AUTOMATYCZNE POZYSKIWANIE WIEDZY KONSULTANTÓW Z PAKIETÓW SYSTEMU SAP R/3

Rafał KLAUS, Bartosz ŻYLIŃSKI

Streszczenie: Celem artykułu jest przedstawienie idei i głównych elementów systemu automatycznego pozyskiwania wiedzy z pakietów programistycznych systemu SAP R/3. Prezentowany system jest próbą rozwiązania problemu uzyskania, gromadzenia i wykorzystania w firmie wiedzy i doświadczenia pracowników. Ze względu na specyfikę konstrukcji SAP R/3 zagadnienie to jest wielopłaszczyznowe nie tylko technicznie ale również z punktu widzenia stosunków międzyludzkich. Rozwiązanie tego problemu, w zakresie prezentowanego w artykule materiału zakończyło się sukcesem.

Słowa kluczowe: SAP R/3, pozyskiwanie modeli, metryki oprogramowania.

1. Wprowadzenie

Istnieje wiele powodów tworzenia dokumentacji poprojektowej. Automatyzacja pozyskiwania danych do tworzenia tej dokumentacji jest uzależnione od rodzaju środowiska w którym realizowane było zlecenie [3].

W ogólności po zakończonym wdrożeniu nowego systemu istotną kwestią jest przygotowanie dokumentacji z projektu oraz przygotowanie repozytorium z kodem źródłowym aplikacji wchodzących w skład systemu. Pozwala to na łatwiejsze prowadzenie poprawek i zmian w systemie oraz udostępnia przyszłym programistom, deweloperom oraz konsultantom historię projektu, która może zostać wykorzystana w przyszłości w innym projekcie jako źródło wiedzy oraz źródło gotowych, sprawdzonych rozwiązań.

W przypadku projektów tworzonych w takich technologiach jak *JAVA*, pierwszorzędną sprawą jaką trzeba zagwarantować zespołowi programistycznemu jest możliwość udostępniania efektów swojej pracy innym członkom zespołu. Wynika to z faktu, że każdy członek zespołu wykorzystuje lokalna instalację środowiska programistycznego. Elementy projektu stworzone przez każdego z członków zespołu składowane są w lokalnych systemach plików, na lokalnych maszynach przez co ich udostępnianie jest problemem. W tym celu stosuje się takie systemy wersjonowania jak *CVS* (Concurrent Versions System), *SVN* (Subversion). Mają one zagwarantować możliwość dzielenia się pracą z pozostałymi członkami zespołu.

W przypadku projektów tworzonych dla systemu *SAP R/3* sprawa wygląda całkowicie inaczej. Centralnym repozytorium dla elementów projektu jest sam system. Wystarczy się do niego zalogować by mieć dostęp do każdego elementu projektu tworzonego przez członków zespołu [1]. Sam system posiada zintegrowane środowisko deweloperskie przez co nie ma konieczności składowania elementów w lokalnym systemie plików. Elementy są edytowane na bieżąco w repozytorium.

Jednak w takiej sytuacji pojawia się poważny problem. Jeżeli z jakiegokolwiek powodu członek zespołu nie ma dostępu do systemu to nie ma możliwości sprawdzenia stanu projektu ani pobrania elementów wchodzących w jego skład. Taka sytuacja zdarza się bardzo często wśród konsultantów wdrożeniowych. Najczęściej swoją pracę wykonują oni bezpo-

średnio w siedzibie klienta. Będąc podłączonym do lokalnej sieci komputerowej nie mają dostępu do innych systemów niż ten, na którym pracują. Często zdarza się, że klient bojąc się o swoje dane nie daje konsultantowi dostępu do sieci zewnętrznej. Problem stanowią również różne wersje aplikacji korzystających z technologii *VPN* (Virtual Private Network) blokujące dostęp do innych zasobów sieciowych niż ten, z którym konsultant aktualnie pracuje. Wszystkie te sytuacje sprawiają, że użytkownik nie ma dostępu do danych, których jest właścicielem.

Istnieje również możliwość´ przekopiowania danych z jednego systemu na drugi poprzez ich tymczasowe połączenie na czas kopiowania, jak również przenoszenie plików zleceń transportowych. Takie sytuacje zdarzają się jednak bardzo rzadko ponieważ właściciele systemów źródłowych obawiają się kradzieży danych. Podczas kopiowania jednego z pakietów na system docelowy istnieje możliwość pobrania innych danych niezwiązanych z aktualnym projektem np. danych dotyczących zatrudnienia pracowników jak również ich danych osobowych.

Celem artykułu jest pokazanie powstałej w wyniku współpracy Instytutu Informatyki Politechniki Poznańskiej i firmy BCC idei systemu pozwalającego na skopiowanie danych dotyczących projektu konfiguracji, zapisanie ich bezpośrednio w lokalnym systemie plików na lokalnej maszynie konsultanta a następnie odtworzenia ich za pomocą aplikacji działającej lokalnie. Tak powstały system ma za zadanie umożliwić przeglądanie elementów projektu bez konieczności logowania się do systemu jak również pozwolić na generowanie dokumentacji do projektu i umożliwić analizę elementów projektu pod względem ich złożoności, przydatności bądź innej cechy charakterystycznej dla danego elementu. Koncepcja takiego systemu została dobrze przyjęta przez środowisko konsultantów systemów *SAP R/3*, którzy od dawna potrzebowali narzędzia które ułatwiłoby im zapisywanie historii swojej pracy oraz generowanie dokumentacji dla osób nie biorących bezpośredniego udziału w projekcie.

2. Pakiety systemu SAP R/3

System *SAP R/3* jest sztandarowym oprogramowaniem klasy *ERP* (ang. Enterprise Resource Planing) firmy *SAP AG*. System ten został zaprojektowany do koordynowania zasobów, informacji oraz czynności niezbędnych do wykonywania procesów biznesowych. *SAP R/3* jest jednym z najpopularniejszych systemów klasy *ERP* na rynku *IT*. System SAP R/3 oparty jest na architekturze trójwarstwowej typu klient-serwer. Warstwy tej architektury to:

- serwer prezentacji (klienta),
- serwer aplikacji,
- serwer bazy danych.

Serwerem prezentacji dla systemu *SAP R/3* jest program o nazwie *SAPGUI*. Jest on instalowany na stacjach roboczych użytkowników. Jego zadaniem jest umożliwienie połączenia z systemem oraz wyświetlanie danych odebranych z systemu. Wyróżniamy trzy różne wersje tego programu:

- SAP GUI for Windows – program zaprojektowany specjalnie dla systemów operacyjnych z rodziny Windows.
- SAP GUI for the Java Environment – program oparty na technologii *JAVA* przeznaczony dla systemów operacyjnych obsługujących wirtualne maszyny *JAVA*.
- SAP GUI for HTML – umożliwia korzystanie z systemu poprzez przeglądarkę internetową.

Serwer aplikacji zajmuje się interpretowaniem i zarządzaniem programami napisanymi w języku *ABAP* oraz udostępnianiem im wejścia i wyjścia. Zadaniem serwera aplikacji jest odpowiednie przygotowywanie odwołań do serwera bazy danych, wysyłanie odwołań, pobieranie danych od serwera bazy danych, przekazywanie ich do programów a następnie przesyłanie wyników do serwera prezentacji. Serwer bazy danych odpowiada za zarządzanie bazą danych. Baza danych zarządzana przez serwer pochodzi od innego dostawcy oprogramowania. Najczęściej wykorzystywane są bazy danych *MSSQL*, *MYSQL* oraz bazy danych firmy Oralce.

Pakiet systemu *SAP R/3* jest podstawowym kontenerem logicznym dla obiektów deweloperskich [2,4,6]. Obiekty te są powiązane ze sobą poprzez wspólne przeznaczenie, funkcję, warstwę transportową, należą do jednego działu, wchodzą w skład jednej aplikacji bądź jej części. Przykładem takich obiektów są programy, klasy, schematy tabele baz danych. Pakiety mogą być grupowane za pomocą pakietów strukturalnych. Pakiet strukturalny jest najwyższą jednostką w hierarchii pakietów. Nie przechowuje żadnych obiektów deweloperskich a jedynie wiąże inne pakiety i jest wykorzystywany do strukturyzacji projektu programistycznego. Przykładem pakietów strukturalnych są *BASIS*, *HR*.

Logiczna struktura pakietu [7]:

- Obiekty Słownika ABAP (ang. ABAP Dictionary Objects)
  - Tabele bazy danych (ang. Database tables)
  - Typy tabel (ang. Table Types)
  - Struktury (ang. Structures)
  - Elementy Danych (ang. Data elements)
  - Domeny (ang. Domains)
  - Obiekty blokowania (ang. Lock objects)
  - Transakcje (ang. Transactions)
  - Pomoce Wyszukiwania (ang. Search helps)
  - Makra (ang. Macros)
  - Grupa typów (ang. Type group)
- Biblioteka Klas (ang. Class Library)
  - Klasy (ang. Classes)
  - Interfejsy (ang. Interfaces)
- Programy (ang. Programs)
- Biblioteka BSP (ang. BSP Library)
  - Aplikacje BSP (ang. BSP Applications)
    - Kontrolery (ang. Controllers)
    - Widoki (ang. Views)
    - Strony z logika˛ przepływu (ang. Pages with flow logic)
    - Fragmenty strony (ang. Page fragments)
  - Rozszerzenia BSP (ang. BSP Extensions)
  - Temat BSP (ang. BSP Theme)
- Grupy Funkcyjne (ang. Function Groups)
  - Moduły funkcyjne (ang. Function modules)
- Biblioteka WebDynpro (ang. Web Dynpro Library)
  - Aplikacje Web Dynpro (ang. Web Dynpro Applications)
  - Komponenty Web Dynopro (ang. Web Dynpro Components)
    - Komponent kontroler (ang. Componentcontroller)
    - Kontrolery niestandardowy (ang. Custom controller)
    - Okna (ang. Windows)

- Widoki (ang. Views)
- Kontrolery konfiguracyjne (ang. Configuration controllers).

3. Idea rozwiązania

Cel to wytworzenie narzędzia, do pozyskiwania pakietów programistycznych systemu *SAP R/3*. Dodatkowymi zadaniami tego narzędzia jest możliwość wyświetlania, przeglądania i generowania dokumentacji dla pobranych wcześniej pakietów. Projekt składa się z następujących części:

- raport napisany w języku *ABAP* działający po stronie systemu *SAP R/3*,
- aplikacja desktopowa napisana w języku *JAVA* działająca po stronie klienta – użytkownika

Zadaniem raportu systemu *SAP R/3* jest pobieranie pakietów programistycznych z systemu i zapisywanie go w podanej lokalizacji przez użytkownika. Raport języka *ABAP* jest akronimem programów innych strukturalnych języków programowania.
Zadaniem aplikacji klienckiej jest przetworzenie wcześniej zapisanego pakietu z podanej lokalizacji na dysku lokalnym i wyświetlenie jej w przystępny sposób użytkownikowi. Dodatkowym zadaniem aplikacji jest wygenerowanie dokumentacji technicznej dla wyświetlanego pakietu oraz wygenerowanie zestawu metryk obiektowych umożliwiających ocenę zawartych w pakiecie elementów obiektowych(klasy, interfejsy).

3.1. Przypadki użycia

Aktorzy:
UŻYTKOWNIK – użytkownik raportu oraz aplikacji
SERWER – system *SAP R/3*
1. Operacje przeprowadzane po stronie systemu *SAP R/3*.

UC1.1 Utworzenie raportu w systemie

Sytuacja: Użytkownik tworzy nowy raport w systemie, który będzie pobierał pakiety programistyczne.

Przebieg:

1.1.1 Użytkownik uruchamia systemową transakcję *SE36* służącą do tworzenia nowych raportów.

1.1.2 Użytkownik kopiuje z pliku lokalnego kod raportu i wkleja do nowoutworzonego raportu.

1.1.3 Użytkownik zapisuje zmiany.

UC1.2 Ręczne uruchomienie raportu pobierającego pakiety

Sytuacja: Użytkownik uruchomił raport w celu pobrania pakietu.

Przebieg:

1.2.1 Raport przedstawia listę pakietów które użytkownik może pobrać z systemu.

1.2.2 Użytkownik zaznacza pakiet lub pakiety, które chce pobrać.

1.2.3 Użytkownik uruchamia pobieranie pakietów przez naciśniecie odpowiedniego przycisku.

Alternatywy:

1.2.1 Raport nie odnalazł podanych pakietów lub z powodów błędów nie było możliwe pobranie pakietów. Zostaje wyświetlona lista błędów użytkownikowi.

UC1.3 Ustawienie automatycznego uruchomienia raportu pobierającego pakiety

Sytuacja: Użytkownik chce zdefiniować automatyczne, okresowe uruchamianie raportu.

Przebieg:

1.3.1 Użytkownik uruchamia systemową transakcję *SM36*.

1.3.2 Użytkownik tworzy nowe zdarzenie okresowe i definiuje nowe parametry uruchomieniowe dla niego.

1.3.3 Użytkownik przypisuje wcześniej utworzony raport do pobierania pakietów do zdarzenia okresowego.

1.3.4 Użytkownik definiuje parametry uruchomieniowe raportu (nazwy pobieranych pakietów oraz ścieżki w systemie plików, gdzie pakiety mają być zapisywane.

Alternatywy:

1.3.1 Raport nie został utworzony, wróć do UC1.1.

2. Operacje przeprowadzane po stronie aplikacji desktopowej.

UC2.1 Wyświetlenie pakietu systemu *SAP R/3*

Sytuacja: Użytkownik chce obejrzeć pobrany pakiet systemu *SAP R/3*

Przebieg:

2.1.1 Użytkownik uruchomił aplikację desktopową.

2.1.2 Użytkownik naciska przycisk służący do importowania pakietów.

2.1.3 Użytkownik w nowo otwartym oknie wskazuje lokalizację w systemie plików pobranego wcześniej pakietu.

2.1.4 Aplikacja wczytuje pakiet z podanej lokalizacji.

2.1.5 Aplikacja wyświetla użytkownikowi elementy składowe pakietu oraz ich szczegóły i właściwości.

Alternatywy:

2.1.1 Użytkownik nie posiada zainstalowanej maszyny wirtualnej języka *JAVA*. Użytkownik musi pobrać ´środowisko uruchomieniowe.

2.1.4 Podana lokalizacja jest błędna: ścieżka nie istnieje lub w podanej ścieżce nie istnieje prawidłowa wersja pakietu systemu *SAP R/3*. Wróć do 2.1.3 i/lub UC1.2.

UC2.2 Generowanie dokumentacji pakietu systemu *SAP R/3*

Sytuacja: Użytkownik pobrał z systemu *SAP R/3* pakiety i zaimportował je do aplikacji.

Przebieg:

2.2.1 Użytkownik zaznacza wybrany pakiet, który wcześniej zaimportował.

2.2.2 Użytkownik naciska przycisk służący do generowania dokumentacji pakietów.

2.2.3 Użytkownik wybiera rodzaj generowanej dokumentacji.

2.2.4 Użytkownik wybiera elementy wchodzące w skład pakietu, dla których chce wygenerować dokumentację.

2.2.5 Użytkownik wybiera ścieżkę w systemie plików, gdzie nowoutworzona dokumentacja będzie zapisana.

2.2.6 Aplikacja generuje dokumentacje techniczną i zapisuje ją w podanej lokalizacji.

Alternatywy:

2.2.4 Użytkownik nie wybrał żadnego elementu wchodzącego w skład pakietu. Dokumentacja nie może zostać wygenerowana.

2.2.5 Użytkownik podał błędną ścieżkę lub w podanej lokalizacji nie ma wystarczająco wolnego miejsca na zapisanie dokumentacji.

2.2.6 Aplikacja zwraca błędy podczas generowania dokumentacji. Wróć do 2.2.3.

UC2.3 Generowanie metryk obiektowych dla pakietu systemu *SAP R/3*

Sytuacja: Użytkownik pobrał z systemu *SAP R/3* pakiety i zaimportował je do aplikacji.

Przebieg:

2.3.1 Użytkownik zaznacza wybrany pakiet, który wcześniej zaimportował.

2.3.2 Użytkownik naciska przycisk służący do generowania metryk obiektowych pakietów.

2.3.3 Użytkownik wybiera rodzaj generowanych metryk.

2.3.4 Użytkownik wybiera elementy wchodzące w skład pakietu, dla których chce wygenerować dokumentację.

2.3.6 Aplikacja generuje metryki obiektowe i wyświetla je użytkownikowi.

Alternatywy:

2.3.4 Użytkownik nie wybrał żadnego elementu wchodzącego w skład pakietu. Dokumentacja nie może zostać wygenerowana lub pakiet nie posiada żadnych elementów obiektowych (klasy, interfejsy).

Opis procedur przedstawionych w modelach przypadków użycia zawierają kolejne rozdziały artykułu.

4. Pobieranie pakietów

Program do pobierania pakietów został napisany w języku programowania *ABAP*, podstawowym języku programowania wykorzystywanym przez system *SAP R/3*. Za jego pomocą można odwoływać się do składowych systemu oraz baz danych zintegrowanych z systemem. Ponieważ jednak *ABAP* do uruchomienia wymaga *SAP NetWeaver Application Server* nie został użyty przy tworzeniu całego projektu.

Zadaniem raportu jest pobranie z systemu pakietu programistycznego wraz ze wszystkimi jego składowymi i zapisanie na dysku lokalnym użytkownika. Argumentami wejściowymi dla raportu są:

- nazwa pakietu który będzie pobierany,
- ścieżka w lokalnym systemie plików użytkownika gdzie pakiet i jego składowe zostaną zapisane,

Efektem wyjściowym raportu jest zbiór plików i katalogów utworzonych w systemie plików będących lokalną wersją pakietu systemu. Do zapisywania elementów pakietu w systemie plików wykorzystywany jest standardowy moduł funkcyjny *GUI_DOWNLOAD*. Do przetwarzania tabel wewnętrznych i struktur na obiekty XML wykorzystywana jest standardowa systemowa klasa *CL_XML_DOCUMENT*. Podczas zapisywania pakietu w systemie plików odwzorowywana jest struktura elementów pakietu. Dla każdej grupy elementów tworzone są niezależne drzewa katalogowe.

5. Aplikacja do wyświetlania danych pakietów

W katalogu programu zostały umieszczone cztery podkatalogi: *repository*, *conf*, *img* oraz *doc*. Katalog *repository* stanowi repozytorium dla importowanych pakietów. Pakiety podczas importowania zostają skopiowane do tego katalogu i późniejsze wszelkie odwołania dokonywane są do pakietów umieszczonych w tym katalogu. Katalog *conf* (configuration) zawiera *XML-OWA*¸ konfigurację aplikacji. Katalog *img* (images) stanowi repozyto-

rium dla plików graficznych wykorzystywanych przez aplikację. Zawiera podkatalog *packagetree*. W katalogu tym znajdują się trzy podkatalogi: *small, medium, large*. W każdym z nich znajduje się zestaw ikon wykorzystywanych przez aplikację podczas wyświetlania drzewa pakietów. Aktualnie wykorzystywany zestaw jest zdefiniowany w pliku konfiguracyjnym. Katalog *doc* (documentation) stanowi repozytorium dla dokumentacji, która zostanie wygenerowana dla poszczególnych pakietów.

W celu możliwości wykorzystywania danych, dostarczonych przez oprogramowanie *raport* pobierający elementy pakietu, utworzono specjalną klasę w języku *JAVA*. Jej zadaniem jest zamodelowanie danych zawartych w plikach *XML* pakietu w postaci obiektów języka *JAVA*. Takie podejście pozwala na jednokrotne wczytanie danych z plików *XML* a następnie pozwala w całej aplikacji na odwoływanie się do nich zamiast odczytywania danych bezpośrednio z pliku. Każdy z tych obiektów może agregować w sobie obiekty reprezentujące inne elementy.

Za zarządzanie pakietami w całej aplikacji odpowiada klasa *PackagesManager*. Do jej zadań należy wczytywanie pakietów znajdujących się w repozytorium oraz importowanych przez użytkownika. Jeżeli pakiet jest importowany przez użytkownika to w pierwszej kolejności następuje jego skopiowanie do katalogu będącego głównym repozytorium pakietów. Następnie sprawdzane jest czy istnieją jakiekolwiek pliki dla danej grupy elementów pakietu. Jeżeli tak to następuje ich wczytanie. Zajmują się tym specjalne klasy zwane *loader'ami*. Dla każdej grupy elementów została stworzona jedna taka klasa. Następnie dla każdego elementu wchodzącego w skład tej grupy następuje sprawdzenie czy istnieją jakiekolwiek pliki reprezentujące ten element. Jeżeli istnieją to następuje wczytanie tych plików i przekształcenie ich do odpowiedniego obiektu reprezentującego dany element.

Do tworzenia poszczególnych elementów wchodzących w skład danej grupy wykorzystano wzorzec projektowy *metoda wytwórcza*. Jest to jeden z konstrukcyjnych wzorców projektowych. Wykorzystanie tego wzorca pozwala na konstruowanie nowych obiektów, które będą wchodziły w skład pakietu zależnie od aktualnie odczytanych danych z plików *XML*. Do przetwarzania plików *XML* na konkretne obiekty została wykorzystana dodatkowa biblioteka języka *JAVA* o nazwie *XStream*. Powodem wykorzystania tej biblioteki w projekcie były następujące jej cechy:

- Prostota użycia – dzięki temu został zaoszczędzony czas na naukę obsługi biblioteki jak również czas na napisanie kodu ją wykorzystującego.
- Brak potrzeby generowania dodatkowych mapowań – tworzenie mapowań zostało ograniczone do minimum. Nie potrzeba tworzyć dokładnych mapowań pomiędzy elementami pliku *XML* a atrybutami klas.
- Wydajność´ – obiekty generowane są szybko i nie obciążają mocno systemu operacyjnego. Jest to istotna cecha w przypadku bardzo dużej liczby pakietów do wczytania i\lub pakietów składających się z bardzo dużej liczby elementów (rzędu kilkuset, kilku tysięcy).
- Brak konieczności modyfikacji obiektów modelujących pakiet dla potrzeb biblioteki – obiekty mogły pozostać w niezmienionej strukturze ponieważ biblioteka pozwala na wstrzykiwanie danych do atrybutów prywatnych klas oraz na serializację pól typu *final* oraz klas wewnętrznych.
- Dobry system obsługi błędów – pozwala na szybkie odnalezienie błędów w plikach *XML* oraz podczas mapowań.
- Konfigurowalne strategie konwersji – możliwe jest zdefiniowanie własnych strategii konwersji plików *XML* na obiekty i odwrotnie. Można zdefiniować jakie struktury danych i typy danych będą używane podczas konwersji.

5.1. Tworzenie dokumentacji technicznej pakietu

Aplikacja umożliwia wygenerowanie dokumentacji technicznej dla każdego pakietu. Generowaniem dokumentacji zajmują się odpowiednie klasy do tego oddelegowane. Dokumentacja wygenerowana dla danego pakietu jest przechowywana w katalogu zdefiniowanym w pliku konfiguracyjnym jako repozytorium dokumentacji technicznej. Podstawowym typem dokumentacji jaki można generowany jest dokumentacja w formacie *HTML*. Przypomina ona dokumentacje generowaną dla pakietów języka *JAVA* za pomocą aplikacji *Javadoc*. Powodem takiego podobieństwa jest fakt, że wielu użytkowników systemu *SAP R/3* a zwłaszcza programistów jest zaznajomionych z tego typem dokumentacją.

W pierwszej kolejności następuje wygenerowanie głównej strony dokumentacji pakietu. Strona ta ma postać trzech ramek. W północno-zachodniej ramce znajduje się podstroną z linkami, które sterują widokiem ramki południowo-zachodniej. Linki te prezentują wszystkie główne grupy elementów pakietu. Naciśniecie któregoś z nich powoduje przeładowanie ramki południowo-zachodniej i wyświetlenie w niej jedynie listy linków do elementów, które należą do danej grupy elementów. Naciśnięcie któregoś z linków w ramce południowo-zachodniej powoduje, że w trzeciej, centralnej ramce zostają wyświetlone dane szczegółowe dotyczące wybranego elementu. Możliwe jest również wyświetlenie w ramce południowo-zachodniej wszystkich elementów wchodzących w skład pakietu. Do zapisywania plików *HTML* w systemie plików została wykorzystana dodatkowa biblioteka języka *JAVA Commons IO*.

5.2. Generowanie metryk dla pakietu

Kolejną funkcjonalnością udostępnianą przez aplikację jest możliwość´ generowania metryk oprogramowania dla elementów pakietu. Mogą one być tworzone jedynie dla elementów zawierających kod źródłowy języka ABAP. Metryki [5] generowane przez aplikację można podzielić´ na metryki dające się zastosować´ do każdego elementu pakietu zawierającego kod źródłowy języka *ABAP* oraz takie które dają się zastosować jedynie do jego obiektowych elementów. Do elementów obiektowych zaliczamy: klasy, interfejsy, kontrolery Web Dynpro.

Do metryk, które można wygenerować´ za pomocą aplikacji zaliczamy:

- złożoność cyklomatyczną McCabe'a (CC)
- liczba efektywnych linii kodu (eLOC)
- Decision Density (DD)
- zestaw metryk MOOD (Metrics for Object-Oriented Design):
  - Attribute Hiding Factor (AHF)
  - Method Hiding Factor (MHF)
  - Attribute Inheritance Factor (AIF)
  - Method Inheritance Factor (MIF)
- Zestaw metryk CK (Chidamber'a i Kemerer'a)
  - Response For a Class (RFC)
  - Weighted Methods per Class (WMC)
  - Lack of Cohesion of Methods (LCOM)

Pierwsze dwie z wymienionych metryk mają zastosowanie do wszystkich elementów, natomiast pozostałe jedynie do elementów obiektowych. W przypadku metryki *LCOM* do implementacji została wybrana jej wersja zwana *LCOM3*.

Pierwszą z implementowanych metryk jest metryka złożoności cyklomatycznej McCa-be'a. Metryka ta opiera się na mierzeniu liczby niezależnych ścieżek przebiegu programu, co bezpośrednio przekłada się na łatwość jego przetestowania. Podczas obliczania metryki stosowany jest wzór: CC = d + 1, gdzie d to liczba węzłów decyzyjnych, w których podejmowana jest decyzja o charakterze binarnym (tak/nie). Decyzjami są określane deklaracje warunkowe języka *ABAP*. Listę takich deklaracji przedstawia tabela 1.

Tab. 1. Wpływ konstrukcji języka ABAP na zmianę wartości metryki CC

Konstrukcja	Wpływ na CC	Wyjaśnienia
IF	+1	Deklaracja IF jest pojedynczą decyzją
CHECK	+1	Deklaracja CHECK jest decyzją równoważną IF
ELSEIF	+1	Deklaracja ELSEIF dodaje nową decyzję do deklaracji IF
CASE WHEN	+1 dla każdej deklaracji WHEN	Każda deklaracja WHEN dodaje nową decyzję
LOOP… WHERE	+1	Decyzja występuje podczas ograniczania zakresu poszukiwania
DO... ENDDO	0	Brak decyzji w pętli
WHILE...ENDWHILE	+1	Występuje decyzja w warunku pętli
CATCH	+1	Każda deklaracja CATCH dodaje nową ścieżkę wykonania

Została przyjęta zasada, że dodatkowe operatory logiczne występujące w warunku nie wpływają na wartość metryki. Metryka *CC* wyznaczana jest dla każdego z elementów z osobna. Wyjątek stanowią klasy, ponieważ w ich przypadku metryka CC wyznaczana jest dla każdej metody wchodzącej w ich skład z osobna. Drugi wyjątek stanowią interfejsy, ponieważ nie posiadają zaimplementowanych metod. Po wygenerowaniu metryki *CC* następuje interpretacja jej wartości. Złożoność elementu określa się na podstawie tabeli 2 natomiast prawdopodobieństwo wystąpienia błędu w elemencie określane jest na podstawie tabeli 3.

Tab. 2. Określanie złożoności elementu na podstawie wartości metryki CC

Wartość CC	Typ struktury elementu	Zagrożenie
1 - 4	Prosta struktura elementu	Bardzo niskie
5 - 10	Element stabilny, o dobrej strukturze	Niskie
11 - 20 ´	Element złożony	Średnie
21 - 50	Element bardzo złożony, zbyt skomplikowana struktura	Wysokie
> 50	Element niestabilny, struktura bardzo złożona	Bardzo wysokie

Tab.3. Określanie prawdopodobieństwa występowania błędu w elemencie na podstawie wartości CC

Wartość CC	Prawdopodobieństwo występowania błędu
1 - 10	5%
11 - 20	10%
21 - 30	20%
31 - 50	30%
51 - 100	50%
> 100	65%

Tak otrzymane rezultaty są wyświetlane użytkownikowi. W przypadku klas wynik prezentowany jest dla każdej metody z osobna jak również dla całej klasy. Złożoność całej klasy wyznaczana jest na podstawie wzoru:

$$S = \sum_{i=0}^{n} M(CCi) + \sum M + 1 \quad (1)$$

gdzie n oznacza liczbę metod w klasie a M(CCi) wartość metryki *CC* dla metody M. Wzór ten wynika z faktu, że wartość metryki *CC* wyznacza się na podstawie liczby deklaracji decyzyjnych elementu plus 1. W przypadku klasy należy zsumować wartości metryki każdej jej metody a następnie odjąć liczbę będącą sumą metod w klasie i dodać jeden.

Następną wyznaczaną metryką jest Liczba efektywnych linii kodu (*ELOC*). Jest to jedna z najprostszych metryk generowanych przez aplikację. Jej wartość oblicza się na podstawie liczby linijek kodu z jakich składa się element. Podczas zliczania linii, pomijane są linie puste oraz linie komentarzy ponieważ nie zawierają one w sobie żadnych instrukcji wykonywalnych języka ABAP.

Ostatnią metryką możliwą do wygenerowania dla każdego elementu jest Decision Density (DD). Oblicza się ją na podstawie dwóch wcześniej wymienionych metryk: złożoności cyklomatycznej McCabe'a (CC) oraz liczbie logicznych linii kodu (LLOC) za pomocą wzoru:

$$DD = \frac{CC}{LLOC} \quad (2)$$

Pierwszą zaimplementowaną metryką obiektową ze zbioru metryk MOOD (Metrics for Object-Oriented Design) jest metryka o nazwie Attribute Hiding Factor. Wartość tej metryki jest wyliczana ze wzoru:

$$AHF = \frac{\sum_{i=1}^{TA} A_h(Ci)}{\sum_{i=1}^{TA} A_d(Ci)} = 1 - \frac{\sum_{i=1}^{TA} A_v(Ci)}{\sum_{i=1}^{TA} A_d(Ci)} \quad (3)$$

Gdzie TA oznacza całkowitą liczbę atrybutów klasy C, *Ad* dowolny atrybut klasy C, *Ah* atrybut ukryty klasy C, *Av* atrybut widoczny klasy C oraz *Ad = Ah + Av* .
Metryka ta wyznaczana jest dla każdej klasy z osobna jak również dla wszystkich klas w pakiecie. Nie posiada ona jednostki miary. Jej wartość jest procentowym udziałem atrybutów ukrytych w klasie lub pakiecie.

Drugą zaimplementowana metryką obiektową ze zbioru metryk MOOD (Metrics for Object-Oriented Design) jest metryka o nazwie Method Hiding Factor. Wartość tej metryki jest wyliczana ze wzoru:

$$MHF = \frac{\sum_{i=1}^{TM} M_h(Ci)}{\sum_{i=1}^{TM} M_d(Ci)} = 1 - \frac{\sum_{i=1}^{TM} M_v(Ci)}{\sum_{i=1}^{TM} M_d(Ci)} \quad (4)$$

Gdzie TM oznacza całkowitą liczbę metod klasy C, *Md* dowolną metodę klasy C, *Mh* metodę ukrytą klasy C, *Mv* metodę widoczną klasy C oraz *Md = Mh + Mv* .

Trzecią zaimplementowaną metryką obiektową ze zbioru metryk MOOD jest metryka o nazwie Attribute Inheritance Factor. Wartość tej metryki jest wyliczana ze wzoru:

$$AIF = \frac{\sum_{i=1}^{TA} A_h(Ci)}{\sum_{i=1}^{TA} A_d(Ci)} = 1 - \frac{\sum_{i=1}^{TA} A_n(Ci)}{\sum_{i=1}^{TA} A_d(Ci)} \quad (5)$$

Gdzie TM oznacza całkowitą liczbę atrybutów klasy C, *Ad* dowolny atrybut klasy C, *Ai* atrybut odziedziczony przez klasę C z nadklasy, *An* atrybut niedziedziczony czyli zdefiniowany w klasie C oraz *Ad = Ai + An*.

Czwartą zaimplementowaną metryką obiektową ze zbioru metryk MOOD jest metryka o nazwie Method Inheritance Factor. Wartość tej metryki jest wyliczana ze wzoru:

$$MIF = \frac{\sum_{i=1}^{TM} M_h(Ci)}{\sum_{i=1}^{TM} M_d(Ci)} = 1 - \frac{\sum_{i=1}^{TM} M_n(Ci)}{\sum_{i=1}^{TM} M_d(Ci)} \quad (6)$$

Gdzie TM oznacza całkowitą liczbę metod klasy C, *Md* dowolną metodą klasy C, *Mi* metodę odziedziczoną przez klasę C z nadklasy, *Mn* metodę nieodziedziczoną, czyli zdefiniowaną w klasie C oraz *Md =Mi +Mn*.

Pierwszą zaimplementowaną metryką obiektową ze zbioru metryk CK (Chidamber'a i Kemerer'a) jest metryka o nazwie Response For a Class. Jej wartość jest obliczana na podstawie wzoru:

$$RFC = \sum_{i=1}^{TM} M_i + \sum_{i=1}^{TS} \sum_{j=1}^{TC_j} M_j \quad (7)$$

Gdzie TM oznacza całkowitą liczbę metod klasy C, TS liczbę podklas klasy C, *TCj* liczbę metod w podklasie j.

Drugą zaimplementowaną metryką obiektową ze zbioru metryk CK jest metryka o nazwie Weighted Methods per Class. Jej wartość jest obliczana na podstawie wzoru:

$$WMC = \sum_{i=1}^{TM} [M_i x\ CC(M_i)] \quad (8)$$

Gdzie TM oznacza całkowitą liczbę metod klasy C, *Mi* wagę metody klasy C, CC złożoność cyklomatyczna metody *Mi* . W przypadku gdy waga dla każdej metody wynosi 1 to wartością tej metryki jest suma złożoności cyklomatycznych metod danej klasy.

Trzecią zaimplementowaną metryką obiektową ze zbioru metryk CK jest metryka o nazwie Lack of Cohesion of Methods. Do implementacji została wybrana trzecia wersja tej metryki o nazwie *LCOM3*. Jej wartość jest obliczana na podstawie wzoru:

$$LCOM = \frac{TM - \frac{\sum_{i=1}^{TA} MA_i}{TA}}{TM-1} \quad (9)$$

Gdzie TM oznacza całkowitą liczbę metod klasy C, TA liczbę atrybutów w klasie C, *MAi* liczbę metod klasy C odwołujących się do atrybutu *Ai* .

6. Zakończenie

System składa się z dwóch elementów. Pierwszy napisany w języku ABAP odpowiedzialny jest za pozyskiwanie i obróbkę pakietów z SAP R/3. Drugi jest aplikacją umożliwiającą wizualizację , generowanie dokumentacji technicznej pakietów i metryk oprogramowania dla jego składowych.

Wszystkie części składowe projektu zostały stworzone z myślą o wygodzie użytkownika, przez co ich wykorzystywanie w codziennej pracy będzie łatwe. Raport do pobierania pakietów jest prosty w użyciu, interfejs użytkownika przypomina standardowe raporty systemowe. Aplikacja do przetwarzania pobranych danych posiada intuicyjne menu. System w wersji alfa został dostarczony grupie użytkowników do testowania. Skład grupy testującej stanowili konsultanci i programiści pracujący od wielu lat z systemem *SAP R/3*. Został on bardzo dobrze przyjęty a cenne uwagi testerów pozwoliły na jeszcze lepsze

dopracowanie projektu. Uwagi były głównie skierowane do elementów *GUI* jak również danych elementów pakietu jakie wykorzystywane są podczas codziennej pracy konsultanta/programisty.

Podczas tworzenia projektu największa trudność sprawiło odnalezienie sposobu na odczytanie danych z systemu *SAP R/3*. Wynikało to z bardzo obszernej dokumentacji systemu jak również występujących w niej braków na temat opisu pakietu i jego elementów składowych, zwłaszcza miejsc przechowywania danych.

Projekt jest cały czas rozwijany. Dodawane są nowe funkcjonalności i możliwości takie jak:

- możliwość generowania dokumentacji w postaci plików *PDF* (Portable Document Format),
- zintegrowanie system *CVS* lub *SVN* z projektem by umożliwić prowadzenie historii zmian w pakietach,
- pobieranie danych z repozytorium *MIME*,
- możliwość pobierania pakietu bez konieczności bezpośredniego logowania do systemu i uruchamiania raportu do pobierania danych pakietu.

Literatura

1. Błażewicz J., Jankowski M., Klaus R.: Organizacja przedsięwzięcia wdrożeniowego SAP R/3, Komputerowo Zintegrowane Zarządzanie pod redakcją Ryszarda Knosali, WNT W-wa 2003, tom 1, str. 93-100, ISBN 83-204-2809-2.
2. Ehrensperger W., Staader J.: New Dynamic Test Tools for ABAP Developers, SAP AG, 2004.
3. Johnson R., Gamma E., Helm R., Vlissides J.: Inżynieria oprogramowania: Wzorce projektowe. WNT, Warszawa, 2008.
4. Helfen M., Lauer M. Trauthwein M.: Testing SAP Solutions, Gaileo Press, 2007.
5. Kan S. Metryki i modele w inżynierii jakości oprogramowania. Wydawnictwo Naukowe PWN, 2006.
6. Klaus R., Maćkowski T.: Automatyczne testy funkcjonalne w aplikacji SAP, Komputerowo Zintegrowane Zarządzanie pod redakcją Ryszarda Knosali, tom I, str. 523-531, Oficyna Wydawnicza PTZP, Opole, 2009, ISBN 978-83-923797-7-5.
7. SAP AG. Sap r/3 documentation - dictionary objects: Data elements. [on-line] http://help.sap.com/saphelp_nw70/helpdata/EN/90/8d72feb1af11d194f600a0c929b3c3/content.htm.

Dr inż. Rafał KLAUS
Instytut Informatyki
Politechnika Poznańska
60-965 Poznań, ul. Piotrowo 2
e-mail: rafal.klaus@cs.put.poznan.pl

Mgr inż. Bartosz ŻYLIŃSKI
Business Consulting Center Sp. z o.o.
62-002 Poznań- Złotniki, ul. Krzemowa 1
www.bcc.com.pl