

Projekt zespołowy - wstępne rozpoznanie problemu

Sławomir Blatkiewicz Jakub Górniak Piotr Piechal Bartosz Pieńkowski
Barnaba Turek Michał Zochniak

24 października 2010

Spis treści

1	Przedmiot projektu	2
2	Ograniczenia projektu	2
2.1	Zasoby czasowe	2
2.2	Zasoby ludzkie	2
3	Rozpoznanie problemu	3
3.1	Język	3
3.2	Parser	3
3.3	Generator plików graficznych	3
4	Założenia projektu	3
4.1	Cel podstawowy	3
4.2	Cele dodatkowe	3
5	Proponowane rozwiązania	4
6	Proponowane technologie	4

1 Przedmiot projektu

Projekt obejmuje cały proces powstawania oprogramowania umożliwiającego generowanie diagramów UML w postaci graficznej na podstawie plików tekstowych o określonej strukturze opartej na konkretnej składni.

2 Ograniczenia projektu

2.1 Zasoby czasowe

Projekt ma trwać około 9 miesięcy. Nieprzekraczalny termin zakończenia prac nad projektem to 10 czerwca 2011.

2.2 Zasoby ludzkie

Do realizacji projektu przydzielony został zespół 6 programistów w składzie:

- Sławomir Blatkiewicz
- Jakub Górniak
- Piotr Piechal
- Bartosz Pieńkowski

- Barnaba Turek
- Michał Zochniak

3 Rozpoznanie problemu

Podstawą całego projektu jest stworzenie języka, który umożliwi precyzyjny opis elementów diagramu klas w notacji UML, zgodnie ze standardami wersji 2.0. Dodatkowymi modułami potrzebnymi do realizacji tego rozwiązania będą parser oraz generator plików graficznych.

3.1 Język

Język powinien umożliwiać opisanie podstawowych elementów diagramu klas, opisanych w specyfikacji notacji UML 2.0:

- klasa
- relacja (asocjacja, agregacja, generalizacja) wraz z określeniem liczebności i ról
- notatka
- paczka/moduł

Dodatkowo zakładamy iż język powinien umożliwiać definiowanie:

- relacji n-arnej
- klasy asocjacyjnej
- ograniczenia (constraint)

3.2 Parser

3.3 Generator plików graficznych

Generator plików graficznych powinien domyślnie korzystać z formatu PNG. Dodatkowo powinien automatycznie optymalizować ułożenie elementów na diagramie.

4 Założenia projektu

4.1 Cel podstawowy

Celem podstawowym jest stworzenie programu sterowanego z linii komend, który wygeneruje plik graficzny zawierający diagram klas odwzorowujący wskazany plik tekstowy zawierający kod w utworzonym języku.

4.2 Cele dodatkowe

Celami dodatkowymi, których realizacja rozważona zostanie po osiągnięciu celu podstawowego są:

1. utworzenie zintegrowanego środowiska programistycznego (IDE) do tego języka, w skład którego wchodziłyby następujące elementy:
 - edytor tekstowy oferujący kolorowanie składni, oraz inteligentne formatowanie tekstu
 - podgląd diagramu na bieżąco
2. rozszerzenie funkcjonalności IDE o możliwość redefiniowania położenia poszczególnych elementów na diagramie w trybie graficznym (*drag and drop*).

5 Proponowane rozwiązania

6 Proponowane technologie

Proponujemy do osiągnięcia celu głównego wykorzystanie technologii języka Python. Za takim rozwiązaniem przemawiają następujące argumenty:

1. przenośność rozwiązania spowodowana skryptowym charakterem języka
2. łatwość użytkowania - brak potrzeby instalacji oprogramowania do jego poprawnego działania
3. aspekt dydaktyczny - chęć zapoznania się z proponowaną technologią

Do realizacji IDE proponujemy użycie biblioteki Qt.