

TOOL FOR SOFTWARE SYSTEMS DESIGN

Karel Hala

Bachelor Degree Programme (4), FIT BUT

E-mail: xhalak00@stud.fit.vutbr.cz

Supervised by: Kočí Radek

E-mail: koci@fit.vutbr.cz

Abstract: The main idea of this tool is to combine graphical expression of programming. And make workflow of developing more abstract. This is achieved by combination of 3 parts of this tool. Use case [1], for graphical expression of each action. Class diagram [2] for displaying further implementation. And Objected oriented petri nets [3], for describing behavior of each class.

Keywords: workflow, system design

1. ÚVOD

Moje práce se zabývá vývojem aplikace Nástroj pro podporu vývoje softwarových systémů. Hlavním účelem tohoto nástroje je zjednodušit práci programátora, zpřehlednit jeho návrh, odhalit problémy ještě předtím, než se k nim dostane a také pochopit zákaznickovy myšlenky.

Při implementaci jakékoliv aplikace či systému se často naráží na problémy nepochopení zákazníka a programátora. Tomuto lze předcházet použitím tohoto nástroje a to zejména v části diagramu užití (Use case). Dále se často stává osudným špatný návrh programu a v neposlední řadě samotná implementace takového návrhu. Díky této aplikaci dojde k propojení diagramu užití (pro porozumění daného problému), diagramu tříd (pro vytvoření návrhu) a objektově orientovaný Petriho sítí (pro popis jednotlivých tříd).

2. SEZNÁMENÍ S NÁSTROJEM

Nástroj dokáže editovat a zobrazit diagramy užití pro lepší komunikaci mezi zákazníkem a programátorem. Tato část by měla být jednoduchá a intuitivní, aby s ní zákazník mohl rychle a snadno pracovat. Dále nástroj umožňuje propojení takto vytvořených diagramů s dalšími částmi aplikace.

V diagramu tříd se nachází editace a vytvoření tříd. Některé třídy jsou vytvořeny automaticky na základě diagramu užití a jiné si zhotoví uživatel sám. Třídy, jejich atributy a propojení mezi třídami lze plně editovat.

Takto vytvořený diagram tříd je možné znázornit pomocí objektově orientovaných Petriho sítí, ve kterých lze důkladně vidět volání mezi třídami a jejich chování.

3. POPIS JEDNOTLIVÝCH ČÁSTÍ

Nejdůležitějším prvkem tohoto nástroje je průběžné automatické generování uživatelských změn v daném systému. Například změna přidání nového případu užití vede k promítnutí do diagramu tříd a to vytvořením adekvátní třídy. Při odebrání objektu dá aplikace uživateli jasně graficky najevo, že něco není v pořádku.

3.1. DIAGRAM UŽITÍ

Diagram užití slouží jako nástroj pro komunikaci programátora a zákazníka, konkrétně pro lepší porozumění zákaznickovým potřebám a nápadům.

3.2. DIAGRAM TŘÍD

Tato část nástroje je učena k vytvoření návrhu implementace, rozvrhnutí tříd, jejich atributů a vzájemnému chování a komunikaci tříd mezi sebou.

3.3. OBJEKTOVĚ ORIENTOVANÉ PETRIHO SÍTĚ

Pro popis jednotlivých tříd a jejich metod je v tomto nástroji použito objektivě orientovaných Petriho sítí. Tyto sítě znázorňují komunikaci v objektu, volání jeho metod, vytváření objektů a volání mezi dalšími třídami. Tato část je nejdůležitější pro pozdější implementaci daného systému a znázorňuje programátorovi průběh celého systému a jeho chování.

4. STAV A PRŮBĚH IMPLEMENTACE NÁSTROJE

Hlavní přednost aplikace je intuitivnost a jednoduchost. Nejsou zde žádné zbytečné editační možnosti, ani další prvky, které by vedly pouze k zneprůhlednění celého nástroje. Hlavní částí nástroje je editor a pár okolních prvků, které budou snadno použitelné.

4.1. GRAFICKÉ ROZDĚLENÍ NÁSTROJE

Aby byl nástroj jednoduchý, obsahuje na hlavním okně pouze pár ovládacích prvků. Jedním z těchto prvků jsou záložky pro jednotlivé projekty, které v sobě obsahují podsložky pro jednotlivé části aplikace.

Otevření a uzavření projektu probíhá intuitivně. Po vzoru moderních prohlížečů bylo zvoleno přidání klávesových zkratk pro otevření a uzavření projektů a doplněny tlačítka vedle záložek.

Samotný obsah aplikace je soustředěn do zbytku hlavního okna. Toto okno je rozděleno na 3 části (levá část, spodní část a střed aplikace), které zastávají v aplikaci různé role, ale základní rozvržení je použito v celé aplikaci.

Levá část obsahuje tlačítka rozdělená do 2 skupin – jedna skupina pro přidávání nových objektů a druhá skupina pro spojování objektů na editačním plátně.

Spodní část aplikace je pro jednotlivé části nástroje odlišná. V diagramu tříd se zde nachází možnost přidání a odebrání proměnných a metod, dále nastavení jejich viditelnosti pro ostatní třídy. V dolní části objektivě orientovaných Petriho sítí se nachází možnost přidání nových pravidel do stráží, vytvoření nové třídy, nebo použití již existující třídy. V diagramu užití je zobrazeno okolí pro vybraný objekt.

Střed aplikace, též nazýván editační plátno, je stejný pro všechny části nástroje. Po zakliknutí odpovídajícího tlačítka a kliknutí na editační plátno, se vyvolá odpovídající akce – například se přidá nová třída do diagramu tříd nebo se spojí místo se stráží v případě objektivě orientovaných Petriho sítí.

Spojování jednotlivých prvků na editačním plátně je jednoduché, při tažení hrany od jednoho prvku k druhému je tento prvek zvýrazněn. Je zde možnost zakřivit hrany pro větší přehlednost diagramu. To se provádí tak, že při tažení hrany je kliknuto na pravé tlačítko myši. Tyto hrany jsou zakončeny adekvátními šipkami, odpovídající UML definicím.

Znázornění dědičnosti je v nástroji zakončeno bílým trojúhelníkem. K řešení výpočtu místa, kde se hrana protíná s krajem objektu, bylo použito shodnosti trojúhelníků. Tím bylo docíleno uchycení šipek ke krajům objektu a možnosti „klouzání“ po okraji objektu, když je s ním manipulováno po editačním plátně.

Při spojování prvků jsou vzaty do úvahy určitá pravidla – například spojení účastníka a případu užití lze pouze za použití asociace. Nicméně tato asociace musí být aplikována ve směru od účastníka k případu užití. Po odebrání účastníka jsou odstraněny všechny hrany vedoucí k případům užití. V případě, že nevede žádná hrana k objektu nebo z objektu, je toto dáno uživateli adekvátně graficky najevo.

4.2. PROPOJENÍ ČÁSTÍ NÁSTROJE

Propojení jednotlivých částí slouží k automatické vzájemné kontrole dalších částí. Odstranění jednoho prvku v jedné části je promítnuto do druhé části tak, že je odpovídající prvek graficky znázorněn. Přidání nového prvku samozřejmě vede k vytvoření nových objektů.

5. ZÁVĚR

Výsledkem je aplikace, která je jednoduchá na používání jak zákazníkem, tak programátorem, ale není neomezuující ani pro jednu stranu v možnostech editace a vytváření. Nejdůležitější částí aplikace je vzájemné propojení, čímž se práce s tímto nástrojem značně urychlí.

ACKNOWLEDGEMENT

I'd like to thank my supervisor for sharing his knowledge about several problems and explaining connections between each part of application.

REFERENCE

- [1] KETTENIS, Jan. Getting Started With Use Case Modeling. In: *An Oracle White Paper* [online]. 2007 [cit. 2014-03-04]. Dostupné z: <http://www.oracle.com/technetwork/developer-tools/jdev/gettingstartedwithusecasemodeling-133857.pdf>
- [2] KETTENIS, Jan. Getting Started With UML Class Modeling. In: *An Oracle White Paper* [online]. 2007 [cit. 2014-03-04]. Dostupné z: <http://www.oracle.com/technetwork/developer-tools/jdev/gettingstartedwithumlclassmodeling-130316.pdf>
- [3] KOČÍ, Radek, Vladimír JANOUŠEK a František ZBOŘIL. Bject Oriented Petri Nets — Modelling Techniques Case Study. [1] *International Journal of Simulation Systems, Science & Technology*. 2009 [online]. 2009, roč. 10, č. 3 [cit. 2014-03-04]. Dostupné z: ijssst.info/Vol-10/No-3/paper4.pdf