

WOJCIECH RZEPKOWSKI*, TOMASZ SERAFIŃSKI*,
SZYMON GRABOWSKI**

**ZARZĄDZANIE ZMIANĄ OPROGRAMOWANIA
NA PRZYKŁADZIE MAGAZYNU WYSOKIEGO SKŁADOWANIA**

**SOFTWARE CHANGE MANAGEMENT IN SYSTEM FOR A HIGH – BAY
WAREHOUSE**

STRESZCZENIE: Praca przybliży problematykę zarządzania zmianą oprogramowania w przedsiębiorstwie i przedstawia propozycję systemu zarządzania zmianami dla magazynu wysokiego składowania. Ponadto omówione są nowoczesne technologie wspomagające tworzenie formularzy, oparte na standardzie XML.

ABSTRACT: The paper contains a brief outline of software configuration management. A diagram proposal for software change handling is given. Moreover, modern XML-based technologies ifacilitating processing user forms are presented.

1. Wprowadzenie

Wymóg zarządzania zmianami oprogramowania jest jednym z podstawowych warunków efektywnej realizacji złożonych przedsięwzięć informatycznych. W niniejszej pracy przedstawiamy opracowane przez nas procedury i formularze do zarządzania zmianami oprogramowania na przykładzie systemu magazynu wysokiego składowania. Procedury te mają istotne znaczenie w procesie zarządzania oprogramowaniem w nowoczesnym, w pełni zautomatyzowanym magazynie. Wprowadzenie procedur zarządzania zmianami owocuje wzrostem produktywności zespołu informatyków zajmujących się systemem informatycznym. Każdy wie, co ma do zrobienia i w każdej chwili może sięgnąć do bazy zmian, żeby sprawdzić, jakich zmian dokonano w ostatnim czasie. Ponadto przy pomocy odpowiednich narzędzi (formularze) można w łatwy sposób wygenerować raporty dotyczące zmian w systemie informatycznym. Przy pracy nad dokumentem tekstowym potrzeba istnienia przycisku „cofnij” jest oczywista; analogicznie – jeśli wolno nam użyć takiego sformułowania – przydałby się odpowiednik takiego przycisku podczas

* Katedra Informatyki Stosowanej Politechniki Łódzkiej

** Społeczna Wyższa Szkoła Przedsiębiorczości i Zarządzania, Łódź

dokonywania zmian w programie, bardzo wygodny w sytuacji, w której wprowadzona zmiana zakłóciła pracę systemu. Mechanizmem umożliwiającym wygodne, kontrolowane wprowadzanie i anulowanie zmian będzie dokumentacja zmian w systemie. Zazwyczaj zarządzanie zmianami dotyczy firm produkujących oprogramowanie, ale w rozpatrywanym przypadku dotyczy użytkownika systemu, który chce panować nad swoim systemem informatycznym.

W Sekcji 2 przedstawimy pokrótce notacje opisujące procesy zachodzące podczas zarządzania zmianami oprogramowania: UML, EPC i metodę prof. Scheera, wykorzystywaną w szeroko stosowanym programie ARIS. Sekcja 3 zawiera propozycję systemu zarządzania zmianami oprogramowania dla magazynu wysokiego składowania. Sekcja 4 dotyczy wyboru narzędzia do raportowania i zgłaszania zmian.

2. Notacje

Język UML

Im bardziej skomplikowany system, tym ważniejsza staje się komunikacja między wszystkimi osobami zaangażowanymi w jego tworzenie i wdrażanie. W zeszłej dekadzie UML stał się standardowym językiem do tworzenia planów oprogramowania, używanym przez analityków, projektantów i programistów.

Podstawą języka UML (*Unified Modeling Language*) jest dziewięć rodzajów diagramów modelowania (Rumbaugh i Jacobson 2001), które można podzielić na dwie grupy: diagramy struktury i diagramy zachowania. W skład pierwszej grupy wchodzi diagramy klas, obiektów, komponentów i diagramy wdrożenia, natomiast do diagramów zachowania należą diagramy sekwencji, współpracy, przypadków użycia, stanów i aktywności. Diagramy sekwencji i diagramy współpracy można ogólnie określić mianem diagramów interakcji. Diagramy struktury opisują statyczne aspekty systemu, natomiast diagramy zachowania, zgodnie z nazwą, jego aspekty dynamiczne.

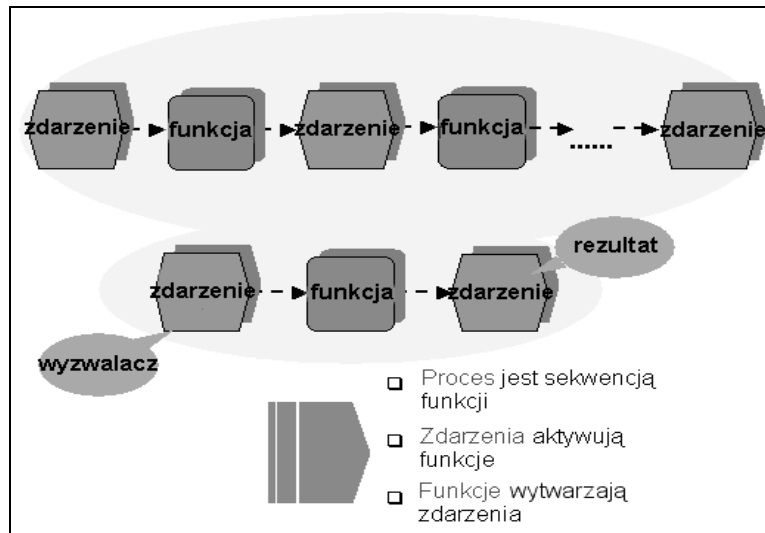
Metoda EPC

Metoda *Event-Driven Process Chain* (EPC) została opracowana w Instytucie Systemów Informatycznych (IWi) Uniwersytetu Saarland w Niemczech przy współpracy z SAP AG. Jest ona kluczowym elementem koncepcji modelowania SAP R/3 dla celów *business engineering*. Metoda ta opiera się na koncepcji sieci stochastycznych (<http://www.rtcn.org.pl>).

Metoda EPC skupia się na modelowaniu biznesowym wykonywanym przez operatora. Jest ona opracowana do modelowania ogólnych zachowań łańcucha procesu.

EPC umożliwia modelowanie procesu jako trwającej w czasie i/lub logicznej sekwencji funkcji. W metodzie EPC proces może być rozumiany jako pewna ilość funkcji wywołanych przez jedno lub więcej zdarzeń. Funkcje są połączone przez zdarzenia. Zdarzenie może być scharakteryzowane przez swój status i datę. Zdarzenia mogą wyzwać funkcje. Zdarzenia kontrolują funkcje – jednak mogą być one również rezultatem funkcji. Powinny być rozumiane jako główny element kontroli procesu. Wiele funkcji może być

wynikiem jednego zdarzenia. Z drugiej strony, niekiedy kilka funkcji musi się zakończyć przed wywołaniem jakiegoś zdarzenia (p. Rysunek 1). Funkcję rozpoczyna i kończy co najmniej jedno zdarzenie. Wskutek tego zdarzenie wyzwalające nazywane jest zdarzeniem początkowym, a zdarzenie powstałe w wyniku wykonania funkcji nazywane jest zdarzeniem końcowym (p. Rysunek 1).



Rys. 1. EPC – Funkcje i zdarzenia – źródło: <http://www.rtcn.org.pl>

W celu umożliwienia modelowania złożonej sekwencji operacji z zachowaniem jasności i przejrzystości, metoda EPC została poszerzona. Rozszerzona metoda *extended Event-Driven Process Chain* (eEPC) umożliwia modelowanie łańcuchów procesów sterowanych zdarzeniami. Dodatkowo w metodzie tej możliwe jest modelowanie obrazu organizacji i danych. Tak więc metoda eEPC zawiera podstawowe elementy metody EPC.

Metoda prof. Scheera

Spośród wielu metod opisu procesów wyróżnia się koncepcja stworzona przez prof. Augusta Wilhelma Scheera (Scheer 1992) z Uniwersytetu Saarbrücken, który od wielu lat pracuje nad opisem metod umożliwiających mapowanie, analizę i reorganizację procesów gospodarczych. Na stworzonej przez niego koncepcji oparte jest jedno z wiodących narzędzi, służących mapowaniu i reorganizacji procesów – pakiet programów ARIS. Koncepcja stworzona przez Scheera i zaimplementowana w ARISie zakłada, że model procesów może być przedstawiony w czterech perspektywach:

- organizacji (*organization view*), w której ukazane są elementy struktury organizacyjnej organizacji;
- danych (*data view*), przedstawiająca system informacyjny organizacji;

- funkcji (*function view*), ukazująca występujące w procesie funkcje i powiązania między nimi;
- sterowania (*control view*), łącząca wydarzenia, funkcje i wszystkie pozostałe elementy występujące w poprzednich trzech perspektywach; pozwala na przedstawienie wzajemnych powiązań pomiędzy pozostałymi perspektywami.

W ramach każdej z czterech perspektyw wyróżnić można trzy poziomy (*descriptive levels*) opisujące przejście od problemu biznesowego do technologii informacyjnej:

- zdefiniowanie wymagań (*requirements definition*) – na poziomie tym określa się wymagania dla technologii informacyjnych;
- specyfikacja projektowa (*design specification*) – na tym poziomie powstaje specyfikacja systemu informacyjnego, który spełni postawione na pierwszym poziomie wymagania;
- opis implementacji (*implementation description*) – w ramach tego poziomu specyfikacja przekształcana jest we wdrożenie odpowiedniego sprzętu komputerowego i oprogramowania.

Z punktu widzenia modelującego i przekształcającego proces analityka najważniejszy jest poziom pierwszy, czyli zdefiniowanie wymagań dla systemu informacyjnego.

3. Opracowanie diagramu zarządzania zmianami

Aby kontrola zmian w systemach komputerowych była możliwa, należy wcześniej spełnić wiele warunków, które można skrótowo określić mianem dobrej praktyki nadzorowania systemów komputerowych. Należy do niej m. in. odpowiednio opracowany przebieg procesu nadzorowania zmianą. Dlatego w procesie kontrolowania zmian przedstawiciel działu informatycznego będzie nieodzowny do określenia związku przyczyna – skutek w zmienionym systemie, natomiast przedstawiciele działów zgłaszających zmianę sprawdzą wpływ na elementy istotne z ich punktu widzenia. Dla przedstawienia tego procesu spośród przedstawionych notacji wybraliśmy diagram EPC, utworzony w oparciu o metodę prof. Scheera. Metoda ta jest stosunkowo prosta i ma zastosowanie dla procesów biznesowych.

Opracowany diagram dotyczy procesu zarządzania zmianami w systemie informatycznym magazynu wysokiego składowania. Niestety, z przyczyn technicznych nie możemy w pracy zreprodukować wynikowego diagramu. Poniżej opisujemy jednak logikę procesu.

W pierwszej kolejności pracownik działu zgłaszającego zmianę wypełnia formularz zmiany wymagań (FZW) dostępny na serwerze, który po akceptacji kierownika działu jest przedkładany do działu informatyki (symbol: TK) i przyjmowany przez pracownika tego działu. Następnie uprawniony pracownik TK przeprowadza wstępną analizę wniosku i sprawdza, czy zgłaszana zmiana mieści się w funkcjonalności systemu. Jeśli tak, to dokonanie zmiany jest na tyle proste, że można ją wprowadzić bez żadnych problemów, odnotowując jedynie fakt jej zaistnienia w raporcie końcowym formularza wprowadzonych zmian (FWZ). Jeśli jednak zmiana nie mieści się w funkcjonalności systemu, to uprawniony pracownik TK analizuje dokumentację systemu oraz kod, w celu określenia, jakiego nakładu pracy i środków wymaga ta zmiana wymagań. W przypadku nakładu znacznych środków finansowych bądź istotnych zmian w systemie, wniosek FZW kierowany jest do rozpatrzenia przez dyrektora pionu. Po jego akceptacji wraca do dalszej analizy i propozycji rozwiązania zgłoszonego problemu. Jeśli zmiana nie wymaga decyzji

dyrektora pionu, to propozycja rozwiązania przedkładana jest przez uprawnionego pracownika TK na formularzu propozycji rozwiązania (FPR) do akceptacji kierownikowi działu zgłaszającego. W przypadku odrzucenia tej propozycji, formularz FPR jest jeszcze raz generowany dla innego rozwiązania tego samego problemu. Następnie przechodzimy do przeprowadzenia testów, które w przypadku stwierdzenia nieprawidłowego działania systemu cofają wniosek propozycji rozwiązania do ponownej analizy.

Po przeprowadzeniu testów wprowadzonej zmiany i ich pozytywnym wyniku przechodzimy do wprowadzenia zmiany na stałe i wygenerowaniu raportu końcowego FWZ.

4. Formularze

4.1 Narzędzia do tworzenia formularzy

Środowisko MS Office System 2003 oferuje technologie, pozwalające na konstruowanie „inteligentnych” rozwiązań klienckich. Rozwiązania, o których mowa, to aplikacje korzystające z informacji udostępnianych przez usługi internetowe XML. Mogą one w sposób dynamiczny łączyć się z serwisami internetowymi w celu analizowania, współdzielenia lub przetwarzania danych. W rezultacie zastosowania tych mechanizmów powstają elastyczne, dające się dostosować do potrzeb użytkownika aplikacje, oferujące zawsze aktualne informacje, które zarówno w trybie online, jak i offline, zachowują swoją pełną funkcjonalność. Dzięki programom pakietu MS Office 2003, aplikacje klienckie mogą stać się ogniwami łańcucha zarządzania informacjami przetwarzanymi w przedsiębiorstwie, nie zaś jedynie statycznymi kontenerami, przeznaczonymi na dane binarne. Aplikacje typu „smart client” mogą być integralnymi komponentami infrastruktury informatycznej przedsiębiorstwa, scalonymi z logiką biznesową i mechanizmami przetwarzania danych, bazującymi na wykorzystaniu usług internetowych XML. Dzięki technologii XML, pozwalającej na oddzielenie danych od innych elementów dokumentu, prezentacji lub arkusza, rozwiązanie utworzone przez pracowników może być ponownie wykorzystane w innych aplikacjach. Ponieważ informacje przetwarzane w różnych aplikacjach mogą być opisane przez taki sam niestandardowy (tj. zdefiniowany przez użytkownika) schemat XML, programistom łatwiej jest wykorzystywać te informacje podczas konstruowania inteligentnych rozwiązań klienckich.

4.2 Inteligentne dokumenty

Technologia kryjąca się pod pojęciem „inteligentnego dokumentu” (ang. *smart document*) składa się na nowy model rozwiązania, bazujący na dokumencie, skoroszybie bądź szablonie aplikacji MS Word 2003 lub MS Excel 2003, posiadającym odpowiednią strukturę XML oraz wyposażonym w dostosowane do potrzeb użytkownika okienko zadań (ang. *task pane*). Struktura XML inteligentnego dokumentu oparta jest na wykorzystaniu niestandardowego schematu XML, którego zadaniem jest wyświetlanie odpowiednich informacji, wraz z przemieszczaniem kursora po różnych obszarach dokumentu. Przykładowo, w panelu zadań mogą pojawiać się informacje pomocnicze, dane odnoszące

się do treści dokumentu, narzędzia pozwalające przetwarzać dokument czy przydatna grafika.

Ponieważ dokumenty inteligentne pobierane są zazwyczaj z zaufanego serwera (ang. *trusted server*), ich kod – zwany pakietem rozszerzającym XML (ang. *XML expansion pack*) – może być również aktualizowany z tego serwera. Każdy dokument inteligentny posługuje się tzw. plikiem-wykazem (ang. *manifest file*), który jest niczym innym, jak plikiem XML definiującym lokalizację kluczowych komponentów dokumentu inteligentnego. Ponieważ informacja o lokalizacji pliku-wykazu zarejestrowana jest w dokumencie inteligentnym, użytkownik otwierający dokument zostaje zapytany o to, czy mają zostać zainstalowane komponenty wyszczególnione w odpowiednim pliku-wykazie.

4.3 Inteligentne tagi

Użytecznym elementem inteligentnych rozwiązań klienckich okazują się również tagi inteligentne (ang. *smart tags*), za pomocą których można znakować pewne typy danych, a następnie wykonywać na tych danych określone czynności (na przykład przeszukiwanie lub aktualizowanie). W pakiecie MS Office 2003 tagi inteligentne, których działanie można dostosować do potrzeb użytkownika, obsługiwane są m. in. przez aplikacje Word, Excel, PowerPoint i (częściowo) Access. Przykładowo, tagów inteligentnych można używać w stosunku do określonych komórek arkuszy kalkulacyjnych Excela 2003 lub pewnych pól tabel Accessa 2003.

Wspólne korzystanie w programach pakietu MS Office 2003 z udoskonaleń oferowanych przez tagi inteligentne przyczynia się do rozszerzenia użyteczności inteligentnych rozwiązań klienckich. Przyporządkowanie czynności wykonywanych przez tagi inteligentne do elementów XML oraz samoistne wyzwalanie akcji wskutek rozpoznania inteligentnego znacznika pozwala rozwiązaniom „smart client” na automatyczne zbieranie metadanych już podczas ich wprowadzania lub na aktualizowanie w czasie rzeczywistym jednej sekcji dokumentu, w oparciu o dane wprowadzane do innej.

4.4 Strona dostępu do danych w MS Access

MS Access umożliwia tworzenie różnych typów stron sieci Web. Aby wykonywać operacje na danych bezpośrednio w bazie danych, należy użyć strony dostępu do danych; aby wyświetlać aktualne dane tylko do odczytu, należy rozważyć zastosowanie generowanych na serwerze plików ASP lub IDC/HTX; aby wyświetlać migawki danych, należy użyć formatu statycznych plików HTML. Aby zapewnić spójność wyglądu tworzonych stron sieci Web, można użyć plików szablonów HTML.

Strona dostępu do danych to strona sieci Web bezpośrednio połączona z danymi w bazie danych. Strony takie umożliwiają wyświetlanie, edycję, aktualizację, usuwanie, filtrowanie, grupowanie i sortowanie danych pobieranych na bieżąco z bazy danych programu MS Access lub bazy danych programu MS SQL Server w programie Internet Explorer 5 lub nowszej jego wersji. Strona może także zawierać składniki takie jak arkusz kalkulacyjny, lista tabeli przestawnej lub wykres.

Stronę dostępu do danych tworzy się jako obiekt bazy danych zawierający skrót do miejsca, w którym znajduje się odpowiadający tej stronie plik HTML.

Strony dostępu do danych można zapisywać także jako pliki HTA (HTML Application). Pliki HTA umożliwiają wykorzystanie technologii dynamicznego języka HTML (DHTML) „poza przeglądarką” w celu tworzenia aplikacji autonomicznych. Używając plików HTA można tworzyć aplikacje w taki sam sposób, w jaki tworzone są strony sieci Web przy użyciu języka DHTML i skryptów, lub można pobrać zawartość, która była pierwotnie przeznaczona do użycia w sieci Web i przetworzyć ją na aplikację.

Aby udostępnić swoje strony dostępu do danych w sieci WWW, należy je opublikować w folderach sieci Web lub na serwerze sieci Web. Należy także udostępnić użytkownikom danej strony bazę danych programu Access lub SQL Server.

Internet Explorer musi pobrać stronę z serwera Web tylko raz, aby umożliwić użytkownikowi przeglądanie i pracę z danymi na stronie. Ponieważ strona wykorzystuje pliki DHTML, dostęp do bazy danych w środowisku klient/serwer jest zazwyczaj bardzo wydajny.

Aby inni użytkownicy mogli przeglądać i pracować z danymi na stronie dostępu do danych przy użyciu przeglądarki sieci Web lub wiadomości pocztowej, baza danych MS Access lub Microsoft SQL Server, która jest źródłem danych OLE DB dla tej strony dostępu do danych, musi być umieszczona na udostępnionym serwerze lub komputerze.

Dobrze jest umieścić bazę danych na udostępnianym serwerze lub w komputerze przed utworzeniem strony. Jeśli przeniesienie bazy danych, z którą strona jest połączona nastąpi po utworzeniu strony, konieczne będzie dokonanie zmiany połączenia źródła danych OLE DB strony w oknie dialogowym *Połączenie tej strony dostępu do danych*.

Chociaż źródłem danych strony może być tylko baza danych Access lub SQL Server, dane w innych formatach można także udostępnić z poziomu strony, korzystając z tabel połączonych bazy danych Access.

Aby możliwie dobrze zabezpieczyć strony dostępu do danych, bazę Access lub SQL Server należy umieścić na tym samym serwerze Web, co stronę dostępu do danych.

Należy sprawdzić, że ścieżka do źródła danych strony została określona przy użyciu konwencji nazewnictwa UNC, a nie litery mapowanego dysku sieciowego używanej w Eksploratorze Windows. Litera dysku może się zmieniać zależnie od komputera albo może być niezdefiniowana, natomiast ścieżka UNC jest niezawodnym i dobrym sposobem odszukania źródła danych przez stronę.

5. Literatura

Scheer A.W. 1992: *Architecture of Integrated Information Systems*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Rumbaugh J., Jacobson I. 2001: *UML, podręcznik użytkownika*, WNT, Warszawa.

<http://www.rtcn.org.pl>: *Metody, narzędzia i techniki*.