

Zastosowanie podejścia ukierunkowanego na architekturę przy projektowaniu oprogramowania eksperymentalnego

T. GÓRSKI

e-mail: gorski@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

W artykule przedstawiono zastosowanie podejścia ukierunkowanego na architekturę systemu informatycznego w procesie projektowania oprogramowania eksperymentalnego do budowy modeli ścieżek klinicznych. Artykuł przedstawia także wyniki analizy zasadności zastosowania standardów BPMN, GELLO, UML, OCL, XML oraz HL7 w kontekście ich przydatności do modelowania architektury systemu informatycznego. Wybrany zestaw standardów oceniany jest pod kątem uzyskania pożądanych efektów budowy architektury systemu informatycznego oraz tworzenia modeli ścieżek klinicznych w oprogramowaniu eksperymentalnym.

Słowa kluczowe: architektura systemu informatycznego, zarządzanie wymaganiami

1. Zakładane efekty projektu

W założeniach projektu POIG.01.03.01-00-145/08 [5] przyjęto, że zostaną wykonane analiza i opracowanie zestawów standardów niezbędnych do budowy repozytorium wytycznych i ścieżek klinicznych. Ponadto, w wyniku projektu miał być utworzony model repozytorium wytycznych i ścieżek klinicznych umożliwiający:

- zdalne tworzenie, edycję i zarządzanie poszczególnymi wytycznymi przez wiele rozproszonych terytorialnie zespołów naukowców i praktyków
- tworzenie i publikowanie wytycznych w języku polskim.

Zasadniczym tematem projektu było badanie możliwości i sposobów automatyzacji tworzenia oraz dystrybucji krajowych wytycznych, a także opracowanie projektu standardów ich elektronicznej postaci.

W związku z tym istotne było odpowiednie skonfigurowanie procesu projektowania systemu informatycznego, który w swym charakterze jest oprogramowaniem eksperymentalnym. Ponadto, należało dobrać odpowiedni zestaw standardów modelowania architektury systemu informatycznego oraz zestaw standardów modelowania ścieżek klinicznych.

2. Konfiguracja procesu projektowania

W realizacji projektu POIG.01.03.01-00-145/08 zastosowano wybrany podzbiór dyscyplin metodyki firmy IBM Rational Unified Process (RUP). U podstaw RUP leży sześć kluczowych reguł projektowania systemów informatycznych ukierunkowanego na zastosowanie biznesowe:

- dostosowanie procesu
- bilansowanie przeciwstawnych priorytetów udziałowców
- współpraca między zespołami
- demonstrowanie wartości w sposób iteracyjny
- dostosowanie poziomu abstrakcji
- ciągłe skupienie na jakości.

Za regułami tymi kryją się podstawowe dla inżynierii oprogramowania praktyki:

- wytwarzaj iteracyjnie
- zarządzaj wymaganiami
- używaj architektury komponentowej
- modeluj wizualnie (UML2.0)
- stale weryfikuj jakość
- zarządzaj zmianami.

Na podkreślenie zasługują praktyki związane z zarządzaniem wymaganiami, modelowaniem wizualnym z UML 2.0 oraz używaniem architektury komponentowej. Głównie tymi praktykami kierowano się przy projektowaniu oprogramowania eksperymentalnego.

W ramach RUP występuje dziewięć dyscyplin, począwszy od modelowania biznesowego, a skończywszy na utrzymaniu środowiska produkcyjnego. W każdej z dyscyplin określone są czynności i role odpowiedzialne za wykonanie tych czynności, produkty potrzebne do realizacji czynności a także produkty będące wynikiem realizacji czynności.

W projekcie skoncentrowano uwagę na działaniach w następujących dyscyplinach [6]:

- zarządzanie wymaganiami
- analiza i projektowanie
- implementacja
- testy
- wdrożenie.

W ramach prac zespołu projektowego WAT, współpracującego z wykonawcą zewnętrznym najistotniejsze dyscypliny, których działania musiały zostać wykonane, to:

- zarządzanie wymaganiami
- analiza i projektowanie
- testy.

Model architektoniczny „4+1” [6] obejmuje następujące widoki architektoniczne systemu informatycznego:

- przypadków użycia
- logiczny
- wdrożeniowy
- procesów
- implementacyjny.

W związku z tym, że w ramach prac WAT skupiono się na trzech ww. dyscyplinach, architekturę projektowanego systemu informatycznego przedstawiono z punktu widzenia następujących widoków architektonicznych modelu „4+1” [6]:

- przypadków użycia
- wdrożeniowy.

W niewielkim stopniu opisywano system informatyczny w widoku Logicznym, pozostawiając decyzje projektowe zespołowi zewnętrznemu. Prace zarówno zespołu WAT, jak i zespołu zewnętrznego zostały zorganizowane i podzielone na iteracje, dostarczające wcześniej określone przyrosty oprogramowania eksperymentalnego.

3. Wybrany zestaw standardów

Oprócz konfiguracji procesu projektowania systemu informatycznego, należało dokonać analizy zasadności zastosowania w tym projekcie zestawu standardów modelowania architektury oprogramowania oraz modelowania ścieżek klinicznych. Analizie poddano następujący zestaw standardów [1]:

- BPMN – Business Process Modelling Notation
- GELLO – Object-Oriented Query and Expression Language for Clinical Decision Support
- UML – Unified Modelling Language
- OCL – Object Constraint Language
- XML – Extensible Markup Language
- HL7 – Health Level Seven.

W wyniku prac analitycznych i budowy prototypu wybrano następujący zestaw standardów opisu architektury systemu informatycznego:

- UML
- GELLO – rozszerzenie OCL
- XML.

Zestaw ten umożliwia modelowanie architektury systemu informatycznego, specyfikację komunikatów wymienianych między komponentami oprogramowania oraz formalną specyfikację ograniczeń w modelach architektonicznych.

W zakresie modelowania ścieżek klinicznych zdecydowano, że głównym standardem modelowania i zapisu ścieżki klinicznej będzie standard GLIF (Guideline Interchange Format). Standard ten został dodany do wstępnego zestawu. Podyktowane było to tym, że jest to standard dedykowany do modelowania i wymiany ścieżek klinicznych między systemami informatycznymi. Podstawową jego cechą jest ustandaryzowany sposób opisu ścieżki klinicznej. Ponadto GLIF jest modelowany w UML i wraz z tym standardem uzyskuje się opracowany model danych do przechowywania informacji o ścieżkach klinicznych w sposób trwały w systemie informatycznym. Był to jeden z elementów, który miał być zapewniony przez budowane repozytorium. Kolejną pożądaną cechą tego standardu było obsługiwane tworzenia ścieżki klinicznej za pomocą formatek z wypełnianymi zestawami danych o czynności w ramach ścieżki klinicznej. Ten sposób był zakładany z punktu widzenia łatwości użycia budowanego systemu przez użytkowników końcowych. Utrzymano także zastosowanie standardu BPMN ze względu na łatwość prezentacji przepływu ścieżki klinicznej wyrażonej w tym standardzie. Pełne uzasadnienie wyboru takiego zestawu standardów zostało przedstawione w pracy [1].

4. Specyfikacja wymagań

Pierwszy z widoków architektonicznych stanowi widok Przypadków użycia. W ramach tego widoku opracowano Model przypadków użycia.

Model ten, stanowiący specyfikację wymagań na budowany system, został opracowany w języku UML i uzupełniony opisem tekstowym. Zastosowano diagramy przypadków użycia oraz diagramy aktywności języka UML.

W ramach projektu zidentyfikowano następujących aktorów systemu:

- projektant ścieżek klinicznych
- przeglądający
- administrator
- analityk.

Rysunek 1 przedstawia diagram przypadków użycia z aktorami systemu.

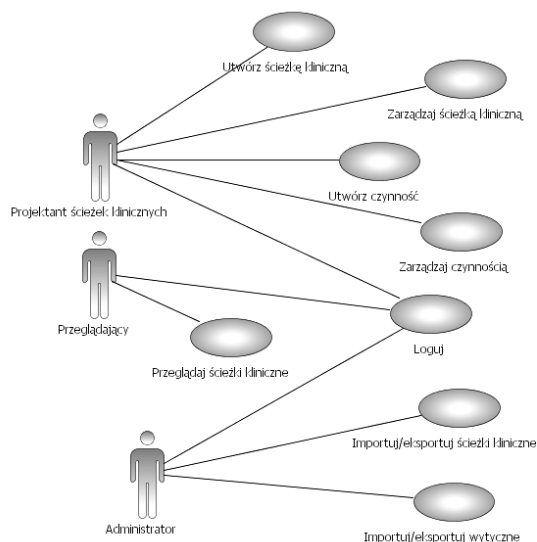


Rys. 1. Aktorzy systemu

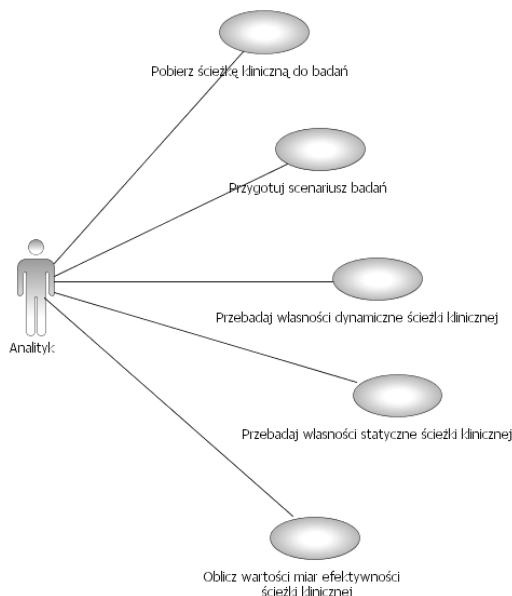
Zaproponowany początkowy zakres funkcjonalny projektowanego systemu zawierał następującą listę przypadków użycia:

- PU.1 – utwórz ścieżkę kliniczną
- PU.2 – zarządzaj ścieżką kliniczną
- PU.3 – utwórz czynność
- PU.4 – zarządzaj czynnością
- PU.5 – przeglądaj ścieżki kliniczne
- PU.6 – importuj/eksportuj ścieżki kliniczne
- PU.7 – importuj/eksportuj wytyczne
- PU.8 – loguj
- PU.9 – pobierz ścieżkę kliniczną do badań
- PU.10 – przygotuj scenariusz badań
- PU.11 – przebadaj własności dynamiczne ścieżki klinicznej
- PU.12 – przebadaj własności statyczne ścieżki klinicznej
- PU.13 – oblicz wartości miar efektywności ścieżki klinicznej.

Na rysunku 2 przedstawiono diagram przypadków użycia systemu dla modułu definicji ścieżek klinicznych.



Rys. 2. Diagram przypadków użycia systemu dla modułu definicji ścieżek klinicznych
Na rysunku 3 przedstawiono diagram przypadków użycia systemu dla modułu symulatora ścieżek klinicznych



Rys. 3. Diagram przypadków użycia systemu dla modułu symulatora ścieżek klinicznych

Zdefiniowano przypadki użycia, określając przepływy zdarzeń oraz przedstawiono zdefiniowane przepływy na diagramach aktywności języka UML.

Poniżej przedstawiono zdefiniowany przepływ zdarzeń dla przypadku użycia PU.6 „Importuj/eksportuj ścieżki kliniczne”:

WP – użytkownik został poprawnie zalogowany do systemu oraz wybrał opcję „Importuj/eksportuj ścieżki kliniczne”.

PG.1 – system wyświetla formatkę do importu/eksportu danych.

Zakres wyświetlanych danych: wybór opcji import/eksport:

- import: nazwa pliku, system wyświetla listę uprzednio wykonanych importów
- eksport: nazwa ścieżki klinicznej, system wyświetla listę uprzednio wykonanych eksportów.

PG.2 – użytkownik wybiera opcję „import” i podaje nazwę pliku XML (podpowiedź w formie standardowej formatki systemowej do lokalizacji plików w systemie operacyjnym) oraz zatwierdza wybór pliku.

PG.3 – system weryfikuje plik XML w zakresie składni XML i formatu pliku wymiany oraz wyświetla podgląd wczytanego pliku. Na dole formatki wyświetla komunikat o poprawności wczytanego pliku. System uaktywnia przycisk „import”.

PG.4 – użytkownik naciska przycisk „import”.

PG.5 – system importuje poprawne ścieżki kliniczne, a błędne wstawia do tabel buforowych.

PG.6 – użytkownik wybiera ostatnio wykonany import.

PG.7 – system wyświetla błędne ścieżki kliniczne.

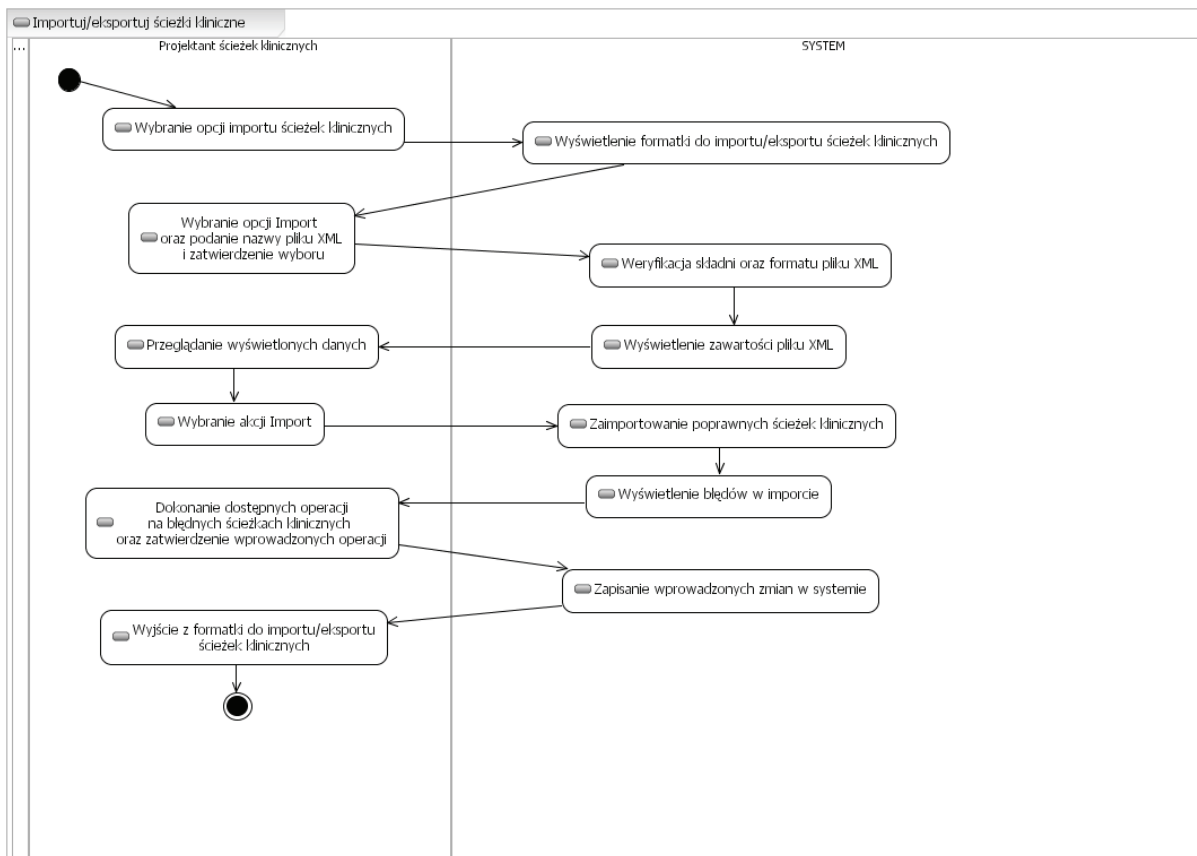
PG.8 – użytkownik dokonuje dostępnych operacji na błędnych ścieżkach klinicznych oraz zatwierdza wprowadzone operacje.

PG.9 – system zapisuje zmiany w bazie danych.

PG.10 – użytkownik wychodzi z formatki do importu/eksportu danych.

WK – system zaimportował poprawne ścieżki kliniczne z pliku XML oraz wprowadził dodatkowo do bazy danych te ścieżki kliniczne, dla których użytkownik dokonał korekty operacji.

Zamieszczony powyżej przepływ zdarzeń przypadku użycia PU.6 „Importuj/eksportuj ścieżki kliniczne” przedstawiono na diagramie aktywności języka UML (rys. 4).



Rys. 4. Diagram aktywności z przepływem głównym przypadków użycia PU.6

Przyjęte zostały następujące założenia na moduł zasilania repozytorium danymi (repozytorium projektu):

- moduł w postaci niezależnej aplikacji typu Enterprise na serwerze aplikacyjnym, np. strona JSP jako View, serwlet jako

Controller, EJB (np. Hibernate) jako synchronizacja modelu z bazą danych. Innym rozwiązaniem może być technologia .NET Microsoft

- moduł posiadający następujące interfejsy: Import – operacja – importuj SK (nazwa-

- pliku* : String) : true; Eksport – operacja – eksportuj SK (*ścieżki* : *Collection*) : true
- źródło danych dla modułu: sformatowany plik XML (zbiór wytycznych SFC – projekt duński)
 - dostępne operacje wymiany: Import z pliku do systemu, Eksport podzbioru danych do pliku XML
 - lista możliwych akcji w imporcie: Wstaw nową ścieżkę, Wstaw nową wersję ścieżki, Modyfikuj istniejącą wersję ścieżki, Anuluj istniejącą wersję ścieżki
 - możliwe sytuacje błędne podczas importu: WS – próba wstawienia nowej ścieżki, która już istnieje; WW – próba wstawienia nowej wersji ścieżki, która już istnieje; MS – próba modyfikacji ścieżki, której nie ma; AS – próba anulowania ścieżki, której nie ma; AA – próba anulowania ścieżki, która jest w statusie „Anulowana”
 - możliwe akcje w sytuacjach błędów w imporcie: WS – Modyfikuj dane istniejącej ścieżki; WW – Modyfikuj dane istniejącej wersji ścieżki; MS – Wstaw nową ścieżkę; AS – Wstaw nową ścieżkę ze statusem „Anulowana”; AA – Brak dostępnych akcji.

Koncepcja architektury oprogramowania określona została w pracy [2].

5. Projekt systemu

Projekt systemu wykonawca zewnętrzny konsultował z zespołem projektowym z WAT. W ramach projektu systemu wykonawca przedstawił widok Logiczny architektury oprogramowania, a w nim Model projektowy.

W trakcie projektowania systemu zestaw przypadków użycia modułu definicji ścieżek klinicznych został rozszerzony o przypadki pomocnicze (umożliwiające zarządzanie systemem).

Końcowa lista przypadków użycia modułu definicji ścieżek klinicznych obejmuje:

- KPU.1 – utwórz ścieżkę kliniczną
- KPU.2 – zarządzaj ścieżką kliniczną
- KPU.3 – utwórz czynność
- KPU.4 – zarządzaj czynnością
- KPU.5 – przeglądaj ścieżki kliniczne
- KPU.8 – loguj
- KPU.6 – importuj wytyczne
- KPU.7 – importuj ścieżki kliniczne
- KPU.9 – zarządzaj źródłami wiedzy
- KPU.10 – przeglądaj historię importów
- KPU.11 – eksportuj ścieżki kliniczne
- KPU.12 – zarządzaj kontami użytkowników.

Architektura końcowa systemu została zaproponowana przez wykonawcę zewnętrznego i po weryfikacji zastosowanych wzorców architektonicznych i podziału na warstwy oprogramowania dokonanej przez WAT zaakceptowana, jako obowiązująca dla rozpatrywanego projektu. Rysunek przedstawiający diagram wdrożeniowy języka UML z widokiem Wdrożeniowym architektury systemu zamieszczono w pracy [4].

6. Testy systemu

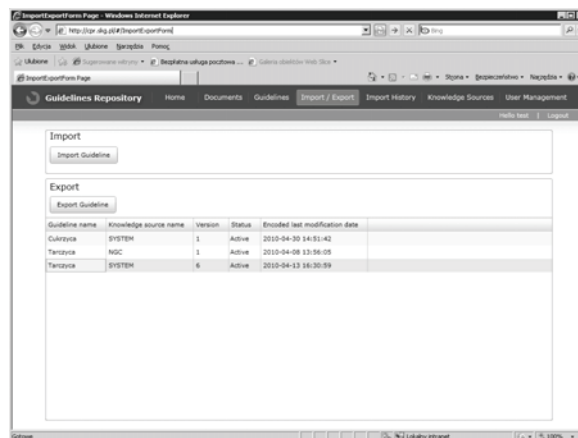
Po wykonaniu systemu dokonano testów funkcjonalnych. Testy przeprowadzono na podstawie przepływów zdarzeń określonych w przypadkach użycia projektowanego systemu.

Oprogramowanie eksperymentalne przeszło pozytywnie serię testów funkcjonalnych. Wyniki testów modułu definicji ścieżek klinicznych zawarto w pracy [3]. Dodatkowo sposób korzystania z opracowanej aplikacji został przedstawiony w opracowaniu [4].

Poniżej przedstawiono funkcjonowanie aplikacji odpowiedzialnej za wykonanie przypadków użycia:

- KPU.7 – Importuj ścieżki kliniczne
- KPU.10 – Przeglądaj historię importów
- KPU.11 – Eksportuj ścieżki kliniczne.

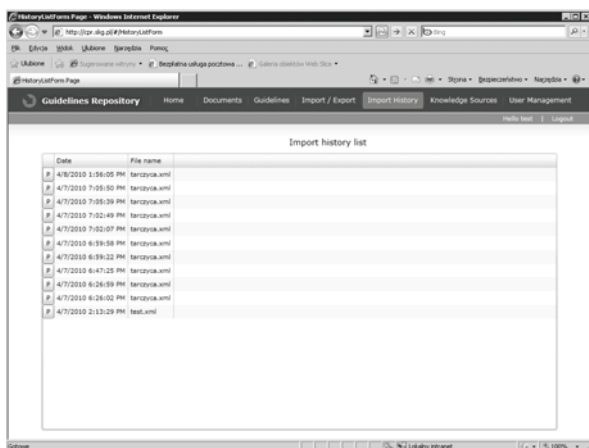
W celu zaimportowania ścieżki klinicznej zgodnej z GLIF 3.5 z zewnętrznych baz informacji medycznej należy w oknie „Import/Eksport ścieżek klinicznych” (rys. 5) nacisnąć przycisk „Importuj ścieżkę kliniczną” („Import Guideline”). Następnie należy wskazać plik w formacie RDF zawierający ścieżki kliniczne podlegające importowi oraz wybrać źródło wiedzy, z którego pochodzą importowane ścieżki kliniczne.



Rys. 5. Okno import/eksport ścieżek klinicznych

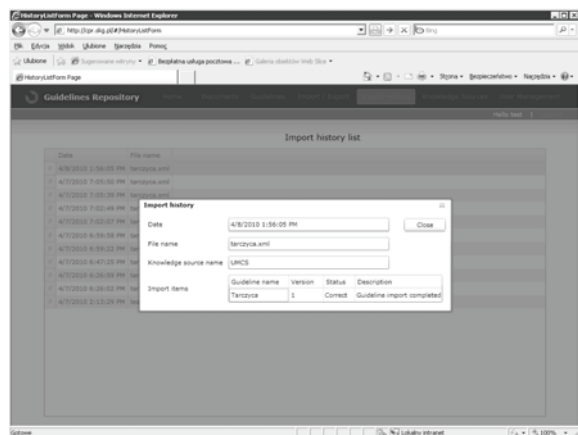
Okno zawierające historię wykonanych importów przedstawione zostało na rysunku 6.

Zawiera ono wykaz wszystkich wykonanych importów.



Rys. 6. Lista importów

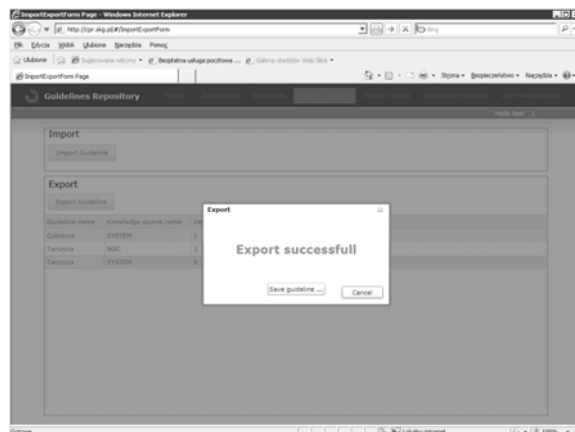
W celu uzyskania bardziej szczegółowych informacji należy nacisnąć przycisk oznaczony literą „P” obok daty importu. Wyświetlone zostanie okno (rys. 7) zawierające szczegółowe informacje o przeprowadzonym imporcie. Okno „Historia przeprowadzonego importu” (rys. 7) zawiera szczegółowe informacje o wybranym imporcie, takie jak: data importu, nazwa importowanego pliku, źródło wiedzy medycznej, a także informacje o każdej ścieżce klinicznej znajdującej się w pliku i poprawności importu określonej ścieżki do repozytorium.



Rys. 7. Okno z historią przeprowadzonego importu

Z repozytorium ścieżek klinicznych z poziomu interfejsu wizualnego można eksportować ścieżki kliniczne będące w statusie „Aktywna”. W tym celu w oknie „Import/Eksport ścieżek klinicznych” (rys. 5) należy zaznaczyć ścieżkę kliniczną, a następnie nacisnąć przycisk „Eksportuj ścieżkę” (Export Guideline). Po poprawnym wygenerowaniu pliku ze ścieżką wyświetlone zostanie okno (rys. 8). W oknie tym

należy nacisnąć przycisk „Zapisz ścieżkę” i wskazać lokalizację dla wygenerowanego pliku. Ścieżki kliniczne eksportowane są w standardzie GLIF 3.5 w postaci pliku RDF.



Rys. 8. Eksport ścieżki zakończony powodzeniem

Na uwagę zasługuje fakt uzyskania praktycznie pełnej zakładanej funkcjonalności przy zachowaniu łatwości definiowania poszczególnych elementów ścieżki klinicznej. Wydaje się, że przyjęty sposób definiowania ścieżki przy zastosowaniu podejścia GLIF będzie łatwy i preferowany przez użytkownika końcowego.

Istotne jest, że udało się uzyskać całościową współpracę systemu (testy konsolidacyjne). Połączono elementy systemu wykorzystujące standard GLIF i BPMN dzięki zastosowaniu pliku wymiany RDF, który jest rozszerzeniem XML.

7. Podsumowanie

Efektom końcowym projektu jest funkcjonująca, stabilna aplikacja do projektowania i zarządzania wytycznymi i ścieżkami klinicznymi. Zasadne było zastosowanie zestawu standardów wywodzących się z małego zbioru podstawowego: UML, XML, GELLO (OCL). Takie podejście ułatwiało budowę systemu i wymianę informacji między jego elementami. Ponadto, zastosowanie GLIF ze zweryfikowanym modelem danych oraz sposobem edycji ścieżek pozwoliło uzyskać łatwość używania aplikacji końcowej oraz możliwość ustandaryzowanej wymiany danych między modułami projektowanego systemu oraz między projektowanym systemem a systemami zewnętrznymi.

Bardzo pozytywny wpływ na sukces projektu, czyli uzyskanie w pełni funkcjonalnego oprogramowania eksperymentalnego, które zbudowano w harmonogramie i w budżecie, było zastosowanie dostosowanej

do potrzeb projektu metodyki Rational Unified Process. Sprawdziło się w tym projekcie podejście ukierunkowane na architekturę z jasno zdefiniowanymi widokami architektonicznymi i jednolitym językiem modelowania (Unified Modelling Language) architektury systemu informatycznego. Język UML jest standardem modelowania systemów informatycznych w środowisku akademickim oraz w firmach komercyjnych. Na sukces projektu miało także wpływ zastosowanie podejścia z jasno zdefiniowanym zestawem wymagań na system informatyczny. W trakcie trwania projektu wymagania były dopracowywane w pracach zespołu projektowego WAT oraz wykonawcy zewnętrznego, przez co uzyskane było ciągle porozumienie nad zebrany zestawem wymagań. Testy i odbiór oprogramowania eksperymentalnego realizowane były na podstawie tego zestawu.

8. Bibliografia

- [1] T. Górski, „Analiza przydatności wybranych standardów do modelowania architektury system informatycznego”, *Biuletyn ISI*, Nr 4, 17-28 (2009)
- [2] T. Górski, „Wykonanie koncepcji architektury i wymagań dla modułu zasilania repozytorium danymi”, projekt POIG.01.03.01-00-145/08, opracowanie własne, 2010.
- [3] T. Górski, „Weryfikacja zgodności implementacji modułu zasilania danymi źródłowymi z założeniami projektowymi z zadania 3, weryfikacja wykonania konsolidacji oraz przeprowadzenie testów modelu repozytorium”, projekt POIG.01.03.01-00-145/08, opracowanie własne, 2010.
- [4] P. Michałowski, „Wykonanie dokumentacji techniczno-użytkowej, zadanie 4. projektu POIG.01.03.01-00-145/08”, opracowanie wewnętrzne WAT, Warszawa, 2010.
- [5] G. Bliźniuk, „Raport z realizacji zadania 4. projektu POIG.01.03.01-00-145/08”, opracowanie własne, 2010.
- [6] P. Kruchten, *The Rational Unified Process, An Introduction*, Addison-Wesley, USA, 1999.
- [7] J. Barnes, *Implementing the IBM Rational Unified Process and Solutions: A Guide to Improving Your Software Development Capability and Maturity*, IBM Press, 2007.
- [8] M. Fowler, *UML Distilled Third Edition*, Addison-Wesley, 2005.

Architecture-driven approach in experimental software designing

T. GÓRSKI

The paper describes the use of an information system architecture-driven approach in the process of experimental software design to build models of clinical paths. This article presents the results of the analysis of the applicability of BPMN, GELLO, UML, OCL, XML, HL7 and GLIF standards in the context of their suitability for modelling system architecture. Selected set of standards is evaluated in terms of obtaining the desired effects of construction of information system architecture and creation of clinical paths models in the experimental software.

Keywords: information system architecture, requirements management