



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Editor konfiguračních souborů Flow123d

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 1802T007 – Informační technologie

*Autor práce:* **Bc. Tomáš Křížek**

*Vedoucí práce:* doc. Ing. Jiřina Královcová, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Editor for Flow123d configuration files

## Master thesis

*Study programme:* N2612 – Electrotechnology and informatics

*Study branch:* 1802T007 – Information technology

*Author:* **Bc. Tomáš Křížek**

*Supervisor:* doc. Ing. Jiřina Královcová, Ph.D.



Tento list nahrad'te  
originálem zadání.

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

## **Abstrakt**

Český abstrakt

## **Abstract**

English abstract

## **Poděkování**

# Obsah

Seznam zkratek . . . . .	8
<b>1 Problematika</b>	<b>11</b>
1.1 Software Flow123d . . . . .	11
1.2 Konfigurační soubory . . . . .	12
1.3 Formát pro výměnu dat . . . . .	15
<b>2 Analýza</b>	<b>19</b>
2.1 Specifikace konfiguračních souborů Flow123d . . . . .	19
2.2 Formát YAML pro konfigurační soubory . . . . .	19
<b>3 Návrh</b>	<b>20</b>
3.1 Autokonverze . . . . .	20
3.2 Validace . . . . .	20
3.3 Kontextová dokumentace . . . . .	20
3.4 Automatické doplňování textu . . . . .	20
3.5 Funkce editoru . . . . .	20
3.6 GUI . . . . .	20
<b>4 Implementace a testování</b>	<b>21</b>
<b>5 Uživatelská příručka</b>	<b>22</b>
5.1 Zápis konfiguračních souborů pomocí YAML . . . . .	22
5.2 Vytváření a editace konfiguračních souborů . . . . .	22
5.3 Validace a odstranění chyb . . . . .	22
5.4 Nastavení . . . . .	22
<b>Literatura</b>	<b>24</b>

## Seznam zkratk



# Úvod

Existuje celá řada softwarů, která pro zajištění požadované funkce potřebuje správné nastavení, podle kterého pak daný program přizpůsobí svoji činnost. Může se jednat o počáteční konfiguraci, jako je tomu například u serverových aplikací nebo e-mailových klientů. Tato konfigurace se zpravidla dále nemění, pokud nedojde k nějakým podstatným změnám.

Oproti tomu existují programy, od kterých se očekává, že budou spouštěny s širokou škálou různých nastavení. U těchto aplikací se typicky konfigurace předává při spuštění jako jeden ze vstupních parametrů. Činnost těchto programů se pak zásadně liší dle zvolené konfigurace.

Takovou aplikací je například simulátor Flow123d, který se používá pro modelování procesů v horninovém prostředí. Vstupem do této aplikace je výpočetní síť společně se zadáním úlohy. Tato úloha je definovaná pomocí konfiguračního souboru, který vytváří uživatel. Po zadání vstupních dat provede simulátor výpočty na dané síti a výsledky uloží do datového souboru, který je výstupem z aplikace.

Software Flow123d podporuje různé typy úloh. Konfigurace jednotlivých úloh vyžaduje odlišné nastavení a může tedy dojít k tomu, že uživatelem zadaná konfigurace je nevalidní – například kvůli tomu, že definice dané úlohy neobsahuje některé povinné parametry a je tedy neúplná. V souboru může vzniknout i syntaktická chyba, která způsobí, že zadaná data nelze správně interpretovat.

Popis formátu konfiguračních souborů pro Flow123d je poměrně rozsáhlý – samotná referenční příručka, která ho popisuje, obsahuje několik desítek stran. To klade na uživatele velké nároky. Pokud chce například ověřit, že byly zadány všechny povinné parametry, buď musí mít se softwarem rozsáhlé zkušenosti, nebo musí trávit velké množství času prohledáváním dokumentace.

Celá situace je dále komplikována tím, že formát konfiguračních souborů se mění s tím, jak se vyvíjí nové a upřesňují stávající funkcionality softwaru Flow123d. Může dojít k tomu, že některé změny ve formátu konfiguračních souborů nemusí být zpětně kompatibilní. Uživatel tedy potřebuje znovu prostudovat rozsáhlou referenční doku-

mentaci, aby zjistil, jakým způsobem zadat dříve realizovanou úlohu pro novou verzi Flow123d.

Pokud se stane, že uživatel spustí Flow123d s nevalidní konfigurací, potom během inicializace dojde k chybě, o které se uživatel dozví pomocí textového rozhraní, ve kterém se Flow123d spouští. Jelikož se může jednat o výpočetně náročné úlohy, které se často pouští na vzdáleném výpočetním clusteru, je tento proces poměrně časově náročný a uživatelsky nepříjemný.

Při vzdáleném spouštění Flow123d se úloha zařadí do fronty na výpočetním clusteru, kde dále čeká na přidělení zdrojů. Ty se přidělují na základě aktuálního vytížení. Bud' jsou k dispozici okamžitě, nebo je nutné čekat na dokončení některých předchozích úloh. Může tedy nastat situace, kdy uživatel zařadí úlohu do fronty a poté čeká na výsledky několik hodin nebo dokonce dní, a teprve potom zjistí, že v konfiguračním souboru, který vytvořil, byla chyba. Kvůli nemohlo dojít k inicializaci úlohy a tím pádem ani neproběhla simulace.

Tyto důvody byly hlavní motivací ke vzniku speciálního editoru pro konfigurační soubory Flow123d, který práci s nimi značně zjednoduší a usnadní. Editor zrychlí proces odstranění chyb tím, že je odhalí už v průběhu vytváření nebo upravování konfiguračních souborů. To uživateli umožní chyby odstranit ještě před tím, než předloží konfigurační soubor softwaru Flow123d. Tím dojde ke značné časové úspoře obzvláště v případech, kdy se výpočetní úloha spouští vzdáleně.

Součástí editoru má být grafické uživatelské rozhraní. Jedním z jeho hlavních přínosů bude zjednodušení přístupu k dokumentaci. Uživatel bude mít k dispozici tu část dokumentace, která bezprostředně souvisí s právě upravovanou částí konfiguračního souboru. Tato forma nápovědy by měla uživateli poskytnout alternativu k prohledávání rozsáhlé referenční dokumentace.

Dále bude editor umožňovat zobrazit datovou strukturu, která tvoří konfigurační soubor. Kromě toho bude editor poskytovat základní funkce pro práci s textovými soubory, jako je podpora operací se schránkou, možnost vrátit či opakovat změny, vyhledávání či nahrazení textu a další. Editor má podporovat platformy Windows a Linux.

# 1 Problematika

## 1.1 Software Flow123d

Flow123d je software, který slouží k výpočtu proudění v porézním médiu, transportu látek nebo transportu tepla. Jedná se o aplikaci, která je orientována na práci s daty, a vzhledem k tomu neobsahuje žádné grafické uživatelské rozhraní. Uživatel tedy s aplikací pracuje v textovém režimu prostřednictvím terminálu, kde může aplikaci předat vstupní soubory a případně další parametry.

Na obrázku 1 jsou znázorněny vstupy a výstupy simulátoru Flow123d spolu s pomocnými aplikacemi, které uživatelé často používají. Vstupem do simulátoru Flow123d jsou dva soubory. První z těchto souborů popisuje výpočetní síť pomocí seznamu uzlů a elementů. Jedná se o textový soubor ve formátu `.msh`. Tuto síť generují softwary GMSH nebo SALOME. Druhým vstupním souborem je konfigurační soubor, který popisuje řešenou úlohu. Tento soubor si prozatím uživatelé tvořili sami pomocí obvyklých textových editorů.

Pro tento konfigurační soubor vzniká v rámci této práce specializovaný editor s označením ModelEditor, který má oproti obvyklému textovému editoru poskytnout např. validaci zadaných dat, zobrazení kontextové dokumentace nebo automatické doplňování textu. Vytváření a editace konfiguračních souborů se tak uživateli značně zjednoduší. ModelEditor je jednou ze součástí aplikace GeoMop, která obsahuje i další komponenty.

GeoMop má sloužit jako nástroj, který usnadní práci se simulátorem Flow123d. Jeho další komponenty mají za úkol např. zajišťovat vzdálené spouštění Flow123d na výpočetních clusterech. To bude zajišťovat modul JobsScheduler, který sjednotí rozhraní a postup spouštění Flow123d na různých výpočetních clusterech. Další součástí aplikace GeoMop bude modul Analysis, který umožní úlohy parametrizovat a dále je potom provádět pro různé sady hodnot. GeoMop je aktuálně ve vývoji a je možné, že se bude rozšiřovat o další funkce.

Výstupem ze softwaru Flow123d je datový soubor, který obsahuje výsledky si-



Obrázek 1: Simulátor Flow123d a pomocný software

mulace. U tohoto souboru si uživatel může vybrat požadovaný formát, podle toho, kterou aplikaci chce použít pro zpracování výsledků. Typicky uživatelé používají buď opět GMSH nebo ParaView. Existuje také celá řada jednoúčelových nástrojů, které si uživatelé často tvoří sami, nebo vznikají v rámci různých projektů.

## 1.2 Konfigurační soubory

V současné době (verze Flow123d 1.8.2), se pro specifikaci úlohy používá jeden konfigurační soubor, který obsahuje všechna potřebná data pro definici a inicializaci úlohy. Z pohledu Flow123d je úloha definovaná pomocí konkrétních objektů, které mají nastavené různé atributy na požadované hodnoty.

Vzhledem k tomu, že úlohy definují lidé, je potřeba určit nějaké společné rozhraní, pomocí kterého budou moci definovat tyto objekty a jejich obsah. Tato defi-

nice zároveň musí být strojově čitelná, aby ji simulátor Flow123d mohl zpracovat a nakonfigurovat se podle ní do správného počátečního stavu pro zahájení výpočtu.

Jelikož se pro předávání dat používají soubory, existují v principu dvě možnosti, jak předat tato data. Formát souboru může být buď binární, nebo textový. Vzhledem k tomu, že soubory mají vytvářet lidé, tak by bylo krajně nepraktické, kdyby se použil binární formát souboru.

Textová reprezentace konfiguračních souborů s sebou kromě čitelnosti přináší i další výhody. Oproti binárnímu formátu není závislá na architektuře, jelikož všechna data jsou kódována ve formě textu. Navíc díky tomu, že textový soubor umožňuje kromě přenosu samotných dat i tyto data nějakým způsobem popsat, potom se změny v interní struktuře Flow123d nemusí nutně projevit ve formátu konfiguračních souborů.

### 1.2.1 Datová struktura

Použití textového formátu konfiguračních souborů s sebou však přináší otázku, jakým způsobem tato data v textu reprezentovat. Je důležité, aby pomocí vybraného formátu bylo možné inicializovat libovolnou strukturu tříd v C++. Takové třídy obsahují atributy, které mají název (dále označován jako klíč), typ a hodnotu. Ve většině případů platí, že klíč jednoznačně implikuje typ. Potom je tedy dostačující ukládat dvojici klíč a hodnota.

Existují i situace, kdy z názvu atributu nelze jednoznačně určit jeho typ. To je způsobené použitím polymorfismu. Z klíče lze tedy odvodit pouze jakého typu musí být předek. Pokud má tento předek více potomků, pak je nutné vybraný typ explicitně uvést. Tyto situace jsou prozatím zanedbány a jsou popsány samostatně v kapitole X.

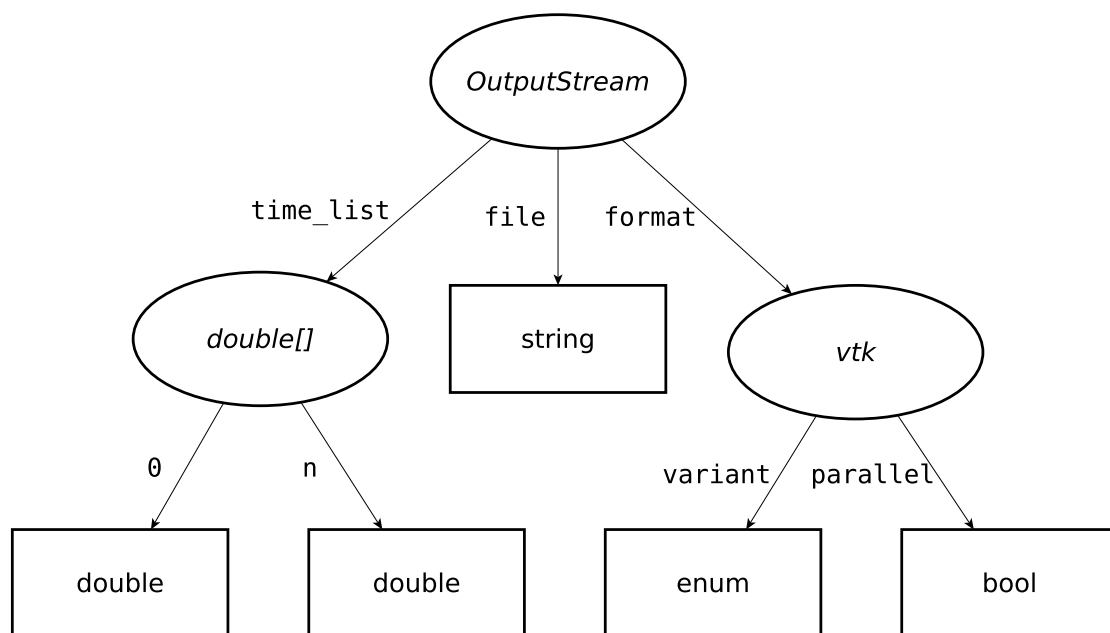
V konfiguračních souborech je tedy potřeba ukládat dvojice klíč a hodnota. Hodnotou může být buď jednoduchého nebo složeného datového typu. Reprezentace jednoduchých datových typů je většinou triviální a spočívá pouze v převodu hodnoty na textový řetězec, pokud jím není. Povolené jednoduché datové typy v rámci konfiguračních souborů jsou následující:

- booleovské hodnoty,
- celá čísla,
- desetinná čísla,
- hodnoty výčtového typu (tzv. enum),

- řetězce<sup>1</sup>.

Složeným datovým typem z pohledu použitých konfiguračních souborů může být buď homogenní pole, nebo jiný objekt. Tím pádem vzniká hierarchická datová struktura, která může mít teoreticky nekonečný počet vnořených úrovní. V praxi je samozřejmě počet úrovní vždy konečný, ale důležité je, aby použitý formát umožňoval reprezentovat libovolný počet vnoření.

Na obrázku 2 je znázorněna hierarchická struktura složeného datového typu *OutputStream*. Pro názornost jsou vynechány některé atributy. Datový typ *OutputStream* obsahuje řetězec *file*, což je jednoduchý datový typ. Dále obsahuje pole desetinných čísel *time\_list*, které dále obsahuje konkrétní desetinná čísla. Posledním znázorněným atributem objektu *OutputStream* je *format*, který obsahuje referenci na objekt typu *vtk*. Objekt tohoto typu pak dále obsahuje atributy *variant* a *parallel*, které jsou jednoduchých datových typů.



Obrázek 2: Příklad složeného datového typu s různými typy atributů

<sup>1</sup>Konkrétní implementace typu řetězec je typicky pole znaků, tedy složený datový typ. Z pohledu konfiguračních souborů se však jedná o jednoduchý datový typ, protože je dále nedělitelný.

## 1.3 Formát pro výměnu dat

### 1.3.1 Formát CON

Ve verzi Flow123d 1.8.2 se pro reprezentaci výše popsané datové struktury používá speciální formát CON, který byl navržen vývojáři Flow123d pro účel zápisu konfiguračních souborů. Jedná se o formát, který vychází z JavaScript Object Notation, který je specifikován standardem ECMA-404. Oproti tomuto standardu se liší v několika detailech.

Příklad části konfiguračního souboru ve formátu CON je znázorněn na obrázku 3. Jednou z ihned zřejmých odlišností od formátu JSON je použití znaku „=“ místo „:“ pro oddělení klíče a hodnoty. Dále není nutné psát názvy klíčů do uvozovek. Další odlišnosti a kompletní specifikaci formátu CON lze nalézt v dokumentaci k Flow123d 1.8.2.

Během používání tohoto formátu se ale jeho odlišnost od JSON projevila jako jeden z nedostatků. Kvůli nekompatibilitě formátu s formátem JSON nelze použít pro zpracování CON formátu standardní knihovny. To je jeden z důvodů, proč bylo rozhodnuto, že ve verzi Flow123d 2.0 bude použit nějaký standardní formát pro výměnu dat.

To však nebylo jediným nedostatkem tohoto formátu. Uživatelé, kteří tento soubor upravovali, naráželi často na dva problémy. Bylo pro ně velice nepohodlné neustále hlídat správné uzávorkování objektů nebo zda na konci řádku nebyla zapomenuta čárka pro oddělení položek. To představoval problém obzvlášť u rozsáhlejších konfiguračních souborů, které obsahovaly velký počet úrovní vnoření. Jelikož formát JSON sdílí tyto vlastnosti, tak byl zavržen jako možný nástupce formátu CON.

### 1.3.2 Jazyk XML

Extensible Markup Language (XML) je rozšiřitelný značkovací jazyk, který slouží pro popis dat. Tento jazyk vznikl z jazyka Standard Generalized Markup Language (SGML), z kterého je odvozen i jazyk HTML. Všechny správně zformátované XML dokumenty tedy zároveň i SGML dokumenty. Popis a doporučení týkající se jazyka XML lze nalézt na webových stránkách konsorcia W3C.

Použití jazyka XML pro zápis konfiguračních souborů bylo jednou ze zvažovaných možností. Ukázku takového zápisu lze vidět na obrázku 3, na němž si lze zároveň všimnout odlišností zápisu XML oproti jiným možnostem.

### Soubor ve formátu CON

```
1 output = {
2   output_stream = {
3     file = "./flow_test16.pvd",
4     format = {
5       TYPE = "vtk",
6       variant = "ascii"
7     },
8     name = "flow_output_stream"
9   },
10  output_fields = [ "pressure_p0",
11                   "pressure_p1",
12                   "velocity_p0" ]
13 }
```

### Soubor v jazyce XML

```
1 <output>
2   <output_stream>
3     <file>./flow_test16.pvd</file>
4     <format type="vtk">
5       <variant>ascii</variant>
6     </format>
7     <name>flow_output_stream</name>
8   </output_stream>
9   <output_fields>pressure_p0</output_fields>
10  <output_fields>pressure_p1</output_fields>
11  <output_fields>velocity_p0</output_fields>
12 </output>
```

### Soubor v jazyce YAML

```
1 output:
2   output_stream:
3     file: ./flow_test16.pvd
4     format: !vtk
5     variant: ascii
6     name: flow_output_stream
7   output_fields:
8     - pressure_p0
9     - pressure_p1
10    - velocity_p0
```

Obrázek 3: Ukázky různých formátů pro zápis konfiguračního souboru



Velkou výhodou tohoto jazyka je, že umožňuje nadefinovat si vlastní strukturu dokumentu. Lze tedy specifikovat kde se mohou vyskytnout jaké elementy, jaké mohou mít atributy a tak podobně. K tomu slouží DTD nebo XML Schema, které má oproti DTD více funkcí, např. dokáže omezit počet výskytů elementů.

Použití XML Schema by velice usnadnilo validaci datové struktury a navíc existuje celá řada nástrojů, které jsou schopné ověřit, zda je daný XML dokument validní pro dané XML Schema. Úskalím použití této validace by však byly tzv. autokonverze, které jsou popsány v kapitole ??.

Jedná se o speciální zápis, který lze použít v datové struktuře při zapisování polí nebo záznamů. V těchto speciálních případech lze místo pole či záznamu zapsat pouze hodnotu. To znemožňuje běžnou validaci pomocí XML Schema a bylo by nutné validaci XML upravit tak, aby byla schopná brát ohled na autokonverze. To znamená, že by bohužel nebylo možné použít univerzální nástroje pro validaci XML.

Nevýhodou tohoto formátu je jeho „výřečnost“. Té si lze na první pohled všimnout na obrázku 3. Jazyk XML vyžaduje z uvedených možností pro zápis stejných dat nejvíce znaků. Hlavní nevýhodu v použití jazyka XML pro zápis konfiguračních souborů vývojáři Flow123d viděli v jeho zásadní odlišnosti oproti dosavadně používanému formátu CON. Dle jejich názoru by to pro uživatele Flow123d byla příliš velká změna.

### 1.3.3 Jazyk YAML

Poslední z uvažovaných možností bylo použití jazyka YAML, který je zobecněním formátu JSON. Kterýkoliv JSON dokument je tím pádem i YAML dokumentem. Oproti formátu JSON ovšem umožňuje syntaktický zápis, který byl speciálně navržen s ohledem na to, aby ho mohli jednoduše zapisovat lidé.

Toho si lze všimnout opět na obrázku 3. Ze všech uvedených možností je zápis v jazyce YAML nejkratší, a to jak počtem napsaných řádek, tak i počtem potřebných znaků. Zároveň ovšem elegantně řeší problémy původně použitého formátu CON, resp. formátu JSON.

Jelikož pro zápis vnořených dat používá odsazení, není nadále nutné používat závorky pro uzavření záznamů a polí. Dále podporuje zápis polí pomocí odrážek, což je dobře čitelné a pohodlné pro zápis. Oproti formátu JSON také odpadá nutnost psaní čárek na koncích řádků pro oddělení klíčů v záznamů nebo položek v poli. Dále není nutné psát řetězce do uvozovek, protože se datové typy odlišují implicitně<sup>2</sup>.

Všechna tato vylepšení vedou k tomu, že soubory zapsané v jazyce YAML jsou

---

<sup>2</sup>V případě potřeby lze datový typ specifikovat i explicitně.

velice dobře čitelné a jednoduché pro zápis. Navíc se jazyk YAML od původního formátu CON neliší natolik, jako jazyk XML. To vedlo k rozhodnutí, že ve verzi Flow123d 2.0 bude použit jazyk YAML pro zápis konfiguračních souborů. Vytvářená aplikace tedy bude pracovat s konfiguračními soubory napsanými v jazyce YAML. Předchozí formát CON bude možné v rámci aplikace importovat<sup>3</sup>, ale již nebude podporován export ve formátu CON.

---

<sup>3</sup>Import formátu CON je nad rámec této diplomové práce a v rámci projektu ho řešil Ing. Pavel Richter.

## **2 Analýza**

### **2.1 Specifikace konfiguračních souborů Flow123d**

#### **2.1.1 Datové typy**

Primitivní datové typy

Pole

Záznamy

Abstraktní záznamy

#### **2.1.2 Autokonverze**

#### **2.1.3 Autokonverze na pole**

#### **2.1.4 Autokonverze na záznam**

#### **2.1.5 Transpozice**

### **2.2 Formát YAML pro konfigurační soubory**

#### **2.2.1 Serializace a deserializace**

#### **2.2.2 Zápis datových typů**

## **3 Návrh**

### **3.1 Autokonverze**

### **3.2 Validace**

### **3.3 Kontextová dokumentace**

### **3.4 Automatické doplňování textu**

### **3.5 Funkce editoru**

### **3.6 GUI**

## **4 Implementace a testování**

## **5 Uživatelská příručka**

### **5.1 Zápis konfiguračních souborů pomocí YAML**

### **5.2 Vytváření a editace konfiguračních souborů**

### **5.3 Validace a odstranění chyb**

### **5.4 Nastavení**

## Závěr

V rámci této diplomové práce byl vytvořen editor konfiguračních souborů pro Flow123d. Jedná se o samostatně funkční aplikaci, která je ovšem navržena s ohledem na její použití jako součást softwarového balíku GeoMop, který obsahuje další nástroje, které usnadňují práci uživatelům Flow123d.

Editor uživatelům zjednodušuje vytváření a upravování konfiguračních souborů. Umožňuje ověřit správnost zadané konfigurace pro zvolenou verzi Flow123d a případně uživatele upozornit na detekované chyby. Tato funkce uživateli přináší časovou úsporu a uživatelsky příjemnější rozhraní při odhalování chyb.

Editor dále uživatelům poskytuje kontextovou dokumentaci a našeptávač. Obě tyto funkce přizpůsobují svůj obsah na základě pozice kurzoru v textu, tedy oblasti, kterou uživatel zrovna upravuje. Pro uživatele to představuje značné zjednodušení, jelikož může využít tyto funkce místo prohledávání rozsáhlé dokumentace.

V neposlední řadě editor obsahuje komponentu pro grafické znázornění datové struktury, která poskytuje alternativní pohled na zadaná data, a umožňuje rychlejší orientaci v rozsáhlých konfiguračních souborech. Kromě těchto stěžejních funkcí editor poskytuje i běžné nástroje pro manipulaci s textem, jako jsou například operace se schránkou, možnost vrácení provedených změn, vyhledávání a nahrazení nebo změna úrovně odsazení.

Aplikace je multiplatformní a podporuje systémy Windows (XP nebo novější) a Linux. S ohledem na požadavek multiplatformní aplikace byl pro vývoj použit jazyk Python 3 a grafická knihovna PyQt 5. K aplikaci byly vytvořeny instalační balíčky pro Windows a Debian.

V rámci budoucího vývoje jsou plánovány další dodatečné funkce. Jedná se např. o zlepšení zvýraznění syntaxe, které se by se mohlo přizpůsobit přímo formátu Flow123d. Další možné vylepšení spočívá v rozšíření funkcionality komponenty pro vizualizaci datové struktury. Ta by mohla v budoucnu podporovat kromě zobrazení i editaci dat nebo vylepšené zobrazení speciálních datových typů.

## Literatura

- [1] BŘEZINA, Jan et al. *Flow123d version 1.8.2: Documentation of file formats and brief user manual*. [online]. Liberec, 2015 [cit. 2016-02-24]. Dostupné z: [http://flow.nti.tul.cz/packages/1.8.2\\_release/flow123d\\_1.8.2\\_doc.pdf](http://flow.nti.tul.cz/packages/1.8.2_release/flow123d_1.8.2_doc.pdf)