

ERD Andrzej

METODA WYKORZYSTANIA UML DO OPISU DYNAMIKI SYSTEMU WSPOMAGANIA EKSPLOATACJI ZŁOŻONYCH OBIEKTÓW TECHNICZNYCH

W pracy podjęto próbę zastosowania metod modelowania za pomocą języka UML do projektowania komputerowego systemu eksploatacji obiektów złożonych. W szczególności położono nacisk na opis dynamiki systemu rozumianej jako zmienność zachowań w wyniku interakcji z otoczeniem. Jako przykład aplikacyjny wybrano system wspomagania eksploatacji kolejowych pojazdów trakcyjnych.

DYNAMICS OF THE COMPUTER-AIDED SYSTEM FOR COMPLEX TECHNICAL OBJECT MAINTENANCE IN THE UML LANGUAGE

This paper is an attempt to use modeling methods employing the UML language for computer modeling of the complex object maintenance systems. A particular emphasis was put on the description of the system dynamics understood as changeable behavior resulting from the interaction with environment. The application case which has been selected here is a computer-aided system for railway track vehicle maintenance.

1. WSTĘP

W gospodarce każdego współczesnego kraju istnieje wiele obiektów o dużej wartości i relatywnie długim czasie eksploatacji w ramach którego następują przemienne fazy wykorzystywania i obsługi zgodne z zasadami eksploatacji. (przykład - środki transportu takie jak pojazdy lądowe, statki, samoloty). Koszt nabycia obiektu jest często znacznie mniejszy niż późniejsze łączne nakłady na eksploatację w trakcie jego użytkowania. Racjonalizacja eksploatacji może w pewnym momencie prowadzić do wycofywania z użytkowania obiektów zdalnych technicznie lecz już przestarzałych ze względu na obecność na rynku obiektów o znacznie wyższych parametrach technicznych bądź też o zdecydowanie niższych kosztach eksploatacji[4].

Syntetyczne ujęcie zjawisk i podejmowanie prawidłowych decyzji z tym związanych jest niemożliwe bez odpowiednich systemów komputerowych stanowiących zaplecze dla procesów biznesowych. Potrzeba tworzenia Komputerowych Systemów Wspomagania Eksploatacji (KSWE) jest również skutkiem zmian w procesach obsługi i zastępowania tradycyjnych cykli przeglądowo naprawczych obsługą wg aktualnego stanu opartą o bieżącą diagnostykę. Tym samym istotne jest bieżące zbieranie danych niezawodnościowych oraz parametrów eksploatacyjnych z poszczególnych obiektów będących przedmiotem eksploatacji. Pomocą są w tym przypadku pokładowe systemy

diagnostyczne, jednak w celu uzyskania danych odnośnie całej populacji obiektów i wyciągnięcia wniosków bardziej ogólnych konieczna jest ich agregacja.

O ile reprezentacja zdarzeń ekonomicznych jest stosunkowo dobrze opanowana, o tyle zdarzenia techniczne są często niedostatecznie zilustrowane i w praktyce wiele prób utworzenia systemów ponosi klęskę [8]. Zasadniczą przyczyną takiego stanu rzeczy jest ogromna złożoność systemów [7]. Oprócz wielu innych przyczyn wymienionych w pracy [7] wynika ona z:

- dużej liczby koniecznych do uwzględnienia zdarzeń o znacznej różnorodności.
- problemów komunikacji wewnątrz zespołów projektowych, a także pomiędzy jego członkami oraz użytkownikiem w fazie formułowania założeń.

W efekcie powstały produkt wymaga wielokrotnych zmian, poprawek i uzupełnień co prowadzi do znaczącego wydłużenia czasu realizacji i wielokrotnego przekroczenia zakładanego budżetu [8]. W efekcie systemy o ile w ogóle powstają mają wersję znacznie okrojoną w stosunku do pierwotnych założeń i koszt ich wytworzenia jest relatywnie bardzo wysoki.

Kolejną poważnym utrudnieniem, jest tworzenie za każdym razem systemu od początku bez wykorzystywania istniejących rozwiązań częściowych. Wyniki częściowe są najczęściej niedostępne gdyż stanowią własność firm zamawiających bądź producentów oprogramowania.

Spośród wielu [2] możliwych dróg ograniczenia problemów złożoności, jedną z najbardziej obiecujących jest modelowanie systemu przed podjęciem jego realizacji oraz w dalszej konsekwencji tworzenie wzorców projektowych. Praca tu przedstawiona ma na celu próbę zastosowania metod modelowania za pomocą języka UML w projektowaniu systemu eksploatacji obiektu złożonego. Przykłady aplikacyjne utworzono opierając się na systemie wspomagania eksploatacji pojazdów szynowych.

2. MODELOWANIE SYSTEMÓW

Tworzenie modeli systemów ma tę zaletę, że model jest jednocześnie fragmentem dokumentacji powstającego systemu i możliwe jest prześledzenie poprawności oraz kompletności powstającego produktu nim jeszcze zostanie on wytworzony. Tym samym ograniczana jest liczba niezbędnych modyfikacji wymaganych do przygotowania produktu końcowego.

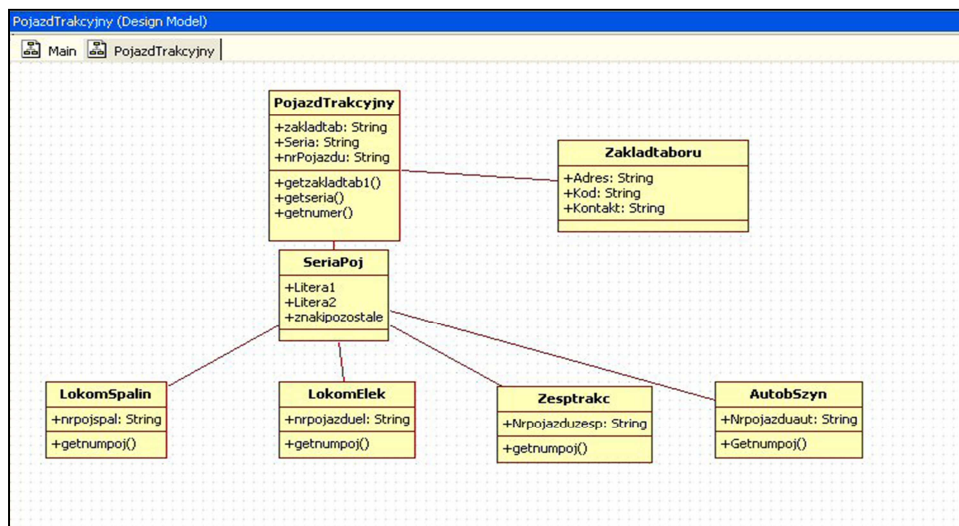
Wraz ze wzrostem komplikacji systemów rośnie liczba osób zaangażowanych w projekt, a w związku z tym pojawiają się istotne bariery komunikacyjne i modele są wówczas podstawowym narzędziem dokumentowania uzgodnień. Oczywiście wymagane są do tego celu narzędzia programistyczne z grupy CASE. Mimo ich najczęściej wysokiego kosztu, w ostatnim czasie pojawiła się grupa dostępnych bezpłatnie.

Na model systemu składają się:

- ❖ Składniki informacyjne – opis statyczny – konstrukcja bazy danych
- ❖ Informacje wynikowe oraz ich sposób prezentacji – interfejs użytkownika
- ❖ Sposób przetwarzania danych – logika aplikacji

Jest to jeden z częściej stosowanych podziałów systemu na warstwy. Istotą architektury warstwowej jest taka konstrukcja aplikacji by funkcje poszczególnych warstw były od siebie niezależne.

Niektórzy autorzy wyróżniają większą liczbę warstw np. koncepcja architektury 5 warstwowej jest pokazana w pracy [6].



Rys. 1 Uproszczony model klas opisu pojazdu trakcyjnego

Znacząca część opisu statycznego jest możliwa do wykonania za pomocą diagramów klas. Pokazują one składające się na klasę atrybuty oraz metody na nich operujące. Na rys. 1 pokazano prosty model składających się na opis pojazdu trakcyjnego będącego podmiotem KSWE pojazdów trakcyjnych.

Model ten pozwala na ujęcie zależności pomiędzy atrybutami klas potomnych wchodzących w skład opisu, oraz metodami pozwala także na rozróżnienie metod prywatnych i publicznych

Zaletą diagramów klas i modeli za ich pomocą wykonanych, jest możliwość automatycznej generacji na ich podstawie kodu programu.

Kod może być generowany w zależności od stosowanego narzędzia w różnych językach np. C/C++, Java, Visual Basic, Delphi, JScript, VBScript, C#, VB.NET. Dla pokazania przykładu wykorzystano program StarUML i ustawiono generację na język C++. Efekt działania zaprezentowano na rys 2. oraz 3.

Zwraca uwagę fakt utworzenia wszystkich koniecznych plików kodu (.cpp) wraz z plikami nagłówkowymi (.h). W utworzonych plikach zawarte są deklaracje atrybutów oraz metod, co jest widoczne w przedstawionym pliku PojazdTrakcyjny.cpp. Oczywistym jest, że w docelowym modelu systemu zbiory atrybutów każdej z klas powinny być bardziej obszerne [3] podobnie jak lista dostępnych metod.

Niezmiernie ważnym jest sprawdzanie poprawności deklaracji przez programy narzędziowe. Poprawność jest sprawdzana zarówno od strony formalnej jak również badana jest kompletność wszystkich deklaracji oraz spójność pomiędzy poszczególnymi

```

Katalog: C:\..\Badawcze\UMLowe\PojazdTrakcyjnyPlikiCpp

2011-06-16 23:53 <DIR>      .
2011-06-16 23:53 <DIR>      ..
2011-05-11 01:56           247 AutobSzyn.cpp
2011-05-11 01:56           336 AutobSzyn.h
2011-05-11 01:56           247 LokomElek.cpp
2011-05-11 01:56           335 LokomElek.h
2011-05-11 01:56           253 LokomSpalin.cpp
2011-05-11 01:56           343 LokomSpalin.h
2011-05-11 01:56           355 PojazdTrakcyjny.cpp
2011-05-11 01:56           451 PojazdTrakcyjny.h
2011-05-11 01:56           340 SeriaPoj.h
2011-05-11 01:56           356 Zakladtaboru.h
2011-05-11 01:56           247 Zesptrakc.cpp
2011-05-11 01:56           337 Zesptrakc.h

```

Rys. 2 Pliki wygenerowane na podstawie diagramu klas pokazanego na rys. 1

```

//
// Generated by StarUML(tm) C++ Add-In
//
// @ Project : PojazdTrakcyjny
// @ File Name : PojazdTrakcyjny.cpp
// @ Date : 2011-05-11
// @ Author : AndrzejErd
//
//
#include "PojazdTrakcyjny.h"

void PojazdTrakcyjny::getzakladtabl() {

}

void PojazdTrakcyjny::getseria() {

}

void PojazdTrakcyjny::getnumer() {

}

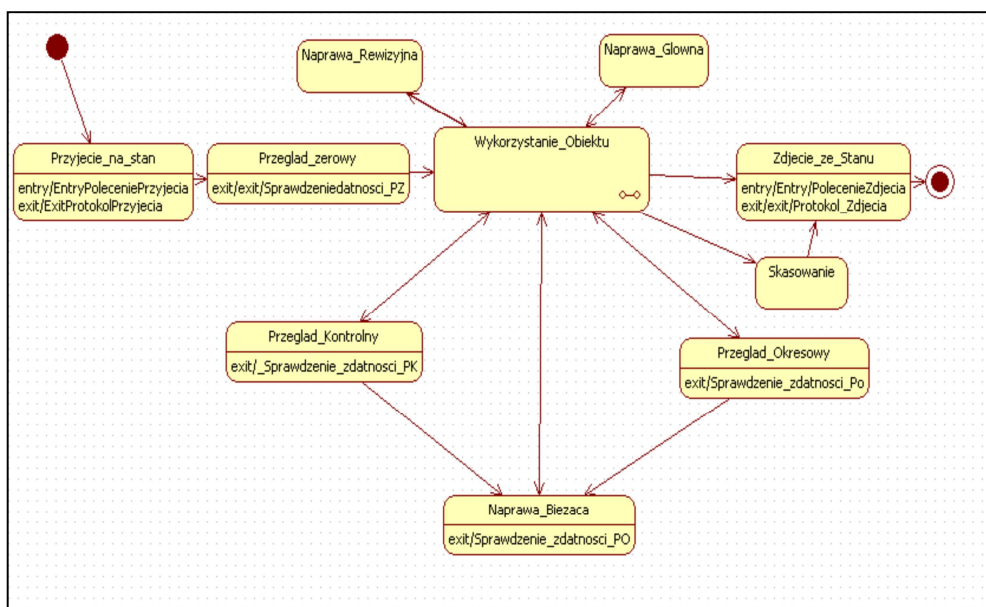
```

Rys. 3 Wnętrze pliku PojazdTrakcyjny.cpp

klasami. Oczywiście jest że kod metod jest zależny od woli programisty i musi być on przez niego uzupełniony, jednak i tak pomoc ze strony narzędzi jest znacząca.

3. MODELE DYNAMIKI SYSTEMU

Przez dynamikę systemu rozumiana jest zmienność systemu w sensie nie tylko parametrów ale również w sensie zachowań. To co dzieje się z systemem jest widoczne na zewnątrz poprzez interfejs użytkownika. Ilustracja oddziaływania użytkowników na system, jak również systemu na otoczenie jest przedstawiana jako diagram przypadków użycia (Use Case Diagram UCD) . W diagramie tym poszczególne grupy użytkowników tzw. aktorzy są łączeni za pomocą strzałek z owalami reprezentującymi czynności. Zakres i stopień przetworzenia informacji zależy od odbiorcy i przeznaczenia. Stąd diagramu przypadków użycia pozwala na uchwycenie wszelkich form współpracy pomiędzy otoczeniem a systemem. W opisie czynności będącym nieodzownym uzupełnieniem UCD powinny być wyspecyfikowane wszystkie aspekty wykonywanej przez użytkownika czynności.



Rys. 4 Diagram maszynowej stanowej wykorzystania obiektu.

W wyniku wprowadzonych przez użytkownika danych, (lub pozyskanych samodzielnie przez system z zewnątrz od innych systemów) zachodzą zmiany jego elementów. W sensie ogólnym poszczególne elementy charakteryzują się pewnymi parametrami pozwalającymi je zaklasyfikować do rozróżnialnych grup. Przynależność do grup jest zmieniana w zależności od czynności wykonywanych nad elementami systemu. Stąd można powiedzieć,

że poszczególne elementy znajdują się w kolejnych stanach a ich zmienność jest ilustrowana poprzez Diagramem maszyny stanowej (State Diagram).

W prostokącie ilustrującym stan oprócz jego nazwy można dodać akcję która ma być wykonana w trakcie wejścia oraz akcję wykonywaną na zakończenie pobytu w określonym stanie. W przypadku bardziej rozbudowanych diagramów stanów mogą one zawierać sub-stany tak jak pokazano na rys. 5. Stan Przegląd_Okresowy zasadniczo jest wykonywany gdy nastąpi przekroczenie przebiegu określonego odrębnymi przepisami dla danej serii pojazdów (stanowi to warunek wejścia do stanu)

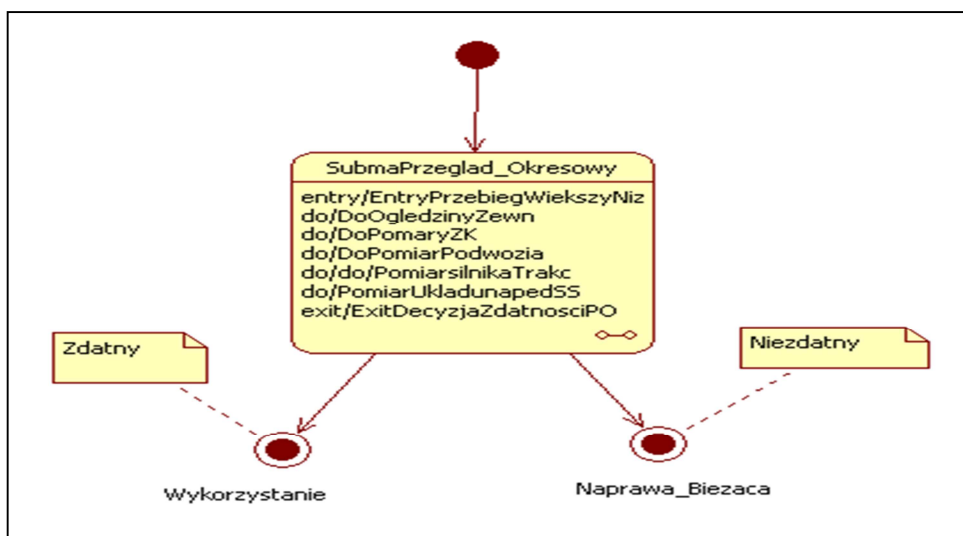


Fig. 5. Diagram sub-stanu Przegląd_Okresowy

Kolejne metody takie jak DoOgledzinyZewn (ogledziny zewnętrzne), DoPomaryZK (pomiary zestawów kołowych), DoPomiarPodwozia (Pomiary podwozia). DoPomiarsilnika-Trac (Pomiary silników trakcyjnych), DoPomiarUklNapedSS (Pomiar Układu Napędowego Silnika Spalinowego/Pojazd Spalinowy), są wykonywane w celu realizacji stanu nadrzędnego. W praktyce mogą one stanowić:

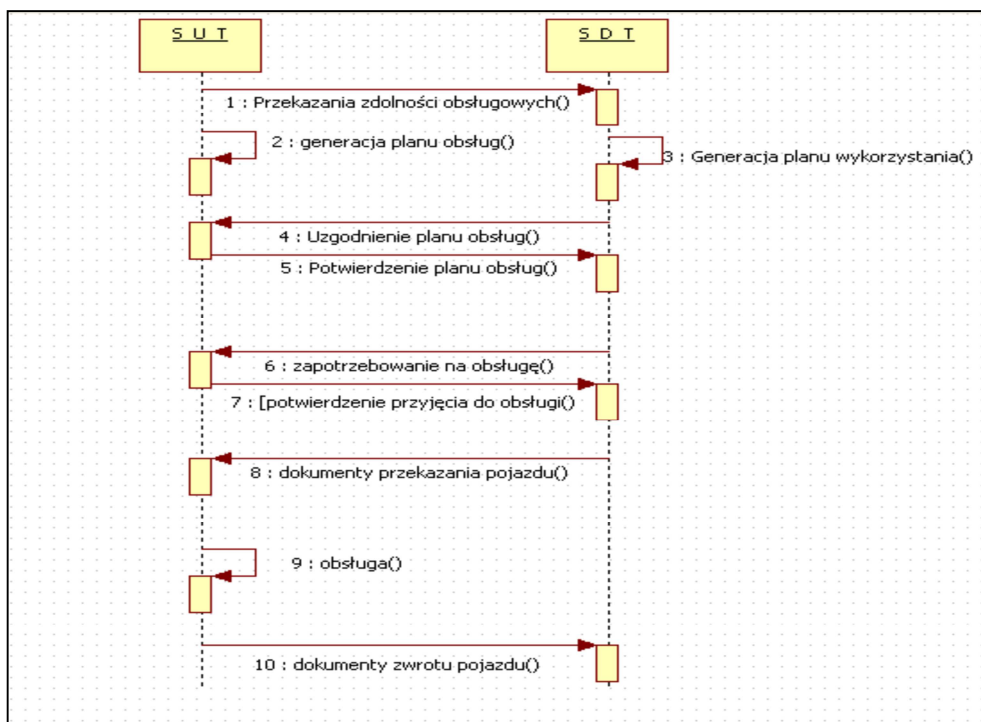
- arkusze ewidencyjne wykonania poszczególnych czynności pomiarowych nad obiektem.
- metody wartościujące stan poszczególnych podzespołów na podstawie danych posiadanych przez system.
- zadania do wykonania przez zewnętrzne systemy np. pomiarowo kontrolne.

W zależności od tego czy metoda może być zrealizowana natychmiast czy też jej wynik zostanie określony w czasie przyszłym warunkiem wyjścia ze stanu jest podjęcie decyzji o zdatności /lub nie, obiektu do dalszej eksploatacji.

Przykładowo metoda Pomiar Układu Napędowego Silnika Spalinowego jest odrębnym zadaniem składającym się z szeregu procedur składowych obejmujących pomiary:

- ❖ charakterystyki w stanie bez obciążenia ale przy różnych prędkościach obrotowych.
- ❖ charakterystyki zewnętrznej w stanie obciążenia $U=f(I)$ napięcia prądnicy głównej w funkcji prądu
- ❖ charakterystyki dynamicznej zespołu napędowego
- ❖ jednostkowego zużycia paliwa silnika spalinowego w charakterystycznych punktach pracy
- ❖ godzinowego zużycia paliwa przy obciążeniu nominalnym

Z zestawu koniecznych do wykonania zadań można wnioskować, że uzyskanie wyniku końcowego może trwać nawet kilka godzin. Tym samym pobyt w stanie Przeglad_Okresowy może być równie długi. Podobnie stan Naprawa_Biezaca może trwać kilka dni. W takiej sytuacji jest istotne uwzględnienie zależności czasowych zdarzeń występujących w systemie, do tego celu znakomicie są dostosowane diagramy sekwencji (sequence diagram).



Rys.6 Diagram sekwencji komunikatów przekazywania pojazdów do naprawy

KSWE powinien ilustrować działanie z jednej strony Zakładu Napraw i z drugiej Działu zajmującego się wykorzystaniem, tym samym wydaje się racjonalnym rozdzielenie systemu na dwa podsystemy.

Jeden zajmuje się Dyspozycją Taboru (SDT) i jego wykorzystywaniem, drugi natomiast zajmuje się czynnościami utrzymaniowymi (SUT). Łącznikami pomiędzy systemami są ze strony Działu Dyspozycji zamówienia na obsługi, a ze strony Działu Utrzymania potwierdzenia wykonania i zgłoszenia gotowości do ich wykonania. Ponieważ obsługi muszą być wykonywane w określonych czasokresach w związku z tym konieczny jest nadrzędny plan eksploatacji obejmujący harmonogramowanie wykorzystania i obsług. Zależności czasowe związane z tymi operacjami pokazano na rys. 5.

4. PODSUMOWANIE

W specyfikacji UML 2.0 (standard ISO/IEC 19501) przedstawionej jako wzorzec przez Object Management Group, organizację zrzeszającą twórców metod obiektowych, występuje jeszcze wiele innych rodzajów diagramów. Pozwalają one uwzględnić w modelu mniej powszechne aspekty działania systemów. np. diagramy komponentów, współpracy czy wdrożenia. Początkowa tendencja do tworzenia odrębnych metod obiektowych w zależności od zastosowania[1] została wyparta przez Język UML. Język ten pomyślany początkowo jako narzędzie modelowania i dokumentowania oprogramowania wydaje się być znakomitym narzędziem również w zastosowaniu do modelowania procesów eksploatacyjnych i biznesowych. Dalszą konsekwencją stosowania modeli systemów wspomagania eksploatacji wydaje się być opracowanie wzorców projektowych takich systemów o znacznie większej uniwersalności niż aktualnie istniejące wersje komercyjne.

5. BIBLIOGRAFIA

- [1] Booch G., Rumbaugh J., Jacobson I., Język UML. Przewodnik użytkownika, Seria: Inżynieria oprogramowania, Warszawa, WNT, 2002.
- [2] Erd Andrzej – Elements of the UML Model of the Rail Vehicle Maintenance System – Journal of POLISH CIMAC vol. 4. Gdansk University of Technology, Polish Academy of Science - Transport and Mechanical Engineering Committee's Gdansk 2009.
- [3] Erd Andrzej, Szycha Leszek – Model systemu eksploatacji uwarunkowany diagnostycznie. Praca niepublikowana Politechnika Radomska, Radom 2010.
- [4] Life Cycle Analysis Handbook - State of Alaska - Department of Education & Early Development 1st Edition - Education Support Services / Facilities 1999.
- [5] Miles Russ, Hamilton Kim – Learning UML 2.0 O'Reilly Media Inc. 2006. wydanie polskie – Helion Gliwice 2007.
- [6] Shalloway A.,Trott James R.,Projektowanie zorientowane obiektowo. Wzorce projektowe. Gliwice, Helion, 2005.
- [7] Subieta Kazimierz- Obiektowość w projektowaniu systemów baz danych. Akademicka Oficyna Wydawnicza PLJ Warszawa 1998.
- [8] Yourdon Edward - Marsz ku klęsce. Wydawnictwa Naukowo Techniczne Warszawa 2007.