

XV Krajowa Konferencja Automatyki

Tom II



**Redaktorzy:
Zdzisław Bubnicki
Roman Kulikowski
Janusz Kacprzyk**

XV Krajowa Konferencja Automatyki Tom II



Redaktorzy:
Zdzisław BUBNICKI
Roman KULIKOWSKI
Janusz KACPRZYK

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓŁORGANIZATORZY

Politechnika Warszawska

Przemysłowy Instytut Automatyki i Pomiarów

Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓLORGANIZATORZY

Politechnika Warszawska
Przemysłowy Instytut Automatyki i Pomiarów
Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

KOMITET PROGRAMOWY

Przewodniczący	Zdzisław BUBNICKI
Zastępca Przewodniczącego	Roman KULIKOWSKI

CZŁONKOWIE

Stanisław BAŃKA	Michał BIAŁKO
Mikołaj BUSŁOWICZ	Władysław FINDEISEN
Ryszard GESSING	Henryk GÓRECKI
Jakub GUTENBAUM	Jerzy JÓZEFczyk
Stanisław KACZANOWSKI	Tadeusz KACZOREK
Janusz KACPRZYK	Jerzy KLAMKA
Józef KORBICZ	Zbigniew KOWALSKI
Krzysztof KOZŁOWSKI	Juliusz L. KULIKOWSKI
Krzysztof KUŹMIŃSKI	Kazimierz MALANOWSKI
Krzysztof MALINOWSKI	Wojciech MITKOWSKI
Antoni NIEDERLIŃSKI	Władysław PEŁCZEWSKI
Tadeusz PUCHAŁKA	Leszek RUTKOWSKI
Stanisław SKOCZOWSKI	Roman SŁOWIŃSKI
Jerzy ŚWIĄTEK	Andrzej ŚWIERNIAK
Ryszard TADEUSIEWICZ	Piotr TATJEWSKI
Krzysztof TCHOŃ	Leszek TRYBUS
Jan WĘGLARZ	Andrzej P. WIERZBICKI

KOMITET ORGANIZACYJNY

Przewodniczący	Roman KULIKOWSKI
Zastępcy Przewodniczącego	Janusz KACPRZYK
	Stanisław KACZANOWSKI
	Tadeusz KACZOREK
	Krzysztof MALINOWSKI
Członkowie	Roman OSTROWSKI
	Tadeusz PUCHAŁKA
	Dariusz WAGNER
Sekretarze naukowci	Jan STUDZIŃSKI
	Jan W. OWSIŃSKI

ISBN 83-89475-01-4

Copyright © Instytut Badań Systemowych Polskiej Akademii Nauk
All rights reserved

Druk: ARGRAF, Warszawa

STEROWANIE
I TECHNIKA KOMPUTEROWA

UML-OWY MODEL SYSTEMU NADZOROWANIA TRANSMISJI DANYCH PRZEZ STEROWNIK

Karol GRANDEK*, Andrzej GACEK**

* Politechnika Opolska, Wydział Elektrotechniki i Automatyki
ul. K.Sosnkowskiego 31, 45-272 Opole, e-mail: grandek@po.opole.pl

** Instytut Tele- i Radiotechniczny
ul. Ratuszowa 11, 03-450 Warszawa, e-mail: agacek@itr.org.pl

Streszczenie: W referacie przedstawiono możliwości zastosowania metodyki UML do analizy, projektowania i implementacji systemu czasu rzeczywistego, którym jest system nadzorowania transmisji danych przez sterownik. Model systemu w metodyce UMLowej obejmuje szereg diagramów obrazujących różne perspektywy procesu przetwarzania danych. W zależności od wymagań model zawiera wybraną grupę diagramów i innych narzędzi. Podstawowymi są diagram przypadków użycia, diagram klas i obiektów, diagramy interakcji. W referacie opisano reprezentatywne diagramy dla wymiany danych opartej na zasadzie Master-Slave.

Słowa kluczowe: Sterowniki programowalne, model UML-owy.

1. WPROWADZENIE

W referacie przedstawiono możliwości zastosowania metodyki UML do zaprojektowania i implementacji systemu informatycznego nadrzędnego, który - pełniąc rolę MASTERA w połączeniu ze sterownikiem - wykonuje następujące zadania:

- ustawia parametry transmisji,
- ustawia ręcznie lub dynamicznie czasy oczekiwań na odpowiedź dla poszczególnych ramek,
- ustawia długości ramek w zależności od ilości przesyłanych danych,
- ustawia wartości pojedynczych rejestrów lub grupy rejestrów,
- wyświetla odebrane dane w różnych formatach,
- testuje poprawność transmisji, wysyła polecenia sterujące.

Sterownik - stacja typu slave - używa jako protokołu komunikacyjnego standard MODBUS z portami szeregowymi RS 232 i RS 485. Systemy komunikacyjne należą do grupy systemów wbudowanych a metodyka UML jest obecnie coraz częściej stosowana do ich modelowania. UML oferuje szeroką gamę narzędzi, które mogą być opcjonalnie wybrane dla prawidłowego wyspecyfikowania określonego systemu. Narzędziami używanymi w każdym niemal systemie są: diagram przypadków użycia (PU), diagram klas i obiektów oraz diagramy sekwencji dla przypadków użycia, diagram

współpracy dla klas/obiektów i diagramy stanów. Dwa pierwsze diagramy modelują statyczne aspekty natomiast pozostałe - aspekty dynamiczne systemu. Wspieranie tych narzędzi przez oprogramowanie CASE zapewnia im zastosowanie na wielu etapach (fazach) cyklu życia oprogramowania (od etapu analizy/modelowania do etapu implementacji).

2. ANALIZA SYSTEMU NADZOROWANIA TRANSMISJI DANYCH

Standard komunikacyjny MODBUS opracowała firma Modicon. Protokół jest standardem przesyłania danych w sieciach komputerów i sterowników przemysłowych (w tym PLC), urządzeń kontrolno-pomiarowych. Wymiana danych w MODBUS oparta jest na zasadzie Master - Slave. Urządzenie Master (inicjator komunikacji) pełni rolę systemu nadrzędnego wysyłającego zapytania oraz informacje sterujące do urządzeń (stacji) typu Slave, które odpowiadają na zapytania i wykonują określone akcje. Wysyłane i odbierane dane są sformatowane w postaci ramek zawierających dodatkowo adres urządzenia oraz bity synchronizujące, kontroli błędów i korygujące (CRC). Protokół MODBUS posiada cztery rodzaje przekazywanych informacji (formatów ramek): zapytanie, odpowiedź, odpowiedź o błędzie, rozgłoszenie oraz dwa tryby kodowania ramek: tryb ASCII i tryb RTU. Przykład ramki zapytania dla funkcji o kodzie 3 (0x03) - czytanie binarnej zawartości grupy rejestrów pokazano w tabeli 1.

Tabela 1. Ramka odczytu wartości rejestrów począwszy od 0x1002

Pole ramki	Wartość (hex)
Adres urządzenia	0F
Funkcja	03
Adres rejestru startowego Hi	10
Adres rejestru startowego Lo	01
Liczba rejestrów Hi	00
Liczba rejestrów Lo	08
LRC lub CRC	--

Projektowanie systemu z zastosowaniem metodyki UML obejmuje kilka faz, z których pierwsza to analiza systemu. Analiza systemu rozpoczyna się od kompletnego opisu systemu. Jego uproszczona wersja przedstawiona poniżej powinna umożliwić identyfikację aktorów i sformułować przypadki użycia.

Komunikacja z użyciem protokołu MODBUS polega na przesyłaniu informacji pomiędzy systemem nadrzędnym a sterownikiem lub jego symulatorem. Dla systemu nadrzędnego przewidziano dwa podstawowe typy aktorów: jest nim użytkownik – określany dalej jako „USER”, którego podstawową rolą jest obsługa systemu nadrzędnego oraz administrator – określany dalej jako „SERVIS”, który posiada większe możliwości obsługi systemu, a także ma znacznie większy dostęp do określonych danych, odczytywanych ze sterownika/symulatora. Aby przystąpić do użytkowania systemu należy się zalogować z odpowiednimi uprawnieniami w celu uzyskania dostępu do różnych funkcji systemu. System pozwala użytkownikowi na skonfigurowanie portu komunikacyjnego. Użytkownik ma do dyspozycji kilka portów komunikacyjnych typu COM. Po wybraniu odpowiedniego portu użytkownik może skonfigurować jego parametry transmisji lub przyjąć domyślne. Przy starcie systemu wartościami domyślnymi są ostatnie wartości ustawione w aplikacji. W oknie komunikacji dostępne są trzy zakładki, na których użytkownik może sprawdzić aktualny stan pracy systemu, aktualnie obsługiwaną rankę transmisji, ustawienia portu oraz dostępne sterowniki, które zostały wykryte w sieci. Na jednej z zakładek znajdują się informacje dotyczące stanu poszczególnych portów komunikacyjnych. Po ustawieniu parametrów transmisji użytkownik może przejść do wykrywania urządzeń typu slave dostępnych w sieci. Wykrywanie może odbywać się na dwa sposoby: poprzez ustawienie konkretnego adresu urządzenia slave lub też poprzez skanowanie całego zakresu adresów (od 0 do 247). Wykrywanie dostępnych urządzeń polega na wysyłaniu ramki identyfikacyjnej o danym adresie i odpowiednim kodzie. Jeżeli sterownik otrzyma taką ramkę zawierającą jego adres, wysyła informację zwrotną do systemu. W informacji tej przesyłany jest kod identyfikacyjny urządzenia na podstawie, którego urządzenie to jest identyfikowane przez system nadrzędny. Informacje wstępne dotyczące danego urządzenia przechowywane są w pliku. Po zidentyfikowaniu urządzenia, system ładuje biblioteki danych odczytane plików i wysyła ramkę odczytującą grupę rejestrów, zawierającą informacje szczegółowe, dotyczące identyfikacji danego sterownika. Użytkownik zostaje powiadomiony o przeprowadzonej identyfikacji i zaczyna się cykliczne odpytywanie sterownika/symulatora o poszczególne grupy rejestrów. Użytkownik może wybrać daną grupę zasobów i przeglądać jej zawartość. Przeglądanie zawartości rejestrów jest możliwe na trzy sposoby: poprzez wartości domyślne, wartości w jednostkach oraz wartości heksadecymalne. Każde z okienek zawierających daną grupę rejestrów jest przystosowane dla tej grupy. Okienko takie zawiera charakterystyczne dla danej grupy rejestrów polecenia oraz format informacji graficznej. Użytkownik posiada możliwość zapisu i odczytu szablonów i zasobów danych.

3. MODEL SYSTEMU

Model systemu w metodyce UMLowej obejmuje szereg diagramów obrazujących różne perspektywy procesu przetwarzania danych. Perspektywy opisują strukturę i dynamikę systemu. Wymagania modeluje się za pomocą przypadków użycia oraz diagramów sekwencji. Znajdując przypadki użycia należy zwrócić uwagę na fakt, że sens istnienia przypadku użycia leży w istnieniu aktora, który go potrzebuje. Przypadki użycia realizują funkcje oferowane przez system dla danego aktora. W tab. 2 przedstawiono niektóre PU dla projektowanego systemu.

Tabela 2. Wybrane przypadki użycia systemu nadzorowania transmisji

Nr PU	Nazwa	Opis
2.	Identyfikacja szczegółowa	Wybranie przez użytkownika wcześniej zidentyfikowanego sterownika, którego dane znajdują się w oknie komunikacji inicjuje odczyt poszczególnych grupy rejestrów
3.	Cykliczny odczyt rejestrów	Przypadek użycia zaczyna się w momencie zakończenia PU2 „Identyfikacja szczegółowa”. Polega on na cyklicznym odczycie poszczególnych grup rejestrów. Jeżeli grupa nie zostanie prawidłowo odczytana, użytkownik dostaje informacje o zerwaniu łączności ze sterownikiem.
12.	Ustaw parametry portu	Ta funkcja systemu wywołwana jest z okna komunikacji dla danego portu. W oknie ustawień parametrów transmisji użytkownik może dokonać zmian parametrów, a także włączyć lub wyłączyć dany port.

Identyfikacja klas i obiektów pozwala wyróżnić trzy ich rodzaje:

1. Klasy/Obiekty businessowe (inaczej aplikacji) – części systemu prezentujące elementy bierne lub trwałe dane istniejące również po zakończeniu trwania przypadku użycia (produkty, dokumenty); tabela 3.
2. Klasy/Obiekty interfejsu – części systemu (osoby lub urządzenia w obrębie firmy), z którymi komunikuje się aktor w trakcie realizacji przypadku użycia; tabela 4,
3. Klasy/Obiekty sterujące – aktywne elementy systemu nie komunikujące się bezpośrednio z aktorami, ale wykonujące ważne czynności, niezbędne do przeprowadzenia przypadku użycia; tabela 5.

Tabela 3. Obiekty aplikacji systemu nadzorowania

Nazwa obiektu	Typ	Opis
Sterownik	TSterownik	Urządzenie, z którym komunikuje się system nadrzędny
Obiekt COM	COM	Jest to obiekt, który zawiera parametry dla poszczególnych portów komunikacyjnych
Obiekt C_Opoz	C_Opoz	Jest to obiekt zawierający dane dotyczące opóźnień dla poszczególnych typów ramek
Model	TModel	Obiekt reprezentujący dane z grupy rejestrów Model
Nastawy/ Pomiar/ Liczniki/ Zdarzenia	Tnastawa/Tpomiar/ Tlicznik/TZdarzenie	Obiekt reprezentujący dane z grupy rejestrów Nastawy/ Pomiar/ Liczniki/ Zdarzenia

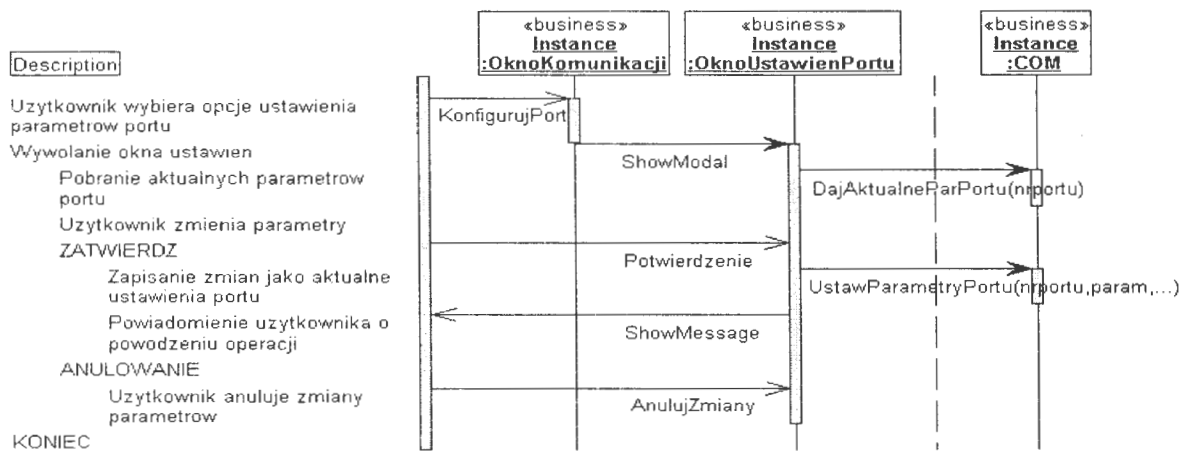
Tabela 4. Obiekty interfejsu systemu nadzorowania

Nazwa obiektu	Typ	Opis
Okno główne	TOknoGlowne	Okno aplikacji zawierające menu główne, z którego użytkownik wywołuje poszczególne funkcje systemu
Okno komunikacji	TOknoKomunikacji	Okno komunikacji, w którym użytkownik może wywoływać funkcje nawiązywania łączności ze sterownikiem
Okno ustawień portu	TOknoUstawienPortu	Okno, w którym użytkownik może zmieniać parametry transmisji dla danego portu
Okno grupy Zdarzenia	TOknoGrupyZdarzenia	Okno służące do zarządzania grupą rejestrów Zdarzenia
Okno grupy Liczniki	TOknoGrupyLiczniki	Okno służące do zarządzania grupą rejestrów Liczniki

Tabela 5. Obiekty sterujące systemem nadzorowania

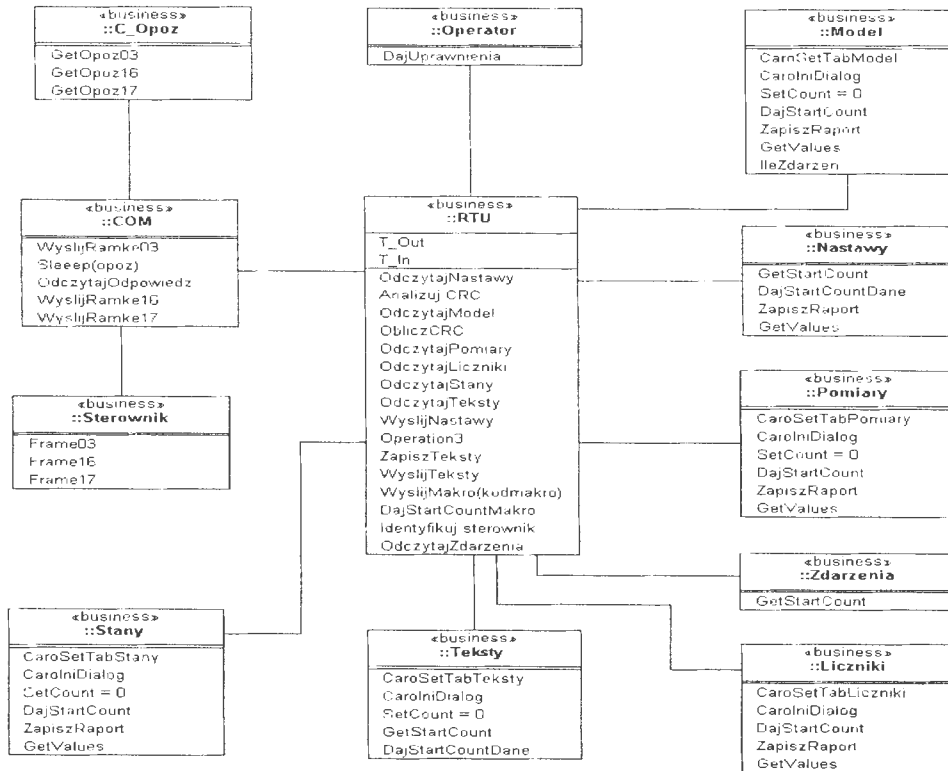
Nazwa obiektu	Typ	Opis
Zarządca aplikacji	TApplication	Obiekt sterujący całą aplikacją, (przetwarzanie komunikatów, klawiszy skrótów, zarządzanie oknami, systemem pomocy, drukowaniem itd.)
Obiekt RTU	RTU	Obiekt odpowiadający za sterowanie komunikacją ze sterownikiem

Dla każdego przypadku użycia formułuje się diagram sekwencji; dla PU nr 12 diagram sekwencji pokazano na rys.1, natomiast diagram klas całego systemu przedstawia rys. 2.



Rys.1. Diagram sekwencji przedstawiający przypadek użycia „Ustawienie parametrów portu”

Diagram obiektów sterujących i aplikacji



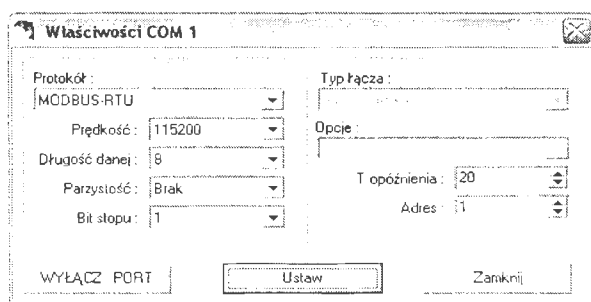
Rys.2. Diagram obiektów sterujących i aplikacji

5. CHARAKTERYSTYKA METODYKI SELECT PERSPECTIVE

Diagramy na rysunkach 1 i 2 są wykonane z zastosowaniem UMLowego narzędzia SELECT Perspective, które obejmują pełny proces projektowania SI, począwszy od analizy wymagań, poprzez projektowanie i implementację aż do końcowego produktu (sprawnie działający system).

Metodyka SELECT charakteryzuje się tym, że system jest budowany fragmentami nazywanymi przyrostami, a więc jest to przyrostowy sposób budowy systemu.

Oferuje ona bogatą notację, służącą opisowi klas i obiektów. Poza opisem samej klasy, jej atrybutów i metod notacja przewiduje wyróżnienie metod i klas abstrakcyjnych oraz ustalenia praw dostępu do atrybutów i metod. Model obiektów służy do przedstawienia budowy (architektury) systemu informatycznego.



Rys.3. Okno ustawień parametrów transmisji dla portu (interfejs przypadku użycia PU12).

6. UWAGI KOŃCOWE I WNIOSKI

Modelowany system zaimplementowano w języku Borland C++ Builder, który jest wizualnym i obiektywnym środowiskiem programistycznym. C++ Builder łączy w sobie charakterystyczne cechy środowiska RAD z funkcjonalnością ANSI C++. Przykładowe okno interfejsu dla PU12 pokazano na rys.3.

Modelowany system, pierwotnie przeznaczony do sterowników MUPASZ [3], jest systemem otwartym i może się komunikować z innymi urządzeniami, obsługującymi protokół MODBUS RTU (po odpowiedniej modyfikacji bibliotek *.ben). System spełnia założenia funkcjonalne, które zostały postawione w etapie określania wymagań w postaci diagramu przypadków użycia.

UML DEFINED MODEL OF SUPERVISING DATA TRANSFER OF CONTROLLER

Abstract. In the paper there is presented an approach of applying of an UML methodology for analysis, design and implementation of real-time system, a case study of which is the supervisory system of data exchange of a controller. The model of the system, defined as a UML design, consists of sequel of diagrams reflecting different of the data processing perspectives. Depending on requirements the model contains a dedicated group of diagrams and other tools. The basic of them are: diagram of use cases, diagram of classes and objects, diagram of interaction. In the paper have been described representative diagrams for data transfer based on Master-Slave mode.

Literatura

1. Dumnicki R., Kasprzyk A., Kozłowski M. (1998) *Analiza i projektowanie obiektowe*. Helion, Gliwice.
2. Mielczarek W. (1993) *„Szeregowe interfejsy cyfrowe*. Helion, Gliwice.
3. Instytut Tele- i Radiotechniczny - *„Mikroprocesorowe urządzenia do pomiarów automatyki, sterowania i zabezpieczeń”*, ITR, 2003.



Instytut Badań Systemowych
Polskiej Akademii Nauk

ISBN 83-89475-01-4