosti zaslán na centrální server.
- Data jsou na centrální server ukládána postupně tak, jak jsou po příchodu ze sériové linky zpracována.

4.1.1.4 Požadavky na zálohu dat

Při nemožnosti uložit data na centrální server jsou data uložena na disku v rámci tzv. cache databáze. Z tohoto disku jsou pak při nejbližší možné příležitosti data přesouvána na server. Data musí být do lokální databáze uložena ve své nezměněné podobě a po úspěšném přesunu-

tí je záznam z lokální databáze smazán. Pokud se nepodaří přesunout některý záznam, je přesun zastaven a opakován při nejbližší možné příležitosti.

4.1.1.5 Požadavky na instalaci

Instalace programu musí být jednoduchá a nesmí vyžadovat žádné nezvyklé zásahy do operačního systému. Po instalaci musí být možno instalátor bez problémů odstranit a spustit nainstalovanou aplikaci. Postup instalace je následovný:

1. Správce nainstaluje podpůrné aplikace (framework, databázový systém, ...).
2. Po úspěšné instalaci podpůrných aplikací zahájí instalaci klienta.
3. Po úspěšné instalaci klienta spustí a nastaví.

4.1.1.6 Požadavky na aktualizaci

- Aktualizace je prováděna pravidelně, resp. pravidelně je kontrolováno, zda jsou k dispozici nové aktualizace.
- Aktualizovat je možno knihovny, interní databázi, konfigurační soubory a další komponenty systému.
- Aktualizace operačního systému je nezávislá na klientovi a je možno ji provádět podle potřeby.
- Po provedení aktualizace jsou všechny dočasné soubory smazány.
- Po provedení aktualizace je odeslán report o úspěšné aktualizaci.

4.1.2 Nefunkční požadavky klienta

Nefunkční požadavky se týkají použitých technologií, časových limitů, rychlosti odezvy systému a podobně. Jsou uváděny formou výčtu.

4.1.2.1 Požadavky na HW a SW

- Běžný počítač vyhovující požadavků .NET Framework 3.5 SP1.
 - Nejlépe MS Windows 7, 2GHz procesor, 2GB operační paměť.
- Stabilní připojení k internetové síti rychlostí min. 1Mb/s.
- Možnost přístupu přes vzdálenou plochu.
- Možnost připojení sériových linek.
- Není nutný monitor ani další periferie, nutné jsou pouze při prvotní instalaci.
- Záložní zdroj či jiný nezávislý a stabilní zdroj elektrické energie.

4.1.2.2 Požadavky na použité technologie

Aplikace může být naprogramována v libovolném vyšším programovacím jazyce, který umožňuje spojení přes sériovou linku a zasílání požadavků pomocí protokolu HTTPS. Použitý jazyk musí být moderní a rychlý. Výsledné knihovny programu musí být chráněny proti přečtení informací obsažených v těchto knihovnách (ochrana proti tzv. dekompilátorům).

Program pro svoji instalaci nesmí vyžadovat velké množství podpůrných aplikací třetích stran.

Výsledný systém bude schopný sbírat data z následujících zařízení:

- SMA měnič:
 - WR7KTL11
 - WR7KTL12
 - WR11TL08
 - WR09TL08
 - WRTL1EB9
- SMAS Sensorbox:
 - SENS0500
 - SENS0700
- Modul Quido SPINEL97:
 - RS 10/1

4.1.2.3 Výkonnostní požadavky

- Sběr dat musí být prováděn cyklicky v co nejrychleji na sebe navazujících intervalech.
- Získání dat z jednoho zařízení nesmí trvat déle než 5 sekund.
- Aktualizace systému musí být kontrolovány minimálně jednou za hodinu.
- Aktualizace nesmí trvat déle než 5 minut.
- Licence musí být kontrolována jednou denně, nejlépe v nočních hodinách.
- Klient musí být schopen přijímat data z neomezeného množství sériových linek.
- Klient musí být schopen uložit data na server během max. 1s.
- Klient musí být schopen ukládat do zálohy všechna neodeslaná data až do velikosti 2GB.
- Spuštění klienta nesmí trvat déle než 2 minuty.
- Vypnutí klienta nesmí trvat déle než 30 sekund.
- Instalace programu nesmí trvat déle než 5 minut.
- Záloha dat je na server přesouvána jednou za 5 minut.

- Kontrola spojení s centrálním serverem je prováděna jednou za 5 minut.
- Kontrola připojení do sítě internet je prováděna jednou za 5 minut.
- Kontrola připojení do sítě je prováděna jednou za 5 minut.

4.1.2.4 Požadavky na zabezpečení

- Nastavení klienta je uchovááno v zašifrovaném konfiguračním souboru.
- Aktualizační balíčky jsou chráněny heslem.
- Zjištění nových aktualizací je prováděno pomocí protokolu HTTPS.
- Veškerý datový přenos mezi klientem a serverem je uskutečněn pomocí protokolu HTTPS.
- Při komunikaci s centrálním serverem je klient ověřován pomocí jména a hesla.
- Lokální databáze je chráněna silným heslem a je šifrována.

4.1.2.5 Funkční požadavky serveru

Funkční požadavky na server definují aktivity, úkony, události, které musí serverová část aplikace provádět za svého běhu. Následující přehled požadavků je sestaven na základě úvodní studie a s pomocí diskuse s reálnými uživateli systému.

4.1.2.6 Požadavky na design

- Administrační rozhraní obsahuje přihlašovací formulář.
- Rozhraní je rozděleno na dvě části, vedlejší obsahující menu, hlavní obsahující obsah vybraných kategorií.
- Rozhraní je zobrazeno ve webovém prohlížeči, design co nejvíce připomíná běžnou desktopovou aplikaci, aby uživatel nebyl nucen k práci v neznámém prostředí.
- Design administrace obsahuje grafy, tabulky, vstupní formuláře.
- Administrace obsahuje možnost odhlášení a je zobrazena informace o přihlášeném uživateli a aktuálním času.

4.1.2.7 Požadavky na přehled údajů

- Úvodní stránka obsahuje přehled všech důležitých informací, jako je aktuální výkon, zisk, množství vyrobené energie, nejnovější reporty, nalezené ilegální měniče, předpověď počasí na další dny, historické grafy zisku.
- Jednotlivé sekce s informacemi si může uživatel podle potřeby přesouvat a řadit.
- Sekce s informacemi se dají odstranit.

4.1.2.8 Požadavky na statistickou analýzu dat

- Statistická analýza dat je zobrazena formou přehledných grafů.
- Analýza je měsíční, denní, roční.
- Grafy zobrazují průměrné hodnoty, minimum, maximum.
- U grafů jsou popsány osy X a Y a jsou zobrazeny jednotky.
- Grafy jsou sloupcové, případně liniové.
- Jsou analyzována data z měničů, meteorologických stanic a trafostanic.

4.1.2.9 Požadavky na diagnostiku

- Diagnostikovány jsou poruchy zařízení na straně elektrárny.
- Zobrazeny jsou případné chybové stavy zařízení.
- Zobrazeny jsou přijaté reporty z klientských jednotek.
- Data jsou zobrazena v přehledných tabulkách s možností řazení a stránkování.

4.1.2.10 Požadavky na archiv dat

- Data jsou uložena všechna, tak jak jsou přijata od klientských jednotek.
- Data je možno detailně procházet, řadit a stránkovat.
- Zobrazena jsou data o měničích, meteorologických stanicích a trafostanicích.
- Data v archivu není možno upravovat.

4.1.2.11 Požadavky na správu

- Spravovány jsou licence klientských jednotek. Je možno určit licenční klíč, platnost klíče od data do data, název klientské jednotky.
- Spravovány jsou zařízení na straně elektrárny.
- U měniče lze určit sériové číslo, datum legalizace, zda byl měnič legalizován, název měniče v projektovém plánu, typ měniče, počet připojených panelů, typ připojených panelů, výrobce měniče.
- U meteorologické stanice lze nastavit sériové číslo, typ a výrobce.
- U trafostanice lze nastavit sériové číslo.

4.1.3 Nefunkční požadavky serveru

Nefunkční požadavky obsahují seznam použitých technologií, časových limitů, rychlosti odezvy systému. Tyto požadavky jsou v následujících odstavcích uvedeny formou seznamu.

4.1.3.1 Požadavky na použité technologie

- Aplikace bude implementována pomocí technologie Microsoft ASP.NET MVC 2.
- SolarServer bude běžet na serveru Microsoft IIS.
- SolarServer bude využívat databázi Microsoft SQL Server 2005.
- Administrační rozhraní bude možno spustit v běžném internetovém prohlížeči.
- K implementaci administračního prostředí bude použit framework Ext JS.

4.1.3.2 Výkonnostní požadavky

- Spuštění administrace nebude trvat déle než 1 minutu při rychlosti internetového připojení 1 Mb/s.
- Zobrazení jednotlivých sekcí administrace nebude trvat déle než 30 sekund.
- Zjišťování nových dat bude prováděno každých 5 sekund.
- Administrace bude umožňovat souběžné připojení až 100 uživatelů.
- Odhlášení z administrace nebude trvat déle než 1 minutu.

4.1.3.3 Požadavky na zabezpečení

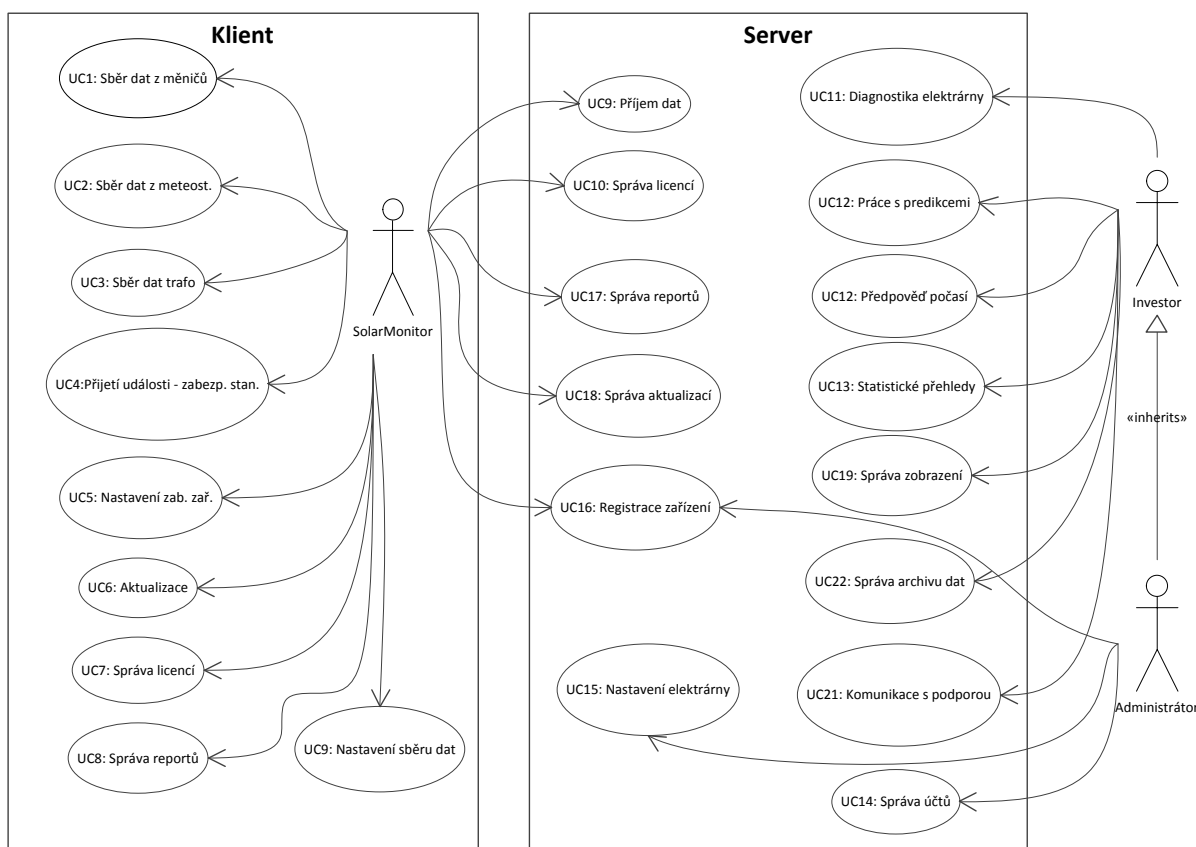
- Přístup do administrace je podmíněn platným jménem a heslem.
- Heslo musí být delší než 8 znaků a musí obsahovat kromě znaků i číslice.
- Přihlašování do administrace musí probíhat přes zabezpečený protokol HTTPS.

4.2 Případy užití

Kapitola popisuje případy užití systému pomocí tzv. grafů případů užití. Nejdříve je analýza provedena v menším detailu a později se vybranými případy užití zabýváme podrobněji. Diagramy vznikly postupnou analýzou úvodní rešerše a odborného článku.

Následující obrázky obsahují USE CASE diagramy, známé také jako diagramy případů užití. V diagramech jsou obsaženy informace o tom, jaké chování a funkce bude výsledný systém obsahovat, bez toho aniž by bylo odhaleno, jak budou tyto funkce realizovány.

Případy užití slouží k lepšímu pochopení funkčnosti systému, lépe definují požadavky na systém a specifikují jednotlivé role aktérů v systému. Umožňují rychlé zjištění hranic systému. Při tvorbě katalogu požadavků vycházím právě z těchto případů užití a samozřejmě i z rešerše, odborného článku a jiných materiálů.

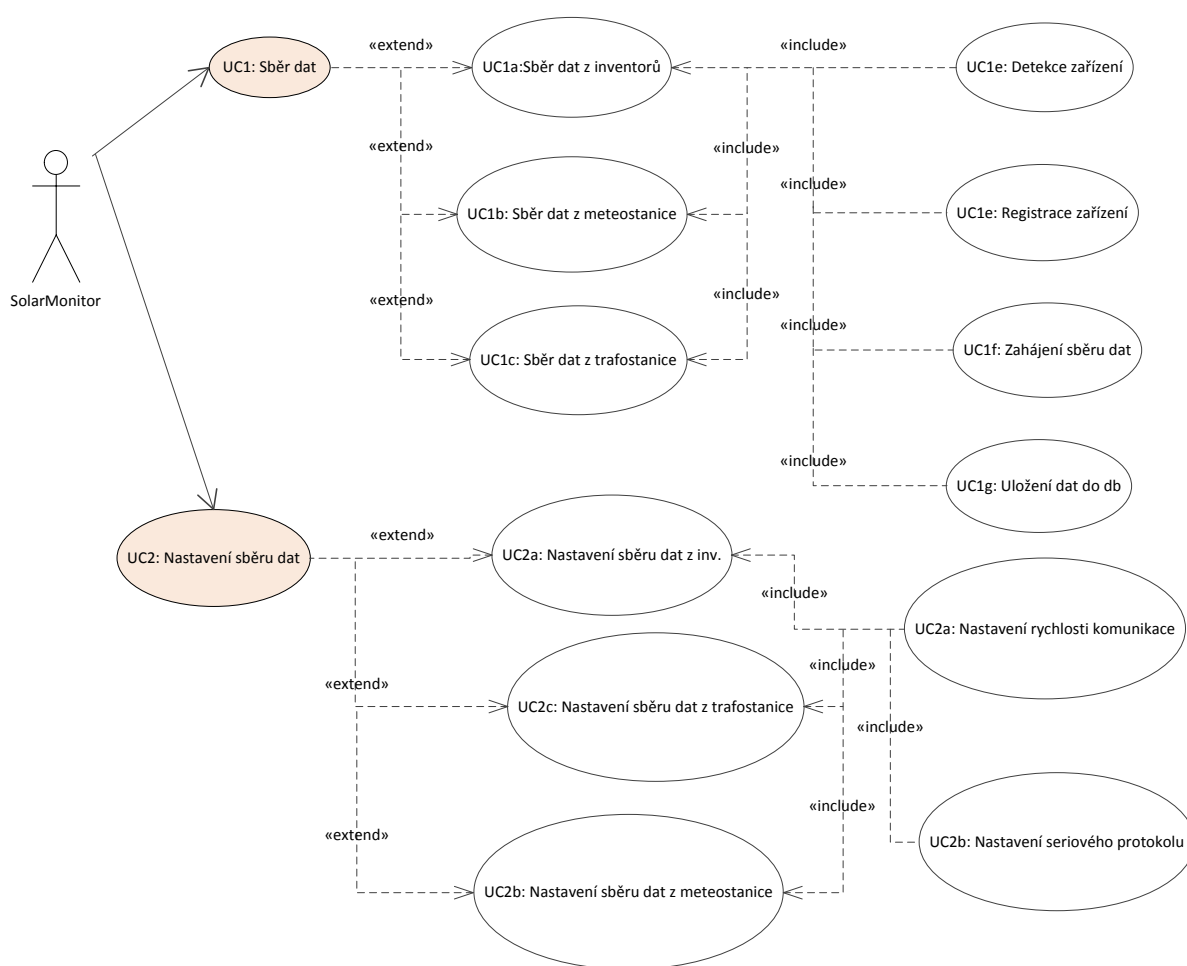


Obrázek 2 Obecný diagram případu užití

Na obrázku 2 je zobrazena poměrně nízká úroveň zobrazující činnosti nad systémem v kontextu uživatelských rolí a aktérů systému. Důležité je však rozdělení systému na klientskou a serverovou část a vzájemná interakce. Výše uvedený diagram obsahuje velké množství funkcí systému, detailní popis a implementace všech by překročila rozsah této práce, proto budou v následujících kapitolách podrobně rozpracovány jen důležité funkce systému.

4.2.1 Detailní pohled na diagram případu užití

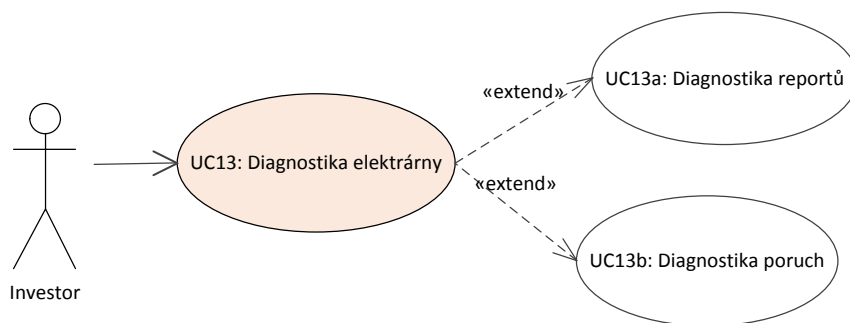
Obecný pohled uvedený dříve neobsahuje všechny důležité detaily potřebné k vypracování důkladné analýzy a později také ke správné implementaci návrhu. Proto následuje detailní pohled na vybrané případy užití, který poskytne lepší představu o rozsahu systému a jeho vlastnostech. Na obrázku 3 je znázorněn aktér SolarMonitor, který využívá funkcí pro sběr a nastavení. Sběr dat je rozšířen o možnosti sběru dat z různých zařízení, jako je měnič, meteorologická stanice a trafostanice. Jednotlivé sběry dat zahrnují detekci zařízení, registraci zařízení, zahájení sběru dat a hlavně uložení dat do centrální databáze. V rámci nastavení sběru dat je opět možno nastavit různá zařízení, konkrétně měnič, trafostanici a meteorologickou stanici. U těchto zařízení se určuje zejména rychlost komunikace a využitý sériový protokol.



Obrázek 3 Detailní pohled na UC Sběr dat a UC Nastavení sběru dat

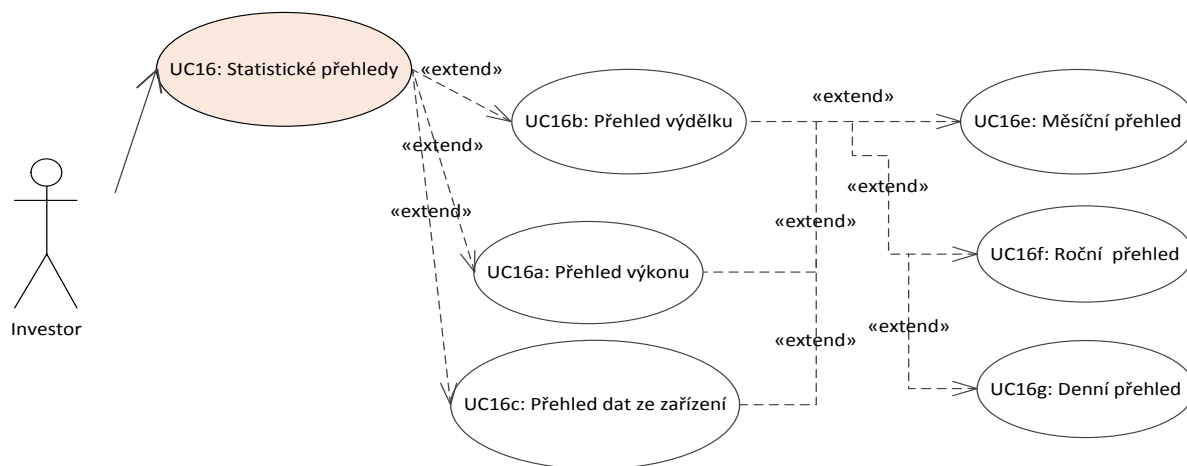
Obrázek 4 obsahuje diagram případu využití diagnostických funkcí serverové části systému. Diagnostický vlastnosti využívá aktér v podobě investora, který si může vybrat, jaké části chce diagnostikovat. Pokud investora zajímá přehled nejnovějších reportů zaslaných z monitorovacích jednotek do centrální databáze, zvolí diagnostiku reportů. Jestliže je investor inte-

resován spíše ve sledování poruch jednotlivých zařízení na straně solární elektrárny, zvolí diagnostiku poruch.



Obrázek 4 Detailní pohled na UC Diagnostika elektrárny a UC Práce s predikcemi

Serverová část systému obsahuje také funkce pro statistické přehledy. Následující obrázek 5 znázorňuje diagram případu užití pro konkrétní funkce statistických přehledů. Aktér investor využívá statistické přehledy, které zahrnují přehledy výdělku, výkonu, dat ze zařízení. Jednotlivé přehledy zahrnují také možnost zvolit si denní, měsíční nebo roční přehled.



Obrázek 5 Detailní pohled na UC Statistické přehledy

Po nastínění případů užití navrhovaného systému je nutné k jednotlivým případům sepsat scénáře. Následující kapitoly budou popisovat vybrané scénáře případů užití, které blíže popíší funkce systému, zejména nastíní, jak budou tyto funkce vykonávány a jaké úkony budou v rámci jejich zpracování prováděny. Hlavní myšlenkou scénářů případů užití je již blíže navrhnout vnitřní logiku systému. Díky scénářům bude návrh jednotlivých komponent systému jednodušší a zejména implementace bude snazší.

4.2.2 Scénáře k vybraným případům užití

V této kapitole se věnuji scénářům k vybraným případům užití. Vybrány jsou ty případy užití, které mají složitější průběh a jsou z pohledu této práce zajímavé.

4.2.2.1 Sběr dat

1. Sběr dat je spuštěn automaticky nebo servisním pracovníkem.
2. Po zahájení probíhá detekce zařízení.
3. Pokud byla nalezena zařízení, tak jsou zaregistrována.
4. Pokračuje detekce a registrace zařízení, dokud není nalezeno nové zařízení.
5. Cyklicky je vysílán požadavek na nová data.
6. Při sběru dat ze zabezpečovací stanice je pouze nasloucháno na sériové lince a data jsou zpracovávána.

4.2.2.2 Nastavení sběru dat

1. Z roletkového menu je vybrán sériový port.
2. Portu je nastaven protokol a rychlost.
3. Pokud se jedná o protokol trafostanice, jsou v dialogovém okně nastaveny počáteční hodnoty čítačů.

4.2.2.3 Nastavení zařízení

1. Nejdříve je vybrána sériová linka, na které se nastavované zařízení nachází.
2. Následně je vyhledáno nastavované zařízení.
3. Je odeslán požadavek na nastavení.
4. Přijata odpověď o výsledku nastavení.

4.2.2.4 Aktualizace klienta

1. Pravidelná kontrola nových aktualizací na straně klienta.
2. Klient se přihlásí pomocí jména a hesla na centrální server.
3. Pokud je zjištěna nová aktualizace, je stažen aktualizací balík.
4. Po rozbalení aktualizací balíku je spuštěn aktualizací program.
5. Po aktualizaci je smazán aktualizací balík a spuštěn sběr dat.

4.2.2.5 Správa licencí

1. Na straně klienta probíhá pravidelná kontrola platnosti licence.
2. Klient se přihlásí pomocí jména a hesla na centrální server.
3. Pokud je licence úspěšně ověřena tak může aplikace pokračovat v práci.
4. Pokud se nepodaří licenci ověřit, tak je tato informace uložena do logu a ověření se provede znovu.
5. Pokud je licence neplatná, je tato informace uložena do logu a program se vypne a znehodnotí.

4.2.2.6 Správa reportů

1. Systém je schopen generovat reporty a ukládat je do centrální databáze.
2. Pokud se nepodaří odeslat report je tato informace uložena do logu a report je uložen do interní databáze.
3. Při nejbližší příležitosti je report odeslán znovu.

4.2.2.7 Příjem dat

1. Data jsou přijímána přes jednotné API pomocí webové služby.
2. Server přijme data.
3. Data jsou ověřena.
4. Data jsou převedena do potřebné podoby
5. Data jsou uložena.
6. Pokud se některá z výše uvedených procedur nezdaří, je klientu vrácen odpovídající stavový kód s informační zprávou.

4.2.2.8 Registrace zařízení

1. Pokud server zjistí, že přišel požadavek na uložení dat od neznámého zařízení, tak je toto zařízení vedeno v databázi jako nelegální.
2. Administrátor se přihlásí do systému a vyplní potřebné legalizační údaje u daného zařízení.
3. Pokud zařízení nelze zlegalizovat, je kontaktována technická podpora.

4.2.2.9 Diagnostika elektrárny

1. Uživatel zvolí v administraci sekci „Diagnostika“.
2. Zvolí, zda chce diagnostikovat „Poruchy“, „Reporty“.
3. Uživateli jsou zobrazeny diagnostické informace.

4.2.2.10 Statistické přehledy

1. Uživatel v administraci zvolí „Statistiky“.
2. Uživateli je nabídnuta možnost zvolit „Přehled výkonu“, „Přehled výdělku“, „Přehled dat ze zařízení“.
3. Data jsou uživateli zobrazena za daný den, měsíc a rok.
4. Uživatel si vybere konkrétní časový úsek.
5. Uživatel agregační funkci pro určení minima, maxima, průměru.

4.3 Datová analýza

V kapitole se budu věnovat popisu datových úložišť klientské části a serverové části. K analýze je využit ER diagram a přehledové tabulky. V ER diagramu jsou vynechány pro přehlednost informace o datových typech sloupců. Jsou zobrazeny pouze názvy entit, vazby mezi entitami a primární klíče. Podrobnější popis entit s komentáři k jednotlivým sloupcům je proveden v přehledových tabulkách.

4.3.1 Klientská část

Klientská část systému využívá databázi k dočasnému uložení dat, která z nějakého důvodu nešla přesunout na server. Struktura databáze je tedy navržena k co nejrychlejšímu uložení a následovněm velice rychlému výběru a odstranění přesunutých záznamů. Návrh databáze proto v některých ohledech nesplňuje 3NF. Tabulky mezi sebou nemají žádnou vazbu, proto zde nemá, smysl zobrazovat ER diagram a rovnou je uveden detailní popis jednotlivých tabulek.

4.3.1.1 Tabulka *mon_sensor_box*

Tabulka *mon_sensor_box* obsahuje záložní informace o naměřených datech, které se nepodařilo uložit do centrální databáze. V tabulce 1 jsou informace jak o meteorologické stanici, ze které byla data získána, tak i konkrétní naměřená data.

Atribut	Typ	Klíč ¹	Hodnota ²	Komentář
mon_sensorbox_id	bigint	PK	NN	
mon_sensorbox_serial_number	int		NN	Sériové číslo.
mon_sensorbox_intsolirr	float			Intenzita slunečního záření.
mon_sensorbox_optm	float			Celková provozní doba.
mon_sensorbox_tmp_amb_c	float			Teplota okolí.

¹ Klíč označuje, zda je sloupec primárním klíčem (PK) nebo cizím klíčem (FK).

² Hodnota označuje, zda sloupec nemůže nabývat hodnoty NULL (značeno NN – Not NULL) nebo může (prázdná buňka).

mon_sensorbox_tmp_mdul_c	float			Teplota modulu.
mon_sensorbox_wind_vel	float			Rychlost větru.
mon_sensorbox_type	nvarchar(50)			Typ meteorologické stanice.
mon_sensorbox_timestamp	datetime			Datum a čas získání záznamu.

Tabulka 1 Detail atributů tabulky mon_sensor_box

4.3.1.2 Tabulka mon_substation_items

Tabulka *mon_substation_items* obsahuje záložní informace o naměřených datech, které se nepodařilo uložit do centrální databáze. V tabulce 2 jsou obsaženy informace o trafostanici jako je její sériové číslo, tak i konkrétní naměřená data.

Atribut	Typ	Klíč	Hodnota	Komentář
mon_substation_items_id	bigint	PK	NN	
mon_substation_serial_number	int		NN	
mon_substation_items_input1 až mon_substation_items_input10	float(8)			
mon_substation_items_timestamp	datetime			

Tabulka 2 Detail atributů tabulky mon_substation_items

4.3.1.3 Tabulka mon_reporting

Tabulka *mon_reporting* obsahuje záložní informace o vzniklých reportech, které se nepodařilo uložit do centrální databáze. Následující tabulka 3 popisuje sloupce tabulky *mon_reporting*.

Atribut	Typ	Klíč	Hodnota	Komentář
mon_reporting_id	bigint(8)	PK	NN	
mon_reporting_id_error_code	int(4)			
mon_reporting_type	nvarchar(100)			
mon_reporting_id_type_device	int (4)			
mon_reporting_priority	int (4)			
mon_reporting_note	ntext			
mon_reporting_date_record	datetime(8)			
mon_reporting_device_identification	nvarchar(100)			

Tabulka 3 Detail atributů tabulky mon_reporting

4.3.1.4 Tabulka mon_monitoring

Tabulka *mon_monitoring* obsahuje záložní informace o datech z měničů, které se nepodařilo uložit do centrální databáze. V tabulce 4 jsou informace o měniči, ze kterého byla data získána. Další sloupce obsahují konkrétní naměřená data. Tabulka *mon_monitoring* je velice obsáhlá a kopíruje svoji strukturu uložení dat na serveru, může obsahovat více hodnot, nejsou zasilány všechny.

Atribut	Typ	Klíč	Hodnota	Komentář
mon_monitoring_id	bigint(8)	PK	NN	
mon_inventors_serial_number	int			Sériové číslo.
mon_monitoring_di	float(8)			Diferenční proud. zařízení.
mon_monitoring_e_total	float(8)			Suma dodané energie.
mon_monitoring_fac	float(8)			Frekvence za měničem.
mon_monitoring_h_on	float(8)			Celkový součet prov. hodin.
mon_monitoring_h_total	float(8)			Součet hod. v režimu dodávky.
mon_monitoring_netz_ein	float(8)			Celkový součet připojení do sítě.
mon_monitoring_pac	float(8)			Výstupní výkon [W].
mon_monitoring_riso	float(8)			Izol. odpor před napojením do sítě.
mon_monitoring_status	float(8)			Indikace akt. provozního stavu.
mon_monitoring_uac	float(8)			Střídavé napětí [V].
mon_monitoring_u_fan	float(8)			Napájecí napětí ventilátoru [V].
mon_monitoring_upv_ist	float(8)			Vstupní napětí na stringu [V].
mon_monitoring_upv_soll	float(8)			Požadované napětí [V].
mon_monitoring_time	datetime			Datum a čas získání záznamu.
mon_monitoring_type	nvarchar(20)			Typ zařízení.

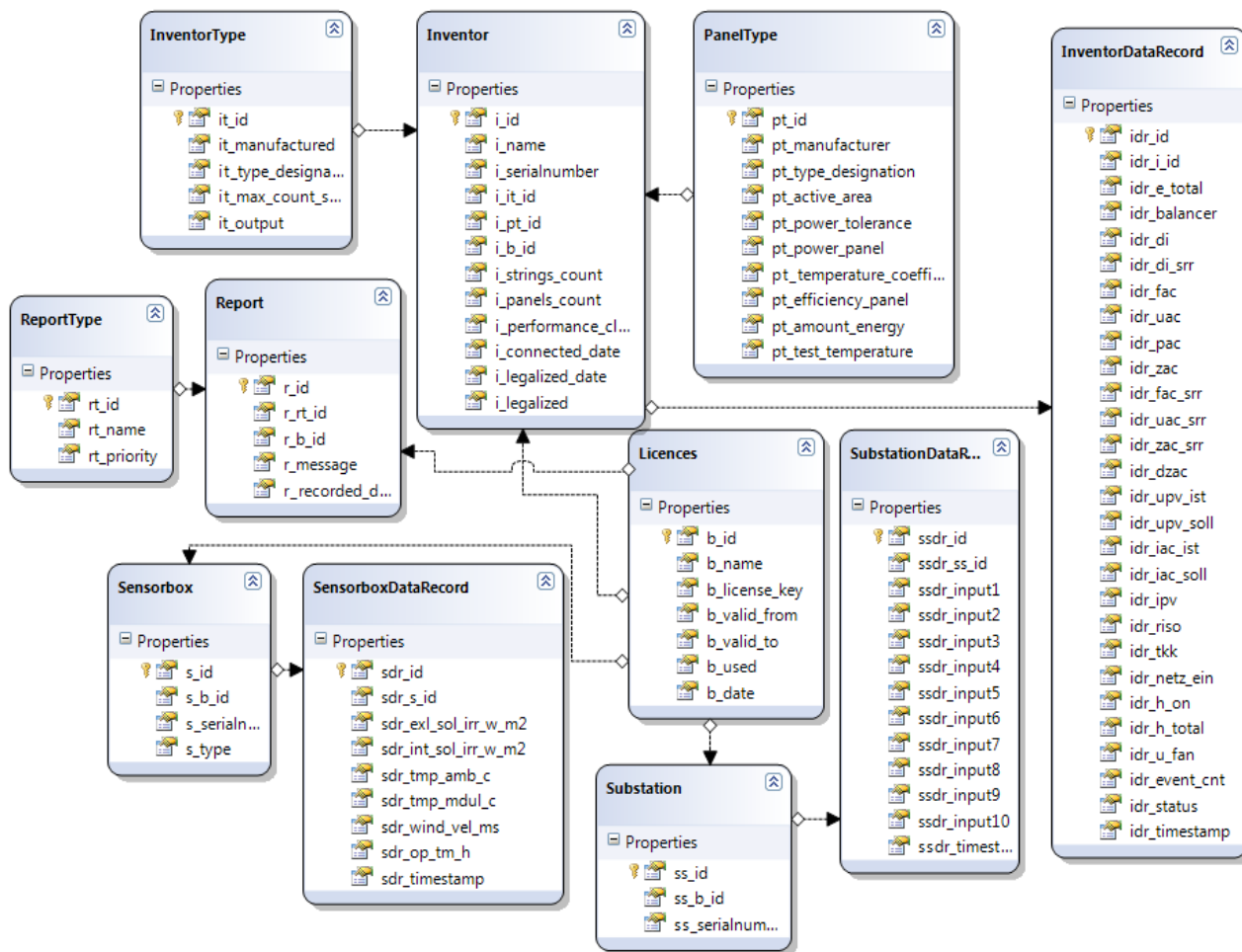
Tabulka 4 Detail atributů tabulky mon_monitoring

4.3.2 Serverová část

Serverová část systému využívá dvě databáze. Jednu s názvem ASPNETDB, která je využívána k ukládání údajů o uživatelích serveru, jejich účtech, právech, rolích a podobně. Tato databáze byla pro zjednodušení práce automaticky vygenerována vývojovým nástrojem Visual Studio 2008 a je velice rozsáhlá. Druhá databáze je navržena přímo pro potřeby monitorovacího systému a nese pojmenování SOLARSERVER. Toto oddělení přináší několik výhod, zejména přehlednější správu i přenositelnost na jiný aplikační server. Jelikož databáze „ASPNETDB“ není z pohledu práce tolik důležitá, nebude v této části popisována. Detailně bude popsána databáze serveru, která má pro tuto práci větší význam. V této databázi jsou uchována naměřená data, z nichž se vytváří analýzy a statistiky. Databáze obsahuje například informace o jednotlivých měničích, jednotlivé naměřené záznamy ze zařízení, informace o licencích klientských jednotek, atd. Všechny tabulky obsahují primární klíč využívající automatické inkrementace. Při pojmenování sloupců jsou využity prefixy k zajištění unikátnosti názvu sloupce v rámci databáze. Prefix je vytvářen z prvního znaku názvu tabulky, někdy je však rozšířen i o další znaky v závislosti na názvu tabulky. Vazby mezi tabulkami znázorňuje ER diagram na obrázku 7, kde vazba 1:N je znázorněna dle obrázku 6.



Obrázek 6 Znázornění vazby 1:N



Obrázek 7 ER diagram vztahů entit v databázi serveru

4.3.2.1 Tabulka Licences

Tabulka *Licences* obsahuje informace o licencích, které jsou investorem zakoupené a jsou potřeba k provozování monitorování elektrárny. Z této tabulky není možno řádky mazat, pokud jsou na záznamy této tabulky vázány nějaké jiné záznamy v dalších tabulkách. Tabulka má nastavena referenční omezení, která mazání zabraňují. Funkce jednotlivých sloupců jsou popsány v následující tabulce 5.

Atribut	Typ	Klíč	Hodnota	Komentář
b_id	smallint	PK	NN	
b_name	nvarchar(50)		NN	Název SolarMonitoru
b_license_key	nvarchar(50)		NN	Licenční klíč.
b_valid_from	datetime		NN	Klíč je validní od data.
b_valid_to	datetime		NN	Klíč je validní do data.
b_used	bit			Byl klíč využit (0 = ne, 1 = ano)?
b_date	datetime			Datum vystavení licence.

Tabulka 5 Detail atributů tabulky Licences

4.3.2.2 Tabulka Sensorboxes

Tabulka *Sensorboxes* obsahuje informace o meteorologických stanicích nainstalovaných na straně elektrárny. Pokud je na záznamy této tabulky odkazováno z jiných tabulek, není možno záznamy díky referenčnímu omezení mazat. Tabulka je vázána na tabulku *Licences* a je popsána v tabulce 6.

Atribut	Typ	Klíč	Hodnota	Komentář
s_id	smallint	PK	NN	
s_b_id	smallint	FK	NN	Cizí klíč tabulky Licences.
s_serialnumber	int		NN	Sériové číslo.
s_type	nvarchar(50)		NN	Typ meteorologické stanice.

Tabulka 6 Detail atributů tabulky Sensorboxes

4.3.2.3 Tabulka Reports

Tabulka *Reports* obsahuje data reportů o stavu systému. Tabulka využívá data uložená v tabulce *ReportTypes*. Reporty jsou důležité informace o nejnovějších stavech solární elektrárny. Každý report je určen svým typem, od kterého se odvíjí priorita takového reportu. Doprovodnou informací reportu je také čas zaslání a hlavně obsah reportu ve formě textové zprávy.

Atribut	Typ	Klíč	Hodnota	Komentář
r_id	bigint	PK	NN	
r_rt_id	smallint	FK	NN	Cizí klíč tabulky ReportTypes.
r_b_id	smallint	FK	NN	Cizí klíč tabulky Licences..
r_message	text		NN	Zpráva obsažena v reportu.
r_recorded_date	datetime		NN	Datum vytvoření reportu.

Tabulka 7 Detail atributů tabulky Reports

4.3.2.4 Tabulka ReportTypes

Tabulka *ReportTypes* obsahuje informace o typech reportů. Jelikož jsou obvykle záznamy této tabulky zatíženy referenčním omezením, nelze je obvykle mazat bez zrušení vazeb. Tabulka má nastavena referenční omezení, která mazání zabraňují. Sloupce jsou popsány v tabulce 8.

Atribut	Typ	Klíč	Hodnota	Komentář
rt_id	smallint	PK	NN	
rt_name	nvarchar(50)		NN	Název typu reportu.
rt_priority	smallint		NN	Priorita reportu v systému.

Tabulka 8 Detail atributů tabulky ReportTypes

4.3.2.5 Tabulka Substations

Tabulka *Substations* obsahuje informace o trafostanicích připojených na straně elektrárny. Pokud jsou na záznamy této tabulky vázány nějaké jiné záznamy v dalších tabulkách, nelze z této tabulky řádky mazat. Tabulka je vázána na tabulku *Licences*.

Atribut	Typ	Klíč	Hodnota	Komentář
ss_id	smallint	PK	NN	
ss_b_id	smallint	FK	NN	Cizí klíč na tabulku Licences.
ss_serialnumber	int		NN	

Tabulka 9 Detail atributů tabulky Substations

4.3.2.6 Tabulka Inventors

Tabulka *Inventors* obsahuje informace o měničích připojených na straně elektrárny. Tabulka má nastavena referenční omezení, která zabraňují odstranění řádků, pokud jsou tyto záznamy odkazovány z jiných tabulek. Tabulka je vázána na tabulky *Licences*, *InventorTypes* a *PanelTypes*. Záznamy obsažené v této tabulce jsou nositeli důležitých informací o jednotlivých měničích na straně elektrárny. Hlavní informací je sériové číslo měniče, spolu s informací o legalizaci a datu připojení do vnitřní sítě elektrárny. Další poskytované informace o měniči je jeho výkonová třída, počet připojených solárních panelů a počet jednotlivých stringů s panely. Díky vazbám na další tabulky je možno zjistit informace o výrobci měniče a konkrétním typu.

Atribut	Typ	Klíč	Hodnota	Komentář
i_id	smallint	PK	NN	
i_serialnumber	int		NN	Sériové číslo měniče.
i_it_id	smallint	FK	NN	Cizí klíč na tabulku InventorTypes.
i_pt_id	smallint	FK	NN	Cizí klíč na tabulku PanelTypes.
i_b_id	smallint	FK	NN	Cizí klíč na tabulku Licences.
i_strings_count	smallint		NN	Počet stringů s panely.
i_panels_count	smallint		NN	Počet panelů.
i_performance_class	nvarchar(50)		NN	Výkonová třída měniče.
i_connected_date	datetime		NN	Datum připojení do gridu.
i_legalized_date	datetime		NN	Datum legalizace (zanesení do projektu).
i_legalized	bit		NN	Byl měnič legalizován (0 = ne, 1 = ano)?

Tabulka 10 Detail atributů tabulky Inventors

4.3.3 Tabulka InventorTypes

Tabulka *InventorTypes* obsahuje informace o typech měničů, které existují v systému. Z tabulky nelze díky referenčnímu omezení mazat, obvykle je totiž na tuto tabulku odkazováno z jiných tabulek. Tabulka obsahuje informace o výrobcích jednotlivých měničů. Zejména o typu, maximálním počtu připojitelných stringů se solárními panely a výstupním napětím.

Atribut	Typ	Klíč	Hodnota	Komentář
it_id	smallint	PK	NN	
it_manufactured	nvarchar(255)		NN	Výrobce.
it_type_designation	nvarchar(255)		NN	Typ měniče.
it_max_count_strings	smallint		NN	Maximální počet stringů s panely.
it_output	int		NN	Výstupní napětí.

Tabulka 11 Detail atributů tabulky **InventorTypes**

4.3.3.1 Tabulka PanelTypes

Tabulka *PanelTypes* obsahuje informace o typech panelů, které existují v systému. Informace jsou důležité zejména při určování výrobce panelu, aktivní plochy panelu, výkonové tolerance a výkonu panelu.

Atribut	Typ	Klíč	Hodnota	Komentář
pt_id	smallint	PK	NN	
pt_manufacturer	nvarchar(50)		NN	Výrobce panelu.
pt_type_designation	nvarchar(50)		NN	Typ panelu.
pt_active_area	float		NN	Činná plocha panelu [mm ²].
pt_power_tolerance	smallint		NN	Tolerance výkonu [%].
pt_power_panel	smallint		NN	Výkon panelu [Wp].
pt_temperature_coefficient	float		NN	Teplotní výkonový koeficient.
pt_efficiency_panel	float		NN	Účinnost panelu [%].
pt_amount_energy	smallint		NN	Množství dopadající energie [W].
pt_test_temperature	smallint		NN	Testovací teplota [°C].

Tabulka 12 Detail atributů tabulky **PanelTypes**

4.3.3.2 Tabulka SensorboxDataRecords

Tabulka *SensorboxDataRecords* obsahuje datové záznamy pořízené při monitorování solární elektrárny a to konkrétně z meteorologické stanice. Důležitými informacemi v tabulce jsou hlavně sloupce obsahující intenzitu osvětlení, rychlost větru, teplotu okolí a modulu, tedy fotovoltaického článku. Každý záznam obsahuje časové razítko, které určuje, kdy přesně byl záznam ze zařízení získán. Záznamy není doporučeno mazat ani upravovat.

Atribut	Typ	Klíč	Hodnota	Komentář
sdr_id	bigint	PK	NN	
sdr_s_id	smallint		NN	Cizí klíč na tabulku Sensorboxes.
sdr_int_sol_irr_w_m2	float			Množství dopadajícího světla [W.m ⁻²].
sdr_tmp_amb_c	float			Teplota okolí [°C].
sdr_tmp_mdul_c	float			Teplota modulu [°C].
sdr_wind_vel_ms	float			Rychlost větru [m.s ⁻¹].
sdr_op_tm_h	float			Provozní čas [h].
sdr_timestamp	datetime		NN	Časový otisk záznamu.

Tabulka 13 Detail atributů tabulky **SensorboxDataRecords**.

4.3.3.3 Tabulka SubstationsDataRecords

Tabulka *SubstationsDataRecords* obsahuje datové záznamy pořízené při monitorování solární elektrárny a to konkrétně z trafostanice. Sloupce této tabulky obsahují hodnoty naměřené na jednotlivých vstupech modulu Quido, o kterém bude řeč později.

Atribut	Typ	Klíč	Hodnota	Komentář
ssdr_id	bigint	PK	NN	
ssdr_ss_id	smallint	FK	NN	Cizí klíč na tabulku Substations.
ssdr_input1 až ssdr_input10	int			Vstup z trafostanice.
ssdr_timestamp	datetime		NN	Časový otisk záznamu.

Tabulka 14 Detail atributů tabulky SubstationsDataRecords

4.3.3.4 Tabulka InventorsDataRecords

Tabulka *InventorsDataRecords* obsahuje datové záznamy pořízené při monitorování solární elektrárny a to konkrétně z měničů. Každý záznam je opatřen časovým razítkem, určujícím dobu získání dat ze zařízení.

Atribut	Typ	Klíč	Hodnota	Komentář
idr_id	bigint	PK	NN	
idr_i_id	smallint	FK	NN	Cizí klíč na tabulku Inventors.
idr_e_total	float			Suma dodané energie
idr_performance	float			Aktuální výkon.
idr_di	float			Diferenční proud zařízení.
idr_fac	float			Frekvence za měničem.
idr_uac	float			Střídavé napětí [V].
idr_pac	float			Výstupní výkon [W].
idr_upv_ist	float			Vstupní napětí na stringu [V].
idr_upv_soll	float			Požadované napětí [V].
idr_riso	float			Izolační odpor před napojením do sítě [Ohm].
idr_netz_ein	float			Celkový součet připojení do sítě.
idr_h_on	float			Celkový součet provozních hodin.
idr_h_total	float			Celk. součet prov. hod. v režimu dodavky do sítě.
idr_u_fan	float			Napájecí napětí ventilátoru [V].
idr_event_cnt	int			Počet nastalých událostí.
idr_status	int			Indikace aktuálního provozního stavu.
idr_timestamp	datetime			Časový otisk záznamu.

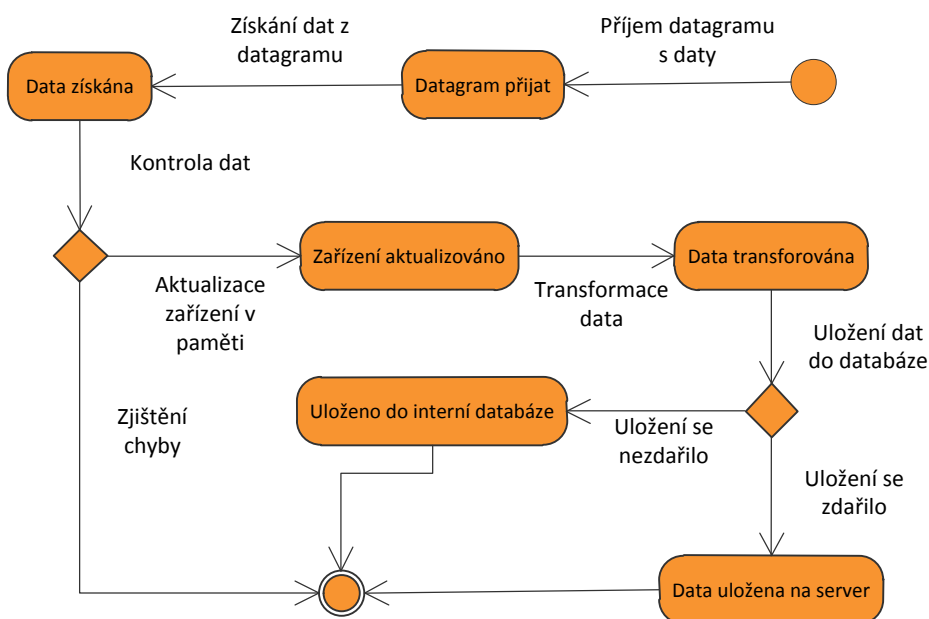
Tabulka 15 Detail atributů tabulky InventorsDataRecords

4.4 Analýza dynamického chování

Analyzovány budou dynamické části systému, a to s pomocí dynamických UML diagramů. K analýze jsou vybrány takové části, které jsou z pohledu zadání práce zajímavé a jejich chování je netriviální.

4.4.1 Klient – stavový diagram sběru dat

Z pohledu klienta je zajímavou částí sběr dat, kdy systém prochází různými stavy. Na obrázku 8 je znázorněn stavovým diagramem proces příjmu datagramu od zařízení, zpracování dat a uložení dat na centrální server.

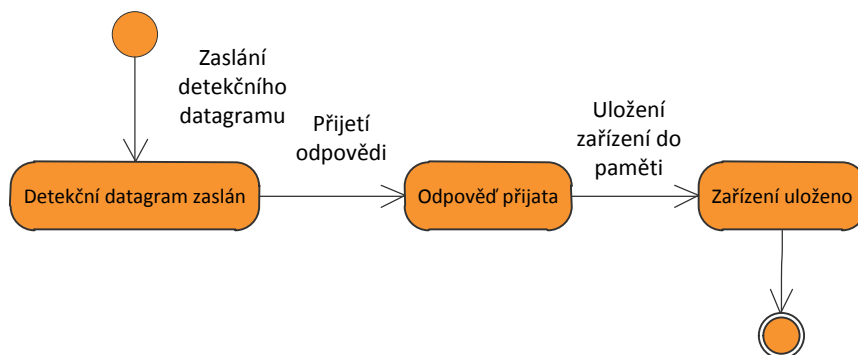


Obrázek 8 Stavový diagram zpracování datagramu

Nejdříve je přijat datagram obsahující data ze zařízení. Systém je ve stavu přijetí datagramu. Z datagramu jsou získána data. Data jsou následně validována, zda jsou správně doručena. Pokud ano, provede se aktualizace informací o zařízení na základě těchto dat. Poté jsou aktualizací data transformována pro potřeby uložení do centrální databáze a následně také uložena. Pokud se nepodaří data odeslat na server, jsou dočasně uložena v lokální databázi. Cyklus zpracování a uložení dat je následně ukončen.

4.4.2 Klient – stavový diagram detekce zařízení

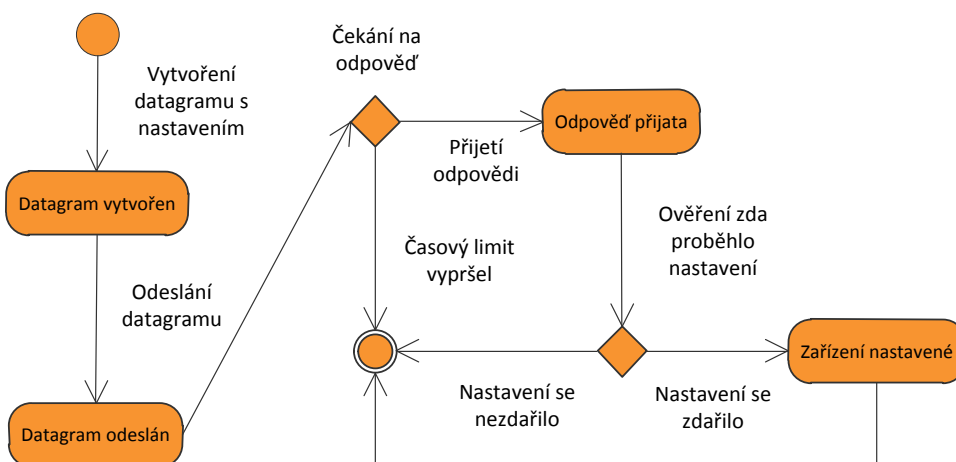
Před samotným sběrem dat ze zařízení je však nutné toto zařízení detekovat. Jak takový proces probíhá, znázorňuje stavový diagram na obrázku 9. Nejdříve je zaslán detekční datagram, na který je přijata od zařízení odpověď. Zařízení je na základě své odpovědi uloženo do paměti a detekce je ukončena.



Obrázek 9 Stavový diagram detekce zařízení

4.4.3 Klient – stavový diagram nastavení zřízení

Po provedení činnosti znázorněné v předchozím stavovém diagramu jsou zařízení detekována a uložena v paměti. Zařízení potřebují být nastavena (síťová adresa nebo jiné další parametry) před samotným zahájením sběru dat. Na obrázku 10 je znázorněn stavový diagram nastavení zařízení.



Obrázek 10 Stavový diagram nastavení zařízení

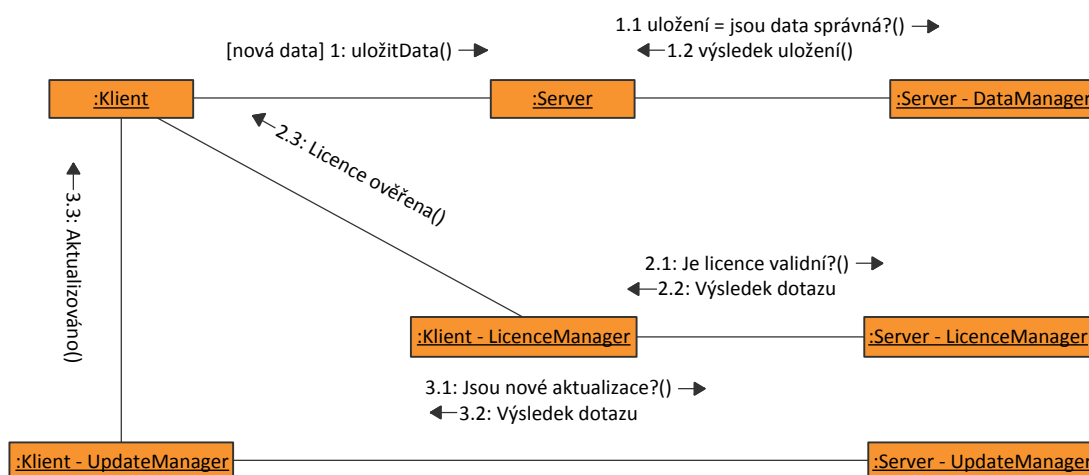
Na začátku je vytvořen datagram s nastavením. Tento datagram je odeslán a je na něj očekávána odpověď. Pokud je přijata odpověď, je ověřeno, zda úspěšně proběhlo nastavení, násled-

ně je zařízení považováno za nastavené. Při vypršení časového limitu na přijetí odpovědi nebo chyby v nastavení je proces ukončen.

4.4.4 Diagram spolupráce klient – server

Diagram spolupráce je řazen mezi interakční diagramy a klade si za cíl zobrazit uspořádání jednotlivých spolupracujících objektů. K lepšímu pochopení spolupráce mezi klientskou a serverovou částí slouží digram na obrázku 11.

Na straně klienta se nachází objekt *Klient*, který obsahuje objekty *LicenceManager* a *UpdateManager*. Na straně serveru je vlastní *Server* s objekty *DataManager*, *LicenceManager* a *UpdateManager*. Zprávy související se stejným objektem mají na začátku stejnou číslici.



Obrázek 11 Diagram spolupráce klient - server

Ve chvíli, kdy má *Klient* připravena data k odeslání na server, odešle zprávu 1. Touto zprávou požádá server o uložení dat. Na serveru přijme data *DataManager*, který data ověří, uloží do databáze a vrátí výsledek uložení.

Na straně klienta běží *LicenceManager*, který se stará o ověřování licence a pravidelně žádá zprávou 2 o ověření licence na serveru. *LicenceManager* na serveru licenci ověří a vrátí zprávu výsledkem operace. *LicenceManager* na straně klienta oznámí ověření licence *Klientu*.

Klientský *UpdateManager* se stará o pravidelnou kontrolu aktualizací na serveru. *UpdateManager* vytvoří zprávu 3, pomocí které se detekuje, zda jsou na serveru nové aktualizace. *UpdateManager* na serveru vrátí informace s výsledkem kontroly. *Klient UpdateManager* předá zprávu o nových aktualizacích *Klientu*.

Z výše uvedeného diagramu tedy vyplývá snaha o co nejmenší interakci mezi klientem a serverem, aby byla zajištěna minimalizace přenosu dat a zvýšena stabilita celého systému.

5 Návrh

Kapitola je věnována návrhu systému, tedy klientské části i serverové části. Dále je v této kapitole obsažen rozbor sériových rozhraní, tak aby je bylo možno použít v implementaci. Kapitola také obsahuje návrh webových služeb pro komunikaci mezi klientem a serverem a popis jednotlivých komponent systému. V kapitole je také navržen princip sběru dat ze zařízení na straně elektrárny. V neposlední řadě se také zabývám návrhem algoritmů pro vyhodnocení získaných dat a jejich zobrazení uživateli.

5.1 Návrh webových služeb

Při komunikaci mezi klientem a serverem bylo využito tzv. webových služeb. Princip webových služeb je založen na vzdáleném volání funkcí ze serveru, předávání parametrů a zpracování vrácených výsledků. Ke komunikaci se používá protokol HTTP. Nezáleží na tom, v jakém programovacím jazyce je webová služba napsaná, ani z jakého jazyka jsou procedury volány. Data se mezi serverem a klientem zasílají nejčastěji ve formátu XML, klient a server se však musí předem dohodnout na tvaru přijatých a zasláných dat.

Princip webových služeb perfektně odpovídá potřebám této práce, proto jsem je také implementoval. Místo standardních webových služeb založených například na SOAP využívám architekturu REST (Representational State Transfer). Ta je orientována oproti XML-RPC nebo SOAP datově, ne procedurálně. Proto je REST výhodnější pro časté zasílání a získávání dat.

Architektura REST definuje čtyři základní metody, které jsou označeny jako CRUD, tedy vytvoření dat (Create), získání dat (Retrieve), změna dat (Update) a smazání dat (Delete). CRUD metody jsou implementovány pomocí metod HTTP protokolu. Navržené API serveru je popsáno v následující tabulce Tabulka 16 a obsahuje několik metod pro práci s daty na straně serveru i pro získávání informací.

K ukládání dat slouží funkce začínající prefixem *save*. Je možno uložit data z měniče, trafostanice, meteorologické stanice a data o vzniklé události – report. Po provedení akce je vrácen výsledek v podobě odpovídajícího stavového kódu, tedy 200. Obsahem odpovědi je také hodnota *true* nebo *false*. Pokud dojde při zpracování požadavku k chybě, je zaslána chybová zpráva s odpovídajícím stavovým kódem.

Data ze serveru lze také získávat, například pomocí funkce *getLicence()*, která přijímá jako parametr licenční klíč a vrací pole obsahující informace o vyžádané licenci.

Název	Metoda	Parametry	Návratová hodnota	Popis
saveInventor	POST	Array data[]	Bool true/false	Uložení dat z měniče. Vrací true pokud se uložení provede v pořádku, v opačném případě false.
saveSensorbox	POST	Array data[]	Bool true/false	Uložení dat z měniče. Vrací true pokud se uložení provede v pořádku, v opačném případě false.
saveReport	POST	Array data[]	Bool true/false	Uložení dat z měniče. Vrací true pokud se uložení provede v pořádku, v opačném případě false.
saveSubstation	POST	Array data[]	Bool true/false	Uložení dat z měniče. Vrací true pokud se uložení provede v pořádku, v opačném případě false.
getLicence	GET	String licenční klíč	Array licence-Data[]	Získání informací o licenci na základě klíče. Vrací pole obsahující data o licenci.
updateLicenceUsed	PUT	String licenční klíč	Bool true/false	Nastavení licence jako použité na základě licenčního klíče.

Tabulka 16 Popis API serveru

Příklad volání metody *getLicence*:

```
GET /rest/get/?action=getLicence&key=xxx
Host: example.com
```

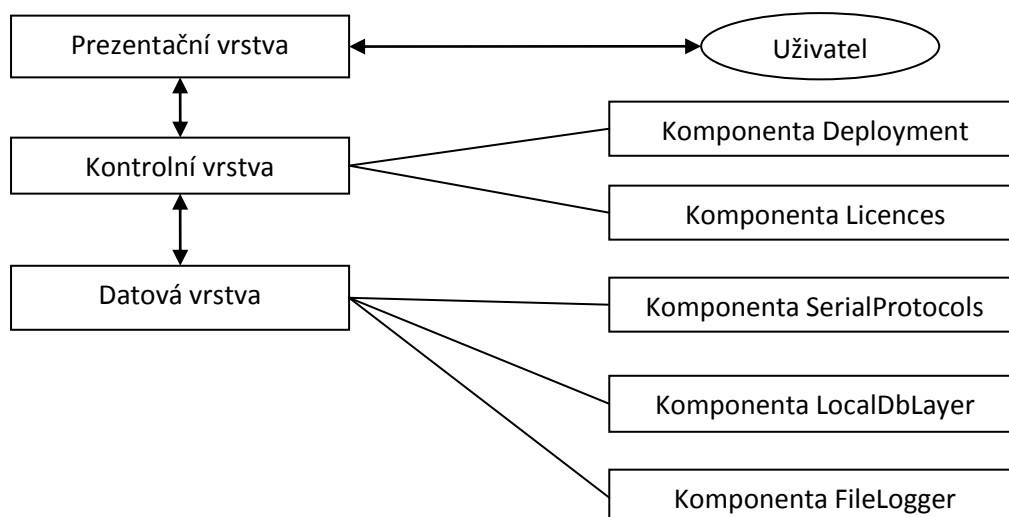
5.2 Komponenty klienta

Klientská část systému se skládá z několika komponent, které byly navrženy tak, aby byly jednoduše znovu použitelné a rozšiřitelné. Klient je celkově rozdělen na několik jmenných prostorů, kde každá komponenta má svůj vlastní. V této kapitole budou popsány nejdůležitější komponenty, ostatní nejsou z pohledu této práce významné, a proto nebudou popisovány.

5.2.1 Hlavní komponenta SolarMonitor

Hlavní nejdůležitější částí klienta je komponenta SolarMonitor. Tato komponenta obsahuje několik tříd, které využívají další komponenty. SolarMonitor je rozdělen do několika vrstev, kde datovou vrstvou zastupují komponenty pracující se sériovým přenosem a s interní databází. Datová vrstva také obsluhuje konfigurační soubory. Nadřazená kontrolní vrstva se stará o transformaci dat před zasláním do centrální databáze na serveru. Poslední prezentační vrstva zobrazuje stavy systému uživateli a přijímá od uživatele údaje, zejména pro nastavení sběru dat a spojení se serverem. Spuštění sběru dat probíhá paralelně na všech zvolených sériových portech. K tomuto účelu jsou využívány vlákna, na kterých se spouští inicializační metody

komponent. Obrázek 12 zobrazuje systémové rozdělení vrstev spolu s příslušností komponent k jednotlivým vrstvám.



Obrázek 12 Diagram komponent klientské části

5.2.2 Komponenta SerialProtocols

Tato komponenta je jádrem systému. Obsahuje implementaci všech sériových protokolů využívaných systémem. Komponenta je určena k řízení provozu na sériové lince, vytváří dotazy a zpracovává odpovědi, drží si informaci o nalezených zařízeních a jejich parametrech. Dále komponenta obsahuje informace o typech zařízení, se kterými dokáže komunikovat a ze kterých je schopna získat data. Důležitou součástí této komponenty jsou časovače, starající se o periodickou žádost o data. Po jejich získání jsou data následně předána vyšší vrstvě k uložení na server.

5.2.3 Komponenta RestClient

Komponenta *RestClient* se připojuje k REST serveru a zasílá mu data pomocí protokolu HTTPS. Komponenta implementuje metody dle navržených webových služeb. Při zasílání dat je nutno nejdříve provést autorizaci a přihlásit se tak k REST serveru. Pokud se spojení nezdaří je následně využita komponenta *LocalDbLayer*, která se uloží data do lokální databáze. Komponenta zpracovává také příchozí data ze serveru, kontroluje hlavičku odpovědi a podle stavových kódů odpovědi vyhazuje případné výjimky.

5.2.4 Komponenta LocalDBLayer

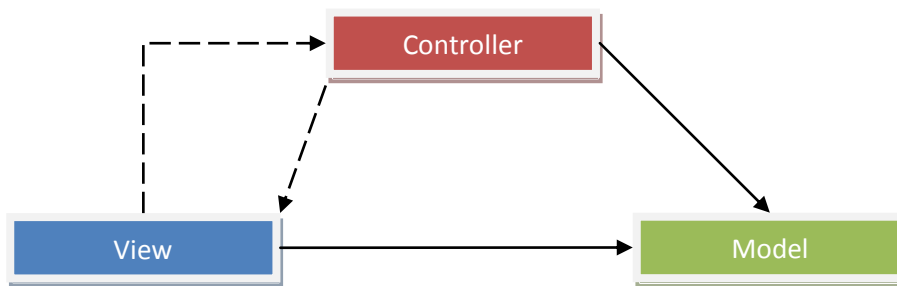
Pokud se nepodaří uložit data na server pomocí komponenty *RestClient*, například z důvodu výpadku internetového spojení, nebo chyby databázového serveru, uloží se data do interní databáze. K uložení dat do interní databáze se využívá služeb komponenty *LocalDBLayer*, který poskytuje přístup k interní databázi. Díky této komponentě je možno data nejen ukládat, ale i mazat nebo aktualizovat. Aktualizace však není využívána. Aktivně se využívá pouze přidání i smazání dat, a to až poté, co jsou úspěšně přesunuta na server.

5.2.5 Komponenta Deployment

Komponenta *Deployment* obsahuje dvě části, *Deployer* a *Updater*. Část *Deployer* se stará o aktualizaci klienta. Kontroluje pravidelně nové aktualizace na serveru a stahuje je. Spouští také druhou část *Updater*, která se nachází vždy v aktualizacím balíku. Po spuštění *Updateru* se celý klient ukončí a zahájí aktualizace. *Updater* překopíruje všechny soubory z aktualizacního balíku do složky s klientem, spustí klientskou aplikaci a sám se smaže spolu s celým aktualizacním balíkem. Aktualizační balíky jsou zabalené a mají příponu *.zip, kromě toho je výsledný archiv ještě zaheslovaný. K rozbalení se využívá externí knihovna *Ionic.Zip.dll*

5.3 Komponenty serveru

Serverová část se skládá z vrstev modelu MVC. Těmito vrstvami je Model (model), View (pohled) a Controller (kontroler). Model reprezentuje informace uložené v databázi, resp. informace se kterými aplikace pracuje. Model také obsahuje business logiku aplikace, někdy také nazývanou jako doménová logika. Pohled je zobrazení dat z modelu uživateli, resp. převádí data z modelu do zobrazitelné podoby. Kontroler reaguje na události od uživatele a přijímá od něho data, transformuje je podle potřeby a předává je k uložení modelu. Kontroler také zajišťuje změny v pohledu nebo v modelu. V následujících podkapitolách budou vysvětleny některé zajímavé komponenty serveru, komponentou je právě myšlen odpovídající pohled s kontrolerem a modelem (využívaných modelů může být více). Blíže MVC návrh zobrazuje následující obrázek 13.



Obrázek 13 Návaznost jednotlivých komponent v modelu MVC

5.3.1 Komponenta Statistics

Komponenta *Statistics* je určena ke zpracování či zobrazení statistických dat. Data z modelu zobrazuje uživateli v přehledných grafech. Na žádost uživatele pak zobrazuje požadovaný časový úsek, typ grafu, apod. Komponenta se skládá z kontroleru *StatisticsController*, který využívá několik modelů k čerpání dat. V této komponentě jsou prováděny navržené analýzy dat, výsledky jsou pak předány pohledu serializované pomocí JSON.

5.3.2 Komponenta Diagnostics

Komponenta *Diagnostics* zpracovává a zobrazuje diagnostická data. Jsou jimi například chybová hlášení, nebo například reporty nastalých událostí. Komponenta obsahuje kontroler *DiagnosticsController*, který využívá několik modelů a předává požadovaná data vrstvě pohledu, která je zobrazí uživateli. V tomto kontroleru jsou prováděny také navržené diagnostické algoritmy. Data jsou mezi vrstvou view a controller přenášena pomocí JSON serializace.

5.3.3 Komponenta Archive

Komponenta je využita tehdy, pokud je potřeba zobrazit podrobná data za určité časové období např. den, měsíc nebo rok. K získání dat komponenta využívá několik modelů, řízených kontrolerem *ArchiveController*. Ten provádí konverzi dat a předává výsledky vrstvě pohledu, přičemž od uživatele zároveň přijímá požadavky na zobrazení dat. Data jsou mezi vrstvami view a controller přenášena pomocí JSON serializace.

5.4 Princip sběru dat ze zařízení

Kapitola popisuje návrh sběru dat ze zařízení. Postup sběru dat se však může lišit podle konkrétního typu zařízení nebo použitého sériového protokolu.

Při sběru dat z jednoho zařízení se postupuje následovně:

1. Detekce zařízení – je vyslán detekční datagram, který zařízení přijme, zpracuje a odešle zpět odpověď obvykle se svým sériovým číslem, síťovou adresou a typem.
2. Registrace zařízení – po přijetí odpovědi na detekční datagram je zařízení zaregistrováno. Jedná se o uložení informace o zařízení do paměti.
3. Nastavení zařízení – pokud je to potřeba, je možné nastavit zařízení síťovou adresu, nebo nějaké další parametry. Nastavení se provádí zasláním datagramu s potřebnými parametry.
4. Žádost o data – zaslání žádosti o data konkrétnímu zařízení pomocí datagramu. Následuje zpracování příchozí odpovědi s daty.
5. Opakování bodu 4 podle potřeby a v potřebných intervalech.

Pokud potřebujeme sbírat najednou data z více zařízení, provádíme předchozí kroky postupně pro každé zařízení. Z principu přenosu po sériové lince nelze tyto procedury provádět paralelně. Výše uvedený nástin je pouze obecným postupem sběru dat. Při implementaci různých protokolů můžeme pozorovat odchylky. Příklad implementace konkrétního protokolu je uveden v následující kapitole.

5.5 Rozbor sériového protokolu SMA

Komunikace se zařízeními na straně elektrárny probíhá převážně pomocí sériového přenosu (sériová link RS485, případně RS232). Zařízení, jakým je například měnič nebo meteorologická stanice, mají implementovaný konkrétní protokol, pomocí kterého mohou zařízení komunikovat jak mezi sebou, tak i se zařízeními třetích stran. Tímto protokolem je například protokol SMA DATA. Ten bylo nutno pro potřeby této práce nastudovat a implementovat. V této části bude rozebrán popis protokolu spolu s návrhem komunikace se zařízeními na sériové lince. Strukturu SMA Net Telegramu zobrazuje tabulka 17.

Rámec			Obsah		Rámec	
Start byte	Adresa	Kontrola	Hlavička protokolu	SMA Data	FCS	Stop byte
<i>1 bajt</i>	<i>1 bajt</i>	<i>1 bajt</i>	<i>2 bajty</i>	<i>až 255 bajtů</i>	<i>2 bajty</i>	<i>1 bajt</i>
0x7E	0xFF	0x03				0x7E

Tabulka 17 Struktura SMA Net Telegramu

Název	Popis	Výchozí hodnota (hexa)
Start byte	Startovní bajt označující začátek nového datagramu.	0x7E
Adresa	Síťová adresa, vždy nastavena jako broadcast.	0xFF
Kontrola	Kontrolní pole, vždy hodnota 0x03, podle HDLC Standard se jedná o „neočíslovaný informační příkaz“.	0x03
Obsah	Obsah datagramu, skládá se z: 1. 2 bajtů hlavičky protokolu, 2. až 255 bajtů dat (SMA Data).	
FCS	16 bitový kontrolní součet.	
Stop byte	Ukončovací bajt označující konec datagramu.	0x7E

Tabulka 18 Popis struktury SMA Net Telegramu

Název	Popis	Výchozí hodnota (hexa)
Adresa odesílatele	Adresa odesílatele datagramu.	
Adresa příjemce	Adresa příjemce datagramu.	
Kontrola	Kontrolní pole. Označuje typ datagramu, zda se jedná o broadcast nebo je datagram určen jednomu příjemci, zda se jedná o dotaz nebo odpověď, atd.	
Počet paketů	Počet paketů, které budou následovat po tomto datagramu.	0x00-0xFF
Číslo příkazu	Číslo příkazu, který je zaslán tímto datagramem.	0x00-0xFF
Data	Přenášená data, délka od 0 až do 255 bajtů, v závislosti na typu příkazu.	

Tabulka 19 Popis struktury SMA Data Telegramu vkládaného do SMA Net Telegramu

5.5.1 Příkaz pro start síťové konfigurace

V originální dokumentaci protokolu SMA označován jako CMD_GET_NET_START. Tento příkaz je zasílán vždy na začátku komunikace se zařízeními. Po přijetí příkazu každé zařízení implementující protokol SMA (měnič, meteorologická stanice apod.) resetuje svůj interní příznak „již dotázán“ a odpoví zpět. Tento příkaz je používán pouze na začátku komunikace.

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	
0x01 0x00	0x00 0x00	0x80	0x00	0x06	

Tabulka 20 Příklad dotazu pro start síťové komunikace

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Sériové číslo, typ zařízení 9380933, WR700-07
0x01 0x00	0x00 0x00	0x40	0x00	0x06	0x45 0x24 0x8F 0x00, 0x57 0x52 0x37 0x30 0x30 0x2D 0x30 0x37

Tabulka 21 Příklad odpovědi na dotaz pro start síťové dokumentace

5.5.2 Příkaz pro síťovou konfiguraci

Tento příkaz je zaslán v rámci detekce ještě nenastavených zařízení v síti. Každý účastník síťové komunikace, který ještě nemá nastavený příznak „již dotázán“ odpoví zasláním své síťové adresy, sériového čísla a typu zařízení. Aby se předešlo kolizi při zasílání odpovědí, je odpověď zaslána náhodně po přenosové pause 85 ms + náhodná hodnota od 0 do 4765 ms.

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	
0x01 0x00	0x00 0x00	0x80	0x00	0x01	

Tabulka 22 Příklad dotazu pro síťovou konfiguraci

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Sériové číslo, typ zařízení 9380933, WR700-07
0x01 0x00	0x00 0x00	0x40	0x00	0x01	0x45 0x24 0x8F 0x00, 0x57 0x52 0x37 0x30 0x30 0x2D 0x30 0x37

Tabulka 23 Příklad odpovědi na dotaz pro síťovou konfiguraci

5.5.3 Příkaz pro nastavení síťové adresy

Každé zařízení v síti musí mít přiřazenou unikátní adresu. Přiřazení se provádí následujícím příkazem. Poté co zařízení obdrží dotaz s nastavením síťové adresy, nastaví si tuto adresu a zároveň si nastaví příznak „již dotázán“.

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Sériové číslo, síťová adresa 9380933, 3
0x01 0x00	0x00 0x00	0x80	0x00	0x03	0x45 0x24 0x8F 0x00, 0x03 0x00

Tabulka 24 Příklad dotazu pro nastavení síťové adresy

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Sériové číslo
0x03 0x00	0x01 0x00	0x40	0x00	0x03	0x45 0x24 0x8F 0x00

Tabulka 25 Příklad odpovědi na dotaz pro nastavení síťové adresy

5.5.4 Příkaz pro synchronizaci dat

Před každou žádostí o data ze zařízení je nutno zaslat synchronizační datagram. Po jeho přijetí zařízení zkopíruje stav své paměti v daném čase a připraví tak data na odeslání. Na tento dotaz není zaslána odpověď.

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Unixový formát času 843504044
0x01 0x00	0x00 0x00	0x80	0x00	0x0A	0x32 0x46 0x09 0xAC

Tabulka 26 Příklad dotazu pro synchronizaci dat

5.5.5 Příkaz pro získání dat

Data jsou ze zařízení získána následujícím dotazem, který obsahuje masku typu kanálu a číslo kanálu. Pokud je číslo kanálu rovno nule, je zaslán obsah všech kanálů dané masky. Pokud je číslo kanálu větší než nula, je zaslán obsah konkrétního kanálu. Odpověď na tento dotaz může přijít ve více datagramech.

Hlavička protokolu					Data
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Typ kanálu, Kanál
0x01 0x00	0x02 0x00	0x00	0x00	0x0B	0x0F 0x09, 0x00

Tabulka 27 Příklad dotazu pro získání dat

Hlavička protokolu					Data					
Adresa odesílatele	Adresa příjemce	Kontrola	Počet paketů	Číslo příkazu	Typ kanálu, Kanál	Čas	Jednotka času	Kanál 1	...	Kanál N
0x02 0x00	0x01 0x00	0x40	0xNN	0x0B	0x0F 0x09, 0x00	32 bit	32 bit	N byte	...	N byte

Tabulka 28 Příklad odpovědi na dotaz pro získání dat

5.5.6 Návrh principu sběru dat – protokol SMA

V před samotnou implementací sběru dat ze zařízení bylo nutno navrhnout, jakým způsobem bude sběr dat probíhat. Algoritmus jsem navrhl tak, aby sběr dat ze zařízení byl rychlý a efektivní. Zároveň však bylo nutné dodržet pravidla pro registraci zařízení dle protokolu SMA a nastavit dobu čekání mezi jednotlivými dotazy tak, aby nedocházelo ke kolizím. Při určování časových rozestupů mezi zasíláním dotazů jsem vycházel z vlastní zkušenosti s komunikací po sériové lince.

Při sběru dat ze zařízení se v rámci protokolu SMA postupuje následujícím způsobem:

1. Je zaslán a zpracován Příkaz pro start síťové konfigurace.
2. Čekání po obdržení poslední odpovědi – 10 sekund.
3. Je zaslán a zpracován Příkaz pro nastavení síťové adresy.
4. Čekání mezi jednotlivými dotazy – 1200 ms.

5. Je zaslán a zpracován Příkaz pro síťovou konfiguraci.
6. Čekání po obdržení poslední odpovědi – 10 sekund.
7. Opakování bodu 3 až 5 dokud nebudou všechna zařízení registrována.
8. Zahájení sběru dat.
9. Je zaslán a zpracován Příkaz pro synchronizaci dat.
10. Je zaslán a zpracován Příkaz pro získání dat.
11. Čekání mezi jednotlivými dotazy – 1200 ms.
12. Opakování bodu 9 až 11, dokud je potřeba získávat data.

Výše uvedený postup sběru jsem navrhl jako základní, následně jsem jej rozšířil ještě o následující pravidla.

1. Pokud není na Příkaz pro start síťové konfigurace obdržena odpověď, je tento příkaz zaslán po 10 sekundách znovu.
2. Přesně ve 12 hodin je přerušen sběr dat a je započata znovu rutina pro detekci a registraci zařízení, následně je opět zahájen sběr dat. Toto pravidlo je aplikováno z důvodu detekce nově připojených nebo odpojených zařízení.
3. Pokud zařízení odpoví na dotaz patřící do fáze, která již neprobíhá, je toto zařízení oznámeno jako potencionálně špatně zapojené a sběr dat probíhá bez jeho přítomnosti. Toto pravidlo je aplikováno k zabránění možných kolizí síťových adres z důvodu špatného zapojení.
4. Pokud bylo zjištěno spuštění klienta v noci, je po prvním validním přijatém datagramu sběr dat zastaven a započat až za 30 minut. Tímto je detekováno svítání, v tuto dobu nejsou všechny měniče plně provozuschopné, je zde tedy riziko registrace jen některých z nich.

5.6 Rozbor sériového protokolu SPINEL97

Dalším protokolem, který byl využit při realizaci monitorování solární elektrárny, je protokol hardwarového modulu SPINEL. Protokol SPINEL existuje ve verzi 66 (ASCII) a 97 (binární). Ve své práci využívám protokol SPINEL97. Hardwarový modul Quido obsahuje deset vstupů, na každém je čítač. Klient se dotazuje modulu na stav čítačů, jež může také resetovat. Hodnota čítačů na vstupu se zvyšuje vždy o jeden impuls. Při monitorování impulsů přicházejících z trafostanice je využíváno právě těchto hardwarových čítačů. Jeden impuls se rovná 100 W.

Název	Popis
PRE	Prefix, 0x2A (znak “*“).
FRM	Číslo formátu 97 (0x61).
NUM	Počet bytů instrukce od následujícího bajtu do konce rámce.
ADR	Adresa modulu, kterému je posílán dotaz nebo který posílá odpověď.
SIG	Podpis zprávy - libovolné číslo od 0x00 do 0xFF. Stejně číslo, které bylo posláno v dotazu, se vrátí v odpovědi, čímž lze snadno rozpoznat, na který dotaz odpověď přišla.
ACK	Potvrzení dotazu (Acknowledge), zda a jak byl proveden. ACK jsou z intervalu 0x00 až 0x0F.
INST	Kód instrukce.
DATA	Data.
SUMA	Kontrolní součet. Součet všech bytů instrukce (sčítají se úplně všechna odesílaná data kromě CR) odečtený od 255.
CR	Zakončovací znak (0x0D).

Tabulka 29 Popis formátu protokolu SPINEL97

V následujících podkapitolách budou popsány instrukce protokolu potřebné k získání dat z modulu Quido.

5.6.1 Instrukce pro čtení parametrů

Použití této instrukce je určeno pro zjištění nastavené adresy v případě, kdy není známa. Dotaz se přitom posílá na univerzální adresu 0xFE. Pokud není známa ani komunikační rychlost, je třeba vyzkoušet všechny komunikační rychlosti zařízení. Na lince ale nesmí být připojeno žádné další zařízení. Instrukce vrací adresu a komunikační rychlost.

PRE	FRM	NUM	NUM	ADR	SIG	INST	SUMA	CR
0x2A	0x61	0x00	0x05	0xFE	0x02	0xF0	0x7F	0x0D

Tabulka 30 Příklad instrukce Čtení komunikačních parametrů

PRE	FRM	NUM	NUM	ADR	SIG	ACK	DATA	SUMA	CR
0x2A	0x61	0x00	0x07	0x04	0x02	0x00	0x04, 0x06	0x5D	0x0D

Tabulka 31 Příklad odpovědi - adresa 04H, komunikační rychlost 9600Bd

5.6.2 Instrukce pro čtení výrobních údajů

Po provedení instrukce vrátí zařízení výrobní údaje. Výrobními údaji se rozumí číslo výrobku a sériové číslo.

PRE	FRM	NUM	NUM	ADR	SIG	INST	SUMA	CR
0x2A	0x61	0x00	0x05	0xFE	0x02	0xFA	0x75	0x0D

Tabulka 32 Příklad instrukce Čtení výrobních údajů

PRE	FRM	NUM	NUM	ADR	SIG	ACK	DATA	SUMA	CR
0x2A	0x61	0x00	0x0D	0x35	0x02	0x00	0x00, 0xC7, 0x00, 0x65, 0x20, 0x05, 0x09, 0x23	0xB3	0x0D

Tabulka 33 Příklad odpovědi – číslo-výrobku 199 (=00C7H), sériové číslo 101 (=0065H)

5.6.3 Instrukce pro nastavení vysílání

Povoluje nebo zakazuje automatické vysílání zprávy při změně logické úrovně na vstupech. Tato instrukce umožňuje automaticky informovat nadřazený systém o změně stavu některého ze vstupů. Není pak nutné se například opakovaně dotazovat na stav vstupů instrukcí. Z výroby je automatické vysílání vypnuto.

PRE	FRM	NUM	NUM	ADR	SIG	INST	DATA	SUMA	CR
0x2A	0x61	0x00	0x06	0x01	0x02	0x10	0x01	0x5A	0x0D

Tabulka 34 Příklad aktivování samovolného vysílání zprávy; adresa 01H, podpis 02H:

PRE	FRM	NUM	NUM	ADR	SIG	ACK	SUMA	CR
0x2A	0x61	0x00	0x05	0x01	0x02	0x00	0x6C	0x0D

Tabulka 35 Příklad odpovědi na aktivování samovolného vysílání

PRE	FRM	NUM	NUM	ADR	ACK	DATA	SUMA	CR
0x2A	0x61	0x00	0x07	0x01	0x00	0x0D, 0x00, 0x00	0x5E	0x0D

Tabulka 36 Automaticky odeslaná odpověď s daty

5.6.4 Instrukce pro čtení čítačů

Instrukce čte stav počítadel změn na vstupech. Čítač umožňuje počítat jednotlivé změny stavu vstupu. Za změnu je považována změna logického stavu (nebo stavu připojeného kontaktu). Každý vstup má vlastní čítač. K hodnotě čítače je přičtena jednička při vybraných změnách na příslušném vstupu (změna z 1 do 0; změna z 0 do 1; případně obě změny).

PRE	FRM	NUM	NUM	ADR	SIG	INST	DATA	SUMA	CR
0x2A	0x61	0x00	0x06	0x31	0x02	0x60	0x00	0xDB	0x0D

Tabulka 37 Příklad pro čtení čítačů

PRE	FRM	NUM	NUM	ADR	SIG	ACK	DATA	SUMA	CR
0x2A	0x61	0x00	0x1A	0x31	0x02	0x00	0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, , 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00	0x17	0x0D

Tabulka 38 Odpověď na čtení čítačů (příklad z modulu Quido ETH 10/1)

5.6.5 Instrukce pro nastavení čítačů

Instrukce nastavuje parametry počítadel změn na vstupech. Čítač umožňuje počítat jednotlivé změny stavu vstupu. Za změnu je považována změna logického stavu (nebo stavu připojeného kontaktu). Každý vstup má vlastní čítač. K hodnotě čítače je přičtena jednička při vybraných změnách na příslušném vstupu (změna z 1 do 0; změna z 0 do 1; případně obě změny).

PRE	FRM	NUM	NUM	ADR	SIG	INST	DATA	SUMA	CR
0x2A	0x61	0x00	0x06	0x31	0x02	0x6A	0x80	0x51	0x0D

Tabulka 39 Příklad nastavení čítačů

PRE	FRM	NUM	NUM	ADR	SIG	ACK	SUMA	CR
0x2A	0x61	0x00	0x05	0x31	0x02	0x00	0x3C	0x0D

Tabulka 40 Odpověď na nastavení čítačů – ok

5.6.6 Princip sběru dat – protokol SPINEL97

Pro získání dat z čítačů modulu Quido bylo nutno navrhnout odlišný postup, než byl použit například u sběru dat z měničů. Při návrhu komunikace se zařízením Quido, jsem opět postupoval tak, aby výsledný algoritmus byl efektivní a sběr dat byl rychlý.

Vzhledem k tomu, že není možno čítačům modulu Quido nastavovat počáteční hodnoty ale pouze je resetovat, je nutné tyto hodnoty získané od uživatele uložit do konfiguračního souboru. Hodnoty, které byly přečtené z čítačů, budou následně k těmto známým hodnotám před odesláním do centrální databáze přičítány. Pokud by došlo k přerušení sběru dat, například z důvodu výpadku monitorovací stanice, budou počty impulzů přicházejících z trafostanice stále uloženy v hardwarovém modulu Quido a je velice snadné je opětovně získat a zaslat do centrální databáze k dalšímu zpracování. Postup sběru dat z hardwarových čítačů je nastíněn v následujícím přehledu.

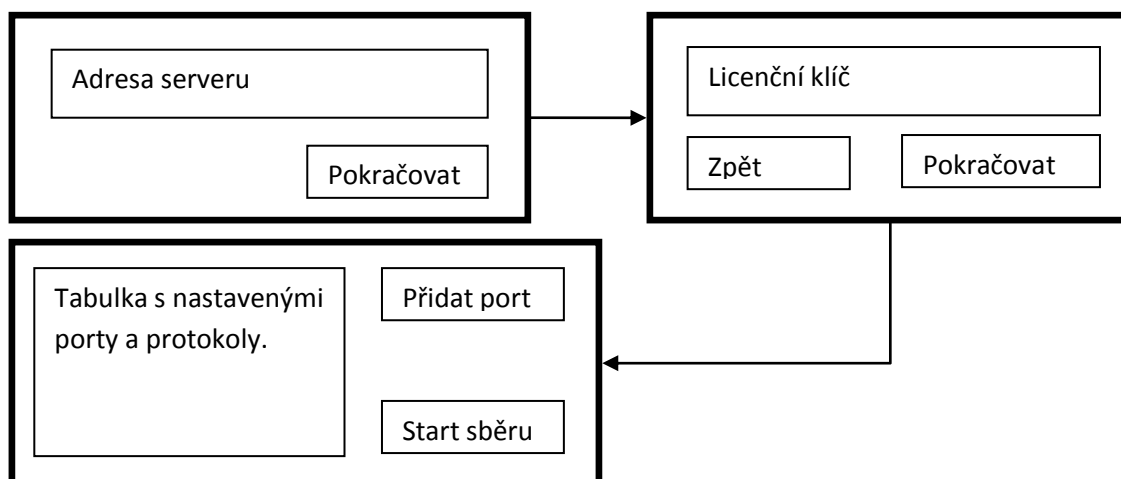
1. Zaslání a zpracování
2. Instrukce pro čtení parametrů.
3. Zaslání a zpracování Instrukce pro čtení výrobních údajů.
4. Zaslání a zpracování
5. Instrukce pro nastavení čítačů.
6. Zaslání a zpracování
7. Instrukce pro nastavení vysílání.
8. Zaslání a zpracování Instrukce pro čtení čítačů.
9. Čekání mezi dotazy – 3 sekundy.
10. Opakování bodu 5, dokud je potřeba získávat data.

5.7 Návrh uživatelského rozhraní

Uživatelské rozhraní je nedílnou součástí každé moderní aplikace. Pro potřeby výsledného systému jsem musel navrhnout dvě uživatelská rozhraní. Jedno je určeno pro klientskou část a druhé pro část serverovou. Následuje popis navržených uživatelských rozhraní.

5.7.1 Rozhraní klientské části

Rozhraní klientské části je navrženo s ohledem na efektivnost, hlavním záměrem je intuitivní rozhraní, které bude schopen ovládat každý běžný uživatel. Základní zaškolení však bude nutné vždy. GUI klientské části se skládá ze tří oken, které na sebe postupně navazují, tak jak uživatel prochází jednotlivé kroky nastavení. Po úspěšném nastavení aplikace je uživateli zobrazena potvrzující hláška o zahájení sběru dat. Nejdříve je uživateli zobrazena žádost o zadání adresy serveru. Pokud je tato adresa validní, je zobrazena žádost o licenční klíč (pokud již nebyl zadán). Pokud nebyl licenční klíč již použit je zobrazeno nastavení jednotlivých sériových portů, poté je možno zahájit sběr dat. Na následujícím obrázku 14 je načrtnut návrh uživatelského rozhraní klientské části.



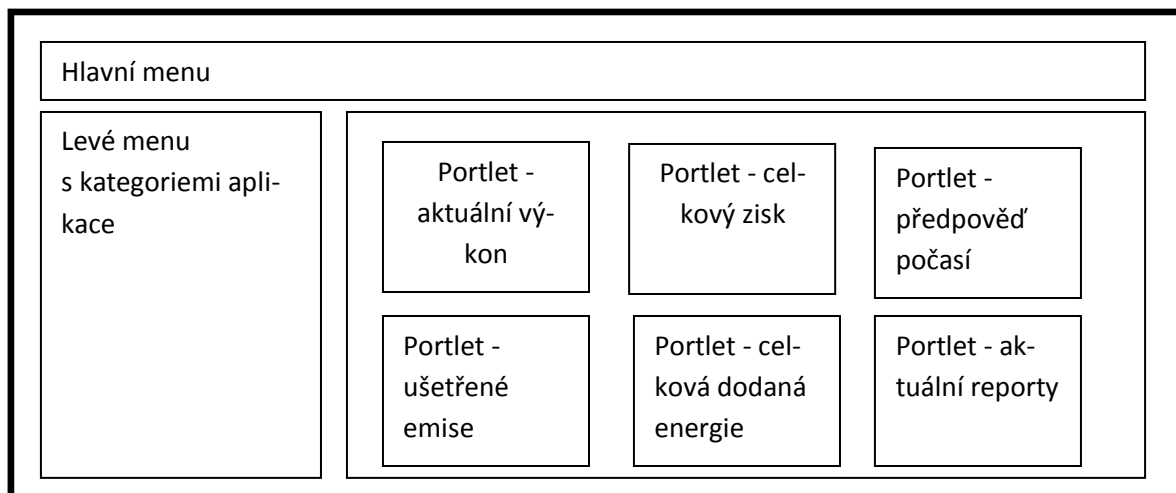
Obrázek 14 Návrh uživatelského rozhraní klientské aplikace

5.7.2 Rozhraní serverové části

Design administračního rozhraní je navržen tak, aby bylo ovládání co nejvíce přirozené a nároky na znalosti uživatele byly co nejmenší. Návrh rozhraní má připomínat klasickou desktopovou aplikaci, kdy je pracovní plocha rozdělena na tři části. Horní menu obsahuje možnost odhlášení. V levé části je menu tvořené stromovou strukturou složek s kategoriemi administrace a v pravé centrální části se zobrazuje jednotlivý obsah zvolených sekcí.

Úvodní stránka administrace obsahuje jednoduché portlety, určené k přehlednému zobrazení nejdůležitějších informací, jako je třeba aktuální výkon, zisk a předpověď počasí. Na obrázku 15 je zobrazen návrh vybraných částí administračního rozhraní.

Hlavní okno aplikace



Obrázek 15 Návrh GUI administračního rozhraní

5.8 Rozbor vybraných algoritmů a výpočtů

Pokud jsou v databázi uložena „surová“ data, je potřeba je nějakým způsobem analyzovat. Uživateli je možno zobrazovat aktuální informace, nejlépe i informace za nějaké období. Kapitola obsahuje návrh a popis některých algoritmů, které jsou použity k analýze dat a vytváření statistických přehledů.

5.8.1 Výpočet aktuálního výkonu

Aktuální výkon P je pro investora solární elektrárny jedna ze stěžejních informací. Důležitou veličinou pro výpočet aktuálního výkonu je tzv. PAC, tedy výstupní výkon [W] na AC straně měniče. Jelikož se jedná o aktuální výkon, zajímá nás PAC z posledního sběru dat, resp. poslední naměřená hodnota PAC daného měniče. Hodnota aktuálního výkonu je pak součet všech PAC měničů, viz následující vzorec:

$$P = \sum_{n=1}^n PAC_n^t$$

Vzorec 1 Výpočet aktuálního výkonu

kde n je počet měničů, t je čas posledního sběru.

5.8.2 Výpočet aktuálního zisku

Další zajímavou informací pro investora je aktuální zisk M , tedy jaký je zisk z prodané elektrické energie. Informaci můžeme získat pomocí součtu celkové dodané energie ETOTAL [kWh] v daném čase, vynásobené výkupní cenou za 1 kWh. Výpočet se provádí podle následujícího vzorce:

$$M = \sum_{n=1}^n \text{ETOTAL}_n^t \cdot VC$$

Vzorec 2 Výpočet aktuálního výkonu

kde n je počet měničů, t je čas posledního sběru, VC je aktuální výkupní cena.

5.8.3 Výpočet ztrát

Obvykle investor potřebuje znát ztráty své elektrárny při výrobě energie. Pro zjištění ztrát postupujeme následovně. Zjistíme jednotlivé ztráty na vedení, měničích, trafostanici, bleskojistkách, proudových chráničích, apod. Takto vypočtené ztráty můžeme porovnat s teoretickou hodnotou vyrobené energie, kterou zjistíme z dat meteorologické stanice. Při určování dílčích ztrát využíváme například následujících vzorců:

$$R = R_0 \cdot \left(\frac{1}{s}\right)$$

kde R_0 je měrný elektrický odpor, s je poloměr vodiče, délka vodiče je 1 m.

Vzorec 3 Výpočet odporu kabelů z panelu k měniči - ztráty DC (stejnoseměrném) vedení

$$\eta = 1 - \left(\frac{\text{PAC}}{\text{PDC}}\right) \cdot 100$$

Vzorec 4 Výpočet ztrát na měniči (účinnost měniče) – ztráty přeměnou DC na AC

kde PAC je výstupní výkon na AC straně a PDC je výstupní výkon na DC straně.

5.8.4 Výpočet emisních povolenek

Emisní povolenky jsou prostředkem, jak zamezit globálním změnám klimatu. Jejich účelem je motivovat společnosti k přechodu na tzv. nízkouhlíkovou ekonomiku. S emisními povolenkami je možno obchodovat, proto každého investora zajímá, kolik emisních povolenek má díky své elektrárně v držení. Při výpočtu zisku z emisních povolenek je nutno nejdříve vypo-

čítat kolik kg oxidu uhličitého elektrárna ušetřila životnímu prostředí. Výpočet se provede následujícím vzorcem:

$$EP = \sum_{n=1}^n ETOTAL_n^t \cdot 0,7$$

Vzorec 5 Výpočet množství ušetřeného CO2

kde n je počet měničů, t je čas posledního sběru, ETOTAL je celkový dodaný výkon.

5.8.5 Algoritmus pro zjištění poruch

Pro investory je velice důležité mít solární elektrárnu v provozu vždy, když je intenzita slunečního záření dostatečná. Zařízení solární elektrárny však mohou vypovědět službu a tím se sníží výrobní kapacita a zisk z prodeje elektrické energie. Aby bylo možno redukovat dobu odstávky zařízení na minimum, je nutno detekovat poruchy těchto zařízení a rychle tyto poruchy odstranit. K detekci poruch v mnou navrhovaném systému využívám algoritmus navržený a popsáný v této kapitole.

Při návrhu algoritmu pro zjištění poruch na zařízeních jsem vycházel ze dvou předpokladů. Zařízení je v poruchovém stavu, zasílá tuto informaci do centrály, tedy komunikuje, ale nemusí vyrábět elektrickou energii. Zachycení této události je vcelku jednoduché. Stačí si kontrolovat odpovídající příchozí data a chybové stavy reportovat administrátorovi.

Další možností, na kterou jsem se zaměřil, je, že zařízení díky poruše není schopno vyrábět energii ani komunikovat s centrálou. Zařízení však nemusí být vyřazeno z provozu kvůli poruše, je možné, že bylo nelegálně odpojeno. Z globálního pohledu se však jeví jako vyřazené z provozu, protože nezasílá data. Při detekci této události jsem postupoval následovně:

1. Zjištění zda je intenzita dopadajícího světla větší než 10 W.m^{-2} , protože je nutno vyloučit pravidelné vypnutí měničů z důvodu nedostatku slunečního záření.
2. Pokud je sluneční záření dostatečné, je nutno ke každému zařízení (trafostanice, měnič) zjistit dobu posledního zaslání dat.
3. Pokud doba uplynulá od posledního zaslání dat je větší než 1 hodina, je zařízení prohlášeno za poruchové a nahlášeno administrátorovi.

Výše uvedený algoritmus dobře detekuje měniče neschopné provozu nebo odpojené. Je však na administrátorovi, zda ještě před vysláním technického pracovníka k opravě poruchy, vyhodnotí situaci jinak, zejména na základě dalších reportů přicházejících z monitorovacího zařízení. Je také možné, že výpadek je plánovaný a zařízení byla odpojena zcela legálně.

6 Implementace

Po úvodní studii a nastínění požadavků na systém bylo přistoupeno k analýze a návrhu vlastního řešení. Implementace je dalším logickým krokem před samotným otestováním naprogramovaného systému a následným nasazením. V této kapitole bude uveden průběh implementace, spolu s popisem použitých technologií, včetně jejich zhodnocení.

6.1 Použité technologie

Při vývoji navrženého systému jsem se snažil preferovat technologie firmy Microsoft. Pro implementaci klientské části systému byl zvolen jazyk C#, který je čistě objektový a firma Microsoft ho aktivně vyvíjí již od roku 2000. Jako programovou platformu jsem zvolil .NET Framework ve verzi 3.5 SP 2. Framework .NET je nadstavbou operačního systému, zkvalitňuje tak služby OS a zvyšuje bezpečnost. Výhodou je, že obsahuje vlastní Garbage Collector, který programátorům velice usnadňuje práci.

Technologie společnosti Microsoft jsem se rozhodl využívat proto, že je dobře znám a konkrétně v prostředí frameworku .NET jsem vytvořil několik aplikací, nejen v rámci studia na vysoké škole.

Serverová část je napsána také v jazyce C# s použitím ASP.NET MVC 2 Frameworku. Předností tohoto Frameworku je implementace návrhového vzoru MVC, který zpřehledňuje a usnadňuje programování webových aplikací. Aplikace založené na ASP.NET jsou rychlejší, neboť jsou předkompilovány do DLL souborů. Tím se liší od skriptovacích jazyků, kde jsou stránky při každém přístupu opětovně parsovány.

Jako datové úložiště byl využit Microsoft SQL Server Compact 3.5 a Microsoft SQL Server 2008. První jmenovaný server je určen pro uložení interní databáze dat v klientské aplikaci. Tato databáze je výhodná v tom, že je kompaktní a není potřeba při nasazení instalovat celý server, ale pouze jen knihovny pro práci s touto databází. Databáze Microsoft SQL Server 2008 je využit v serverové části aplikace. Zde se ukládají nasbíraná data a z této databáze se vytvářejí analýzy a statistické přehledy.

Pro tvorbu webového rozhraní byl využit skvělý Framework Ext JS 3.2.1 vytvořený v jazyce JavaScript. Tento framework obsahuje velké množství předpřipravených komponent, což velice usnadňuje práci vývojářů. Rozhraní vytvořená pomocí tohoto frameworku se nejčastěji hodí na administraci nějakého serveru, protože komponenty mají jednotný vzhled, když se dají stylovat podle potřeby.

6.2 Použité nástroje

Nejvíce využívaným nástrojem při implementaci bylo vývojové prostředí Microsoft Visual Studio 2008. Toto IDE se velice osvědčilo a pro implementaci programů založených na platformě .NET je bezkonkurenční. Ke správě databází byl využíván nástroj SQL Server Management Studio Express. Rovněž vynikající nástroj pro správu Microsoft SQL Serveru.

Dalším využitým nástrojem byl program Skater .NET Obfuscator. Tento program je určen pro tzv. zmatení nebo zatemnění kódu. Aplikace postavené na platformě .NET, resp. jejich DLL knihovny a EXE soubory, jsou jednoduše dekompilovatelné. Kdokoliv si tak může dekompilovat knihovnu a získat tak původní zdrojový kód. Je tedy reálné riziko, zejména ze strany konkurence, že by mohla získat přístup k mému programu a získat tak duševní vlastnictví ve formě zdrojových kódů. Proto jsem využil program od firmy Skater a zamezil jsem dekompilaci a získání zdrojových kódů v případě nasazení mého systému do produkčního prostředí.

Jako nástroj pro vytváření instalátoru byl využit program Inno Setup, který dokáže při instalaci nastavovat práva složek, vytvářet ikony na ploše a podobně. Tento program je zdarma a má velice obsáhlou a podrobnou dokumentaci.

6.3 Popis implementace

V této kapitole bych rád zmínil části programu, kterými mohou být zdrojové kódy, datové struktury, apod. V neposlední řadě také použité SQL dotazy. Bohužel popis všech částí programu by silně přesahoval rozsah této práce, proto se zaměřím pouze na zajímavé části aplikace.

6.3.1 Použité datové struktury

Při vývoji aplikace bylo použito několik datových struktur. Drtivá většina pochází přímo z .NET Frameworku, vlastní struktury jsem nenavrhol. Kolekce .NET Frameworku byly plně dostačující. V následujícím přehledu uvedu popis využitých zajímavých struktur, jejich implementaci v .NET Frameworku a příklad využití v mém systému.

6.3.1.1 Kolekce Dictionary<T,T>

Prvky jsou v kolekci Dictionary<T,T> uloženy v páru, tedy klíč a hodnota. Třída Dictionary<T,T> je generická a jedná se implementaci třídy HashTable, která je realizována pomocí pole s hashovanými indexy. Tuto strukturu jsem využil například pro ukládání informací o

zařizování, kdy klíčem je unikátní sériové číslo a hodnotou je třída typu zařízení. Složitosti operací:

- Vlož – pokud je počet prvků ve struktuře menší než kapacita pak $O(1)$, pokud musí být při vkládání kapacita navýšena pak $O(n)$.
- Odeber – přibližně $O(1)$.
- Najdi – přibližně $O(1)$.

6.3.1.2 Kolekce Queue<T>

Fronta, kterou jsem použil ve svém programu je implementována pomocí generické kolekce Queue<T> z .NET Frameworku. Ta je interně representována kruhovým polem, díky tomu není nutné při každém výběru přerovnávat pole. Kolekci Queue<T> jsem využil například pro příchozí a odchozí datagramy. Tato struktura se ukázala jako velice efektivní a pro zvolený případ užití naprosto vyhovující. Složitosti operací:

- Vlož – pokud je počet prvků ve struktuře menší než kapacita pak $O(1)$, pokud musí být při vkládání kapacita navýšena pak $O(n)$.
- Odeber – $O(1)$.

6.3.2 Komunikace pomocí sériové linky

Pro lepší pochopení komunikace pomocí sériové linky uvedu příklad odeslání datagramu a přijetí odpovědi. Následující kód je převzat z implementace komunikace se zařízeními pomocí protokolu SMA.

```
public class SmaMonitor
{
    private SerialPortManager serialPortManager = new SerialPortManager();
    private Queue<byte> recievedBytesQueue = new Queue<byte>();

    public SmaMonitor()
    {
        serialPortManager.DataRecieved += serialPortDataReceived;

        if (!serialPortManager.OpenPort(portName, boudRate, true,
true))
        {
            throw new Exception("Nemohu otevřít port " + this.portName
+ ". Zkontrolujte zda není blokován jinou aplikací.");
        }
    }
}
```

```

    }
}
Public SendRequest ()
{
SmaDatagram datagram = new SmaDatagram ();
    datagram.Source = 0;
    datagram.Destination = 0;
    datagram.Ctrl = 0x80;
    datagram.PktCnt = 0x00;
    datagram.Cmd = 0x02;
    datagram.Type = SmaDatagrams.Request;
    datagram.Command = SmaCommands.CmdSearchDev;
    datagram.Data = BitConverter.GetBytes (serialNumber);

    serialPortManager.WriteData (datagram);
}

private void serialPortDataReceived (object sender, EventArgs e)
{
    byte[] data = serialPortManager.ReadData ();

    if (data != null && data.Length > 0)
    {
        data.ToList ().ForEach (b => recievedBytesQueue.Enqueue (b));
    }
}
}

```

Část kódu 1 Ukázka odeslání a přijetí odpovědi v protokolu SMA

Výše uvedený segment zdrojového kódu je příkladem vytvoření dotazu pro nalezení zařízení na sériové lince a přijetí odpovědi od tohoto zařízení. Třída obsahuje dva privátní členy, objekt *SerialPortManager* a seznam přijatých bajtů. V konstruktoru se zaregistruje událost *serialPortDataReceived*, přičemž se otevře seriový port. Zavoláním veřejné metody *SendRequest()* je odeslán vyhledávací datagram. Odpověď dorazí obvykle s menším zpožděním, proto jsou jednotlivé dorazivší bajty ukládány do bufferu, který je tvořen kolekcí *Queue<T>*. Z tohoto bufferu je pak možno jednotlivé přijaté bajty vybírat a sestavovat z nich příchozí datagram, který je následovně vložen do fronty pro příchozí datagramy, kde čeká na zpracování. Při asynchronní komunikaci často vzniká problém jak detekovat konec přijímaných dat. K této činnosti se hodí timer, kdy je nastaven interval a timer resetován vždy při obdržení dat. Pokud dokončí timer svůj nastavený interval, byla přijata všechna data a buffer je zpracován.

6.3.3 Použité SQL dotazy

Při zpracování dat na straně serveru byly využity procedury a SQL dotazy, které bych rád v této části vysvětlil. Pro zajímavost uvedu jen některé vybrané dotazy.

6.3.3.1 Trigger pro výpočet výkonu

Následující trigger doplňuje přírůstek výkonu pro každý záznam z měniče, který byl přidán do tabulky *InventorsDataRecords*. Samozřejmě je možné výkon dopočítávat zpětně, ale vzhledem k tomu, že v databázi jsou statisíce řádků, bylo by to velice časově náročné. Proto jsem zvolil tuto cestu a vypočítávám přírůstky dopředu. Trigger také zjišťuje nové, ještě neschválené měniče a označuje je jako ilegální. Trigger je spouštěn vždy po vložení do tabulky *InventorsDataRecords*.

```
CREATE TRIGGER after_insert_inventor_record ON mon_monitoring
AFTER INSERT AS BEGIN
DECLARE @record_1 INT;
DECLARE @record_2 INT;
DECLARE @last_timestamp DATETIME;
DECLARE @last_e_total1 FLOAT;
DECLARE @last_e_total2 FLOAT;
DECLARE @last_inventor_id INT;
DECLARE @last_inventor_serialnumber INT;
SELECT @last_e_total1 = i_e_total,@record_1=idr_id,@last_inventor_id =
idr_i_id FROM InventorDataRecords WHERE
idr_id=IDENT_CURRENT('InventorDataRecords');
SELECT TOP 1 @last_e_total2 = idr_e_total,@record_2=idr_id FROM InventorDa-
taRecords WHERE idr_timestamp < @last_timestamp ORDER BY idr_timestamp
DESC;
IF (@last_e_total2 > 0 AND (@last_e_total1-@last_e_total2 ) < 2 AND
(@last_e_total1-@last_e_total2 ) > 0)
UPDATE InventorDataRecords SET i_performance = (@last_e_total1-
@last_e_total2 )WHERE idr_id = @record_1
ELSE
UPDATE InventorDataRecords SET i_performance = 0 WHERE idr_id = @record_1

IF (@last_inventor_id= 0)
BEGIN
INSERT INTO Inventors VALUES(NULL, "illegal", 0, 1,1,4,0,0,"null",
NOW(),NOW(),0);
END
END
```

Část kódu 2 Trigger pro výpočet přírůstků výkonu a zjištění nelegálních měničů

6.3.3.2 Agregace naměřených hodnot

Některé naměřené hodnoty je nutno pro potřeby statistických analýz agregovat. Například v grafu průměrných naměřených hodnot rychlosti větru je využita agregační funkce `AVG()`. Hodnoty jsou agregovány při zpracování SQL dotazu, není proto potřeba je následně upravovat. V následujícím příkladu je uveden dotaz pro získání průměrných hodnot všech naměřených veličin z meteorologické stanice. Dotaz je napsán pomocí LINQ to SQL, což je objekto-vě-relační mapper. S daty lze jednoduše manipulovat a vytvářet složité dotazy, které se vykonávají, až když je to opravdu potřeba. Do té doby se pracuje s kolekcí `IQueryable<T>`. Díky tomu jsou dotazy na databázi efektivnější a není nutno přenášet velké množství dat. Obvykle se dotaz provede až při převodu výše zmíněné kolekce na jinou, například `IList<T>` nebo `IDictionary<T,T>` případně při procházení kolekce pomocí konstrukce `foreach`. Výsledný dotaz, kterým jsou získána data, je uveden pod odpovídajícím LINQ to SQL zápisem.

```
var q = (from i in db.SensorboxDataRecords
        where i.sdr_timestamp.Date == date.Date
          && i.Sensorbox.s_id == meteoId
        group i by i.sdr_timestamp.Hour into t

        select new
        {
            time = t.Key,
            values = new double?[]{
                t.Avg(x => x.sdr_int_sol_irr_w_m2),
                t.Avg(x => x.sdr_tmp_amb_c),
                t.Avg(x => x.sdr_tmp_mdul_c),
                t.Avg(x => x.sdr_wind_vel_ms)
            }
        }

        );

//--- výsledný dotaz

{SELECT AVG([t2].[sdr_int_sol_irr_w_m2]) AS [value],
AVG([t2].[sdr_tmp_amb_c]) AS [value2], AVG([t2].[sdr_tmp_mdul_c]) AS
[value3], AVG([t2].[sdr_wind_vel_ms]) AS [value4], [t2].[value] AS [time]
FROM (
    SELECT DATEPART(Hour, [t0].[sdr_timestamp]) AS [value],
[t0].[sdr_timestamp], [t1].[s_id], [t0].[sdr_int_sol_irr_w_m2],
[t0].[sdr_tmp_amb_c], [t0].[sdr_tmp_mdul_c], [t0].[sdr_wind_vel_ms]
FROM [dbo].[SensorboxDataRecords] AS [t0]
```

```

INNER JOIN [dbo].[Sensorboxes] AS [t1] ON [t1].[s_id] = [t0].[sdr_s_id]
) AS [t2]
WHERE (DATEADD(HOUR, -DATEPART(HOUR, [t2].[sdr_timestamp]), DATEADD(MINUTE,
-DATEPART(MINUTE, [t2].[sdr_timestamp]), DATEADD(SECOND, -DATEPART(SECOND,
[t2].[sdr_timestamp]), DATEADD(MILLISECOND, -DATEPART(MILLISECOND,
[t2].[sdr_timestamp]), [t2].[sdr_timestamp]))) = @p0) AND ([t2].[s_id] =
@p1)
GROUP BY [t2].[value]
}

```

Část kódu 3 LINQ to SQL dotaz pro získání agregovaných hodnot

6.3.4 Pravidla použitá při implementaci

Při implementaci jsem se řídil zavedenými doporučeními pro konvence pojmenování přímo z .Net Frameworku. Nicméně některé odlišnosti jsou možné. Proto zde uvádím souhrn svých pravidel, kterými jsem doplnil doporučené .NET konvence.

- Každá třída je ve vlastním souboru, případně soubor obsahuje související logické celky (výčtové typy, apod.)
- Dlouhé řetězce jsou rozděleny do více řádků, stejně tak dlouhé příkazy.
- Dlouhé podmínky jsou rovněž rozděleny do více řádků.
- Zdrojový kód je psán anglicky.
- Kód je vhodně okomentován.
- Zprávy zobrazené uživateli jsou pokud možno krátké a srozumitelné.
- Ovládací prvky formulářů jsou popisovány stručně.
- Nepoužívají se odborné termíny (s výjimkou termínů vztahujících se k oboru, pro který je program navržen) ani slangové výrazy.
- Uživateli se netyká.
- V případě SQL dotazu jsou klíčová slova napsána velkými písmeny.
- Názvy tabulek a sloupců jsou napsány malými písmeny.
- Sloupec v tabulce začíná prefixem založeným na názvu tabulky.

6.4 Vlastnosti klientské aplikace

Při vývoji klientské aplikace se mi podařilo implementovat většinu zamýšlených vlastností. Proto bych rád v této kapitole shrnul funkce systému, které jsou z pohledu práce zajímavé. Stručně je popsáno, jakým způsobem jsou klíčové vlastnosti systému řešeny.

6.4.1 Podpora různých sériových protokolů

Hlavním požadavkem byla možnost do budoucna jednoduše rozšiřovat klientskou aplikaci o další sady sériových protokolů a podporu různých typů zařízení na straně solární elektrárny, ale i v základu byl již požadavek na implementaci dvou hlavních sériových protokolů s podporou více různých typů zařízení. Tyto požadavky jsem splnil díky rozdělení klientské aplikace na jednotlivé komponenty. Navrhl jsem také několik rozhraní, která jednotlivé komponenty implementují. Je pak velice snadné vytvořit novou komponentu, která bude implementovat dané rozhraní a díky tomu bude snadno včlenitelná do již existujícího systému. Kromě toho je celý systém rozdělen na několik vrstev, které mezi sebou mají volnou vazbu právě díky rozhraní. Komunikace mezi vrstvami je zajištěna pomocí událostí.

6.4.2 Aktualizace

Jedním z hlavních požadavků na klientskou aplikaci byla automatická aktualizace. Program je nyní sám schopen se aktualizovat. Nové aktualizace se stahují jednou za hodinu, tedy vždy po spuštění programu se spustí hodinový časovač, který kontroluje, zda jsou na serveru nové aktualizace a ty následně stáhne a rozbalí. V aktualizacím balíku se nachází aktualizací program, který překopíruje soubory do adresáře s hlavní aplikací a sám se ukončí a smaže. Aktualizace se stahují přes zabezpečený protokol HTTPS a rovněž aktualizací balík je zaheslovaný silným 20 místným heslem.

6.4.3 Licence

Pokud je nainstalována aplikace na solární elektrárnu, získá tak investor roční licenci na využívání programu. Po roce si musí licenci obnovit. Každý den se proto kontroluje, zda je licence ještě platná. Pokud se zjistí, že licence již vypršela, je o tom informován administrátor serveru. Klientská aplikace zastaví svou činnost a automaticky se z hostitelského počítače odstraní. Licence jsou kontrolovány vždy v noci, přibližně ve 3 ráno. Pokud se nepodaří získat informace o licenci, například z důvodu přerušeno spojení se serverem, opakuje se ověření

licence následující den ve stejnou dobu. Informace o nemožnosti ověření licence se zašle na server, pokud je to možné.

6.4.4 Kontrola spojení

Aplikace si sama dokáže kontrolovat spojení se serverem. Pro administrátory serveru je důležité mít přehled, zda jsou jejich klientské aplikace stále připojeny do sítě internet a mohou tak odesílat data do centrální databáze. Pokud by docházelo k častým výpadkům ve spojení, právě z důvodu nestabilního internetového připojení, budou administrátoři o tomto problému informováni. Klient pravidelně kontroluje, zda je připojen do sítě internet a zda je dostupný centrální server. Pokud tomu tak není, pokusí se odeslat report o tomto problému. Když se report neodešle, je uložen do interní databáze a odeslán při nejbližší příležitosti. Kontroly jsou prováděny pravidelně každých 5 minut, dále je zachycena událost vyvolaná odpojením ze sítě.

6.4.5 Ochrana před dekompilací

Jak již bylo zmíněno v některé z předchozích kapitol, knihovny vytvořené pomocí .NET Frameworku lze velice snadno dekompilovat. Vzhledem k silnému konkurenčnímu prostředí bylo potřeba se případné krádeži duševního vlastnictví ve formě zdrojových kódů chránit. Nástrojem pro ochranu je program Skater .NET Obfuscator, který perfektně slouží svému účelu. Bohužel díky tomuto požadavku nebylo možno použít instalátor integrovaný v IDE Visual Studio ani technologii ClickOnce pro snadné nasazení a aktualizaci.

6.5 Vlastnosti serverové části aplikace

Serverová část obsahuje zajímavé funkce, které bych rád v této části popsal. Budu se zabývat popisem vlastností uživatelského rozhraní ale i schopnostmi jádra aplikace.

6.5.1 Aktuální stavy

Webové rozhraní serveru nabízí uživateli mnoho informací o aktuálním stavu elektrárny. Mezi tyto informace patří aktuální výkon elektrárny, aktuální množství vyrobené elektrické energie, aktuální zisk, apod. Dále jsou zobrazeny aktuální informace o počasí v dané lokalitě a to na základě dat z meteorologické stanice umístěné v prostoru solární elektrárny. Uživateli se také zobrazují chybové zprávy a aktuální provozní stavy měničů.

Aktuální stavy jsou zjišťovány v průměru každých pět sekund, u informací o nových měnících je to však jedna minuta, protože tato informace se nemění tak často. V přehledech lze

nalézt také informace o předpovědi počasí na další dny. Tato informace je získávána z placeného serveru a je vztažena k lokalitě dané elektrárny. Předpověď počasí je přenášena pomocí souboru XML, který je na serveru je rozparsován a předán k zobrazení. Pro účely své diplomové práce jsem však využil ukázkový XML soubor od poskytovatele předpovědi počasí, takže data v administraci nejsou aktuální. Nic však nebrání nahrazení ukázkového souboru souborem aktuálním, avšak placeným.

6.5.2 Statistické přehledy

Statistické přehledy sumarizují do přehledných grafů různé informace. Mezi něž patří denní, měsíční i roční výnos či výkon. V přehledech uživatel najde také sumarizaci dat z meteorologické stanice, tedy průměrné teploty, množství dopadajícího světla a rychlost větru za předchozí den, měsíc či rok. Je také možno navolit různou agregaci dat, tedy zobrazení průměru, minima a maxima.

6.5.3 Diagnostika

V rámci administračního rozhraní může uživatel získat také diagnostické informace. Těmito informacemi se rozumí zobrazení poruch na straně elektrárny, detailní reporty o stavu elektrárny, detailní provozní stav zařízení (trafostanice, meteorologická stanice, měniče). V sekci diagnóza tak můžeme najít informace o stavech připojených klientů na straně elektrárny, zda dochází k výpadkům ve sběru dat (například z důvodu výpadku internetového připojení). Diagnostika poruch zařízení se provádí zjištěním poslední zaslané hodnoty ze zařízení a porovnáním času odeslání s aktuálním časem. Pokud je rozdíl větší, než jedna hodina je zařízení prohlášeno za poruchové. Zohledněna je samozřejmě denní doba.

6.5.4 Archiv dat

V archivu dat si může uživatel vyhledat zpětně data vztažená ke konkrétnímu časovému období a konkrétnímu zařízení (měnič, trafostanice, meteorologická stanice). Data jsou zobrazena v přehledných tabulkách. Uživatel může filtrovat jednotlivé sloupce, řadit je podle potřeby, anebo seskupovat například podle data. Data jsou získávána z centrální databáze a nejsou před zobrazením nijak upravována. Pro zachování původních hodnot nelze data v administraci mazat ani nijak upravovat.

7 Závěr

Závěr této práce je věnován shrnutí dosažených výsledků a doporučení pro další pokračování. V kapitole je také uveden přínos navrženého systému a jeho nasazení v praxi.

V rámci vývoje klientské aplikace byla také vypracována uživatelská dokumentace určená pro technické pracovníky, kteří budou v budoucnu systém instalovat na vybrané solární elektrárny. Tato uživatelská dokumentace nebyla z důvodu velkého rozsahu zahrnuta do tištěné verze diplomové práce, ale je obsažena v příloze.

7.1 Doporučení pro další pokračování práce

Na svou závěrečnou práci bych rád v budoucnu ještě navázal. Zejména proto, že mne téma solárních systémů zajímá a rád bych do něj pronikl hlouběji. V této problematice je stále ještě dostatek prostoru pro inovace i nové postupy, zejména sledování solárních elektráren je v ČR ještě na začátku svého vývoje.

Aplikaci bude potřeba do budoucna ještě rozšiřovat o schopnost monitorování dalších zařízení. Z toho vyplývá potřeba implementovat nové sériové protokoly a algoritmy sběru dat. Důležitou vlastností, je také vzdálené ovládání solárního systému, tedy nastavování jednotlivých parametrů podle vyhodnocených dat tak, aby byl výkon systému co nejlepší.

Velký prostor pro pokračování vidím v metodách analýzy dat. Zejména určování ztrát a predikce výkonu je velmi obsáhlá problematika, která by si zasloužila hlubší zpracování. Například predikce výkonu je zajímavou částí této problematiky. Je možno se zaměřit na souvislost mezi výkonem solární elektrárny a povětrnostními podmínkami v dané lokalitě. Díky případným studiím by bylo možno predikovat výkon solární elektrárny krátkodobě v řádech dnů a dlouhodobě v řádech týdnů.

Serverovou část by bylo v budoucnu dobré také vylepšit o propracovanější systém uživatelských rolí, aby byla zajištěna vyšší míra zabezpečení. Dále by bylo vhodné doplnit přehledové tabulky o filtraci jednotlivých hodnot, což vede ke zkvalitnění práce s velkým množstvím dat. Tyto úpravy nebyly vzhledem k zadání práce tolik podstatné a nepřinesly by žádný nový pohled na dané téma, proto jsou uvedeny na tomto místě, jako návrhy na další zlepšení.

Další kapitolou budoucího vývoje jsou malé solární systémy určené převážně na rodinné domy, panelové domy, apod. Taková monitorovací aplikace je svojí architekturou dosti odlišná, ale v základě by se dal využít systém navržený v této práci.

7.2 Shrnutí dosažených výsledků

Úvodní studie definovala velice rozsáhlý systém, podařilo se mi však analyzovat nejdůležitější části systému, které byly následně navrženy a implementovány.

Přínos mé práce je zejména v uceleném souhrnu znalostí z oboru monitorování fotovoltaických systémů. Na úvod jsem se zaměřil na rešerši stávajících komerčních řešení spolu s porovnáním jednotlivých systémů.

Mnou navržený systém je díky svým vlastnostem, jako je například sběr dat ze zcela odlišných typů zařízení, analýza těchto dat a monitorování provozního stavu solární elektrárny, unikátním svého druhu na českém trhu. V současné době mi není znám nikdo, kdo by systém podobného zaměření navrhl a úspěšně nasadil v praxi.

Již zmíněné sériové protokoly jsou implementovány pouze v proprietárních zařízeních firem, které tyto protokoly vyvinuly. Zatím jich využití v rámci konkrétního projektu podobného zaměření nebylo publikováno.

Při práci na svém systému, zvláště při vývoji klientské části, jsem musel překonávat různé problémy. Vzhledem k nemožnosti přímého vývoje, kdy bych mohl být fyzicky přítomen v prostoru solární elektrárny, bylo nutno využít přístupu přes vzdálenou plochu. Výpadky internetového spojení byly však velmi časté a velmi zdržovali postup vývoje.

Nejdříve bylo nutno věnovat čas kontrole zapojení celé solární elektrárny, na níž vývoj probíhal, kdy byla provedena revize desítek metrů kabelů a zkontrolováno více jak sto zařízení. Až po vyloučení případných hardwarových chyb bylo možno zahájit vývoj vlastního systému.

Nicméně i přes výše uvedené překážky věřím, že se mi podařilo navrhnout a implementovat systém, který je kvalitní a přínosný. Domnívám se, že systém bude dobře plnit úkol, pro který byl vytvořen a že bude dále úspěšně rozšiřován a vylepšován.

Myslím si, že díky svým unikátním vlastnostem systém dobře obstojí proti již existujícím řešením a získá si tak stabilní pozici na trhu se systémy pro monitorování a zpracování dat. Mou domněnku však prokáže jen čas a podpoří ji další práce na systému.

Výsledkem této práce je systém navržený a implementovaný podle zadání. Díky možnosti nasazení systému v praxi se ukázala kvalita implementace. V současné době je díky mému systému monitorováno 5 velkých fotovoltaických elektráren, které čítají řádově stovku měničů a dalších zařízení, denně jsou sesbírány a zanalyzovány stovky MB dat. Monitorovací systém již dokázal odhalit několik výpadků zařízení a snížit tak ztráty na zisku z prodeje vyrobe-

né elektrické energie. Samozřejmě je celý tento systém monitorování a zpracování dat stále ve vývoji a v testování, bude však postupně nasazován na další elektrárny.

Závěrem zmíním ještě přínos této práce pro mne samotného. Získal jsem zkušenosti při vývoji komplexního systému, seznámil jsem se zajímavou technologií a úspěšně jsem se nasadil monitorovací systému do provozu. Mohu říci, že do budoucna se jistě budu problematice monitorování solárních elektráren a vyhodnocování získaných dat dále věnovat, a budu své znalosti dále rozvíjet.

8 Seznam použitých zdrojů

8.1 Seznam literatury

- (1) GOOK, Michael. *Hardwarová rozhraní - průvodce programátora*. 1. vyd. Brno: Computer Press, 2006. 464 s. ISBN 80-251-1019-2.
- (2) NAGEL Christian, EVJEN Bill, GLYNN Jay, WATSON Karli, SKINNER Morgan. *C# 2008 Programujeme profesionálně*. 1. vyd. Brno: Computer Press, 2009. 1904 s. ISBN 978-80-251-2401-7.
- (3) EVJEN Bill, HANSELMAN Scott, RADER Devon. *ASP.NET 3.5 v jazycích C# a Visual Basic - Programujeme profesionálně*. 1. vyd. Brno: Computer Press, 2009. 1600 s. ISBN 978-80-251-2069-9.
- (4) CONOLLY Thomas, BEGG Carolyn, HOLOWCZAK Richard. *Mistrovství - Databáze, Profesionální průvodce tvorbou efektivních databází*. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
- (5) SMA Regelsysteme GmbH, *SMA Data Specification – Definition and Description of the Data Telegram Format and Communication Protocol*. Version 1.25. SMA, 2003. 83 s. SMADAT-12:ZE2203.
- (6) Papouch s. r. o., *Quido Spinel - Kompletní popis komunikačního protokolu I/O modulů Quido*. Verze 2.11. Papouch s. r. o., 2007. 48 s.
- (7) BERÁNEK, Václav. *Postup a příprava pro realizaci fotovoltaického systému pro typické domovní instalace do výkonu 5 kWp: Fotovoltaická elektrárna „snadno a rychle“*. *Energie 21*. 2008, 3, s. 28-31.
- (8) BERÁNEK, Václav. *Fotovoltaická elektrárna v Rokytnici nad Jizerou. Jemná mechanika a optika*. 2009, 9, s. 14-20.
- (9) BERÁNEK, Václav. *Solarmon*. Archiv autora, plánované vydání 9/2010, periodikum neznámé.

8.2 Seznam internetových zdrojů

- (10) *Fotovoltaika princip* [online]. 2009, [cit. 2010-03-12]. Dostupné z:
< http://www.ceska-solarni.cz/fotovoltaika_princip.php >.
- (11) *Fotovoltaika - Princip fotoelektrického jevu* [online]. 2010, [cit. 2010-03-13]. Dostupné z: < <http://www.solartec.cz/cs/fv-systemy/o-fotovoltaice/fotovoltaika.html> >.
- (12) *Fotovoltaika v ČR: jaké jsou zkušenosti majitelů solární elektrárny?* [online]. 12. 08. 2009, [cit. 2010-03-15]. Dostupné z:
< <http://www.nazeleno.cz/energie/fotovoltaika-1/fotovoltaika-v-cr-jake-jsou-zkusenosti-majitelu-solarni-elektrarny.aspx> >.
- (13) *SOAP vs. REST API Implementation* [online]. 17. 12. 2008, [cit. 2010-04-20]. Dostupné z: <http://www.fliquidstudios.com/2008/12/17/soap_vs_rest_api/>
- (14) MALÝ Martin. *REST: architektura pro webové API* [online]. 3. 8. 2009, [cit. 2010-04-22]. Dostupné z: < <http://zdrojak.root.cz/clanky/rest-architektura-pro-webove-api/> >

9 Seznam příloh

Příloha A – Obsah přiloženého CD.

Příloha B – Obrazová dokumentace částí systému.

Příloha A

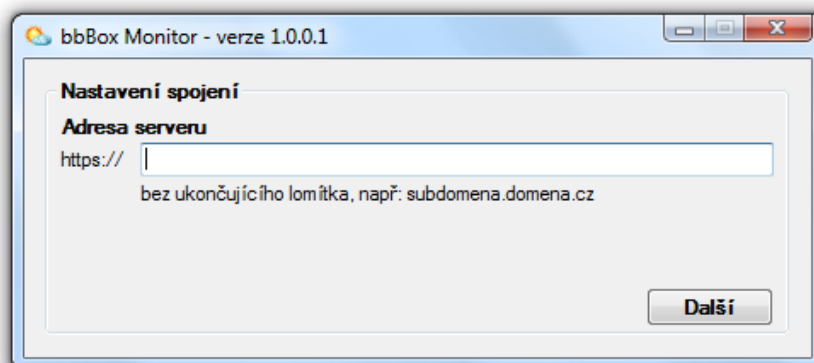
1. Aplikace

- a. SolarMonitor.zip (archiv zip se zdrojovými kódy klientské aplikace)
- b. SolarServer.zip (archiv zip se zdrojovými kódy serverové aplikace)

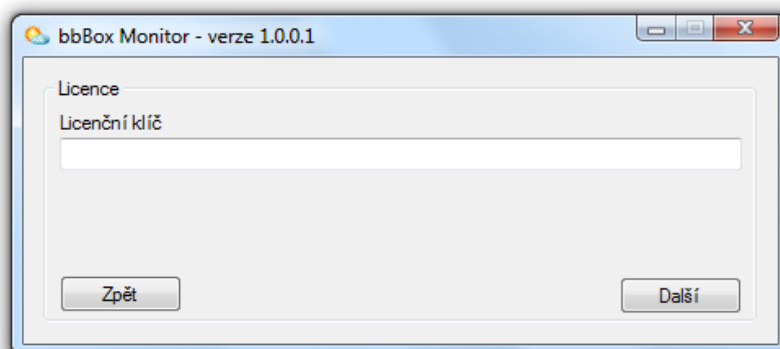
2. Dokumenty

- a. DiplomovaPrace.pdf (text samotné diplomové práce)
- b. UserDoc.pdf (uživatelská příručka)

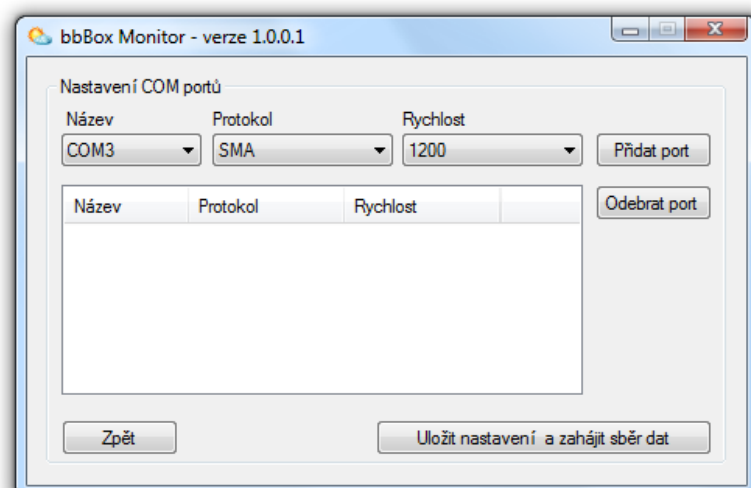
Příloha B



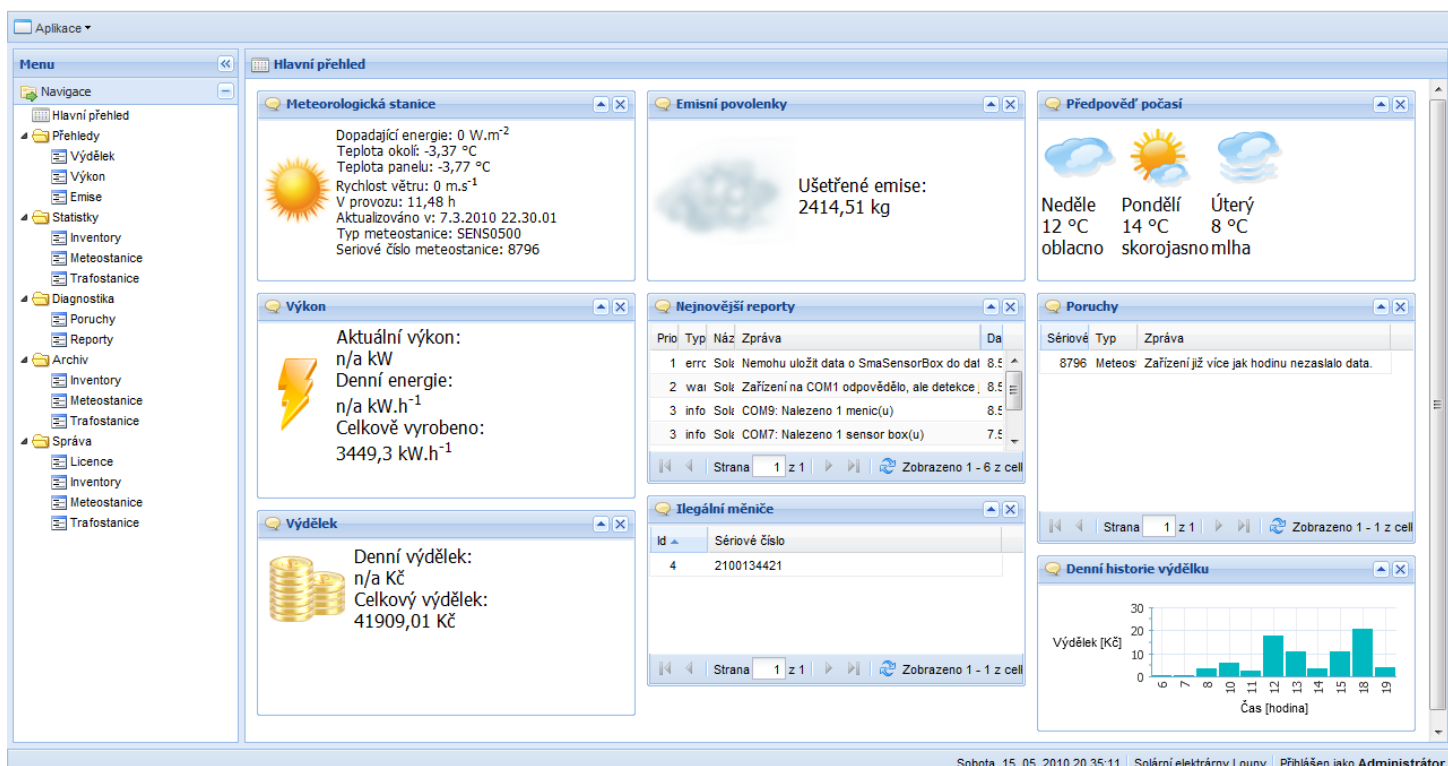
Obrázek 16 Klient - zadání adresy serveru



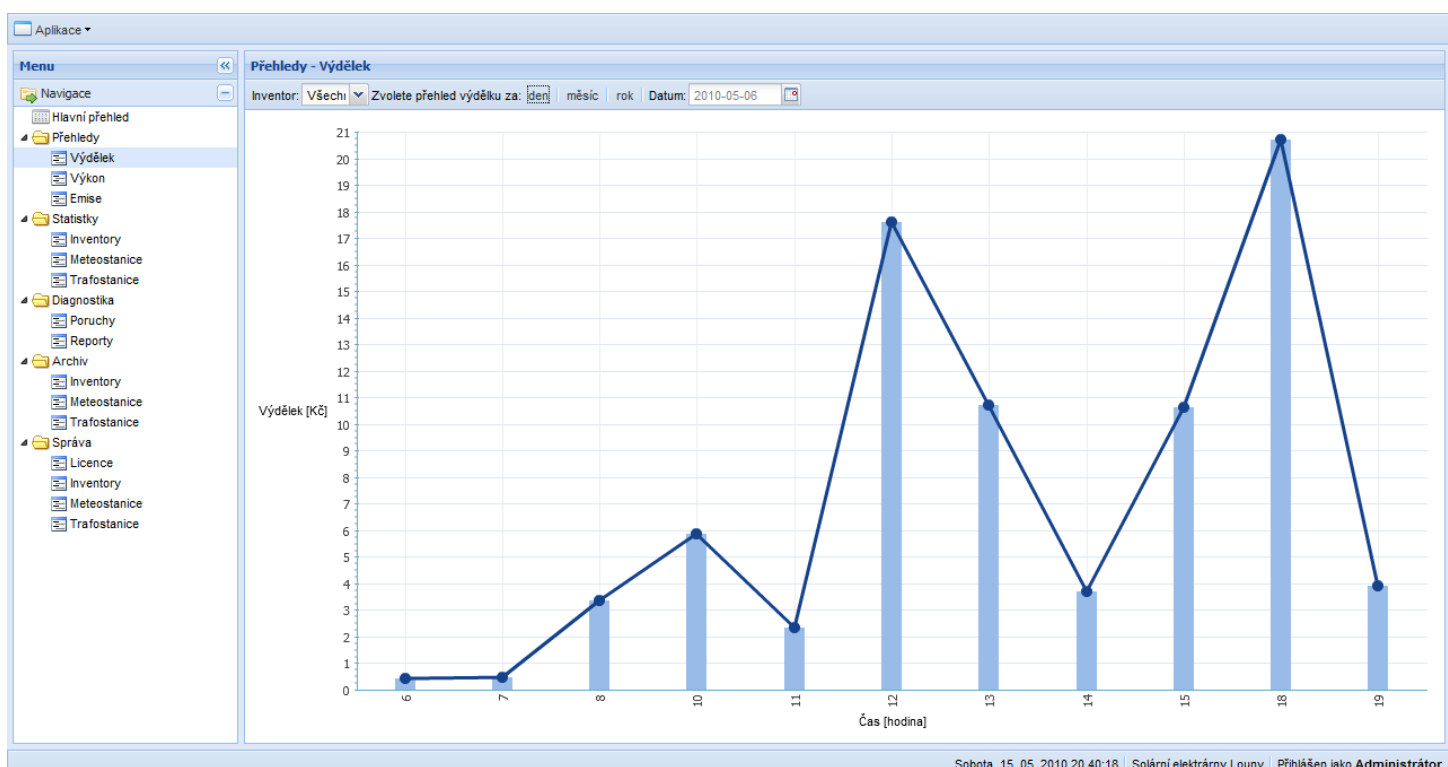
Obrázek 17 Klient - zadání licenčního klíče



Obrázek 18 Klient - nastavení sériových portů



Obrázek 19 Server - úvodní strana administrace obsahující nejdůležitější informace



Obrázek 20 Server - Graf hodinového vývoje zisku v konkrétním dni

Applikace ▾

Menu <<

Navigace
 Hlavní přehled
 Přehledy
 Výdělěk
 Výkon
 Emise
 Statistky
 Inventory
 Meteostanice
 Trafostanice
 Diagnostika
 Poruchy
 Reporty
 Archiv
 Inventory
 Meteostanice
 Trafostanice
 Správa
 Licence
 Inventory
 Meteostanice
 Trafostanice

Správa - Inventory

Nový inventar Odebrat inventar

Id	Monitor	Název	Sériové číslo	Výrobce	Typ	Max. string	Výstup	Výkon. tř.	Počet panelů	Legalizován?	Legalizováno
4	SolarMonitor1	INV 1	2100134421	SMA Solar Technology AG	SB 5000TL-20	4	5000	A	10	Ne	19.03.2010
5	SolarMonitor1	INV 2	2100134235	SMA Solar Technology AG	SB 5000TL-20	4	5000	A	10	Ano	19.03.2010
6	SolarMonitor1	INV 3	2100132255	SMA Solar Technology AG	SB 5000TL-20	4	5000	A	10	<input checked="" type="checkbox"/>	19.03.2010
7	SolarMonitor1	INV 10	2001081737	SMA Solar Technology AG			9000	A	13	Ano	19.03.2010
8	SolarMonitor1	INV 11	2001037804	SMA Solar Technology AG			9000	A	13	Ano	19.03.2010
9	SolarMonitor1	INV 12	2000990234	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010
10	SolarMonitor1	INV 4	2001157248	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010
11	SolarMonitor1	INV 5	2001177090	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010
12	SolarMonitor1	INV 6	2001177050	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010
13	SolarMonitor1	INV 7	2001157434	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010
14	SolarMonitor1	INV 8	2001177045	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010
15	SolarMonitor1	INV 9	2001157228	SMA Solar Technology AG	SMC 11000TL-10	5	11000	A	13	Ano	19.03.2010

Uložit Zrušit

Strana 1 z 1

Zobrazeno 1 - 12 z celkových 12

Sobota, 15. 05. 2010 20:42:45 | Solární elektrárny Louňv | Přihlášen jako Administrátor

Obrázek 21 Server - úprava informací o měniči