CASE UML .FRI

Ján Janech[1] , Ľubomír Sadloň[2]

[1]Katedra informatiky, FRI, Žilinská Univerzita Žilina,
010 26, Žilina
Jan.Janech@fri.uniza.sk

[2]Katedra softvérových technológií, FRI, Žilinská Univerzita Žilina,
010 26, Žilina
Lubomir.Sadlon@fri.uniza.sk

Abstrakt. Príspevok obsahuje popis UML/DSM CASE nástroja UML .FRI. Nástroj je vyvíjaný v rámci projektovej výučby študentmi inžinierskeho stupňa štúdia v študijného programu Informačné systémy na Fakulte Riadenia a Informatiky Žilinskej Univerzity v Žiline. CASE je vyvýjaný špeciálne s ohľadom na potreby univerzitného prostredia. Nástroj je tvorený v jazyku Python s použitím GUI knižnice GTK+, čo umožňuje jeho využitie na rôznych hardvérových i softvérových platformách. Nástroj sa snaží abstrahovať od použitého metamodelu a je ho teda možné použiť i na tvorbu ľubovoľných iných modelov ako UML.

Kľúčové slová: CASE, UML, DSM, python

1 Úvod

Tak ako sa objektové programovanie pomaly stáva v praxi nutnosťou, začína sa čoraz viac používať i počas výučby na vysokých školách. S tým prichádza aj nutnosť využívať pokročilé *CASE* nástroje, ktoré by študentom uľahčili objektovú analýzu a návrh pri práci na školských projektoch. Vizuálna reprezentácia systému pomocou UML diagramov takisto uľahčí prácu učiteľom pri hodnotení.

Pred dvoma rokmi bola situácia na Fakulte riadenia a informatiky ŽU ohľadom *UML CASE* nástrojov dosť zlá. Drvivá väčšina predmetov žiadny *CASE* pri výuke nepoužívala. Iniciatívnejší študenti sťahovali z internetu časovo obmedzené trial verzie komerčných nástrojov, alebo použili niektorý z freeware-ových, ktoré ale poväčšine nemali potrebnú funkčnosť. Na zvyšných predmetoch používali na každom iný *CASE*.

Riešení bolo niekoľko:

- *Zakúpiť licencie na niektorý s komerčných nástrojov.* Problémom pri tomto riešení boli – financie. Licencie na akýkoľvek softvér do všetkých učební na fakulte (+pre učiteľov) sú dosť veľká položka. Okrem toho sa nerieši situácia pre študentov, ktorý by si museli zaobsatarať legálnu kópiu tohoto nástroja sami (napr. Trial verzie).
  Zvažované možnosti:

Sparx Enterprise Architect
IBM Rational Rose UML
Select Component Factory

- *Vybrať s pomedzi CASE nástrojov, ktoré sú poskytované zdarma.* Týmto by sa vyriešila otázka financií ako pre fakultu, tak i pre študentov. Problém by nastal pri zvýšených požiadavkách, ktoré by vybraný (a v horšom prípade už používaný) nástroj nedokázal splniť.
  Možnosti:
  Umbrello
  ArgoUML
- *Vytvoriť vlastný nástroj*. Táto možnosť vyhovuje ako po finančnej stránke tak i po funkcionálnej stránke. Pridať akúkoľvek novú funkcionalitu by principiálne nebol problém. Vytvorenie takéhoto nástroja však vyžaduje množstvo času.

Rozhodnutie bolo: skúsiť to s treťou možnosťou. Bol vytvorený projekt v rámci projektovej výučby, ktorý mal za cieľ nástroj vytvoriť a v budúcnosti udržiavať *CASE* nástroj na tvorbu *UML*. Jeho názov je *UML .FRI*.

Nástroj je vyvýjaný generačne. Každý rok pribúdajú nový členovia týmu, ktorý spolupracujú na projekte nasledujúce 3 semestre (jeden a pol roka). Tým je zabezpečené, že predchádzajúca generácia môže odovzdať projekt (a myšlienky) tej nasledujúcej. Momentálne je nástroj vo vývoji 2 roky a od budúceho semestra sa začne používať na niektorých predmetoch.

2 Technológie, na ktorých stojí UML .FRI

CASE je naprogramovaný v jazyku *Python*. Je to plne objektový dynamicky typovaný programovací jazyk. Neznamená to však, že by vyžadoval od programátora čisté objektové programovanie. Je to multiparadigmový jazyk. Podporuje štrukturované, procedurálne, (čiastočne) funkcionálne, modulárne a objektové programovanie. Tým ponecháva programátorovi slobodu rozhodnúť sa, ktorý štýl zápisu mu vyhovuje najviac. Väčšinou sa využíva kombinácia všetkých týchto prístupov. *Python* je jazyk multiplatformový, vďaka čomu je možné využívať *CASE* nástroj *UML .FRI* na rôznych operačných systémoch aj na rôznych hardvérových platformách.

Na tvorbu *GUI* bola použitá knižnica *GTK+*. Rovnako ako jazyk *Python* dokáže *GTK+* fungovať na rôznych platformách. Je to však za cenu zníženej rýchlosti vykreslovania *GUI*.

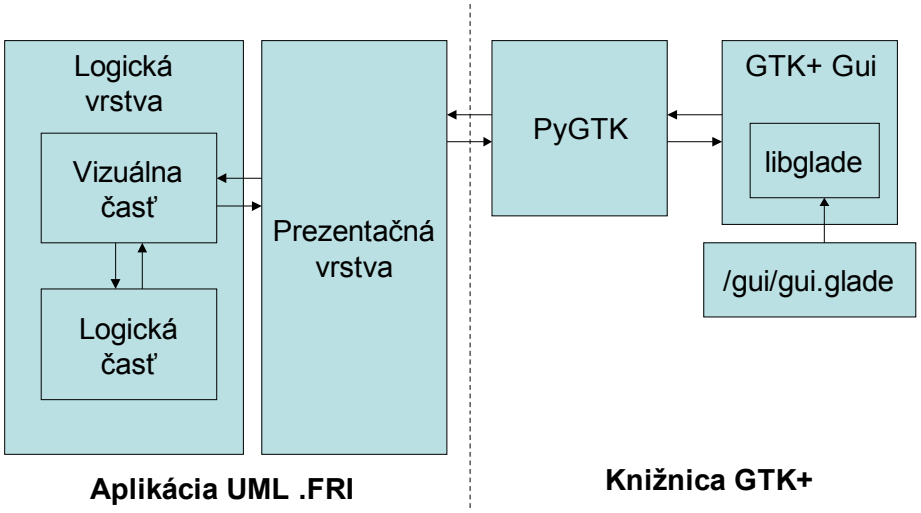
Všetky konfiguračné súbory sú uložené pomocou značkovacieho jazyka *XML*. Na jeho parsovanie sa používa knižnica *minidom*. Jedná sa o jednoduchú implementáciu objektového modelu *DOM*. Vďaka tejto kombinácii (*XML+DOM*) je možné jednoduché načítanie a ukladanie štrukturovaných dát do súboru a aj ich jednoduchá úprava človekom. Knižnica *minidom* je priamo súčasťou základného distribučného

balíčka jazyka python, takže nie je nutné doinštalovávať žiadne ďalšie balíky do systému.

Výstupné súbory obsahujúce modely sú (po vzore *Open Document Format* a *OpenXML*) uložené ako *XML* súbory komprimované do *ZIP* archívu. Umožňuje to pridávať dodatočné informácie (ako náhľad, ikonu, metadáta a podobne) priamo do súboru. Tieto informácie sú potom prístupné aj pre iné programy, ako napríklad vyhľadávače.

Program striktne dodržiava n-vrstvovú architektúru (obrázok 1), čo sprehľadňuje kód a zjednodušuje zásahy. Ako je vidieť, logická vrstva sa delí na dve nezávislé časti:

- *Logická časť* – implementuje operácie nad modelom a metamodelom, ako je vytváranie diagramov, vytváranie elementov, zmena ich vlastností, načítanie/ ukladanie modelov do súboru a podobne...
- *Vizuálna časť* – vykreslovanie diagramov, označovanie elementov na diagrame, práca zo schránkou...



Obr. 1. Architektúra nástroja UML .FRI

UML .FRI je postavený na „samonosnej“ objektovej štruktúre, teda sa jedná z väčšej časti o objektový prístup. Na zjednodušenie a sprehľadnenie kódu sa však využívajú aj funkcionálne zápisy ako napríklad *list comprehension* (generátory zoznamov).

Na návrh nástroja bol použitý program *Enterprise Architect*. Vo fáze analýzy boli vytvorené diagramy prípadov použitia, diagramy aktivít a sekvenčné diagramy. Pomocou programu bola pokrytá aj fáza návrhu, za pomoci diagramov tried. V budúcnosti sa model prepíše do *UML .FRI*, aby sa nástroj stal úplne nezávislý od iných produktov.

2.1 UML metamodel

Projekt bol od začiatku navrhovaný tak, aby sa oddelila programová výkonná časť, ktorá by bola upravovaná tímom okolo nástroja *UML .FRI*, od metamodelu *UML*. Tým nástroj dosahuje maximálnu flexibilitu. V prípade chýbajucej implementácie niektorej časti normy *UML*, poprípade po zmene normy, môže upraviť metamodel ktokoľvek bez zásahu do programu.

To bola pôvodná myšlienka tohoto oddelenia. Týmto oddelením vznikol vedľajší efekt, že sa nástroj priblížil nástrojom *DSM* (*Domain Specific Modeller*). Filozofia *DSM CASE* vychádza z toho, že nieje možné vytvoriť modelovací jazyk, ktorý by vyhovoval každému a bol by vhodný (resp. použiteľný) na každý typ úlohy. Preto dávajú tieto nástroje užívateľovi možnosť vytvoriť si vlastný metamodel a teda navrhnúť vlastný modelovací jazyk. Tento sa potom dá použiť na riešenie špecifického problému, alebo typu problémov. Okrem toho nie je problém použiť jeden nástroj na editáciu viacerých typov diagramov. Napríklad *UML* a *E-R*.

Metamodel je v *UML .FRI* zapisovaný prostredníctvom systému *XML* súborov. Každý súbor reprezentuje jeden typ objektov použiteľných v modeli (elementov/spojení/diagramov).

Príklad XML súboru s elementom (poznámka v UML):

```
<?xml version="1.0" encoding="utf-8"?>
<ElementType id="Note">
    <Icon path="icons/note.png" />
    <Connections>
        <Item value="Note Link" with="*" />
    </Connections>
    <Attributes>
        <Item value="Name" type="str" propid="name" />
        <Item value="Notes" type="note" propid="note" />
    </Attributes>
    <Appearance>
        <Shadow padding="3" color=" #505050">
            <Rectangle fill=" lightyellow" border=" black"
righttop="10 lightyellow note">
                <Sizer minwidth="50">
                    <Padding padding="5">
                        <TextBox text="#note" font=" Arial 10"
color=" black" />
                    </Padding>
                </Sizer>
            </Rectangle>
        </Shadow>
    </Appearance>
</ElementType>
```

2.2 Generovanie kódu a diagramov

V rámci diplomových prác v školskom roku 2006/2007 riešili Ing. Miroslav Špigura a Ing. Pavol Kovalík *generovanie kódu* a *reverzné inžinierstvo* pre nástroj *UML .FRI*. V rámci diplomovej práce je však z časových dôvodov nemožné dokončiť také komplexné projekty. Výsledky ich práce (aj keď sú s časti funkčné) preto nie sú zatial zahrnuté do hlavnej vývojovej vetvy programu.

Tak ako modelovanie diagramov, aj generovanie kódu/diagramov je plne nastaviteľné pomocou šablón. Vďaka tomu je možné generovať kód, alebo diagramy z kódu pri použití ľubovoľného metamodelu. V súčastnosti sú k dispozícii šablóny na generovanie dokumentácie z *UML* a generovanie kódu/diagramov pre jazyky *C++*, *Delphi* a *Python*.

3 Záver

Projekt je uvolnený ako *Open Source* pod licenciou *GNU GPL v2*. Z toho vyplívajú dve výhody:

- *Program je možné použiť aj mimo fakulty FRI ŽU*. Ak by sa podarilo nástroj rozšíriť aj mimo univerzitného prostredia, Urýchlilo by sa testovanie a nahlasovanie prípadných chýb.
- *Do projektu sa môžu priamo zapojiť ľudia mimo fakulty*, čo urýchli vývoj nástroja.

Účastníci projektovej výučby na fakulte však zostávajú ako hlavný členovia tímu a teda majú právo určovať smer vývoja. Nejedná sa teda o plne otvorený model vývoja.

Referencie

1. Python Software Foundation. http://www.python.org/.
2. DSM Forum. http://www.dsmforum.org/.