

Dariusz R. AUGUSTYN, Łukasz WARCHAŁ
Politechnika Śląska, Instytut Informatyki

ADAPTACJA NARZĘDZIA TYPU CASE DO SPECYFIKI PROCESU WYTWARZANIA OPROGRAMOWANIA

Streszczenie. Artykuł porusza zagadnienie dostosowania narzędzia typu CASE do specyfiki przyjętego procesu wytwarzania oprogramowania. Na przykładzie systemu Enterprise Architect opisano sposób wykorzystania standardowego mechanizmu profili UML i dodatkowego, własnego, modułu programowego (wtyczki), co sprawia, że narzędzie CASE wspiera także te obszary procesu wytwórczego, które mają swoją specyfikę względem przyjętych standardów. Artykuł pokazuje podejście do problemu integracji narzędzia CASE z zewnętrznym systemem, wspomagającym organizację procesu wytwarzania oprogramowania.

Słowa kluczowe: proces wytwórczy, analiza, zarządzanie pracą, przypadki użycia, automatyzacja integracji, Enterprise Architect, SWDPS, wtyczka do EA

CUSTOMIZING CASE TOOL TO SPECIFIC SOFTWARE DEVELOPMENT PROCESS

Summary. This paper considers how to customize CASE tool to specific software development process. It outlines how to combine standard UML profile mechanism with custom software plug-in for Enterprise Architect to achieve better support of specific parts in development process. It also describes how using this concept a CASE tool can be integrated with other system supporting development process management, to gain a control over the process.

Keywords: development process, system analysis, workflow managements, use cases, automation of integration, Enterprise Architect, SWDPS, EA plug-in

1. Wprowadzenie

Wzrost liczby wytwarzanych systemów informatycznych spowodował pojawienie się licznych metodyk opisujących proces ich tworzenia. Większość z nich ma charakter ogólny

i jest jedynie zbiorem dobrych praktyk oraz zaleceń. Dopiero osadzenie ich w konkretnej rzeczywistości procesu wytwórczego sprawia, że nabierają praktycznego znaczenia. Narzędzia typu CASE (ang. *Computer-Aided Software Engineering*) mają za zadanie ułatwiać i wspierać dostosowanie przyjętej metodyki do specyfiki takiego procesu.

Przykładem takiego narzędzia CASE jest Enterprise Architect. W niniejszym artykule pokazano, jak dostosować to narzędzie, aby najlepiej wspierało przyjęty, specyficzny proces wytwarzania oprogramowania.

Artykuł zbudowany jest w następujący sposób.

W punkcie 2 opisano zaproponowane rozszerzenia narzędzia Enterprise Architect, pozwalające na jego użycie zgodnie z przyjętym sposobem analizy, wykorzystującym przyjęte formaty dokumentu wizji i specyfikacji wymagań systemowych oraz opartym na dostosowanych artefaktach typu cecha, przypadek użycia, wymaganie.

W punkcie 3 opisano pewne narzędzie wspomagania procesu wytwórczego – SWDPS. W szczególności omówiono rozszerzenie tego systemu pod kątem integracji pojęć w nim wykorzystywanych (zgłoszenia, problemy), z artefaktami powstałymi w procesie analizy (cechy, przypadki użycia, wymagania). Pokazano, w jaki sposób organizacja pracy w SWDPS (wyrażona przez zgłoszenia, problemy, zadania) może być sterowana/napędzana przez artefakty analizy (cechy, przypadki użycia).

W punkcie 4 przedstawiono implementację autorskiej wtyczki do narzędzia Enterprise Architect, która wspomaga wiązanie artefaktów analizy, o których mowa w punkcie 2, z odpowiadającymi im pojęciami z SWDPS, opisanymi w punkcie 3.

Użycie wtyczki sprawia, że utworzenie/modyfikacja artefaktów analitycznych w narzędziu CASE powoduje automatyczne utworzenie/modyfikację odpowiednich artefaktów wykorzystywanych w niezależnym narzędziu do zarządzania procesem wytwórczym. Taka integracja systemów pozwala na organizację procesu wytwórczego napędzanego artefaktami analizy, w szczególności napędzanego przypadkami użycia (ang. *use-case driven development process*).

2. Wsparcie procesu analizy – własne rozszerzenia funkcjonalności narzędzia Enterprise Architect

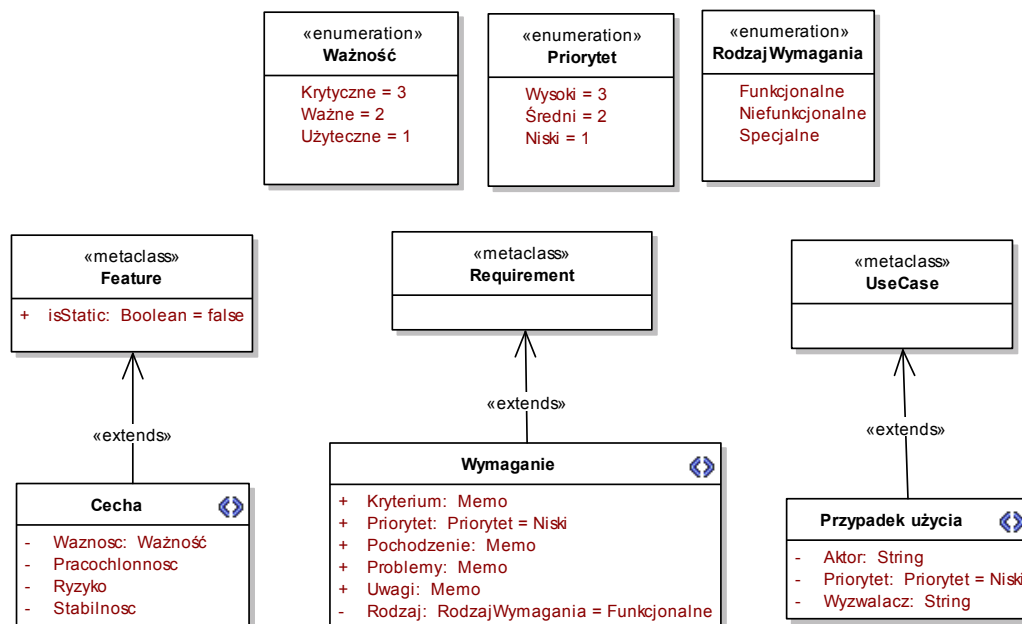
Metodyki wytwórcze oparte na podejściu UP (ang. *unified process*) [2] zakładają tworzenie opisu systemu informatycznego za pomocą ogólnego dokumentu wizji (ang. *vision document*) oraz szczegółowej specyfikacji wymagań systemowych (ang. *system requirement specification*). Na najwyższym poziomie abstrakcji opisu systemu często używane są tzw. cechy (ang. *features*), opisujące skrótowo, ogólnie, najważniejsze wymagania. Cechy te wymienia-

ne są w dokumencie wizji. Dokładny analityczny, opis systemów, oparty na diagramach z aktorami i przypadkami użycia (ang. *actors, use cases*) [1, 3], oraz szczegółowe wymagania (ang. *requirements*), zawarte są w dokumencie SWS (Specyfikacji Wymagań Systemowych).

Każda firma wytwarzająca oprogramowanie na ogół dokonuje własnej adaptacji zaleceń, zrębów faz i procesów oraz ról projektowych czy wzorców artefaktów proponowanych w oficjalnych dokumentach, opisujących bazową, standardową, rozbudowaną metodykę wytwórczą. Ma to podłoże ekonomiczne – wynika zarówno z aktualnej kultury organizacyjnej samej firmy, dojrzałości procesów wytwórczych, jak i rodzaju oraz „rozmiaru” poszczególnych projektów realizowanych w danej firmie. Adaptacje te mogą dotyczyć nawet pojedynczych własności poszczególnych artefaktów.

Profile w UML 2.0 [3, 5] pozwalają na dostosowanie elementów analizy/architektury/projektu do potrzeb opisu systemu informatycznego przyjętego w danej organizacji.

W rozwiązaniu opisywanym w niniejszym opracowaniu, zakładającym użycie popularnego narzędzia CASE Enterprise Architect (EA) [4], dzięki mechanizmom profili i znaczników (ang. *tagged values*) dokonano redefinicji cech, przypadków użycia, wymagań, czyli artefaktów analitycznych. Podobnie uczyniono w bardziej rozbudowanym rozwiązaniu opisywanym w [9], gdzie redefinicja dotyczyła artefaktów wykorzystywanych przy tworzeniu projektu technicznego systemu.

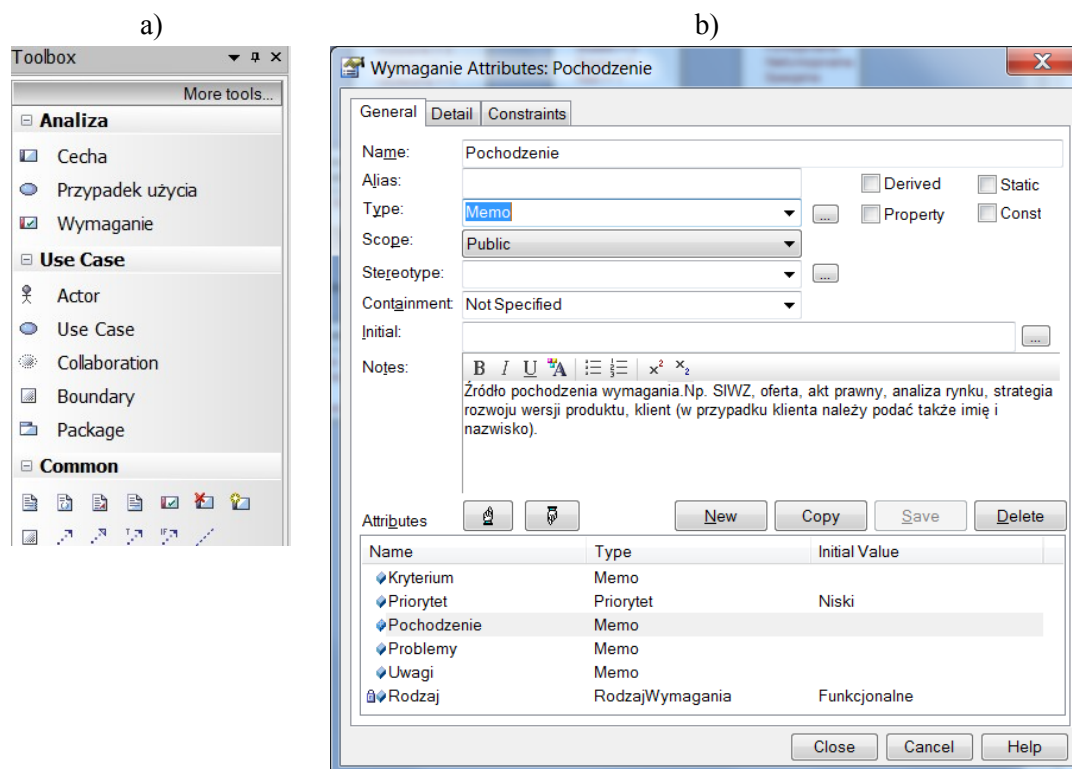


Rys. 1. Pakiet <<profile>> Analiza zawierający elementy dostosowane do przyjętego sposobu analizy
 Fig. 1. <<profile>> Analiza class packet including customized elements adopted for given analysis method

Wykorzystując EA, stworzono model klas (na potrzeby mechanizmu profili UML), określający nowe definicje artefaktów, tj. nowe typy wyliczeniowe oraz nowe elementy – „Ce-

cha”, „Wymaganie”, „Przypadek użycia” – rozszerzające standardowe elementy. Dzięki temu pojawiły się nowe atrybuty (polskojęzyczne) w nowych elementach (rys. 1, 2a).

Model klas z rys. 1 może być zapisany w pliku wymiany XML, a następnie zaimportowany do dowolnego projektu EA, przez co w nowym projekcie dostępny będzie nowy przybornik „Analiza”, a w nim zdefiniowane elementy (rys. 2a) .



Rys. 2. Wykorzystanie nowych artefaktów w Enterprise Architect: a) przybornik „Analiza” z nowymi elementami. b) nowe atrybuty elementu „Wymaganie” (pokazany domyślny opis atrybutu „Pochodzenie”)

Fig. 2. Usage new of analysis artifacts in Enterprise Architect: a) toolbox „Analiza” with new elements, b) new attributes of “Wymaganie” element (default description of “Pochodzenie” attribute is shown)

W omawianym rozwiązaniu rozszerzono definicje:

- „Cechy” o następujące atrybuty:
 - ważność (z punktu widzenia udziałowców/użytkowników),
 - pracochłonność (opcjonalna, bardzo wstępnie szacowana liczba osobodni realizacji),
 - ryzyko (wpływ niezrealizowania cechy na powodzenie całego przedsięwzięcia),
 - stabilność (pewność/zaufanie określenia/opisu cechy).
- „Przypadku użycia” o następujące atrybuty:
 - aktor (nazwa aktora głównego),
 - priorytet (opcjonalne określenie ważności; może być istotne dla uszeregowania ważności przypadków realizujących tę samą cechę),
 - wyzwalacz (opis zdarzenia uruchamiającego dany przypadek użycia).

- „Wymagania” o następujące atrybuty:
 - kryterium (jednoznaczny, słowny opis warunku uznania wymagania za spełnione),
 - pochodzenie (jak w opisie na rys. 2b),
 - priorytet,
 - problemy (wykaz problemów, które pozostały do wyjaśnienia na danym etapie wersji wymagania; w zatwierdzonej wersji wymagania, wartość atrybutu powinna być pusta),
 - rodzaj (typ wymagania),
 - uwagi.

Wyżej wymienione atrybuty dostępne są w EA jako znaczniki „Cech”, „Wymagań” czy „Przypadków użycia”.

EA pozwala na tworzenie szablonów raportów, np. w formacie RTF. EA udostępnia edytor szablonów RTF. Zdefiniowany szablon może być dołączony do projektu EA (i widoczny w zakładce *Resource* w węźle *Templates/RTF Style Templates*). Dla przykładu, na rys. 3 pokazano definicję szablonu raportu wspomagającego tworzenie dokumentu wizji, tzn. raportu będącego fragmentem dokumentu wizji w zakresie ewidencji cech.

```
package >
element >
```

Tabela cechy: {Element.Name}

Opis	{Element.Notes}
Identyfikator	{Element.Alias}
Status	{Element.Status}
Ważność	{Element.valueOf(Waznosc)}
Pracochłonność	{Element.valueOf(Pracochlonnosc)}
Ryzyko	{Element.valueOf(Ryzyko)}
Stabilność	{Element.valueOf(Stabilnosc)}

```
child elements >
< child elements
< element
< package
```

Rys. 3. Przykład prostego szablonu raportu RTF – definicja tabeli z wybranymi atrybutami cechy
 Fig. 3. Simple sample RTF report template – definition of table with selected attributes of “Cecha” element

W przedstawionym szablonie widać, że wykorzystano standardowe atrybuty cechy (wynikające z dziedziczenia po elemencie *Feature*, np. *Name* oraz te wprowadzone przez mechanizm profili i znaczników (np. *Stabilnosc*).

Stworzenie konkretnego raportu wymaga ustawienia wskaźnika myszy na pakiecie z cechami w oknie/zakładce *Project Browser*, po czym następują: wybór pozycji menu *Documentation/Rich Text Format (RTF) Report...* oraz wybranie nazwy szablonu raportu i podanie nazwy wynikowego pliku. Wynikowy raport powstaje po naciśnięciu przycisku *Generate*. Te

czynności prowadzą do stworzenia pliku RTF, opisującego w zadany sposób wszystkie cechy wskazanego pakietu. Plik RTF może być automatycznie włączany (przez odpowiedni kod sterujący {INCLUDETEXT ...}) do dokumentu wizji tworzonego przy użyciu MS Word. Zakłada się, że niektóre części dokumentu wizji są tworzone bezpośrednio w edytorze, a inne automatycznie generowane z narzędzia EA.

Podobnie jak w przypadku dokumentu wizji, istotne elementy dokumentu SWS są tworzone w EA i potem automatycznie włączane do dokumentu SWS. Na potrzeby tworzenia SWS zdefiniowano specjalne szablony raportów, które pozwalają na stworzenie raportów dotyczących:

- przypadków użycia (obejmujących diagram aktorów i przypadków użycia oraz szczegółowy opis każdego przypadku użycia, nawet z dokładnością do pojedynczego scenariusza),
- wymagań,
- mapowania cech na wymagania lub przypadków użycia (na potrzeby śledzenia artefaktów określonych w SWS i dokumencie wizji).

Zakres artykułu nie pozwala na szczegółowe opisanie ww. trzech definicji szablonów raportów (z pewnością bardziej złożonych od tego pokazanego na rys. 3).

Omawiane dostosowanie EA do potrzeb organizacji obejmuje dodatkowo:

- mechanizm automatycznego generowania unikalnych identyfikatorów cech, aktorów, przypadków użycia i wymagań (ustawienia dostępne przez opcje *Settings/Auto Name Counters...*),
- utworzenie tablic translacji wartości (stałych) z języka angielskiego na język polski (dotyczy to wartości standardowych atrybutów, np. wartość atrybutu *Status* jest równa *Zaakceptowane* zamiast *Accepted*).

3. SWDPS – system zarządzania procesem wytwórczym – rozszerzenie w zakresie obsługi cech oraz przypadków użycia/wymagań

W dużych organizacjach i złożonych projektach niezbędne staje się wykorzystanie systemów informatycznych wspomagających zarządzanie procesem wytwórczym. Istnieje wiele takich gotowych rozwiązań, jak np. Manits, Jira, Mylyn, Microsoft Team Foundation i inne. Zaletą takich rozwiązań jest dojrzałość produktu i dość powszechna znajomość jego funkcjonalności. Większość z nich ma pewne możliwości dostosowania do konkretnego procesu wytwórczego, ale może nie spełniać w pełni wymagań, jakie stawiane są przez organizację. Wtedy rozwiązaniem może być adaptowalny system własny (w pełni „kustomizowalny”).

Do takiej kategorii należy SWDPS, czyli system wspomagający działanie produkcji i serwisu, którego pomysłodawcą jest jeden ze współautorów niniejszego artykułu. SWDPS, opisany w [6, 7, 8], był i jest nadal wykorzystywany w niektórych oddziałach firm, takich jak: Computerland SA., Sygnity SA., ABG SA., Asseco Polska SA., Global Services Sp. z o.o.

3.1. Zakres zastosowań systemu SWDPS – wprowadzenie

System SWDPS (szczegółowo opisany w [7]) wspiera zarządzanie procesem wytwórczym w szerokim zakresie, obejmującym m.in.:

- ewidencję zgłoszeń (zewnętrznych, wewnętrznych),
- ewidencję problemów (typu: błąd krytyczny, błąd, usterka, zmiana, rozszerzenie),
- ewidencję zadań (typu: analiza, projektowanie, programowanie, testowanie, serwisowanie, dokumentowanie itd.),
- ewidencję wytwarzanych systemów informatycznych oraz modułów wchodzących w skład tych systemów (oraz obsługę wersji systemów i ich modułów),
- obsługę faz wytwarzania systemów,
- obsługę testów integracyjnych, systemowych, regresji (w szczególności definiowanie wielokrokowych scenariuszy testowych i zarządzanie procesem testowania na podstawie wybranych, automatycznie aktywowanych scenariuszt).

U podstaw działania SWDPS leży model danych oparty na pojęciach zgłoszenie, problem, zadanie. Zakłada się w nim, że zgłoszenie (na ogół generowane przez klienta) stanowi wysokopoziomowy opis zagadnienia (nierzadko nawet kilku zagadnień). W wyniku analizy opisu zgłoszenia następuje jego dekompozycja na problem/problemy, czyli element lub elementy opisane językiem analityczno-projektowym, wymagające podjęcia działań. Problem rozpisywany jest na zadania. Cykl życia każdego z elementów (zgłoszenia, problemu, zadania), określony przez *stan*, jest kontrolowany przez SWDPS. Zamknięcie (albo anulowanie) zadań pozwala na zamknięcie (lub anulowanie) problemu skojarzonego z zadaniami, a zamknięcie (albo anulowanie) problemów pozwala na zakończenie (albo anulowanie) zgłoszenia skojarzonego z problemami.

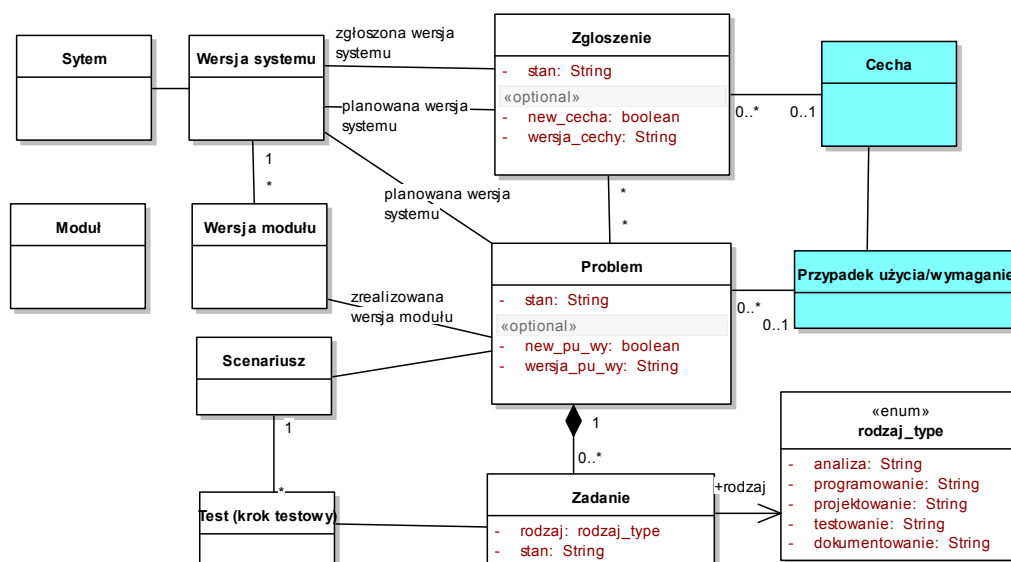
Z elementem zgłoszenie/problem/zadanie skojarzone są wskazane osoby rejestrujące, realizujące, zatwierdzające. Dla problemów oraz zadań podawane są szacunkowe i rzeczywiste (post factum) czasy realizacji oraz daty zakończenia realizacji. Podawane są wersje systemów i modułów, dla których problem wystąpił (tzn. został zgłoszony) oraz te, dla których faktyczna funkcjonalność określona w problemie została zrealizowana/naprawiona.

Baza danych systemu SWDPS stanowi podstawę do raportowania produktowego (np. jaka funkcjonalność (lista opisów problemów) jest zrealizowana w kolejnych przyrostach sys-

temów pomiędzy wskazanymi wersjami) czy procesowego (np. jaka jest czasochłonność poszczególnych typów zadań czy rozkład zadań w rozbiu na realizujące je osoby).

3.2. Integracja zgłoszeń i problemów z artefaktami analizy

Uproszczony model danych rozszerzonego systemu SWDPS został pokazany na rys. 4. Jest on zgodny z modelami prezentowanymi w [7, 8], ale uwzględnia nowe klasy – Cecha i Przypadek_uzycia/wymaganie. Cecha może odpowiadać „realizacyjnie” Zgłoszeniu, a Przypadek_uzycia/wymaganie – Problemowi. Zakłada się, że zarówno Cecha, jak i Przypadek_uzycia/wymaganie polegają zmianom, stąd będzie wynikać odwzorowanie 1:N „do” Zgłoszenia i Problemu. Cecha i Przypadek_uzycia/wymaganie to pojęcia z zakresu analizy, natomiast Zgłoszenie i Problem to pojęcia z zakresu zarządzania procesem wytwórczym w SWDPS.



Rys. 4. Model danych rozszerzonego funkcjonalnie systemu SWDPS

Fig. 4. Data model of functionally extended SWDPS

W ramach niniejszego podpunktu omówiony zostanie sposób integracji artefaktów analizy z elementami SWDPS.

W momencie powstania opisu cechy systemu w EA, w SWDPS zostaje zarejestrowane nowe zgłoszenie (rys. 5).

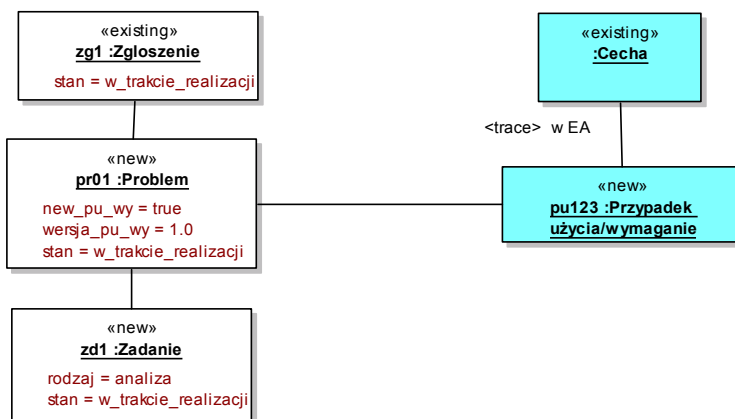


Rys. 5. Diagram obiektów – skojarzenie nowych elementów – cecha i zgłoszenie

Fig. 5. Object diagram – association of new elements – a feature and a request

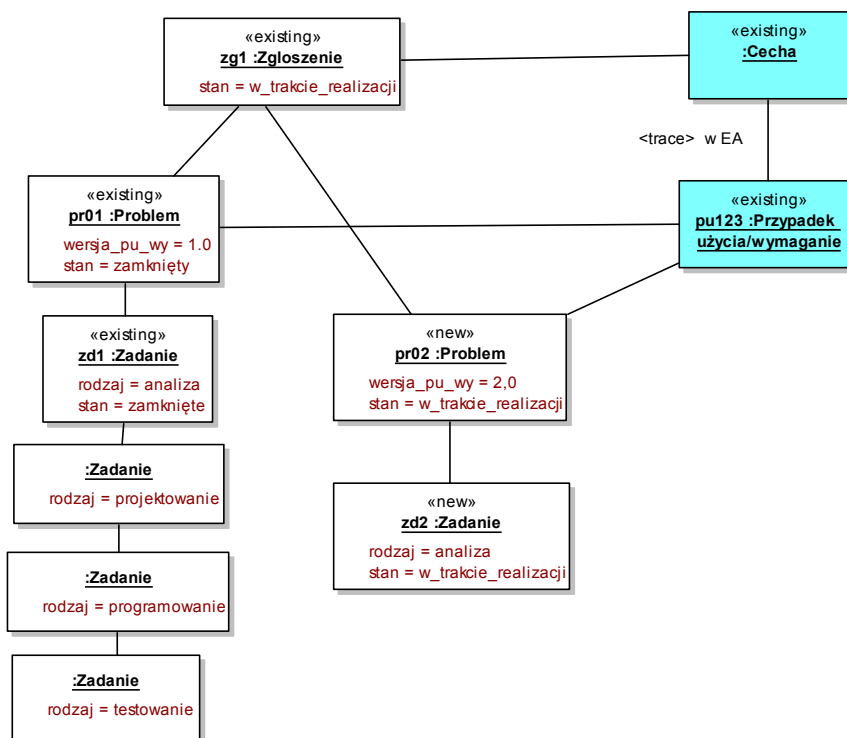
W ramach pracy nad zgłoszeniem kolejno powstają uszczegółowiające je przypadki użycia lub wymagania. Rysunek 6 pokazuje, jak nowy przypadek użycia *pu123*, powiązany z odpowiednią cechą w EA, zostaje skojarzony z odpowiednim problemem *pr01*. Przy okazji,

w ramach określania zadań związanych z problemem, definiowane jest zadanie analityczne (*stan*= 'w_trakcie_realizacji'), które odzwierciedla pracę nad opisywaniem przypadkiem użycia *pu123*.



Rys. 6. Nowy przypadek użycia w ramach istniejącej cechy
 Fig. 6. A new use case for the existing feature

W momencie zakończenia opracowywania przypadku użycia (zmiana stanu przypadku użycia na 'zaakceptowany'), odpowiednie zadanie analitycznie w SWDPS zostaje zamknięte, np. zamknięte zadanie *zd1*, związane z analizą przypadku użycia *pu123* (w wersji 1.0) z rys. 7.

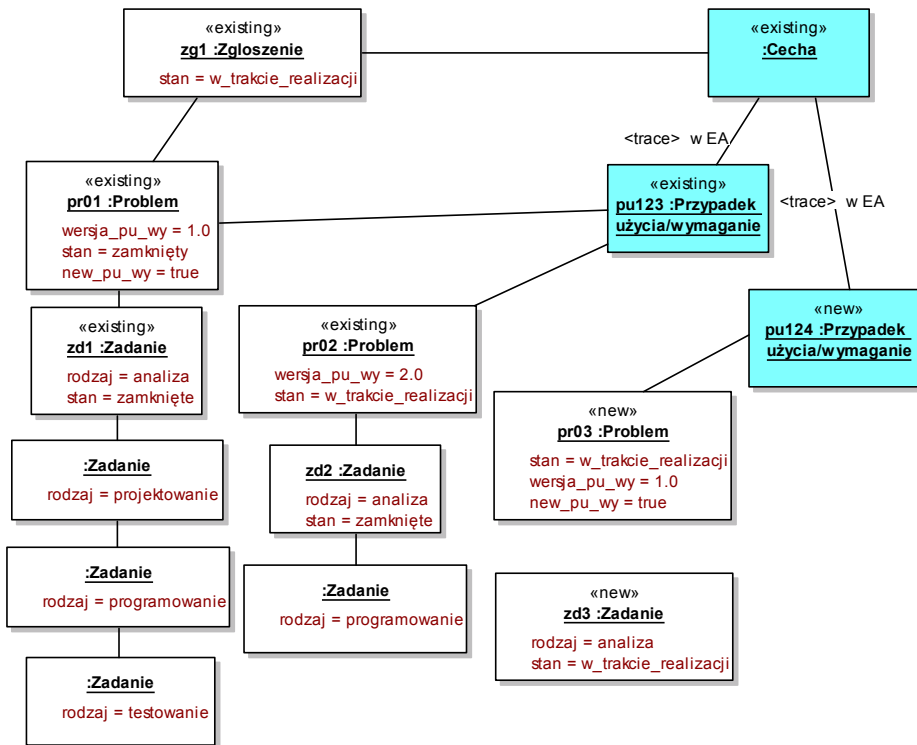


Rys. 7. Realizacja zmiany istniejącego przypadku użycia
 Fig. 7. Realization of change of the existing use case

Dalsze prace – tzn. prace realizacyjne, jak: projektowanie, programowanie, testowanie, wynikające z implementacji przypadku użycia *pu123*, obsługiwane poza EA – są ewidencjonowane w SWDPS w ramach zdań problemu *pr01* (rys.7).

W sytuacji gdy przypadek użycia określony w EA zostanie istotnie zmieniony (np. nastąpi zmiana czy dodanie nowego scenariusza), to w SWDPS powinien pojawić się nowy problem, w ramach którego będą realizowane zadania związane z obsługą nowej wersji tego przypadku użycia. Przykładem tego może być przypadek użycia *pu123* z rys. 7, gdzie poprzednia wersja tego przypadku (wersja 1.0 w problemie *pr01*) została zaimplementowana w ramach zadań w zamkniętym problemie *pr01*, a nowa wersja (wersja 2.0 w problemie *pr02*) jest obecnie opracowywana w ramach problemu *pr02* (w ramach zadania analitycznego *zd2*).

Oczywiście w ramach opracowywania cechy mogą powstawać nowe przypadki użycia i wymagania. Przykładowo, na rys. 8 pokazano obiekty SWDPS związane z obsługą cechy uszczegółowianej w EA przez dwa przypadki użycia: istniejący *pu123* (wersje 1.0 i 2.0) oraz nowy *pu124* (wersja 1.0).



Rys. 8. Kolejny, nowy przypadek użycia w ramach istniejącej cechy
Fig. 8. Yet another new use case for the existing feature

Opisany powyżej, rozszerzony model danych SWDPS, pozwala na skojarzenie elementów obiektowej analizy z pojęciami SWDPS. Te skojarzenia można utworzyć, bezpośrednio korzystając z GUI aplikacji SWDPS, ale jeszcze łatwiej uzyskać je automatycznie, korzystając z zaproponowanego rozszerzenia EA, tzw. plug-in-a do integracji EA-SWDPS, opisanego w następnym punkcie.

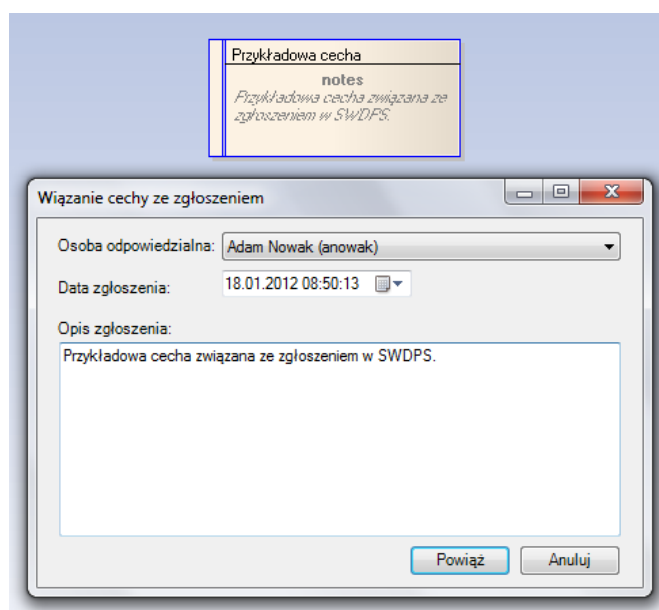
4. Integracja SWDPS i EA – plug-in do asocjacji zgłoszeń i problemów z artefaktami analizy definiowanymi w EA

Narzędzie CASE Enterprise Architect (EA) jest wykorzystywane w wielu firmach do wspomagania analizy i projektowania systemów informatycznych. Jedną z jego zalet jest łatwość dostosowania poszczególnych składników do specyfikacji procesu wytwarzania oprogramowania danego przedsiębiorstwa, co ilustrują przykłady pokazane w punkcie 2. Możliwe jest także dodanie do EA zupełnie nowej funkcjonalności dzięki mechanizmowi wtyczek (ang. *plug-in*) [10]. Opisany w podpunkcie 3.2 sposób powiązania artefaktów analizy z pojęciami używanymi w SWDPS został zautomatyzowany przy użyciu takiej właśnie wtyczki. Automatyzacja jest możliwa dzięki wykorzystaniu biblioteki DLL, dostarczanej przez twórców EA [10], która pozwala operować na elementach repozytorium projektu. Dzięki temu można dokonać asocjacji informacji zawartych w diagramach UML z odpowiadającymi im pojęciami SWDPS, odwzorowanymi w bazie danych.

W kolejnych podpunktach zaprezentowano przykłady użycia w EA omawianej wtyczki w celu aktualizacji bazy danych SWDPS, na podstawie informacji pochodzących z artefaktów analitycznych tworzonych w repozytorium projektu w EA.

4.1. Wiązanie cechy ze zgłoszeniem

Wiązanie cechy umieszczonej na diagramie UML ze zgłoszeniem w SWDPS odbywa się w momencie, gdy będąc w EA, wskazujemy wybraną cechę i wybieramy opcję *Powiąz ze zgłoszeniem* omawianej wtyczki. Następuje wtedy sprawdzenie, czy ta cecha posiada już odpowiadające jej zgłoszenie. Jeżeli nie, zostaje wyświetlone okno dialogowe, w którym użytkownik wypełnia niektóre dane nowego zgłoszenia. Część wypełniana jest automatycznie na podstawie informacji z cechy (np. autor cechy zostaje osobą odpowiedzialną za zgłoszenie, opis cechy staje się opisem zgłoszenia itp.). Przykład użycia wtyczki dla przykładowej cechy prezentuje rys. 9.



Rys. 9. Powiązanie nowej cechy ze zgłoszeniem w SWDPS, przy użyciu omawianej wtyczki do EA
 Fig. 9. Binding a new feature to a new SWDPS request using proposed EA plug-in

4.2. Wiązanie przypadku użycia z problemem

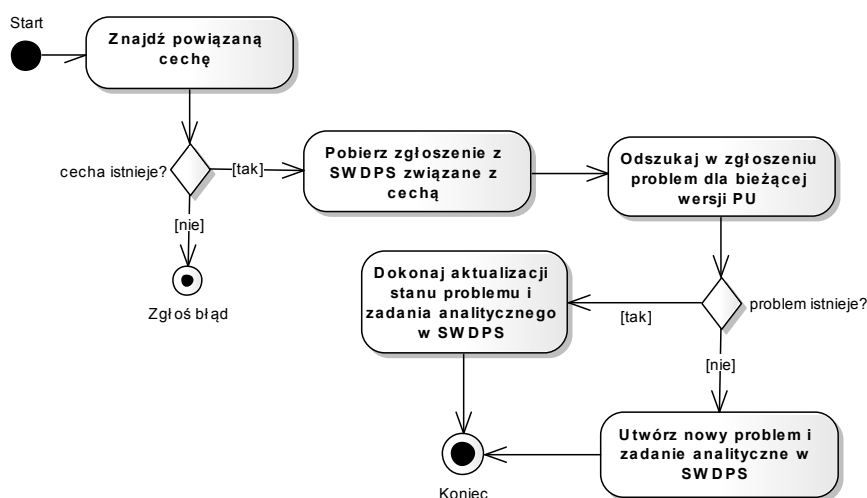
W ramach opracowywania wybranej cechy systemu następuje na ogół jej dekompozycja na kilka przypadków użycia. Każdy z przypadków użycia może zostać związany z problemem (a pośrednio z zadaniami realizacyjnymi – patrz rys. 6), co pozwala na monitorowanie w narzędziu SWDPS postępu prac nad jego analizą i realizacją. Automatyzację procesu wiązania zapewnia omawiana wtyczka do EA.

Związanie przypadku użycia z problemem w SWDPS jest nieco bardziej skomplikowane niż np. opisane wcześniej wiązanie cechy ze zgłoszeniem.

Związanie jest możliwe tylko dla tych przypadków użycia, które opisują działanie systemu w ramach którejś z cech – wymagane jest więc połączenie (ang. *link*) pomiędzy nimi, oznaczone stereotypem <<trace>>.

Później, już po związaniu, przypadek użycia może ulec zmianie i konieczna jest synchronizacja danych pomiędzy EA a SWDPS.

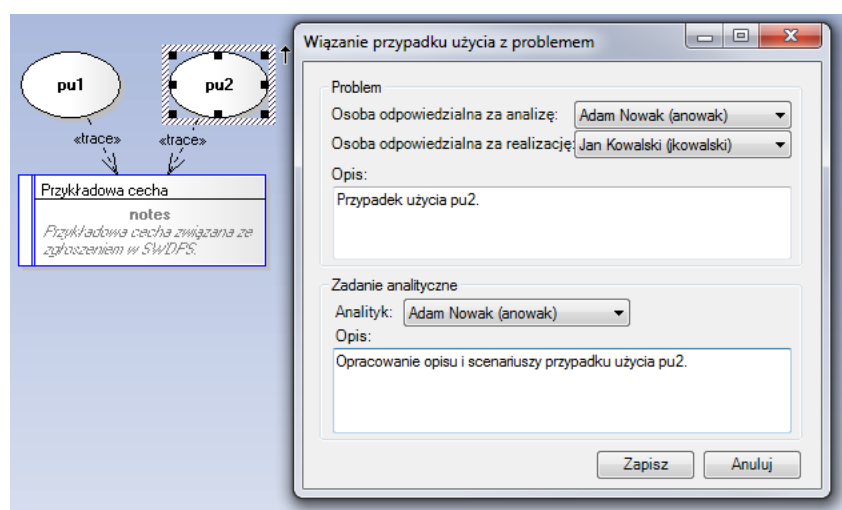
Proponowana wtyczka do EA realizuje powiązanie i synchronizację przypadku użycia z problemem w SWDPS wg algorytmu opisanego diagramem czynności z rys. 10.



Rys. 10. Diagram czynności przedstawiający algorytm synchronizacji powiązania przypadku użycia z problemem w SWDPS

Fig. 10. Activity diagram showing algorithm which synchronizes binding between use case and problem in SWDPS

Po uruchomieniu synchronizacji „na” wybranym przypadku użycia, w pierwszej kolejności następuje odszukanie związanej z nim cechy. W tym celu używane są jego połączenia oznaczone stereotypem `<<trace>>`. Jeśli znaleziono cechę, która ma powiązanie ze zgłoszeniem w SWDPS, wyszukiwany jest należący do niej problem, który odpowiada bieżącej wersji przypadku użycia. Jeżeli problem został znaleziony, badamy, czy nastąpiła zmiana stanu przypadku użycia. Jeśli został on w EA zamknięty w sensie analizy (stan ‘zaakceptowany’), następuje zamknięcie związanego z nim zadania analizy (patrz rys. 7). W przypadku gdy problem nie został znaleziony (zmieniła się wersja przypadku użycia), tworzony jest nowy problem wraz z zadaniem analizy.



Rys. 11. Przykład wiązania nowego przypadku użycia z problemem i zadaniem analizy w SWDPS

Fig. 11. Example of binding a new use case with problem and analyze task in SWDPS

5. Podsumowanie

W artykule opisano sposób dostosowania narzędzia CASE Enterprise Architect (EA) do elementów przyjętego procesu wytwórczego.

Opisano, jak rozszerzyć EA, aby zadośćuczynić przyjętej metodzie analizy (tzn. przyjętemu sposobowi opisu systemu przez cechy, przypadki użycia, wymagania, oraz sposobowi raportowania). Artykuł pokazuje, jak za pomocą mechanizmu profili i znaczników UML rozbudować artefakty analizy wg potrzeb.

Artykuł odnosi się również do pewnego systemu wspomagającego zarządzanie pracą w ramach procesu wytwórczego – SWDPS. Pokazano jak system ten rozszerzyć, aby sterowanie przebiegiem prac mogło wynikać z obsługi artefaktów analitycznych. Zaproponowane rozszerzenie EA – wtyczka (EA plug-in) – pozwala na automatyzację integracji systemu EA (wykorzystywanego w ramach analizy) z systemem SWDPS, wykorzystywanym do zarządzania pracą.

Artykuł opisuje konkretne rozwiązania programowe, jednak autorzy chcieliby dodatkowo zwrócić uwagę na pewne aspekty o charakterze uniwersalnym.

Część artykułu, dotycząca rozszerzania EA w zakresie analizy, pokazuje łatwość takiej rozbudowy, realizowanej głównie przez konfigurację tego narzędzia. W szczególności artykuł ilustruje prostotę takiej adaptacji, pozwalającą na dostosowanie do innego niż pokazany w punkcie 3 sposobu określania artefaktów analizy i ich raportowania.

Zaproponowana wtyczka EA (moduł utworzony w języku C#) może być łatwo dostosowana do innego niż SWDPS systemu zarządzania procesem wytwórczym. Wystarczy dostarczyć implementację odpowiedniego interfejsu programowego, która będzie wykorzystywała API docelowego systemu zarządzania pracą. Interfejs ten, którym wewnętrznie posługuje się zaproponowana wtyczka, dotyczy synchronizacji cech, wymagań i przypadków użycia. Jego implementację dla systemu SWDPS opisano w punkcie 4.

Oczywiście istnieją nieliczne specjalizowane rozwiązania w zakresie integracji EA z konkretnymi systemami typu *bug and issue tracking*. Przykładem może być EA Connector for Jira® [11]. Rozwiązanie to dotyczy synchronizacji wielu typów artefaktów EA i jest w znacznym stopniu konfigurowalne, ale oczywiście dotyczy jedynie systemu JIRA [12].

Na potrzeby omawianej integracji system SWDPS został rozbudowany w ten sposób, że udostępnia na poziomie serwera bazy danych interfejs do synchronizacji Cech i Przypadków_uzycia/wymagań w postaci pakietu procedur PL/SQL. Dzięki temu może być „sterowany” przez inne narzędzia typu CASE, jeśli tylko istnieje możliwość ich rozszerzania, w podobny sposób jak to zostało pokazane w niniejszym artykule (np. poprzez mechanizm wtyczek).

BIBLIOGRAFIA

1. Booch G., Rumbaugh J., Jacobson I.: UML przewodnik użytkownika. Wydawnictwa Naukowo-Techniczne, Warszawa 2002.
2. Kroll P., Kruchten P.: Rational Unified Process od strony teoretycznej. Wydawnictwa Naukowo-Techniczne, Warszawa 2006.
3. Pilone D., Pitman N.: UML 2.0 in a Nutshell. O'Really Media, 2005.
4. Enterprise Architect, <http://www.sparxsystems.com>, 2010.
5. Sparx Systems – Resources – Developers – UML Profiles, http://www.sparxsystems.com/resources/developers/uml_profiles.html, 2010.
6. Chabik J.: Proces pod informatyczną kontrolą. ComputerWorld Polska. Nr 3/559, 2003.
7. Augustyn D. R., Wyciślik Ł.: Zintegrowany system wspomaganie zarządzania procesami wytwarzania i utrzymania oprogramowania. Bazy danych. Modele, technologie, narzędzia. Analiza danych i wybrane zastosowania. WKŁ 2005, s. 329÷338.
8. Augustyn D. R.: Koncepcja systemu wspomaganie zarządzania procesem wytwórczym oprogramowania. Bazy danych: Nowe technologie. Bezpieczeństwo, wybrane technologie i zastosowania, WKŁ, 2007, s. 351÷361.
9. Augustyn D. R., Warchał Ł.: Moduł wspomagający projektowanie i wytwarzanie złożonych systemów informatycznych opartych na Spring i OSGi. Integracja oprogramowania w procesach integracji systemów informatycznych, Technologie informatyczne – Zarządzanie, oprogramowanie. PWNT, 2010, s. 17÷24.
10. Introduction to Enterprise Architect's Automation Interface, <http://www.sparxsystems.com/resources/developers/autint.html>, 2010.
11. EA Connector for Jira® – Atlassian Plugin Exchange, <https://plugins.atlassian.com/plugins/atlassian.confluence.plugins.eaconnector>, 2012.
12. JIRA – Track bugs, tasks, and projects for software development, <http://www.atlassian.com/software/jira>, 2012.

Wpłynęło do Redakcji 31 stycznia 2012 r.

Abstract

Number of software development methodologies grows. Most of them is mainly set of general guidelines and recommendations. There is a need to adopt them to the specific development process. CASE tools can help in this task, but must be first tuned or even extended to achieve maximum support and automation. This paper describes, taking Enterprise Architect

as an example CASE tool, how to use UML profiles mechanism to modify standard analyze artifacts such as features, use cases or requirements to be consistent with concepts used in adopted and modified UnifiedProcess-based methodology. It is also shown how this solution can help in creating complex reports or documents. This paper shows also how to extend Enterprise Architect with custom plug-in, to integrate it with existing application lifecycle management system (SWPDS).

Adresy

Dariusz R. AUGUSTYN: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, draugustyn@polsl.pl.

Łukasz WARCHAŁ: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, lukasz.warchal@polsl.pl.