

Próbna obrona pracy inżynierskiej

Temat: Aplikacja WYSIWYG do tworzenia prezentacji
multimedialnych z wykorzystaniem klasy dokumentów `BEAMER`
`LATEX-a`

Jan Adam Kumor



Instytut Automatyki i Informatyki Stosowanej

23 października 2015

Temat pracy

Autor: Jan Kumor, semestr 7. specjalność SID

Opiekun: prof. nzw. dr hab. inż. Maciej Ławryńczuk
Zespół Technik Sterowania
Zakład Automatyki i Inżynierii Oprogramowania
Instytut Automatyki i Informatyki Stosowanej

Temat: *Aplikacja WYSIWYG do tworzenia prezentacji
multimedialnych z wykorzystaniem klasy
dokumentów BEAMER L^AT_EX-a*

Konspekt

- 1 Cel pracy i możliwe zastosowania
- 2 Istniejące rozwiązania
- 3 Zapropownowane rozwiązanie i wynik pracy
- 4 Wnioski i podsumowanie

Główne założenia

- Graficzne środowisko do tworzenia prezentacji multimedialnych
- Zaadaptowanie systemu \LaTeX i klasy dokumentów `BEAMER`
- Systemy WYSIWYG (*What You See Is What You Get*)

Możliwe zastosowania

- Tworzenie dokumentów klasy `BEAMER` bez znajomości \LaTeX -a
- Szybkie prototypowanie prezentacji
- Nauka \LaTeX

Konspekt

- 1 Cel pracy i możliwe zastosowania
- 2 **Istniejące rozwiązania**
- 3 Zapropownowane rozwiązanie i wynik pracy
- 4 Wnioski i podsumowanie

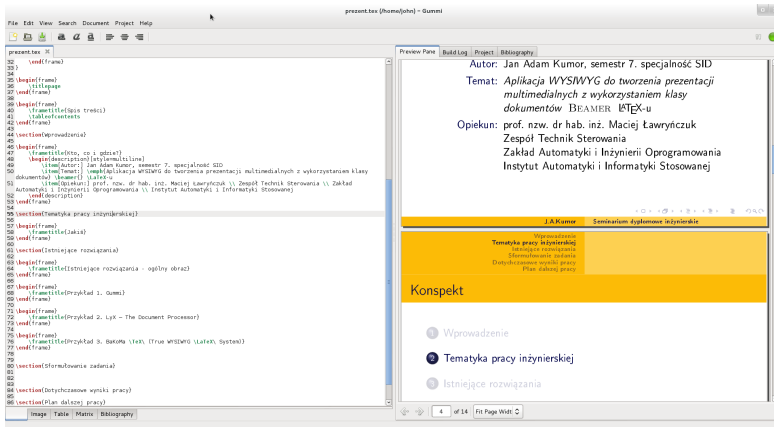
Przykład 1. Gummi¹



- Typowe dla dokumentów L^AT_EX workflow - plik .tex i podgląd wyniku kompilacji
- Automatyczna kompilacja dokumentu po każdej wprowadzonej zmianie
- Dostępne podstawowe operacje na tekście:
 - **pogrubienie**, *kursywa* oraz podkreślenie tekstu
 - wyrównanie tekstu do lewej i prawej strony oraz wyśrodkowanie
- Wsparcie GUI przy wstawianiu grafiki, tabel, macierzy i bibliografii

¹<http://gummi.midnightcoding.org/>

Przykład 1. Gummi



Rysunek: Okno programu Gummi

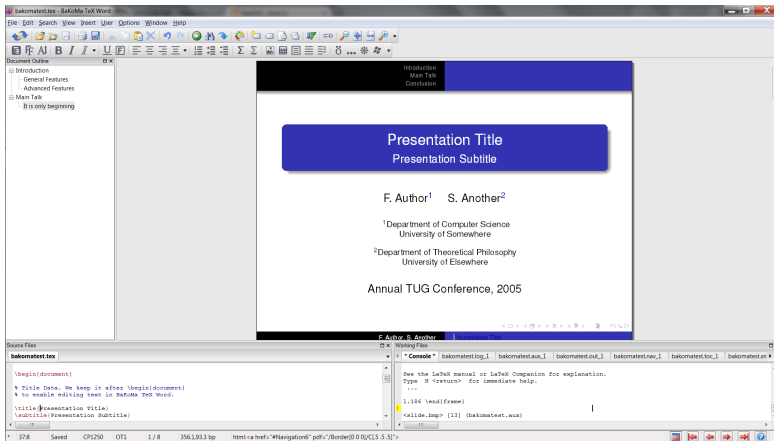
Przykład 2. BaKoMa T_EX (True WYSIWYG L^AT_EX System)²



- Pakiet aplikacji komercyjnych
- Wsparcie GUI dla większości funkcji L^AT_EX
- Obsługa wielu klas dokumentów z poziomu GUI
- Prawdziwy WYSIWYG w czasie rzeczywistym
- W tle działa silnik T_EX- tylko dla aktualnie zmienianej strony

²<http://www.bakoma-tex.com/>

Przykład 2. BaKoMa T_EX (True WYSIWYG L^AT_EX System)



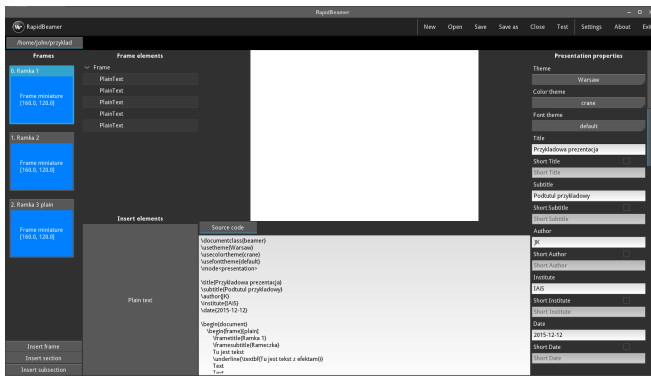
Rysunek: Okno programu BaKoMa Word

Konspekt

- 1 Cel pracy i możliwe zastosowania
- 2 Istniejące rozwiązania
- 3 Zapropownowane rozwiązanie i wynik pracy**
- 4 Wnioski i podsumowanie

Aplikacja Rapidbeamer

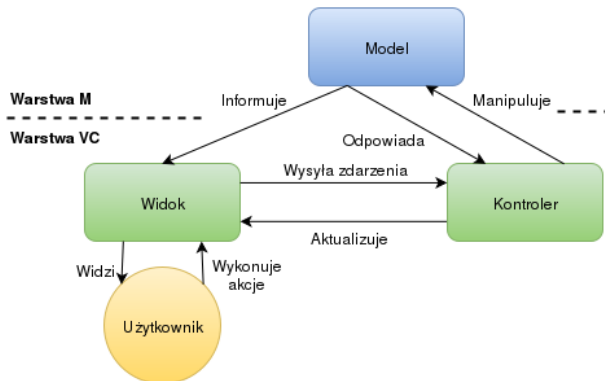
- język programowania - Python 3.5
- framework Kivy



Rysunek: Okno aplikacji Rapidbeamer

Architektura rozwiązania

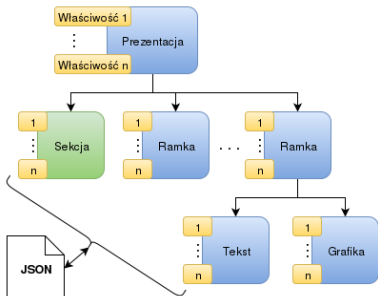
Wzorzec MVC pozwala na wygodne rozdzielenie zakresów odpowiedzialności poszczególnych części aplikacji.



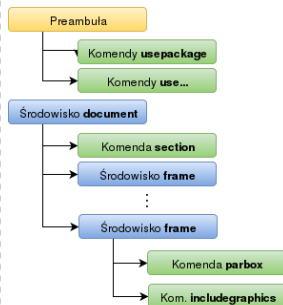
Warstwa M - Model

- Python 3.5
- Trójwarstwowa struktura modelu
 - Warstwa abstrakcyjna
 - Warstwa dokumentu \TeX
 - Warstwa generowanego kodu

Warstwa abstrakcyjna



Warstwa dokumentu \TeX



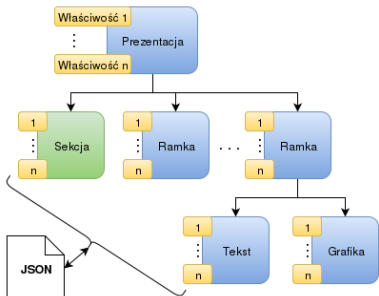
Warstwa generowanego kodu



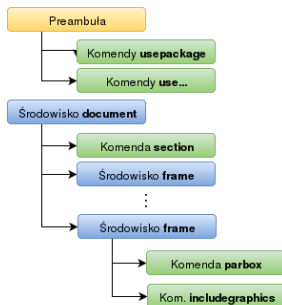
Warstwa M - Model

- Python 3.5
- Trójwarstwowa struktura modelu
 - Warstwa abstrakcyjna
 - Warstwa dokumentu \TeX
 - Warstwa generowanego kodu

Warstwa abstrakcyjna



Warstwa dokumentu \TeX



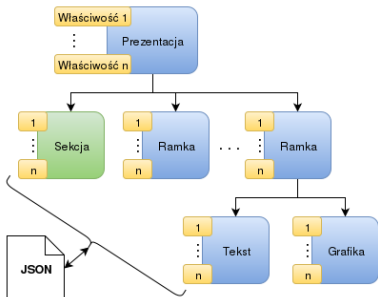
Warstwa generowanego kodu



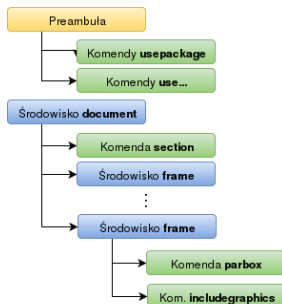
Warstwa M - Model

- Python 3.5
- Trójwarstwowa struktura modelu
 - Warstwa abstrakcyjna
 - Warstwa dokumentu \TeX
 - Warstwa generowanego kodu

Warstwa abstrakcyjna



Warstwa dokumentu \TeX

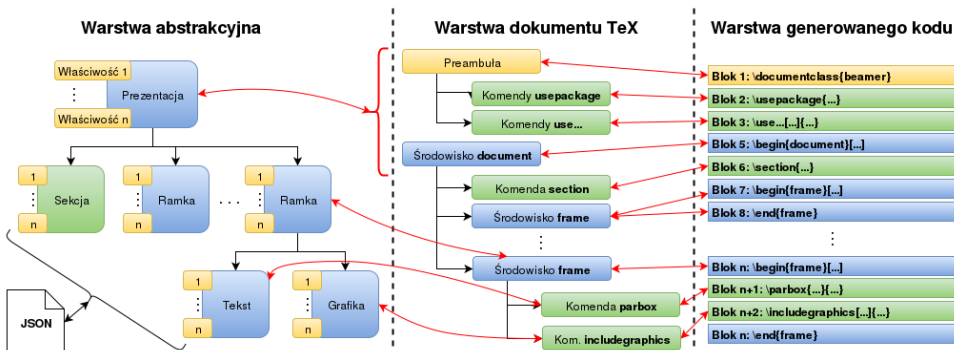


Warstwa generowanego kodu



Warstwa M - Model

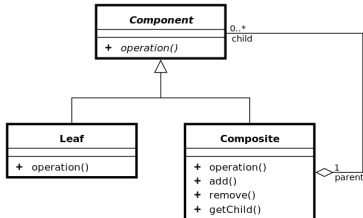
- Python 3.5
- Trójwarstwowa struktura modelu
 - Warstwa abstrakcyjna
 - Warstwa dokumentu \TeX
 - Warstwa generowanego kodu



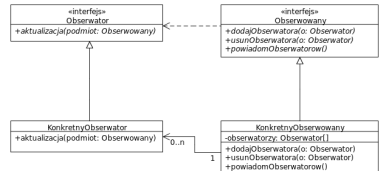
Warstwa M - Model

Realizuje większość logiki aplikacji

- Generowanie kodu \LaTeX - wzorzec kompozytu
- Zapis i odczyt projektu - format JSON
- API: metody manipulacji modelem + wzorzec obserwatora



Rysunek: "Composite UML class diagram"
autorstwa Trashtoyderivative



Rysunek: "Observer classes pl"
autorstwa Zyxist

Warstwa VC - Widok i kontroler

Umożliwia interakcję użytkownika z aplikacją.

- Sterowanie zdarzeniami
- Dwa współpracujące wątki: widoku i kontrolera
- widok zaimplementowany z użyciem frameworku Kivy

Warstwa VC - Widok i kontroler cd



Framework Kivy³

- Otwarta biblioteka języka Python do tworzenia graficznego interfejsu użytkownika
- Wieloplatformowość: *Linux, Windows, OS X, Android i iOS*
- Szerokie wsparcie dla NUI (*Natural User Interfaces*)
- Dedykowany język opisu interfejsu *kivy language*

³<http://kivy.org/>

Budowa interfejsu

The image shows the RapidBeamer application interface, which is a presentation editor. The interface is divided into several panels:

- Top Bar:** Contains the application name "RapidBeamer" and a menu with options: New, Open, Save, Save as, Close, Test, Settings, About, and Exit.
- Left Panel:** Contains a list of frames and a list of elements to insert.
 - Frames:** A list of frames, each with a title and a miniature. The frames are: 0. Ramka 1, 1. Ramka 2, and 2. Ramka 3 plain. Each frame has a miniature showing a blue rectangle with the text "Frame miniature [160.0, 120.0]".
 - Insert elements:** A list of elements to insert, including "Plain text".
- Center Panel:** A large white area for editing the presentation content.
- Right Panel:** Contains the "Presentation properties" section, which includes fields for Theme, Color theme, Font theme, Title, Short Title, Subtitle, Podtutul przykładowy, Short Subtitle, Author, Short Author, Institute, Short Institute, Date, and Short Date.

The "Presentation properties" section is currently set to the following values:

- Theme: Warsaw
- Color theme: crane
- Font theme: default
- Title: Przykładowa prezentacja
- Short Title: ☐
- Subtitle: Podtutul przykładowy
- Short Subtitle: ☐
- Author: JK
- Short Author: ☐
- Institute: IAIS
- Short Institute: ☐
- Date: 2015-12-12
- Short Date: ☐

The "Insert elements" section is currently set to "Plain text".

Moduł prezentacji

Stan zaawansowania pracy nie pozwala powiedzieć aktualnie nic na ten temat...

Konspekt

- 1 Cel pracy i możliwe zastosowania
- 2 Istniejące rozwiązania
- 3 Zapropownowane rozwiązanie i wynik pracy
- 4 Wnioski i podsumowanie

Wnioski i podsumowanie

- Udało się stworzyć aplikację umożliwiającą wygodne i szybkie tworzenie prezentacji multimedialnych.
- Więcej wniosków mogłoby dotyczyć testów wydajnościowych generatora kodu lub mechanizmu prezentacji

Dziękuję