

Przedmowa do wydania trzeciego¹

Od najdawniejszych czasów najbardziej utalentowanym architektom i najwybitniejszym projektantom było znane prawo oszczędności (zwane też brzytwą Okhama). Niezależnie od tego, czy wyraża się je przez paradoks („mniej oznacza więcej”), czy w postaci buddyjskiego koanu („umysł zen to umysł początkującego”), jego mądrość jest ponadczasowa — zredukować każdą rzecz do jej istoty, tak aby forma harmonizowała z jej przeznaczeniem. Od piramid egipskich do budynku Opery w Sydney, od architektur von Neumanna do Uniksa i Smalltalka, najlepsi architekci i projektanci dążyli do przestrzegania tej uniwersalnej i wiecznej zasady.

Znam zalety cięcia brzytwą Okhama, więc gdy projektuję i czytam, wówczas zawsze szukam projektów i książek zgodnych z prawem oszczędności. Z tego względu muszę też pochwalić książkę, którą właśnie macie w rękach.

Moja ostatnia uwaga może się z początku wydawać zaskakująca. Często kojarzy się mnie z obszernymi i bogatymi specyfikacjami, które definiują zunifikowany język modelowania (UML). Specyfikacje te pozwalają twórcom narzędzi na zaimplementowanie UML-a, a metodologom na stosowanie go. Przez siedem lat przewodziłem wielkim międzynarodowym zespołom standaryzacyjnym, które określiły UML 1.1 i UML 2.0 i przygotowały w międzyczasie kilka pośrednich wydań tego języka. W tym czasie wzrosły możliwości UML-a do wyrażania pojęć oraz jego precyzja, ale też wzrosła jego złożoność wynikająca z procesu standaryzacji. Niestety procesy standaryzacyjne są raczej znane z kompromisów koniecznych przy projektowaniu przez komitety, a nie z eleganckiej oszczędności.

Czego ekspert UML-a znający najgłębsze tajniki specyfikacji mógłby nauczyć się z przewodnika po języku UML 2.0 proponowanego przez Martina? Okazuje się, że całkiem sporo. Po pierwsze Martin sprytnie zredukował ogromny i złożony język do pragmatycznego podzbioru, którego efektywności dowiódł w swojej własnej praktyce. Nie poszedł przy tym na łatwiznę i nie dodał po prostu kilku następnych stron do poprzedniego wydania swojej książki. Mimo że język UML się rozrósł, Martin pozostał wierny swojemu celowi poszukiwania „najbardziej przydatnej części UML-a” i o niej właśnie pisze. Ta część to mityczne 20% UML-a, która pozwala na wykonanie 80% pracy, i znalezienie tej części jest nie lada osiągnięciem!

Jeszcze większe wrażenie robi to, że Martin osiągnął ten cel, nie rezygnując z pisania ujmującym, gawędziarskim stylem. Dzieląc się z nami swoimi opiniami i anegdotami, sprawił, że książkę czyta się przyjemnie. Autor przypomina nam, że projektowanie i przygotowywanie systemów powinno być zarówno twórcze, jak i produktywne. Jeśli w pełni podążymy za regułą oszczędności, to odkrywamy, że modelowanie w UML-u może być tak przyjemne, jak zajęcia z plastyki w szkole podstawowej. UML powinien być piorunochronem gromadzącym naszą kreatywność i laserem pozwalającym na precyzyjne formowanie projektów, tak aby inni mogli na ich podstawie budować rzeczywiste systemy. Ten ostatni cel jest próbą ogniową dla każdego prawdziwego języka projektowania.

¹ To wydanie jest wydaniem trzecim oryginalnym, a drugim polskim. (przyp. wyd.)

Zatem, mimo że jest to mała książka, nie jest ona trywialna. Z podejścia Martina do modelowania i jego opisów języka UML 2.0 można się bardzo wiele nauczyć.

Ogromną przyjemność sprawiła mi praca z Martinem nad wybieraniem funkcji języka UML 2.0 do tego wydania książki i poprawianiem przykładów. Pamiętajmy o tym, że wszystkie żywe języki, zarówno naturalne, jak i sztuczne, muszą ewoluować lub zanikają. Dokonywane przez Martina wybory nowych funkcji oraz preferencje użytkowników są najważniejsze w procesie korygowania języka UML. Utrzymują one vitalność języka i pomagają mu ewoluować na zasadach doboru naturalnego na rynku.

Zanim tworzenie oprogramowania za pomocą modeli stanie się dominującym nurtem minie jeszcze wiele czasu, ale jestem bardzo entuzjastycznie nastawiony do takich książek, jak niniejsza — takich, które jasno prezentują podstawy modelowania w UML-u i stosują je pragmatycznie. Mam nadzieję, że skorzystacie z niej co najmniej w takim stopniu, jak ja, a pomysły, na które przy okazji wpadniecie, wykorzystacie do udoskonalenia swoich własnych metod modelowania oprogramowania.

Cris Kobryn
Prezes UML 2.0 Submission Team w U2 Partners
Główny specjalista, Telelogic

Przedmowa do poprzednich wydań

Gdy rozpoczynaliśmy prace nad zunifikowanym językiem modelowania — w skrócie UML (ang. Unified Modeling Language), mieliśmy nadzieję, że uda nam się stworzyć standardowe narzędzie do wyrażania projektów oprogramowania, które nie tylko by odzwierciedlało najlepsze osiągnięcia informatyki, ale też „odczarowało” proces modelowania systemów informatycznych. Wydawało się nam, że dostępność standardowego języka modelowania zachęciłaby informatyków do modelowania systemów przed ich kodowaniem. Szybka i szeroka akceptacja UML-a pokazuje, że korzyści z modelowania są w rzeczy samej znane informatykom.

Tworzenie UML-a było procesem iteracyjnym i przyrostowym podobnym do modelowania dużego systemu informatycznego. W efekcie otrzymaliśmy standard odzwierciedlający pomysły wielu osób oraz wkład wielu osób i firm. My rozpoczęliśmy prace nad UML-em, ale inni pomogli je pomyślnie zakończyć; jesteśmy wdzięczni za ich wkład.

Opracowanie i uzgodnienie standardowego języka modelowania jest samo w sobie znaczącym wyzwaniem. Równie znaczącym wyzwaniem jest upowszechnienie wiedzy o nim w środowisku informatyków w sposób, który jest zarówno przystępny, jak i osadzony w kontekście procesu tworzenia oprogramowania. Udało się to Martinowi Fowlerowi w tej niedużej książce, zaktualizowanej po to, aby uwzględnić najnowsze zmiany w UML-u.

Martin jasno i bez przybierania tonu mentora omawia najważniejsze elementy UML-a, a także wyraźnie pokazuje, jaką rolę odgrywa język modelowania w procesie tworzenia oprogramowania. Po drodze daje nam nieocenione wskazówki do-

tyczące modelowania, oparte na jego ponaddwunastoletnim doświadczeniu w projektowaniu i modelowaniu systemów informatycznych.

Wynikiem jest książka, która wprowadziła tysiące informatyków w podstawy UML-a, rozbudzając ich ciekawość i chęć poznania korzyści płynących z modelowania w tym standardowym obecnie języku.

Polecamy tę książkę wszystkim informatykom, którzy są zainteresowani wprowadzeniem do UML-a i poznaniem ważnej roli, jaką odgrywa on w projektowaniu oprogramowania.

Grady Booch
Ivar Jacobson
James Rumbaugh

Specjalnie dla Czytelników pierwszego wydania polskiego

Zawsze miło usłyszeć o kolejnym tłumaczeniu swojej książki — tak też mi się zrobiło, gdy usłyszałem o tym. W Europie Środkowej jest mnóstwo utalentowanych programistów, a ja gorąco wierzę, że to właśnie oni przyczyniają się do zasympyowania przepaści dzielącej Wschód od Zachodu (gdzie ja się wychowywałem). Wprawdzie nie wydaje mi się, aby UML rozwiązał wszystkie problemy, jednakże standard języka graficznego służy usprawnieniu porozumiewania się zespołów programistów. Uważam również, że to tłumaczenie będzie choćby niewielkim kroczkiem usprawniającym porozumiewanie się między narodami. Mam też nadzieję, że kiedyś nadarzy się okazja do odwiedzenia Polski i dowiedzenia się o niej czegoś więcej.

Martin

O Autorze

Martin Fowler jest pionierem zastosowań technik obiektowych w systemach komputerowych przeznaczonych dla przedsiębiorstw. W firmie ThoughtWorks, zajmującej się opracowywaniem i wdrażaniem rozwiązań wielkoskalowych, jest głównym specjalistą. Jako autor książek i opracowań na temat wzorców, języka UML, refaktoryzacji oraz metod lekkich, w przystępny i zajmujący sposób dzieli się swoim bogatym, praktycznym doświadczeniem. Mieszka w Massachusetts z żoną Cindy oraz z przedziwnym kotem.