



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

Validace konfiguračních souborů Flow123d

Magisterský projekt

M14000168

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Tomáš Křížek**

Vedoucí práce: Jiří Vraný, Ph.D.





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

Validation of Flow123d configuration files

Project report

M14000168

Study programme: N2612 – Electrical Engineering and Informatics

Study branch: 1802T007 – Information Technologies

Author: **Bc. Tomáš Křížek**

Supervisor: Jiří Vraný, Ph.D.



Tento list nahrad'te
originálem zadání.

Prohlášení

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

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasa- huje do mých autorských práv užitím mého magisterského projektu pro vnitřní potřebu TUL.

Užiji-li magisterský projekt nebo poskytnu-li licenci k jeho 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.

Magisterský projekt jsem vypracoval samostatně s použitím uve- dené literatury a na základě konzultací s vedoucím mého magis- terského projektu a konzultantem.

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

Datum:

Podpis:

Abstrakt

Tento projekt se zabývá problematikou validace konfiguračních souborů Flow123d, jejichž struktura je dynamicky definována v závislosti na verzi Flow123D. V projektu jsou popsány specifiky formátu konfiguračních souborů, jako jsou reference nebo automatické konverze. Dále je navržena datová struktura pro zpracování konfiguračních souborů a dynamicky definovaných validačních pravidel. Nakonec je rozebrána problematika samotné validace.

Abstract

This project addresses the issues surrounding validation of Flow123d configuration files. Data structure of these files is dependent on the version of Flow123d. This paper describes the specifics of the configuration file format. These include references or automatic conversions. Next, a data structure for processing of configuration files and dynamically defined validation rules is designed. Lastly, the issues of validation are discussed.

Obsah

Seznam zkratek	7
1 Úvod	8
2 Formát konfiguračních souborů	9
2.1 Autokonverze	10
2.2 Reference	10
3 Struktura konfiguračních souborů	11
3.1 Skalární hodnoty	11
3.2 Pole	12
3.3 Záznam	12
3.4 Abstraktní záznam	12

Seznam zkratek

JSON JavaScript Object Notation

1 Úvod

V mém magisterském projektu jsem řešil problematiku validace konfiguračních souborů pro Flow123d. Tyto soubory mají dynamicky definovanou strukturu, která se liší podle verze Flow123d. Cílem je vytvořit nástroj, který určí, zda zadaný konfigurační soubor je validní pro danou verzi.

Konfigurační soubory jsou ve formátu CON, který se podobá standardnímu formátu JSON. Od tohoto formátu se mírně odlišuje syntaxí. Konkrétní odlišnosti jsou popsány v kapitole ???. Kromě odlišné syntaxe také umožňuje vytvářet reference, se kterými je potřeba náležitě pracovat při zpracování dat. Práce s referencemi jsou popsány v kapitole ??. Formát CON také umožňuje v některých případech použít zkrácený zápis, který se poté čtečkou formátu automaticky zkonvertuje na požadovaný tvar. Problematikou autokonverzí se zabývá kapitola ??.

Jelikož se struktura konfiguračních souborů se liší podle verze Flow123d, validace nespočívá pouze v ověření dat podle pevně dané struktury. Pro validaci je nutné dynamicky načíst sadu pravidel, která strukturu popisuje. Sada těchto pravidel je dodána ve formátu JSON. V něm se nachází specifikace jednotlivých pravidel, které společně tvoří stromovou strukturu, stejně jako vstupní konfigurační soubor. Problematika použití sady pravidel popisující strukturu je popsána v kapitole ??.

2 Formát konfiguračních souborů

Konfigurační soubory, které je potřeba zvalidovat, jsou často psány ručně a nejsou strojově generovány. Formát CON, ve kterém jsou napsány, byl tomuto přizpůsoben. Syntaxe CON je podobná standardnímu formátu JSON, ale pro snazší zápis se mírně liší. Rozdíly v syntaxi oproti formátu JSON jsou následující.

- Klíče neobsahující mezeru mohou být napsány bez uvozovek.
- Klíč může být od hodnoty oddělen znakem „=“ místo „:“.
- V souboru se mohou vyskytovat komentáře. Jsou povoleny víceřádkové komentáře uzavřené mezi sekvencemi `/*` a `*/`, nebo řádkové komentáře, které jsou uvozeny sekvencí `//`.

Formát CON byl navržen pro inicializaci datových typů v C++. Kromě syntaktických rozdílů tedy předpokládá následující sémantická pravidla.

- Záznam má určitý typ, který definuje seznam povolených klíčů.
- Každý klíč má daný typ, jméno a implicitní hodnotu.
- Klíče napsané velkými písmeny mají speciální význam a jsou zpracovány čtečkou CON formátu. Názvy ostatních klíčů jsou malými písmeny.
- Povolené datové typy jsou skalární hodnoty, pole nebo záznamy.
- Pole jsou vždy tvořeny prvky stejného typu.
- U záznamů lze použít polymorfismus, kde klíč `TYPE` udává typ záznamu.
- Lze použít reference na data v rámci CON souboru pomocí speciálního klíče `REF`.

Bližší specifikace formátu CON lze nalézt v dokumentaci Flow123D.

2.1 Autokonverze

2.2 Reference

3 Struktura konfiguračních souborů

Jak již bylo zmíněno v kapitole 2, data v konfiguračním souboru podléhají určitým sémantickým pravidlům. Uvedená pravidla jsou pouze obecná a jejich konkrétní podoba je závislá na verzi Flow123D. Požadavky na strukturu jsou popsány v dokumentaci Flow123D a pro potřeby aplikace je jejich popis dodán ve strojově čitelném formátu JSON.

Tento popis obsahuje pravidla, která slouží pro validaci jednotlivých položek v konfiguračních souborech. Pravidla mohou definovat několik datových typů. Prvními z nich jsou skalární hodnoty. Jejich typy a pravidla pro validaci jsou popsány v tabulce 1. Dalším datovými typy jsou pole (*Array*), záznam (*Record*) a abstraktní záznam (*AbstractRecord*), který se využívá pro polymorfismus.

Tabulka 1: Datové typy skalárních hodnot

Datový typ	Popis	Pravidlo pro validaci
<i>Integer</i>	celé číslo	hodnota musí být v rozmezí min a max
<i>Double</i>	desetinné číslo	hodnota musí být v rozmezí min a max
<i>Bool</i>	přepínač	–
<i>String</i>	řetězec	–
<i>Selection</i>	výběr z množiny	musí obsahovat hodnotu z množiny povolených hodnot
<i>FileName</i>	cesta k souboru	–

U skalárních typů spočívá validace v ověření správného datového typu a ověření validačního pravidla, pokud nějaké existuje. Pro číselné hodnoty se kontroluje rozmezí, které je určeno hodnotami **min** a **max** u specifikace datového typu. Speciální validaci vyžaduje také typ *Selection*. U tohoto datového typu je dána množina povolených hodnot (**values**), kde každá hodnota má jméno (**name**), které ji jednoznačně identifikuje. Zadaná hodnota v *Selection* je validní, pokud se shoduje se jménem některé z povolených hodnot.

3.1 Pole

3.2 Záznam

3.3 Abstraktní záznam