

Tomasz GRATKOWSKI

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Wymagania i założenia dla prototypu systemu wspierającego analityka systemowego w procesie projektowania

Mgr inż. Tomasz GRATKOWSKI

Skończył studia na Wydziale Elektrycznym Politechniki Zielonogórskiej, kierunek Elektrotechnika, specjalność Inżynieria Systemów Komputerowych. Jest pracownikiem Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego oraz współpracuje z firmami z branży informatycznej. Głównym tematem zainteresowań naukowych są projektowanie systemów informatycznych, metodyki używane w inżynierii oprogramowania oraz systemy rozproszone.



e-mail: T.Gratkowski@iie.uz.zgora.pl

Streszczenie

Celem prowadzonych badań jest zbudowanie prototypu systemu, wspierającego analityka systemowego w procesie projektowania wielowarstwowych systemów informatycznych. Przyjęto, iż proces projektowania realizowany będzie zgodnie z metodyką Cheesmana-Danielsa, która została opracowana z myślą o systemach wytwarzanych w technologiach komponentowych. Zadaniem prototypu będzie automatyzacja wybranych zadań realizowanych przez analityka zgodnie z wytycznymi metodyki Cheesmana-Danielsa. W prezentowanym artykule przedstawiono wstępne wymagania i założenia dla prototypu systemu wspierającego analityka systemowego w procesie projektowania. W założeniach przyjęto również, iż w procesie automatyzacji zostanie wykorzystany system regułowy. Zastosowanie systemu regułowego pozwoli zautomatyzować wybrane zadania, do których rozwiązania, wymagana jest wiedza człowieka, a którą można zapisać przy użyciu reguł.

Słowa kluczowe: metodyka Cheesmana-Danielsa, systemy regułowe, projektowanie wielowarstwowych systemów komponentowych.

Requirements and Principles for the Prototype of the System Supporting the System Analyst in the Designing Process

Abstract

The aim of the research is to build the prototype of the system supporting the system analyst in the process of designing the n-tier informatic systems. The process of designing is carried out according to the Cheesman-Daniels methodology, which has been worked out for designing the n-tier component system. The main task of the prototype will be automatization of selected tasks that are carried out by the analyst, according to the Cheesman-Daniels methodology. This paper presents the introductory requirements and principles for the prototype of the system supporting the system analyst in the process of designing. According to the principles, the rule-based system will be used in the process of automatization. Using the rule-based system will allow to automatize the selected tasks that are solved with help of human knowledge which can be written down as a system of rules.

Keywords: Cheesman-Daniels methodology, rule-base systems, projecting n-tier component systems.

1. Wstęp

Proces projektowania systemu informatycznego jest istotnym etapem podczas wytwarzania oprogramowania. Zastosowanie etapu projektowania przyczynia się do wytwarzania lepszego oprogramowania oraz zmniejsza ryzyko niepowodzenia projektu. Proces projektowania jest etapem trudnym, wymagającym wiedzy i doświadczenia. Dodatkowo, wraz ze wzrostem złożoności projektowanego systemu, również proces projektowania staje się bardziej złożony.

W celu uproszczenia procesu wytwarzania oprogramowania, co za tym idzie uproszczenia procesu projektowania, opracowano szereg metodyk, które definiują czynności do wykonania oraz ścieżki przejść pomiędzy czynnościami. Proces wytwarzania stał się prostszym, dzięki podziałowi na mniejsze prostsze czynności. Jednak mimo uzyskanego uporządkowania procesu projektowania, jest on nadal procesem złożonym i trudnym. Przyczynia się to do częstego pomijania etapu projektowania. Dlatego dla wybranego procesu wytwarzania oprogramowania, postanowiono opracować mechanizm wspierający analityka systemowego na etapie projektowania systemu. Poniższy artykuł przedstawia wstępne wymagania i założenia dla prototypu systemu wspierającego w procesie projektowania (wwPP). W kolejnych punktach przedstawiono główne wymagania stawiane systemowi, główne modele definiujące działanie systemu oraz opisano oczekiwane korzyści płynące z dostępności proponowanego narzędzia.

2. Wymagania

W pierwszej kolejności dokonano analizy wymagań projektowych. W poniższej tabeli przedstawiono główny problem, który zostanie rozwiązany poprzez zastosowanie projektowanego prototypu.

Element	Opis
problem polega na...	skomplikowanym procesie projektowania wielowarstwowych systemów komponentowych
problem dotyczy...	analityków systemowych, projektantów systemów wielowarstwowych
rezultatem problemu jest...	pomijanie fazy projektowania, błędy w projekcie przekreślające możliwość pozytywnego zakończenia projektu
korzyści rozwiązania problemu...	większa dostępność zespołów informatycznych do metodyki projektowania systemów wielowarstwowych

Na bazie postawionego problemu, opracowano zbiór wymagań funkcjonalnych, który zawiera trzy główne wymagania:

F1. **Opracowanie artefaktów.** W ramach systemu możliwe będzie opracowanie artefaktów przy pomocy narzędzia do modelowania typu CASE [1]. Rodzaje artefaktów zależą od przyjętej metodyki projektowania. Przyjęto, iż proces projektowania realizowany będzie zgodnie z metodyką Cheesmana-Danielsa [2] (CD). Założenie to zostało opisane poniżej w ramach definicji wymagań niefunkcjonalnych.

F2. **Przetwarzanie artefaktów.** Zgodnie z metodyką CD oraz ze specyfikacją SPEM [7], najmniejszym elementem w strukturze procesu jest krok. Definiując ogólnie, kroki służą do przetwarzania artefaktów wejściowych do artefaktów wyjściowych, zgodnie z przyjętymi dla kroku regułami. Projektowany system powinien wspierać proces przetwarzania artefaktów w ramach kroków.

F3. **Zarządzanie przepływem.** W ramach metodyki zostały zdefiniowane ścieżki przejść pomiędzy krokami, które wykonywane są w czynnościach. Natomiast czynności podlegają uporządkowaniu w etapach, które opisują przepływ pomiędzy czynnościami. W celu uproszczenia wykorzystania projektowanego prototypu, należy wyposażyć go w mechanizm zarządzania przepływem na poziomie czynności, etapów oraz dziedziny [2].

Dodatkowo opracowano zbiór wymagań niefunkcjonalnych, które precyzyjnie wskażą zakres projektowanego systemu:

NF1. System dla metodyki Cheesmana-Danielsa. Na podstawie przeprowadzonej analizy dostępnych metodyk [6], oraz przy założeniach, iż metodyka powinna: wspierać projektowanie wielowarstwowych systemów komponentowych, posiadać precyzyjnie zdefiniowaną strukturę powiązanych czynności oraz być nieskomplikowaną, zdecydowano o wyborze metodyki CD.

NF2. Opracowywanie artefaktów jako niezależny moduł. W celu łatwiejszego zaimplementowania zdefiniowanej funkcjonalności w ramach wymagań funkcjonalnych, każde z trzech głównych wymagań będzie zrealizowane jako niezależny moduł.

NF2.1. Eksport danych do XMI 1.1 [2.1, DI 1.0]. Dla eksportowanych z systemu plików reprezentujących artefakty przyjęto dokładnie takie same założenia jak w NF2.1.

NF2.2. Import danych z XMI 1.1 [2.1, DI 1.0]. Dla elastyczności rozwiązania przyjęto, iż w pierwszej wersji system będzie umożliwiał importowanie artefaktów z plików zapisanych zgodnie ze standardem XMI 1.1 [9]. Przyjęte założenie wynika z faktu, iż w chwili obecnej większość darmowych narzędzi wykorzystuje właśnie wspomniany standard. Docelowo w kolejnych wersjach dostępny będzie standard XMI 2.1 [10] oraz Diagram Interchange 1.0 [3]. Przyjęty standard powinien umożliwić zapisanie wykorzystywanych w metodyce artefaktów.

NF3. Przetwarzanie kroku jako niezależny moduł. Jednym z ważniejszych elementów projektowanego prototypu jest proces przetwarzania artefaktów, który odbywa się w ramach kroku. Przyjęto, iż przetwarzanie zrealizowane będzie poprzez niezależny moduł. Z funkcji modułu przetwarzanie kroku będzie korzystać moduł zarządzający przepływem.

NF3.1. Zdalny dostęp do modułu. Ponieważ moduł przetwarzania kroku udostępnia swoją funkcjonalność innym modułom przyjęto, iż dostęp do modułu odbywać się będzie w sposób zdalny.

NF3.2. Przetwarzanie z wykorzystaniem systemu regułowego. Na bazie przeprowadzonych badań [4, 5] zdefiniowano, iż proces przetwarzania artefaktów wykorzystywać będzie zaprojektowany do tego celu system regułowy. W pracy [5] opisano architekturę proponowanego systemu regułowego. Zastosowanie systemu regułowego pozwoli odwzorować reguły, którymi powinien posługiwać się analityk systemowy zgodnie z metodyką CD. Poprzez zastosowanie niezależnego zbioru reguł, można je będzie rozbudowywać i poprawiać, przez co powinna wzrosnąć precyzyjność systemu.

NF3.3. System regułowy zbudowany przy użyciu darmowego, otwartego środowiska do budowy systemów regułowych. Dzięki zastosowaniu otwartego darmowego środowiska do budowy systemów regułowych, zmniejszone zostaną koszty implementacji.

NF3.3. Repozytorium do przechowywania artefaktów. Zaimportowane artefakty oraz uzyskane artefakty podczas przetwarzania dokonanego w kroku, są przechowywane w repozytorium w przygotowanej do tego celu strukturze.

NF4. Zarządzanie przepływem jako niezależny moduł. Zarządzanie przepływem jest głównym modułem, scalającym pozostałe moduły, zaimplementowanym niezależnie.

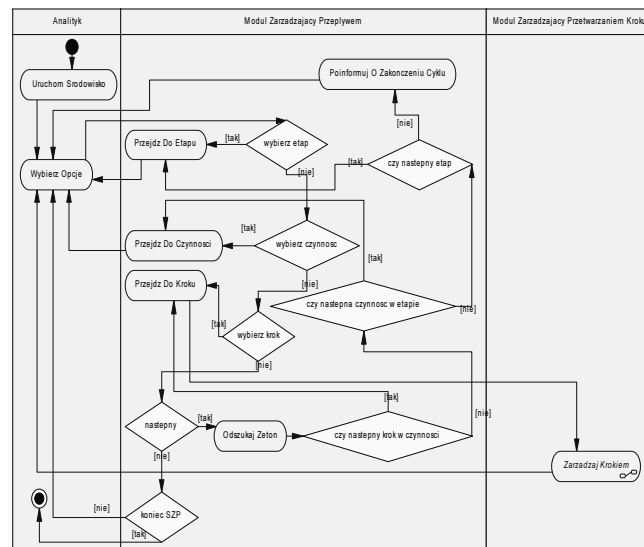
NF4.1. Wizualizacja przepływu przy użyciu diagramów aktywności. W celu zwiększenia czytelności modułu zarządzanie przepływem zostanie przedstawiony przy użyciu diagramów aktywności [7]. Proces przetwarzania będzie kontrolowany poprzez przekazywanie żetonu pomiędzy zadaniami. Po wykonaniu zadania żeton zostaje przekazany do kolejnego zadania, zgodnie z przepływem zadań w metodyce CD. Żeton w zadaniu informuje o bieżącym wykonywanym zadaniu.

NF4.2. Odzworowanie hierarchii. Zarządzanie przepływem powinno zostać zrealizowane na trzech poziomach: dziedziny, etapu i kroku [2].

NF4.3. Dodatkowa wizualizacja przy użyciu drzewa. Dodatkowo moduł powinien przechowywać informację o etapach, czynnościach, krokach, artefaktach wejściowych i wyjściowych w formie drzewa. Drzewo powinno odzwierciedlać hierarchię z metodyki CD oraz wskazywać artefakty wejściowe i wyjściowe dla kroków.

3. Model systemu

Na bazie przedstawionych w punkcie drugim wymagań opracowano model biznesowy [2] opisujący działanie projektowanego systemu. Przy użyciu diagramu aktywności na rysunku 1 przedstawiono przepływ zadań w ramach modułu System Zarządzający Przepływem. Głównym klientem modułu jest analityk systemowy, który porusza się po kolejnych zdefiniowanych w metodyce CD krokach (opcja następny) lub samodzielnie steruje przepływem (pozostałe opcje). Przejście do zadania Przejdź Do Kroku skutkuje przejściem do modułu System Zarządzający Przetwarzaniem Kroku. System zarządzania przepływem odpowiedzialny jest za zarządzanie żetonem, wskazującym aktualnie wykonywany krok.

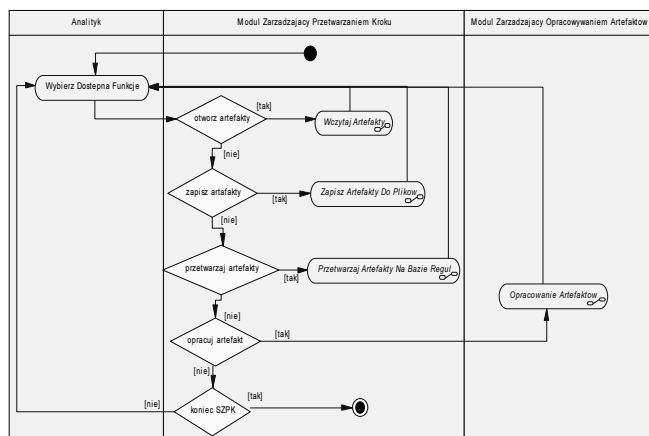


Rys. 1. Diagram aktywności przedstawiający działanie modułu zarządzającego przepływem

Fig. 1. Activity diagram describing the flow-managing module

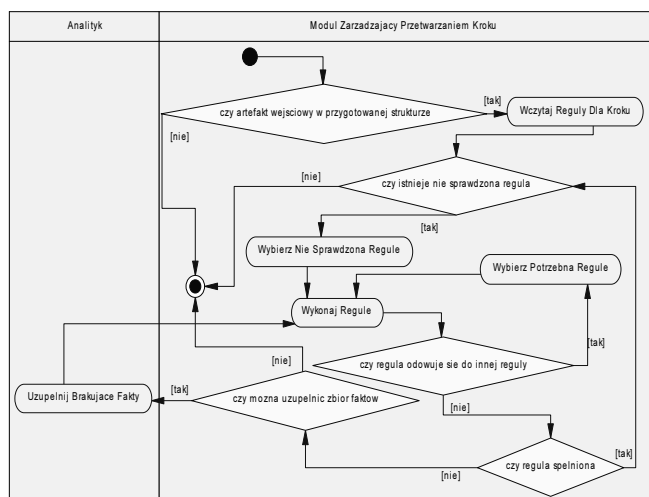
Na rysunkach 2 i 3 przedstawiono przepływ zadań w module System Zarządzający Przetwarzaniem Kroku. Rysunek 2 przedstawia główne zadania wykonywane w ramach modułu. Do wykonania każdego kroku wymagane są artefakty wejściowe, dlatego analityk może zaimportować potrzebne artefakty z plików (opcja Wczytaj Artefakty, wymaganie NF2.1). Może również po wykonaniu kroku eksportować artefakty do plików (opcja Zapisz Artefakty Do Plików, wymaganie NF2.2). Analityk

może również przejść do zewnętrznego modułu System Zarządzający Opracowywaniem Artefaktów (opcja Opracowanie Artefaktów). Głównym zadaniem wykonywanym w ramach omawianego modułu jest przetwarzanie artefaktów w kroku (opcja Przetwarzaj Artefakty Na Bazie Regul). Przepływ dla tego zadania został przedstawiony na rysunku 3. W ramach kroku następuje przetworzenie artefaktów wejściowych na bazie zdefiniowanych dla kroku reguł oraz informacjach zawartych w artefaktach. Jednak mogą zaistnieć sytuacje gdy system nie będzie w stanie podjąć decyzji. Dlatego algorytm postępowania uzupełniono o dodatkową interakcję z analitykiem, który będzie mógł uzupełnić zbiór faktów potrzebny do realizacji reguły (blok decyzyjny czy można uzupełnić zbiór faktów).



Rys. 2. Diagram aktywności przedstawiający działania modułu zarządzającego przetwarzaniem w ramach kroku

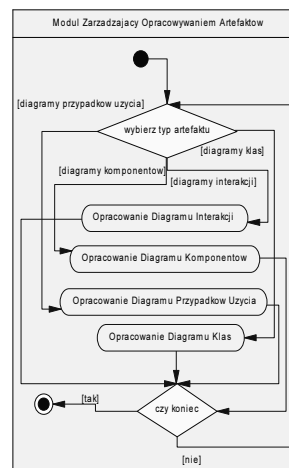
Fig. 2. Activity diagram describing the module managing processing within one step



Rys. 3. Diagram aktywności przedstawiający przetwarzanie artefaktów w ramach kroku

Fig. 3. Activity diagram describing artifacts processing within one step

Ostatnim elementem systemu jest moduł Zarządzający Opracowywaniem Artefaktów, przedstawiony na rysunku 4. Moduł jest odpowiedzialny za rysowanie artefaktów. Jest to moduł, którego złożoność jest największa. Z tego też powodu, w początkowych wersjach systemu moduł ten zostanie pominięty. Artefakty będą przygotowywane w zewnętrznych narzędziach do modelowania w języku UML [8], które umożliwiają export diagramów do plików zgodnych ze standardem XMI 1.1.



Rys. 4. Diagram aktywności przedstawiający działanie modułu opracowującego artefakty

Fig. 4. Activity diagram describing a module preparing artifacts

4. Podsumowanie

Poniższy artykuł zawiera definicję głównych wymagań dla prototypu systemu wspierającego analityka systemowego w procesie projektowania wielowarstwowych systemów komponentowych. Na bazie zdefiniowanych wymagań przedstawiono główne algorytmy dla poszczególnych modułów, na które został podzielony prototyp. Zbiór wymagań i algorytmów pozwoli na precyzyjniejsze zaprojektowanie modułów. Zaimplementowany prototyp pozwoli na empiryczną weryfikację przyjętego teoretycznego modelu automatyzacji procesu projektowania. W przyszłości prototyp może stać się podstawą do utworzenia profesjonalnego narzędzia.

5. Literatura

- [1] Definicja CASE z Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/legacy/case/case_what.html.
- [2] Chessman J., Daniels J.: Komponenty w UML, Wydawnictwa Naukowo-Techniczne 2004.
- [3] OMG, Diagram Interchange version 1.0, OMG, April 2006, <http://www.omg.org/cgi-bin/apps/doc?formal/06-04-04.pdf>.
- [4] Gratkowski T.: Automatyczne generowanie artefaktów wyjściowych dla czynności określenia interfejsów i operacji systemowych w metodzie Cheesmana-Danielsa z wykorzystaniem systemu ekspertowego, Computer Methods and Systems - CMS '05, Oprogramowanie Naukowo-Techniczne 2005 - Vol. 2, 483–488.
- [5] Gratkowski T.: Formalizacja kroków dla czynności utworzenia modeli typów z metodyki Cheesmana-Danielsa na potrzeby systemu regułowego, Pomiary Automatyka Kontrola - 2006 nr 6, 36–38.
- [6] Gratkowski T.: Cykl życia projektu informatycznego, Pierwsza Konferencja Naukowa 'Informatyka - sztuka czy rzemiosło', Oficyna Wydawnicza Uniwersytetu Zielonogóskiego 2004, 113–118.
- [7] OMG, Software Process Engineering Metamodel v1.1, OMG, January 2005, <http://www.omg.org/cgi-bin/apps/doc?formal/05-01-06.pdf>.
- [8] OMG, UML: Superstructure version 2.0, OMG, August 2005, <http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04>.
- [9] OMG, XML Metadata Interchange (XMI) version 1.1, OMG, November 2000, <http://www.omg.org/cgi-bin/apps/doc?formal/00-11-02.pdf>.
- [10] OMG, XML Metadata Interchange, OMG, September 2005, <http://www.omg.org/cgi-bin/apps/doc?formal/05-09-01.pdf>.