Posudek *oponenta* diplomové práce

Jméno a příjmení autora posudku: Jakub Klímek

Jméno a příjmení autora práce: Roman Betík

Název práce: Automatic Generation of Synthetic XML Documents

Práce se zabývá automatickým generováním XML dokumentů na na základě XML schématu ve formátu DTD a XSD a pak také v situaci kdy schéma k dispozici není. Práce obsahuje dostatečný přehled a srovnání existujících technik, bohužel přehledové tabulky srovnání na stranách 29 a 30 jsou chybně vytištěny a části jich chybí.

Kapitola 3 obsahuje popis toho, co by navrhovaný algoritmus XmlGenerator pro generování XML dokumentů měl dělat. Kapitola je také protkána vymezeními se vůči existujícímu generátoru ToXgene, který dle autora většinou nabízí lepší možnosti generování dat za cenu složitějšího nastavení. Ve světle tohoto srovnání mi nepřijde jako vhodná volba implementovat vlastní generátor, o kterém autor dopředu říká, že bude umět méně a ospravedlňuje to tím, že půjde snáze konfigurovat. Lepší cesta by v tomto případě byla vyvinout nástroj usnadňující konfiguraci ToXgene.

V kapitole 4 je popisován XML konfigurační soubor a je zde popsán způsob, jakým lze do XmlGeneratoru přidat vlastní generátor distribuce náhody a generátor jmen a hodnot elementů a atributů. Dále kapitola pokračuje sekcí s pseudokódy algoritmů generujících XML dokumenty, které vesměs používají rekurzivních volání pro generování XML stromu s hloubkou a šíří danou parametrizovatelným náhodným generátorem. Navrhovaný algoritmus ovšem nepřináší nic, co by již nebylo implementováno v ostatních generátorech. Demonstrační aplikace běží na příkazové řádce a jako vstup bere popsaný konfigurační soubor. Autor implementoval okno s tlačítkem, které tuto aplikaci spouští, viz. screenshoty na straně 62, což působí zcela nedostatečně, obzvláště vzhledem k deklarovanému záměru navrhnout systém, který bude snadno a rychle použitelný. Přitom by stačil alespoň předvyplněný grafický formulář pro konfigurační soubor a poté alespoň přehled statistik o vygenerovaných souborech.

V kapitole 5 autor srovnává XmlGenerátor s ostatními generátory tím, že se snaží XmlGenerátorem emulovat ostatní generátory, nicméně proč zvolil zrovna tento způsob srovnání v práci popsáno není. Srovnání končí opět špatně vytištěnou tabulkou na straně 85.

Ačkoliv zadání práce obsahuje implementaci konstrukcí IDs, IDREFs, konfiguraci distribuce jednotlivých konstruktů a nastavení jejich komplexity, řešení vzájemného ovlivňování parametrů, v implementaci ani v algoritmu toto řešeno není a je pouze konstatováno, že toto algoritmus neumí, bez dalšího vysvětlení. Dále zadání obsahuje možnost konfigurační parametry extrahovat ze sady XML dokumentů a XML dotazů, nicméně tomu se autor v práci také nevěnuje. Nakonec je v zadání, že by generátor měl být rychlý, ale o jeho výkonu se v práci také nepíše.

Zadání práce splněno nebylo, jazyková úroveň práce je nižší, práce nepřináší žádné nové výsledky a implementace nesplňuje cíl snadné použitelnosti.

Otázka k obhajobě:

1) Proč autor neimplementoval to, co měl v zadání, tedy například konstrukce IDs a IDREFs v DTD (případně key, keyref v XSD)?

Doporučení k obhajobě:

Z výše uvedených důvodů práci *nedoporučuji* k obhajobě.

Vynikající práce vhodná pro soutěž studentských prací	ANO ☐

V Praze dne: 15.5.2015

Podpis: