

Jacek MAŚLANKOWSKI
Uniwersytet Gdański, Wydział Zarządzania

KONCEPCJA WIELOWARSTWOWEGO MODELOWANIA SYSTEMÓW KLASY BUSINESS INTELLIGENCE

Streszczenie. Niniejszy artykuł prezentuje autorską koncepcję wielowarstwowego podejścia do budowy systemów informatycznych klasy Business Intelligence. Przedmiot badań jest wynikiem analizy stosowanych architektur tej klasy systemów w kontekście występujących niepowodzeń podczas ich wdrożeń. Proponowana architektura wychodzi naprzeciw oczekiwaniom, jakie stawiane są współczesnym systemom informatycznym w zakresie dostosowywania się do zmieniających się warunków otoczenia biznesowego, takich jak zwiększone potrzeby informacyjne użytkowników.

Słowa kluczowe: hurtownie danych, business intelligence, modelowanie, architektura

THE CONCEPT OF MULTILAYER MODELLING APPROACH FOR BUSINESS INTELLIGENCE SYSTEMS

Summary. The article presents original concept of multilayer approach in building Business Intelligence systems. The subject of the study is the result of the analysis of existing architectures of such systems in the context of implementation failures. The architecture proposed in this article towards the expectations of modern information systems, in the scope of changing conditions in business environment, such as changing user requirements.

Keywords: data warehouse, business intelligence, modelling, architecture

1. Wprowadzenie

Architekci współczesnych systemów informatycznych bardzo często borykają się z problemami tworzenia rozwiązań, które w sposób elastyczny mogłyby się dopasowywać do zmieniających się uwarunkowań otoczenia biznesowego. W przypadku systemów klasy Business Intelligence zmiany potrzeb informacyjnych bardzo często skutkują koniecznością ponownego

wdrożenia jego składowych, takich jak hurtownie danych czy bazy operacyjne. Pełna ocena realizacji wymagań użytkowników przez system klasy Business Intelligence jest możliwa poprzez analizę jakości danych zawartych w kluczowym repozytorium systemu, jakim zwykle jest hurtownia danych. Wynika to często z braku możliwości dokonania oceny jakości danych, rozumianej m.in. jako niewystępowanie danych brakujących, niespójnych czy zduplikowanych, zanim dane pochodzące z różnych systemów ulegną pełnemu zintegrowaniu¹. Rezultatem takiego stanu są wysokie koszty ponoszone przez organizacje gospodarcze, związane z utrzymywaniem stosowanych analitycznych systemów informatycznych, polegającym na zastępowaniu obecnych komponentów (hurtownie danych, bazy operacyjne) ich udoskonalonymi implementacjami w przypadku zmieniających się wymagań użytkowników². Według badania nt. systemów klasy Business Intelligence, przeprowadzonego przez grupę Gartnera wśród CFO (ang. *Chief Financial Officer* – dyrektor finansowy), systemy Business Intelligence zajmowały czołową pozycję w zakresie inwestowania we wdrożenia rozwiązań informatycznych w firmach w 2012 roku i zainteresowanie CFO inwestycjami w te rodzaje systemów będzie się prawdopodobnie utrzymywać w 2013 roku³. Stąd bardzo ważne są dalsze prace środowiska naukowego, zmierzające do rozwoju i ułatwienia wdrażania systemów tej klasy.

Rozwiązaniem pozwalającym na obniżanie kosztów związanych z utrzymywaniem tego typu systemów może być zmiana architektury poprzez budowanie systemów informatycznych klasy Business Intelligence na zasadzie niezależnych komponentów, dla których zmiana wymagań biznesowych nie będzie skutkowałą koniecznością wymiany całego systemu, co ma miejsce w proponowanych obecnie rozwiązaniach⁴. Jednym z takich rozwiązań jest propozycja zastosowania architektury SOA (ang. *Service Oriented Architecture*) w hurtowniach danych⁵. Pomimo wdrażania tego typu takich rozwiązań, wciąż zmiany w jednym komponencie, np. hurtowni danych, mogą pociągać za sobą konieczność zmian w innym komponencie, np. narzędziu raportującym. Zatem istotne jest przejście do koncepcji wielowarstwowości, tj. aby zmiana na jednym poziomie nie wymagała ingerencji w pozostałe warstwy systemu.

Teżą postawioną w niniejszym artykule jest założenie, że system klasy Business Intelligence może zostać zaprojektowany w sposób umożliwiający jego dostosowywanie do zmieniających się potrzeb informatycznych użytkowników z niego korzystających. W tym celu wykorzystano koncepcję wielowarstwowości.

¹ Por. Inmon W.H.: *Building the Data Warehouse*. Wiley Publishing Inc., 2005, s. 21.

² Maślankowski J.: *Zastosowania systemów hurtowni danych w organizacjach gospodarczych – Infobazy 2008*. Systemy, aplikacje, usługi. CI TASK, Gdańsk 2008, s. 46.

³ Van Decker J.E.: *What CFOs Want from IT*. *Financial Executive*, Vol. 28, Issue 6, July 2012, s. 66.

⁴ Inmon W.H., Strauss D., Neushloss G.: *DW 2.0. The Architecture for the Next Generation of Data Warehousing*, Morgan Kaufmann, Burlington 2008, s. 45÷47.

⁵ Moseley M.: *Eliminating Data Warehouse Pressures with Master Data Services and SOA*. *Business Intelligence Journal*, Vol. 14, Issue 2, 2009, s. 36.

Celem niniejszego artykułu jest zaprezentowanie koncepcji rozwiązania klasy Business Intelligence, które niewielkim kosztem może być wdrożone w dowolnej organizacji. Istotą przyjętego rozwiązania jest stosowanie koncepcji bazującej na zmiennych oraz plikach XML, co umożliwi jego dostosowywanie się do nowych potrzeb informacyjnych użytkowników.

2. Obecny stan wiedzy na temat systemów klasy Business Intelligence

Pomimo spotykanych w ostatnich latach systemów analitycznych klasy Business Intelligence, korzystających ze zwykłych relacyjnych baz danych, za kluczowy komponent systemów tej klasy wciąż uważana jest hurtownia danych⁶. Wynika to z faktu, że uzyskiwane raporty o bieżącej sytuacji w przedsiębiorstwie zwykle bazują na szeregach czasowych, które są naturalną postacią przechowywania danych w architekturze zgodnej z koncepcją hurtowni danych – każdy schemat hurtowni danych zawiera wymiar czasowy. W takim raporcie kondycja przedsiębiorstwa jest najczęściej rozpatrywana jako poziom efektywności organizacji w zakresie produkcji, sprzedaży czy też w najprostszej postaci – osiąganego w danym okresie przychodu, dochodu lub zysku. Wówczas istnieje możliwość szybkiej reakcji na niekorzystne zmiany pojawiające w organizacji, prowadzące do spadku jej efektywności, na którą składają się m.in. wydajność, sprawność czy produktywność⁷.

Należy również nadmienić, że hurtownie danych, stanowiące rdzeń dla wielu systemów klasy Business Intelligence, zaczęły się upowszechniać w przedsiębiorstwach wraz ze spadkiem ceny wdrożenia takiego systemu, na co pośredni wpływ miało zwiększenie wydajności rozwiązań serwerowych przy jednoczesnym spadku ich ceny. Jednocześnie należy zaznaczyć, że hurtownie danych wciąż bazują na koncepcji zaproponowanej w 1991 roku przez Williama H. Inmona⁸, wraz z regułami dotyczącymi wolno zmieniających się wymiarów (ang. Slowly Changing Dimensions), zaproponowanymi przez Ralphi Kimballa⁹.

Istotne zmiany w podejściu do wdrażania systemów analitycznych miały zajść w pierwszej dekadzie dwudziestego pierwszego wieku, kiedy próbowano stosować zwykłe relacyjne bazy danych jako źródło analiz¹⁰. Było to przedmiotem dyskusji na wielu konferencjach

⁶ Por. Rus V.R., Toader V.: Business Intelligence for Hotels' Management Performance. *International Journal of Business Research*, Vol. 8, Issue 4, s. 151, 2008.; Krishna P.R., Kumar De S.: A Fuzzy Approach to Build an Intelligent Data Warehouse. *Journal of Intelligent & Fuzzy Systems*, Vol. 11, No. 1-2/2001, s. 23; oraz obecne propozycje rozwiązań Business Intelligence, dostępne na stronach dostawców tych usług: www.microsoft.com oraz www.oracle.com z dnia 31 grudnia 2012 r.

⁷ Skurzyńska-Sikora U.: Poprawa efektywności organizacji przy wykorzystaniu modelu PEMM. *Organizacja i Zarządzanie*, nr 3/2008, Wydawnictwo Politechniki Śląskiej, Gliwice 2008, s. 8÷9.

⁸ Inmon W.H.: *Building the Data Warehouse*. Wiley Publishing Inc., 2005, s. 29.

⁹ Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B.: *The Data Warehouse Lifecycle Toolkit*. Wiley Publishing, Inc., 2008, s. 257÷261.

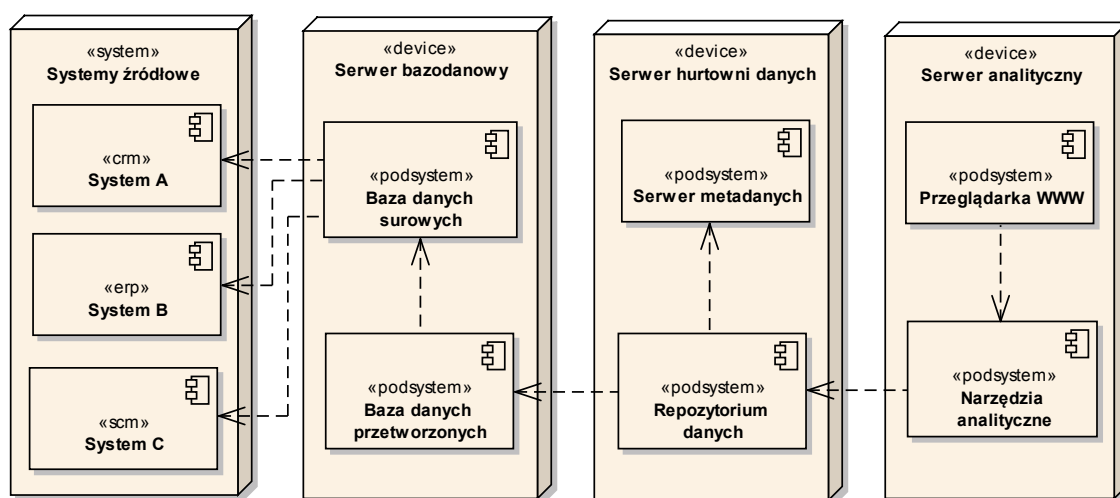
¹⁰ Por. www.sqlmag.com/article/data-management/bi-without-the-data-warehouse z dnia 1 grudnia 2012 r., www-304.ibm.com/businesscenter/cpe/html0/223221.html z dnia 3 grudnia 2012 r.

i sympozjach naukowych. Jednak, jak wspomniano wcześniej, model taki jest stosowany rzadko, gdyż struktura danych w typowych relacyjnych bazach danych zwykle nie jest przystosowana do wykonywania na niej złożonych analiz szeregów czasowych.

3. Przesłanki do stosowania wielowarstwowego modelu

Typowe modułowe podejście do wdrażania systemów klasy Business Intelligence wyróżnia podstawowe komponenty systemu, takie jak: źródła danych, baza operacyjna (obszar przejściowy), hurtownia danych oraz narzędzia analityczno-raportujące. Ralph Kimball w swojej publikacji „The Data Warehouse Lifecycle Toolkit” nazwał te cztery obszary następująco: operacyjne źródła danych, obszar przejściowy danych, obszar prezentacji danych, narzędzia dostępu do danych¹¹. Powiązania pomiędzy tymi obszarami zostały zaprezentowane na diagramie rozlokowania, znajdującym się na rysunku 1, zgodnie z nazewnictwem przyjętym przez autora niniejszego artykułu.

deployment Architektura



Rys. 1. Komponenty typowego systemu klasy Business Intelligence.

Źródło: Opracowanie własne, na podstawie: Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B.: The Data Warehouse Lifecycle Toolkit. Wiley Publishing, Inc., 2008, s. 7.

Fig. 1. Components of the typical Business Intelligence system.

Source: Own elaboration based on: Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B.: The Data Warehouse Lifecycle Toolkit. Wiley Publishing, Inc., 2008, s. 7.

Te komponenty zwykle mają wiele podsystemów, mających wiele interfejsów wejściowych i wyjściowych. W przypadku zmian zachodzących w otoczeniu, interfejsy te umożliwiają zastępowanie istniejących podsystemów nowymi implementacjami, bardziej ukierunkowanymi na bieżące potrzeby informacyjne użytkowników. Takie zmiany wywoływane są

¹¹ Por. Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B.: The Data Warehouse Lifecycle Toolkit. Wiley Publishing, Inc., 2008, s. 7.

najczęściej pozyskaniem nowego źródła danych, którego efektem końcowym jest dostarczenie nowej wiedzy kadrze zarządzającej lub analitykom korzystającym z zasobów informacyjnych pozyskiwanych z tych źródeł. Może się wydawać, że dodanie kolejnego źródła danych przełoży się jedynie na wystąpienie zmiany w komponencie nazwanym powyżej Systemy źródłowe i wyprowadzenie z niego interfejsów wyjściowych, jednak praktyka pokazuje, że zmiany wymagane są nierzadko w pozostałych komponentach – bazie operacyjnej, hurtowni danych oraz narzędziach raportujących. Te zmiany nie ograniczają się jedynie do przeliczenia nowych zestawień i przygotowania raportów. Bardzo często dodanie nowego źródła danych sprawia, że przeliczeń wymagają również wcześniej przygotowane zestawienia danych. Sytuacja taka ma miejsce, gdy jakość istniejących wskaźników, prezentowana wcześniej w raportach, może zostać poprawiona poprzez uwzględnienie tego dodatkowego źródła (np. przez potwierdzenie wystąpienia danego zjawiska). Jakość wskaźników w niniejszym artykule rozumiana jest jako rzetelność i wiarygodność prezentowanych informacji.

Aby uniknąć problemów związanych z dostosowywaniem systemów informatycznych klasy Business Intelligence do zmieniającego się otoczenia, należy zastosować model wielowarstwowo, który umożliwi oddzielenie poszczególnych składowych systemów tej klasy. Należy zauważyć, że zalety wielowarstwowego traktowania technologii informatycznych zostały dostrzeżone w wielu różnych dziedzinach informatyki. Przykładem jest chociażby model referencyjny OSI lub model praktyków TCP/IP dla sieci komputerowych. Modele te, składające się odpowiednio z siedmiu (OSI) lub czterech (TCP/IP) warstw, pozwalają na współpracę wielu protokołów różnych dostawców, które są projektowane niezależnie. Są to powszechnie przyjęte modele, chociaż w literaturze naukowej można spotkać próby ich udoskonalania¹².

Zasadne jest zatem podjęcie próby ustandaryzowania architektury systemów informatycznych, która umożliwi projektowanie poszczególnych ich składowych niezależnie, jednak w sposób spójny i technicznie niezależny. Takie ogólne rozwiązanie może być trudne do zaproponowania dla ogółu systemów informatycznych, ze względu na indywidualny charakter różnych rodzajów oprogramowania. Należy przy tym podkreślić, że duży odsetek wdrażanych systemów informatycznych to rozwiązania dedykowane dla konkretnej firmy, dlatego w niniejszym artykule skoncentrowano się jedynie na propozycji takiego modelu dla systemów klasy Business Intelligence.

¹² Day S.A.: The best of both worlds, *Telephony*; 06/07/99, Vol. 236 Issue 23, 1999, s. 174.

4. Proponowany model wielowarstwowy

Proponowane rozwiązanie zakłada utworzenie wielowarstwowego modelu dla systemów klasy Business Intelligence, pozwalającego na zarządzanie zmianami pojawiającymi się w otoczeniu biznesowym. Zmiany te mogą być wynikiem m.in. dodania do systemu nowych źródeł danych czy też zdefiniowania raportów poprzez zastosowanie innego podejścia do analizowania kondycji przedsiębiorstwa. W rozwiązaniu tym przyjęto, że stosowany będzie model scentralizowany, gdyż jest on jednym z najczęściej występujących architektur hurtowni danych, a jednocześnie wdrożenia takiego modelu relatywnie często kończą się powodzeniem w porównaniu do innych architektur¹³.

Proponuje się wyróżnić sześć zasadniczych warstw systemu klasy Business Intelligence, numerowanych w porządku malejącym:

- 6) warstwa prezentacji odpowiedzialna za aplikację internetową, pozwalającą na przeglądanie danych pobieranych z warstwy analitycznej proponowanego modelu wielowarstwowego;
- 5) warstwa analityczna, stosująca metody statystyki opisowej lub wnioskowania statystycznego w celu przekazywania zestawień danych do warstwy prezentacji;
- 4) warstwa danych przetworzonych, czyli właściwe repozytorium hurtowni danych;
- 3) warstwa integracji danych, odpowiedzialna za stosowanie algorytmów naliczania i przetwarzania danych;
- 2) warstwa danych nieprzetworzonych, znana powszechnie jako baza operacyjna bądź obszar przejściowy;
- 1) warstwa źródeł danych, niezależna od pozostałych warstw, na którą składają się liczne systemy źródłowe.

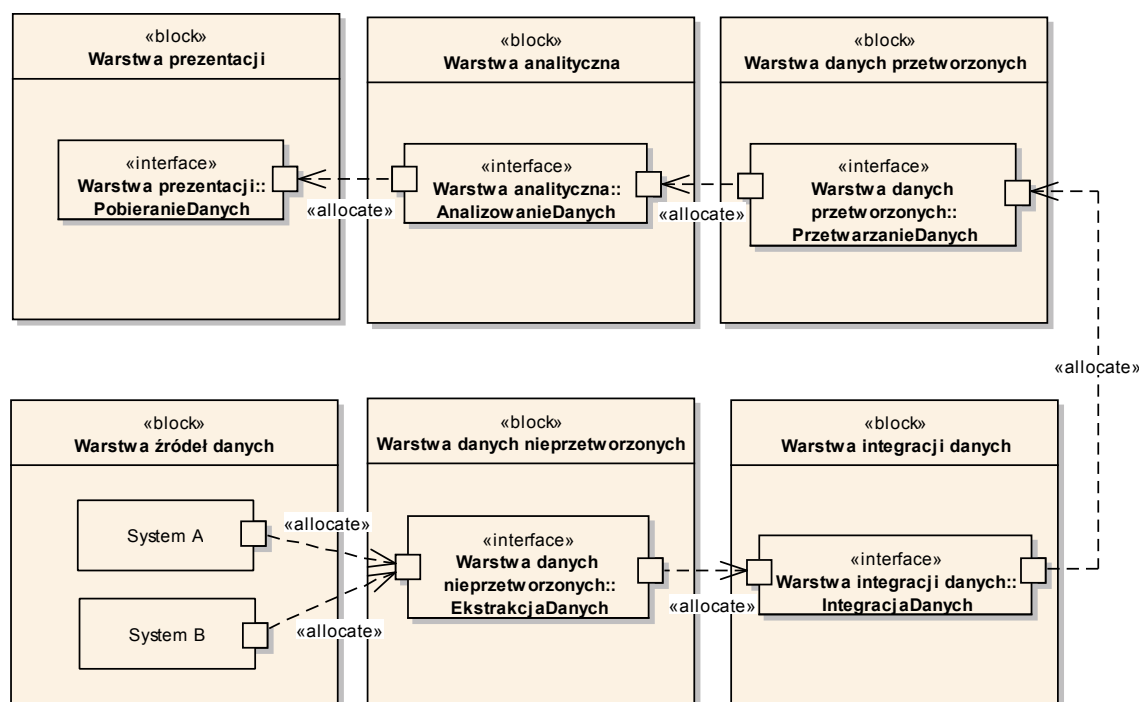
Na styku pomiędzy tymi warstwami istnieją interfejsy wejściowe oraz wyjściowe z zastosowaniem odpowiedniej struktury danych. Nie jest zatem możliwe przejście bezpośrednio z warstwy źródeł danych do warstwy prezentacji, co niekiedy ma miejsce w przypadku systemów analitycznych, kiedy np. dane z systemu ERP (ang. Entity Relationship Management) lub CRM (ang. Customer Relationship Management) są pobierane bezpośrednio przez narzędzia analityczne i podlegają różnym operacjom matematycznym oraz statystycznym, takim jak uśrednianie czy sumowanie. Można domniemywać, że jakość tych analiz może nie być wystarczająca. Wpływ na obniżoną jakość może mieć duże prawdopodobieństwo nieporównywalności poszczególnych zmiennych o tych samych nazwach, ale innym znaczeniu metodologicznym.

¹³ Ariyachandra T., Watson H.: Which Data Warehouse Architecture is Best? Communications of the ACM, Vol. 51, Issue 10, October 2008, s. 147.

Opis przejść pomiędzy poszczególnymi warstwami odbywa się w prezentowany poniżej sposób. Warstwa pierwsza, czyli źródeł, danych jest zbiorem danych nieuporządkowanych, często również nieposiadających identycznej struktury na przestrzeni wielu lat – problemem jest zwłaszcza importowanie danych archiwalnych, gdzie w tabelach może być stosowane inne nazewnictwo zmiennych, mogą występować również złączone kolumny. Zatem wyprowadzenie danych z tych źródeł musi być realizowane poprzez warstwę drugą, czyli danych nieprzetworzonych, w typowych hurtowniach danych nazywana bazą operacyjną lub obszarem przejściowym (ang. staging area). Proponuje się w tej warstwie zdefiniować zarówno protokoły wejściowe, jak również wyjściowe, np. z wykorzystaniem opisu struktury danych w standardzie XML. Warstwa ta powinna wyprowadzać dane w tym samym schemacie, tj. poprzez zmienne zakodowane w szeregu czasowym, w pliku XML. Plik ten zostałby uzupełniony stosownymi metadanymi. Przykładem standardu wyprowadzenia danych w pliku zgodnym z XML, powiązanego z metadanymi i klasyfikacjami, jest SDMX (Statistical Data and Metadata Exchange), którego celem jest dostarczanie danych statystycznych m.in. do europejskiego urzędu statystycznego – Eurostatu¹⁴.

Uproszczony model wielowarstwowy wdrożenia wygląda zatem zgodnie ze schematem przedstawionym na diagramie bloków wewnętrznych na rysunku 2.

ibd [SysML Internal Block] Architektura [Interfejsy]



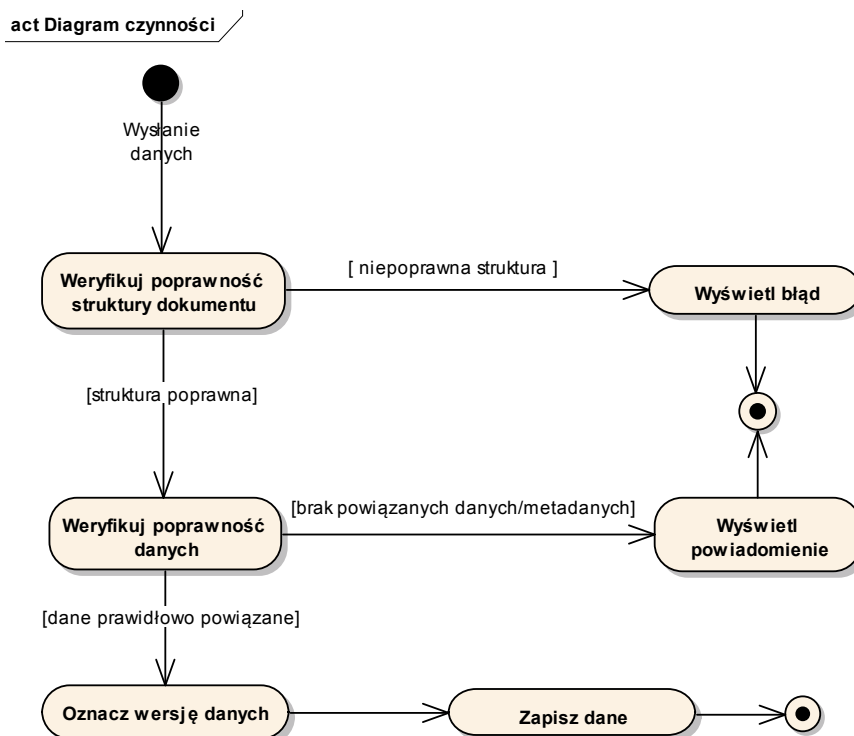
Rys. 2. Proponowane komponenty systemu klasy Business Intelligence
Fig. 2. Suggested components of the Business Intelligence system

¹⁴ sdmx.org z dnia 14 grudnia 2012 r.

Warstwa integracji danych stanie się tym samym tylko odbiorcą danych już wstępnie przetworzonych. Dane te byłyby integrowane w warstwie trzeciej i wyprowadzane również w postaci plików XML do warstwy czwartej, czyli danych przetworzonych. Tam byłyby przechowywane w hurtowni danych. Interfejs wyprowadzający dane do kolejnej warstwy umożliwiłby podanie reguł dotyczących agregacji danych. Reguły te wykorzystane zostałyby poprzez narzędzia analityczne dostępne w kolejnej warstwie. Wyliczone agregaty byłyby pobierane poprzez interfejs występujący pomiędzy warstwami piątą i szóstą, co pozwoliłoby przeglądarce internetowej, działającej na warstwie aplikacji, wykonywanie tego typu obliczeń.

Zgodnie z powyższym rysunkiem, dane z warstwy źródeł danych są przekazywane do warstwy danych nieprzetworzonych za pomocą standardowych interfejsów, takich jak JDBC/ODBC. Odpowiednikiem tej warstwy w typowej architekturze hurtowni danych (rysunek 1) jest baza operacyjna. Istnieje również możliwość przekazywania danych z warstwy źródeł danych do warstwy danych nieprzetworzonych za pomocą plików XML, co zostało krótko opisane w części 5 niniejszego artykułu. Zadaniem warstwy integracji danych jest przetworzenie istniejących danych.

Przykład działania interfejsów pomiędzy warstwami został zademonstrowany na podstawie przekazywania danych pomiędzy warstwą integracji danych a warstwą danych przetworzonych na rysunku 3.



Rys. 3. Diagram czynności przeniesienia danych – warstwa integracji a warstwa danych przetworzonych

Fig. 3. Activity diagram of moving data – integration and processed data layer

Zatem lista interfejsów dostępnych w poszczególnych warstwach jest następująca:

- Warstwa prezentacji – interfejs dostępowy do warstwy analitycznej.
- Warstwa analityczna – interfejs dostępowy, umożliwiający wykonywanie zapytań do warstwy danych przetworzonych.
- Warstwa danych przetworzonych – dostarczanie danych w odpowiedniej strukturze dla warstwy analitycznej.
- Warstwa integracji danych – dostarczanie danych dla warstwy danych przetworzonych.
- Warstwa danych nieprzetworzonych – dostarczanie danych i metadanych dla warstwy integracji danych, pobieranie danych z warstwy źródeł danych.
- Warstwa źródeł danych – niezależna od pozostałych warstw; model nie ma na nią wpływu, chociaż wskazane jest stosowanie tam wyzwalaczy automatycznie aktualizujących zawartość warstwy danych nieprzetworzonych. Alternatywne dla wyzwalaczy jest stosowanie rozwiązań, takich jak Oracle CDC (Change Data Capture), których celem są szybka identyfikacja i przetwarzanie tylko danych zmienionych w celu późniejszego wykorzystania, bez konieczności importowania całych tabel¹⁵.

Interfejs warstwy integracji jest odpowiedzialny za przekazanie zintegrowanych danych do warstwy danych przetworzonych. Na warstwie danych przetworzonych następuje weryfikacja poprawności struktury dokumentu, mająca na celu sprawdzenie, czy przekazany plik XML jest w prawidłowym formacie. Jeżeli jego struktura jest poprawna, wówczas następuje sprawdzenie danych. Weryfikacja ta ma na celu wykrycie braków w repozytorium metadanych lub powiązanych danych, co pozwoli na uniknięcie problemu przechowywania danych zduplikowanych lub niemających odzwierciedlenia w metadanych. Na końcowym etapie działania importowania oznaczana jest wersja danych, co pozwala na aktualizowanie danych znajdujących się już w hurtowni. Finalną czynnością jest umieszczenie wysyłanych danych w repozytorium danych przetworzonych.

5. Zastosowanie proponowanego rozwiązania w praktyce

Proponowany model wielowarstwowy hurtowni danych wychodzi naprzeciw tendencjom, jakie pojawiają się we współczesnych rozwiązaniach technologicznych w tym zakresie. Obecnie oferowane hurtownie danych, np. MS SQL Server 2012 firmy Microsoft¹⁶, DB2 firmy IBM¹⁷ czy baza danych firmy Oracle¹⁸, pozwalają na wsparcie w zakresie natywnej

¹⁵ docs.oracle.com/cd/B10500_01/server.920/a96520/cdc.htm z dnia 1 marca 2013 r.

¹⁶ msdn.microsoft.com/en-us/library/cc645993.aspx z dnia 4 listopada 2012 r.

¹⁷ www-01.ibm.com/software/data/db2/linux-unix-windows/xml/ z dnia 1 marca 2013 r.

¹⁸ www.oracle.com/technetwork/database-features/xmldb/overview/index.html z dnia 1 marca 2013 r.

obsługi plików XML. Zatem dane i ustawienia mogą być importowane bezpośrednio z plików XML, co znacząco ułatwia zastosowanie proponowanego rozwiązania.

Aby zrealizować proponowane rozwiązanie w praktyce, wykorzystano system MS SQL Server 2008 z dodatkiem Analysis Services. Ze względu na ograniczoną objętość niniejszego artykułu, zaprezentowano tutaj jedynie przejście pomiędzy warstwą źródeł danych a warstwą danych nieprzetworzonych. Dostarczany za pomocą wyzwalaczy plik z warstwy źródeł danych do warstwy danych nieprzetworzonych ma w dużym uproszczeniu następującą postać:

```
<DaneZrodlowe symbolZrodla="A1" daneWersja="1.1">
  <Dane>
    <Wiersz id="12112702381288" wartosc="200" />
    <Wiersz id="12112704381288" wartosc="100" />
    <Wiersz id="12112702391288" wartosc="300" />
    ...
  </Dane>
</DaneZrodlowe>
```

W powyższym schemacie id jest kodem oznaczającym kolejno datę (121127), województwo (02, 03), numer zmiennej (381288, 391288).

W przypadku gdy zostanie dodany nowy produkt bądź gdy będzie trzeba rozszerzyć projektowany system o nowe źródło danych, wystarczy w tym źródle zaprojektować wyzwalacz, który będzie wykorzystywał strukturę zgodną z opisywaną wcześniej. Stosowanie wyzwalacza jest najprostszym sposobem na uniknięcie podwójnego importowania danych do obszaru przejściowego. W wielu przypadkach dodanie nowego źródła danych będzie skutkowało koniecznością dodania kolejnych zmiennych w warstwie danych przetworzonych, jednak nie spowoduje to zmiany struktury dokumentu XML. Kluczowe jest jednak traktowanie każdego zjawiska zapisywanego przez hurtownię danych w kategoriach zmiennej (może to być produkt, wskaźnik, indeks itp.), opisanej poprzez metadane biznesowe, pozwalające na zrozumienie istoty analizowanego zjawiska.

Poza standardowymi tabelami wymiarów, jakimi są czas i region, powinna zatem koniecznie zostać wyodrębniona tabela o nazwie zmienna. Tabela ta mogłaby zawierać standardowe atrybuty, takie jak: id, nazwa, opis, kategoria, podkategoria itd. Dodatkowo, jeżeli zachodzi taka konieczność, schemat gwiazdy powinien zostać uzupełniony o tabelę o nazwie jednostka miary, gdy zmienne reprezentowane są w różnych jednostkach – zwłaszcza dotyczy to wskaźników mikro- i makroekonomicznych.

Zaletą takiego rozwiązania jest możliwość rozszerzenia zakresów przedmiotowego i podmiotowego systemu bez konieczności dostosowywania schematu systemu do nowych struktur biznesowych, poprzez dodawanie nowych tabel czy interfejsów. Należy jednak podkreślić, że rozwiązanie takie może być stosowane raczej w systemach przechowujących niezbyt duże ilości danych, ze względu na duży narzut wynikający ze stosowania plików XML oraz przetwarzania dużej liczby zmiennych zawartych w systemie bazodanowym. Wydajność może zostać poprawiona poprzez partycjonowanie bazy danych oraz stosowanie bardziej

wydajnych mechanizmów przekazywania danych z plików XML, jednak ustępuje to wydajności oferowanych obecnie przez hurtownie danych mechanizmów przetwarzania danych.

6. Wnioski

Zalety płynące z niniejszej propozycji wielowarstwowego traktowania wdrożeń mogą mieć istotny wpływ na rozwój i utrzymywanie systemów klasy Business Intelligence. Należy zwrócić uwagę, że najczęściej wdrażane rozwiązania tej klasy, zwłaszcza dedykowane, nie pozwalają w wystarczającym stopniu na ich łatwą rozbudowę o kolejne moduły, co oznacza, że próba rozszerzenia jego zakresu przedmiotowego lub podmiotowego kończy się decyzją o wymianie tego systemu na nowszy. Przyczyn można się dopatrywać również w konieczności wymiany infrastruktury informatycznej, gdyż, jak powszechnie wiadomo, sprzęt informatyczny starzeje się znacznie szybciej niż stosowane oprogramowanie.

Rozwiązanie zaproponowane w niniejszym artykule może być stosowane w niewielkich systemach typu Business Intelligence, działających w firmach, gdzie potrzeby informacyjne użytkowników tego systemu zmieniają się w sposób znaczący. Problemy mogą pojawić się w większych systemach tego typu, gdy wymagane jest przetwarzanie dużej liczby danych – wówczas proponowany sposób przekazywania danych w postaci plików XML może okazać się mało wydajny. Pomimo że opisywane rozwiązanie funkcjonuje jedynie w środowisku testowym, wykorzystując przy tym standardowe aplikacje, to jednak pozwoliło na pokazanie, że system Business Intelligence może być projektowany w sposób alternatywny do typowych rozwiązań, bardziej elastyczny w zakresie dostosowania się do zmian potrzeb informacyjnych użytkowników. Przede wszystkim nadaje się do systemów, gdzie istnieje duża różnorodność przechowywanych informacji (wskaźników makroekonomicznych, indeksów itp.), połączonych np. z wynikami sprzedaży poszczególnych produktów, gdy również lista tych wskaźników stale się powiększa.

Stosując proponowaną koncepcję wielowarstwowości systemu z wykorzystaniem plików XML, można potwierdzić tezę postawioną we wprowadzeniu do niniejszego artykułu, głoszącą, że system klasy Business Intelligence może zostać zaprojektowany w sposób umożliwiający jego dostosowywanie do zmieniających się potrzeb informatycznych użytkowników z niego korzystających. Jak stwierdzono w środowisku testowym, w przeciwieństwie do wyników badań prowadzonych przez autora artykułu, zmiany potrzeb informacyjnych w danych zawartych w repozytorium nie muszą skutkować koniecznością projektowania poszczególnych komponentów systemu od podstaw.

BIBLIOGRAFIA

1. Ariyachandra T., Watson H.: Which Data Warehouse Architecture is Best? *Communications of the ACM*, Vol. 51, Issue 10, 2008, s. 147.
2. Day S. A.: The best of both worlds. *Telephony*, Vol. 236, Issue 23, 1999, s. 174.
3. Inmon W. H., Strauss D., Neushloss G.: DW 2.0. The Architecture for the Next Generation of Data Warehousing. Morgan Kaufmann, Burlington 2008, s. 45÷47.
4. Inmon W. H.: Building the Data Warehouse, Fourth Edition. Wiley Publishing Inc., 2005, s. 29.
5. Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B.: The Data Warehouse Lifecycle Toolkit. Wiley Publishing, Inc., 2008, s. 257÷261.
6. Krishna P. R., Kumar De S.: A Fuzzy Approach to Build an Intelligent Data Warehouse. *Journal of Intelligent & Fuzzy Systems*, Vol. 11, No. 1-2, 2001, s. 23.
7. Maślankowski J.: Zastosowania systemów hurtowni danych w organizacjach gospodarczych. *Infobazy 2008. Systemy, aplikacje, usługi*. CI TASK, Gdańsk 2008, s. 46.
8. Moseley M.: Eliminating Data Warehouse Pressures with Master Data Services and SOA. *Business Intelligence Journal*, Vol. 14, Issue 2, 2009, s. 36.
9. Rus V. R., Toader V.: Business Intelligence for Hotels' Management Performance. *International Journal of Business Research*, Vol. 8, Issue 4, 2008, s. 151.
10. Skurzyńska-Sikora U.: Poprawa efektywności organizacji przy wykorzystaniu modelu PEMM. *Organizacja i Zarządzanie*, nr 3/2008, Wydawnictwo Politechniki Śląskiej, Gliwice 2008.
11. Van Decker J. E.: What CFOs Want from IT. *Financial Executive*, Vol. 28, Issue 6, 2012, s. 66.
12. docs.oracle.com/cd/B10500_01/server.920/a96520/cdc.htm, z dnia 1 marca 2013 r.
13. msdn.microsoft.com/en-us/library/cc645993.aspx, z dnia 4 listopada 2012 r.
14. sdmx.org, z dnia 14 grudnia 2012 r.
15. www.sqlmag.com/article/data-management/bi-without-the-data-warehouse, z dnia 1 grudnia 2012 r.
16. www-304.ibm.com/businesscenter/cpe/html0/223221.html, z dnia 3 grudnia 2012 r.
17. www.microsoft.com, z dnia 31 grudnia 2012 r.
18. www.oracle.com, z dnia 31 grudnia 2012 r.
19. www.oracle.com/technetwork/database-features/xmldb/overview/index.html z dnia 1 marca 2013 r.
20. www-01.ibm.com/software/data/db2/linux-unix-windows/xml/, z dnia 1 marca 2013 r.

Wpłynęło do Redakcji 16 stycznia 2013 r.

Abstract

The article presents original concept of multilayer approach in building Business Intelligence systems. The subject of the study is the result of the analysis of existing architectures of such systems in the context of implementation failures. The architecture proposed in this article towards the expectations of modern information systems, in the scope of changing conditions in business environment, such as changing user requirements. The thesis of the article was to show that Business Intelligence can adapt itself to changes in user requirements after its implementation. To confirm that thesis the multilayer approach in building Business Intelligence systems was developed and implemented in the tested environment. The model of Business Intelligence system was divided into 6 layers which are: presentation, analytical, processed data, data integration, unprocessed data and data sources. The key of the approach is to use XML files to transfer the data between different layers. It allows adding new data sources as well as new variables without changing the structure of the system.

Adres

Jacek MAŚLANKOWSKI: Uniwersytet Gdański, Wydział Zarządzania,
ul. Armii Krajowej 101, 81-824 Sopot, Polska, jacek@ug.edu.pl.