III Krajowa Konferencja Naukowo-Techniczna: 7-10 wrzesień, 1998, Jurata k/Gdańska
Diagnostyka Procesów Przemysłowych

Analiza bezpieczeństwa przemysłowych zastosowań informatyki z wykorzystaniem metody FMEA

Tadeusz CICHOCKI*, Janusz GÓRSKI**

*Adtranz Zwus, ul. Modelarska 12, 40-142 Katowice, e-mail: tadeusz.cichocki@plsig.mail.abb.com
**Politechnika Gdańska, Katedra Zastosowań Informatyki,
ul. Narutowicza 11/12, 80-952 Gdańsk, e-mail: jango@pg.gda.pl

Streszczenie. W artykule przedstawiono wykorzystanie analizy FMEA w ramach systemu zapewniania bezpieczeństwa aplikacji informatycznych. Wskazano na problemy związane ze stosowaniem tej metody w sposób gwarantujący wysoką wiarygodność wyników. Prezentacji dokonano na bazie doświadczeń związanych z zastosowaniem metody w Adtranz Zwus.

1. WSTĘP

Oprogramowanie i sprzęt komputerowy dynamicznie zwiększają swój udział w przemysłowych systemach sterujących. Umożliwiają one łatwiejszą realizację funkcji sterujących i monitorujących, zwiększenie elastyczności w dopasowywaniu do nowych potrzeb, obniżenie kosztów wytwarzania i eksploatacji, wzrost wydajności systemów. Rozwojowi temu towarzyszy wzrost znaczenia problemu zapewnienia bezpieczeństwa systemów. Dla przykładu, według ocen NASA [10], utrzymanie obecnej intensywności wypadków lotniczych wynikających z błędów systemów sterowania ruchem doprowadzi, wobec wzrostu natężenia ruchu lotniczego, do statystycznie jednego poważnego wypadku lotniczego na tydzień, już w roku 2005.

Analiza bezpieczeństwa (ang. *safety analysis*) koncentruje się na potencjalnych negatywnych skutkach działania systemu dla jego środowiska i obejmuje:

- analizę zagrożeń związanych z systemem,
- analizę ryzyka wynikającego z tych zagrożeń,
- identyfikację działań, które należy podjąć w celu eliminacji lub kontroli zagrożeń oraz redukcji ryzyka.

W efekcie analizy bezpieczeństwa

- definiowane są założenia i cele bezpieczeństwa,
- uzasadniane są przyjęte rozwiązania służące osiąganiu bezpieczeństwa (strategie wykrywania i łagodzenia skutków zdarzeń prowadzących do zagrożeń).

Bezpieczeństwo jest w znacznym stopniu uzależnione od współpracy elementów z różnych dziedzin: komputerowych systemów sterujących, urządzeń mechanicznych i elektrycznych, obsługi operatorskiej i konserwacji. W związku z tym, niezbędnym jest oparcie analizy bezpieczeństwa na modelu o poziomie abstrakcji dostatecznie wysokim, aby adekwatnie obejmował wszystkie te dziedziny. Ze względu na rosnącą złożoność struktur systemów, z udziałem sprzętu, oprogramowania i ludzi, pojawiają się nowe zagrożenia, związane z niepożądanymi charakterystykami struktury systemu i niejawnymi interakcjami pomiędzy jej elementami.

Oprogramowanie jest ze swej natury „nieciągłe": niewielka zmiana kodu lub danych wejściowych może doprowadzić do znacznej różnicy w przebiegu programu i w danych wyjściowych. Zmiany takie mogą mieć wiele przyczyn, takich jak:

- defekty projektowania,
- błędy kompilacji,
- błędy danych wejściowych, lub szerzej, środowiska programu.

Istnieje więc problem niepewności związanej z funkcjonowaniem oprogramowania, a w szczególnym przypadku z programową realizacją podstawowej zasady bezpieczeństwa strukturalnego głoszącej, że

awaria elementu musi prowadzić do przejścia systemu do ustalonego stanu bezpiecznego.

2. ANALIZA FMEA

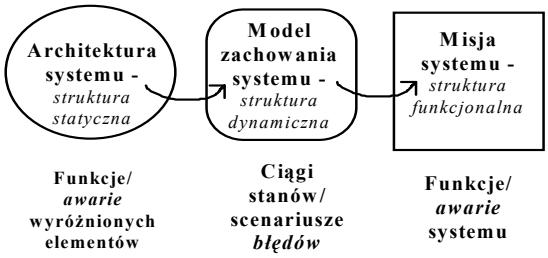
Analiza FMEA (ang. *Failure Mode and Effect Analysis*) opiera się na wnioskowaniu indukcyjnym, które na podstawie niepełnych przesłanek, skończonej liczby obserwacji lub ograniczonych doświadczeń formułuje stwierdzenia ogólne, dotyczące całego systemu. Przedmiotem analizy FMEA jest architektura systemu oraz składające się na nią elementy. Architektura ta jest „próbkowana" na bazie wybranego zestawu potencjalnych awarii elementów (odstępstw od założonego funkcjonowania), w celu wykrycia wynikających stąd awarii całego systemu. Spośród nich wyróżniane są te, które naruszają bezpieczeństwu systemu.

Analiza FMEA rozpoczyna się podjęciem decyzji dotyczącej poziomu elementów bazowych struktury, tzn. tych elementów których wewnętrzna struktura nie jest interesująca z punktu widzenia celów prowadzonej analizy. Potencjalne awarie tych elementów oraz ocena ich skutków są dokumentowane w *tabeli FMEA*. Tabela ta jest strukturą danych, w której gromadzone są wyniki dokonanych analiz. Identyfikowane są scenariusze rozwoju błędów w systemie, zainicjowanych wystąpieniem awarii poszczególnych elementów. Skutki awarii danego elementu są rozpatrywane w odniesieniu do następnych (wyższych) warstw struktury funkcjonalnej, aż do poziomu zewnętrznych interfejsów systemu. Dokumentowane są również powiązania tych scenariuszy z propozycjami działań mających na celu eliminację lub redukcję związanego z nimi ryzyka.

Na Rys.1. podano ogólny schemat obrazujący zakres analizy FMEA.

Istota analizy FMEA polega na weryfikacji architektury systemu poprzez identyfikację powiązań awarii elementów systemu z awariami całego systemu. W tym celu wykorzystywane są scenariusze propagacji błędów w systemie. Skuteczność analizy jest uwarunkowana następującymi czynnikami:

- trafnym wyborem elementów systemu i związanych z nimi typów awarii,
- utworzeniem modelu architektury adekwatnie reprezentującego zależności funkcjonalne pomiędzy poszczególnymi składowymi systemu,
- identyfikacją scenariuszy propagacji błędów inicjowanych przez awarie elementów bazowych.



Rys.1. Zakres analizy FMEA

Analiza FMEA może być podejmowana wcześnie (w ramach cyklu życia systemu), nawet wtedy gdy procesy projektowania i realizacji nie są jeszcze na tyle zaawansowane aby utworzyć kompletny model architektury. W takiej sytuacji analiza jest prowadzona w stosunku do „architektury wyidealizowanej", odzwierciedlającej wstępne intencje i założenia projektantów i ogólne szkice projektowanej struktury, np. wyrażone w postaci diagramów blokowych. Opisy te są na ogół niewystarczające do pełnego zrozumienia natury i konsekwencji awarii elementów niskiego poziomu. Wczesne podjęcie analizy umożliwia decyzje projektowe związane z ujawnionymi zagrożeniami. Wraz z postępem prac analiza jest rozbudowywana w kierunku zwiększenia jej precyzji, np. dla obszarów ocenianych jako najbardziej krytyczne. Na każdym etapie, analiza FMEA dostarcza aktualnych informacji dotyczących potencjalnych zagrożeń wynikających z podjętych decyzji projektowych. Staje się przez to narzędziem, które w sposób iteracyjny wspomaga podejmowanie nowych decyzji, już od najwcześniejszych faz rozwoju systemu.

W całokształcie działań składających się na analizę bezpieczeństwa systemu, analiza FMEA spełnia następującą rolę:

- prowadzi do identyfikacji możliwych, być może nie rozpoznanych wcześniej, stanów systemu,
- umożliwia identyfikację modułów, procedur, procesów lub funkcji, które są krytyczne ze względu na bezpieczeństwo,

- dokonuje przeglądu zdarzeń w systemie na drodze sygnałów od czujników do ich końcowych odbiorców (i w ten sposób prowadzi do lepszego zrozumienia systemu),
- wspomaga formułowanie i weryfikację wymagań dotyczących funkcji związanych z bezpieczeństwem,
- wspomaga projektowanie i sprawdzanie poprawności implementacji funkcji krytycznych ze względu na bezpieczeństwo,
- umożliwia wykazanie niezależności funkcji związanych z bezpieczeństwem od innych funkcji systemu (zależność taka może wystąpić np. poprzez wspólne zasoby),
- wspomaga identyfikację tych obszarów systemu, poprzez które awarie elementów „przedostają się" na poziom funkcji całego systemu,
- wspomaga projektowanie przypadków testowych mających na celu weryfikację scenariuszy błędów prowadzących do istotnych zagrożeń,
- wspomaga formułowanie wniosków diagnostycznych i zaleceń dla bezpiecznej obsługi systemu.

Proces związany z realizacją analizy FMEA odpowiada pierwszym trzem z sześciu funkcji zdefiniowanych w ramach modelu Ciągłego Zarządzania Ryzykiem (ang. *Continuous Risk Management*) przez SEI [8]:

- *Identyfikacja* (poszukiwanie ryzyka zanim przerodzi się ono w rzeczywisty problem),
- *Analiza* (przekształcanie danych dotyczących ryzyka w informację umożliwiającą podejmowanie decyzji),
- *Planowanie* (wykorzystanie tej informacji w podejmowaniu decyzji i działań mających na celu ich odwzorowanie w implementacji systemu).

3. FMEA DLA OPROGRAMOWANIA

Powstaje pytanie czy analiza FMEA może być bezpośrednio zastosowana w odniesieniu do tych części systemu, które są realizowane w postaci oprogramowania. Według normy IEC 812 [9], co potwierdzają również próby podejmowane już od końca lat 70-tych, np. w NASA lub w Boeing Corporation, FMEA może być zastosowana do identyfikacji stanów systemu sprzętowego, które mają wpływ na wymagania względem oprogramowania sterującego tym systemem. Natomiast przeniesienie analizy „w głąb" oprogramowania napotyka na znaczne trudności. Można wymienić trzy podstawowe źródła tych trudności:

- brak powszechnie zaakceptowanego i jednoznacznego modelu architektury oprogramowania,
- niska jakość kodu wynikowego,
- zależność oprogramowania od sprzętu i środowiska wykonania.

Poniżej dokonano bardziej szczegółowej dyskusji tych problemów.

Architektura oprogramowania

Otwarty jest problem identyfikacji elementów programowych niskiego poziomu, które mogą stanowić punkt wyjścia dla analizy FMEA oraz klasyfikacji typów awarii tych elementów. Nie istnieje obecnie teoria wyjaśniająca czym są takie elementy oraz jak je opisywać w celu umożliwienia zrozumienia utworzonego z nich systemu.

Elementy konstrukcyjne współczesnych języków programowania, nazywane modułami, strukturami lub klasami, nie mają jednoznacznej i pełnej semantyki. Stanowią one konstrukcje syntaktyczne i służą przede wszystkim do grupowania związanych ze sobą definicji opisujących organizację danych i związanych z nimi akcji. Opis interakcji między modułami wyrażonymi w klasycznych językach programowania nie jest zwykle jawny i jednoznaczny. Przykładami różnych rodzajów zależności i oddziaływań takich modułów programowych mogą być:

- *zależności proceduralne i funkcjonalne* - procedury i funkcje mogą zawierać odwołania do elementów (globalnych deklaracji typu, globalnych zmiennych i stałych, innych funkcji), które są zdefiniowane poza daną procedurą lub funkcją,
- *zależności typów i elementów danych* - podstawowe typy danych mogą być używane do tworzenia typów bardziej złożonych, stosowanych w różnych modułach; zmienne i stałe mogą reprezentować określony typ wartości nadany im poprzez jawną deklarację w programie lub (niejawnie) gdy są rezultatem wykonywanych operacji,
- *interferencja danych* - powodowana wzajemną, nieplanowaną modyfikacją wspólnych danych wynikającą z błędów

projektowych, błędów kodowania, błędów danych zewnętrznych, usterek sprzętowych,

- *interferencja usług systemowych* - wynikająca na przykład z wzajemnego blokowania dostępu do procesora,
- *interferencje sterowania programów* - realizacja przerwań, procesów równoległych sterowanych sygnałami zewnętrznymi, a także w wyniku błędów w sterowaniu programów prowadzących do wywołania niewłaściwych procedur,
- *awarie o wspólnych przyczynach* - ponieważ moduły programowe wykorzystują wspólne zasoby (co nie jest uwidocznione w tekście programu) ich awarie są często silnie skorelowane gdyż awaria współdzielonego zasobu powoduje awarię wszystkich modułów programowych wykorzystujących ten zasób.

Możliwość przeprowadzenia skutecznej analizy FMEA dla oprogramowania, na poziomie elementów niskiego poziomu (struktur języka programowania widocznych dla projektantów) jest zależna od zdefiniowania odpowiedniej semantyki tego oprogramowania. Potrzebna jest semantyka przyporządkowująca programom własności w sposób wyjaśniający wpływ jaki program wywiera w swoim środowisku, a nie to, jak program działa.

Jakość kodu wynikowego

Dla systemów sprzętowych, wnioski z analizy FMEA ze znaczną pewnością odpowiadają rzeczywistości (tzn. opisują stan rzeczywistego systemu). Analiza obejmuje swym zakresem awarie elementów wynikające z defektów losowych, czyli takich które powstają w trakcie użytkowania (starzenie się elementów, zużycie, oddziaływania elektromagnetyczne, wibracje, zapylenie, itp.). Awarie takie są zwykle dobrze rozumiane w sensie ich intensywności w odniesieniu do typu elementu i warunków jego eksploatacji.

W wypadku oprogramowania, sytuacja jest zasadniczo różna. Oprogramowanie nie zawiera defektów losowych (nie podlega procesom starzenia się i zużycia). Podstawowy rodzaj defektu oprogramowania to defekt projektowania. W ogólnym przypadku nie możemy być pewni, że kod wynikowy tworzonego oprogramowania jest pozbawiony defektów (wprowadzonych przez projektantów, programistów lub oprogramowanie narzędziowe). Te właśnie defekty mają zasadniczy wpływ na to, że zachowanie oprogramowania różni się od oczekiwanego, wyrażonego specyfikacją danego modułu. Tak więc, oprócz analizy dotyczącej możliwych awarii wynikających z defektów zasobów fizycznych będących „nośnikiem" programu (procesory, pamięć, linie komunikacyjne) konieczne jest objęcie analizą procesu wytwarzania, który jest odpowiedzialny za defekty projektowania wprowadzane do oprogramowania. Analiza FMEA dotycząca architektury systemu z oprogramowaniem powinna więc być rozszerzona analizą FMEA procesu wytworzenia danego systemu. Do przeprowadzenia takiej analizy konieczna jest znajomość efektywności stosowanych technik i metod („elementów" tego procesu) oraz zależności między nimi. Wykonywanie takiej analizy jest postulowane przez standardy, np. standard DS 00-55 [2]. Podobnie jak miało to miejsce w wypadku „klasycznej" analizy FMEA, analiza procesu wytwarzania powinna być prowadzona od najwcześniejszych etapów cyklu życia oprogramowania, a wynikające z niej wnioski mogą być wykorzystane do poprawy tego procesu, poprzez modyfikacje i zabiegi mające na celu zwiększenie skuteczności stosowanych metod wytwarzania.

Uzależnienie od środowiska

Często defekt oprogramowania prowadzi do poważnych skutków (wypadku) w przypadku jednoczesnej usterki sprzętowej, pomyłki człowieka (użytkownika, operatora) lub określonego wpływu ze strony środowiska. W badaniach, na które powołano się w [1], 35 % wszystkich awarii oprogramowania wynikała z zależności programowo-sprzętowych. Według innych danych eksperymentalnych, znaczna liczba awarii ma charakter chwilowy (zwykle pojedyncze zdarzenia powodowane przez przemijające warunki - na przykład, element sprzętowy ulega chwilowej awarii i niszczy dane). W takich sytuacjach możliwie jest przywrócenie poprawnego funkcjonowania, np. poprzez wznowienie programu od (zachowanego wcześniej) stanu poprawnego [11]. Standardy EN 50128, DS 00-55 i IEC 812 [4, 2, 9] postulują łączne traktowanie sprzętu i oprogramowania podczas identyfikacji nieprawidłowości działania i wykonywanie FMEA dla całego systemu.

4. FMEA W ADTRANZ ZWUS

Analiza dotychczasowych doświadczeń była podstawą do zdefiniowania programu zapewniania bezpieczeństwa systemów, sformułowanego w podsumowaniu etapu szkoleniowo-badawczego w Adtranz Zwus. Adtranz Zwus jest producentem systemów sterowania ruchem kolejowym, przeznaczonych zarówno na rynek krajowy jak i międzynarodowy. Systemy te muszą charakteryzować się bardzo wysokim poziomem bezpieczeństwa wynikającym z norm i regulacji branżowych i międzynarodowych. W związku z tym, poszukiwano efektywnych technik zapewniania niezawodności i bezpieczeństwa i badano możliwości spełnienia wymagań bezpieczeństwa zawartych w normach CENELEC EN 50126/-8/-9 oraz PKP/CNTK [3, 4, 5, 12]. W procesie szkoleń oraz w trakcie projektów pilotowych kształtowano u projektantów zrozumienie zaawansowanych technik analitycznych oraz ich związku z bezpieczeństwem systemów.

Jednym z projektów pilotowych, w którym wdrożono elementy programu jest projekt realizacji *systemu komputerowej blokady liniowej* SHL-1. W projekcie tym zdefiniowano następujące cele analizy FMEA:

- identyfikacja reakcji systemu na pojedyncze awarie wyróżnionych elementów (w tym awarii podczas planowanych reakcji systemu na inne awarie),
- identyfikacja i ocena zagrożeń związanych z tymi awariami,
- sformułowanie wniosków dla testowania lub dalszych działań projektowych,
- uzasadnienie poprawności proponowanej struktury (opisanej schematami blokowymi), ze względu na wymagania niezawodności i bezpieczeństwa),
- uzupełnienie pakietu bezpieczeństwa (ang. *safety case*) systemu.

Za wymaganie podstawowe uznano oparcie prac o spójną i jednoznaczną dla realizatorów analizy terminologię (por. [7]), zgodną z pozostałą dokumentacją i działaniami projektowymi. Przed rozpoczęciem analizy zdefiniowano stan bezpieczny systemu i stany bezpieczne rozproszonych geograficznie elementów. Ustalono tryby pracy systemu, logiczne granice systemu i definicje innych stosowanych terminów.

Zdefiniowano tabelę dokumentacji końcowej wyników analizy. W tabeli tej zawarto następujące kolumny: *rodzaj* awarii, *efekt* awarii (lokalny, wtórny, końcowy), *przyczyna* awarii, *ocena skutków* (ryzyka dla tej awarii), *zalecane działania* (naprawcze, zapobiegawcze), *sposób i czas ujawnienia* awarii, *przejście do stanu bezpiecznego*.

W analizie wybrano elementy odpowiadające poziomowi struktury systemu, na którym podejmowane są reakcje na zagrożenia bezpieczeństwa systemu (bezpieczeństwo strukturalne). Do tej listy dołączono istotne moduły współpracujące z systemem (udostępniające dane lub sterowane przez system) jako elementy środowiska.

Awarie wyróżnionych elementów zostały zdefiniowane jako zaprzeczenie usług (funkcji) tych elementów świadczonych na rzecz innych elementów. Przyjęto następującą ogólną definicję usługi dostarczanej przez dany element: *właściwa operacja (dane, sygnały) we właściwym czasie*. Dane o awariach zebrano w następującym zestawieniu:

Tabl.1. Schemat dokumentacji awarii elementów

Lp.	Element	Lista awarii elementu	Uzasadnienie listy

Przy ustalaniu list awarii elementów posługiwano się listą kontrolną (ang. *checklist*). Lista ta obejmowała:

- błędy użytkowników/operatorów lub systemów współpracujących,
- utratę danych wejściowych (w tym konfiguracyjnych),
- programowo-sprzętowe awarie związane z błędami synchronizacji, błędami wykrywania i naprawy innych błędów, błędami interakcji, błędami kontroli współbieżności,
- zalecenia o awariach systemów transmisji zawarte w normie EN 50159 [6].

Przyjęto, że awarie połączeń nie wymienione jawnie w analizie, ograniczone są do braku (zaniku) danego połączenia i przypisane są odpowiedniemu elementowi przynoszącemu daną usługę.

W identyfikacji scenariuszy błędów wyróżniono trzy podstawowe typy błędów:

- błędy (synchronizacji) czasu usługi,
- błędy wartości parametrów/sygnałów,

- błędy wykorzystania zasobów/informacji.

Przyjęto zasadę, że podczas tworzenia scenariusza błędów, w przypadkach wątpliwych lub niejednoznacznych, uwzględniany jest rozwój scenariusza w kierunku „najgorszego przypadku".

Za podstawę oceny krytyczności awarii przyjęto stopień degradacji funkcji dotkniętej daną awarią:

- możliwy zakres strat wynikający ze skutków awarii,
- czas trwania zakłócenia/utraty/degradacji funkcji systemu,
- dostępność środków alternatywnych umożliwiających (bezpieczną) kontynuację ruchu przy ograniczonej funkcjonalności systemu.

W uzupełnieniu rozważano ocenę i możliwości diagnostyczne operatora systemu. Przyjęto następującą skalę krytyczności awarii:

Tabl.2. Skala krytyczności awarii

Skala krytycz-ności	Obiektywna ocena sytuacji	Ocena z punktu widzenia użytkownika/operatora
5	Źle.	Użytkownik myśli, że jest dobrze.
4	Źle.	Użytkownik nie wie, co jest źle.
3	Źle.	Użytkownik wie o tym (niefunkcjonalność).
2	Umiarkowanie źle.	Użytkownik wie o tym (zakłócenie).
1	Minimalnie źle.	Nie ma znaczenia.

W ramach procesu analizy przyjęto następującą strategię:

- pomimo tego, że analiza struktury wysokiego poziomu wskazuje, że jest ona bezpieczna, należy uwzględnić, że interakcje elementów niższego poziomu mogą również prowadzić do zagrożeń [6],
- każda awaria brana pod uwagę w trakcie analizy powinna być rozważana jako symptom pewnego procesu, który się „pod nią" kryje.

Z listy wyróżnionych elementów wybrano trzy elementy *krytyczne*, dla których przeprowadzono lokalną analizę FMEA. Konsekwencje awarii ich elementów funkcjonalnych (modułów programowo-sprzętowych) weryfikowały (wyjaśniały lub rozszerzały) wcześniej utworzoną listę awarii elementów krytycznych.

W stosunku do krytycznych modułów programowych, w ramach zapewniania bezpieczeństwa systemu przewidziano zrealizowanie zalecenia normy EN 50128 przewidującego zastosowanie techniki testowania architektury SEEA (ang. *Software Error Effect Analysis*) [4]. Technika ta polega na rozważeniu konsekwencji możliwych błędów na poziomie wybranych instrukcji lub bloków kodu źródłowego.

W ramach programu przewidziano również, że w uzupełnieniu do analizy FMEA zostanie zrealizowana analiza drzew błędów FTA (ang. *Fault Tree Analysis*). Punktem wyjścia dla ustalenia zbioru zdarzeń szczytowych, dla których opracowane zostaną drzewa błędów są awarie zidentyfikowane w trakcie analiz FMEA.

5. PODSUMOWANIE

W artykule przedstawiono idee wykorzystania analizy FMEA w ramach programu zapewniania bezpieczeństwa systemów, których znaczącym składnikiem jest oprogramowanie. Ich częściowe wdrożenie umożliwia wykonanie kolejnego kroku. Postulat dla tego kroku można sformułować następująco - należy wzmacniać bazę dla analizy FMEA poprzez:

- poprawę jakości źródeł danych do procesu analizy, a w szczególności poprawę jakości powstających specyfikacji i modeli,
- integrację procesu FMEA z innymi (równolegle wykonywanymi) działaniami w ramach procesu projektowo-wytwórczego,
- poprawę efektywności procesu zapewniania jakości tworzonego oprogramowania.

Jednym z rozważanych kierunków jest wykorzystanie technologii obiektowej w procesie projektowania i wytwarzania oprogramowania. Do zalet technologii obiektowej, w kontekście problemów analizy FMEA, można zaliczyć:

- nacisk na jednoznaczność dekompozycji systemu i niezależność elementów (ich interfejsy mają formę precyzyjnych 'kontraktów'),
- modelowanie z wielu perspektyw (np. w ramach metodyki OMT [13] tworzone są modele z perspektywy statycznej, dynamicznej i funkcjonalnej),
- elastyczność i stabilność; modele łatwo poddają się modyfikacjom, a dokonywane

zmiany lokalizują się w ograniczonej ilości obiektów,

- możliwość „gładkiego" przejścia od analizy do projektowania i kodu programów; obiekty zidentyfikowane na etapie analizy mają swoją bezpośrednią reprezentację w implementacji.

LITERATURA

[1] BOWEN J., STAVRIDOU V.: Safety-Critical Systems, Formal Methods and Standards. *Oxford University Computing Laboratory Technical Report*, PRG-TR-5-92, 1992.

[2] Defence Standard 00-55, Requirements for Safety Related Software in Defence Equipment (Part 1&2), Issue 1, UK Ministry of Defence, 1997.

[3] EN 50126: Railway applications. The Specification and Demonstration of Dependability, Reliability, Availability, Maintainability and Safety (RAMS), CENELEC, Final Draft version, June 1997.

[4] EN 50128: Railway applications. Software for railway control and protection systems, CENELEC, Final Draft version, June 1997.

[5] EN 50129: Railway applications. Safety Related Electronic Systems for Signalling, CENELEC, ver. 1.0, January 1997.

[6] EN 50159-1/2: Railway applications. Requirements for Safety Related Communication in Closed/Open Transmission Systems, CENELEC, ver. 0.7, April 1996.

[7] GÓRSKI J.: Extending Safety Analysis Techniques with Formal Semantics, w: *Technology and Assessment of Safety Critical Systems (F. J. REDMILL, Ed.)*. Springer-Verlag, 1994.

[8] HIGUERA R. P., HAIMES Y. Y.: Software Risk Management, *CMU, Software Engineering Institute Technical Report*, CMU/SEI-96-TR-012. June 1996.

[9] IEC 812 (1985): Procedure for failure mode and effects analysis (FMEA), TC56 (polskie tłumaczenie: PN-IEC 812: 1994).

[10] LEVESON N., SANDYS S., and others: Safety Analysis of Air Traffic Control Upgrades, Final Report of NASA Langley, NASA Ames & Univ. of Washington (CS) Project. Sept. 1997.

[11] OWRE S., RUSHBY J., SHANKAR N., von HENKE F.: Formal Verification for Fault-Tolerant Architectures: Prolegomena to the Design of PVS, *IEEE Trans. on Software Eng.*, vol. 21, no. 2, February 1995, ss. 107-125.

[12] PKP / Centrum Naukowo - Techniczne Kolejnictwa, Zadanie nr 1060/23: Wymagania bezpieczeństwa dla urządzeń sterowania ruchem kolejowym. Warszawa, wrzesień 1997.

[13] RAMBAUGH J., BLAHA M., PREMERLANI W., EDDY F., LORENSEN W.: *Object-Oriented Modelling and Design*, Prentice-Hall Int., 1991.

Application of FMEA to Safety Analysis of Industrial Computer Applications

Abstract: The article presents problems involved in the application of the FMEA method within the context of safety assurance framework for software systems. Some experiences related to the attempt to use FMEA to the computerised railway signalling applications in Adtranz Zwus are also reported.