

Filip Pawłowski

Hubert Soroka

Maciej Tymoftejewicz

ZPRP – Design Proposal

1. Temat projektu

Celem projektu jest zaimplementowanie i opublikowanie biblioteki, umożliwiającej edytowanie diagramów UML, wczytywanych z powszechnie przyjętego do ich reprezentacji formatu XML, poprzez edycję obiektów w Pythonie.

2. Planowane funkcjonalności

Chcemy wspierać różne rodzaje diagramów UML, na ten moment są to diagramy:

- Klas
- Sekwencji
- Komponentów

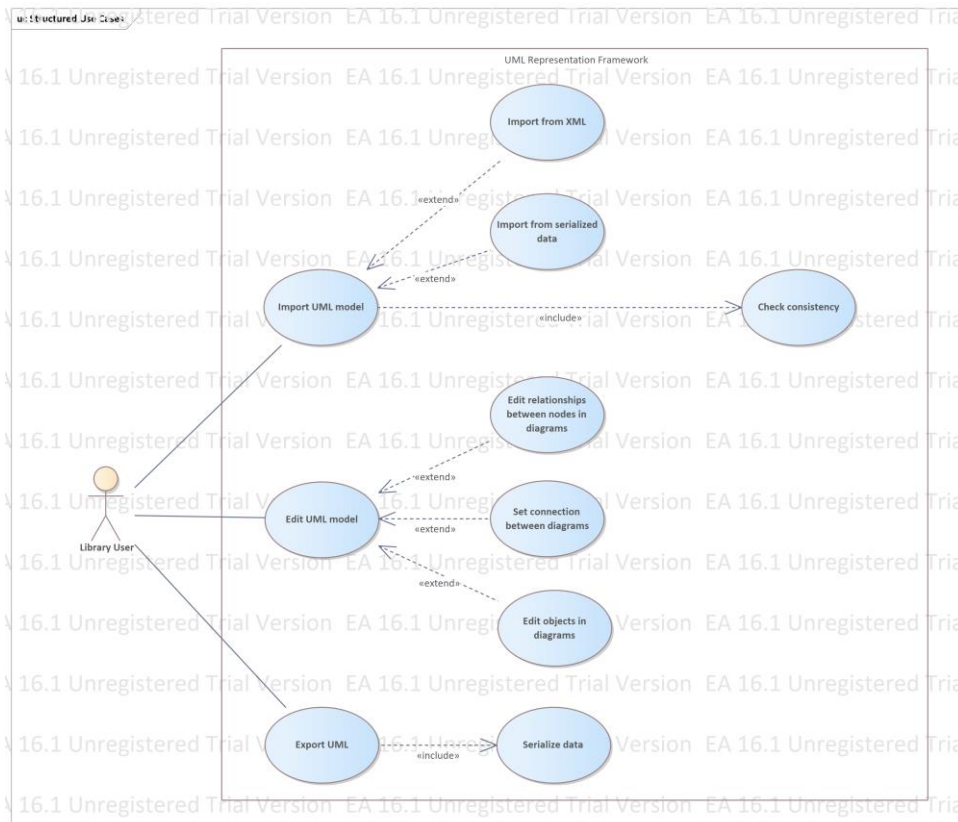
Rodzaje diagramów wymienione są w kolejności zgodnej z naszymi priorytetami. W przypadku braku czasu lub innych problemów, ograniczone zostaną dostarczone funkcjonalności diagramów komponentów.

Dokładny zakres zaimplementowanej specyfikacji UML pozostawiono na dalsze etapy projektu, w związku z koniecznością poznania dziedziny od podstaw oraz niepewnościami z tym związanymi. Docelowo oczekiwane jest, żeby biblioteka wspierała wszystkie podstawowe elementy wymienionych diagramów. Jako punkt odniesienia przyjęto ten projekt sam w sobie – a dokładniej możliwość edycji diagramów UML opisujących proponowany system z jego użyciem.

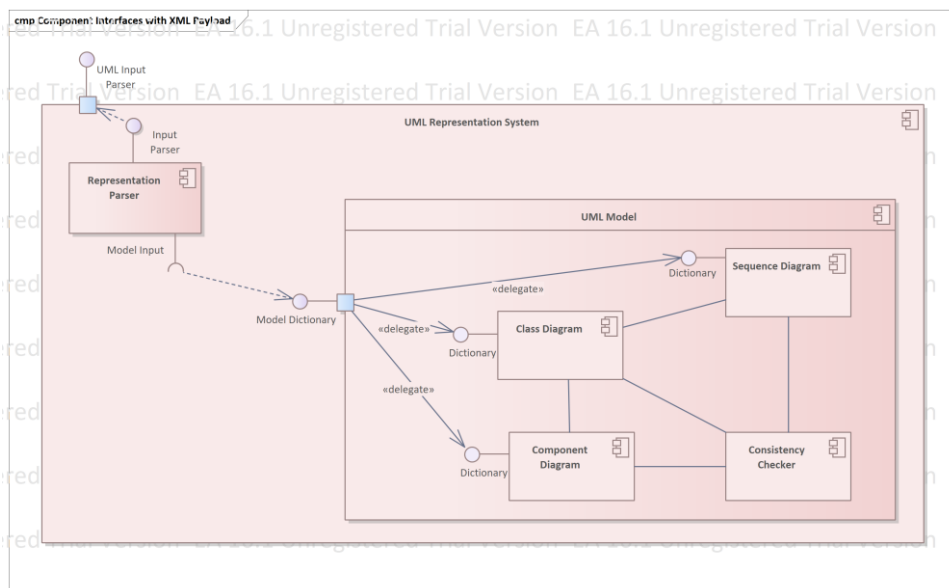
3. Model systemu

- a. Class Diagram

b. Usecase Diagram



b. Component Diagram



Chcemy umożliwić użytkownikowi odczyt diagramu UML z pliku XML po podaniu ścieżki (najpewniej pojawi się funkcja przyjmująca ścieżkę do pliku i zwracająca odczytany model). Użytkownik mógłby wtedy zmodyfikować obiekty składające się na model diagramu i zapisać

nowy diagram UML do pliku (analogiczna funkcja zapisująca model do pliku). W gruncie rzeczy chcemy wydać narzędzie przetwarzające XML w obiekty i z powrotem.

Planujemy też moduł do sprawdzania poprawności modelu w formie obiektów Pythonowych, aby użytkownik dowiedział się o błędach i braku spójności diagramu, bez konieczności otwierania pliku wynikowego zewnętrznym narzędziem.

Nie ma powszechnej, jednolitej reprezentacji diagramu UML (zwłaszcza, że diagramy UML są bardzo zróżnicowane) w formacie XML, dlatego chcemy skupić się na obsługiwaniu plików eksportowanych przez Enterprise Architect, jako narzędzie wskazane w źródłach jako jedno z najbardziej popularnych. Oczywiście w implementacji uwzględnimy możliwość wsparcia większej liczby formatów w perspektywie dalszego rozwoju. Serializator i deserializator do formatów Enterprise Architect'a będą zatem pierwszymi implementacjami interfejsów serializatora i deserializatora.

4. Stos technologiczny i publikacja

Do zarządzania zależnościami, bibliotekami i środowiskiem wirtualnym pythona będziemy używać Poetry. Ostateczną wersję biblioteki będziemy chcieli umieścić na PyPI.

Zgodnie z zaleceniami projektowymi, do testów na wielu wersjach języka użyjemy tox, a do dokumentowania MkDocs.

Do pracy przy plikach XML planujemy użyć wbudowanego modułu xml. Wydaje się dostatecznie rozbudowany na nasze potrzeby, ale nie wykluczamy użycia zewnętrznych bibliotek, jeśli pojawi się trudne do obsłużenia zagadnienie, co jest jednak bardzo wątpliwe.

5. Grafiki

Każda data to piątek w tygodniu, w którym chcielibyśmy zakończyć wymienione po myślniku zadania.

03.11.2023 - konfiguracja środowiska, narzędzi, wtyczek itd., szkielet aplikacji, zdefiniowanie interfejsów, przykładowe testy jednostkowe

10.11.2023 - kompletny model reprezentacji diagramów w pythonie (na podstawie zamieszczonych diagramów UML), początki prac nad deserializatorem (tylko diagramy klas)

24.11.2023 - demo, umożliwiające wczytanie prostego diagramu klas UML z pliku i otrzymanie modelu w obiektach

01.12.2023 - wprowadzenie uwag po prezentacji demo, rozpoczęcie prac nad serializatorem (tylko diagramy klas)

15.12.2023 - ukończenie deserializatora dla wszystkich diagramów, ukończenie serializatora dla diagramu klas

22.12.2023 - wstępna wersja dokumentacji

03.01.2024 - ukończenie serializatora dla wszystkich diagramów

10.01.2024 - ukończenie projektu, publikacja na PyPI, wraz z dokumentacją, ostatnie testy

Warto zaznaczyć, że ciężko oszacować dokładnie ilość pracy potrzebną do wykonania deserializatora i serializatora do konkretnego formatu XML wspieranego przez Enterprise Architecta, ponieważ żaden z nas nie ma doświadczenia w obróbce tego formatu plików. Występuje w tym czynnik odkrywczy.

Testy jednostkowe będą pisane równolegle do modułów aplikacji, stąd brak konkretnego terminu ich wykonania.

6. Bibliografia

- i. ISO Central Secretary, "Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2", International Organization for Standardization, Geneva, CH, Standard ISO/IEC 19501:2005, 2005. ISO UML Standard
- ii. Extensible Markup Language 1.0 (Fifth Edition) (XML), 2008. XML Standard
- iii. OMG: XML Metadata Interchange (XMI), 2015.
- iv. M. Ozkaya and F. Erata, "A survey on the practical use of UML for different software architecture viewpoints," *Information and Software Technology*, vol. 121, pp. 106-275, 2020. ISSN: 0950-5849. DOI UML Survey
- v. H. Koc, A. Erdoğ an, Y. Barjakly, and S. Peker, "UML Diagrams in Software Engineering Research: A Systematic Literature Review," *Proceedings*, vol. 74, p. 13, Mar. 2021. DOI
- vi. I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang, "What Industry Needs from Architectural Languages: A Survey," *IEEE Transactions on Software Engineering*, vol. 39, no. 6, pp. 869–891, 2013. DOI Industry Survey
- vii. J. Hutchinson, J. Whittle, and M. Rouncefield, "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure," *Science of Computer Programming*, vol. 89, pp. 144–161, 2014. ISSN: 0167-6423. DOI Model-driven Engineering
- viii. M. H. Osman and M. Chaudron, "UML Usage in Open Source Software Development: A Field Study," 2013.
- ix. ISO Central Secretary, "Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 1: Infrastructure," International Organization for Standardization, Geneva, CH, Standard ISO/IEC 19505-1:2012, 2012. ISO OMG UML
- x. M. Petre, "UML in practice," in *2013 35th International Conference on Software Engineering (ICSE)*, 2013, pp. 722–731. DOI UML in Practice