STUDIA INFORMATICA 2003
Volume 24 Number 4 (56)

Marcin GORAWSKI
Politechnika Śląska, Instytut Informatyki

CHARAKTERYSTYKA PROCESU EKSTRAKCJI DANYCH

Streszczenie. W artykule zostały omówione podstawowe pojęcia związane z procesem ekstrakcji. W oparciu o doświadczenie zdobyte w trakcie realizacji wielu projektów hurtowni danych przedstawione zostały problemy (oraz ich przykładowe rozwiązania), z jakimi można się zetknąć w trakcie wykonywania rzeczywistych zadań ekstrakcji.

Słowa kluczowe: proces ekstrakcji danych, hurtownie danych

THE CHARACTERISTICS OF DATA EXTRACTION PROCESS

Summary. The article discusses the basic concepts connected with data extraction process. The experience gained during the realization of many data warehouse projects allowed to present problems (and example solutions) one could encounter performing real extraction tasks.

Keywords: extraction process, data warehouses

1. Wprowadzenie

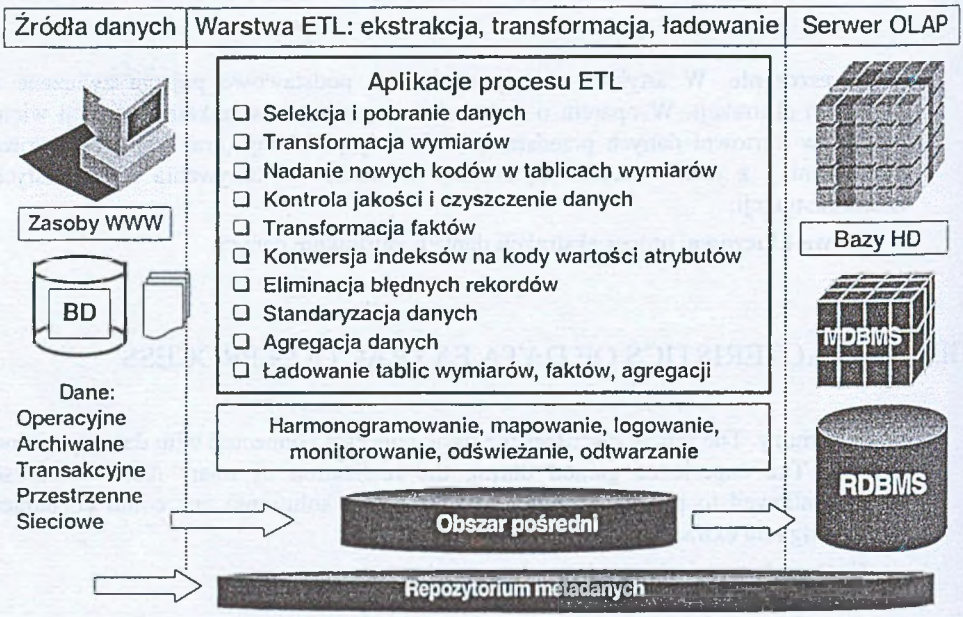
Zasadniczą koncepcją hurtowni danych (*ang. Data Warehouse*) jest połączenie danych z różnych, istniejących i pozostających nadal w użyciu baz w odpowiednio jedną kolekcję danych. Według W.H. Inmona "hurtownia danych to zorientowana obiektowo, zintegrowana, zmienna w czasie i nie ulotna kolekcja danych wspierająca proces podejmowania decyzji na szczeblu kierowniczym" [1].

Dane w hurtowni danych (HD) nie są uaktualniane w czasie rzeczywistym, jak w bazach transakcyjnych (operacyjnych), lecz są aktualizowane w pewnych odstępach czasu. Oznacza to, że dane w hurtowni nie są w pełni zsynchronizowane z danymi zawartymi w systemach transakcyjnych.

Zespół czynności związanych z przenoszeniem danych z systemów źródłowych do hurtowni danych zwany jest potocznie ekstrakcją danych. W procesie ekstrakcji wyróżnia się trzy odrębne etapy:

- pobranie danych z systemu źródłowego (*ang. Extraction*),
- przekształcenie danych do pożądanej postaci (*ang. Transformation*),
- załadowanie danych do hurtowni (*ang. Load*).

Stąd też często programy ekstrakcji określane są mianem systemów ETL.



Rys. 1. Kroki budowania hurtowni danych: proces ETL
Fig. 1. Step of Building a Data Warehouse: the ETL process

Na rysunku 1 pokazano wielowarstwową architekturę hurtowni danych z zaznaczonym procesem ETL. Warstwę ETL tworzą aplikacje procesu ETL, aplikacje zarządzające (harmonogramowanie, mapowanie, logowanie, monitorowanie, odświeżanie, odtwarzanie), obszar pośredni (*ang. Data Stage Area*) [6] oraz repozytorium metadanych. Aplikacje procesu ETL tworzą zbiór kolejno następujących po sobie zadań ekstrakcji, tj.:

- Selekcja i pobranie danych (ekstrakcja wymiarów i faktów)
- Transformacja wymiarów
- Nadanie nowych kodów wartości atrybutów w tablicach wymiarów

- Kontrola jakości i czyszczenie danych
- Transformacja faktów
- Konwersja indeksów na kody wartości atrybutów
- Eliminacja błędnych rekordów
- Standaryzacja danych
- Agregacja danych
- Ładowanie tablic wymiarów, faktów, agregacji:
  - ładowanie tablic wymiarów (przeniesienie danych do wybranej statycznej tablicy wymiarów i zmiennej tablicy wymiarów oraz przeniesienie danych do pozostałych tablic wymiarów);
  - ładowanie tablic faktów (algorytmy załadowania danych historycznych i ładowania przyrostowego;
  - ładowanie agregatów (algorytmy agregacji danych);
- Analizę logów bazy danych (automatyzacja procesu ładowania (głównie przyrostowego)).

W obszarze pośrednim dane są przygotowywane, przetwarzane, a następnie przenoszone do hurtowni danych. Jest to miejsce, gdzie dokonywana jest selekcja i pobranie danych, a następnie ich transformacja, standaryzacja i czyszczenie. Obszar pośredni może być zrealizowany jako: relacyjna baza danych, zbiór plików lub może być rozumiany jako pamięć operacyjna komputera, na którym wykonywana jest ekstrakcja. To, który z wariantów zostanie wybrany, zależy od wielu czynników, np. ilości danych oraz ich pochodzenia. Gdy danych jest niewiele lub znajdują się na tym samym komputerze co hurtownia danych, można wybrać metodę wykorzystującą pamięć operacyjną komputera. Obszar pośredni nigdy nie stanowi bazy danych, do której kierowane są zapytania użytkowników.

Gdy dane zostały już wstępnie przetworzone i zgromadzone w obszarze pośrednim, następuje ich integracja zgodnie z formatem wymaganym w hurtowni danych oraz załadowanie do systemu docelowego. Z zagadnieniem procesu ETL wiąże się również problem przyrostowego, okresowego ładowania danych do hurtowni, czyli odświeżania danych oraz ładowania uzupełniającego danych, w których wcześniej wykryto błędy.

Aby rozpocząć przenoszenie danych, muszą być wcześniej spełnione określone w [2,4,13] warunki:

- rozpoznanie struktury systemów transakcyjnych,
- wstępna selekcja danych - specyfikacja danych źródłowych do hurtowni,

- rozpoznanie wymagań stawianych hurtowni danych i określenie transformacji wykonywanych na danych źródłowych,
- zdefiniowanie architektury i modelu danych,
- implementacja baz hurtowni danych.

Dopiero po spełnieniu tych wymagań możliwe jest rozpoczęcie analizy problemu ekstrakcji danych.

Ekstrakcja jest procesem silnie obciążającym systemy komputerowe, stąd celowe jest stosowne do potrzeb przygotowanie zasobów przeznaczonych na rzecz tworzenia hurtowni danych. Warto również ustalić środowisko pracy programistów, terminologię i nazewnictwo. Wszystkie te elementy powinny być zdefiniowane przed przystąpieniem do właściwych prac projektowych i realizacyjnych. Ze względu na wielość i różnorodność zadań duży nacisk należy położyć na szczegółowe dokumentowanie tego procesu.

Generalnie projektowanie procesu ekstrakcji zamyka się w następujących zadaniach:

- Selekcja (i pobranie danych z heterogenicznych źródeł zewnętrznych)
- Czyszczenie i standaryzacja (wykrywanie błędów i ich poprawa)
- Transformacja (konwersja danych do formatu obowiązującego w HD)
- Integracja (budowa map transformacji)
- Ładowanie (sortowanie, sumowanie, konsolidowanie, wyliczanie widoków, sprawdzanie integralności, budowanie indeksów oraz partycji)
- Odświeżanie (uaktualnianie danych w hurtowni na podstawie danych źródłowych)
- Odtwarzanie (ładowanie uzupełniające HD po jej „upadku")
- Zarządzanie ekstrakcją
- Ekstrakcja równoległa

Poniżej omówimy niektóre z nich.

2. Wybrane zagadnienia procesu ekstrakcji

2.1. Selekcja danych

Podstawowym zadaniem jest odszukanie i identyfikacja danych przydatnych w tworzonej hurtowni. Wiele danych transakcyjnych nie jest potrzebnych w hurtowni. Dopiero na bazie reguł biznesowych można ocenić przydatność danych, znajdujących się w archiwach. Zadanie to jest żmudne i wymaga dużej cierpliwości. Podczas analizy danych historycznych warto zbierać informacje statystyczne o systemach transakcyjnych, z których będą czerpane informacje. Dane te są przydatne do oszacowania czasu ładowania i lepszego rozplanowania zadań.

2.2. Czyszczenie danych i standaryzacja

W całym procesie ekstrakcji najważniejsza jest kontrola poprawności danych. Przed jej rozpoczęciem należy określić kryteria, wg których będziemy oceniać poprawność załadowanych danych. Definiując te kryteria należy uwzględnić:

- zgodność danych pomiędzy systemami źródłowymi a danymi w hurtowni,
- kompletność danych (odpowiada na pytanie: czy wszystkie dane w systemach źródłowych mają swoje odpowiedniki w hurtowni danych),
- wewnętrzna spójność danych w hurtowni,
- unikalność, która gwarantuje niepowtarzalność danych w hurtowni,
- częstotliwość aktualizacji danych w hurtowni (powinna być wystarczająca na potrzeby analizy danych i wspomagania podejmowania decyzji).

W danych pochodzących z systemów źródłowych występuje wiele usterek, które wymagają korekty. Typowe usterki to [2, 4]:

- niekonsekwentne używanie kodów i znaczników,
- niektóre identyfikatory mogą mieć różną interpretację,
- dane mogą być zapisane niepoprawnie albo ten sam obiekt może być dwukrotnie wprowadzony do bazy z różnymi nazwami (duplikaty, rekordy sprzeczne).

Niektóre błędy można usunąć przed rozpoczęciem ekstrakcji przez:

- identyfikację i wybór najbardziej wiarygodnych źródeł danych,
- sprawdzenie, jaka część danych źródłowych jest niepoprawna,
- współpracę z projektantami systemu transakcyjnego przy usuwaniu wad w danych,
- współpracę z użytkownikami systemów transakcyjnych w zakresie uzupełniania danych.

Pozostałe usterki należy usuwać w obszarze pośrednim przed załadowaniem danych do HD. Ogólnie, czyszczenie danych dotyczy kilku faz [12], tj.:

- analizy danych - celem wykrycia wszystkich rodzajów błędów i niekonsekwencji, które są do usunięcia; programy analizujące powinny korzystać z metadanych;
- definiowania strumienia transformacji i reguł mapowania - w zależności od liczby źródeł, stopnia ich heterogeniczności i stopnia „zabrudzenia" danych musi być wykonana duża ilość transformacji danych i akcji czyszczenia. Schemat translacji używany jest do mapowania źródeł we wspólny model danych HD w postaci relacyjnej. Pierwsze kroki czyszczenia danych mogą poprawić jakość przygotowywanych danych w pojedynczym źródle danych do ich integracji. Dalsze kroki zajmują się schematami/danymi integracji i usuwaniem problemów wystąpień wielu źródeł, np. duplikatów. Dla HD kontrola i strumień danych dla tych

transformacji oraz kroki czyszczenia powinny być określone w strumieniu prac, który definiuje proces ETL;

- weryfikacji - poprawność i efektywność strumienia transformacji i definicji transformacji powinny być testowane i oceniane np. na próbce lub kopii danych źródłowych, aby poprawić definicje, jeśli będzie to konieczne. Wiele iteracji kroków analizy, projektowania i weryfikacji może być potrzebnych, ponieważ np. tylko kilka błędów pozostało widocznych po zastosowaniu kilku transformacji;
- poprawiania danych - powrotny strumień czyszczonych danych - po usunięciu błędów oczyszczone dane powinny zastąpić „brudne" dane również w oryginalnych źródłach w celu uniknięcia potrzeby ponownej pracy czyszczenia dla przyszłych ekstrakcji danych.

Fazy czyszczenia powinny być określone przez deklaratywny język zapytań i mapowania, aby umożliwić automatyczną generację kodu transformacji.

Standaryzacja danych zapewnia konwersje wartości atrybutów w spójny i jednolity format, co ułatwia dopasowanie i integrację danych. Na przykład wpisy daty i czasu powinny być zebrane w specyficznym formacie; nazwy i inne ciągi danych powinny być przekonwertowane na duże albo małe znaki itd. Dane tekstowe mogą być zebrane i zunifikowane przez wykonanie tematyzacji, usuwanie prefiksów, sufiksów i słów przystankowych. Co więcej, skróty i schematy kodowania powinny być spójnie rozwiązywane przez konsultacje ze słownikami synonimów lub zastosowanie predefiniowanych reguł konwersji.

2.3. Transformacja

Proces transformacji wymaga oczywiście dużych ilości metadanych, takich jak: schematy, poziomy wystąpień charakterystyk danych, mapowania transformacji, definicji przepływu danych, informacje o dziedziczonych obiektach transformowanych i ich zmianach itd. Dla spójności, elastyczności i łatwości ponownego użycia te informacje powinny być utrzymywane w repozytorium metadanych (rys.1).

Przykładami transformacji danych z obszaru pośredniego do postaci wymaganej w HD mogą być operacje:

- nadania unikalnych indeksów,
- konwersji typów danych (np. z EBCDIC na ASCII),
- dopracowania szczegółów (np. kapitalizacja pierwszych liter w imionach i nazwiskach).

O ile ostatnie dwa przykłady wymagają jedynie klasycznych mechanizmów konwersji, to nadanie nowych indeksów ma znaczenie szczególne. W systemach transakcyjnych indeksy

mają różnorodną postać, często nieodpowiednią dla przetwarzania w HD. Zdarza się, że dany wymiar pochodzi z wielu źródeł. Rzadko przy tym poszczególne wartości atrybutów wymiarowych mają unikalne indeksy. Konieczne jest wówczas stworzenie nowych indeksów, unikalnych w ramach hurtowni. Dlatego opracowuje się specjalną mapę konwersji: z indeksów transakcyjnych na indeksy hurtowni. Każda tabela wymiarów, dla której tworzone są indeksy, powinna posiadać taką mapę.

2.4. Integracja

Pierwszym zadaniem jest stworzenie schematów ekstrakcji. Wstępny schemat powinien zawierać bardzo ogólne założenia: z jakich modułów, jakie dane (jakościowo i ilościowo) oraz dokąd są przenoszone. Kolejne uściślenia poszczególnych ścieżek pozwalają na otrzymanie szczegółowej mapy przekształceń danych transakcyjnych na dane hurtowni (mapa transformacji). Taka strategia tworzenia schematów ekstrakcji i transformacji metodą zstępującą chroni nas przed przypadkową utratą pewnych połączeń. Dopiero wówczas możliwe będzie zdefiniowanie szczegółowych algorytmów transformacji danych. Staną się one podstawą praktycznej implementacji.

Zajmowanie się problemami wielu źródeł wymaga integracji schematów, zawierającej takie kroki, jak: dzielenie, scalanie, składanie, rozkładanie atrybutów i tablic [12]. Na przykład zadanie eliminacji duplikatów jest typowo wykonywane po większości transformacji i kroków czyszczenia pojedynczych źródeł danych. Eliminacja jest wykonywana na co najmniej dwóch czyszczonych źródłach w tym samym czasie lub na pojedynczym już zintegrowanym zbiorze danych. Eliminacja duplikatów wymaga identyfikacji podobnych rekordów dotyczących tego samego obiektu, a następnie ich scalania w jeden rekord zawierający wszystkie istotne atrybuty (bez redundancji). Na tym etapie pracy warto dokonać wyboru narzędzi programistycznych. Im wcześniej takiego wyboru dokonamy, tym bardziej nasz schemat będzie dostosowany do narzędzi, których będziemy używać. Pozwoli to ograniczyć konieczność ponownego projektowania niektórych transformacji.

2.5. Ładowanie tablic wymiarów

Ładowanie tablic wymiarów jest drugim etapem procesu ekstrakcji danych. Na początek należy wybrać najprostszą statyczną tablicę wymiaru. Statyczną, tzn. niezmienną, w której zestaw atrybutów jest stały i niezmienny. Powodem, dla którego dobieramy najprostszy przypadek, jest ograniczenie ewentualnych trudności do problemów związanych z komunikacją, bezpieczeństwem i transferem danych. Istotna jest tutaj kwestia wyboru metody

tworzenia i transmisji wyodrębnionych danych. Stosuje się zasadniczo dwie metody: przez plik i przez strumień.

Kiedy dane zostaną poprawnie przygotowane, można przystąpić do ładowania tablic wymiarów. Dzięki przygotowaniu danych w obszarze pośrednim proces ładowania jest stosunkowo prosty, jednak warto stosować pewne stałe reguły postępowania [2]:

- Należy unikać posługiwania się klasycznymi poleceniami wstawiania rekordów do bazy jako nieefektywnymi z powodu transakcyjnych cech relacyjnych baz danych; większość RDBMS'ów realizuje transakcje przez jednoczesne zapisywanie zmian w rejestrach transakcyjnych, co sprawia, że operacje te są bardzo czasochłonne.
- Należy używać programów do masowego wprowadzania danych (*ang. loader*).
- Wyłączyć OLTP – nawet, jeśli ładowanie się nie powiedzie, to możliwe jest powtórzenie ładowania – pliki źródłowe znajdują się w obszarze pośrednim.
- Posortować pliki źródłowe – operacja ta znacznie przyśpiesza tworzenie indeksów, natomiast należy unikać przekształcania danych podczas ładowania – nawet wykorzystanie dedykowanych programów wspomagających ładowanie może być nieefektywne, bowiem rekordy są przetwarzane i ładowane pojedynczo.
- Usunąć indeksy przed ładowaniem i utworzyć ponownie po zakończeniu ładowania – to zalecenie jest istotne, gdy ładowana jest duża porcja danych. Jako granicę przyjmuje się poziom objętości indeksów rzędu 10-15 procent bazy, po przekroczeniu którego warto indeksy usunąć i po ładowaniu odtworzyć.
- Poprawnie zdefiniować parametry bazy danych – większość obecnych baz danych automatycznie alokuje miejsce pod tworzenie nowych zapisów w bazach HD.
- Określić odpowiednie wartości wskaźników wypełnienia i przygotować odpowiednią ilość wolnego miejsca.

Gdy ładowanie najprostszej tablicy wymiarów zakończy się powodzeniem, można przystąpić do opracowania algorytmu ekstrakcji jednej wybranej zmiennej tablicy wymiarów.

2.5.1. Zarządzanie wymiarami zmiennymi

Wymiary zmienne charakteryzują się zmianami opisów, dotyczących tych samych wartości atrybutów, które znajdują się już w bazach hurtowni. Wyróżnia się trzy rodzaje obsługi sytuacji ze zmieniającymi się wymiarami [6]:

- Typ 1: Nadpisanie – „stara" wartość atrybutu znajdująca się w tablicy wymiarowej jest nadpisywana przez wartość „nową". Ten model postępowania powoduje utratę poprzedniej wartości atrybutu. Przykładem takiego atrybutu jest zmodyfikowana nazwa tego samego produktu.
- Typ 2: Utworzenie nowego rekordu w tablicy wymiarów – nowa wartość wymiaru otrzymuje nowy indeks w tablicy hurtowni wraz z nowym opisem.

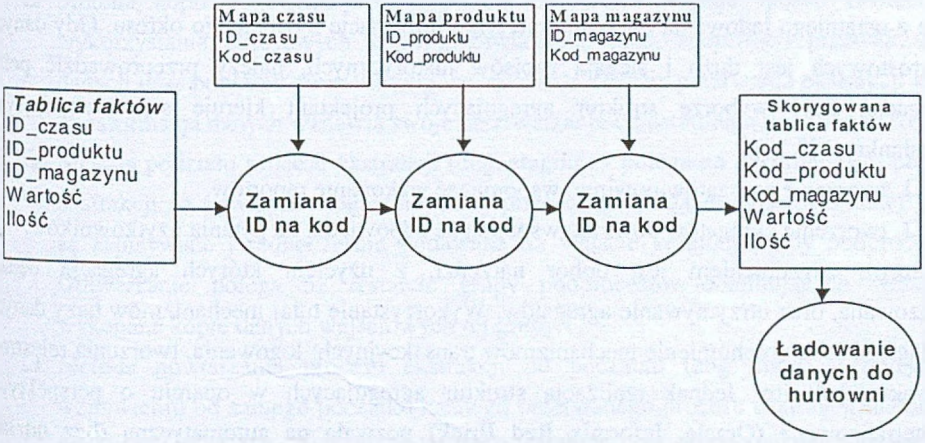
- Typ 3: Umieszczenie nowego opisu w dodatkowych polach rekordu tablicy wymiarów – w tym przypadku poprzedni i nowy opis atrybutu egzystuje w różnych polach tego samego rekordu tablicy wymiarów.

Zarządzając tablicami wymiarów można posłużyć się każdą z przedstawionych technik. Wszystkie te techniki zakładają posiadanie wiedzy o tym, że poszczególne wymiary zostały zmienione. Najczęściej stosowana, zapewniająca najlepsze efekty, jest technika typu 2.

Po przeprowadzeniu ładowania jednej z wybranych tablic wymiarów: statycznej i zmiennej, należy przeprowadzić ekstrakcję pozostałych wymiarów. Jest to realizowane w oparciu o doświadczenia uzyskane w dotychczasowych działaniach. Niektóre tablice wymiarów są znacznych rozmiarów i ich ekstrakcja charakteryzuje się cechami wspólnymi z ekstrakcją dużych tablic faktów.

2.6. Ładowanie tablic faktów

Proces ekstrakcji musi uwzględniać podział na ładowanie pierwsze i przyrostowe. Ładowanie pierwsze zakłada przeniesienie do nowo tworzonej hurtowni danych historycznych i aktualnych. Ekstrakcja danych historycznych to proces ładowania i transformacji dużych zbiorów informacji. Ładowanie przyrostowe (zwane też odświeżaniem) to przeniesienie do istniejącej hurtowni danych aktualnych, na bieżąco zbieranych w systemach transakcyjnych.



Rys. 2. Proces przekształcania tablic faktów
Fig. 2. The process of fact tables transformation

2.6.1. *Ładowanie danych historycznych*

Pierwsze ładowanie oznacza zebranie danych historycznych, czasami pochodzących z wielu źródeł (dodatkowym utrudnieniem może być fakt przechowywania tych danych na nośnikach archiwizacyjnych np. taśmach). Proces przetwarzania tablic faktów ilustruje rys.2.

Tablice faktów muszą odnosić się do indeksów hurtowni, zawartych w tablicach wymiarów. Podczas przetwarzania tablic wymiarów zostaną utworzone nowe kody wartości atrybutów każdego z wymiarów. Indeksy w źródłowych tablicach systemów transakcyjnych należy odpowiednio przekształcić na nowe kody.

Cały proces ekstrakcji należy tak kontrolować, aby w tablicy faktów nie znalazły się wartości kodów, których nie ma w tablicy wymiarów. Sprawdzenia tego należy dokonywać programowo, a nie z wykorzystaniem wbudowanych mechanizmów relacyjnej bazy danych, jakimi są więzy integralności. Postępowanie takie jest wskazane, ponieważ ładowanie każdego rekordu pociąga za sobą jednocześnie sprawdzenie więzów integralności, co z kolei wydłuża czas działania.

2.6.2. *Agregacja*

Końcowym etapem ładowania danych do hurtowni jest agregacja danych. Rekordy w tablicach agregacji tworzone są na podstawie już wygenerowanych tablic faktów. Problem pojawia się, gdy ilość danych do agregacji jest bardzo duża. Nie można wówczas zagregować wszystkich danych, a jedynie te, które zostały załadowane podczas ostatniego ładowania. Mamy tu do czynienia z agregacją przyrostową. Problemu nie można łatwo rozwiązać, gdy dane z ostatniego ładowania obejmują nie tylko informacje z ostatniego okresu. Gdy danych przyrostowych jest dużo i sięgają zapisów historycznych, należy przeprowadzić pełną agregację. Przy wyborze struktur agregujących projektant kieruje się następującymi przesłankami [9,10]:

- tworzenie agregatów powinno wspomagać wykonanie raportów,
- tworzenie agregatów powinno wspomagać odpowiedzi na pytania użytkowników.

Innym zagadnieniem jest dobór narzędzi, z użyciem których agregacja będzie realizowana, oraz utrzymywanie agregatów. Wykorzystanie tutaj mechanizmów bazy danych pociąga za sobą uruchomienie mechanizmów transakcyjnych: logowania, tworzenia rejestrów transakcyjnych itp. Jednak realizacja struktur agregujących w oparciu o perspektywy zmaterializowane (Oracle, Informix Red Brick) pozwala na automatyczną (bez udziału operatora) aktualizację struktur bezpośrednio po wypełnieniu struktur bazowych [4,11].

2.7. Odtwarzanie – powtórne ładowanie

Jeśli w trakcie ładowania danych wystąpi błąd, to jednym z możliwych sposobów postępowania jest cofnięcie całej operacji i powtórzenie jej od nowa. Lepszym rozwiązaniem jest kontynuowanie ładowania od miejsca, w którym zostało przerwane. W tym celu należy rozpoznać, do jakiego miejsca ładowanie było poprawne i nie wymaga powtarzania. Niektóre algorytmy wznowienia ładowania wykorzystują śledzenie poprawności ładowania i wymagają dodatkowego narzutu na operacje: ładowania, specyficznych transformacji danych lub wysokich kosztów odtwarzania danych.

Akcja wznowienia procesu ekstrakcji, polegająca na dokończeniu przerwanego procesu ładowania danych, nazywa jest odtwarzaniem.

Jeżeli rozważymy tylko błędy na poziomie systemowym (np. błąd RDBMS, błąd oprogramowania, brak miejsca na dysku) i przyjmiemy, że jeśli w jednym z procesów wystąpił błąd, ładowanie danych zostanie wstrzymane, to jedynymi danymi stanowiącymi podstawę wznowienia procesu ładowania są dane załadowane do hurtowni oraz dane źródłowe.

Stosowane są koncepcje postępowania - metody odtwarzania [13], tj.:

- Metoda podziału danych wejściowych (ang. batching) - dane wejściowe dzielone są na bloki danych (ang. *batch*) przetwarzane sekwencyjnie, blok po bloku. W razie przerwania procesu ekstrakcji odtwarzanie jest wznawiane od uszkodzonego bloku, pozostałe poprawnie załadowane bloki nie wymagają powtórnego przetwarzania.
- Metoda kopii migawkowych i punktów powrotu - opisuje sposób tworzenia i wykorzystania okresowych kopii migawkowych (ang. *snapshots*) przetwarzanych danych oraz punktów powrotu (ang. *savepoints*). W razie przerwania ekstrakcji każda transformacja danych wznawia swoje przetwarzanie od ostatniego punktu powrotu.
- Metoda podziału procesu ekstrakcji (ang. staging) - polega na dekompozycji procesu ekstrakcji na powiązane logicznie grupy podprocesów. Dane wyjściowe danej grupy są zapisywane i jednocześnie podawane na wejście kolejnej grupy podprocesów. Odtwarzanie polega na restarcie grupy podprocesów obejmujących wcześniej wykonane kopie danych wejściowych tej grupy.
- Metoda powtarzania procesu ekstrakcji od początku (ang. redo) – polega na wznowieniu od samego początku każdego przerwanego procesu ekstrakcji, natomiast proces ładujący dane do hurtowni filtruje przetworzone już krotki tak, aby do bazy hurtowni nie trafiały dane już tam zapisane przed wystąpieniem przerwania.
- Metoda odtwarzania Design-Resume (DR) - opisuje zmodyfikowaną metodę odtwarzania typu Redo. Algorytm DR wykorzystuje dane już przetworzone i załadowane do hurtowni danych przez przerwany proces ekstrakcji eliminując

potrzebę powtórnego przetwarzania wszystkich danych wejściowych. Dane te są wykorzystywane w czasie filtrowania danych wejściowych i danych przetwarzanych przez wznowiony proces ekstrakcji. Takie podejście zasadniczo skraca czas odtwarzania.

Odrębnym zagadnieniem jest obsługa wyjątków, czyli sytuacji, w których wykryto błąd danych źródłowych. Zwykle procedura ekstrakcji, która rozpoznała, że rekord danych przygotowany do załadowania do hurtowni nie spełnia kryteriów poprawności, zapisuje ten rekord w innym zbiorze i sygnalizuje administratorowi procesu ekstrakcji wystąpienie niepoprawnej sekwencji danych wejściowych.

2.8. Zarządzanie ekstrakcją

Proces ekstrakcji danych nie jest zadaniem jednorazowym. Dlatego bardzo ważne jest zorganizowanie zarządzania ekstrakcją przyrostową (okresowe odświeżanie danych), jak i uzupełniającą (ładowanie danych, które nie zostały umieszczone w hurtowni z powodu wykrycia błędów ładowania). Z praktyki wiadomo, że automatyczna kontrola procesu ekstrakcji powinna obejmować:

- zdefiniowanie zadań do wykonania i określenie zależności między nimi,
- zaplanowanie wykonania zadań,
- monitorowanie i rejestrowanie wyników wykonania procesów elementarnych,
- obsługę wyjątków i błędów,
- szeregowanie procesów – kończący się proces powinien zainicjować następny,
- informowanie o zakończeniu procesów i zadań.

Do przeprowadzenia automatycznej ekstrakcji wykorzystuje się najczęściej wbudowane w system mechanizmy automatycznego uruchamiania zadań. Zastosowanie specjalizowanych narzędzi ekstrakcji, które mają wbudowane mechanizmy kontroli wykonania zadań, znacznie upraszcza zarządzanie i automatyzację procesu ekstrakcji.

2.9. Przetwarzanie równoległe

Równoległe przetwarzanie stwarza ogromne możliwości zwiększenia wydajności procesu ETL. Zadania równoległe wykorzystują różne zasoby: procesor, pamięć, sieć, porty wejścia/wyjścia. Dlatego istotne jest wyodrębnienie niezależnych zadań, które wykorzystują różne zasoby. Jeśli jakaś operacja musi być poprzedzona wykonaniem innej, należy uwzględnić ten porządek wykonania – samo zrównoleglenie zadań nie zawsze powoduje przyśpieszenia wykonania ekstrakcji.

Zagadnienie ekstrakcji równoległej można rozważać w następujących aspektach [4]:

- Wykorzystanie wbudowanych mechanizmów przetwarzania równoległego danych większości relacyjnych baz danych - szczególnie indeksowanie danych może być realizowane równolegle: każdy procesor działa na swojej niezależnej części bazy;
- Zastosowanie mechanizmów bliźniaczych baz danych - utrzymujemy dwie kopie hurtowni: jedną do ładowania danych, drugą do realizacji zapytań. Jeden serwer wykonuje ekstrakcję danych, drugi obsługuje zapytania. Po zakończeniu danego etapu ładowania serwery i dane są automatycznie przełączane na hurtownię załadowaną aktualnymi danymi. Na drugim serwerze może się rozpocząć następne ładowanie;
- Zastosowanie mechanizmów bliźniaczych tabel - serwer bazy danych jest jeden, natomiast utrzymujemy dwie kopie tabel. Ładowanie odbywa się do jednej tabeli, zwanej tablicą ładowania, a zapytania są kierowane do tablicy zapytań. Po załadowaniu nowych danych do tablicy ładowania, następuje zmiana jej nazwy na tablicę zapytań - poprzednia tablica zapytań jest natomiast usuwana.

3. Przykładowy projekt ekstrakcji

Projekt procesu ekstrakcji powinien rozpocząć się od dokładnego poznania schematu hurtowni, do której będą przenoszone dane i analizy systemów i danych źródłowych pod kątem wymagań stawianych przez model danych hurtowni.

Kolejnym etapem powinno być stworzenie mapy transformacji danych. Sposób mapowania danych powinien być wykonany tak, aby umożliwiał zaplanowanie prac nad implementacją poszczególnych zadań składających się na całą ekstrakcję.

Szczegółowy opis projektu ekstrakcji na bazie przykładowego Systemu Analizy Zasobów Ludzkich (HR) i tworzonej HD[HR] przedstawiono w [5]. Tutaj przytoczono tylko najważniejsze fragmenty tego projektu. Istotnym elementem tworzonego systemu ekstrakcji są metadane procesu ETL. Mogą być one zrealizowane jako relacyjna baza danych i zlokalizowane w dowolnym miejscu – na maszynie, na której znajduje się HD lub, na której działa motor ekstrakcji. Metadane powinny być umieszczone w miejscu łatwo dostępnym poprzez sieć. Ma to znaczenie w momencie rozproszenia systemu ekstrakcji w celu zrównoleglenia przetwarzania. Aby poprawić efektywność, każda instancja motoru ekstrakcji powinna pobierać odpowiednie dane z bazy metadanych do pamięci podręcznej.

W metadanych są zwykle zapisywane następujące informacje:

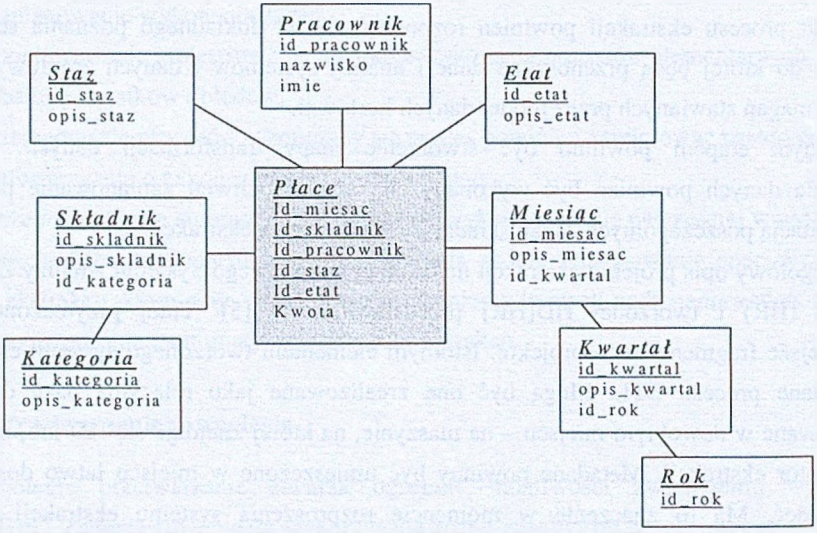
- mapa konwersji kluczy systemów źródłowych na obowiązujące w hurtowni,
- informacje o kolejnych wykonaniach ekstrakcji: ilość poprawnie przetworzonych danych, ilość wykrytych wyjątków, czas wykonania itp.,

- informacje o wykrytych wyjątkach.

Na etapie tworzenia mapy transformacji danych można dokonać podziału ekstrakcji na zadania. Przyjęto [5], iż jedno zadanie wypełnia jedną tabelę w hurtowni lub zbiór tabel ściśle ze sobą połączonych (np. tablice wymiaru Czas). Zgodnie z ogólnymi zaleceniami [7,8] definiowanie zadań ekstrakcji rozpoczynamy od tablic wymiarów. Należy zauważyć, że zawartość części z nich ma zwykle charakter statyczny. Za dynamiczne można uważać tablice wymiaru Czas. Oprócz zadania, które wypełni na początku, konieczne jest stworzenie zadań, które będą wykonywały wypełnianie przyrostowe. Po wymiarach należy rozpatrzyć tablice faktów. Postępuje się z nimi w podobny sposób jak z dynamicznymi tablicami wymiarów (ładowanie pierwsze oraz przyrostowe).

3.1. Opis hurtowni danych

Na potrzeby niniejszego artykułu przedstawiony został uproszczony schemat HD[HR] (rys.3), zawierający tylko wybrane elementy kompletnego systemu. Model ten zawiera następujące wymiary: Czas, Składniki Płacowe, Pracownik, Staż, Etat oraz jedną tablicę faktów: Płace.



Rys. 3. Uproszczony schemat przykładowej HD[HR]
Fig. 3. The simplified schema of example DW [HR]

3.2. Opis systemów źródłowych

Analiza systemów źródłowych powinna prowadzić do pełnego poznania tych ich fragmentów, które będą wykorzystywane w procesie ekstrakcji (tab.1,2).

Tabela 1

Opis tabel systemu źródłowego Kadry

Tabela	Pole	Typ	Znaczenie pola
Pracownik – przechowuje ogólne Informacje o pracowniku.	num_prac	Integer	Unikatowy numer pracownika
	nazwisko	Varchar(30)	Nazwisko pracownika
	Imie	Varchar(30)	Imię pracownika
	Plec	Char(1)	Płeć pracownika ('K','M')
Zatrudnienie – przechowuje ostatni okres zatrudnienia.	num_prac	Integer	Numer pracownika
	Wymiar_etatu	Float	Określa wymiar etatu, w jakim zatrudniony jest dany pracownik.
	data_od	Date	Początek okresu zatrudnienia
	data_do	Date	Koniec okresu zatrudnienia. (Wartość null oznacza, że zatrudnienie nadal trwa).
Zawieszenia – przechowuje okresy zawieszeń w pracy przypadające na okres zatrudnienia.	num_prac	Integer	Numer pracownika
	data_od	Date	Początek zawieszenia
	data_do	Date	Koniec zawieszenia
	Kod	Char(1)	Kod zawieszenia 1 – uwzględniać przy stażu 0 – nie uwzględniać przy stażu
Praca – przechowuje informacje o przebiegu pracy.	num_prac	Integer	Numer pracownika
	Kod	Char(3)	Określa charakter zatrudnienia.
	nazwa_zaklad	Varchar(40)	Określa zakład, w którym był zatrudniony pracownik.
	data_od	Date	Początek okresu zatrudnienia
	data_do	Date	Koniec okresu zatrudnienia

Tabela 2

Opis tabel systemu źródłowego Płace

Tabela	Pole	Typ	Znaczenie pola
StanKontRR_MM – przechowuje informacje o stanach kont.	Typ_konta	Char(12)	Określa rodzaj konta.
	num_prac	Integer	Numer pracownika
	Stan	Float	Podaje aktualny stan konta.
Składniki – przechowuje składniki definiujące postać konta.	kod_skl	Char(3)	Definiuje składnik konta.
	nazwa_skl	Varchar(50)	Podaje znaczenie składnika.
	Rodzaj	Integer	Określa rodzaj składnika.

3.3. Metadane procesu ekstrakcji

Przedstawiono przykładową tabelę (3) przechowującą metadane procesu ETL.

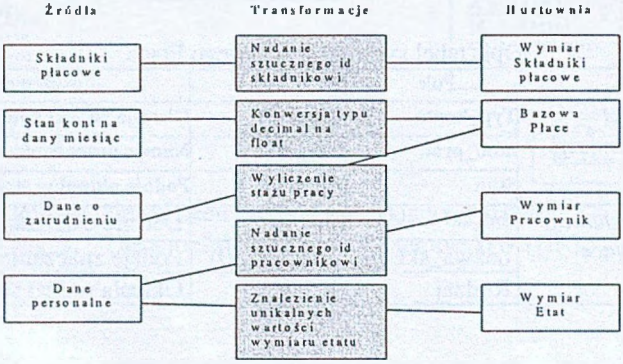
Tabela 3

Opis tabel metadanych ekstrakcji

Tabela	Pole	Typ	Znaczenie pola
Przekodowania – przechowuje mapę konwersji kluczy	rodz	Varchar(20)	Rodzaj przekodowania
	id_hurtowni	Integer	Identyfikator w hurtowni
	id_zrodlo	Varchar(10)	Identyfikator w syst. źródłowym
InstancjeEkstr – informacje o kolejnych wykonaniach ekstrakcji	id_ekstrakcji	Integer	Numer wykonania ekstrakcji
	data_od	Date	Data rozpoczęcia
	godzina_od	Time	Godzina rozpoczęcia
	data_do	Date	Data zakonczenia
	godzina_do	Time	Godzina zakończenia
	statu_wyk	Char(1)	Status wykonania
	ile_wyjatkow	Integer	Ilość wykrytych wyjątków
Wyjątki – informacje o wykrytych wyjątkach	id_ekstrakcji	Integer	Numer wykonania ekstrakcji
	id_zadania	Integer	Numer zadania ekstrakcji
	syst_zrod	Integer	Numer określa system źródłowy
	id_transakcji	Varchar(10)	Id błędnego elementu
	id_blad	Integer	Kod błędu
	Poprawiony	Boolean	Informacja o poprawieniu błędu w syst. źródłowym

3.4. Mapa transformacji danych

Poniżej przedstawiono przykładową mapę transformacji danych (rys. 4) dla schematu HD[HR] (rys. 3) oraz systemów źródłowych (tab.1, 2) [6].



Rys. 4. Mapa transformacji danych
Fig. 4. Map of data transformation

Przedstawiony ogólny plan nie zawiera elementów hurtowni, których zawartość nie pochodzi z systemów źródłowych (wymiar Czas oraz Staż). Zgodnie z zaleceniami omówionymi wcześniej definiowanie zadań ekstrakcji rozpoczynamy od tablic wymiarów.

Poniższy algorytm przedstawia sposób wypełnienia tablicy Pracownik.

1. *Ustaw się na pierwszy rekord w tablicy Kadry:Pracownik (alias z - źródło),*
2. *Jeśli koniec zbioru danych idź do 9,*
3. *Pobierz wartość pól z.num_prac, z.nazwisko, z.imie,*
4. *Jeśli w tablicy Metadane:Przekodowania istnieje wpis (rodz = „num_prac", id_zrodlo = z.num_prac) idź do pkt. 8,*
5. *Wyznacz kolejny id_pracownik,*
6. *Wprowadź do tabeli Metadane:Przekodowania rekord (rodz="num_prac", id_hurtowni=id_pracownik, id_zrodlo=num_prac),*
7. *Wprowadź do tabeli Hurtownia:Pracownik wpis (id_pracownik = z.id_pracownik, nazwisko = z.nazwisko, imie = z.imie),*
8. *Ustaw się na następny rekord w tablicy Kadry:Pracownik i idź do pkt. 2,*
9. *Koniec pracy.*

Mamy tu do czynienia z transformacją polegająca na przekodowaniu, czyli zamianie id obowiązującego w systemie źródłowym na obowiązujące w hurtowni. Jest to dosyć często stosowane przekształcenie.

Następny algorytm przedstawia sposób wypełniania tablicy Płace. Zadanie to zostało oparte na liście pracowników, którą można zrealizować jako tymczasową tabelę w metadanych ekstrakcji lub jako plik tekstowy. Założenie to przyjęto, ponieważ istnieje czasami konieczność powtórzenia ekstrakcji dla części danych pracowników. Gdy wykonuje się ekstrakcję dla wszystkich, listą pracowników może być tablica Pracownik.

1. *Ustaw się na pierwszy rekord w tablicy ListaPrac (alias l .*
2. *Jeśli koniec zbioru danych, idź do ..*
3. *Pobierz wartość pola l.num_prac,*
4. *Jeśli w tablicy Metadane:Przekodowania nie istnieje wpis (rodz = „num_prac", id_zrodlo = l.num_pra,), wygeneruj wyjątek i idź do 1.*
5. *Pobierz id_pracownik z tablicy Metadane:Przekodowania.*
6. *Jeśli w tablicy Kadry:Zatrudnienie nie ma wpisu dla przetwarzanego pracownika wygeneruj wyjątek i idź do 12.*
7. *Pobierz wartość pola wymiar_etatu z tablicy Zatrudnienie.*
8. *Jeśli w tablicy Metadane:Przekodowania nie istnieje wpis (rodz = „etat", id_zrodlo = wymiar_etatu), wygeneruj wyjątek i idź do 12.*
9. *Pobierz id_etatu z tablicy Metadane:Przekodowania.*
10. *Oblicz staż na podstawie tablic Zatrudnienie, Zawieszenie, Praca,*
11. *Pobierz wartości pól skladnik = typ_konta[9-12] oraz kwota z odpowiedniej tablicy StanKont, gdzie typ_konta[1-3] = „111".*
12. *Jeśli w tablicy Metadane:Przekodowania nie istnieją wpisy (rodz = „skladnik", id_zrodlo = skladnik), wygeneruj wyjątek i idź do 15,*
13. *Pobierz id_skladnik z tablicy Metadane:Przekodowania.*
14. *Wstaw rekord do tablicy Płace.*
15. *Ustaw się na następny rekord w tablicy ListaPrac i idź do pkt. 2.*

4. Implementacja procesu ekstrakcji

Przy pobieraniu danych z systemów źródłowych stosuje się dwa podejścia:

- **przez plik** (kroki: generowanie pliku z systemów źródłowych, przeniesienie pliku do obszaru pośredniego, przetworzenie pliku, załadowanie pliku do baz danych obszaru pośredniego) [4],
- **przez strumień** (wariant: pobranie na raz wszystkich danych do pamięci lub pobieranie danych jedynie dla pojedynczego elementu z listy, który jest w danym momencie przetwarzany) [5].

Stosowanie pierwszej z wymienionych metod wymaga posiadania odpowiedniej ilości wolnego miejsca na dysku oraz wymaga dodatkowej pracy związanej z administrowaniem pobranymi plikami. Do zalet tej metody należą: łatwa archiwizacja, możliwość szyfrowania i kompresji danych oraz prosta weryfikacja poprawności transmisji.

Często łatwiej i wygodniej zastosować metodę strumieniową, szczególnie jeśli chodzi o pobieranie i przekształcanie danych (przy relacyjnych bazach źródłowych używa się interfejsu ODBC). Dane są pobierane z systemów źródłowych, a następnie przekazywane do modułu transformacji danych, stąd natomiast są kierowane do baz danych obszaru pośredniego. Zaletą tej metody jest spójność i zwartość rozwiązania, nie jest konieczne dysponowanie dodatkową przestrzenią w pamięciach masowych na przygotowanie pliku transmisyjnego (jak to było w metodzie plikowej). Istotną wadą jest brak kompresji i ochrony danych przed dostępem osób trzecich.

5. Uniwersalne narzędzia ekstrakcji

Charakteryzując proces ETL nie można pominąć uniwersalnych narzędzi ekstrakcji. Główną zaletą uniwersalnych programów ekstrakcji jest fakt, że narzucają one określoną strategię podejścia do procesu ETL. Komplikuje to realizację określonych algorytmów, jednak niemal zawsze przyczynia się do ujednolicenia stworzonych programów.

Schematy i dane są pobierane ze źródeł danych przez rodzime bramki DBMS, jak również przez standardowe interfejsy, jak: ODBC czy EDA. Transformacje danych są definiowane za pomocą intuicyjnego interfejsu graficznego (co w znacznym stopniu zwiększa efektywność prac nad ekstrakcją) z wykorzystaniem bibliotek predefiniowanych funkcji konwersji i języka reguł. Proces transformacji jest wykonywany przez motor, który interpretuje określone transformacje w czasie pracy lub przez kompilowany kod.

Narzędzia ETL oparte na motorze np.: Copy Manager (InformationBuilders), Ardent Data Stage (IBM), Decision Base (CA/Platinium), Power Mart (Informatica), Warehouse Workbench (SystemFabrik), Warehouse Administrator (SAS) posiadają program szeregujący i wspierają strumień czynności z kompleksowym wywołaniem zależności zadań mapujących. Strumień czynności może również wywoływać narzędzia zewnętrzne np. dla zadań czyszczenia danych (eliminacji duplikatów) lub zewnętrzne procedury C/C++.

W oparciu o doświadczenia praktyczne ze stosowania Warehouse Workbench i Data Stage. przedstawiono poniżej ogólną charakterystykę tych narzędzi.

W skład uniwersalnych narzędzi ETL wchodzą zwykle następujące komponenty:

- narzędzia dla twórców programów ekstrakcji, pozwalające zdefiniować struktury źródłowe i zbiory przeznaczenia oraz reguły transformacji,
- narzędzia do harmonogramowania poszczególnych zadań ekstrakcji, które pozwalają ustalić specyficzne parametry tych zadań, jak np. liczbę rekordów odrzucanych w procesie czyszczenia danych,
- właściwy motor wykonujący wcześniej zdefiniowane oraz umieszczone w kolejce zadania; odwołuje się bezpośrednio do repozytorium, uruchamia oraz steruje przebiegiem wszystkich procesów; działanie pozostałych komponentów odbywa się za jego pośrednictwem,
- narzędzie zarządzania obiektami repozytorium,
- narzędzie administrujące, umożliwiające konfigurację projektów oraz użytkowników.

Uniwersalne narzędzia ETL udostępniają dane pochodzące ze źródeł różnego typu, począwszy od najpopularniejszych relacyjnych baz danych, poprzez pliki tekstowe o zdefiniowanym formacie, aż po dane pobierane bezpośrednio z sieci WWW.

Takie podejście wymusza zdefiniowanie warstwy pośredniej w procesie selekcji i ładowania danych, zapewniającej jednolity obraz danych na etapie transformacji.

Warstwa ta dokonuje bezpośredniego tłumaczenia odwołań do systemów źródłowych, które przykładowo może odbywać się poprzez standard ODBC dla baz relacyjnych. W efekcie końcowym otrzymujemy abstrakcyjny obiekt, stanowiący definicję źródła na poziomie programu ekstrakcji, którego składowe mogą już zostać bezpośrednio wykorzystane w definicji transformacji.

Ładowanie danych do hurtowni odbywa się w oparciu o transformację danych źródłowych. Reguły transformacji są dość elastyczne i pozwalają na definiowanie złożonych zadań. Definiując transformację podajemy obiekt źródłowy, z którego dane będą pobierane oraz obiekt przeznaczenia, do którego nastąpi ładowanie przetworzonych danych.

Porównując własności DataStage i Warehouse Workbench należy zwrócić uwagę na następujące różnice i podobieństwa [7]:

- Obydwa pakiety posiadają moduły do harmonogramowania zadań o podobnej funkcjonalności oraz podobne możliwości raportowania określonych czynności poprzez pocztę elektroniczną oraz analogiczne generatory dokumentacji.
- DataStage posiada bardziej przyjazny w użyciu interfejs graficzny użytkownika, a przez to tworzone w nim programy ekstrakcji są czytelniejsze.
- DataStage to większa elastyczność w realizacji wyrafinowanych algorytmów ETL.
- DataStage ładuje i pobiera dane do / z kilku źródeł w ramach pojedynczego zadania, gdy w Warehouse Workbench konieczne jest zdefiniowanie kilku zadań.
- W DataStage brak standardowo wbudowanego modułu do pobierania danych z sieci WWW, który jest w Warehouse Workbench.
- Język Basic używany w DataStage do konstrukcji wyrażeń transformacji jest bardziej ograniczony niż język UTL stosowany przez Warehouse Workbench.
- Brak modułu generowania danych testowych w DataStage i niewielkie możliwości testowania w Warehouse Workbench.
- DataStage posiada zdecydowanie gorsze możliwości śledzenia przebiegu transformacji w poszczególnych jej fazach.

6. Podsumowanie

Ostateczną decyzję co do tego, które z uniwersalnych narzędzi ETL zastosować lub czy implementować ekstrakcję w językach 3GL, należy podjąć po dokładnym rozpoznaniu źródeł danych oraz schematu hurtowni danych. Wtedy dopiero można oszacować złożoność procesów ekstrakcji oraz określić wszystkie korzyści oraz niedogodności każdego z tych rozwiązań.

Zaletą programowania w C++ jest elastyczność zaimplementowanych specjalizowanych narzędzi ETL i co za tym idzie - możliwość wykonania niemal każdej ekstrakcji i najbardziej złożonej transformacji. Jednak ich opracowanie i stosowanie wymaga doświadczonych programistów i projektantów znających dobrze problematykę ekstrakcji i środowiska, w którym mają działać. Natomiast uniwersalne narzędzia ETL na pewno mniej elastyczne i efektywne niż specjalizowane narzucają określoną strategię postępowania niezmiernie przydatną mniej doświadczonym projektantom. Jednak z doświadczeń wynika, że rola specjalizowanych narzędzi ETL będzie rosła wraz ze złożonością architektury hurtowni danych oraz liczebnością wolumenów i źródeł danych.

LITERATURA

1 Inmon H.: Building the Data Warehouse. John Wiley&Sons, 1996.

2 Gorawski M., Konopacki A.: Data Warehouse: Architektura. Software 2.06/1999.

3 Gorawski M., Koziatek A.: Data Warehouse: Ekstrakcja danych. Software 2.09 /1999.

4 Gorawski M., Konopacki A.: Data Warehouse: Analiza porównawcza środowisk. Software 2.10/1999.

5 Gorawski M., Koziatek A.: Systemy DSS: Projekt ekstrakcji danych.Informatyka 7-8/2000.

6 Gorawski M., Grzywacz M.: Systemy DSS: Wybrane zagadnienia implementacji procesu ekstrakcji danych przy użyciu C++. Informatyka 9/2000.

7 Gorawski M., Świtoński A.: Systemy DSS: Uniwersalne narzędzia ETL. Informatyka 10/2000.

8 Kimball R.: The Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, Inc, 1996.

9 Kimball R., Reeves L., Ross M., Thornthwaite W.: The Data Warehouse Lifecycle Toolkit. John Wiley & Sons, Inc 1998.

10 Gorawski M., Frączek J.: Data Warehouse: Modelowanie danych. Software 2.07/1999.

11 Frączek J., Gorawski M., Kozielski S..: Modelowanie struktur wielowymiarowych w hurtowniach danych. Archiwum Informatyki Teoretycznej i Stosowanej PAN, 2000.

12 Rahm E., Hong H.: Data Cleanning: Problem and Current Approaches. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2000.

13 Gorawski M., Wocław A.: Implementacja algorytmu odtwarzania Design-Resume w technologii JavaBeans, ZN Pol. Śl. Studia Informatica Vol. 24, No 1(52), Gliwice 2003.

Recenzent: Dr inż. Arkadiusz Sochan

Wpłynęło do Redakcji 9 maja 2003 r.

Abstract

The article discusses the basic concepts and problems connected with data extraction process. The experience gained during the realization of many data warehouse projects allowed to present real problems and example solutions.

The final decision on whether to use one of the universal ETL tools or to implement the extraction process in the C++ language depends on the complexity of a data warehouse architecture and "troublesome" data sources. Specialized ETL tools require experienced programmers and designers who know well the extraction issues. The universal ETL tools are less flexible and efficient than specialized tools but they impose particular implementing strategy and make the work easier. The role of specialized ETL tools grows rapidly according to the complexity of a data warehouse architecture and to the number of volumes and data sources.

Adres

Marcin GORAWSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, M.Gorawski@polsl.pl.