doi:10.3969/j.issn.1671-637X.2015.07.021

软件健康管理在ADIRU中的应用及验证

宫华伟[1], 李保中[2], 温永强[2]
(1.中国航空工业集团公司洛阳电光设备研究所,河南 洛阳 471000;
2.中国人民解放军驻六一三所军事代表室,河南 洛阳 471000)

摘 要:现代嵌入系统软件复杂性逐渐增加,使发现和处理运行时软件错误的难度也随之增大,实际应用表明,仍有少数的软件运行时错误并不能通过常规测试方法发现。在嵌入系统软件设计中引入了软件健康管理的方法,来增强嵌入系统软件的可靠性,该方法通过在嵌入系统软件中构建运行时故障监测、诊断和恢复模块来实现。为了验证该方法的有效性,以2005年马来西亚航空公司124航班机载ADIRU发生的软件故障为例,使用AADL语言为ADIRU建立软件模型,并通过该模型验证了软件健康管理方法的有效性。

关键词:软件健康管理;软件容错;时间故障传播有向图
中图分类号:TP302.8 **文献标志码**:A **文章编号**:1671-637X(2015)07-0102-05

Application and Demonstration of Software Health Management in ADIRU

GONG Hua-wei[1], LI Bao-zhong[2], WEN Yong-qiang[2]
(1. Luoyang Institute of Electro-Optical Equipment, AVIC, Luoyang 471000, China;
2. Military Representative Office of PLA in No. 613 Institute, Luoyang 471000, China)

Abstract: With the increase of complexity of modern embedded system software, it becomes more difficult to discover and handle all potential software faults. Practical application has shown that there are still some software runtime errors that cannot be discovered by conventional test method. A method was proposed by introducing software health management into the design of embedded system software, to improve the reliability of embedded system software. This method was realized by constructing runtime error monitoring, diagnosis and recovery models in the embedded system software. To verify its effectiveness, taking software errors occurred in 2005 in airborne ADIRU of Malaysian Air Flight 124 as an example, a software model was established for ADIRU by using AADL language, and the effectiveness of software health management was proved through this model.

Key words: software health management; software fault-tolerant; timed failure propagation graph

0 引言

2005年,马来西亚航空公司124航班(Boeing 777飞机)从珀斯飞往吉隆坡途中,飞机在高空突然以大角度爬升,然后失速,高度陡降,紧急断开自动驾驶仪后,才得以重新控制飞机。后来的事故调查表明事故是飞机上主大气数据惯性基准组件(ADIRU)出现的软件故障所导致。ADIRU为飞行控制计算机提供飞机空速、攻角、高度、位置以及姿态数据,是飞机上的关键电子设备。此事件说明,对于这种高安全性要求的嵌入系统,需要一种有效的方法来防止这种软件运行时错误,使系统在错误发生后,能够自动从故障中恢复,避免导致灾难性的后果。这对于构成航空电子设备核心的嵌入系统软件的安全运行具有特别重要的意义。

本文以前述发生故障的ADIRU为例,在为其构建的航空电子设备结构描述(AADL)语言[1]模型中引入了软件健康管理(SHM)方法,以增强嵌入系统软件的可靠性,并通过该模型进行了验证。

1 软件健康管理

软件故障在任何嵌入系统运行中都有可能出现。

收稿日期:2014-10-28 **修回日期**:2015-05-06
作者简介:宫华伟(1978—),男,河南洛阳人,硕士,研究方向为嵌入系统应用、机载光电系统总体设计。

虽然大多数的软件问题在之前可以通过常规的软件测试方法发现，但是仍有少数的软件故障可能在运行时发生。对于高安全性要求的嵌入系统，在设计中普遍使用了冗余和票选机制，但是，对于系统中发生的共模错误和由其他错误引起的系统潜在软件错误，这些机制也无能为力。还需要增加自管理功能，才能使系统在运行中具备对错误和故障的自适应能力，这样的系统称为自适应系统。

软件健康管理是软件容错技术中的一种，目的是为了实现软件对运行中出现故障时的自适应处理能力，构建一个软件自适应系统。此方法通过在软件中增加健康管理模块来进行软件故障监测、故障诊断和故障恢复，使系统软件具备容错能力。

本文在 ADIRU 模型中构建了这样的软件健康管理机制，通过在系统各功能单元中建立监测模块，实时监测软件运行时系统中数据流的有效性，根据发生的运行中错误，采用故障诊断算法确定故障源，最后通过隔离故障单元，使系统从软件错误中恢复正常运行。

2　ADIRU 嵌入系统模型

2.1　ADIRU 组成

Boeing 777 ADIRU 的组成[2]如图 1 所示，具有多层冗余结构。由 4 个容错区(FCA)组成，每个容错区有多个容错单元(FCM)。加速度传感器和陀螺仪各有 6 个容错单元，处理器有 4 个容错单元，供电电源有 3 个容错单元及 ARINC-629 的 3 个容错单元。若每个容错区中只发生一个故障，ADIRU 能够正常工作，不需要维护。若发生两个故障，ADIRU 仍然能够工作，但需要在下一次飞行前维护。

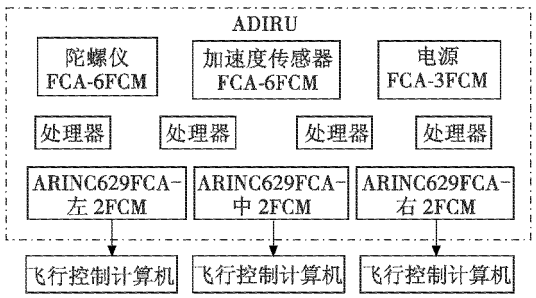

图 1　Boeing 777 ADIRU 组成

Fig. 1　The architecture of ADIRU

2.2　ADIRU AADL 模型

本文使用 AADL 语言为 ADIRU 建模，开发工具使用了 OSATE 2.0。AADL 语言最初应用于航空电子设备嵌入系统开发，现已发展成为嵌入系统建模的一种标准语言(SAE AS5506 标准)，非常适合于航空电子系统建模。ADIRU 模型采用 ARINC-653 标准构架[3]，使用 AADL 语言能够快速建立符合 ARINC-653 标准要求的嵌入系统模型，并通过模型对嵌入系统各方面的功能和性能进行分析。

建立的 ADIRU 模型与实际 Boeing 777 ADIRU 不完全相同，为了突出 ADIRU 所发生的故障，与事故无关的陀螺仪没有包含在模型中，另外模型运行时序也不同于真实系统中的运行时序。

如图 2 所示，模型中包含了加速度传感器单元、处理器单元、比较和显示单元 3 个基本功能单元以及软件健康管理单元。各单元作为一个 ARINC-653 进程，运行在各自独立的时间和内存空间中。3 个基本功能单元用来实现 ADIRU 的基本功能。软件健康管理单元用来实现系统的软件健康管理功能，是本文构建的核心单元。

首先说明 ADIRU 各基本功能单元的 AADL 模型构建，如图 2 所示。

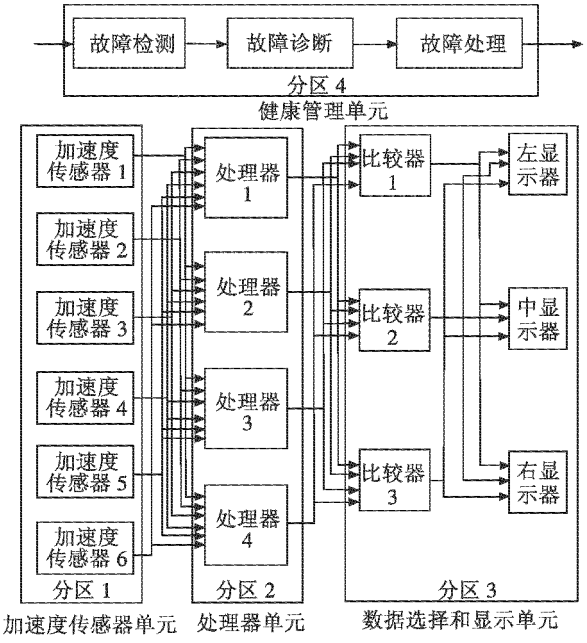

图 2　ADIRU 模型组成

Fig. 2　The architecture of ADIRU model

2.3　加速度传感器单元 AADL 模型

加速度传感器单元的功能是产生加速度数值。作为一个 ARINC-653 进程(acc_process. impl)进行建模，其 AADL 模型如图 3 所示。在进程内包含 6 个 thread 对象(acc1 ~ acc6)，每个对象实现一个加速度传感器的数据输出功能。

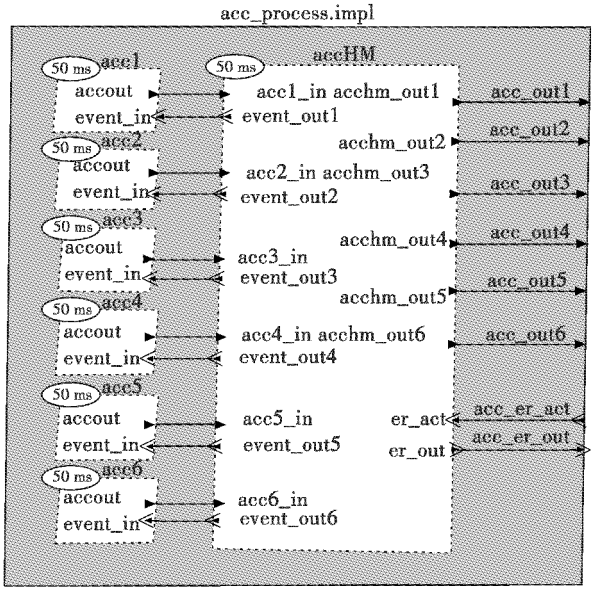

图3 加速度传感器单元 AADL 模型

Fig. 3 The AADL model of accelerometer unit

2.4 ADIRU 处理器单元 AADL 模型

ADIRU 处理器单元中包含了 4 个 ADIRU 处理器。每个处理器接收各个加速度传感器输出的数据并计算得到飞机在其 3 坐标轴向的加速度值，计算式为

$$\begin{cases} a_x = -0.19a_1 + 0.19a_2 - 0.30a_3 - 0.49a_4 - 0.30a_5 + 0.11a_6 \\ a_y = -0.17a_1 - 0.43a_2 - 0.40a_3 + 0.08a_4 - 0.03a_5 - 0.34a_6 \\ a_z = +0.43a_1 + 0.17a_2 - 0.03a_3 + 0.08a_4 - 0.40a_5 - 0.34a_6 \end{cases} \quad (1)$$

其 AADL 模型如图 4 所示。处理器单元由一个 ARINC-653 进程(ADIRUp_process. impl)中的 4 个 thread 对象(solver1_th, solver2_th, solver3_th 和 solver4_th)表示，每个 thread 对象实现一个 ADIRU 处理器功能。

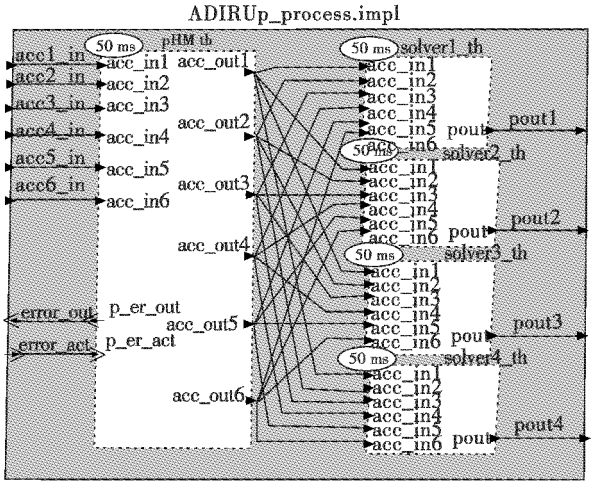

图4 ADIRU 处理器单元 AADL 模型

Fig. 4 The AADL model of processor unit

2.5 比较和显示单元 AADL 模型

比较和显示单元中包含 3 个比较器和 3 个显示器。每个比较器分别接收来自 ADIRU 处理器单元的 4 路数据输出，通过中值运算法则，得到 4 路数据的中值输出给显示器。

其 AADL 模型如图 5 所示，该单元用一个 ARINC-653 进程(voter_display_process. impl)中表示选择器功能的 3 个 thread 对象(voter1, voter2, voter3)和表示显示功能的另外 3 个 thread 对象(display1, display2, display3)表示。

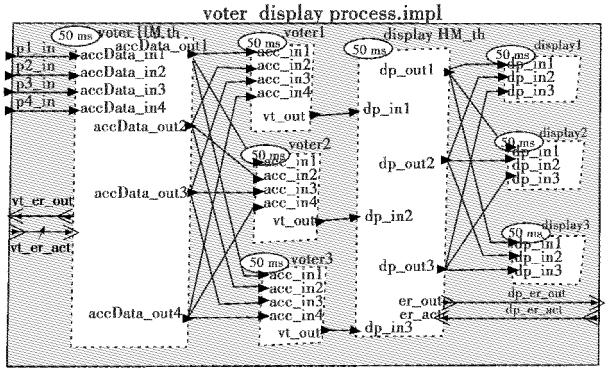

图5 比较和显示单元 AADL 模型

Fig. 5 The AADL model of voter and display unit

3 软件健康管理单元模型

模型中构成软件健康管理功能的网络分为两级结构：部件级健康管理(CLHM)和系统级健康管理(SLHM)。部件级健康管理位于 ADIRU 各功能单元中，用来检测部件内出现的数据异常并执行本地故障恢复；系统级健康管理能够检测各部件级健康管理单元产生的故障信息，执行故障诊断，确定故障源，并在系统范围内进行故障恢复。

3.1 部件级健康管理 AADL 模型

部件级健康管理功能由位于各功能单元中的本地健康管理模块实现，它们的 AADL 模型均用 thread 对象来表示。如图 3 中的 accHM 对象；图 4 中的 pHM_th 对象；图 5 中的 voter_HM_th 和 displayHM_th 对象。本地健康管理模块功能之一是实时监测各自输入端的数据流，如果检测到异常数据则产生相应的警告信息并报告至系统级软件健康管理单元。

监测的数据异常情况包括：1) BDI，数值过大、过小或变化过大；2) NDO，无数据输出；3) LDO，数据延迟输出；4) IDO，无效数据输出；5) VFI，数据的时间戳超过规定值。

本地健康管理模块的另一功能是管理各自功能单元的运行状态和执行故障恢复命令。在模型中使用状态机表示，如图 6 所示。当检测到单元异常后，向系统输出故障代码，并转入故障工作状态；接收到系统恢复指令后，执行相应的恢复动作。

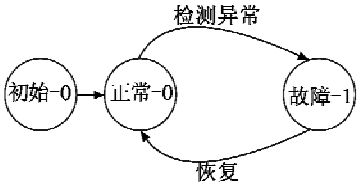

图 6　本地健康管理状态机

Fig. 6　The state machine of CLHM

3.2　系统级健康管理 AADL 模型

该单元用来执行系统级健康管理功能，由故障检测、故障诊断和故障处理三个模块组成。其 AADL 模型如图 7 所示，其中包含 3 个 thread 对象，分别实现各组成模块功能。

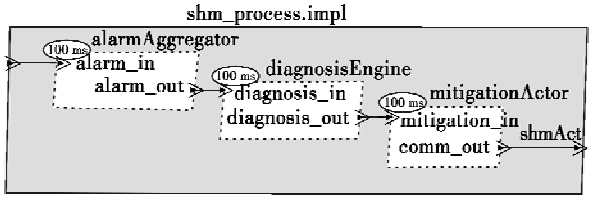

图 7　健康管理单元 AADL 模型

Fig. 7　The AADL model of SHM unit

alarmAggregator 的输入端分别与模型中 3 个基本功能单元中的本地健康管理模块的故障输出端相连，接收各功能单元发出的故障信息，然后将故障位置和代码输出给 diagnosisEngine，由该模块进行故障诊断。该模块采用了基于 TFPG 模型的故障诊断算法[4-5]，系统的 TFPG 模型如图 8 所示。

由该算法确定出真实的故障源之后，将故障源代码输出给 mitigationActor 模块；该模块根据不同的故障源发出相应的故障恢复指令，相关功能单元接收到故障恢复指令后即执行相应的恢复动作，使系统从故障中恢复，继续正常运行。

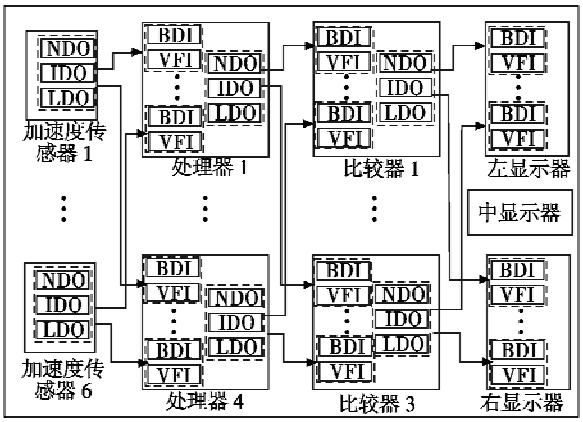

图 8　ADIRU TFPG 模型

Fig. 8　TFPG model of ADIRU

4　模型运行及结果

模型运行过程中故障的发生过程和真实故障发生过程相同。根据事故报告[6]，5#加速度传感器在 2001 年 6 月已经发生故障并一直输出异常的加速度值。这个故障被检测到，并且 5#加速度传感器的值被排除使用。在事故当日，发生了系统重启。之后，6#加速度传感器出现故障，此故障被检测到并且其输出值被排除使用。但是，重启后发生了错误，5#加速度传感器被重新作为有效数使用，因此产生了过大的加速度传感器计算值并输出到飞行控制计算机，导致了事故的发生。

ADIRU 模型通过 Ocarina 编译生成可执行代码后，可在安装于 CentOS 6.4 操作系统计算机平台中的能够运行 ARINC-653 应用程序的操作系统 POK 中运行，运行结果列于表 1 中，从中可以看出故障的发生、检测、诊断以及恢复过程。

表 1　ADIRU 模型运行结果

Table 1　The consequence of the model running

顺序	单元	事件	说明
8	5#加速度传感器	5#加速度传感器故障	5#加速度传感器已经故障，且被处理器排除使用
28	ADIRU 处理器	系统重启	发生了未知重启事件。之后，系统初始化过程中，没有读取 5#加速度传感器的故障信息。此时处理器仍使用 1#,2#,3#,4#和 6#加速度传感器数据。
36	6#加速度传感器	6#加速度传感器输出过大数值	6#加速度传感器出现故障，该错误被 CLHM 检测到
47	6#加速度传感器	6#加速度传感器向健康管理单元发送错误信息	6#加速度传感器故障信息报告至 SHM 单元
59	故障检测模块	故障检测模块接收到错误信息	接收到该故障信息
60	故障处理模块	故障处理模块发送恢复指令	故障诊断之后，发送重置命令给 6#加速度传感器
67	6#加速度传感器	6#加速度传感器 CLHM 接收到恢复指令	加速度传感器单元的 CLHM 接收到该命令，然后发送重置命令给 6#加速度传感器
76	6#加速度传感器	6#加速度传感器正在初始化 6#加速度传感器初始化完成	6#加速度传感器接收到重置命令，然后进行初始化

续表1
Continued from Table 1

顺序	单元	事件	说明
86	6#加速度传感器	6#加速度传感器输出过大数值	重置后,6#加速度传感器仍然发送过大的加速度值
97	6#加速度传感器CLHM	6#加速度传感器CLHM向健康管理单元发送错误信息	错误信息报告至SHM单元
109	故障检测模块	故障检测模块接收到错误信息	接收到该故障信息
110	故障处理模块	6#加速度传感器故障 故障处理模块向ADIRU处理器单元发送故障恢复指令	诊断之后,认为6#加速度传感器永久故障,发送故障恢复命令给处理器单元,排除使用6#加速度传感器输出的数值
118	ADIRU处理器单元CLHM	ADIRU处理器单元接收到故障恢复指令 抛弃6#加速度传感器的输出数据	处理器单元的CLHM接收到该故障恢复命令,6#加速度传感器的输出值被排除使用
128	5#加速度传感器	5#加速度传感器重新使用	此时5#加速度传感器的输出值被处理器单元重新使用
135	5#加速度传感器	5#加速度传感器输出过大数值	5#加速度传感器输出的错误数据被处理器单元的CLHM检测到
147	5#加速度传感器CLHM	5#加速度传感器CLHM向健康管理单元发送错误信息	错误信息报告至SHM单元
159	故障检测模块	故障检测模块接收到错误信息	接收到该故障信息
160	故障处理模块	故障处理模块发送恢复指令	故障诊断之后,发送重置命令给5#加速度传感器
167	5#加速度传感器CLHM	5#加速度传感器CLHM接收到恢复指令	加速度传感器单元的CLHM接收到该命令,然后发送重置命令给5#加速度传感器
175	5#加速度传感器	5#加速度传感器正在初始化 5#加速度传感器初始化完成	5#加速度传感器接收到重置命令,然后进行初始化
179	故障检测模块	故障检测模块接收到错误信息	重置后仍然接收到来自5#加速度传感器的故障信息
180	故障处理模块	5#加速度传感器故障 故障处理模块向ADIRU处理器单元发送故障恢复指令	诊断之后,认为5#加速度传感器永久故障,发送故障恢复命令给处理器单元,排除使用5#加速度传感器输出的数据
188	ADIRU处理器单元CLHM	ADIRU处理器单元接收到故障恢复指令抛弃5#加速度传感器的数据	处理器单元的CLHM接收到该故障恢复命令,5#加速度传感器的输出值被排除使用

5 结论

本文引入了软件健康管理方法来解决在嵌入系统软件运行过程中可能出现的运行中错误,并在为2005年马来西亚航空公司124航班上出现软件故障的ADIRU所建立的AADL模型中进行了验证;模型通过POK和Ocarina在计算机平台上编译运行,模拟运行结果说明软件健康管理能够有效检测软件运行中错误,使系统从错误中恢复正常运行。此方法能够应用于高安全性要求的嵌入系统软件设计,用来增强嵌入系统软件的运行可靠性。

参考文献

[1] FEILER P H, GLUCH D P. Model-based engineering with AADL[M]. Boston: Addison-Wesley, 2012: 196-204.

[2] MCINTYRE M D W, SEBRING D L. Integrated fault-tolerant air data inertial reference system: US, 5297052[P]. 1994-03-22.

[3] ARINC Aeronautical Radio. ARINC 653P1-3 Avionics application software standard interface Part 1, Required services [S]. [S. l.]: ARINC, 2010.

[4] ABDELWAHED S, KARSAI G. Notions of diagnosability for timed failure propagation graphs[C]//IEEE Systems Readiness Technology, Anaheim: IEEE Conference, 2006: 643-648.

[5] ABDELWAHED S, KARSAI G, MAHADEVAN N, et al. Practical considerations in systems diagnosis using timed failure propagation graph models[J]. IEEE Transactions on Instrumentation and Measurement, 2009, 58(2): 240-247.

[6] Aviation Occurrence Report. In-flight upset event 240 km north-west of Perth, WABoeing Company 777-200, 9M-MRG[R]. Canberra: Australian Transport Safety Bureau, 2005.