

# 模型覆盖率在嵌入式软件开发中的应用

马金梭 张荣华 席厚金

爱斯特尔技术公司, 上海市郭守敬路 498 号浦东软件园 12-401, 201203

## 摘要:

在嵌入式软件开发领域,随着对软件质量和开发效率的要求越来越高,传统的软件开发方式已经不能够胜任。基于模型的开发方式通过引入直观、准确的模型,提供大量新的技术,从根本上改善了软件开发活动。其中模型覆盖率分析技术的采用,使开发人员可以在开发早期完成高质量的验证活动,既保证了软件的质量,又提高了开发效率。本文介绍了模型覆盖率的基本思想,并给了一个模型覆盖率分析的应用实例。

**关键词:** 基于代码开发, 基于模型开发, 模型覆盖率, SCADE

## Application of Model Test Coverage in Embedded Software development

Jinsuo Ma, Ronghua Zhang, Houjin Xi

Esterel Technologies, 12-401, SPSP, 498, GuoShoujing Road, Shanghai

## Abstract

Traditional software development way, which focuses on source code, could not meet the more and more compact requirement of high quality and short development cycle of embedded software. Model based development is thus introduced and improves embedded software development activities by employing visual, accurate model and lots of new techniques based model. Among these techniques, Model Test Coverage helps the developer to verify development work efficiently and quantitatively in early stage of the development cycle, and guarantees software quality while saving development time. In this paper, Model Test Coverage is introduced and an example is given to demonstrate Model Test Coverage.

**Key words:** code based development, model based development, model test coverage, SCADE

## 1、背景

嵌入式系统已经覆盖了从工业系统到我们日常生活中普通应用的几乎所有的领域,而且其规模及复杂程度还在不断地提高。特别是随着嵌入式系统在航空航天、国防、交通运输、能源等关键领域的应用日益广泛,嵌入式系统开发面临着越来越苛刻的挑战,主要体现在开发周期要求越来越短,而对质量的要求越来越高。

在硬件处理能力飞速发展的今天,软件设计却面临越来越多的挑战。随着嵌入式系统的功能越来越多地由软件实现,嵌入系统开发的瓶颈主要集中在软件上,软件设计规模上不去,设计质量难于控制,设计周期无限延长……由于嵌入式软件引起的应用开发延期事件以及导致严重后果的应用事故屡见不鲜<sup>[1]</sup>。究其原因,在于嵌入式软件的开发技术不能适应嵌入式应用快速发展的需求。

目前,以程序员的大脑“生产”软件,并通过键盘输入代码,“以代码为中心”的基于代码开发方式仍然是嵌入式软件主要的开发形式。基于代码开发方式在面对需求快速变化,规模日益增长,复杂程度越来越高的嵌入式软件开发时显得力不从心。为了解决这一问题,一些新的软件开发方法和技术被逐步引入到嵌入式软件开发领域,并且已经有了一系列成功

的实践。其中，基于模型开发最为引人注目。基于模型开发是新一代软件开发技术，全面改善了基于代码开发的弊端，为嵌入式软件开发提供了新的手段。在基于模型开发技术的基础上，人们又发展了一些新的软件开发新技术，如，模型覆盖率分析，代码自动生成，形式化验证等，借助这些技术，将嵌入式软件开发活动的质量提高到一个新的水平。下面重点介绍模型覆盖率及其在嵌入软件开发中的作用。

## 2、基于模型开发

### 2.1 基于代码开发

传统的基于代码开发方式以代码开发为中心，需求分析、软件设计为软件编码做准备，代码编写完成后进行单元测试。长期以来，代码编写一直是开发过程的主要阶段，占用了大量的时间和人力。随着人们对软件质量的重视，单元测试在开发过程中的地位日益重要，在目前的开发流程中，越来越多的力量投入到代码的单元测试中<sup>[2]</sup>。无论是哪种情况，代码都是开发过程的中心：代码编写，代码走查，代码测试……

基于代码开发方式的缺点非常明显：

- 开发效率低下

由于代码的抽象程度高，不直观，与软件需求完全脱节（编码工作基于软件设计文档而不是软件需求），因此开发人员不得不花大量的时间和技术细节上，代码的修改也很耗时。另外，文档的编写和维护、代码测试也需要花费大量的时间。

- 质量没有保证

由于软件代码由程序员手工编写，程序员的能力、态度及状态制约了软件的质量。而且随着软件规模、复杂度的不断增加，团队开发情况的出现，代码集成也变得越来越困难。

出现这些缺点的主要原因在于，在基于代码开发方式下，开发工作的早期（需求分析、软件设计）完全依赖于人工工作，其成果以有歧义的自然语言或图表方式描述，无法进行有效的交流和验证，从而成为错误引入的重灾区；而当具有动态行为的代码出现后，对代码的测试是一种事后监督的、费时费力的做法，只能尽可能地发现错误，而无法有效地避免或者是排除错误。

### 2.2 基于模型开发

随着人们对软件认识的逐渐加深及实践中的探索，模型取代代码成为了软件开发流程的中心。直观模型符号被用于需求分析及以软件设计过程，表达分析人员对需求的理解及细化。由于模型远比程序语言的抽象程度低，更易于理解和交流，减少了需求分析和软件设计过程中错误引入的机率，尤为重要，模型还可用于模拟仿真，表达其动态特性，从而使开发人员在开发的早期能确定性地发现并排除错误。

一般来说，基于模型的开发方式具有以下几个特征：

- 形式化的图形符号

直观易懂的图形符号是模型的主要特征，也是模型优于代码的主要原因。借助于图形化的符号组成的模型，阅读者不仅能快速获取模型所表达的含义，而且不会因知识背景的不同产生歧义。

- 模型可运行

可运行的模型为开发人员了解模型的动态行为提供了方便，可以验证设计与需求是否一致。

- 代码自动生成

利用代码生成器，模型可以直接转换成相应的程序语言代码，无需手工编码工作。

在嵌入式软件开发领域，由于有成熟的理论支持以及大量的经验总结，基于模型开发在嵌入式领域中的应用远比其他领域成熟，文档自动生成、模型覆盖率分析、形式验证等或强大的功能都引入到开发过程中，极大地提高了开发活动的质量。

### 3、模型覆盖率

#### 3.1 代码覆盖率

基于代码开发方式中，在编码工作完成以前，开发工作的成果以自然语言及图表方式表达，对其验证完全依赖人工的走查、分析，不仅效率低、效果差，而且无法定量评价验证工作完成得是否足够。

在编码工作完成后，通过单元测试验证手工编码工作，代码运行的结果可以用于确认代码在相应测试用例的驱动下动态行为正确与否，从而找出编码过程中引入的错误。在绝大多数情况下，无法通过遍历所有的测试用例对代码进行完全地测试，只能挑选部分有意义的测试用例进行测试。一般根据对特定代码结构的覆盖来判定测试用例是否合适或足够。常见的覆盖率准则有语句覆盖、分支覆盖或判定覆盖。在一些高安全性应用开发中，如航空软件中，采用 MC/DC（修正条件/判定覆盖）准则。

通过代码的测试可以发现编码中引入的错误，通过代码覆盖率分析可以评估测试是否达到某种完备程度，因此代码覆盖率分析在基于代码开发方式中是最主要的软件质量保证手段。但由于代码出现已经是开发的中后期，此时发现错误后，纠正错误的成本高，而且主要是找出手工编码阶段引入的错误，对于发现开发早期引入的错误贡献有限。

#### 3.2 模型覆盖率

对应于基于代码开发方式，模型为嵌入式软件的验证工作带来了巨大的革新。

模型直接来自于需求的细化和具体化，可以认为模型是需求的另一种无歧义的表达形式。由于模型能运行，利用来自于需求的测试用例驱动模型，可以通过测试的方式验证模型的正确性，即与需求是否相符。与代码的测试类似，测试工作无法穷尽所有测试用例，因此需要借助模型覆盖率来评价测试工作的质量。

模型不同于代码，模型描述的是功能，而代码则是描述功能的实现。在定义模型的覆盖率准则时，除了要考虑模型的结构，捕获模型的动作（在运行过程中激活与否）外，还应考虑模型功能的覆盖。

最基本的模型覆盖率准则可参考代码的覆盖率准则，定义模型分支覆盖、模型判定覆盖、模型 MC/DC 覆盖等，通过分析测试用例是否覆盖不同的模型结构，从而评价测试工作的完备程度。

针对特定应用下的模型，用户可以根据其经验及对应用的理解，将某些功能特征或它们的组合定义为覆盖率准则，实现更为高效的测试。

#### 3.3 模型覆盖率的作用

模型覆盖率在基于模型开发方式中的地位与代码覆盖率在传统开发方式中的地位类似，都是主要的质量保证手段。但模型覆盖率还有着更重要的意义，它的引入，从根本上改变了

传统开发方式中测试工作太晚的弊端，从而为软件质量提供了保障。

模型覆盖率分析带来的一系列好处有：

- 发现需求分析和软件设计工作中引入的错误

借助于模型覆盖率分析，可以对模型按照覆盖率准则进行充分的测试，尽可能地发现模型中的错误，即在需求分析和软件设计中引入的错误。由于测试的结果是确定性的，它远比完全依赖人工走查、分析的验证方式来得高效和准确。

- 发现需求中的错误

进行模型覆盖率分析时，测试用例来自于需求。当结果不正确，而设计正确时，导致结果与预期不一致的原因就是需求不正确。因此，模型覆盖率分析可以帮助找出需求中的冲突、遗漏、错误。

- 在开发的早期完成高质量的验证工作

采用模型进行需求分析和软件设计，可以有效地减少错误的引入，而模型的仿真运行帮助准确地找出错误。在引入了模型覆盖率分析后，覆盖率准则可以指导测试用例开发，还可通过覆盖率分析评价测试的完备性，在开发的早期完成高质量的验证工作，尽可能在进入到下一阶段工作前将错误排除掉。

- 省去代码单元测试工作

由于采用了模型覆盖率分析，模型的正确性得到了足够的保证。代码生成是由机器自动完成的，完全避免了手工编码引入的错误，因此通过代码覆盖率分析来验证代码是否正确的意义不大，特别是当代码生成器通过了认证，代码的测试工作可以完全省略。

总的来说，模型覆盖率不是简单的代码覆盖率“升级”，它改变了嵌入式软件开发流程的重心，从而在多个方面对软件质量保证作出了贡献。

## 4、实现实例

基于模型的开发工具近些年来发展很快，在嵌入式领域主要有 SCADE, Rhapsody, TAU、RoseRT 等，其中 SCADE 因其满足关键应用领域的高安全要求，简单易用等特性，近年来在国内航空航天等高安全性领域被广泛采用。SCADE 还是少数支持模型覆盖率分析功能的嵌入式开发工具。下面以 SCADE 模型为例，介绍模型覆盖率分析的具体实现。

### 4.1 SCADE 简介

SCADE<sup>[3]</sup> (Safety-Critical Application Development Environment) 是一套针对高安全性应用的嵌入式软件开发环境，模型基于严格的数学理论，覆盖了从需求到最终实现的整个开发流程，它的代码生成器还通过了航空业界内最为严格的 DO-178B A 级认证，其主要功能有：需求管理、建模、模型仿真、模型覆盖率分析、形式验证、系统原型生成、代码自动生成、定点运算、编译器验证、文档自动生成、配置管理、RTOS 桥接等。SCADE 适用于不同软、硬件平台，是一套通用的嵌入式软件开发平台。

### 4.2 在 SCADE 中进行模型覆盖率分析

#### 4.2.1 限幅积分器模型

采用 SCADE 开发一个限幅积分器，积分器的数学公式如下：

$$y^n = y^{n-1} + \Delta t \bullet (u^n + u^{n-1}) / 2$$

当积分值大于上限（值为 10.0）时，结果 y 取上限；当积分值小于下限（值为-10.0）

时，结果  $y$  取下限；当积分值落在  $-10.0$  至  $10.0$  之间时，取积分值。第一个时间步长的结果取  $0.0$ ；积分器可通过复位信号复位，复位时结果取  $0.0$ ；时间步长取  $0.1$ 。

SCADE 模型如下，其中输出为  $Y$ ，输入为  $U$ ，TimeCycle，LowLimit，HiLimit，Reset（考虑到通用性，将时间步长，下限和上限作为变量输入，而没有采用常数或常量）。

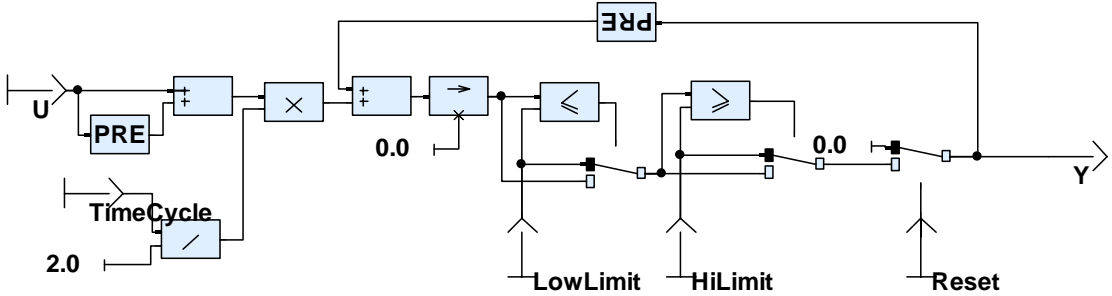




图 1 限幅积分器的 SCADE 模型

上述模型中，**PRE** 表示取前一时间步长的值； 表示初始化，仅在第一个时间步长动作； 为选择操作“IF…ELSE…”；其它符号直观地表示了加法、乘法、除法、比较等基本操作；连线表示数据的流动。

#### 4.2.2 模型覆盖率分析<sup>[4]</sup>

要进行模型覆盖率分析，先要定义覆盖率准则，这里选用系统预定义的模型 MC/DC 准则（用户也可以根据需要自定义覆盖率准则），然后进行模型插装，获得一个能记录模型特征的新模型（插装由系统自动完成）。对插装后的模型进行模拟仿真，即可采集覆盖数据，进行覆盖率分析。

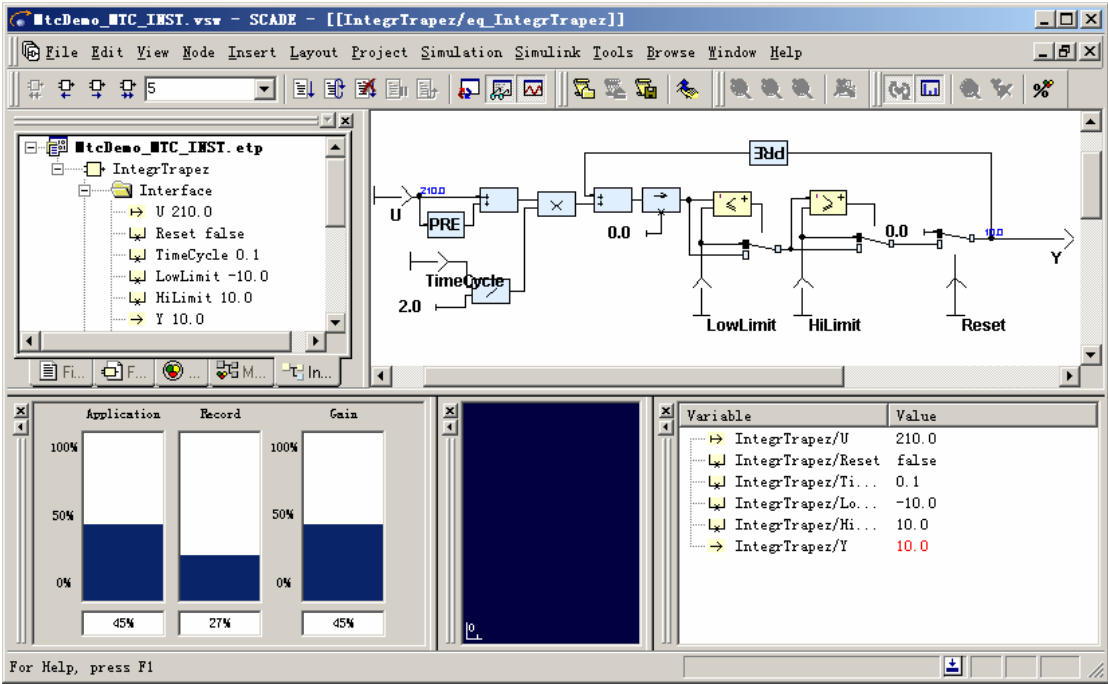


图 2 模型覆盖率数据采集

测试用例可以由用户构造，或者来自于已有的运行数据、原型设计数据。取一组数据驱动模型仿真，并收集覆盖率数，可以发现该模型的 11 个特征只能覆盖到

10 个，其中 MTC1\_IFTHENELSE\_I 3 的特征始终无法达到完全覆盖。在模型中定位到该模型节点，并显示覆盖情况，即可知道原因，Reset 的值没有取“True”导致的（见图 3）。由于实际系统正常运行中，该积分器始终不曾被复位，因此数据中没有积分器复位的测试用例。补充一个复位的测试用例（即 Reset 取“True”值）即可达到 100%覆盖。如果在该积分器的实际应用中，复位操作永远不会发生，则说明需求与实际情况不符，即通过覆盖率分析可以找出需求中的问题。如果需求没问题，而是在某一部分应用中不会出现复位情况，则可以通过覆盖率调整的方式，通过添加覆盖率数据“人为覆盖”该特征，并注明原因，从而达到 100%覆盖。

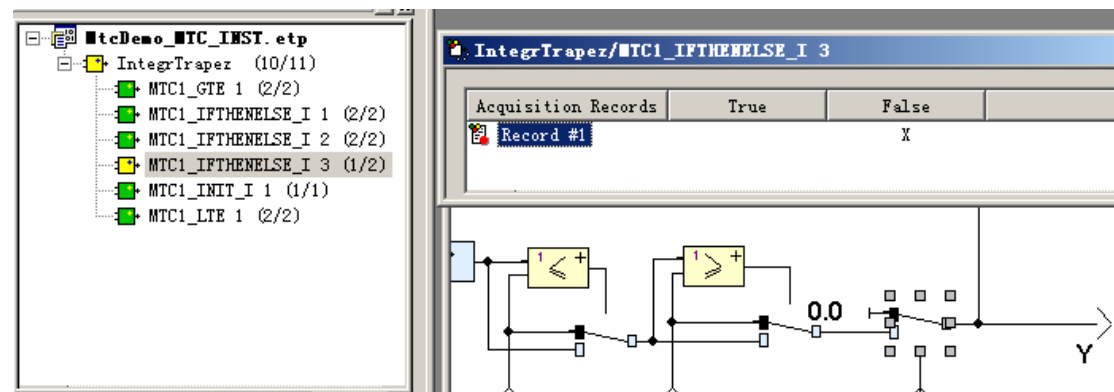


图 3 模型覆盖率数据分析

达到 100%覆盖说明测试工作达到了要求，测试“足够”充分，模型“足够”正确。模型覆盖率分析完成后，还可生成自定义格式覆盖率分析文档。

## 5、总结

基于模型开发是软件开发技术的重大进步，是未来软件开发技术的趋势。在采用模型的基础上，涌现了一批新的软件开发技术，模型覆盖率分析就是其中最重要的技术之一，它改变了传统开发方式中早期工作不能有效验证的弊端，从开发流程、开发手段等多方面为改善软件质量，提高开发效率提供了保证。

## 6、参考文献

- [1] Bruce Powel Douglass 著，柳翔等译，《嵌入式与实时系统开发》，北京：机械工业出版社，2005 年，P60~61。
- [2] 林宁，孟庆余，《软件测试实用指南》，北京：清华大学出版社，2004 年，P9。
- [3] Jean-Louis Camus and Bernard Dion. *Efficient Development of Airborne Software with SCADE Suite*[M]. Esterel Technologies, 2003.
- [4] 爱斯特尔技术公司，<http://www.esterel-technologies.com/products/scade-suite/model-test-coverage>，2006.