Predykcyjno-przyrostowa strategia rozwoju informatycznych aplikacji dedykowanych

Rafał Knopa
Henryk Krawczyk
Kruk Sebastian
Adam Mazurkiewicz
Jan Zieliński
Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

Streszczenie

Przedstawiono doświadczenie związane z budową i rozbudową trzech kolejnych wersji aplikacji informatycznej przeznaczonej dla zastosowań medycznych. Scharakteryzowano przyrostowy model rozwoju aplikacji i skomentowano jego przedatność przy opracowaniu kolejnych wersji na podstawie prognozowanych wymagań. Opisano również finalną architekturę aplikacji jak i jej interfejs z użytkownikiem.

1. Wprowadzenie

Wytwarzanie oprogramowania jest działalnością intelektualną wymagającą różnorakiej wiedzy, w tym co najmniej pochodzącej z dwóch różnych obszarów takich jak:

1. metodologie i narzędzia wytwarzania oprogramowania,
2. specyfiki dziedziny rozwiązywanego problemu.

W wielu podejściach zakłada się, żę metodologia wytwarzania oprogramowania jest na tyle uniwersalna, że można ją wykorzystać do budowy każdej aplikacji [2], [7]. Wyróżnia się kilka modeli cyklu życia (kaskadowy, V, czy spiralny) i precyzuje się zadania dla każdego z etapów całego procesu wytwarzania. Zakłada się, że idealne ich wykonanie prowadzi do sukcesu, co oznacza wytworzenie produktu o oczekiwanych przez użytkownika własnościach. Z praktyki informatycznej wynika jednak, że specyfika dziedziny aplikacji wymaga często podejścia zorientowanego na tę dziedzinę. Stąd też coraz popularniejsze stają się różnego typu wzorce projektowe [9], określone na podstawie konkretnych aplikacji, które są często wykorzystywane i które uzyskały szeroką akceptację. Z uwagi na częstą zmianę technologii, a także modyfikację wymagań użytkownika, mniej istotne staje się dopasowanie architektury systemu do konkretnych (często chwilowych) wymagań, a raczej konstrukcja aplikacji o dużej elastyczności, w znacznie krótszym czasie, które łatwo poddają się potem różnego typu modyfikacjom.

Ogólne rozwiązania wychodzące na przeciw tym ostatnim tendencjom sprowadzają się do modyfikacji poprzednio opracowanych modeli cyklu życia. Przykładem jest strategia małych zespołów realizujących równolegle swoje zadania w sposób iteracyjny lub metodą przyrostową (incremental or iterative product development) [4]. Innym podejściem jest rozwój wielu kolejnych wersji, metodą ewolucyjną (version staged model) [8]. Jeszcze innym znanym podejściem tzw. negocjacyjnym jest rozbudowa modelu spiralnego (WinWin spiral model) [3] o dodatkowe fazy, co umożliwia lepszą koncentrację na realizacji prostszych zadań. Przyrostowe podejście do budowy aplikacji oznacza często składanie jej z kolejno oprowacowywanych komponentów. Złożone aplikacje, integrujące pracę różnych systemów podano w pracy [1]. W tym przypadku szczególną uwagę zwraca się na różny sposób spostrzegania aplikacji przez udziałowców, klientów, architektów, wykonawców i osoby odpowiedzialne za utrzymanie. Dwa pierwsze wyżej wymienione podejścia dotyczyły raczej aplikacji powszechnego stosowania, zaś dwa ostatnie związane były z budową aplikacji dedykowanych, zorientowanych na konkretne dziedziny zastosowań, takie jak aplikacje multimedialne, czy handel elektroniczny.

W pracy przedstawiono budowę komputerowego systemu medycznego wspomagającego rejestrację badań endoskopowych i wspomaganie rozpoznawania chorób gastroenterologicznych. System ten jest rozwijany w Katedrze od 1997 roku. Przedstawiono przyjętą strategię wytwarzania, która wykorzystuje zarówno elementy modelu wersjowego oraz przyrostowego. Co więcej zakłada się predykcje zmiany wymagań. W rozdziale drugim przedstawiono sposób rozwoju kolejnych wersji systemu. W dwóch następnych rozdziałach opisano wersję finalną architektury systemu i interfesju użytkownika. We wnioskach końcowych zwrócono uwagę na konieczność dalszej rozbudowy o specjalizowane funkcje rekomendacji, która dzięki elastycznej architekturze systemu jest już

Tablica 1: Charakterysytka opracowanych wersji aplikacji

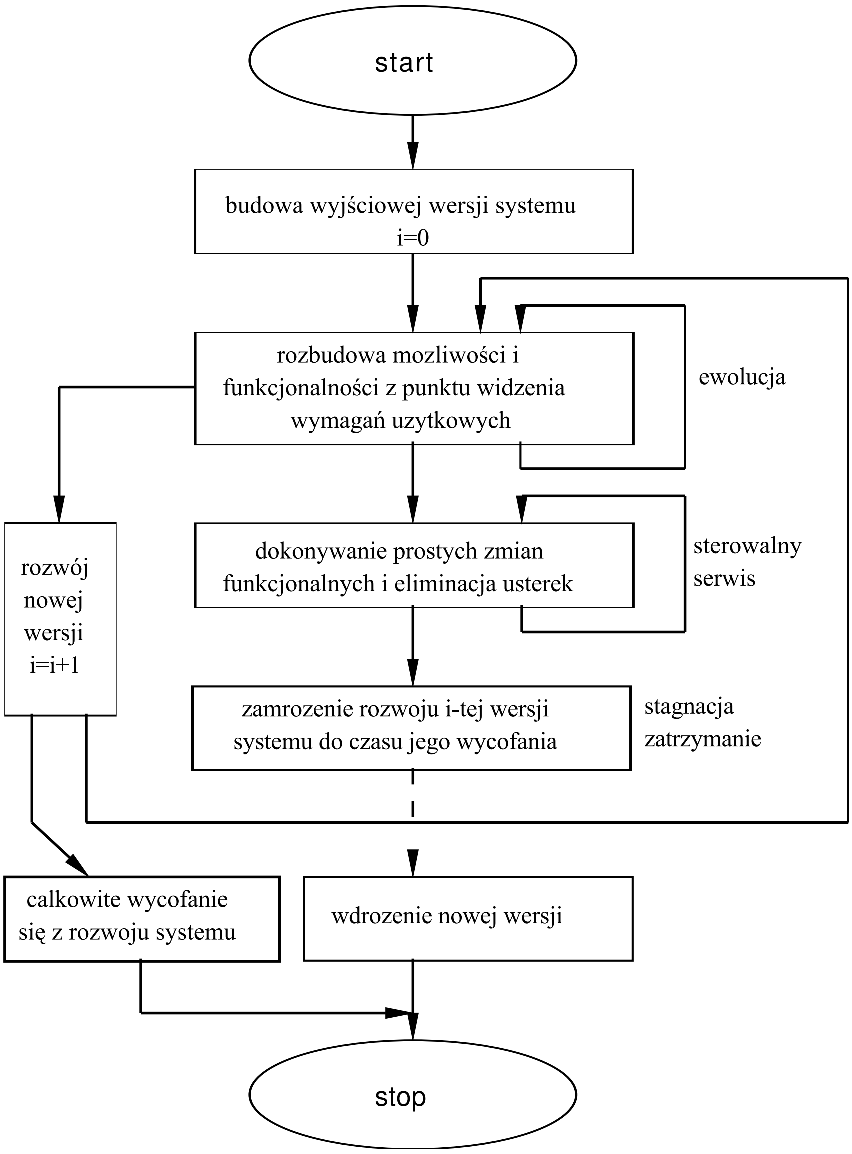
Wersja, rok i czas prod.	Funkcje	Platforma implement.	Koncepcja intefejsu	Uwagi
1, 1997, 2 lata	rejestracja danych pacjenta, rejestracja wyników dla 6 rodz. badań, przegląd i modyfikacja danych, twozenie i drukowanie raportów i statystyk	MS-DOS, Clipper	ukierunkowany na wyodrębnione typy badań; każda akcja ma ściśle określony scenariusz	system zamknięty funkcjonalnie
2, 1999, 1 rok	rejestracja i obróbka zdjęć endoskopowych, konwersja danych do formatu DICOM, porównywanie zdjęć w oparciu o histogramy	MsWindows 98, Java JDK1.2, MsSQL 7.0	ukierunkowany na dowolne badanie pacjenta; możliwość obróbki filmu i zdjęć z badania; koncepcja prostych pluginów porównywania obrazów; rekomendacja zdjęć oparta na jednym algorytmie	-
3, 2001, 6 m-cy	standaryzacja opisu badania wg MST, mechanizm rekomendacji wykorzystujący równoległe algorytmy rozpoznawania obrazów, definicja interfejsów komunikacji przez Internet, mechanizm opisu zdjęć wzorcowych	MsWindows 98/2000, Java JDK1.3, JPVM, PVM, RMI, MySQL 3.23	baza wzorców dostępna w Internecie; rozwinięcie technologii pluginów i zaawansowania rekomendacji; wiele funkcji zespolonych z technologiami Internetowymi (np. zestawienia, raporty)	wyodrębnienie dobrze zdefiniowanych komponentów, środowisko zintegrowane i rozproszone

prosta do zrealizowania.

2. Hybrydowe podejście do budowy oprogramowania

Projektowana aplikacja dla zastosowań medycznych jest aplikacją dedykowaną, gdzie przyszły użytkownik (lekarz) nie jest specjalistą z technologii informatycznej, dlatego trudno jest mu przedstawić szczegółowe wymagania na tego typu aplikację. Spodziewa się on podstawowych korzyści: odciążenia go od pracy administracyjnej, w tym pomoc przy wypełnianiu dokumentów badań oraz sporządzania dodatkowych sprawozdań. Z uwagi na to, że odbiorcą systemu jest Klinika Akademii Medycznej, wspomina się również o pewnych funkcjach dydaktyczno-szkoleniowych. W miarę jednak eksploatacji systemu, użytkownik zauważa szczegółowe problemy do rozwiązania, na przykład zapewnienie wspomagania pewnych zadań badawczych, czy współpracę tego typu systemów dla wielu klinik. Dzięki temu cały projekt jak i aplikacja ulegają z czasem całkowitej przebudowie.

W tablicy 1 przedstawiono opracowane warianty aplikacji ERS (Endoscopy



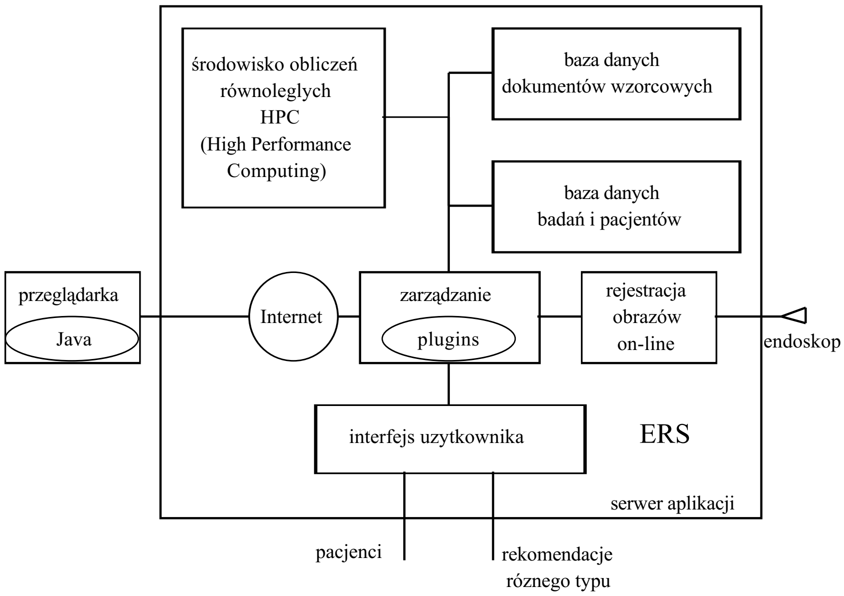
Rysunek 1: Koncepcja modelu predykcyjno-przyrostowego rozwoju oprogramowania

Recommender System [8]). Pierwsza wersja była realizowana w tradycyjnym podejściu związanym z modelem kaskadowym. Następne zostały zrealizowane po dwuletniej eksploatacji pierwszej wersji, przy czym przy ich rozwoju wykorzystano podejście predykcyjno-przyrostowe. Oznacza ono przyjęcie stopniowego rozwoju aplikacji, gdzie kolejno rozbudowuje się jej komponenty w oparcji o dobrze określone fragmenty funkcjonalne (tzw. przyrosty), zaś wizję tych fragmentów opracowuje się równolegle do istniejącej wersji.

Na rys. 1 przedstawiono uproszczony schemat postępowania. Rozwój nowej wersji następuje równolegle z przekazaniem do eksploatacji wersji poprzedniej, zaś wersja bieżąca powinna być ukończona przed wycofaniem z pracy wersji poprzedniej. W przedstawionym przypadku budowa każdej nowej wersji o znacznie lepszych możliwościach funkcjonalnych realizowana była znacznie krócej, mimo, że zespoły ją realizące były inne. Podstawą takiego podejścia jest nie tylko zmiana technologii informacyjnej (z DOS + Clipper na Javę), ale również przyjęcie elastycznej architektury aplikacji. Złożona jest ona z dobrze określonych komponentów, które przy założonych odpowiednich interfejsach mogą być niezależnie rozwijane. W wyniku takiego podejścia powstała nie tylko wymagana aplikacja, ale produkty dodatkowe (jak album zdjęć wzorcowych czy równoległe środowisko do analizy obrazów medycznych), których efekty przekraczają znacznie oczekiwane możliwości. Co istotne koszt i harmonogram realizacji nie był przekroczony.

3. Architektura systemu ERS

System ERS [6] słada się z wielu komponentów. Na rys. 2 przedstawiono schemat architektury systemu. W kolejnych podrozdziałach opisano wybrane funkcje tej aplikacji.



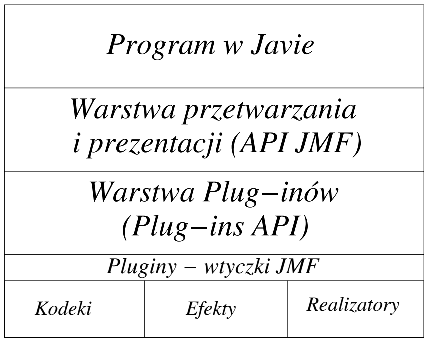
Rysunek 2: Architektura systemu ERS

3.1. Przechwytywanie obrazów

Pacjent poddawany jest badaniu za pomocą endoskopu. Dane z tego urządzenia przesyłane są poprzez złącze video karty graficznej komputera. Obecnie używane są karty niemieckiej firmy Elsa, gwarantujące wysoką jakość procesu przetwarzania grafiki. Proces przechwytywania kontrolowany jest przez klasy pakietu `capture`. Te z kolei wykorzystywane są przez obiekty dostępu do bazy danych (agentów baz danych) w celu pozyskania danych medialnych, które zapisywane są w bazie danych systemu. Do bazy trafiają także podstawowe dane o pacjencie a także informacje pozyskane w trakcie wywiadu. Taka architektura umożliwia dostosowanie struktury bazy danych do indywidualnych potrzeb każdej implementacji systemu bez potrzeby ingerencji w kod pakietu `capture`. Po przeprowadzeniu badania i ewentualnej rekomendacji, zapisy w bazie uzupełniane są szczegółowym opisem badania, zgodnym z normą `MST`, i wynikami procesu rekomendacyjnego.

Proces przechwytywania obrazów łączy w sobie zagadnienia z dziedziny optyki, elektroniki i informatyki. Dodatkowo, komplikuje go fakt wielkiej różnorodności rozwiązań sprzętowych i programowych. Jeżeli na dodatek odbywać ma się on w czasie rzeczywistym, to okazuje się, że trzeba zaangażować weń całkiem spore zasoby sprzętowe. Założeniem systemu ERS była dodatkowo niezależność od platformy, co uzyskano programując całość systemu w języku Java.

Dostrzegając ogólny problem operacji na danych multimedialnych, firma SUN stworzyła pakiet Java Media Framework (JMF), który pozwolił zbudować pomost pomiędzy wysokopoziomowymi klasami Javy a sprzętem multimedialnym. Jego ogólną architekturę przedstawia rysunek 3.



Rysunek 3: Ogólna architektura pakietu JMF

Program napisany w Javie, obojętnie czy jest to aplikacja, aplet, czy też komponent *JavaBeans*, korzysta z funkcji udostępnianych przez warstwę przetwarzania i prezentacji pakietu JMF. Autorzy pakietów odpowiedzialnych za przechwytywanie obrazów w systemie ERS rozważyli użycie kilku możliwych architektur dostępu do sprzętu wideo. Ostatecznie zdecydowano się na użycie standardowego JMF. Klasy pakietu `caputre` dostarczają niezbędnego do przeprowadzenia przechywytywania interfejsu użytkownika oraz ukrywają przed programistami innych częsci systemu szczegóły dostępu do klas JMF. Pakiet nie implementuje żadnych rozszerzeń jego funkcjonalności. Aby zapewnić odpowiednią wydajność procesu przechwytywania, niezbędnym okazało się użycie `Windows Performance Pack`. Jest to pakiet, który za cenę utraty niezależności od platformy, daje większą wydajność JMF.

3.2. Bazy danych

Dane przechowywane przez system ERS dzielą się na dwie grupy: dane jawne, czyli bazy danych obrazów wzorcowych wraz z ich opisem, oraz poufne, takie jak rekordy pacjenta oraz wyniki poszczególnych badań przeprowadzonych przy wsparciu ze strony systemu, a także wyniki rekomendacji. Obie te grupy łączy duży rozmiar danych oraz medyczna tematyka.

Wszystkie dane związane z badaniem zapisywane są w dedykowanej bazie, oznaczonej na rysunku 2 jako *Baza danych badań i pacjentów.* Zawiera ona nie tylko rekord pacjenta i wyniki przeprowadzonego z nim wywiadu, ale także szczegóły dotyczące badania, zlecającej go instytucji medycznej, lekarza je wykonywującego oraz uzyskane obrazy i/lub filmy. Dostęp do bazy danych może odbywać się za pośrednictwem agentów dostępu do baz danych, co daje większą elastyczność w doborze motoru (engine'u) bazy danych, lub bezpośrednio – poprzez odwołania do `JDBC (Java Data Base Connectivity)`.

Baza danych zawierająca wzorce jest ściśle związana ze standardem MST (Minimal Standard Terminology). Wzorce składają się z reprezentatywnych zdjęć i ich opisów. Schemat tej bazy danych jest trójpoziomowy. Najwyżej znajdują się tabele główne: zdjęcia i nagłówki dokumentów MST. Poniżej znajdują się tabele reprezentujące sekcje MST. Każdej sekcji odpowiada oddzielna tabela. Tabele sekcji zawierają kolumnę z identyfikatorem dokumentu, która wiąże dokument MST z zawartościami poszczególnych sekcji. Na najniższym poziomie znajdują się tabele słownikowe. Tabele słownikowe pozwalają uniezależnić tabele sekcji od opisów słownych. Istnieją trzy tabele słownikowe:

- tabela słów ze standardu MST – do niej są dopisywane słowa pochodzące z plików konfiguracyjnych MST,
- tabela opisów – zawiera opisy z części MST pozwalających na wprowadzanie swobodnych opisów,
- tabela jednostek chorobowych – zawiera określenia dla rozpoznań końcowych.

Ponieważ w tabelach sekcji dokumentów MST nie są umieszczane żadne opisy słowne, lecz identyfikatory do pól w tabelach słownikowych, tylko tabele słownikowe mogą zawierać opisy słowne. Takie podejście ułatwia przetwarzanie automatyczne, jak np. tłumaczenie na inne języki, porównywanie słów itd. Ponadto umożliwia to przyspieszenie dostępu do bazy danych (np. poprzez buforowanie w pamięci słowników) oraz upraszcza procedury korekty błędnie wprowadzonych globalnych sformułowań (np. błędów w plikach konfiguracyjnych).

Korzystanie z tak zbudowanej bazy danych jest wydajne. W tabelach prostych wymaga tylko pojedynczego odwołania się do tabeli. W tabelach złożonych pierwsze odwołanie pozwala uzyskać liczbę grup określeń, aby w kolejnych odwołaniach już uzyskiwać konkretne informacje. W przypadku klasycznej relacyjnej bazy danych należałoby wykonywać kolejne zapytania do oddzielnych tabel stworzonych dla poszczególnych określeń. Drugą niekwestionowaną zaletą jest to, że modyfikowacje struktury MST nie wpływają na strukturę bazy danych. Dzieje się tak dlatego, że struktura dokumentu MST nie jest wpisana w strukturę bazy danych, lecz stanowi część danych zapisanych w bazie danych

3.3. Proces rekomendacji

W tym rozdziale przedstawimy krok po kroku jak za pomocą systemu ERS możliwe jest wydanie diagnozy i podjęciu decyzji o dalszym leczeniu badanego pacjenta. Przed przystąpieniem do właściwego badania endoskopowego, należy przeprowadzić wywiad z pacjentem w celu określenia występujących objawów. Dane te po w prowadzeniu do dokumentu pacjenta posłużą w pierwszej części rekomendacji, podczas określania rozpoznania wstępnego.

Na podstawie informacji zawartych w karcie badania pacjenta, w części dotyczącej objawów, system ERS stara się znaleźć kilka najbardziej podobnych pod względem objawów dokumentów z bazy dokumentów wzorcowych. Zaproponowane przez system dokumenty wzorcowe mogą posłużyć lekarzowi w celu ustalenia prawdopodobnej choroby pacjenta. Informacje te mogą okazać się pomocne podczas określania charakteru dalszego badania pacjenta.

Po wykonaniu badania endoskopowego, system ERS w trzech krokach dokonuje rekomendacji wypełnienia pozostałej części karty badania pacjenta, a co za tym idzie, ustalenia diagnozy. Pierwsza część obejmuje rekomendacje jedynie na podstawie zdjęcia (zdjęć) endoskopowych wykonanych podczas badania. Druga i trzecia część korzystają dodatkowo z zawartości (częściowo wypełnionego) dokumentu pacjenta. W ostatnim kroku sugerowane są informacje wybrane na podstawie dwóch poprzednich oraz interakcji lekarza z nimi.

W pierwszym kroku system ERS dokonuje porównania zdjęcia (zdjęć) z badania endoskopowego ze zdjęciami wzorcowymi. W wyniku porównania przedstawiane zostaje lekarzowi maksymalnie pięć dokumentów wzorcowych o zdjęciach (dokładniej mówiąc — regionach zainteresowania ROI oznaczonych na tych zdjęciach) najbardziej zbliżonych do zdjęcia (zdjęć) z danego badania endoskopowego. Jeżeli lekarz dojdzie do wniosku, że jeden z zaprezentowanych dokumentów w sposób dostatecznie zbliżony opisuje obecne badanie, wówczas może on zawarte w nim informacje wykorzystać podczas wypełniania karty badania pacjenta i w tym momencie zakończyć proces rekomendacji. Jeżeli żaden z dokumentów nie opisuje dostatecznie dobrze obecnego badania, wówczas konieczne jest wypełnienie, choćby fragmentarycznie, karty badania pacjenta. Wówczas w drugim kroku rekomendacji system stara się zaprezentować maksymalnie pięć dokumentów o zawartości zbliżonej do tej, którą lekarz już wypełnił w karcie badania oraz zdjęciach zbliżonych do zdjęć znajdujących się w dokumentach wzorcowych. Podobnie jak poprzednio, lekarz ma wówczas możliwość wyboru jednego z dokumentów w celu wypełnienia zawartości karty badania pacjenta, i zakończenia procesu rekomendacji. W ostatnim kroku system dokonuje porównania zwróconych wcześniej dokumentów wzorcowych i prezentuje lekarzowi maksymalnie dwa, które najlepiej pasują do obecnego badania. Prezentowane dokumenty są ostatnimi proponowanymi przez system ERS. Na ich podstawie lekarz ma możliwość wypełnienia karty badania. Krok ten kończy jednocześnie proces rekomendacji. W momencie wyboru jednego z prezentowanych dokumentów wzorcowych jako referencyjnego do wypełnienia karty badania, system podaje również sugerowane procedury diagnostyczne i terapeutyczne oraz prawdopodobne powikłania, które mogą wystąpić.

4. Interfejs użytkownika

4.1. Użyteczność interfejsu

Zgodnie z normą ISO9241 część 11, użyteczność systemu zależy nie tylko od wyglądu interfejsu (postać przycisków), lecz również od sposobu interakcji użytkownika z systemem. Ta interakcja jest opisana za pomocą pięciu podstawowych atrybutów:

- łatwość nauczenia się interfejsu (czas nauki),
- efektywność reakcji systemu (liczba zadań możliwych do realizacji w jednostce czasu),
- zapamiętywanie interfejsu w czasie (możliwość korzystania z interfejsu po dłuższej przerwie),
- intensywność pojawiających się będów (np. obsługa zdarzeń wyjątkowych),

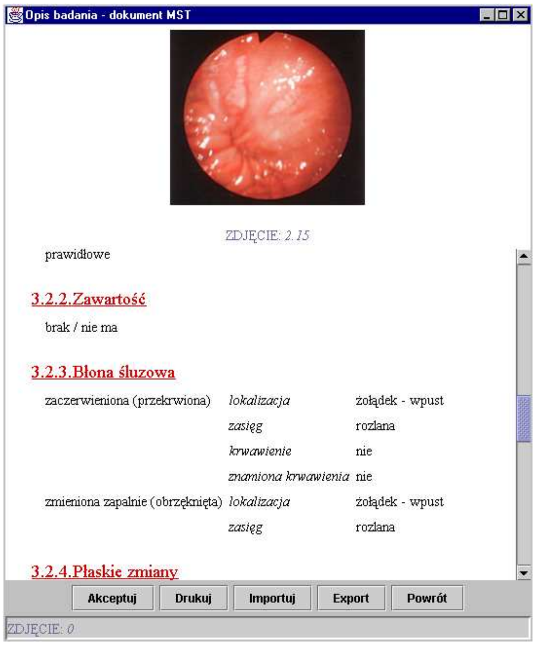
- szeroko rozumiana satysfakcja użytkownika (ogólne wrażenie o systemie).

Powyższe artybuty sa często sprzeczne, np. łatwość uczenia i efektywność. Użyteczność interfejsu jest istotna zwłaszcza w aplikacjach dedykowanych, gdzie użytkownikiem nie jest informatyk. W zaproponowanej metodzie predykcyjno-przyrostowej istnieje również możliwość iteracyjnego projektowania interfejsu wraz z oceną jego jakości przez użytkowników. Wraz z rozwojem funkcjonalnym całego systemu oceniana jest również sposób inicjalizacji wizualizacji wykonania zadań. To z kolei umożliwia sformułowanie nowych sugestii, co do dalszych zadań systemu. Oceną interfejsu dokonuje się na bazie jego szybkiego prototypowania wykorzystując dodatkowo pakiet SOJO [5] do dokumentowania i analizy porównawczej. Warto przy tym nadmienić, że analiza jakościowa w proponowanym modelu rozwoju oprogramowania jest bardzo istotna i dokładność prognozowania jakości w dużej mierze wpływa na proces wytwarzania, co w konsekwencji prowadzi do ulepszenia powstającego produktu. Sposób wykorzystania pakietu SOJO w takim ogólnym przypadku ilustruje cytowana już praca [5].

4.2. Rozwój interfejsu

W trakcie kolejnych faz rozwoju systemu funkcjonalność i wygląd interfejsu zmieniały się, aby zapewnić proste i efektywne wykorzystanie kolejnych funkcji systemu. Determinowane jest to również zmianami systemu operacyjnego i platformy programistycznej. Początkowy kształt interfejsu opierał się na dość ścisłym powiązaniu pól tekstowych i innych elementów operacji użytkownika ze strukturą bazy danych. Rejestracja obrazów z endoskopu do tej wersji systemu została dodana dużo później jako zupełnie osobny moduł i w praktyce nie była wykorzystywana w systemie. W kolejnej wersji, już zorientowanej na kompleksową rejestrację badania, funkcje związane z przechwytywaniem i szeroko rozumianymi modyfikacjami obrazu stały się integralną, ważną częścią systemu, pełniąc rolę elementu scenariusza tworzenia rekordu pacjenta. Rekord pacjenta jest tutaj koncepcją logiczną, ktora wiąże wszelkie dane związane z konkretnym pacjentem w jedną sensowną całość. Rdzeń koncepcji interfejsu użytkownika systemu ERS, dopracowanej i uzupełnionej w wersji trzeciej, opiera się właśnie na przyjaznym przeprowadzeniu operatora systemu przez wszystkie etapy tworzenia i zarządzania rekordem pacjenta. Główną nowością trzeciej wersji systemu jest wykorzystanie w opisie badania standardu MST. Poza tym trzecia wersja systemu sformalizowała korzystanie z rekomendacji jako elementu ułatwiającego tworzenie rekordu pacjenta, z wykorzystaniem określania ROI (fragment obrazu) oraz możliwością dokonywania rekomendacji w dowolnym momencie tworzenia opisu badania.

Standard MST, jest próbą usystematyzowania różnorodnych form opisu przebiegu badania endoskopowego. Jego ścisłe powiązania i hierarchiczna struktura umożliwiły stworzenie przyjaznego interfejsu dla rejestracji wyników konkretnego badania. Przy tym wyróżniono dwie części: pierwsza część, nadrzędna pokazuje aktualny wygląd dokumentu w formie gotowego do wydruku arkusza i, za pomocą odnośników kodu HTML, modyfikuje dynamicznie jego zawartość, wołając w razie potrzeby metody części drugiej. Druga część składa się z okienek dialogowych, które pozwalają podawać wartości dla kolejnych pól standardu MST i jednocześnie prowadzą operatora przez niejednoznaczności, które mogą się pojawić dla danego dokumentu. Wygląd dokumentu w czasie wypełniania pokazany jest na rysunku 4.



Rysunek 4: Wygląd dokumentu MST w trakcie wprowadzania danych

5. Wnioski końcowe

W pracy zaprezentowano strategię wytwarzania dedykowalnej aplikacji medycznej, która została rozbudowana z prostej sekwencyjnej bazodanowej aplikacji została rozbudowana w rozproszoną aplikację realizującą złożone obliczenia równoległe i dostępną z Internetu. Oczywiście taki rozwój nie byłby możliwy, przy uwzględnieniu tylko podstawowych wymagań użytkowych (tutaj lekarzy). W ten sposób realizowana pierwsza wersja już została wycofana z eksploatacji, z uwagi na małą użyteczność. Wersja druga akcentuje rozbudowę funkcji związanych z badaniem gastroenterologicznym pacjenta, w tym rejestrację obrazów endoskopowych na bieżąco. Wersja trzecia posiada duże możliwości użytkowe (przed wszystki dydadaktyczne), jest już dostosowana do współpracy z innymi systemami dzięki stosowaniu standardów MST i DICOM, ale nie jest wiarygodna co do decyzji medycznych generowanych automatycznie. Dlatego w przyszłym roku, po gruntownym przetestowaniu wersji trzeciej, będzie zrealizowana wersja docelowa zapewniająca skuteczną rekomendację. Jej wytworzenie polegać będzie na dopracowaniu odpowiednich *pluginów*, realizujących tego typu funkcje.

Literatura

[1] F. J. Armour, S. H. Kaisler, and S. Y. Liu. Building an enterprise architecture step by step. *IT Pro*, July/August:31–38, 1999.

[2] A. Behforaoz and F.J. Hudson. Software engineering fundamentals. Oxford University Press, New York, 1996.

[3] B. Boehm, A. Egyee, J. Kwan, D. Post, A. Shoh, and R. Madochy. Using the winwin spiral model: A case study. *IEEE Computer*, July:33–44, 1998.

[4] M.A. Cusumono and R.W. Selby. How microsoft builds software. *Comm. of the ACM*, June:53–61, 1997.

[5] P. Czarnul, H. Krawczyk, and A. Mazurkiewicz. Quality driven development methodology for network applications. In *Proceedings of the ISThmus Conference*. 2000.

[6] H. Krawczyk and R. Knopa. Web-based endoscopy recommender system – ers. In *Proc. of IEEE Conference on Parallel Computing in Elec. Engineering*, pages 257–261. Trois Rivieres, Quebec, Canada, 2000.

[7] H. Krawczyk and B. Wiszniewski. *Analysis and Testing Distributed Applications*. RSP Ltd, Baldock, Hertfordshire, England.

[8] V.T. Rajlich and K.H. Bennett. A staged model for the software life cycle. *IEEE Computer*, July:66–71, 2000.

[9] S. Svinivasan. Design patterns in object-oriented frameworks. *IEEE Computer*, Feb.:24–31, 1999.