

Przedmowa

TOMASZ SZMUC

Wykłady zaproszone konferencji *Systemy Czasu Rzeczywistego* mają tradycyjnie charakter szkoleniowy, co często decyduje o doborze tematyki. Dokładniej, wybór tematyki jest określony przez trzy, niżej wymienione podstawowe kryteria:

1. nowość w sensie koncepcji, podejścia do rozwiązania w zagadnieniach związanych z tematyką Konferencji;
2. aktualność języków, metod, metodyk i środowisk wspomagających, szczególnie w kontekście ich aktualnego rozwoju i zastosowań;
3. istotność zagadnienia przy nieco mniejszej popularności (w warunkach polskich) w środowiskach wytwarzających oprogramowanie.

Dwa pierwsze kryteria dotyczą nowych i aktualnych elementów związanych z wytwarzaniem oprogramowania czasu rzeczywistego, natomiast trzecie dopuszcza elementy niekoniecznie najnowsze, lecz mniej ugruntowane w polskiej rzeczywistości i jednocześnie bardzo ważne i stosowane w firmach o zasięgu światowym. Podane uszeregowanie kryteriów nie wynika z próby oceny ich ważności, lecz raczej odpowiada przejściu od elementów ogólniejszych do szczegółowych (1 \Rightarrow 2) oraz osłabieniu wymagania nowości (2 \Rightarrow 3).

Wykłady szkoleniowe (zaproszone) na VII Konferencję *Systemy Czasu Rzeczywistego* (Kraków, 25–28 września 2000) zostały wybrane zgodnie z powyższymi kryteriami:

1. *Zastosowanie UML do projektowania systemów czasu rzeczywistego – metodyka ROPES* – Prof. dr hab. inż. Zbigniew Huzar – Politechnika Wrocławska;
2. *Zastosowanie logiki temporalnej w specyfikacji i weryfikacji oprogramowania – w stronę czasu rzeczywistego* – Dr inż. Radosław Klimek – Akademia Górniczo-Hutnicza;
3. *Planowanie projektu informatycznego* – Dr inż. Stanisław Szejko – Politechnika Gdańska.

Pierwszy z wykładów dotyczy bardzo popularnego już języka modelowania stosowanego w początkowych fazach (analiza, projektowanie) wytwarzania oprogramowania, drugi prezentuje możliwości zastosowania logik temporalnych do wspomagania analizy poprawności oprogramowania, natomiast trzeci opisuje bardzo ważne zagadnienia planowania projektu informatycznego. W myśl podanych wyżej kryteriów wykład pierwszy spełnia drugie z wymienionych, drugi dotyczy raczej pewnej filozofii opisu i koncepcji (pierwsze kryterium), natomiast trzeci wykład został wybrany na podstawie trzeciego kryterium.

Zastosowanie języka UML (ang. *Unified Modeling Language*) zyskuje coraz większą popularność nie tylko przy wytwarzaniu klasycznych systemów informatycznych, lecz również różnych typów systemów specjalizowanych. Powoduje to z jednej strony modyfikację i opracowywanie nowych wersji tego języka, jak również rozwijanie metodyk i narzędzi wspomagających. W wykładzie: *Zastosowanie UML do projektowania systemów czasu rzeczywistego – metodyka ROPES* wykorzystano elementy najnowszej wersji języka (UML 1.4), ogłoszonej w roku 2000. Należy przy tym podkreślić, że główny nacisk położono na prezentację zastosowania języka w kontekście metodyki ROPES (ang. *Rapid Oriented Process for Embedded Systems*). Taki układ wyróżnia to opracowanie względem innych publikacji, opisujących najczęściej sam język i traktujących raczej pobieżnie (z nielicznymi wyjątkami) jego zastosowanie. Główny wątek stanowi zatem opis poszczególnych faz metodyki ROPES: analiza, projektowanie, implementacja, testowanie, przy czym szczegółowo opisane są dwie początkowe (analiza, projektowanie), gdzie język UML ma bezpośrednie zastosowanie. Wspomniane fazy dzielą się na podfazy, odpowiednio analizy wymagań, analizy systemowej i analizy obiektowej dla analizy oraz projektowania ogólnego, pośredniego i szczegółowego dla fazy projektowania. Poszczególne podfazy są opisywane jednolicie, począwszy od ogólnego wprowadzenia i sformułowania celu danej podfazy, następnie opis wykonywanych czynności, specyfikację elementów konstruowanego modelu oraz artefakty wynikowe danej podfazy. Dodatkowo dla każdej z faz tej metodyki opisano przepływ artefaktów między podfazami w konwencji diagramów przepływu danych. Opisana ogólnie metodyka jest następnie szczegółowo ilustrowana na przykładzie sterownika bramek (rogatek) na przejeździe kolejowym. Przykład ten jest dokładnie opisany z rozbiciem na wspomniane (pod)fazy oraz z uwzględnieniem różnych aspektów, w tym bezpieczeństwa systemu.

W dziedzinie metod formalnych wyróżnia się trzy podstawowe klasy stosowane do modelowania i analizy oprogramowania systemów czasu rzeczywistego: sieci Petriego, algebry procesów (CCS, CSP) i logiki temporalne. Dwa pierwsze mają charakter operacyjny, tzn. opisują system w terminach częściowo uporządkowanych akcji (zmian stanów), natomiast podejście trzecie bazuje na logice, co nadaje mu statyczny charakter. Łatwość interpretacji sieci Petriego decyduje o ich największej popularności, z kolei algebry procesów charakteryzują się wprawdzie

mniejszą czytelnością, lecz z drugiej strony dają możliwość dowodzenia równoważności modeli opisywanych z różną dokładnością (bisymulacja). Podejścia bazujące na logice charakteryzują się elegancją i precyzją zapisu. Własności systemu są specyfikowane w postaci odpowiednich formuł, gdzie operatory logiki klasycznej opisują własności stanów, natomiast operatory temporalne specyfikują zależności wzdłuż historii (następstwa stanów). Wykład: *Zastosowanie logiki temporalnej w specyfikacji i weryfikacji oprogramowania – w stronę czasu rzeczywistego* zawiera krótki i kompetentny przegląd podstawowych klas logik temporalnych oraz możliwości ich zastosowań w specyfikacji i weryfikacji oprogramowania czasu rzeczywistego. W części początkowej przedstawiono analizę różnych reprezentacji czasu (ciągły–dyskretny, ograniczony–nieograniczony, punkt czasowy–przedział), struktur czasowych (liniowa, rozgałęziona, równoległa). W kolejnych dwóch podrozdziałach opisywane są modele dwóch podstawowych klas: logik liniowych i logik rozgałęzionych. Ta bardziej klasyczna część jest przedstawiona w układzie: opis składni, semantyki, ilustracja w postaci przykładów oraz krótka charakterystyka danej klasy. W tym klasycznym ujęciu czas nie jest używany *explicite*, lecz decyduje raczej o uporządkowaniu stanów. Bezpośrednie wprowadzenie aspektu czasowego jest realizowane zgodnie z dwoma podstawowymi modelami znanymi jako semantyka przeplotowa (rzutowanie współbieżnych zdarzeń na oś czasu) i semantyka zegarowa (rejestracja czasu wystąpienia zdarzeń). Uzależnienie od czasu jest również uwidocznione w operatorach temporalnych, przy czym wyróżnia się trzy podstawowe podejścia: operatory ograniczone czasowo (ograniczony zakres działania operatorów temporalnych), zmienne czasowe (bezpośrednio opisujące czas i stosowane do formułowania warunków) i ograniczone kwantyfikowanie („zamrożenie” czasu zmiennej kwantyfikowanej). W końcowej części tego wykładu podano klasyfikację własności w kontekście podziału na bezpieczeństwo i żywotność oraz opisano różne techniki dowodzenia w obrębie dwóch zasadniczych podejść: weryfikacji modelowej i metody aksjomatycznej.

Planowanie projektu informatycznego jest szczególnie istotne w kontekście niemal przysłowiowego niedotrzymywania terminów realizacji i przekraczania założonego budżetu przy wytwarzaniu i wdrażaniu oprogramowania. Dość popularne przekonanie o podstawowej roli (ze względu na powodzenie przedsięwzięcia) organizacji procesu wytwarzania winno być jednak zmodyfikowane przez zwrócenie uwagi na istotność planowania projektów informatycznych i ich realizacji zgodnie z założonymi planami. Trzeci z prezentowanych wykładów: *Planowanie projektu informatycznego* opisuje właśnie te zagadnienia. W początkowej części zdefiniowano projekt informatyczny i opisano kontekst jego realizacji: środowisko, strategie prowadzenia, infrastruktura, czynniki ryzyka i sukcesu, zarządzanie projektem. W kolejnej części przedstawiono cele, zadania i poziomy planowania projektu. Zasadniczą część stanowi jednak metodyka planowania, w której po krótkim omówieniu

aspektów ogólnych: przegląd opisu, dobór strategii, uwzględnienie ryzyka, identyfikacja i wyznaczenie zadań – przeanalizowano szczegółowo metody szacowania zadań (pracochłonność, czas trwania projektu, obciążenie ludzi) oraz podano techniki tworzenia harmonogramów projektów (uszeregowanie zadań, określenie zasobów dla zadań, identyfikacja ścieżki krytycznej). Wykład stanowi spójną i zwartą całość prezentującą podstawowe i istotne elementy i metody związane z planowaniem projektu informatycznego.

Opisywane powyżej trzy wykłady zaproszone dotyczą różnych aspektów i obszarów wytwarzania oprogramowania. Niejednokrotnie dość duża odległość tematyczna wykładów wynika z założonego celu – prezentacji wybranych koncepcji, narzędzi, metod ostatnio rozwijanych i stosowanych w szeroko rozumianym wytwarzaniu oprogramowania dla systemów czasu rzeczywistego. W celu pewnego złagodzenia tej różnorodności w niniejszym wstępie przedstawiono nieco szerszy kontekst wykładów, a krótki opis ich zawartości ułatwi wybór i być może zachęci czytelnika do sięgnięcia po niniejsze opracowanie.

Zawartość poszczególnych wykładów została opracowana po wygłoszeniu na VII Konferencji *Systemy Czasu Rzeczywistego*. Wymagało to dodatkowego nakładu pracy ze strony wykładowców i umożliwiło uwzględnienie doświadczeń i niejednokrotnie rozszerzenie zakresu w stosunku do prezentowanego na Konferencji. Pragnę w tym miejscu gorąco podziękować Autorom poszczególnych rozdziałów za podjęcie się tego zadania, dodatkowe opracowanie wykładów i tym samym umożliwienie ich opublikowania.