ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ **Nr 279**
Mechanika z. 83 (2/11) **2011**

Piotr MICHAŁOWSKI
Włodzimierz ZUBKO
Instytut Techniczny Wojsk Lotniczych

MODELOWANIE ARCHITEKTURY ZINTEGROWANYCH SYSTEMÓW AWIONICZNYCH

Coraz większe możliwości sprzętu komputerowego są przyczyną podwyższenia wymagań na nowe, bardziej złożone funkcje zintegrowanych systemów awionicznych. Prowadzi to do znacznego powiększenia stopnia złożoności systemu, a w konsekwencji do ryzyka pojawienia się błędów. Jednym ze sposobów rozwiązania tego problemu jest prowadzenie projektów według metodyki opartej na modelach, np. MDE (*Model Driven Engineering*). W pracy do modelowania został wykorzystany język AADL (*Architecture Analysis & Design Language*), który ze względu na swoje właściwości (opis struktury i parametrów systemu, modyfikowalność) dobrze opisuje architekturę zintegrowanego systemu awionicznego. W artykule przedstawiono model architektury zintegrowanego systemu awionicznego śmigłowca. Omówiono sposoby i narzędzia badania uzyskanego modelu oraz możliwości jego wykorzystania na różnych etapach cyklu projektowego.

Wstęp

Współczesne systemy awioniczne ze względu na realizowane funkcje wpływają na poprawę bezpieczeństwa lotów. Równocześnie coraz to nowsze możliwości sprzętu przyczyniają się do wzrostu wymagań co do bardziej złożonych funkcji systemów awionicznych. Prowadzi to do znacznego powiększenia stopnia złożoności systemu, a w konsekwencji do ryzyka pojawienia się błędów. Z drugiej strony ich niepoprawne działanie może skutkować ofiarami (nawet śmiertelnymi), poważnymi stratami materialnymi lub zniszczeniem środowiska naturalnego. Systemy takie nazywane są systemami krytycznymi dla bezpieczeństwa (*Safety Critical Systems*, SCS). Niekiedy niepoprawne działanie systemu uniemożliwia wykonywanie zadań związanych z jego misją. Systemy posiadające taką właściwość są zaliczane do kategorii systemów krytycznych dla realizowanej misji (*Mission Critical Systems*, MCS). Dlatego proces tworzenia systemów awionicznych należących do klasy SCS lub MCS musi być oparty na metodach i narzędziach umożliwiających wykrywanie i eliminację ewentualnych błędów we wczesnym etapie projektu. Obecnie zalecaną metodą jest wy-

twarzanie systemów według modelu MDE. Takie podejście jest uważane za właściwą metodę tworzenia systemów krytycznych, ponieważ do badania poprawności systemu wykorzystuje formalną weryfikację modeli (*model checking*) opartą na badaniu sieci Petriego [6].

Metoda MDE wymaga tworzenia modeli systemów w języku dziedzinowym. Kładzie ona nacisk na analizę systemu, a następnie na detekcję błędów we wczesnym etapie projektowania [1]. W pracy do modelowania systemów awionicznych został wykorzystany język AADL, który ze względu na swoje właściwości (opis struktury i parametrów systemu, modyfikowalność, rozszerzalność, przyjazność dla użytkownika) dobrze pasuje do opisu architektury systemów. Standard AADL jest często stosowany w analizie i projektowaniu systemów awionicznych [8]. W artykule przedstawiono model architektury zintegrowanego systemu awionicznego śmigłowca. Omówiono sposoby i narzędzia badania uzyskanego modelu oraz możliwości jego wykorzystania na różnych etapach cyklu projektowego.

1. Standard opisu architektury AADL

Informacje wstępne

Standard opisu architektury AADL [10] został opublikowany przez SAE (*Society for Automotive Engineers*). Definiuje on język dziedzinowy, oparty na pojęciu komponentu. Jest on przeznaczony do modelowania, analizy i projektowania systemów współbieżnych, a w szczególności systemów czasu rzeczywistego, systemów wbudowanych, systemów krytycznych ze względu na bezpieczeństwo, realizowaną misję lub wydajność. Ponieważ zintegrowane systemy awioniczne są najczęściej zaliczane do klasy systemów krytycznych, to język AADL może być wykorzystany do ich modelowania, analizy i projektowania. Dodatkowo za pomocą tego języka można budować modele istniejących i działających systemów, a następnie analizować ich zachowanie w zmieniającym się środowisku.

Stworzona w języku AADL specyfikacja modelowanego systemu składa się z ciągu deklaracji opisujących jego strukturę i realizowane funkcje. W celu uporządkowania opisu deklaracje mogą być pogrupowane w pakiety (*packages*). Jak już wspomniano, język AADL jest oparty na pojęciu komponentu. Komponenty reprezentują zarówno sprzętowe, jak i programowe składowe systemu. Dodatkowo wprowadzono pojęcie komponentu hybrydowego, który reprezentuje pojęcie systemu systemów. Pomiędzy poszczególnymi komponentami mogą przepływać dane i zdarzenia. Język AADL umożliwia opis właściwości poszczególnych przepływów. Opisy te mogą być następnie wykorzystywane do analizy zachowań modelowanego systemu. Język pozwala na stworzenie hierarchicznej struktury modelu z wykorzystaniem mechanizmów dziedziczenia typów.

Modelowany w języku AADL system jest zbiorem abstrakcyjnych komponentów programowych, sprzętowych oraz hybrydowych, które są połączone wzajemnymi relacjami, takimi jak połączenia i powiązania. Połączenia realizują przepływy danych i zdarzeń, natomiast powiązania odwzorowują wewnętrzną strukturę systemu i łączą komponenty programowe z komponentami sprzętowymi. Każdy komponent posiada opis definicji typu i co najmniej jeden opis implementacji [7]. Poziom opisu definicji typu zawiera informacje o tym, w jaki sposób dany komponent jest widoczny dla otoczenia, natomiast opis implementacji – o jego zachowaniu (działaniu).

W standardzie AADL zarówno architektura oprogramowania, jak i sprzętowe środowisko wykonawcze mogą być opisywane za pomocą trzech form zapisu: tekstowej, graficznej albo wykorzystującej język XML (*Extensible Markup Language*). Zapis tekstowy stanowi opis referencyjny składni języka AADL. Posiada on największą siłę wyrazu i pozwala na precyzyjne definiowanie komponentów. Zapis graficzny, prezentowany w postaci diagramów, jest zgodny z zapisem tekstowym. Oznacza to, że każdy diagram w systemie może być opisany za pomocą notacji tekstowej. Zapis graficzny umożliwia ogólny przegląd architektury modelowanego systemu. Niestety nie jest on użyteczny w trakcie modelowania dużych i skomplikowanych systemów. Trzecia forma zapisu jest oparta na języku XML. Sposób ten jest w pełni zgodny z zapisem tekstowym. Jest on wykorzystywany przez oprogramowanie do maszynowej analizy i symulacji działania modelowanego systemu.

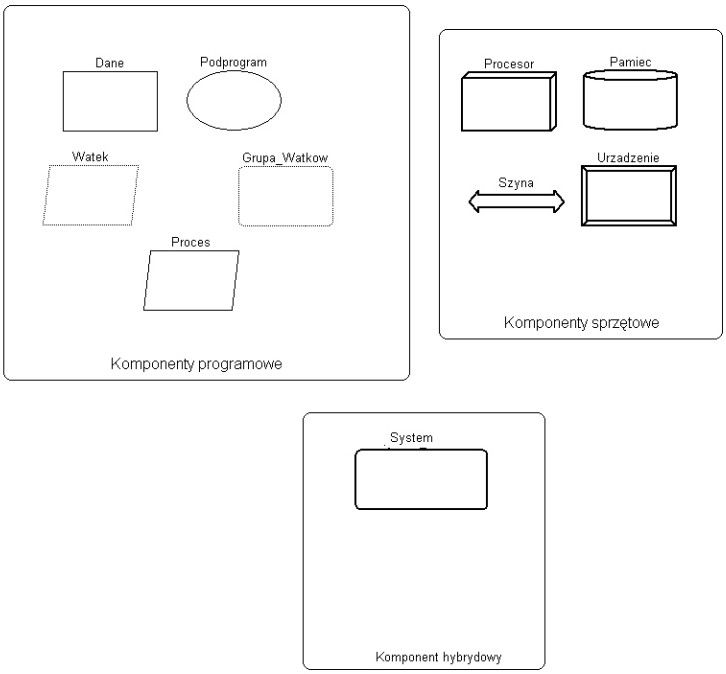
Standard AADL opisuje modelowany system jako zbiór abstrakcyjnych komponentów sprzętowych, programowych oraz hybrydowych (rys. 1.). Do grupy komponentów programowych zaliczane są: wątki, grupy wątków, dane, podprogramy i procesy.

Wątki (*threads*) reprezentują aktywne, współbieżnie wykonywane elementy systemu (podobnie jak wątki standardu POSIX), których wykonanie podlega regułom szeregowania. Wątki mogą być łączone w grupy (*thread groups*), czyli abstrakcyjne struktury umożliwiające ich logiczną organizację. Komponenty typu podprogram (*subprogram*) reprezentują fragmenty wykonywanego sekwencyjnie kodu programowego. Mogą one występować jedynie jako części składowe wątków. Komponenty typu proces (*process*) reprezentują wykonywalne programy, które są odpowiedzialne za ochronę pamięci, i stanowią dla wątków ich środowisko wykonawcze. Każdy proces powinien zawierać co najmniej jeden wątek działający w jego środowisku. W modelowanym systemie komponenty typu dane (*data*) reprezentują potencjalnie współdzielone struktury danych.

Następną grupą komponentów, wchodzących w skład standardu AADL, są komponenty sprzętowe, zwane również komponentami platformy wykonawczej (*hardware*/*execution platform components*). Do tej grupy zalicza się: pamięć, procesor, urządzenie i szynę danych. Komponenty te reprezentują sprzęt komputerowy. Ich opis zawiera strukturę oraz właściwości środowiska wykonawczego.

Standard AADL definiuje metody dowiązywania komponentów programowych do komponentów sprzętowych. Do realizacji takiej funkcji niezbędny jest komponent hybrydowy typu system, który należy do wydzielonej, jednoelementowej grupy. Rola komponentów typu system polega na zapewnieniu możliwości integracji komponentów składowych (zarówno programowych, jak i sprzętowych) w jednolity system.

W języku AADL z opisem danego komponentu zawiązane są następujące elementy: deklaracje komponentów składowych (*subcomponents*), definicje cech (*features*), opisy przepływów (*flows*) i połączeń (*connections*) oraz właściwości (*properties*). Opisy komponentów składowych, cech, połączeń i przepływów służą do definiowania struktury funkcjonalnej modelowanego systemu. Przykładowymi cechami komponentu są jego porty wejścia/wyjścia, które stanowią punkty dostępu do wnętrza danego komponentu. Ze względu na kierunek przechodzenia informacji wyróżnia się trzy rodzaje portów: wejściowe, wyjściowe i dwukierunkowe. Ze względu zaś na rodzaj przesyłanych informacji porty w języku AADL są zaliczane do jednego z trzech typów: portów danych, portów zdarzeń oraz portów mieszanych, modelujących wejścia/wyjścia zdarzeń wraz z danymi, np. kolejki. Porty, które wchodzą w skład połączeń pomiędzy dwoma określonymi komponentami, mogą być grupowane w celu uporządkowania struktury systemu.



Rys. 1. Symbole graficzne opisujące komponenty w języku AADL

W języku AADL właściwości dostarczają informacji o elemencie zawartym w specyfikacji modelowanego systemu. Właściwości są używane do określania np.: rozmiaru porcji danych, okresu wątku cyklicznego, oczekiwanego czasu wykonania wątku lub opóźnienia wnoszonego przez połączenie albo przepływ. Każda właściwość posiada własną nazwę oraz określony typ, który określa zbiór wartości, jakie może przyjmować dana właściwość. Właściwości są w języku AADL deklarowane za pomocą nazwanych zbiorów, które często stanowią rozszerzenie standardu języka. Obowiązuje tu zasada, że właściwość zdefiniowana dla danego komponentu może jednocześnie przyjmować tylko jedną wartość.

Więcej informacji na temat tworzenia modeli w języku AADL znajduje się w pracach [2÷4]. W pracy [1] przedstawiono zastosowanie języka AADL do modelowania systemów spełniających wymagania standardu ARINC 653, dotyczących partycjonowania systemów awionicznych.

Metody analizy modelowanych systemów oraz wykorzystywane narzędzia programowe

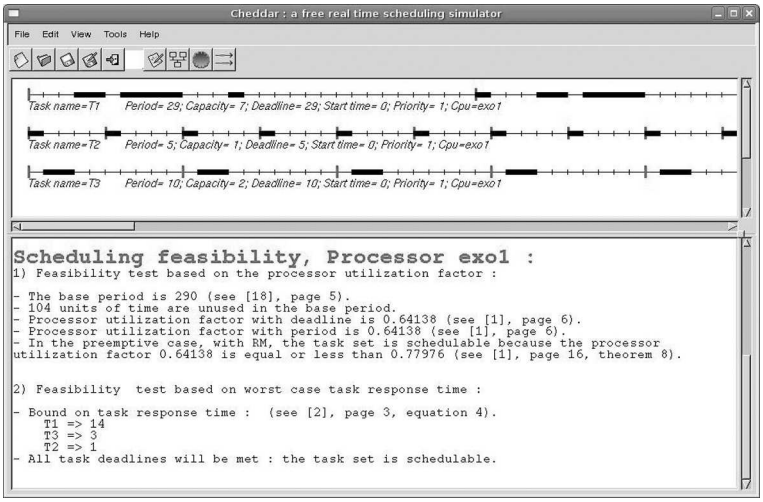
Tworzenie w języku AADL modeli systemów wbudowanych może być wspomagane przez pakiety oprogramowania narzędziowego. Przykładem takiego narzędzia jest opisany w publikacji [9] system OSATE autorstwa Uniwersytetu Carnegie-Mellon.

Utworzone w języku AADL modele systemów wbudowanych (w tym awionicznych) mogą być analizowane pod kątem spełnienia wymagań, które zostały wcześniej zdefiniowane za pomocą cech i właściwości dowiązanych do komponentów modelu. W zależności od przyjętych w modelu zbiorów właściwości, mogą być analizowane różnorakie właściwości, takie jak: wydajność (*performance*), szeregowalność (*schedulability*), sprawność przepływu informacji, zajętość zasobów oraz propagacja błędów.

Przykład analizy wydajności systemu został przedstawiony w pracy [5]. Autorzy omawiają sposób analizy wydajności systemu na etapie jego projektowania. W pracy zastosowano oryginalny zbiór właściwości systemu określający czasowy koszt wykonania pojedynczych instrukcji maszynowych w modelowanym systemie. Na podstawie metody najmniejszych kwadratów podano sposób oszacowania poszczególnych czasów wykonania instrukcji dla określonego typu procesora. Następnie wątki w modelowanych systemach są charakteryzowane wektorem krotności wykonywanych w pojedynczym przebiegu instrukcji maszynowych. Pozwala to na określenie przewidywanego czasu realizacji. Po uzyskaniu oszacowań system jest modelowany za pomocą odpowiednich narzędzi programowych, np. programu Cheddar [12] (rys. 2.).

Pakiet oprogramowania Cheddar został opracowany przez zespół LISyC z Uniwersytetu w Brest we Francji. Jest on przeznaczony do sprawdzania ograniczeń czasowych aplikacji, w tym systemów czasu rzeczywistego, opisanych m.in. w języku AADL. Oprogramowanie wspomaga prototypowanie systemów

i pozwala na symulację (*simulation*) ich działania oraz na analizę możliwości (*feasibility analysis*) spełniania wymagań szeregowalności.



Rys. 2. Okno pakietu Cheddar z przykładowym przebiegiem symulacji; strona internetowa projektu [13]

Niekiedy analiza szeregowalności nie jest wystarczająca do określenia występowania blokad, zakleszczeń i zagłodzeń modelowanego systemu. Dlatego w pracy [6] została podana metoda przekształcania diagramów i opisów języka AADL w symetryczne, kolorowane sieci Petriego, które następnie są przedmiotem weryfikacji metodami teorii grafów. Innym sposobem analizy systemów opisanych w języku AADL jest podana w pracy [11] metoda polegająca na przetłumaczeniu ich na procesy stanowiące elementy algebry procesów czasu rzeczywistego ACSR (*Algebra of Communicating Shared Resources*), a następnie na badaniu uzyskanych obiektów metodami matematycznymi.

Do standardu AADL został dodany aneks zawierający mechanizmy modelowania zjawiska pojawiania się i propagacji błędów w modelowanym systemie. Do języka zostały dodane komponenty reprezentujące stany błędów i stany poprawnego działania. Ponadto został on uzupełniony o zdarzenia powodujące przejścia pomiędzy tymi stanami. Właściwości komponentów zostały uzupełnione o rozkłady prawdopodobieństwa zajścia określonych, zdefiniowanych w modelu zdarzeń. Głównym celem wprowadzenia do języka AADL przedstawionych mechanizmów jest zapewnienie możliwości uproszczonego modelowania niezawodności i bezpieczeństwa systemów. W pracy [7] została podana metoda przekształcania opisu modelowanego systemu z języka AADL w uogólnione stochastyczne sieci Petriego, które następnie mogą być analizowane metoda-

mi rachunku prawdopodobieństwa, np. za pomocą pakietu oprogramowania narzędziowego SURF2 [14].

2. Zintegrowany system awioniczny śmigłowca

W Instytucie Technicznym Wojsk Lotniczych został opracowany zintegrowany system awioniczny dla śmigłowca W3-PL „Głuszec". W artykule zaprezentowano jedynie uproszczony fragment (opisany w języku AADL) modelu wymienionego systemu. Opis został ograniczony do systemu obsługi misji, który stanowi element zarządzający pracą systemu. Jest on odpowiedzialny za wymianę danych pomiędzy zewnętrznymi urządzeniami (podsystemami) typu: terminal, system nawigacyjny, system uzbrojenia oraz system CSAR. Dodatkowo system obsługi misji na potrzeby systemu uzbrojenia realizuje funkcje wyliczeń balistycznych.

System nawigacyjny jest odpowiedzialny za dostarczanie prawidłowych danych o położeniu oraz wektorze ruchu śmigłowca. System uzbrojenia steruje uzbrojeniem podwieszonym do śmigłowca. System CSAR realizuje funkcje poszukiwawczo-ratownicze. Interfejs użytkownika dla wspomnianych urządzeń–systemów jest realizowany przez terminal. System ten wizualizuje dane pochodzące z urządzeń na tle mapy w kontekście przygotowanego wcześniej planu misji. Dodatkowo system terminal umożliwia użytkownikowi wprowadzanie danych i nastaw do zintegrowanego systemu awionicznego.

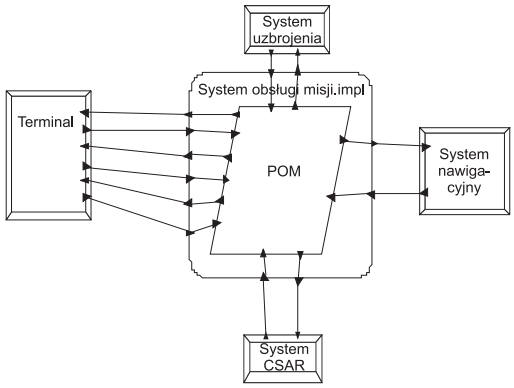
System obsługi misji jest połączony z terminalem za pomocą szybkiej lokalnej sieci komputerowej. Wymiana danych z pozostałymi systemami zewnętrznymi jest realizowana poprzez szyny, których działanie oparto na standardzie MIL 1553B. Każde urządzenie podłączone jest do systemu za pomocą oddzielnej szyny.

Oprogramowanie systemu obsługi misji zostało zrealizowane jako pojedynczy, wielowątkowy proces obsługi misji POM. W skład tego procesu wchodzą trzy współbieżne, kooperujące wątki programowe: WątekUzbr, WątekCSAR i WątekIntegracji.

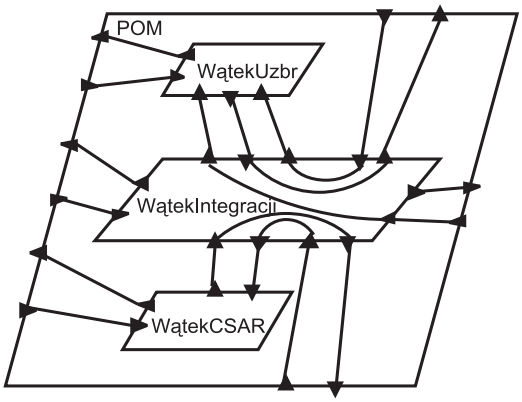
Pierwszy z wymienionych wątków jest odpowiedzialny za generowanie danych dla systemu uzbrojenia, drugi steruje urządzeniami poszukiwawczo-ratowniczymi CSAR. Natomiast ostatni wątek zajmuje się integracją danych przychodzących z sieci lokalnej i szyn. Wątek ten porządkuje dane w pakiety, które są następnie przesyłane do terminala. Strzałki na rys. 3. i 4. przedstawiają przepływy danych i sterowania pomiędzy poszczególnymi elementami systemu.

Każdy wątek zintegrowanego systemu awionicznego jest opisany zbiorem właściwości należących do języka AADL. Określają one typ danego wątku (cykliczny, acykliczny), okres wywołania cyklu, szacowany czas wykonania zadania oraz maksymalne jego ograniczenie. Natomiast zdefiniowane w systemie przepływy danych są opisywane czasami ich realizacji. Struktura systemu oraz

wymienione dane stanowią wejście dla systemów modelowania i symulacji, np. pakietu Cheddar.



Rys. 3. Uproszczony schemat modelu zintegrowanego systemu awionicznego śmigłowca



Rys. 4. Wielowątkowy schemat procesu obsługi misji POM

Zintegrowany system awioniczny śmigłowca W3-PL został opisany za pomocą języka AADL dopiero po etapie jego implementacji. Do stworzenia modelu wykorzystano istniejący kod oprogramowania. Działanie to może być traktowane jako sposób pozyskiwania „wiedzy” z istniejącego systemu. Przyjęty został następujący algorytm postępowania:

- krok 1. – na podstawie istniejącego kodu wytworzenie struktury modelu zintegrowanego systemu awionicznego w języku AADL, wprowadzenie właściwości niefunkcjonalnych (np. ograniczeń czasowych),

- krok 2. – wyznaczenie zbioru „działań elementarnych” (podprogramów i innych sekwencji kodu) realizowanych przez poszczególne wątki, a następnie estymacja czasów ich realizacji metodami opisanymi w pracy [5],
- krok 3. – symulacja działania uzyskanego modelu z wykorzystaniem specjalizowanego narzędzia, np. pakietu Cheddar,
- krok 4. – weryfikacja modelu poprzez porównanie uzyskanych wyników symulacji z zarejestrowanymi wcześniej parametrami pracy zintegrowanego systemu awionicznego.

Obecnie trwają prace nad wyznaczeniem zbioru działań elementarnych oraz weryfikacją zgodności otrzymanego modelu z systemem rzeczywistym. W przyszłości planowane jest wykorzystanie otrzymanego modelu do weryfikacji poprawności działania systemu awionicznego, badania jego ścieżek krytycznych oraz analizowania możliwości jego dalszego rozwoju. Wyniki tych prac zostaną opublikowane w późniejszych artykułach.

3. Podsumowanie

W pracy przedstawiono standard opisu architektury wbudowanych systemów czasu rzeczywistego AADL oraz jego zastosowanie do modelowania i analizy zintegrowanych systemów awionicznych. Standard AADL może być stosowany podczas wszystkich etapów procesu analizy, projektowania, testowania, wytwarzania i eksploatacji zintegrowanych systemów awionicznych. Autorzy wyrażają pogląd, że podejście przedstawione w artykule jest szczególnie użyteczne w procesie planowania modernizacji zintegrowanych systemów awionicznych.

Literatura

[1] Delange J., Pautet L., Plantec A., Kerboeuf M., Singhoff F., Kordon F.: Validate, simulate, and implement ARINC653 systems using the AADL, ACM SIGAda Ada Letters Archive, vol. 29(3), 2009, s. 31÷44.
[2] Feiler P.H., Gluch D.P., Hudak J.J.: The Architecture Analysis & Design Language (AADL): An introduction, Technical report, SEI, Carnegie Mellon University, 2006.
[3] Feiler P.H., Gluch D.P., Hudak J.J., Levis B.A.: Embedded system architecture analysis using SAE AADL, Technical note, CMU/SEI-2004-TN-005, SEI, Carnegie Mellon University, 2004.
[4] Hudak J.J., Feiler P.H.: Developing AADL models for control systems: A practitioner's guide, SEI, Carnegie Mellon University, 2007.
[5] Januzaj V., Mauersberger R., Biechele F.: Performance modelling for avionics aystems, [in:] EUROCAST 2009, R. Moreno-Díaz (ed.), Springer-Verlag, Berlin Heidelberg 2009, s. 833÷840.

[6] Renault X., Kordon F., Huguens J.: From AADL architectural models to Petri nets: Checking model viability, IEEE International Symposium on Object/Component/ /Service-Oriented Real-Time Computing, 2009, s. 313÷320.
[7] Rugina A.E.: Dependability modeling and evaluation – from AADL to stochastic Petri nets, Systèmes. Informatiques. Toulouse: PhD dissertation in LAAS CNRS, Toulouse 2007.
[8] Sioutis C., Nguyen T.: Modelling airborne mission systems using the Architecture Analysis and Design Language, Proc. 18th World IMACS/MODSIM Congress, Cairns, Australia 13÷17 July 2009, http://mssanz.org.au/modsim09.
[9] Society of automotive engineers: Getting started with the AADL and OSATE: An introductory tutorial using the AADL textual and object model editors, SEI, Carnegie Mellon University, 2007.
[10] Society of automotive engineers: SAE standards: Architecture Analysis & Design Language (AADL), AS5506 (11/2004), AS5506/1 (06/2006).
[11] Sokolsky O., Lee I., Clarke D.: Schedulability analysis of AADL models, Proc. 20th IEEE International Parallel & Distributed Processing Symposium, 2006.
[12] http://beru.univ-brest.fr/~singhoff/cheddar/
[13] http://beru.univ-brest.fr/~singhoff/chedlolar/#Ref1
[14] http://homepages.laas.fr/surf4tst/what-uk.html

MODELING THE ARCHITECTURE OF INTEGRATED AVIONICS SYSTEMS

A b s t r a c t

Growing capabilities of the computer hardware are a reason for appearing of requirements to new more complex functions of integrated avionics systems. It is causing that the systems complexity, and in consequence the risk of the appearance of errors are increasing. In order to reduce these problem, it is recommended to conduct projects according to well-known methodology based on e.g. Model- Driven Engineering. In the article, there are discussed a model of the architecture of the integrated avionics system for helicopter and methods of its analysis.

Złożono w Oficynie Wydawniczej w sierpniu 2011 r.