
WYBRANE PROBLEMY INŻYNIERSKIE

NUMER 2

INSTYTUT AUTOMATYZACJI PROCESÓW TECHNOLOGICZNYCH
I ZINTEGROWANYCH SYSTEMÓW WYTWARZANIA

Jerzy ŚWIDER, Mariusz HETMAŃCZYK*

Instytut Automatykacji Procesów Technologicznych i Zintegrowanych Systemów
Wytwarzania, Wydział Mechaniczny Technologiczny, Politechnika Śląska, Gliwice

*mariusz.hetmanczyk@polsl.pl

ZASTOSOWANIE JĘZYKA UML DO MODELOWANIA ZŁOŻONYCH SYSTEMÓW KOMPUTEROWYCH

Streszczenie: Artykuł przedstawia wykorzystanie języka UML (Unified Modeling Language) do celów opracowania diagramów klas [4, 5] pozwalających na projektowanie złożonych struktur systemów komputerowych. Ujęcie modelowania oparto na przykładzie komputerowo zintegrowanego systemu sterowania i diagnostyki napędów rozproszonych, złożonego z niezależnych modułów funkcjonalnych. Artykuł stanowi kontynuację wcześniejszych publikacji autorów [1, 2, 3], a w jego treści ujęto jedynie część elementów opracowanego systemu ograniczoną do oprogramowania komputerowego zainstalowanego w zasobach węzła diagnostycznego. Elementy współdziałające z systemem komputerowym KSSiDNR (sterownik logiczny PLC, przetwornice częstotliwości oraz motoreduktory) zostały pominięte lub opisane w sposób pobieżny, umożliwiając określenie wzajemnych zależności funkcjonalnych oraz powiązań informatycznych.

1. Wstęp

Komputerowe wspomaganie monitorowania, diagnozy obiektów lub procesów przemysłowych znajduje coraz szersze zastosowanie w wielu aplikacjach wspomagających inżynierów utrzymania ruchu. Korzystanie z systemów SCADA lub HMI jest zjawiskiem powszechnym, jednak można zaobserwować tendencję rezygnowania ze środowisk uniwersalnych na rzecz aplikacji (opartych na językach wysokiego rzędu) dopasowanych do konkretnych wymagań użytkowników. Powodem wzrostu zainteresowania systemami dedykowanymi są m.in. dostępność wielu odmian języków programowania (w postaci uniwersalnych lub specjalizowanych platform), możliwość dowolnego kształtowania struktury zależności programowych, rozbudowy, itp.

W przypadku projektowania złożonych struktur komputerowych systemów składających się z wielu klas, powiązanych siecią wzajemnych zależności, przydatność wykazuje język UML [4].

Artykuł przedstawia przykład wykorzystania języka UML do celów opisu struktury komputerowego systemu sterowania i diagnostyki napędów rozproszonych (KSSiDNR) [1, 2, 3], stanowiącego rozbudowane narzędzie ukierunkowane na wspomaganie nadzoru jednostek napędowych.

2. Diagram klas złożonego systemu komputerowego

System KSSiDNR został opracowany jako środowisko dedykowane do celów monitorowania, diagnozy oraz sterowania systemem napędów rozproszonych firmy SEW Eurodrive, połączonych magistralą sieci ProfiBus DP oraz scentralizowanym sterowaniem, w postaci modułowego sterownika logicznego PLC. W środowisku komputerowym zaimplementowano kilka niezależnych modułów programowych, o różnym przeznaczeniu, stanowiących jedno spójne środowisko o wzajemnie uzupełniających się funkcjach.

Na rys. 1 przedstawiono widok diagramu klas systemu KSSiDNR. Na schemacie ujęto również uzupełnienie systemu KSSiDNR o zasoby sterownika logicznego PLC, bez wnikania w szczegółowe funkcje realizowane przez bloki diagnostyczno-sterujące. Klasy oprogramowania węzła diagnostycznego oraz sterownika logicznego PLC potraktowano jako system zależny, w którym poprawne przetwarzanie informacji wymaga współpracy klas składowych oraz wymiany danych zgodnej z jednym ze standardów komunikacji.

W celu zwiększenia przejrzystości część elementów systemu pogrupowano w pakiety. Dodatkowo moduły diagnozy off-line oraz weryfikacji wstępnej ujęto w postaci odrębnych diagramów (diagram klas systemu KSSiDNR, rys. 1, zawiera jedynie pakiet modułu wraz z zależnościami, w odniesieniu do innych klas oraz interfejsów opisywanego systemu).

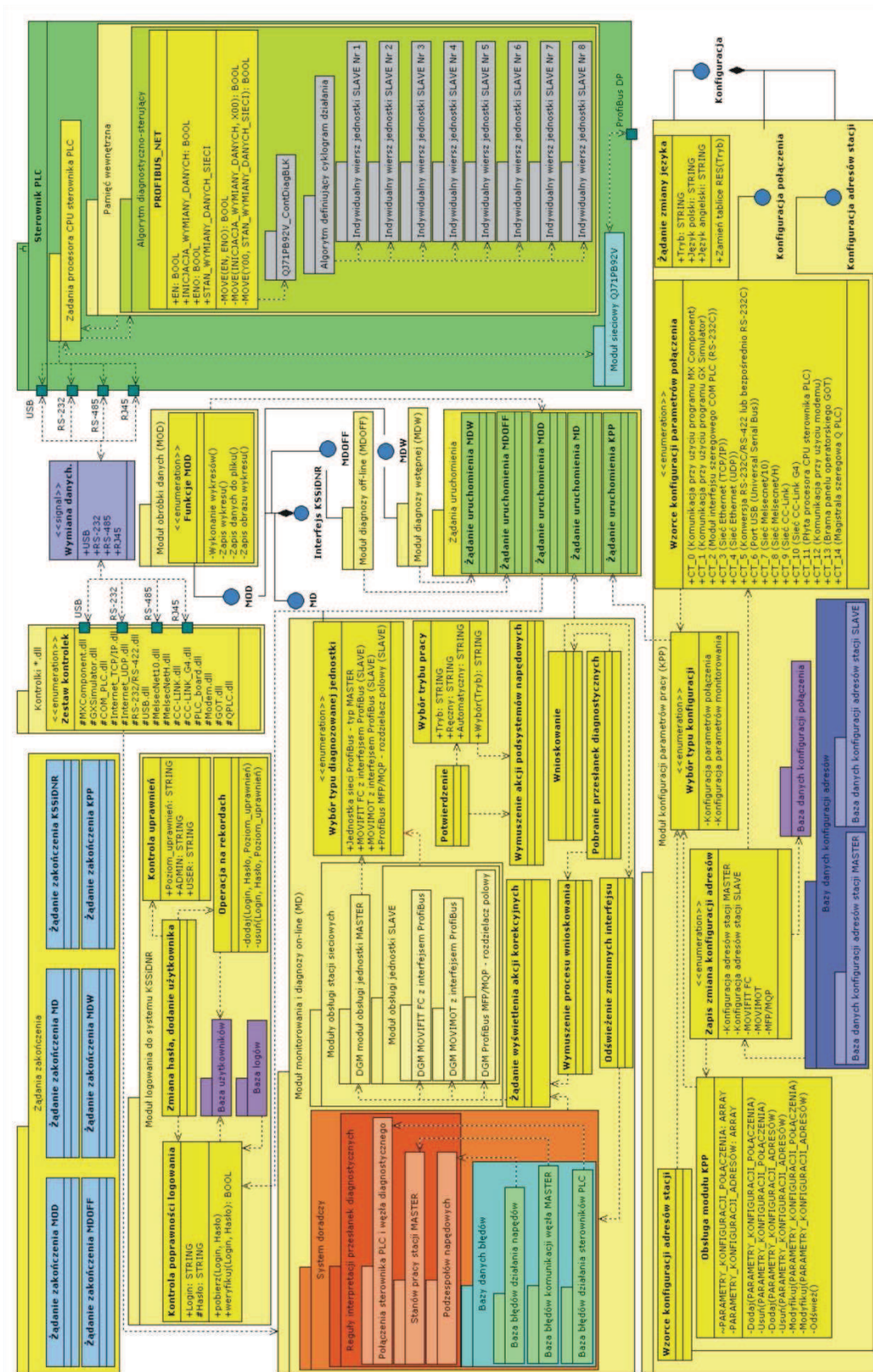
Zaimplementowany system doradczy (ulożony w zasobach komputera PC, pełniący rolę węzła diagnostycznego) jest używany w narzędziach modułu diagnozy off-line, on-line lub modułu obróbki i interpretacji danych historycznych (zapisanych w tablicach danych historycznych).

W przypadku monitorowania wartości bieżących nadzorowanego systemu (stan podglądu lub wymuszania stanów, w trybie on-line) nieodzowne jest wykorzystanie informacji pozyskiwanych przez zintegrowane bloki diagnostyczno-sterujące, zaimplementowane w pamięci sterownika logicznego PLC. Bloki diagnostyczno-sterujące realizują funkcję sterowania zgodnie z algorytmem definiującym cyklogram działania (niezależnie od systemu doradczego i funkcji węzła diagnostycznego).

Zdefiniowana zależność klas polega na przekazywaniu znaczników pozwalających na inicjację aktywności kolejnej klasy lub przekazaniu wartości liczbowych parametrów, pozwalających na wykonanie zdefiniowanych operacji.

Interfejs użytkownika systemu KSSiDNR stanowi element, w którego obrębie wyświetlane są interfejsy pozostałych modułów (zależność pomiędzy interfejsami dodatkowymi, a interfejsem głównym KSSiDNR określono w postaci relacji kompozycji [4, 5]).

Diagram klas modułu diagnozy off-line przedstawiono na rys. 2. Moduł diagnozy off-line [3] wywoływany jest z poziomu menu okna głównego, podobnie jak inne dostępne komponenty systemu KSSiDNR. Interfejs użytkownika modułu diagnozy off-line stanowi integralną część interfejsu systemu KSSiDNR (okno interfejsu głównego systemu KSSiDNR posiada status rodzica w stosunku do okien innych interfejsów).

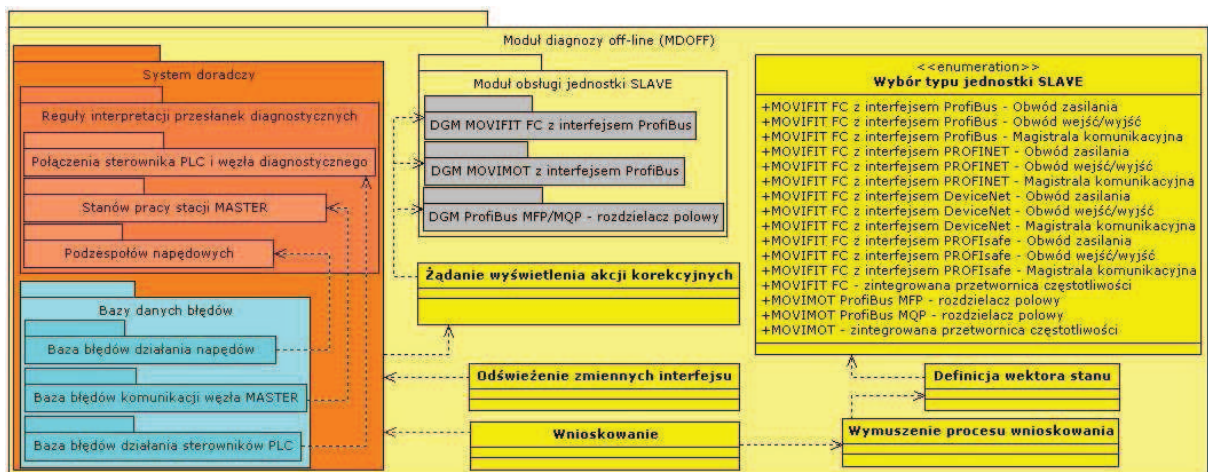


Rys. 1. Widok diagramu klas systemu KSSiDNR
Fig. 1. View of the class diagram of the KSSiDNR system

Czynności realizowane w module diagnostyki off-line, pozwalające na identyfikację błędu, można przedstawić w następujących krokach (rys. 2):

- wybór diagnozowanego podsystemu,
- odwzorowanie stanów urządzenia (definicja postaci wiersza binarnej macierzy diagnostycznej) - każdemu uszkodzeniu przyporządkowano jednoznacznie zdefiniowany zbiór binarnych sygnałów diagnostycznych, stanowiący wzorzec porównawczy funkcji definiujących błąd,
- wykonanie żądania identyfikacji i interpretacji błędów,
- wybór opcji powodującej wygenerowanie akcji korekcyjnych (tj. zbioru czynności pozwalających na identyfikację źródła nieprawidłowości).

Zaprojektowany interfejs użytkownika umożliwia [3] zdefiniowanie wektora stanów, bazującego na wewnętrznym algorytmie diagnozowanych modułów. Dzięki takiemu rozwiązaniu jest możliwe określenie przyczyny błędu bez konieczności bezpośredniego połączenia z diagnozowaną jednostką (nie jest wymagane pobranie pakietu danych z zasobów sterownika logicznego PLC).



Rys.2. Widok diagramu klas modułu diagnostyki off-line
Fig.2. View of the class diagram of the off-line diagnosis module

Diagram klas modułu diagnostyki wstępnej przedstawiono na rys. 3. Inicjacja modułu następuje w oknie głównym w wyniku wyboru odpowiedniej opcji menu (wybór żądania uruchomienia, MDW, rys. 1).

Klasa zawarta w ramach modułu korekcji napędu przejmuje część wartości liczbowych z klasy weryfikacji napędu (dane dotyczące wartości liczbowych parametrów elementów konstrukcyjnych korby, korbowodu, przenośnika rolkowego, itp.), natomiast użytkownik wprowadza wartości liczbowe parametrów elementu napędu będącego przedmiotem korekcji lub weryfikacji (np. zależności momentu znamionowego silnika oraz jego mocy, w odniesieniu do wartości rzeczywistej obciążenia zewnętrznego).

Wynikiem obliczeń są skorygowane wartości parametrów napędu oraz informacja o stanie doboru (dobór poprawny lub niepoprawny).



Rys.3. Widok diagramu klas modułu diagnozy wstępnej
 Fig.3. View of the class diagram of the preliminary diagnosis module

Wzajemne powiązanie funkcji zaimplementowanych w poszczególnych klasach pozwala na zbudowanie kompletnego systemu umożliwiającego obsługę napędów rozproszonych.

3. Podsumowanie

W artykule przedstawiono wykorzystanie diagramów klas zapisanych zgodnie z notacją języka UML. Należy nadmienić, że opisywany język wyposażono w szereg innych możliwości reprezentacji struktur systemowych (diagramy obiektów, przypadków użycia, stanów, przebiegu, czynności, itp.), jednak wybrana przez autorów reprezentacja została uznana za wystarczającą do potrzeb opracowania systemu KSSiDNR. Zastosowanie języka UML w modelowaniu złożonych struktur programów niesie z sobą wiele niezaprzeczalnych korzyści, w postaci możliwości:

- przejrzystego zapisu wzajemnych zależności pomiędzy poszczególnymi klasami,
- zmian, na etapie projektowania, struktury oprogramowania,
- określenia liczby zmiennych, ich typu oraz funkcji przetwarzających pozwalających zrealizować zamierzony algorytm pracy,
- sprawdzenia poprawności toku myślowego projektanta systemu oraz wstępnej weryfikacji modelu, bez nadmiernego nakładu pracy.

Literatura

1. Świder J., Hetmańczyk M.: Method of indirect states monitoring of dispersed electric drives. BINDT, 2009, p. 1171÷1179.
2. Świder J., Hetmańczyk M.: Adaptation of the expert system in diagnosis of the connection of the PLC user interface system and field level. "Solid State Phenomena" 2010, No. 164, s. 201÷206
3. Świder J., Hetmańczyk M.: Komputerowe wspomaganie diagnozy stanów przetwornic częstotliwości. „Problemy Maszyn Roboczych” 2009, nr 33, s. 57÷66
4. O’Docherty M.: Object-oriented analysis and design understanding system development with UML 2.0. John Wiley & Sons Ltd, 2005.
5. Schmuller J.: UML dla każdego. Gliwice: Helion, 2001.

APPLICATION OF UML LANGUAGE FOR MODELLING OF COMPLEX COMPUTER SYSTEMS

Summary: The article presents an application of the UML language (Unified Modeling Language) for an elaboration of class diagrams which allow to design of complex structures of computer systems. An approach based on an example of the Computer Integrated Control and Diagnostic System of Distributed Drives is composed from independent functional modules. The article is continuation of earlier publications of the authors and its content contains only a part of elements of the elaborated system limited to the computer software installed in the resources of the diagnostic node. Elements which co-operate with KSSiDNR system (for example a Programmable Logic Controller, frequency inverters and gear-motors) were omitted or described in a superficial way, allow determining of reciprocal functional relationships or informatics’ dependences.