

**Fakulta matematiky, fyziky a informatiky Univerzity  
Komenského**

# **Návrh na SVOŠA**

## Obsah

0 Úvod .....	3
0.1 Legenda .....	3
1 Konceptuálna analýza.....	3
1.1 Stavové diagramy .....	3
1.2 Entitno-relačný diagram .....	6
1.3 Use-case diagram .....	7
1.4 Užívateľské rozhranie .....	7
2 Analýza technológií .....	7
2.1 Úvod .....	7
2.2 Používané technológie .....	8
3 Komponentný diagram.....	10
3.1 Komponent prvotné nastavovanie .....	10
3.2 Komponent meranie.....	11
3.3 Komponent spracovanie dát .....	11
3.4 Komponent uloženie dát .....	11
3.5 Komponent kreslič grafov.....	11
4 Triedny diagram .....	12
5 Dátový model zobrazený pomocou entitno-relačného diagramu .....	13

## 0 Úvod

Návrh je dokument opisujúci ako bude presnejšie fungovať náš software SVOSA. Tento dokument obsahuje predošlé dokumenty: Diagramy, Konceptuálna analýza, Analýza technológií, návrh rozhrania a popis dátového modelu spolu s podrobnou špecifikáciou komponentov. Tento dokument je určený pre skupinu programátorov, tak by došla z hľadiska funkcionality a vlastností softvéru k rovnakému riešeniu.

### 0.1 Legenda

**SA** - Spektrálny analyzátor.

**SVOSA** - Softvér pre vzdialené ovládanie spektrálneho analyzátora.

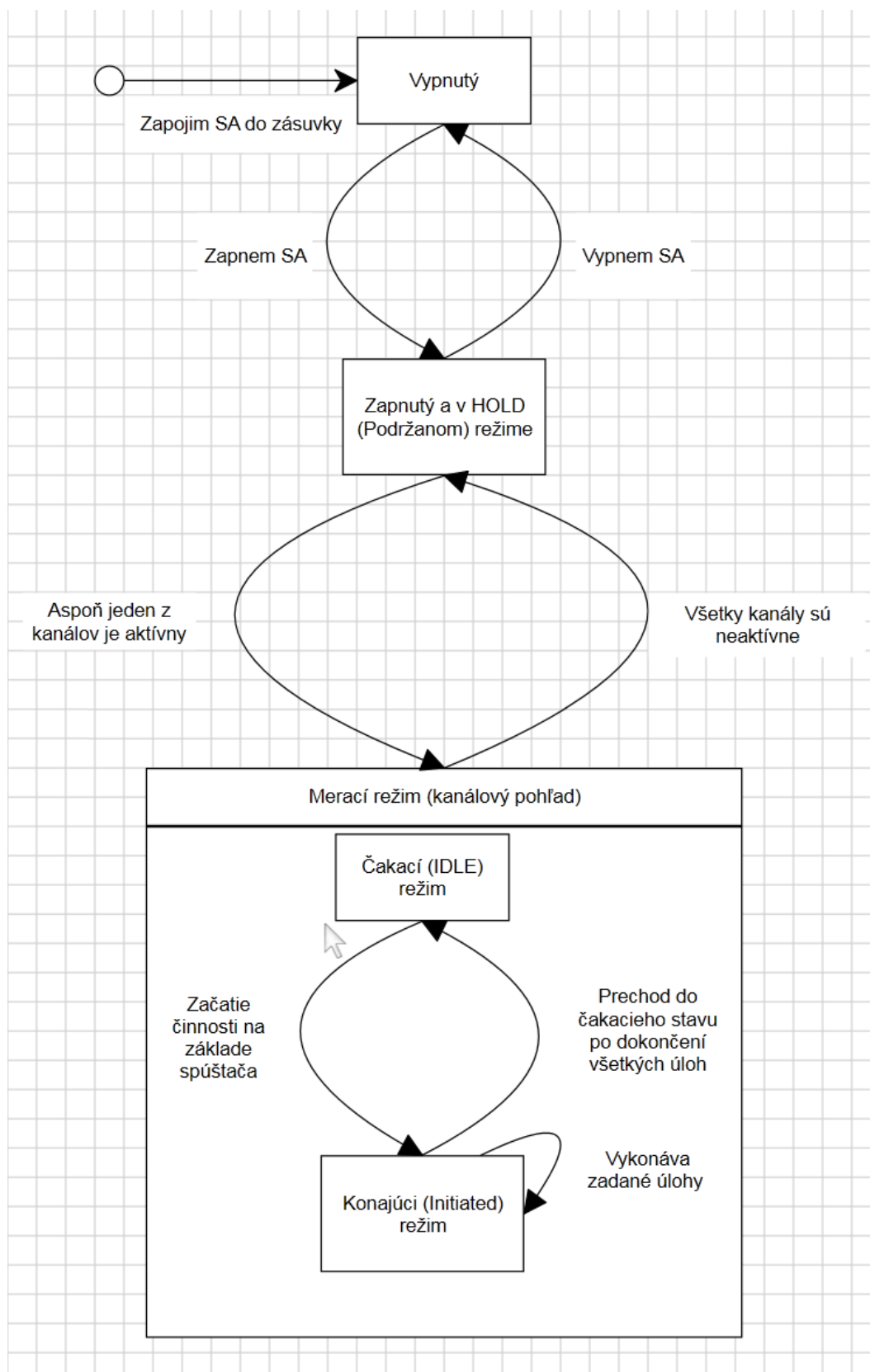
## 1 Konceptuálna analýza

Tento dokument slúži na logické odvodenie požiadaviek z platného katalógu požiadaviek, súvislosti a prezentáciu základných dát. Obsahuje diagramy to entitno-relačný diagram, stavové diagramy, a use-case diagram. Na konci tohto dokumentu je legenda s vysvetlivkami pojmov a popis užívateľského rozhrania.

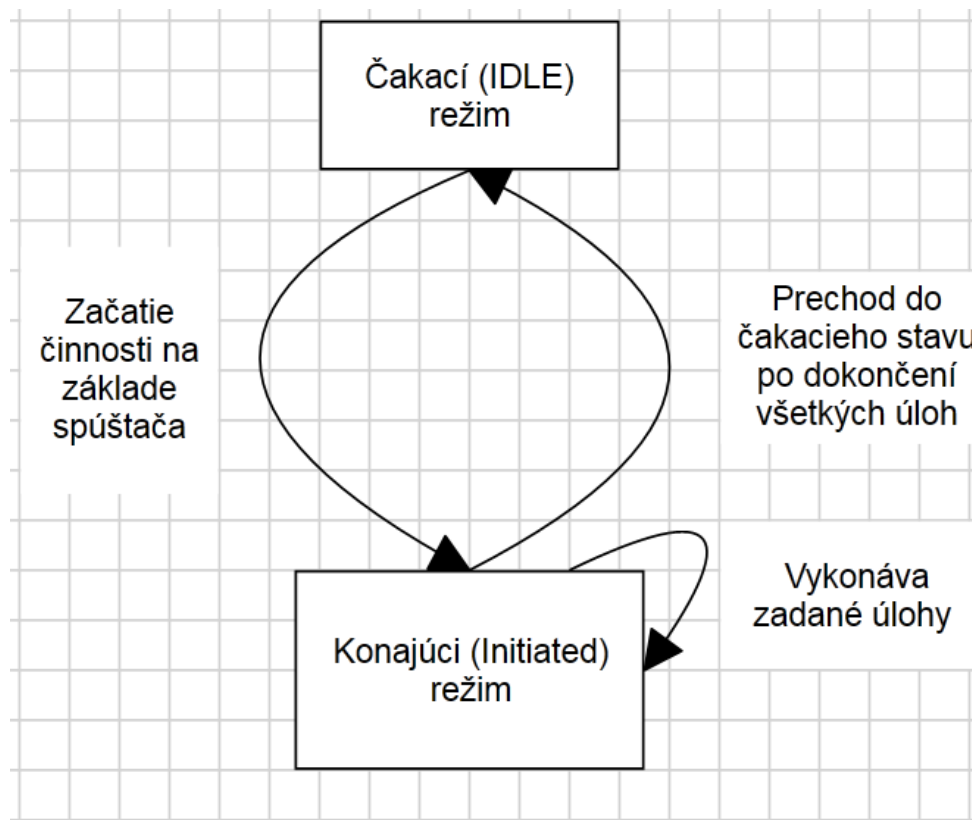
### 1.1 Stavové diagramy

Diagramy vysvetľujú ako z pohľadu SA funguje proces experimentu, teda jeho hlavnej činnosti.

Bližšie detaily sú vysvetlené v dokumentácii [2] na 81 strane.



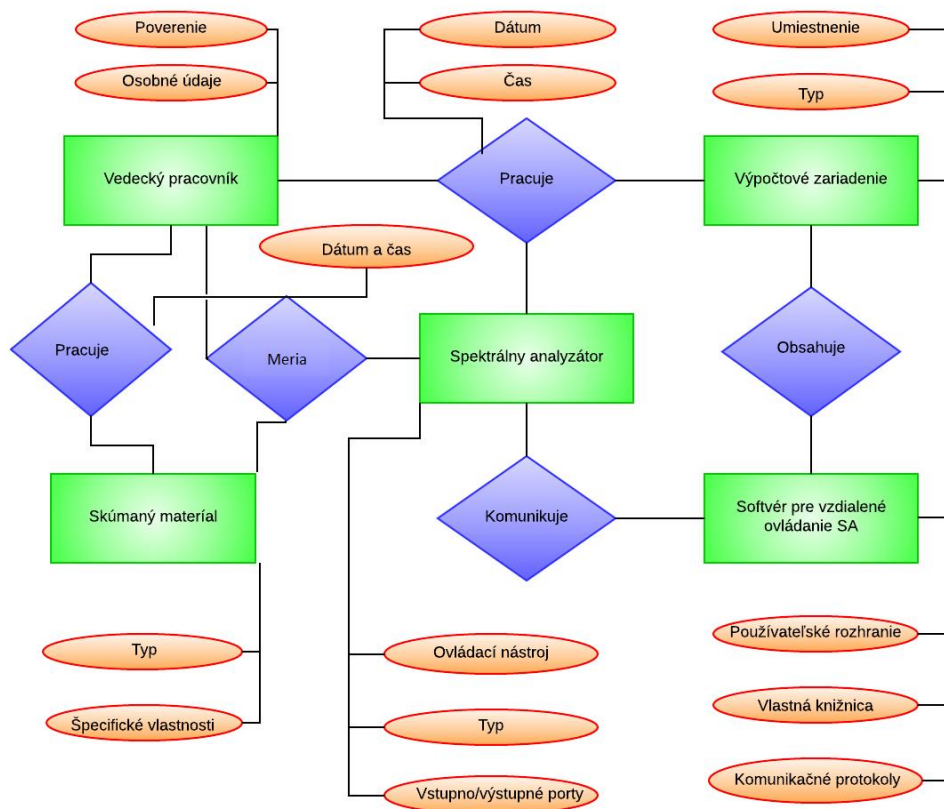
Obrázok 1.1 : Vysvetľuje systémový pohľad na proces experimentu z pohľadu SA.



Obrázok 1.2: Vysvetľuje kanálový pohľad na proces experimentu z pohľadu SA. Samotný SA vie vykonávať experiment vo viacerých kanáloch, čo zvyšuje rýchlosť experimentu, lebo vie robiť viac vecí naraz.

## 1.2 Entitno-relačný diagram

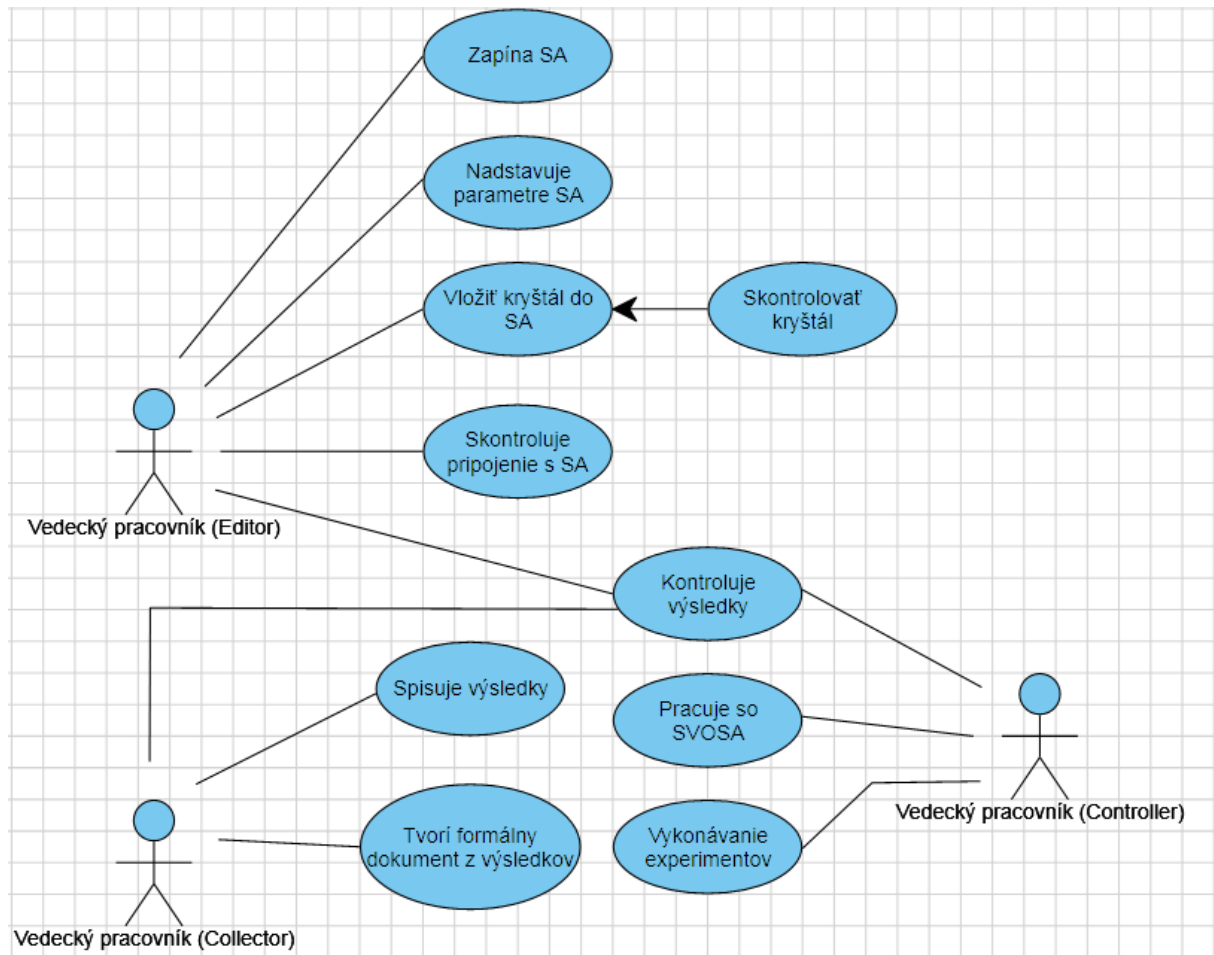
Diagram znázorňuje jednotlivé entity vstupujúce do experimentu a ich vzťahy (relácie) medzi sebou v priebehu tohto experimentu.



Obrázok 1.3: Znázorňuje jednotlivé vzťahy medzi entitami a ich atribúty v procese experimentu.

### 1.3 Use-case diagram

Use-case diagram je najjednoduchší spôsob reprezentácie vzťahov medzi užívateľom systému popísaný stavmi použitia. Tento diagram popisuje, ako užívateľ reaguje počas experimentu so SA a so softvérom SVOSA.



Obrázok 1.4: Popisuje interakcie užívateľov so systémom. Jednotlivé role môže vykonávať aj jeden človek, ale aj skupina.

### 1.4 Užívateľské rozhranie

Užívateľské rozhranie bude spočívať vo forme príkazového riadku, do ktorého používateľ bude volať procedúry a metódy SVOSA. Užívateľ potom môže dodefinovať ďalšie argumenty k programu. Ak uvedenie 1 argument, tak ten je definovaný ako dvojica frekvencie a impedancie. Ak uvedie ešte jeden parameter, bude to meno súboru, do ktorého sa uložia dáta do prehľadnej tabuľky (do relatívneho priečinka odkiaľ bol súbor spustený). Ostatné argumenty budú ignorované.

## 2 Analýza technológií

### 2.1 Úvod

Tento dokument má slúžiť na popis technológií, ktoré sú potrebné na plnú funkcionálnosť ISVOS-y. Taktiež opisuje komponenty, ktoré budú v našom systéme a taktiež zobrazuje dátový model a triedny diagram.

Ako bolo stanovené v dokumente "Katalóg požiadaviek", v sekcii "2.1.1 Používateľské rozhrania" projekt bude písaný v programovacom jazyku Python.

## 2.2 Používané technológie

### 2.2.1 Možnosti technológií pripojenia SVOSA k SA:

1. LAN - (Local Area Network) systém vzdialeného ovládania poskytuje dve metódy :
  - SICL-LAN – v ovládacom systéme používajúcom SICL-LAN server, komunikácia medzi vonkajším ovládačom (klientom) a SA (serverom) je uskutočnená použitím SICL-LAN protokolu. Samotná komunikácia je uskutočnená pomocou SICL (Standard Instrument Control Library). Užívateľ môže ovládať SA programovaním pomocou SICL alebo VISA v jazyku C pod operačným systémom Linux, alebo Visual C++, Visual Basic a VEE pod operačným systémom Windows.
  - Telnet – v ovládacom systéme cez telnet server, komunikácia je uskutočňovaná cez pripojenie medzi zásuvnými modulmi poskytovanými procesmi vonkajšieho ovládača a SA na nadviazanie sieťového spojenia medzi nimi. Ovládací systém cez telnet môže komunikovať dvoma portami a to:
    - port 23 – používa sa na dialógové ovládanie používajúce telnet (užívateľské rozhranie pre TELNET protokol)
    - port 5025 – používa sa na ovládanie z programu
2. GBIP – (General Purpose Interface Bus) je štandardné rozhranie pre pripojenie počítačov a periférnych zariadení, ktoré podporuje nasledujúce medzinárodné štandardy : IEEE 488.1, IEC-625, IEEE 488.2, a JIS-C1901 . GPIB rozhranie umožňuje ovládať Agilent SA z externého počítača. Počítač odosiela príkazy a pokyny na SA a prijíma dáta odoslané z SA cez GPIB .

### Výber technológie pripojenia SVOSA k SA

Na základe špecifikácie požiadaviek sme sa rozhodli uprednostniť LAN technológiu, keďže má byť SA ovládaný cez lokálnu sieť, s použitím metódy SICL-LAN pretože ovládací počítač má bežať pod operačným systémom Linux.



### 2.2.2 Hardwarové technológie

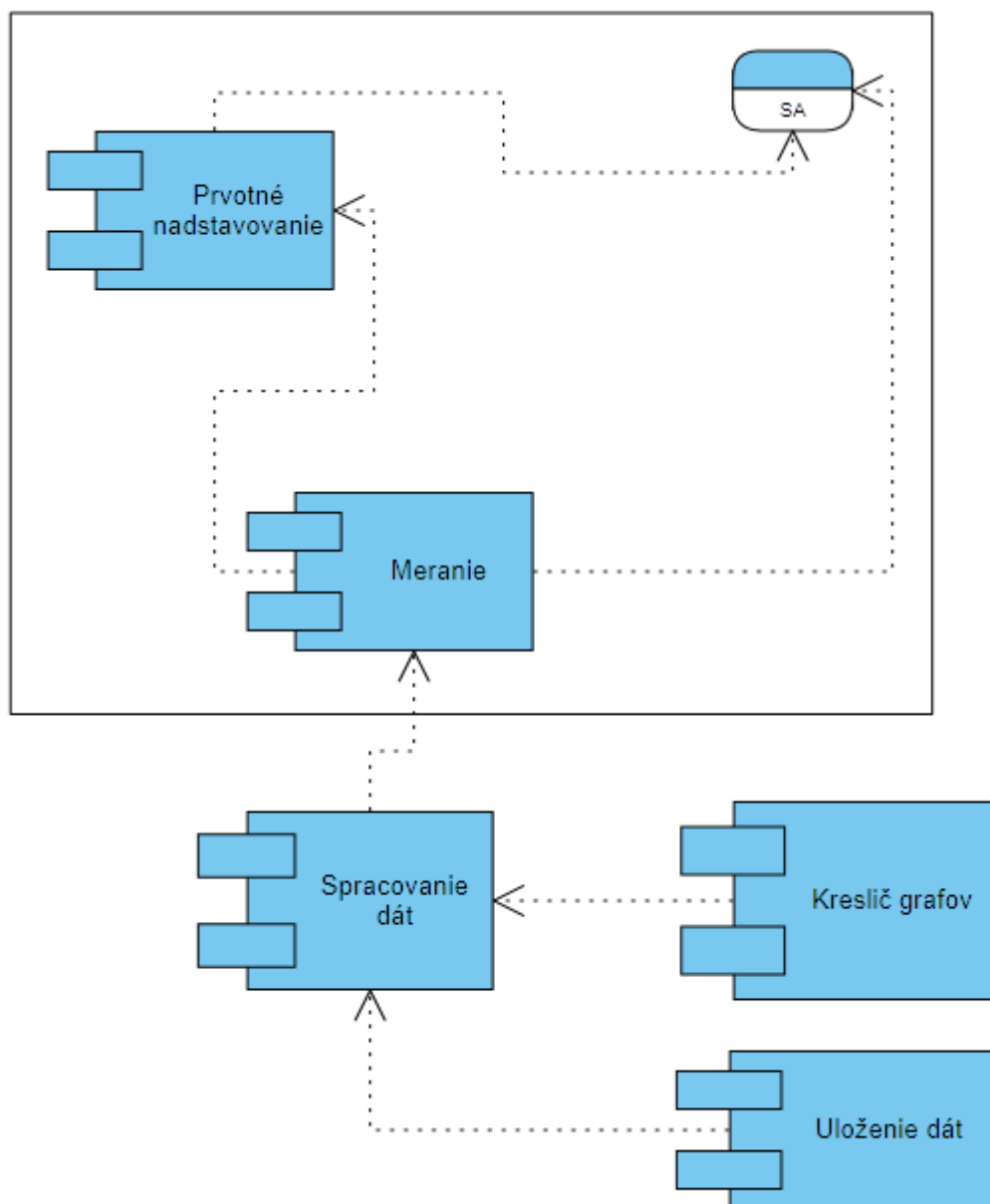
1. Náš systém bude používať technológiu TCP/IP na spojenie sa s koncovým zariadením. Očakáva sa že to bude väčšinou notebook alebo PC. Prípadné rozšírenie je zapojenie zariadenia do sieťového prvku (napríklad rozbočovač).

### 2.2.3 Softwarové technológie

Náš systém bude používať rôzne knižnice pre dosiahnutie konkrétnych požiadaviek

1. Na vykresľovanie grafov budeme používať knižnicu "Gnuplot", ktorá dokáže vykresľovať rôzne typy grafov. Všeobecnejšie sa dá používať aj ako nástroj v Linuxe na kreslenie grafov cez príkazový riadok.
2. Taktiež na nízko úrovňovú sieťovú komunikáciu s SA budeme potrebovať aj knižnicu menom "socket". Budeme vedieť pomocou nej sa pripojiť na konkrétnu IP adresu na konkrétny port nášho SA.
3. Knižnica "math" budeme používať na rôzne výpočty matematických vzorcov a volanie matematických funkcií.

### 3 Komponentný diagram



Obrázok 3.1

#### 3.1 Komponent prvotné nastavovanie

Komponent slúži na nastavenie všetkých potrebných parametrov potrebných na napojenia sa pomocou siete na SA. Okrem iného sa v ňom nastavujú dôležité konštanty ako napríklad adresa zariadenia na sieť či príslušný port, ktoré užívateľ ďalej nemôže meniť. Tu sa budú dať aj nastaviť ostatné parametre ako bolo už definované v katalógu požiadaviek.

### **3.2 Komponent meranie**

Komponent slúži na posielanie pokynov na meranie a ich spätné prijímanie v nadstavenom formáte. Dáta sa ukladajú do internej štruktúry. Spracovanie dát zo štruktúry je už spracúvaný iným systémom, ktorý dátam rozumie.

### **3.3 Komponent spracovanie dát**

Komponent prečíta dáta z internej štruktúry a preloží do ďalších potrebných štruktúr a počítajú sa potrebné veličiny z dodaných dát pre ich ďalšie využitie.

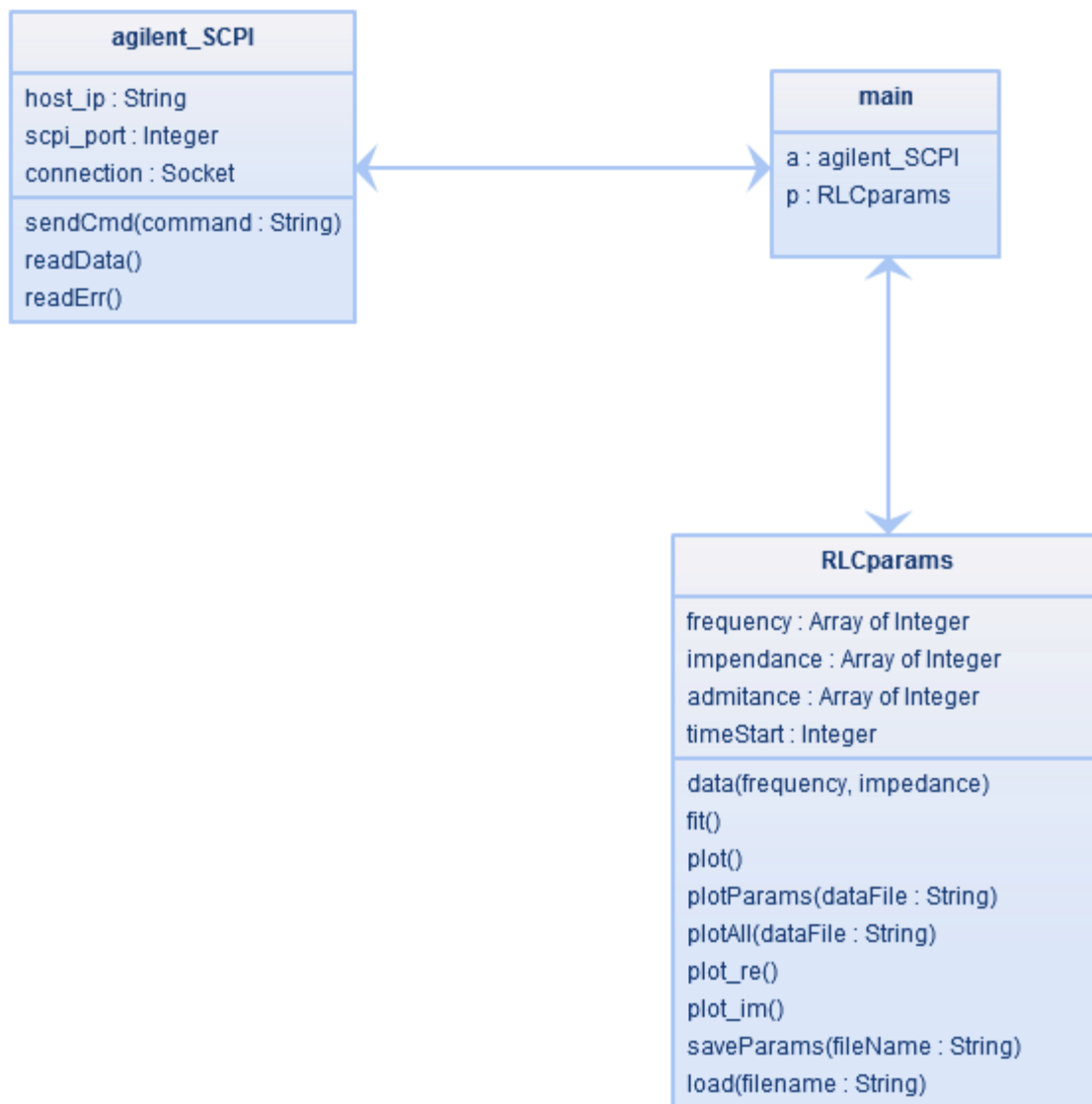
### **3.4 Komponent uloženie dát**

Dáta sa ukladajú do prehľadnej tabuľky, ak sa užívateľ rozhodne inak sa dáta nebudú ukladať.

### **3.5 Komponent kresliť grafov**

Z poskytnutých spracovaných a vyrátaných dát a veličín dokáže tento komponent zobrazíť potrebné grafy.

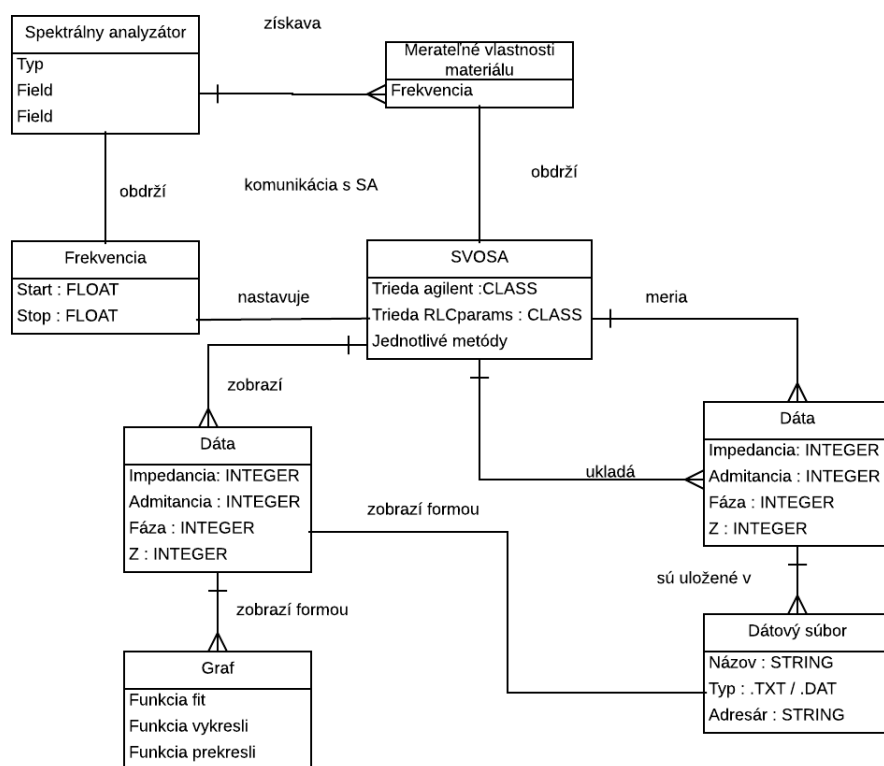
## 4 Triedny diagram



Obrázok 4.1

Triedny diagram znázorňuje štruktúru projektu, popisuje jeho triedy a ich vzájomné prepojenie. Hlavný program main spravuje a inicializuje triedy agilent\_SCPI a RLCparams. SVOSA pomocou triedy agilent\_SCPI nadviaže spojenie zo SA, a následne pomocou tejto triedy aj prijme dáta, ktoré po úspešnom prijatí analyzuje a graficky znázorní pomocou triedy RLCparams.

## 5 Dátový model zobrazený pomocou entitno-relačného diagramu



Obrázok 5.1

Podrobnejší entitno-relačný diagram znázorňujúci dátový model Informačného systému.

SVOSA pracuje a analyzuje získané dáta od SA a tie následne dokáže graficky znázorňovať, alebo ich dokáže ukladať (exportovať) do textového, alebo dátového súboru s ktorým môžu pracovníci ďalej pracovať.