ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 34 Nr kol. 1381

Janusz ŚWIERZOWICZ
Politechnika Rzeszowska, Zakład Informatyki

# ZESPOŁOWE TWORZENIE APLIKACJI KLIENT-SERWER Z WYKORZYSTANIEM NARZĘDZI TYPU L-CASE

**Streszczenie.** W artykule wyjaśniono przyczyny zespołowego tworzenia systemów informatycznych. Dokonano charakterystyki strategii zintegrowanej inżynierii systemowej oraz przedstawiono przykłady oprogramowania narzędziowego do zespołowego tworzenia aplikacji. Omówiono komponenty pakietu TeamWindows, schemat przetwarzania oraz funkcje członków zespołu tworzącego aplikacje. Przedstawiono etapy tworzenia aplikacji oraz kontrolę wersji modułów.

# TEAM PROGRAMMING OF CLIENT/SERVER SYSTEMS WITH L-CASE TOOLS

**Summary.** The paper deals with System Engineering and team development of client/server systems. Description of Integrated Computer Aided System Design has been presented. The paper describes many technological elements of TeamWindows that have been synthesized to enable team programming, from a multi-user open repository to integration with systems modeling and configuration management facilities.

## 1. Wprowadzenie

W latach dziewięćdziesiątych daje się zauważyć rosnące znaczenie systemów obsługi baz danych w architekturze klient-serwer. Zasadniczym zadaniem tych systemów jest udostępnianie uprawnionym pracownikom przedsiębiorstwa informacji w dowolnym miejscu i w dowolnym czasie. Dostęp do danych realizuje się najczęściej w standardzie strukturalnego języka zapytań SQL oraz stosuje się graficzne środowisko pracy dla użytkownika systemu [1, 3, 7, 8]. Z przyczyn ekonomicznych coraz więcej zadań zarówno dla twórców aplikacji, jak

i użytkowników systemów informatycznych, przenosi się na płaszczyznę komputerów osobistych, tak wykazały badania przeprowadzone przez Forrester Group [6].

Wobec twórców aplikacji klient-serwer są stawiane różnorodne wymagania. Muszą oni posiadać wiedzę z zakresu środowisk graficznych oraz projektowania graficznego interfejsu użytkownika (GUI). Twórca powinien zdefiniować klasy opisujące działanie standardowych typów okien, określić parametry każdego okna, zdefiniować funkcje obsługi zdarzeń przekazywanych przez system oraz pętlę obsługującą zdarzenia. Powinien znać zagadnienia sieci lokalnych i rozległych, projektowania i eksploatacji baz danych, komunikacji między procesami, systemy operacyjne serwerów i sieci. Odmienną kwestią jest znajomość zasad zarządzania przedsiębiorstwem i wymagań użytkowników.

Tak wiele wymagań wobec twórców aplikacji prowadzi do specjalizacji informatyków w określonych dziedzinach, gdyż jest mało prawdopodobne, aby jedna osoba mogła zbudować i zintegrować wszystkie elementy systemu informatycznego w czasie zadowalającym użytkownika. Dlatego coraz większą rolę odgrywa zespołowe tworzenie aplikacji klient-serwer zarówno na platformie stacji roboczych [1, 3, 4, 12], jak i komputerów osobistych [2, 5-11, 13].

## 2. Zintegrowana, komputerowo wspomagana inżynieria oprogramoawnia

Zintegrowana, komputerowo wspomagana inżynieria oprogramowania, I-CASE (Integrated-Computer-Aided Software Engineering) to narzędzia służące do tworzenia aplikacji, począwszy od modelowania systemu informatycznego, a skończywszy na generacji kodu wynikowego. Stanowią one integrację oprogramowania CASE niskiego poziomu (L-CASE) i wysokiego poziomu (U-CASE). W trakcie opracowywania aplikacji często zmieniają się wymagania użytkownika. Integracja narzędzi ma na celu utworzenie dwukierunkowego połączenia, służącego do przesyłania informacji pomiędzy poziomami L-CASE a U-CASE. CASE niskiego poziomu przyjmuje informacje z wysokiego poziomu i umożliwia szybkie tworzenie całej aplikacji lub pewnych jej fragmentów.

Pakiety L-CASE często oferują możliwości zespołowego tworzenia programów i zapewniają odpowiednie środowisko do tworzenia aplikacji przez zespół.

Pakiety U-CASE są stosowane w fazie analizy i projektowania systemu informatycznego. Z ich pomocą dokumentuje się komponenty interfejsu użytkownika oraz modele procesu i danych. Możliwość przemieszczania informacji w odwrotnym kierunku (odwrócenia cyklu

projektowania) od CASE niskiego poziomu do wysokiego poziomu pozwala źle udokumentowanym systemom informatycznym na automatyczne odtworzenie dokumentacji.

### 2.1. Przyczyny zmian rynku CASE

Rynek oprogramowania CASE podlega gwałtownej transformacji ze względu na zmiany w metodologii. Zmiany te spowodowane są:

- obiektowo zorientowaną analizą procesów,
- obiektowo zorientowanym projektowaniem,
- rozwojem komputerów osobistych, sieci lokalnych i rozległych,
- przetwarzaniem w technologii klient-serwer,
- powszechnym stosowaniem interfejsów graficznych sterowanych zdarzeniami.

Pierwotnie producenci oprogramowania CASE dostarczali je jako kompletne rozwiązanie, z generatorami kodu aplikacji znakowej, uzyskanego na podstawie specyfikacji projektu oprogramowania. Nie było wówczas potrzeby włączać fazy rozwoju oprogramowania w cykl tworzenia systemu informatycznego. Takie ujęcie często zawodzi w rzeczywistości.

Ideą I-CASE jest połączenie U-CASE z L-CASE umożliwiające przepływ informacji w obie strony. Połączenie jest dwustronne, ponieważ zmieniają się wymagania użytkownika w fazie wdrażania systemu informatycznego. Jedną z najważniejszych zalet metodologii zorientowanych obiektowo jest systematyczne rozpoznawanie tych zmian. Za pomocą tych metodologii systemy informatyczne są tworzone iteracyjnie, powtarzając analizę czynności, projektowanie i wdrażanie prototypu aplikacji. Nawet jeżeli w czasie tworzenia systemu informatycznego nie stosuje się cyklu iteracyjnego, sam fakt częstych zmian specyfikacji podczas wdrażania systemu wymaga uaktualnienia i synchronizacji modeli z aktualnym kodem. W modelu tradycyjnym fazy analizy i projektowania muszą być zakończone przed rozpoczęciem wdrażania systemu - specyfikacja projektowa może się w czasie fazy implementacji zdezaktualizować. Możliwość przenoszenia informacji z L-CASE do U-CASE oznacza również możliwość automatycznej dokumentacji systemów, które formalnej dokumentacji nie miały

### 2.2. Fazowy rozwój aplikacji – modele danych i procesu

Ewolucja technologii I-CASE obejmuje dwie główne fazy. Odpowiadają one dwóm typom informacji w narzędziach U-CASE i L-CASE, wymienialnych między sobą. Pierwszy typ jest zawarty w modelu danych i zmiana tego modelu jest stosunkowo prosta. O wiele bardziej złożonym zadaniem jest współdzielenie informacji w modelach procesu i na tym właśnie koncentruje się I-CASE.

Modelowanie danych skupia się na analizie i projektowaniu systemu informatycznego ze statycznego ujęcia bazy danych. Modele danych zawierają zarówno logiczne, jak i fizyczne projekty bazy danych w postaci diagramów związków encji.

Modelowanie procesów koncentruje się na funkcjonalnym spojrzeniu na system jako na sterowalną sieć działania aplikacji. Do tego celu wykorzystuje się najczęściej diagramy przepływu danych, schematy blokowe, diagramy stanów przejściowych.

W obecnej technologii jest osiągalna wymiana informacji pomiędzy pakietami CASE na poziomie modeli danych:

- o tabelach, kolumnach, relacjach między kluczami głównymi i obcymi,
- o rozszerzonych atrybutach danych, takich jak reguły poprawności, formaty rysunków, kolory i czcionki.

Druga faza rozwoju narzędzi I-CASE zapewnia ściślejszą integrację z modelem procesu. Umożliwiają one generowanie komponentów aplikacji zawierających logikę programu oraz złożone reguły poprawności zawarte w modelach. Innymi słowy, następuje powrót do generacji kodu na podstawie wymagań określonych przez użytkownika, z tą różnicą że pakiety CASE są dostarczane przez różnych producentów. Od narzędzi stosowanych w I CASE zależy, jak jest realizowana wymiana danych.

### 2.3. Przykłady rozwiązań I-CASE

W dalszej części artykułu zostanie przedstawiona inżynieria I-CASE opracowana przez firmę CENTURA. Głównym elementem L-CASE jest pakiet o nazwie TeamWindows [6], współpracujący z pakietami U-CASE wielu czołowych producentów, takich jak:

- Bachman Information Systems (Bachman Generator),
- Evergreen CASE Tools (EasyCASE),
- KASEWORKS, Inc. (KASE:VIP for Windows)
- Learmonth & Burchett Management Systms, Inc. (SE/Open for TeamWindows),
- Logic Works (ERWin/ERX ),
- Popkin Software and Systems, Inc. (System Architect for Windows Team Windows Link),
- Visible Systems Corp. (The Visible Analyst Workbench).

## 3. Charakterystyka pakietu TeamWindows – narzędzia do programowania zespołowego

Pakiet TeamWindows opracowano dla projektantów systemów informatycznych mających praktykę w tworzeniu aplikacji dla komputerów mainframe, a odpowiadających za rozwój aplikacji klient-serwer. Różni się od narzędzi znanych z komputerów mainframe:

- graficznym pośrednictwem użytkownika,
- możliwością funkcjonowania na komputerach PC w sieci lokalnej lub rozległej.

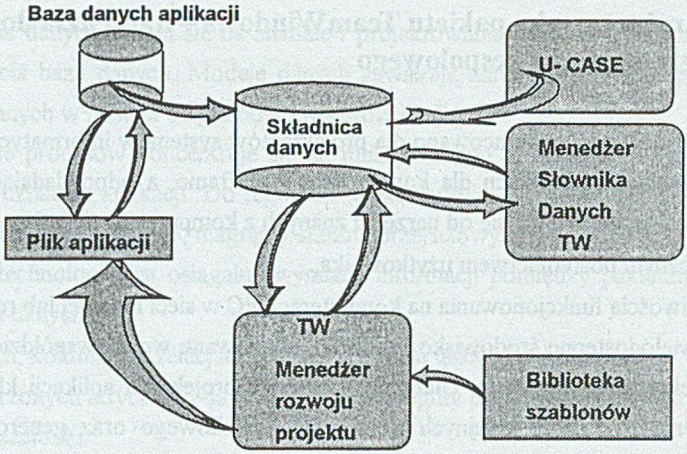
Jest to wielodostępne środowisko graficzne, zbudowane wokół współdzielonej składnicy danych projektu - repozytorium. Ułatwia zarządzanie projektami aplikacji klient-serwer poprzez śledzenie i kontrolę kolejnych wersji kodu źródłowego oraz generowanie aplikacji z wykorzystaniem szablonów i słownika danych. W pakiecie tym została opracowana architektura wymiany informacji z CASE (CASE Interchange Architecture) wokół otwartej składnicy danych (Open Repository).

Składnica danych zawiera rozszerzony słownik danych o aplikacji. Oprócz struktur danych są tam zawarte informacje i pliki związane z projektem, np. moduły i szablony dla środowiska SQLWindows, informacje o zespole projektowym i o prawach dostępu poszczególnych członków zespołu do elementów projektu, dokumenty z edytorów tekstu, ikony, mapy bitowe oraz szczegółowe informacje o każdym ekranie aplikacji. Do głównych zadań wykonywanych za pomocą pakietu zalicza się:

- zarządzanie projektami aplikacji,
- obsługę kodu źródłowego aplikacji tworzonej przez zespół programistów,
- obsługę składnicy danych,
- opracowywanie szablonów i ekranów dla środowiska *SQLWindows*.

### 3.1. Komponenty pakietu TeamWindows

W pakiecie TeamWindows występują (rys. 1): menedżer projektów (TeamWindows Project Development Manager), menedżer słownika danych (Data Dictionary Manager), składnica danych projektu (Data Repository), biblioteki szablonów (Template Libraries).




Rys. 1. Komponenty pakietu TeamWindows
Fig. 1. TeamWindows components

### *3.1.1. Menedżer rozwoju projektu*

Jest to pakiet służący do zarządzania projektami, generowania kodu aplikacji, obsługi biblioteki szablonów oraz do tworzenia ekranów. Głównym elementem pakietu jest moduł okna projektowania. Moduł ten dostarcza twórcom aplikacji graficzne pośrednictwo dla tworzenia formularzy i okien dialogowych, wstawianych do opracowywanej aplikacji. W module występuje pasek menu, pasek narzędzi z przyciskami przywołania często używanych opcji menu oraz tabele zawartości ekranu i atrybutów. Moduł okna projektowania zawiera również paletę narzędzi, z obiektami do umieszczania w formularzu, oraz połączenie SQLLink do ustalania relacji pomiędzy obiektami ekranu i kolumn w bazie danych. Moduł okna projektowania jest podobny do okna projektów w środowisku SQLWindows. Tabela zawartości ekranu zawiera listę kodowanych kolorem ekranów, będących częścią otwartego projektu.

### *3.1.2. Projekt*

Projekt jest centralną koncepcją w pakiecie TeamWindows. Na pulpicie symbolizuje go ikona skoroszytu. Każdy projekt zawiera zarówno moduły kodu źródłowego środowiska SQLWindows, jak i inne pliki, np. ikony, mapy bitowe, modele danych, nagrane dźwięki, dokumenty analizy i projektowania, diagramy, kontrakty, arkusze kalkulacyjne związane z tworzoną aplikacją. Projekty są również skojarzone z bazą danych aplikacji. Jeżeli używamy generatora kodu, zostają wymuszane standardy kodowania, np. konwencja nazw. W projekcie są uwzględniane także informacje o członkach zespołu twórców aplikacji wraz z przypisanymi im poziomami dostępu do danych.

TeamWindows wymaga przestrzegania określonych standardów w tworzeniu i rozwoju projektu. W każdym projekcie występują: osoba zarządzająca projektem - administrator projektu - oraz zespół projektowy.

Aby rozpocząć nowy projekt, należy wykonać szereg administracyjnych czynności, np. należy wprowadzić informacje o zespole opracowującym projekt i o bazach danych używanych przez aplikację.

Projekty są połączone w składnicy danych wraz z danymi o zespole projektowym oraz ze ścieżkami dostępu do trzech katalogów: dla modułów testowanych, współdzielonych oraz modułów w fazie produkcji. Każdy członek zespołu projektowego posiada indywidualny katalog.

Po zainicjowaniu projektu do składnicy danych zostaną dodane po raz pierwszy pliki lub moduły. Można również określić powiązania między modułami, co jest wygodne podczas tworzenia raportów lub gdy moduł jest wypisywany. Istnieje opcja kopiowania wszystkich modułów związanych z modułem wybranym bezpośrednio.

TeamWindows ma również funkcję promocji, która przesuwa moduły z fazy testowania do fazy produkcji. Katalog testowania zawiera programy wykonawcze SQLWindows i inne związane pliki, sprawdzane przez członków zespołu testującego.

Katalog produkcji zawiera moduły wykonawcze promowane do fazy produkcji. Katalog współdzielenia zawiera dołączalne moduły obiektów SQLWindows i bibliotek z funkcjami i oknami dialogowymi wykorzystywanymi przez twórców aplikacji. Na koniec określa się liczbę archiwowanych wersji plików aplikacji i metodę ich przechowywania w kolumnach składnicy danych typu longvarchar albo jako pliki w katalogu. Informacje o składzie zespołu projektowego oraz o strukturze bazy danych aplikacji są również wprowadzone do składnicy.

### *3.1.3. Poziomy dostępu twórców aplikacji do składnicy danych*

Użytkownikowi pakietu TeamWindows zostaje przydzielony jeden z czterech poziomów dostępu do tworzonej aplikacji. Poziomy są hierarchiczne - wyższy poziom ma wszystkie cechy niższego poziomu. Są one zgodne z funkcjami: administratora projektu, menedżera projektu, programisty oraz testera, pełnionymi w zespole projektowym.

Po uruchomieniu pakietu TeamWindows twórca aplikacji podaje identyfikator zespołu projektowego, imię i nazwisko oraz poziom, do którego może promować moduły. W tabeli 1 określono prawa twórców aplikacji, a na rys. 2 przedstawiono model zarządzania projektami.

Tabela 1

Uprawnienia twórców aplikacji

| Rodzaj uprawnienia | Administrator | Menedżer | Projektant | Tester |
|---|---|---|---|---|
| Dostęp do pakietu *TeamWindows* | + | + | + | + |
| Menedżer Słownika Danych | + | + | + | – |
| Obsługa zespołu projektowego | + | – | – | – |
| Import i obsługa wersji bazy danych | + | +* | +* | – |
| Tworzenie nowego projektu | + | – | – | |
| Zarządzanie projektami | + | + | – | – |
| Przydzielanie projektów członkom zespołu | + | + | – | – |
| Obsługa modułów projektu | + | + | + | – |
| Odblokowanie opcji "*check out*" dla modułów projektu | + | + | + | – |
| Obsługa formularzy | + | + | + | – |
| Wypisywanie modułów w fazie testowania od promocji do produkcji | + | +** | +** | + |
| Obsługa i kompilacja szablonów | +*** | +*** | +*** | – |

W tabeli 1 przyjęto następujące oznaczenia uprawnień:

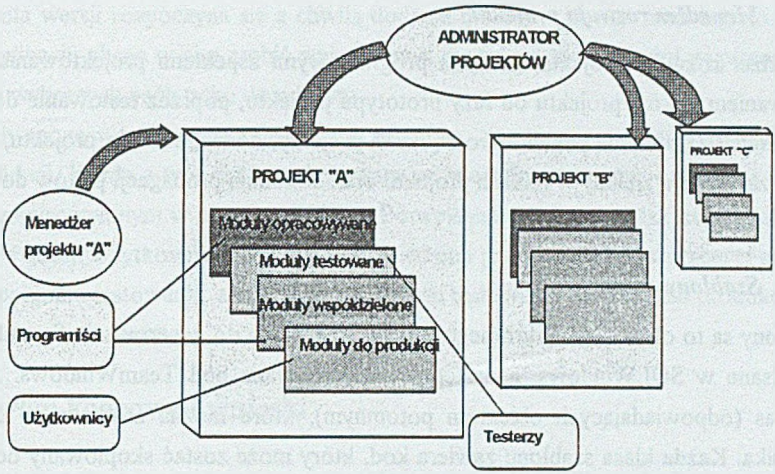
– brak uprawnienia,

\+ posiadane uprawnienie,

+* posiadane uprawnienie administratora bazy danych,

+** uprawnienie odpisywania modułów w fazie testowania,

+*** uprawnienie inżyniera szablonów.




Rys. 2. Model zarządzania projektami w systemie TeamWindows
Fig. 2. A model of projects management in TeamWindows

### *3.1.4. Otwarta składnica danych o projekcie*

Główną funkcją słownika danych jest przechowywanie i obsługa informacji o bazie danych aplikacji. Informacja ta jest przechowywana w otwartej składnicy danych. Istotną jej częścią jest relacyjna baza danych zaimplementowana w systemach SQLBase, Oracle lub Sybase. W składnicy przechowywane są informacje o projektach, plikach projektowych, o strukturach baz projektów oraz o osobach administrujących, tworzących i sprawdzających pliki projektowe. Zawarte są tam nazwy tabel i kolumn, klucze główne i obce, typy kolumn bazy danych, nazwa serwera itp. Mogą być przypisane również trzy typy atrybutów kolumn:

- atrybuty SQL odnoszące się do kolumny (typ i długość),
- atrybuty SQLWindows opisujące charakterystykę kolumn w odniesieniu do aplikacji SQLWindows, tj. kolor, czcionka, nazwa klasy, wyrównywanie,
- atrybuty definiowane przez użytkownika określają dodatkowe charakterystyki obiektów ponadstandardowe atrybuty SQL, np. komunikaty, reguły poprawności.

Później, gdy kolumny są używane w oknie projektowania TeamWindows, gdzie konstruowane są dialogi i formularze, okno potomne jest połączone z kolumną, dziedzicząc bieżące atrybuty wyspecyfikowane w słowniku danych.

### *3.1.5. Menedżer składnicy*

Menedżer składnicy zawiera wiele funkcji umożliwiających importowanie, eksportowanie i obsługę informacji o tabelach i kolumnach używanych w aplikacji.

### *3.1.6. Menedżer rozwoju projektu*

Menedżer rozwoju projektu zarządza programowymi aspektami projektowania i steruje propagowaniem plików projektu od fazy prototypu projektu, poprzez testowanie do produkcji. Projektanci za pomocą pakietu tworzą nowe projekty, dodają pliki do projektu, rozwijają aplikację, zachowują zmiany w plikach projektu oraz dokonują propagacji plików do fazy produkcji.

### *3.1.7. Szablony*

Szablony są to obiekty wielokrotnego użycia, stosowane do generowania formularzy. Zostały napisane w SQLWindows, a następnie skompilowane pod TeamWindows. Zawierają szereg klas (odpowiadających obiektom potomnym), które można zastosować na ekranie użytkownika. Każda klasa szablonu zawiera kod, który może zostać skopiowany do aplikacji SQL-Windows w czasie zapamiętywania ekranów.

## 3.2. Proces tworzenia aplikacji za pomocą TeamWindows

Proces tworzenia aplikacji rozpoczyna się od zdefiniowania projektu. Początkowo, najwięcej wysiłków wymaga dodawanie nowych modułów do projektu. Modułami mogą być formularze lub szablony SQLWindows, pliki ikon używane w przyciskach aplikacji oraz diagramy związków encji z pakietu U-CASE. Gdy moduły zostaną dodane do projektu, rozpoczyna się kontrola wersji. Tylko jedna osoba z zespołu może mieć wypisany moduł. Gdy moduł jest z powrotem wpisany do systemu, otrzymuje nowy numer wersji.

Twórcy aplikacji mogą w modułach zmieniać lub dodawać nowe obiekty ekranowe. Obiektami mogą być pola danych, przyciski, listy wyboru i inne elementy okien potomnych. Szablon, który sprawdza paletę narzędzi, może mieć zdefiniowane klasy okien. Obiekty są kojarzone z jedną z tych klas i mogą być również łączone z kolumnami tabeli bazy danych.

## 3.3. Kontrola wersji

Zarządzanie różnymi wersjami modułów oprogramowania oraz scalanie ich w procesie tworzenia i promowania nowych modułów wykonawczych - uwzględniając testowanie i produkcję - są złożonymi czynnościami. Złożoność uwydatnia się również w przypadku współdzielonych modułów. Kontrola wersji jest jednym z ważniejszych aspektów cyklu tworzenia aplikacji. Jest ona konieczna w zarządzaniu projektem i jest doskonałym narzędziem diagnostyki wtedy, gdy próbujemy prześledzić, w których liniach kod został zmieniony, dodany lub usunięty. Zmiany te można prześledzić w oddzielnych wersjach modułu.

Kontrola wersji rozpoczyna się z chwilą dodania modułów po raz pierwszy do projektu. Twórca aplikacji, chcąc w nim zrobić zmiany, musi moduł wypisać. Moduł wypisany jest dostępny dla wybranych osób tylko do odczytu.

Moduły mogą występować w jednej z trzech faz: rozwoju, testowania-współdzielenia i fazie produkcji. Każda z tych faz jest skojarzona ze specyficzną ścieżką katalogu plików, zazwyczaj umiejscowionym w serwerze plików. Po wpisaniu moduł jest traktowany jako moduł w fazie pierwszej. Użytkownik pakietu z odpowiednimi prawami dostępu przenosi moduł najpierw do poziomu testowania, a następnie z poziomu testowania do poziomu produkcji.

## 4. Uwagi końcowe i wnioski

Systemy I-CASE są stosowane od niedawna i brak jeszcze jednoznacznych stwierdzeń o ich użyteczności. Pakiety U-CASE przechodzą ewolucję od generatorów znakowych aplikacji w językach III generacji do modeli aplikacji sterowanych zdarzeniami, pracujących w architekturze klient - serwer. Zintegrowana inżynieria systemowa jest coraz częściej realizowana na komputerach osobistych, przy czym pakiety U-CASE koncentrują się na analizie i projektowaniu systemów informatycznych, natomiast pakiety L-CASE na wdrażaniu tych systemów. Konkurencja na rynku producentów pakietów U-CASE dla różnych pakietów L-CASE obniża koszty systemów I-CASE, przyczyniając się do wzrostu zainteresowania tego typu oprogramowaniem przez zespoły programistów tworzących duże aplikacje obsługujące bazy danych w środowisku graficznym.

Przedstawiony artykuł daje przykład sposobu rozwiązania problemów, jakie występują podczas projektowania informatycznych systemów zarządzania bazami danych.

Autor dziękuje firmie Techmex S.A. za udostępnienie pakietów Gupta SQLWindows Corporate Edition oraz SQLBase Server w ramach programu promocyjnego „Gupta Scientific Partner".

## LITERATURA

[1] Enabling Team Development in Client/Server Environments with INFORMIX - New Era. Informix Software Inc., July 1995.

[2] Goddard D.: Group Programming with TeamWindows. DBA, July 1993, s. 61-62.

[3] INFORMIX - New Era, Technical Brief, Informix Software Inc., 1995.

[4] Object Team for OMT, Cadre Technologies Inc, 1995.

[5] Russel W.P., Maher D.: SQLWindows Team Programming. Jarrah Technologies Ltd., Gupta Corp, 1994.

[6] Syed R.: Team Programming - concept, requirements, solutions. Gupta Corp., 1994.

[7] Świerzowicz J.: A Computer Modelling of Decision Support System for DBMS Choice, 1-st International Modelling School, Alushta,1996.

[8] Świerzowicz J.: Nowoczesne technologie tworzenia systemów zarządzania bazami danych, Konferencja „Wspomaganie komputerowe w eksploatacji samolotów", Politechnika Warszawska, Warszawa, listopad 1994.

[9] Świerzowicz J.: Podstawy tworzenia graficznego środowiska dostępu do SQL-owych baz danych. ZN Politechniki Rzeszowskiej, Zarządzanie i Marketing Z.1., Nr 143, 1996, s. 108-127.

[10] Świerzowicz J.: Rola TeamWindows w tworzeniu aplikacji klient-serwer, Seminarium Gupta Club 6, Międzybrodzie Żywieckie, Wrzesień 1995.

[11] Świerzowicz J.: Rozwój projektu systemu informatycznego z wykorzystaniem pakietu Team-Windows, Seminarium Gupta Club 7, Bielsko Biała, Listopad 1995.

[12] Świerzowicz J.: Zespołowe tworzenie systemów zarządzania SQL-owymi bazami danych. III Krajowa Konferencja „Komputerowe wspomaganie badań naukowych", WTN, Wrocław 1996,

[13] Vaskevitch D.: Strategie klient-serwer. Systemy informatyczne w procesie przekształcania przedsiębiorstwa. IDG Books, Warszawa 1995.

Recenzent: Dr inż. Marcin Skowronek

Wpłynęło do Redakcji 24 grudnia 1997 r.

**Abstract**

In this paper the reasons for a team development of client/server systems are explained. The characteristics of an Integrated Computer Aided System Engineering -I-CASE strategy is brought about and a module of software Lower CASE L-CASE tool for a team development of application-TeamWindows is presented.

The components of a module are presented in fig.1. A scheme of processing and the function of a team, which develops applications, are discussed and described in fig. 2. and in table 1. Also, the stages of developing of application and the control of module version are described. Many technological elements of TeamWindows, are synthesized to enable team pro-

gramming, from a multi-user open repository to integration with systems modeling and configuration management facilities.