Automatyzacja w prognozowaniu jakości oprogramowania

Anna Bobkowska
Politechnika Gdańska

Streszczenie

Jest hipoteza mówiąca, że poprawę jakości oprogramowania i skrócenie czasu jego wytworzenia można uzyskać dzięki zastosowaniu narzędzi CASE, które wykonają niektóre zadania automatycznie lub przynajmniej będą je inteligentnie wspomagać. Czy jednak jest to możliwe tak niepowtarzalnym przedsięwzięciu jakim jest projekt informatyczny?

W referacie zaprezentowano wyniki badań z zakresu automatyzacji prognozowania jakości we wczesnych fazach wytwarzania oprogramowania. Omówiono zagadnienia związane z możliwością automatyzacji, przedstawiono wizję systemu inteligentnego wspomagania prognozowania jakości i prototyp systemu obliczającego metryki prognozowanej jakości oraz zasygnalizowano podobne prace z tego obszaru.

1. Wstęp

Wytwarzane współcześnie oprogramowania jest coraz bardziej złożone. Jednocześnie obserwowane są dążenia do obniżenia kosztów projektu, skrócenia czasu wykonania i wzrostu jakości rozwiązania. Aby sprostać tym wyzwaniom do wspomagania procesu wytwarzania wykorzystywane są narzędzia CASE (ang. Computer Aided Software Engineering). Niewątpliwą zaletą stosowania narzędzi CASE w firmie jest poprawa jakości dokumentacji i komunikacji w zespole, jednakże zazwyczaj oczekiwania użytkowników są większe niż usługi oferowane przez konkretne produkty, a uzyskiwane w wyniku ich zastosowania zyski czasowe nie są satysfakcjonujące [13]. Powstaje więc pytanie o możliwość automatyzacji zadań na poszczególnych etapach procesu wytwarzania oprogramowania lub przynajmniej inteligentnego ich wspomagania. Przykładem

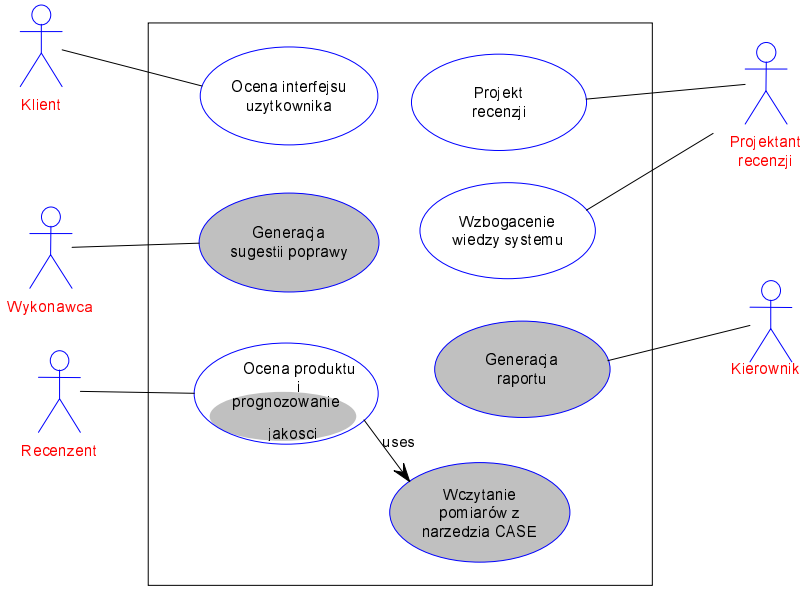
może być dążenie do automatycznej generacji kodu przez projektanta operującego na poziomie dziedziny problemu, sprawdzanie spójności modelu wykonanego w UML, generacja dokumentacji na podstawie modelu, automatyczne wykrywanie określonych typów defektów w kodzie itp. Istnieją nawet propozycje włączenia do narzędzi CASE elementów sztucznej inteligencji [1], takich jak bazy wiedzy, przetwarzanie języka naturalnego i pojęć opartych o pewne struktury, wnioskowanie w warunkach niepewności, modelowanie i budowanie aplikacji z inteligentnych agentów, tzn. obiektów posiadających nie tylko atrybuty i zachowanie, ale także cele i strategie działania oraz pewną znajomość otoczenia lub nawet możliwość uczenia się.

Prezentowane w referacie badania miały na celu analizę możliwości automatyzacji prognozowania jakości oprogramowania. Prognozowanie jakości oprogramowania dotyczy określania jakości oprogramowania w trakcie jego wytwarzania, tzn. wtedy, kiedy nie jest dostępny kod, ale dokumentacja systemu z zastosowaniem notacji UML [14]. Takie postępowanie ma na celu zapewnienie jakości już w fazach analizy i projektowania poprzez pokazanie konsekwencji jakościowych konkretnych rozwiązań oraz ich poprawę jak najniższym kosztem. Gromadzenie wskaźników prognozowanej jakości w kolejnych iteracjach poszczególnych etapów wytwarzania oprogramowania oraz innych wskaźników procesu wytwarzania umożliwia sterowanie jakością, a analiza tych danych może służyć zarządzaniu jakością i ciągłemu doskonaleniu procesu wytwarzania oprogramowania. Prognozowanie jakości na podstawie dokumentacji analizy i projektowania zwiększa prawdopodobieństwo wytworzenia produktu o wymaganej jakości.

2. Wizja systemu inteligentnego wspomagania prognozowania jakości

Idealnym rozwiązaniem byłoby zastąpienie w prognozowaniu recenzenta przez odpowiednią aplikację. W takim przypadku wykonawca po wykonaniu modeli uruchamiałby tą aplikację, która automatycznie znajdowałby defekty w jego modelu i pomagała mu je usunąć. Jednak całkowitej automatyzacji w ogólnym przypadku nie można wykonać ze względu na wymagane kompetencje i dużą różnorodność projektów. Na rysunku 1 przedstawiono diagram przypadków użycia dla systemu realnego wspomagania prognozowania jakości, a w tabelach 1 i 2 zawarto opis aktorów i przypadków użycia. Projektant recenzji wykonuje projekt recenzji, w czasie którego następuje integracja elementów dokumentacji i wymaganej jakości. Gdy wykonawca dostarczy produkty poszczególnych faz, klient może ocenić prototyp interfejsu, a recenzent może ocenić dokumentację projektową z zastosowaniem modeli UML. Po wykonaniu prognozowania jakości, która zawiera również zbieranie metryk diagramów i uwzględnienie ich w obliczeniach, wykonawca może wygenerować sugestie dotyczące poprawy

produktu, a kierownik projektu – raport przydatny do zarządzania projektem. Na diagramie na szaro zaznaczono miejsca potencjalnej automatyzacji. Nie można zautomatyzować zbierania danych o defektach i dokonywania ocen wyrażających trudne do zmierzenia aspekty, ale można automatycznie przekształcać pozyskane informacje (generacja raportu, generacja sugestii poprawy), prognozowanie jakości na podstawie zebranych danych i znajdowanie anomalii w projekcie na podstawie diagramów i wiedzy zawartej w systemie.



Rysunek 1. Diagram przypadków użycia systemu inteligentnego wspomagania prognozowania jakości

Tabela 1. Opis aktorów przedstawionych na rysunku 1.

Aktor	Opis
Klient	Użytkownik dostarczający oceny prototypu interfejsu w zakresie użyteczności, łatwości użycia i satysfakcji.
Wykonawca	Użytkownik wytwarzający system, który oczekuje sugestii dotyczących poprawy jakości dokumentacji.
Projektant recenzji	Użytkownik odpowiedzialny za projekt recenzji w zakresie określenia zakresu zbieranych danych oraz wewnętrznej integracji informacji z różnych źródeł (klient, poszczególni recenzenci, wewnętrzna wiedza systemu).
Recenzent	Użytkownik dokonujący recenzji, którego zadaniem jest znalezienie defektów i dokonanie oceny w określonym zakresie. Jego czynności są wspierane przez szablon oceny produktu. Recenzent może podać sugestie dotyczące poprawy. Recenzenci posiadający różną specjalizację mogą pracować nad innym zakresem dokumentacji i być ukierunkowane na inne defekty.

Kierownik	Użytkownik odpowiedzialny za organizację: planowanie projektu, śledzenie i zapewnienie zasobów. Jest on zainteresowany krótkim raportem o postępach w pracy nad projektem i jakością produktów.

Tabela 2. Opis przypadków użycia pokazanych na rysunku 1.

Przypadek użycia Aktor	Opis
Ocena interfejsu użytkownika Klient	Klient otrzymuje ankietę z prośbą o wyrażenie opinii na temat interfejsu (prototypu lub ostatecznej wersji), które będą pomocne w nanoszeniu korekt, poprawek i zapewnianiu satysfakcji klienta.
Generacja sugestii poprawy Wykonawca	Zwraca wykonawcy listę potencjalnych defektów znalezionych podczas przeglądu jego produktu z sugestiami ich usunięcia. Pozwala także wykonawcy na skomentowanie ich.
Ocena produktu i prognozowanie jakości Recenzent	Wspomaga recenzenta w zbieraniu danych przez dostarczenie zbioru szablonów pomocnych w znajdowaniu defektów i dokonywaniu ocen. Umożliwia wywołanie części prognozowania związanej z wnioskowaniem. *Wykorzystuje:* wczytanie danych z narzędzia CASE
Wczytanie pomiarów z narzędzia CASE	Pobiera metryki z narzędzia CASE. Może obliczać złożone metryki oraz na podstawie ich i wartości oczekiwanych wskazywać na anomalie w projekcie.
Projekt recenzji Projektant recenzji	Dostarcza zbioru szablonów projektantowi recenzji w celu ułatwienia przygotowania recenzji dokumentacji i oceny prototypu przez klienta, integracji oczekiwanych rezultatów z wbudowaną wiedzą oraz zdefiniowania formatu sugestii dla wykonawcy oraz formatu raportu dla kierownictwa.
Wzbogacenie wiedzy systemu Projektant recenzji	Umożliwia wprowadzenie wiedzy systemu i pozyskania wiedzy z danych z projektów poprzednich. Wiedza systemu obejmuje wartości oczekiwane metryk diagramów, reguły wnioskowani na temat jakości, heurystyki projektowe, wzorce i anty-wzorce projektowe, reguły generowania sugestii i reguły integracji informacji pochodzących z różnych źródeł. Mogą one pochodzić z literatury, od lokalnych ekspertów, z poprzednich projektów itp.
Generacja raportu Kierownik	Dostarcza krótkiego raportu dla kierownictwa dotyczącego postępów i jakości jak również wspierającego organizację projektu. Atrybutami bezpośrednio powiązanymi z prognozowaniem są jakość, czas i zasoby.

4. Prototyp systemu obliczającego metryki prognozowanej jakości

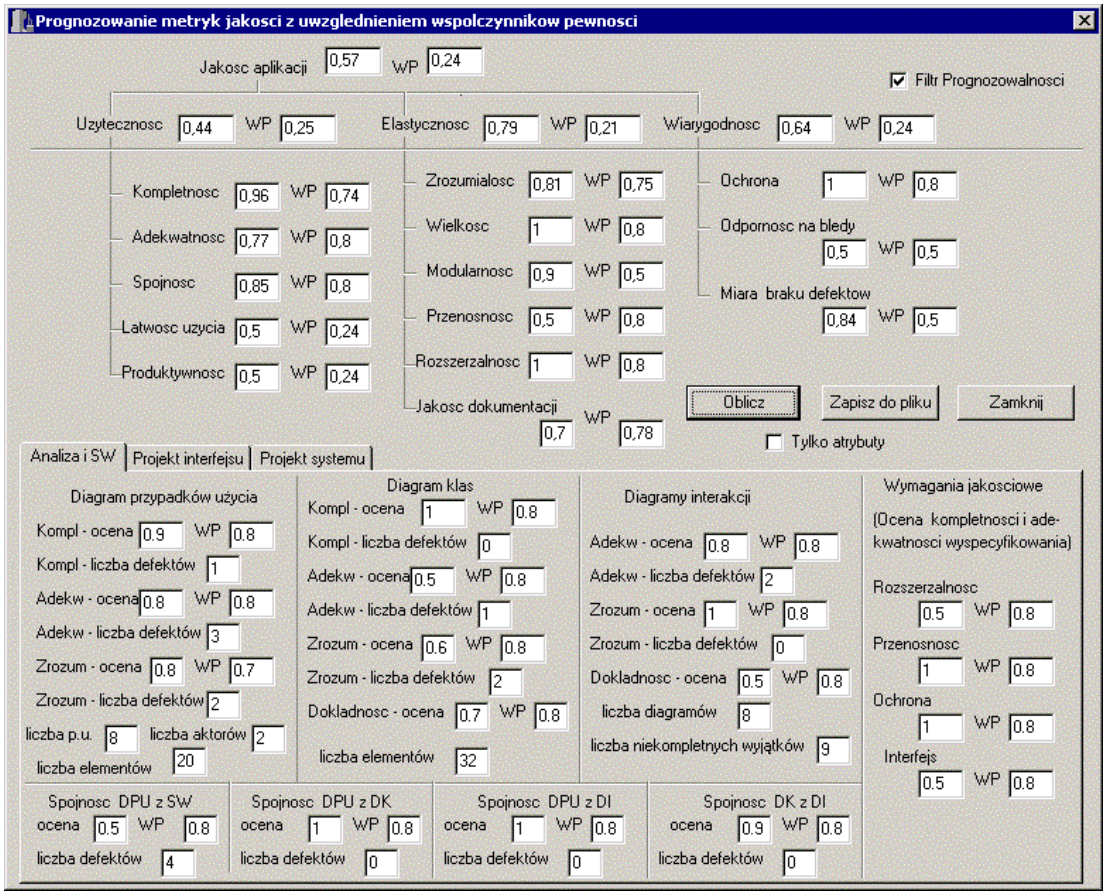
Zakres funkcjonalności prototypu obejmuje przypadek użycia ‘Ocena produktu i prognozowanie jakości’. Udostępnione są następujące usługi:

- Obliczanie metryk charakterystyk i atrybutów jakości w oparciu o elementarne metryki jakości wprowadzone w fazach rewizji analizy, projektu interfejsu użytkownika i projektu systemu;
- Zmiany parametrów w zakresie uwzględnienia wyników z poprzednich faz oraz uwzględnienia prognozowalności;
- Obliczanie samych metryk atrybutów i jakości aplikacji na podstawie metryk określających charakterystyki jakości;
- Zapis metryk jakości dotyczących charakterystyk i atrybutów do pliku.

Podstawą budowy prototypu był model prognozowania i poprawy jakości, który uwzględniał metryki diagramów, oceny i defekty. Tutaj zostały one poddane one kwantyfikacji i uwzględniono niepewność wnioskowania w postaci współczynników pewności (WP). Do przekształcania informacji jakościowej wykorzystano modele przekształceń zaproponowane w [4]. Założono kwantyfikację w przedziale [0..1] i współczynniki pewności również w przedziale [0..1]. Metryki diagramów są odniesione do wartości oczekiwanych i na tej podstawie obliczana jest elementarna metryka jakości związana z adekwatnością poszczególnych aspektów diagramu. Wartości oczekiwane zostały obliczone na podstawie podobnych projektów [3]. Oceny, które są związane z subiektywnymi aspektami albo trudnymi do precyzyjnego zmierzenia występują jako liczby z przedziału [0..1]. Jak elementarne metryki jakości traktowane są także lokalne metryki defektów, tzn. metryki wskazujące liczbę określonego typu defektów, które są później podzielone przez lokalną wielkość.

Interfejs użytkownika prototypu systemu obliczającego prognozowane metryki jakości dla fazy analizy i specyfikacji wymagań przedstawiono na rysunku 2. Zawarte tam dane pochodzą z projektu studenckiego pt. „Rezerwacja miejsc w hotelu przez Internet”. Dla każdej z faz pokazane jest drzewo jakości (będące rozszerzeniem struktury FCM [7]) oraz elementarne dane jakościowe uwzględniane w danej fazie. Możliwość uwzględnienia prognozowalności występuje jako ‘Filtr prognozowalności’, a w fazach projektu interfejsu i projektu systemu jest możliwość uwzględnienia wyników z poprzednich faz poprzez zaznaczenie ‘Łącznie z poprzednimi fazami’. Główną korzyścią wynikającą z zastosowania systemu w porównaniu z wnioskowaniem intuicyjnym jest skrócenie czasu związanego z obliczaniem metryk jakości oraz eliminacja subiektywizmu związanego z indywidualnym wnioskowaniem. Prototyp dostarcza informacji o wskaźnikach jakości i dokładności rezultatów oraz miejsca zwiększonego ryzyka występowania defektów wpływającego na obniżenie jakości

systemu, jednakże jest potrzeba ich odpowiedniego ich zinterpretowania, np. przez odniesienie do progów akceptowalności lub progów prognozowalności.



Rysunek 2. Interfejs użytkownika prototypu systemu obliczającego metryki prognozowanej jakości dla fazy analizy i specyfikacji wymagań.

3. Podobne prace

W literaturze, można znaleźć opis szeregu badań mających na celu zautomatyzowanie znajdowania defektów w modelach obiektowych. Jedną z pierwszych pozycji była praca [10] badająca progi wartości metryk w celu wykrycia anomalii i zidentyfikowania problemu. Z każdym zidentyfikowanym problemem skojarzona jest akcja sugerowanych w celu poprawy. Autorzy dla każdej metryki zdefiniowali następujący szablon:

- statystyczne pomiary ze zbadanych projektów,
- znaczenie metryki - opis, co logicznie wynika ze zmierzenia metryki,
- podobne metryki - mierzą podobne czynniki albo uzupełniające się,

- progi - zakresy pożądanych wartości metryk, które służą do wykrycia anomalii i tym samym wskazują na istnienie problemów,
- sugerowane akcje w przypadku identyfikacji problemu, przy czym decyzja o zastosowaniu sugerowanych akcji należy do projektanta.

Metryki produktów dotyczą głównie diagramu klas. Podobne metryki dla diagramu przypadków użycia zostały zaproponowane w pracy [5].

Ocena jakości modeli UML bazująca na metodzie GQM [2] zaprezentowana została w pracy [12]. Celem badań jest pomiar jakości i poprawa produktów. Metryki mają wskazywać miejsca zwiększonego ryzyka. Model zawiera metryki, progi i heurystyki. Autor twierdzi, że trudno jest znaleźć metryki interpretujące heurystyki. Należy rozważać metryki w kontekście, np. języka programowania, platformy, architektury, używanych mechanizmów, więc nie ma uniwersalnych progów (chociażby biorąc pod uwagę różnice pomiędzy komponentami 'serwera' i 'klienta'). Heurystyki znajdują nie defekty, ale najbardziej zagrożone fragmenty, które wymagają poświecenia im dodatkowego czasu. Ostatnio opublikowana została także praca [15], w której autorzy starają się określać jakość na podstawie diagramów UML z fazy projektu. Poszukują oni bezpośrednich wskaźników jakości oraz starają się wnioskować o jakości na podstawie wzorców projektowych oraz anty-wzorców wskazujących na defekty.

Propozycję automatyzacji pomiarów i ocen w oparciu o bazę wiedzy zaprezentowano w pracy [8]. Celem systemu jest wspomaganie projektantów w obiektywnym wnioskowaniu na temat jakości produktów projektowania i wyboru poszczególnych konstrukcji projektowych, które odbywa się poprzez zautomatyzowane przeglądy w zakresie wskazywania potencjalnych defektów i podawania propozycji, jak rozpoznane defekty mogą być usunięte. Badania polegają na skatalogowaniu wiedzy dostępnej w literaturze o projektowaniu obiektowym, zaleceń stosowanych w przedsiębiorstwach oraz osobistych doświadczeń i sformalizowaniu tej wiedzy o projektach w postaci heurystyk projektowych, reguł przekształcających w lepsze rozwiązania i pomiarów. Następnym etapem jest integracja wyników z efektywną metodą wytwarzania oprogramowania. W tym modelu występują następujące elementy:

- heurystyki - zdroworozsądkowe reguły poprawności projektu, wśród których wyróżnione są heurystyki sprawdzalne, sugerowane heurystyki (czasami nie wiadomo, jaka wartość jest optymalna) oraz niesprawdzalne automatycznie.
- reguły przekształcające fragment projektu - zalecenia jak wyeliminować potencjalne błędy, dla każdej heurystyki są dobrane inne reguły przekształcające;
- pomiary - które pomagają rozwiązywać konflikty tzn. dają możliwość porównania schematów przed i po wprowadzeniu zmian.

Prowadzone są także prace oparte na modelu SPIN. Praca [9] zawiera opis badań związanych z weryfikacją zachowania na podstawie rozszerzonych diagramów stanów (ang. statechart), a w przypadku znalezienia defektów

generuje diagramów sekwencji pokazujących jak znaleźć taki błąd. Weryfikacja poprawności protokołów uwierzytelniania oraz określonych własności poprawności (np. zakleszczenia) dla różnych architektur systemów rozproszonych zostały opisane w pracy [6].

Trwają także prace [11] nad zbudowaniem systemu uczącego się oceniać automatycznie poszczególne decyzje projektowe pod względem adaptowalności i wielokrotnego użycia oraz nad systemem potwierdzania jakości z użyciem automatycznych agentów, który zajmuje się potwierdzaniem prognozowalności, łatwości utrzymania, testowalności, sterowalności, obserwowalności i automatycznej naprawialności (ang. survivability).

4. Podsumowanie

W referacie przeanalizowano możliwość zastosowania automatyzacji w prognozowaniu jakości oprogramowania we wczesnych fazach wytwarzania oprogramowania. Badania pokazują, że całkowita automatyzacja nie jest możliwa ze względu na wymagane kompetencje potrzebne do oceny subiektywnych aspektów dokumentacji oraz znajdowania defektów. Jednakże poszczególne techniki bazujące na wzorcach oraz na danych statystycznych z poprzednich projektów mogą wskazywać miejsca anomalii w dokumentacji, które pokazują miejsca zwiększonego ryzyka występowania defektów. Automatyzacja natomiast może byś zastosowania do przetwarzania zgromadzonej informacji o jakości. Generacja raportów dla wykonawcy powinna pozwalać na poprawę jakości dokumentacji, a generacja raportów dla kierownika projektu powinna zawierać wskaźniki jakości przydatne do zarządzania projektem.

Zaprezentowano także prototyp systemu obliczającego metryki prognozowanej jakości na podstawie elementarnych metryk jakości związanych z modelem prognozowania jakości zawierającym metryki diagramów, oceny i defekty. Metryki defektów oraz oceny są wprowadzane do systemu, natomiast metryki diagramów są przekształcane automatycznie z wykorzystaniem wartości oczekiwanych wyznaczonych na podstawie podobnych projektów. Do przekształcania na metryki charakterystyk jakości wykorzystano badania nad kwantyfikacją jakości z uwzględnieniem niepewności prognozowania. Korzyścią wynikającą z zastosowania systemu w porównaniu z wnioskowaniem intuicyjnym jest skrócenie czasu związanego z obliczaniem metryk jakości oraz eliminacja subiektywizmu związanego z indywidualnym wnioskowaniem. W ramach dalszych prac planowane jest rozszerzenie prototypu o inne metody automatycznego wykrywania anomalii i defektów w dokumentacji projektowej.

Bibliografia

[1]. *Automated Software Engineering*, internet: www.cs.vassar.edu/faculty/welty/ase/

[2]. V.R. Basili, G. Caldiera, D.H. Rombach. The Goal Question Metric Approach, *ISERN* 1996.

[3]. A. Bobkowska. Quantitative and qualitative methods in process improvement and product quality assessment., *Proceedings of ESCOM 2001*.

[4]. A. Bobkowska. Kwantyfikacja prognozowaniej jakości oprogramowania, *Materiały konferencyjne KKIO'2000*.

[5]. A. Bobkowska, Quality Indicators Related to Use Case Model, *Proceedings of ACS'99*.

[6]. P. Bose. Automated Translation of UML Models of Architectures for Verification and Simulation using SPIN, *Proceedings of the 14th IEEE Conference on Automated Software Engineering,* 1998.

[7]. N.E. Fenton. *Software Metrics. A Rigorous Approach*, Chapman&Hall, 1993.

[8]. T. Grotehen, K.R. Dittrich. *The MeTHOOD Approach: Measures, Transformation Rules, and Heuristics for Object-Oriented Design*, Technical Report., internet: http://www.ifi.unizh.ch/groups/dbtg/MeTHOOD/index.html

[9]. J. Lilius, I.P. Paltor. vUML: A Tool for Verifying UML Models, *Proceedings of the 14th IEEE Conference on Automated Software Engineering,* 1998.

[10]. M. Lorenz, J. Kidd. *Object-oriented Software Metrics. A Practical Guide.*, Prentice Hall, 1994.

[11]. T. Menzies. HQkb - *The High Quality Knowledge Base Initiative*, internet: http://research.ivv.nasa.gov/docs/techreports.

[12]. B. Lyons. Partner Perspective: Quantifying Quality in Your OO Models, *Rose Architect,* volume 1, 1998.

[13]. M. Piechówka, S. Szejko, CASE jaki jest, każdy widzi., *Informatyka,* numer 4, 1998.

[14]. Rational Software Corporation, *UML v.1.3.*, internet: www.rational.com

[15]. A.I. Verkamo, J. Gustafsson, L. Nenonen, J. Paakki. Measuring design diagrams for product quality evaluation, *Proceedings of ESCOM 2001*.