

Paweł BUCHWALD  
Wyższa Szkoła Biznesu w Dąbrowie Górniczej

## ZASTOSOWANIE WYBRANYCH MECHANIZMÓW TRANSFORMACJI DANYCH NA POTRZEBY SYMULACYJNE NA PRZYKŁADZIE SQL SERVER

**Streszczenie.** Artykuł przedstawia możliwości w zakresie zastosowania mechanizmów transformacji, ładowania i pozyskiwania danych w celu zapisania ich w jednolitym modelu danych. Tak stworzony jednolity model zapisany w relacyjnej bazie danych podlega łatwiejszej integracji z rzeczywistymi danymi przedsiębiorstwa i pozwala na dokonanie na nim badań symulacyjnych.

**Słowa kluczowe:** mechanizmy transformacji, pozyskiwania i ładowania danych, model symulacyjny systemów zarządzania, SQL Server,

## USE OF SELECTED EXTRACTION, TRANSFORMATION AND LOADING DATA MECHANISM FOR SIMULATION OF MANAGEMENT SYSTEM IN SQL SERVER

**Summary.** This study represents some chosen ETL mechanisms which are possible to be applied in the SQL database server. These mechanisms may be used for an initial preparation of data saved in a relational database. By means of them one may obtain a description of an Integrated Database Management System with a view to its integration with a company's actual systems or its use for simulative applications.

**Keywords:** ETL mechanism, SQL Server, Simulation of management system

### 1. Wstęp

Tworzone obecnie systemy zarządzania są projektowane i implementowane przez szerokie grono osób, które posiadają wiedzę branżową z rozległych dziedzin. Znacznym problemem w osiągnięciu dobrego modelu systemu zarządzania jest stworzenie rozwiązania odpo-

wiadającego idealnie potrzebom jego użytkowników w różnych warstwach funkcjonowania przedsiębiorstwa. Kluczowym warunkiem utworzenia modelu symulacyjnego odpowiadającego potrzebom jego użytkowników jest wykorzystanie dotychczasowej wiedzy o procesach i danych niezbędnych do funkcjonowania przedsiębiorstwa, dla którego jest projektowany system informatyczny. Nie bez znaczenia jest również uzyskanie spójności pomiędzy utworzonym modelem, specyfikacją użytkownika oraz implementacją finalną. Uzyskanie takiej spójności utrudniają rezimy czasowe i finansowe stawiane projektantom oraz brak „wspólnego języka” pomiędzy poszczególnymi specjalistami odpowiadającymi za stworzenie założeń i implementację systemu. W wielu przypadkach rozwiązaniem tego problemu może być wprowadzenie modeli referencyjnych, które mogą w dużym stopniu odpowiadać potrzebom przedsiębiorstw o podobnym charakterze. Bazując na tychże modelach posiadających sformalizowaną postać zapisu danych można stworzyć za pomocą odpowiednich mechanizmów translacji TLP swoisty system przechowywania danych służących do ich analizy przy użyciu mechanizmów decyzyjnych. Takie rozwiązanie często stosuje się w systemach podejmowania decyzji, które jako źródło danych przyjmują kostki OLAP zasilane danymi z tradycyjnych systemów OLTP. Użycie wspólnych modeli referencyjnych bez uwzględnienia dotychczasowej wiedzy o unikalnych aspektach specyfiki danego przedsiębiorstwa może jednak spowodować zbytne upodobnienie go do konkurencji i zniwelować jego przewagę biznesową na rynku. W celu zapobieżenia temu negatywnemu zjawisku istotne jest wykorzystanie wiedzy o dotychczasowym działaniu przedsiębiorstwa. Modelowanie od podstaw procesów biznesowych i danych niezbędnych do jego funkcjonowania spowodowałoby znaczne wydłużenie kosztów czasowych powstania finalnego rozwiązania oraz kosztów związanych z utrzymaniem spójności wielu unikalnych systemów i ich modeli. Ułatwieniem w realizacji tego nietrywialnego zadania może być zastosowanie pewnych metamodeli oraz mechanizmów translacji danych z jednego modelu na inny, które są oferowane przez dostępne obecnie na rynku systemy bazodanowe, a także narzędzia szybkiego projektowania i implementacji aplikacji. W niniejszym opracowaniu starano się przedstawić mechanizmy uznane przez inżynierię oprogramowania i powszechnie używane w stosowanych obecnie narzędziach bazodanowych do budowy spersonalizowanych rozwiązań informatycznych. Zwrócono szczególną uwagę na łatwość w tworzeniu typowych mechanizmów translacji danych oraz zestandaryzowanie postaci danych. Wprowadzenie zunifikowanych standardów zapisu danych na potrzeby projektowania i implementacji systemów informatycznych, zwłaszcza podejmowania decyzji oraz symulacji rozwiązań decyzyjnych, jest ważnym aspektem ze względu na heterogeniczność i różnorodność systemów, które mają być źródłami danych. Z tego względu na popularności zyskują te standardy i modele, które są niezależne od grupy producentów i wspierane przez ogólnosiękatowe otwarte standardy. Jednym z takich standardów jest język XML dający duże możliwości przez swoją uniwersalność. Nie wolno jednak zapominać o innych modelach danych, które mogą pozwolić na wydobycie cennych informacji o organizacji już

delach danych, które mogą pozwolić na wydobycie cennych informacji o organizacji już istniejących systemów wspomagających działanie przedsiębiorstwa. Z tego względu nowoczesne narzędzia translacji danych powinny mieć również możliwość wykorzystania innych modeli oraz ich konwersji na używane powszechnie postacie.

## **2. Usługi transformacji oraz integracji danych**

Działanie mechanizmu SSIS polega na wykorzystaniu specjalnych pakietów służących do selekcjonowania danych, ich translacji oraz zapisu do innych miejsc przechowywania danych w rozproszonym systemie informatycznym. Niezwykle istotna jest rola transformacji samej postaci danych wykorzystywana przez pakiety SSIS. Implementacja pakietów SSIS jest możliwa za pomocą bloków logiki biznesowej udostępnianych w ramach jednego z modułów narzędzia Business Intelligence dostarczanego razem z systemem SQL Server 2008. Wraz z rozwojem zintegrowanych systemów zarządzania wzrosło znaczenie pozyskiwania danych mogących posłużyć do przygotowania informacji niezbędnych dla wspomagania decyzji podejmowanych w przedsiębiorstwie. Dane te są zapisane w rozmaitych źródłach danych często nierelacyjnych lub pochodzą z heterogenicznych systemów takich, jak systemy nadrzędnej kontroli i dystrybucji danych SCADA oraz systemy zarządzania produkcją. Przed możliwością skorzystania z tych danych należy dokonać ich transformacji do pożądanej postaci. Powszechnym rozwiązaniem pozwalającym na dokonanie takiej transformacji są mechanizmy ETL (extraction – transformation – loading) używane do wstępnego przygotowania oraz akwizycji danych dla systemów hurtowni danych. Śledzenie mechanizmów transformacji danych jest równie ważne jak pozyskiwanie informacji na temat procesów biznesowych wykonywanych w informatycznych systemach zarządzania. Decyzje podjęte na podstawie danych nieaktualnych bądź błędnie pozyskanych nie są wartościowe dla użytkowników tego typu systemów informatycznych. Przygotowanie adekwatnego do potrzeb mechanizmu transformacji danych ma często charakter wielorazowy. Ze względu na możliwość modyfikacji struktur zapisu danych w modułach rozproszonego systemu informatycznego bądź podmianę modułu na nowszą wersję mechanizmy wstępnej obróbki i przygotowania danych muszą być modyfikowalne w stosunkowo łatwy sposób. Podczas implementacji logiki biznesowej procedur ETL nie bez znaczenia pozostaje czytelność zaimplementowanego rozwiązania. Czytelność rozwiązania jest ściśle powiązana z możliwością jego przyszłej modyfikacji oraz dostosowania do zmieniającej się struktury systemu informatycznego. Wymienione cechy są spełnione przez usługi SSIS będące częścią środowiska serwera baz danych SQL Server 2008.

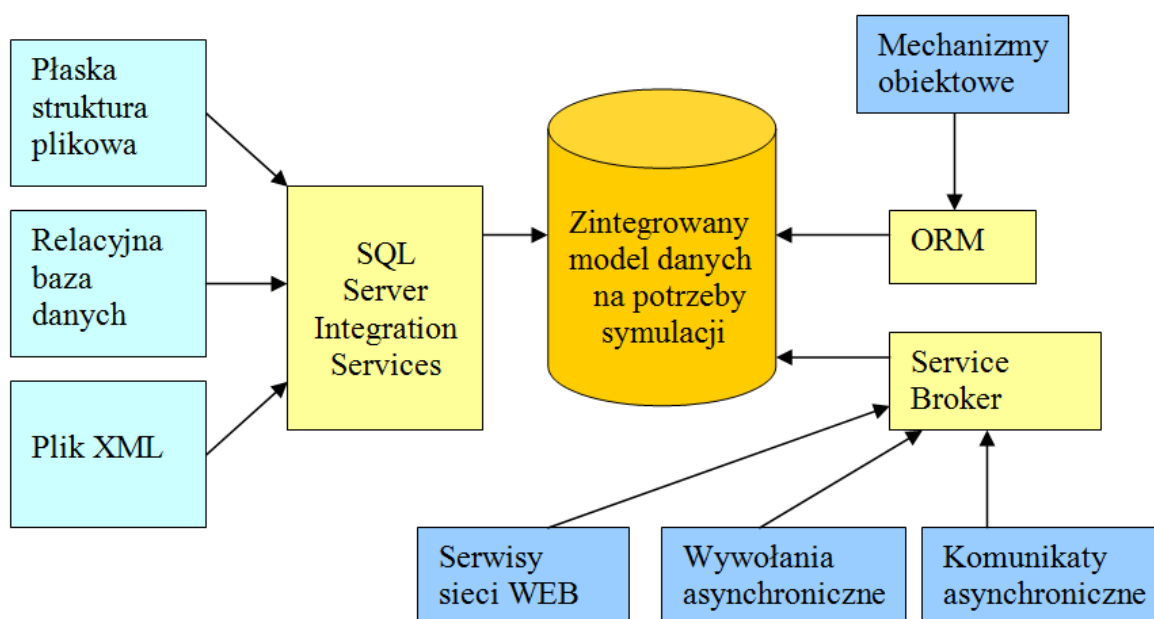
SQL Server Integration Services są następcą popularnych usług Data Transformation Services udostępnianych przez wcześniejsze wersje serwera bazy danych firmy Microsoft. Interfejs graficzny usług transformacji danych zaproponowano w taki sposób, aby reprezentacja działania utworzonego rozwiązania przypominała diagramy workflow używane przy projektowaniu logiki biznesowej oprogramowania. Usługi transformacji danych pozwalają zapisać diagram zaprojektowanej logiki biznesowej wprost jako pakiet w bazie danych lub przestrzeni plikowej serwera. Pakiet ten może być w każdej chwili uruchomiony i wykonany, natomiast stan wykonania pakietu może być monitorowany i zapisywany w tabelach relacyjnej bazy danych. Ze względu na bogaty zbiór dostępnych komponentów, z jakich można projektować logikę biznesową transformacji danych, możliwe jest dostosowanie działania pakietu w sytuacjach wyjątkowych oraz realizacja powiadomień użytkownika o stanie wykonania lub nieprawidłowym przebiegu transformacji danych. We wcześniejszych wersjach często wykorzystywanym sposobem do realizacji powiadomień o sytuacjach wyjątkowych lub uruchamiania logiki biznesowej w odpowiedzi na zaistnienie pewnych zdarzeń w systemie był mechanizm zadań administracyjnych (Job). Jest on jednak przeznaczony dla czynności administracyjnych i implementacja bardziej zaawansowanej logiki biznesowej w oparciu o tę technologię jest niezwykle żmudnym i pracochłonnym zajęciem. Ponadto znikome wsparcie dla wizualizacji logiki biznesowej i mała ilość komponentów pozwalających na realizację zadań czyni zaimplementowane w taki sposób rozwiązania mało czytelnymi, a tym samym trudne do późniejszej modyfikacji. Zarówno komponenty SSIS, jak i elementy dostępne w ramach mechanizmu zadań administracyjnych SQL Servera pozwalają na uruchamianie procesów zewnętrznych. Diagram SSIS jest jednak rozwiązaniem pozwalającym na budowanie logiki biznesowej z graficznych komponentów, przez co jego śledzenie i utrzymywanie spójności z potrzebami użytkowników jest o wiele prostsze. Podstawowe elementy składniowe dostępne w ramach pakietów SSIS, z których można budować funkcjonalność rozwiązania, można podzielić na następujące grupy:

- Adaptery danych źródłowe i docelowe – są to komponenty pozwalające na połączenie z zewnętrznym źródłem danych w dowolnej postaci i pozyskanie z niego danych bądź wprowadzenie danych do innego systemu. Adaptery danych źródłowych i docelowych pełnią w komunikacji taką rolę, jak obiekty klasy DataAdapter przy mechanizmach komunikacji z bazami danych za pomocą bibliotek ADO.NET. Najważniejsze z adapterów danych używanych w SSIS to:
  - Adaptery OLEDB – pozwalające na korzystanie ze źródeł danych kompatybilnych ze standardem Object Linking and Embedding Databases. Dla większości baz danych używanych dzisiaj w nowoczesnych rozproszonych systemach informatycznych można z łatwością pozyskać dedykowane dla nich komponenty OLEDB.

- Adapter XML – pozwala na pracę ze źródłami danych będącymi plikami XML. Należy podkreślić, iż pojęcie pliku jest tu użyte w sensie logicznym, co pozwala na połączenie się ze źródłem danych w postaci XML, które może być obsługiwane za pomocą komponentów charakterystycznych dla pracy z plikami.
- Komponenty zadań elementarnych pozwalające na wykonywanie podstawowych zadań komunikacji z innymi procesami oraz elementarnych zadań przetwarzania danych. Najważniejsze z nich to:
  - Execute SQL – komponent pozwalający na wykonanie polecenia lub procedury składowanej po stronie bazy danych. Pozwala na przekazanie parametrów wejściowych oraz pobranie parametrów wyjściowych z procedur w języku TSQL, a także uruchomienie sparametryzowanych zapytań.
  - Execute Process – komponent stosowany do uruchomienia procesu zewnętrznego podobnie jak w wierszu poleceń. Pozwala na przekazywanie parametrów do procesów oraz potoków.
  - Execute Package – komponent użyty do ukrycia całości logiki biznesowej części pakietu. Pozwala na agregowanie pakietów w podpakiety i ich późniejsze używanie. Za pomocą tego komponentu można podzielić logikę realizowanego rozwiązania na mniejsze części, którymi łatwiej jest zarządzać. Zwiększa to znacznie czytelność budowanego rozwiązania.
  - File System Operations – komponenty pozwalające na wykonywanie operacji w systemie plików takich, jak tworzenie katalogów, kasowanie plików, kopiowanie czy ich przenoszenie.
  - Script – komponent pozwalający na uruchomienie skryptu zaimplementowanego w Visual Basic .NET. Dzięki temu komponentowi możliwe jest utworzenie logiki biznesowej rozwiązań zgodnie ze wszystkimi udogodnieniami platformy .NET Framework, włączając w to obiektowy paradygmat projektowania aplikacji.
  - Data Flow – umożliwiający kontrolę przepływu pomiędzy poszczególnymi komponentami realizowanego rozwiązania. Pozwala na uzależnienie przepływu od skutków wykonania bieżącej operacji lub funkcji warunkowej.
- Komponenty agregujące zwane kontenerami pozwalające na zaimplementowanie logiki sterowania oraz podział skomplikowanych zadań na mniejsze składowe:
  - Sequence – komponent umożliwiający agregowanie podpakietów oraz ich sekwencyjne wykonanie w sposób synchroniczny.
  - For Loop – komponent pozwalający na agregowanie podpakietów oraz ich warunkowe wykonanie w pętli. Komponent pozwala na określenie warunku zakończenia pętli wykonania wszystkich pakietów.

- For Each – komponent pozwalający na realizację zadań w pętli dla wszystkich agregatów.
- Komponenty synchronicznej i asynchronicznej transformacji danych pozwalające na manipulowanie danymi pobranymi z zewnętrznych źródeł oraz ich warunkowe przekazanie do elementów docelowych systemu. W obrębie tej grupy stosowane są takie komponenty, jak:
  - Conditional Split – komponent pozwalający na podstawie wcześniej przygotowanego warunku dla przetwarzanych danych umieścić bieżący rekord w docelowym źródle danych. Pozwala on na realizację warunkowego przesyłu danych.
  - Union All – komponent pozwalający na połączenie danych z wielu źródeł w sposób podobny do operacji unii wykonywanej na danych zwróconych przez kilka zapytań w języku SQL.

Logikę sterowania zaimplementowaną w oparciu o komponenty diagramu SSIS zrealizować można za pomocą kreatora pakietów SSIS udostępnianego w ramach komponentów narzędzia Visual Studio Business Intelligence. Pakiety SSIS są jednym z ważniejszych mechanizmów służących do budowy mechanizmów ekstrahowania transformacji i zapisywania danych.



Rys. 1. Schemat translacji danych przedsiębiorstwa

Fig. 1. Scheme of translation company database

### **3. Realizacja wymiany danych z usługami zorientowanymi serwisowo z zastosowaniem SQL Server Service broker**

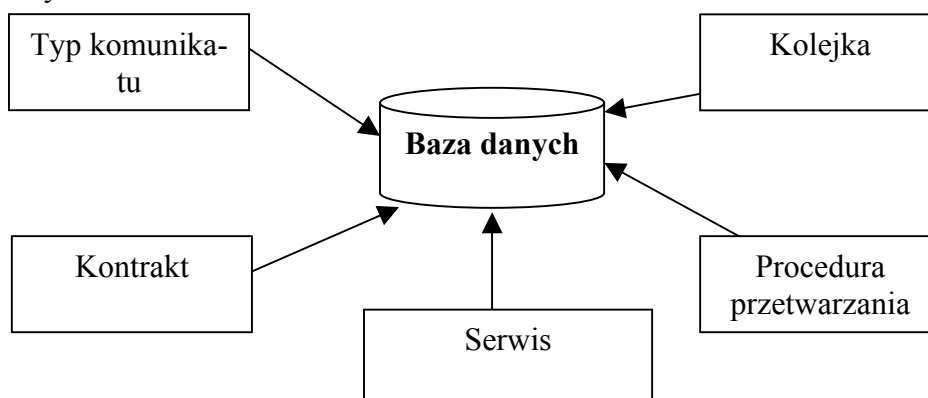
W wielu rozproszonych systemach wymiany informacji stosuje się asynchroniczną obsługę przesyłania danych. Taki sposób komunikacji jest charakterystyczny zwłaszcza dla usług o charakterze serwisowym. Wiele rozproszonych systemów wielowarstwowych wykorzystuje architekturę opartą na serwisach sieci Web oraz protokół SOAP jako warstwę komunikacyjną. Ten sposób budowy rozwiązań logiki biznesowej zdobył popularność między innymi z tego względu, iż mechanizm Web Services nie jest zależny od jednej platformy sprzętowej bądź programowej, ale jest wspierany jako niezależny standard popierany przez międzynarodową organizację World Wide Web Consortium. Asynchroniczna komunikacja bazy danych z usługami sieci Web również może odbywać się przy zastosowaniu protokołu SOAP. Realizację takiego sposobu komunikacji ułatwia znacznie charakter samego protokołu komunikacyjnego, który oparty jest na znacznikach XML. Mechanizmy dostępne w narzędziu SQL Server w ramach usług Service broker udostępniają nie tylko łatwy sposób przetwarzania i wysyłania komunikatów do zdalnych usług serwisowych, ale pozwalają na transformację danych wprost do tabel relacyjnej bazy danych oraz obsługę kolejkowania komunikatów za pomocą dostępnych na serwerze baz danych rozwiązań.

Przy projektowaniu mechanizmów wymiany informacji ze zdalnym serwisem należy zdefiniować postać komunikatów używanych podczas przesyłu danych. Mechanizmy Service broker mają możliwość automatycznego sprawdzania poprawności przesyłanych komunikatów. Każdy z przesyłanych komunikatów jest pewną porcją danych w postaci XML. Do sprawdzenia poprawności składniowej komunikatu stosuje się więc plik schematu XML. Plik ten jest wskazywany podczas tworzenia pojedynczego komunikatu.

Kolejność poszczególnych komunikatów wymienianych pomiędzy bazą danych a serwisami musi być również zdeterminowana w celu osiągnięcia poprawnej wymiany danych. Kolejność wysyłanych i odbieranych komunikatów oraz stronę mogącą zainicjować transmisję opisuje tzw. kontrakt. Kontrakt jest swoistym opisem sposobu posługiwania się wcześniej zdefiniowanymi typami komunikatów.

Treść przesyłanych komunikatów jest przechowywana i buforowana w tabelach relacyjnej bazy danych. Użycie dostępnych tabel zwalnia projektanta od samodzielnej implementacji mechanizmów kolejkowania i przetwarzania danych. Za pomocą usług service broker należy określić miejsce składowania treści komunikatów, możliwość jednoczesnego użycia kolejki przez kilka serwisów oraz procedurę wbudowaną, która ma być uruchomiona w momencie wstawienia nowego komunikatu do kolejki. Zaprojektowanie poprawnej platformy komunikacyjnej pomiędzy bazą danych a usługami zorientowanymi serwisowo w oparciu o mechanizm Service broker wymaga określenia wszystkich przedstawionych wyżej elemen-

tów. Składowe elementy mechanizmów komunikacyjnych używanych przez service broker przedstawia rysunek 2.



Rys. 2. Elementy komunikacyjne mechanizmu service broker  
 Fig. 2. Elements of service broker communication's mechanisms

Ze względu na swe szerokie możliwości dostosowania do różnorodnych zastosowań i łatwość implementacji rozwiązań w oparciu o usługi service brokera jest on jednym z ważnych komponentów pozwalających na realizację mechanizmów ETL zwłaszcza w systemach, dla których wymagana jest komunikacja ze zorientowanymi serwisowo komponentami wykorzystującymi protokoły komunikacyjne oparte na powszechnym standardzie XML.

#### 4. Użycie mechanizmów ORM w przygotowaniu obsługi procesów biznesowych przedsiębiorstwa

Mechanizmy ORM (Object Relation Mapper) pozwalają na wykorzystanie danych przechowywanych w postaci relacyjnej poprzez mechanizmy zgodne z paradygmatem projektowania obiektowego. Szerokie rozpowszechnianie się obiektowo zorientowanych metod projektowania aplikacji, jak również korzystanie z języków graficznych zorientowanych obiektowo, takich jak UML, do modelowania rozwiązań informatycznych w wielu przedsiębiorstwach wymusza integrację danych przechowywanych często w tabelach relacyjnego systemu serwera baz danych z obiektową architekturą implementowanego rozwiązania. Integracja danych zapisanych zgodnie z modelem relacyjnym wymaga mapowania pól implementowanych klas na konkretne kolumny tabel przechowywanych w relacyjnych bazach danych. W większości systemów ORM prawidłowe mapowanie umożliwiają dodatkowe metadane, które muszą być utrzymywane podczas implementacji systemu. Utrzymanie tych metadanych daje możliwość automatycznego mapowania odwołań do pól poszczególnych obiektów, do konkretnych rekordów bazy danych. Dzięki temu projektant może skupić się na realizacji wymaganych procesów biznesowych bez konieczności uciążliwej implementacji mechanizmów selekcjonowania danych z bazy relacyjnej. Wadą automatycznego mapowania obiektów



tów na relacyjną bazę danych jest mniejsza wydajność instrukcji selekcji danych wygenerowanych w sposób mechaniczny przy użyciu komponentów ORM. Wada ta jest widoczna szczególnie przy bardziej skomplikowanych zapytaniach złożonych z agregacji, które wymagają profilowania i modyfikacji w celu uzyskania lepszej wydajności. W przypadku krótkich i mniej skomplikowanych zapytań mechanizmy ORM mogą być z powodzeniem wykorzystywane jako warstwa pośrednicząca w dostępie do danych. Warunkiem użycia automatycznych komponentów mapowania danych relacyjnych na postać obiektową przy budowie rozwiązań symulacyjnych systemów zarządzania jest posłużenie się takim modelem, który w stosunkowo łatwy sposób za pomocą prostych konstrukcji składniowych języka SQL pozwala pozyskać dane o symulowanym przedsiębiorstwie.

Warunek ten jest spełniony przez model ERC. Zapewnia on uniwersalną strukturę organizacyjną zbudowaną w oparciu o relacyjny model bazy danych. Model ten dla każdej z tabel używa stosunkowo niewielkiej liczby pól kluczowych, przez co wyodrębnienie każdego elementu modelowanego przedsiębiorstwa wymaga wykonania prostych instrukcji select języka SQL. Zastosowanie zestandaryzowanego zapisu organizacyjnej struktury przedsiębiorstwa jest niezwykle pomocne w realizacji procedur ETL. Reprezentacja struktury przedsiębiorstwa w oparciu o taki model daje więc możliwości w przygotowaniu mechanizmów migracji do hurtowni danych lub systemów decyzyjnych. Translacja danych przez mechanizmy ORM działająca w oparciu o automatyczne generowanie zapytań w przypadku zastosowania modelu ERC daje stosunkowo mały narzut czasowy. Proste zapytania nie wymagają zastosowania skomplikowanych mechanizmów podwyższania wydajności czasowej generowania odpowiedzi po stronie bazy danych.

Dla większości środowisk do projektowania i uruchamiania oprogramowania w tym również dla platformy .NET powstało wiele rozwiązań typu ORM. Jednym z nich jest zaproponowany przez Microsoft mechanizm LINQ czy wywodzący się z języka Java popularny ORM o nazwie hibernate. Zastosowanie mechanizmów ORM łącznie z serializacją daje możliwość instancjonowania pól obiektów po stronie bazy danych. Jest to istotne w modelowaniu przedsiębiorstwa, a zwłaszcza podczas śledzenia realizacji procesów biznesowych. Dane opisujące proces mogą być w każdej chwili zapisane w bazie danych, przez co stan procesu zostaje „zahibernowany” i może zostać odtworzony w dowolnym momencie. Kopie poszczególnych stanów procesu z określonych przedziałów czasowych stanowią obraz do analizy i symulacji decyzji biznesowych.

## 5. Wnioski

Dane opisujące informacje o procesach w przedsiębiorstwach mogą pochodzić z wielu heterogenicznych źródeł. Pozyskiwanie danych obrazujących stan przedsiębiorstwa odbywa się przez integrację z przemysłowymi systemami czasu rzeczywistego nadzorowanymi przez systemy nadrzędnej kontroli i dystrybucji danych (SCADA – Supervisory Control And Data Acquisition) systemy finansowo-księgowe, systemy wspomaganie decyzji oraz dedykowane systemy wspomaganie bieżącej działalności przedsiębiorstwa. Chcąc w prawidłowy sposób zasymulować działanie przedsiębiorstwa nawet w wąskim aspekcie jego działania, oprócz właściwego modelu pozwalającego na dokonanie symulacji należy dysponować pełnym zestawem danych będącym odzwierciedleniem stanu bieżących procesów w zadanym przedziale czasowym. Symulacja taka może mieć na celu dostrzeżenie pewnych aspektów działania zintegrowanych systemów informatycznych po reorganizacji procesów. Zanim reorganizacja zostanie dokonana w rzeczywistym przedsiębiorstwie, można ocenić jej skutki używając dostępnych na rynku narzędzi symulacyjnych. Aby dokonać takiej symulacji, często trzeba oprzeć się na rzeczywistych danych rozpatrywanej organizacji przy tworzeniu modelu symulacyjnego. Dane te mogą być pozyskane z historii wykonania konkretnych procesów biznesowych mających swoje odzwierciedlenie w istniejących narzędziach informatycznych wspomagających prace analizowanej organizacji. Aby móc w pełni skorzystać z tych danych i zaprezentować je w postaci modelu użytecznego w przeprowadzeniu eksperymentów symulacyjnych, konieczne jest posłużenie się mechanizmami translacji i akwizycji danych. Wiele ze stosowanych narzędzi symulacyjnych używa do reprezentacji danych modelu systemów relacyjnych baz danych, z tego względu uwzględnienie danych pochodzących z heterogenicznych systemów przedsiębiorstwa w modelu symulacyjnym jest możliwe po dokonaniu odpowiedniej konwersji. Na szczęście wiele narzędzi symulowania procesów biznesowych, takich jak ARIS, posiada architekturę zorientowaną serwisowo i wykorzystuje powszechnie stosowane standardy wymiany danych, np. protokół SOAP. Ze względu na różnorodność i heterogeniczność owych danych, jak również możliwość zmiany ich struktury podczas użytkowania różnorodnych rozwiązań, w tym symulacyjnych, potrzebne są mechanizmy budowy logiki biznesowej transformacji danych pozwalające na intuicyjne przedstawienie działania i łatwą modyfikację. Nowoczesne narzędzia bazodanowe oprócz samych mechanizmów zarządzania relacyjnymi bazami danych udostępniają bogaty zestaw narzędzi transformacji i dostosowania danych do konkretnych postaci. Dla prostoty przedstawienia logiki biznesowej transformacji danych, która jednoznacznie przekłada się na łatwość modyfikacji rozwiązania i późniejszą jego przydatność, narzędzia te używają graficznych języków opisu. Tendencję do graficznego przedstawienia logiki biznesowej pokazują takie rozwiązania firmy Microsoft, jak SSIS (ang. *SQL Server Integration Services*), będący częścią produktu serwera baz danych czy Workflow Foundation, przeznaczony do budowy warstwy

duktu serwera baz danych czy Workflow Foundation, przeznaczony do budowy warstwy aplikacyjnej. W powyższym opracowaniu przedstawiono wybrane z nich, które mogą ułatwić dostosowanie danych opisujących symulowaną organizację do wykorzystywanego modelu.

## BIBLIOGRAFIA

1. Zaborowski M.: Sterowanie nadążne zasobami przedsiębiorstwa ERC. Wyd. Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2008.
2. Kozielski S., Małysiak B., Kasprowski P., Mrozek D.: Bazy danych: Nowe technologie, WKŁ, 2007.
3. Vijay P. Mehta.: Pro LINQ Object Relational Mapping in C# 2008, Microsoft Press 2008.
4. Otey M.: Microsoft SQL Server 2008. Podręcznik programisty, Microsoft Press 2009.
5. Veerman E.: Microsoft SQL Server 2008 Integration Wrox, 2009.

Recenzent: Dr inż. Dariusz Mrozek

Wpłynęło do Redakcji 31 stycznia 2010 r.

## Abstract

Currently created management systems are designed and implemented by a wide range of people who have expert knowledge in various disciplines. A significant problem in achieving a good model of management system is creating such a solution which will perfectly match its user's needs on various levels of a company's functioning. The key condition of creating a simulation model which will satisfy its user's needs is applying contemporary knowledge about processes and data necessary for functioning of a company for which an IT system is designed. It is also significant to achieve coherence between a created model, a user's specification and final specification. Achieving such coherence is made difficult for designers on account of time and financial limits as well as the lack of a common language between specialists who are responsible for making assumptions and system implementation. In many cases a solution of this problem may introduce reference models. These models might match needs of similar companies. Relying on these reference models one may create by means of appropriate mechanisms of TLP translation a peculiar data saving system by means of deci-

sion mechanisms. Such solutions are often applied in decision making systems which use as a source of data OLAP cubes supplied with data from traditional OLTP systems. Using common reference models without taking into consideration knowledge about unique aspects of a particular company's specificity may cause it to become similar to other competitive companies and lose its business advantage on the market. In order to prevent this negative phenomenon it is essential to use knowledge about previous and present company's operation. Modelling business processes and data (the data which are indispensable for a company's operation) from the very basics would considerably prolong the time cost of creation of the final solution and costs connected with maintaining coherence of many unique systems and their models. Applying certain metamodels and mechanisms of data transformation from one model to another (which are offered by available commercial database systems), fast design tools and application implementation may make the realisation of this crucial task easier. In this study I tried to present some mechanisms, which are recognised by software engineering and commonly used in database tools for building personalized IT solutions. A particular attention was paid to the easiness of creating typical data translation mechanisms and data standardisation. Introducing unified standards of data saving for designing and implementation of IT systems (particularly for decision making and simulating decision solutions) is an important aspect on account of heterogeneity and multiplatformity of systems which are to be the source of data. In considered standards and models which gain popularity are these which are independent of producers and supported by worldwide open standards. One of such standards is XML language giving great possibilities by its universality. However, one cannot forget about other data models which make it possible to get valuable information about organization of already existing systems supporting functioning of a company. The modern data translation tools should also have a possibility of using other models and enabling their conversions. Standardised IT system supporting company management may be compared with other solutions by simulating suitable actions and by evaluation of received results. Mechanisms of data transformation enabling the representation of various systems based on databases in a uniform form are thus useful to make a simulative assessment of IT solutions.

## **Adres**

Paweł BUCHWALD: Wyższa Szkoła Biznesu w Dąbrowie Górniczej, ul. Ciepłaka 1c,  
44-300 Dąbrowa Górnicza, Polska, pawel@buchwald.com.pl.