

DIDAKTICKÉ DEMONSTRACE MODELŮ PRO POPIS FORMÁLNÍCH JAZYKŮ

Z. Křivka (student DSP, 2. r.), **R. Lukáš** (DSP, 3. r.), **A. Meduna** (akademický pracovník)
ÚIFS, Fakulta informačních technologií, VUT v Brně, E-mail: krivka@fit.vutbr.cz

1 ÚVOD

V současné době se na naší fakultě vyučují dva předměty se zaměřením na formální jazyky a teorii překladačů. V bakalářském studijním programu je to velmi obsažný a tedy i náročný předmět *Formální jazyky a překladače* [1] a v magisterském pak *Výstavba překladačů*. Relativně malý počet přednášek (13) a demonstračních cvičení (6) nám nedovoluje se věnovat dostatečně velkému množství příkladů nutných pro kvalitnější pochopení vykládané látky.

Proto se tento řešitelský tým rozhodl implementovat didakticky účelný nástroj pro názornou demonstraci probíraných modelů a usnadnění studia těžkých pasáží učiva.

2 SPECIFIKACE A ANALÝZA POŽADAVKŮ

Prvotním požadavkem je vytvoření aplikace (**DI**daktické **DE**monstrace **FO**rmálních **M**odelů, DIDEFOM) na platformě Microsoft Windows® pro pedagogicky kvalitní prezentaci funkčnosti formálních modelů a algoritmů pro transformace mezi nimi. Toho bude dosaženo pomocí multimediálních prostředků jako je (a) barevná animace grafů, kterými dané modely reprezentujeme; (b) textový komentář a odkazy na literaturu; případně i (c) zvukový záznam popisující simulovaný děj. Pro úplnost je třeba k programovému dílu vytvořit uživatelskou dokumentaci a vhodný demonstrační soubor příkladů. Vše bude publikováno na webových stránkách projektu [3] ke konci roku 2006 a následně průběžně aktualizováno.

Analýzou specifikovaného problému jsme dospěli k následující klasifikaci demonstrací:

- *Formální model* (matematická struktura definující jazyk resp. třídu jazyků): (1) automat (přijímá věty) nebo (2) gramatika (generuje věty jazyka).
- *Transformace* mezi dvěma formálními modely (například převod nedeterministického konečného automatu na deterministický)
- *Konstrukce řídicího jádra syntaktického analyzátoru* z daného formálního modelu (například konstrukce LL(1) tabulky z LL gramatiky)

Z pohledu uživatele bude aplikace pracovat ve dvou hlavních módech:

- § *Editační* – pro definování a úpravu instancí formálních modelů, které chceme simulovat, případně transformovat (samotné transformace bude možné pouze parametrizovat, nikoli uživatelsky vytvářet a editovat).
- § *Simulační* – demonstrační mód, který bude pracovat buď *interaktivně* (kroky ovlivňuje uživatel), nebo *automaticky* (neinteraktivní simulace zadanou rychlostí); mezi oběma

módy bude možno přecházet. U simulace je nutno také mít možnost určovat granularitu jednotlivých kroků (míra názornosti), jejich sdružování do skupin (hierarchické sdružování pro podporu nastavení granularity) a případného uložení (načtení) rozpracované simulace.

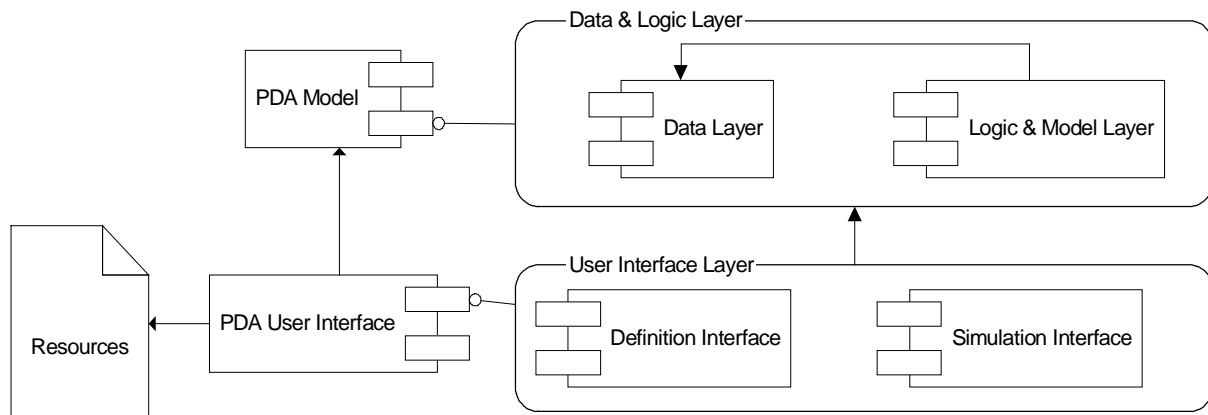
3 REALIZACE

Hlavní částí realizace je implementace popisovaného nástroje s přívětivým uživatelským rozhraním a s přiměřenými možnostmi konfigurace a parametrizace.

3.1 NÁVRH

Projekt je navrhován pro implementaci v prostředí Microsoft .NET, konkrétně jazykem C# ([3]). Výsledkem bude komplexní nativní aplikace pro operační systémy Windows (s podporou platformy .NET) využívající soubor knihoven WinForms a GDI+ pro pohodlné a plně objektově orientované používání Windows API.

Aplikace bude umožňovat práci pouze s jedním modelem v jedné instanci aplikace. Z pohledu starší knihovny MFC se jedná o tzv. SDI aplikaci. Architektura je pseudo-trojvrstvá, kdy ve výhodných případech porušíme zapouzdření mezi vrstvou datovou a logickou (např. načítání nastavení v prezentační vrstvě pomocí datové vrstvy, viz obr. 1).

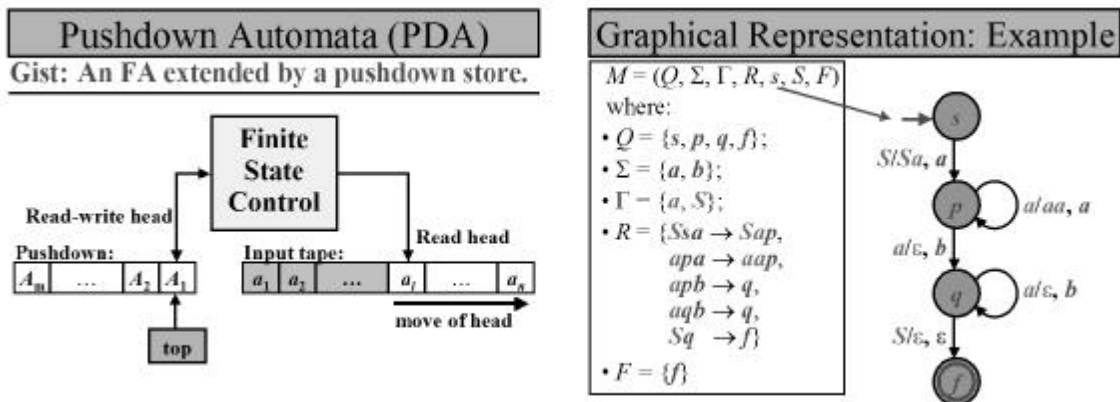


Obr. 1: Architektura aplikace DIDEFOM (PDA značí zásobníkový automat)

Primárně využívaným datovým formátem pro import/export bude masově rozšířené XML. Při použití na popisy grafů reprezentujících formální modely využijeme některé ze standardizovaných XML Schémat (např. *GraphML*, viz [4]). Zde budeme těžit z bohaté výbavy knihoven .NET pro práci s tímto přísně strukturovaným formátem. Díky využití XML (lze vytvářet a upravovat i manuálně) můžeme odložit problém implementace uživatelsky přívětivého editoru instancí formálních modelů do pozdější (méně kritické) fáze projektu.

3.2 UŽIVATELSKÉ ROZHRANÍ

Při tvorbě uživatelského rozhraní se částečně vychází z přednáškových materiálů výše zmíněných předmětů (viz obr. 2). Zobrazování více druhů pohledů nad jedním modelem zajišťuje návrhový vzor Pozorovatel (*Observer* nebo též *Document-View*, [5]). Na kreslení grafů demonstrovovaných modelů využijeme volně dostupné nástroje a knihovny ([6], [7]).



Obr. 2: Ukázka grafické reprezentace zásobníkového automatu v předmětech IFJ a VYP

4 ZÁVĚR

Velká pozornost je při návrhu kladena na možnosti rozšiřitelnosti. V rámci samotného projektu nelze realizovat demonstrace všech formálních modelů a transformací, které jsou v předmětech o teoretické informatice probírány. To náš řešitelský tým vede k myšlence vytvořit dostatečně kvalitní programové prostředí, které bude možno rozšiřovat nejen o samotné příklady, ale i o nové formální modely a algoritmy (v rámci bakalářských projektů; např. nový model zásobníkového automatu (PDA) na obr. 1). V rámci projektu FRVŠ hodláme implementovat všechny v tomto příspěvků zmíněné formální modely a transformace.

V této fázi projektu využijeme dobré možnosti programové reflexe samotného kódu i modulů (tzv. *assembly*) v jazyce C#, které jsou potřebné pro připojení nových modulů, jejichž návrh a struktura nejsou v době vzniku jádra aplikace známy.

Poděkování: Tento příspěvek vznikl za podpory projektu FRVŠ MŠMT FR1909/2006/G1.

LITERATURA

- [1] Formální jazyky a překladače, [stránky předmětu], dostupné na <http://www.fit.vutbr.cz/study/course-1.php.cs?id=23> (květen 2006).
- [2] Didaktické demonstrace modelů pro popis formálních jazyků, [stránky projektu], dostupné na <http://www.fit.vutbr.cz/~krivka/didefom> (květen 2006).
- [3] Robinson, S. a kol., *C# Programujeme profesionálně*, Brno, Computer Press 2003, ISBN 80-251-0085-5.
- [4] The GraphML File Format, [stránka projektu], dostupné na <http://graphml.graphdrawing.org/> (květen 2006)
- [5] Cooper, J., W., *C# Design Patterns: A Tutorial*, Addison-Wesley 2003, ISBN 0-201-84453-2.
- [6] Lew, C., *Building a Simple Diagramming Tool with Windows Forms*. Visual Studio .NET Developer, Issue June 2005, Pinnacle Publishing, dostupné na <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvsdev05/html/vs05f1.asp> (květen 2006).
- [7] The Netron Project, 2002-2006, [stránky projektu], dostupné na <http://netron.sourceforge.net/wp> (květen 2006).