



## 使用Rhapsody软件框架和UML的实时系统开发

来源：单片机及嵌入式系统应用 作者：孙 强 张振华

**摘要：**UML已成为复杂系统建模的工业标准，并可借助代码自动生成工具实现从分析到编码的开发过程自动化。而实时系统的行为特征可以很好地用UML的状态机表示，从而成为自动代码生成、测试、分析和校验的最佳选择。Ilogix公司的Rhapsody是一个基于UML支持模型驱动开发的CASE工具，它使用一套为实时系统量身打造的软件框架实现实时系统软件的开发自动化。

**关键词：**实时系统 UML Rhapsody 状态机

### 引言

面向对象的软件设计方法推崇抽象、可重用且易维护的设计模式。其中，一个普遍采用的做法是使用预定义的框架构建复杂的软件系统。在面向对象领域中，框架是指为某一具体应用领域提供解决方案的类集。程序员使用继承、聚合等技术定制框架并将其用于具体的应用程序设计。基于框架的软件开发具有如下优点：

- \*由于重用了框架提供的代码，目标系统不必从头写起；
- \*规范的框架结构能够指导目标系统的设计；
- \*框架的设计是开放的，因此可以通过子类化等技术定制和扩展。

本文介绍Rhapsody的实时代码生成策略。Ilogix公司提供的Rhapsody是一个基于UML支持实时嵌入式系统全过程面向对象开发的CASE工具。Rhapsody基于一个内置的实时系统软件框架，OXY生成C/C++代码。

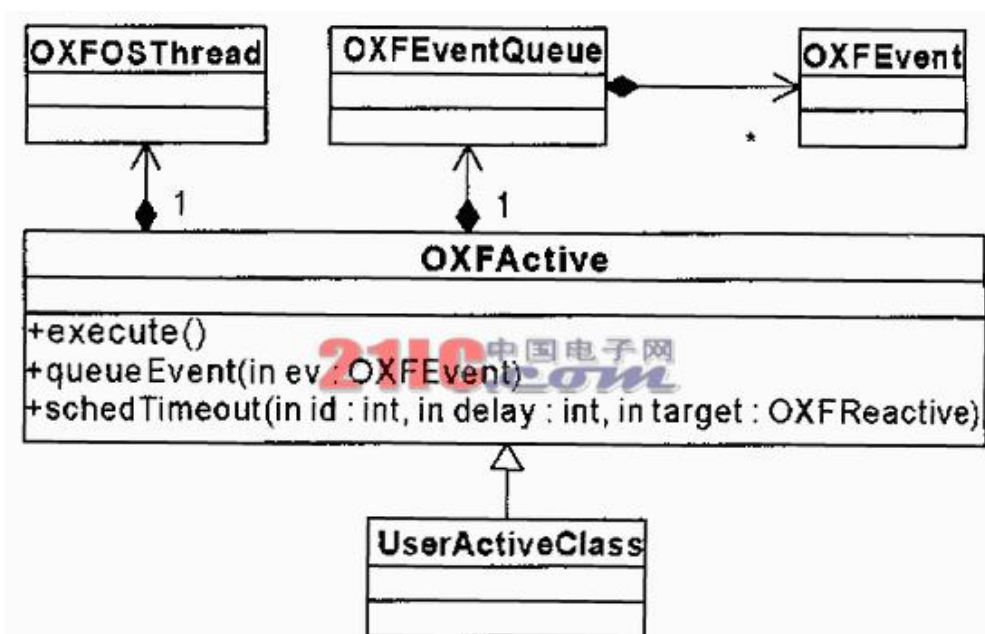


图1 活动类及相关框架元素

## 1 Rhapsody的软件框架

Rhapsody的OXF框架专为实时嵌入式系统而设计，它由一系列UML类组成。这些类相互协作，组成了一个结构严整、跨实时操作系统平台的实时嵌入式软件架构。

### (1) 活动类

活动对象（Active Object）在UML中被定义为：拥有一个线程并能够发起控制活动的对象。Rhapsody给这个一般定义赋予了具体含义，如图1所示。

活动类（Active Classes）在框架中以OXFActive表示，它拥有一个执行线程和一个事件队列，并提供事件调度功能。用户活动类从OXFActive继承。

该活动类包装了实时系统中的事件调度器。Execute方法是一个无限循环。在循环中，事件被不断地从队列中提取出来，发往对应的目标对象。QueueEvent和schedTimeout方法向队列注入新的事件。

### (2) 响应类

响应类（Reactive Classes）是能够对事件做出反应的框架类，即事件消费者。它在Rhapsody的框架中以OXFReactive表示，用户响应类从OXFReactive继承，如图2所示。



可以影响对象的行为及状态。在Rhapsody中，活动类负责事件调度。事件既可以异步（活动类将事件投递给目标对象之后立即返回，目标对象使用自己的线程完成对事件的响应），也可以是同步的（活动类投递事件之后必须等待目标对象处理完成才能返回）。操作代表类本身提供的服务和功能。操作调用总是同步的，即操作总是运行在调用者线程之内。Rhapsody的框架定义了两种事件：

### ①信号事件。

信号事件（Signal Events）是最常用的事件，它代表对象之间传递的一种异步激励信号。在Rhapsody的框架中，它以OXFEvent表示，所有的用户事件类都从OXFEvent继承。

信号事件有自己的参数（以其属性表示），并且和一个响应对象关联。事件管理器依据事件的响应对象判断事件的接收者。

### ②定时事件

定时事件（Timeout Events）在Rhapsody中以OXFTimeout表示。它在OXF框架中的位置如图3所示。

事件管理器（活动对象）的schedTimeout方法负责定时事件的构造和调度。其执行过程如图4所示。



图 3 定时事件及相关框架元素

图3 定时事件及相关框架元素

事件管理器创建一个OXFTimeout对象并将其交由一个定时管理器管理。定时管理器可以管理多个定时事件，它运行于自己的独立线程内且聚合了一个定时器对象。定时器对象每间隔固定的时间片会向定时管理器发通知信号，这时定时管理器检查它管理的定时事件集合中是否有定时事件到时，并将到时的定时事件添加到事件管理器（活动对象）的事件队列中（通过调用queueEvent），之后事件管理器再将这些定时事件投递给对应的响应对象。定时事件本身是被动的，它不包含定时器。

#### ( 4 ) 状态机

Rhapsody支持UML状态机 ( State Machines ) , 包括层次状态分解、带参事件、定时事件、伪状态、完成转移、入口和出口动作等功能。Rhapsody也包含了UML中定义的异步事件处理模型: 带有状态图的类都是响应类, 因此都与一个事件管理器 ( 活动类 ) 关联。事件管理器使用一个队列缓冲收到的事件, 然后将事件投递给响应对象。响应对象按照状态图处理收到的事件。

OXF框架使用OXFEvent Consumer、OXFState、OXFLeafState等实现UML状态机的语义。限于篇幅, 本文对此不作介绍。

#### ( 5 ) 并发控制

Rhapsody的OXF框架使用如下两个类实现对实时系统资源的并发控制 ( Concurrency Control ) 。

\*OXFOSMutex: 包装了实时操作系统的互斥量对象, 支持lock ( ) 和free()操作。互斥量用于管理独占性资源。

\*OXFOSSemaphore: 包装了实时操作系统的信号量对象, 支持wait()和signal()操作。信号量用于管理共享资源。

UML类的方法可以声明为guarded。这时类被称为protected。protected类用于对独占性资源进行建模, 在任何时刻只可有一份guarded方法执行, OXF框架以OXFProtected定义UML protected类, 它充当所有用户protected类的父类。

#### ( 6 ) 模型的调试及分析

Rhapsody允许用户单步执行状态图和顺序图以验证系统的功能和逻辑。系统执行过程中, 对象的方法调用和状态改变都可以在模型图上以动画的形式表现出来。

除了功能和逻辑上的正确性之外, 实时系统还对系统的响应时间有严格的要求。对实时系统响应时间的测试称为调度性分析, 主要使用两种方法。

①经验法: 给系统注入实时测试数据, 然后测量系统的相应性能。

②理论分析: 为系统模块设定足够的时间信息, 然后用数学方法计算系统的整体相应性能。此方法通常需借助专用的软件工具完成。



图4 定时事件的调度和投递

图4 定时事件的调度和投递

Rhapsody支持基于经验的调度性分析。用户需要为对象方法设定估计的执行时间，并编写一个驱动脚本或状态图以模拟系统的触发事件。驱动脚本和状态图启动之后，Rhapsody会按照设定的时间参数和程序逻辑运行模型并输出详细的执行过程报表。用户通过检查执行报表判断系统是否满足要求的响应性能。Rhapsody也允许用户在目标硬件系统运行驱动脚本或状态图，以获得最准确的执行时间报表。

## 结语

实时嵌入式系统应用的开发越来越复杂，因此有必要对嵌入式项目实施严格的软件工程管理。UML已成为复杂系统建模的工业标准，其状态机模型与实时嵌入式系统极好地对应，且便于借助CASE工具实现代码自动生成，能够极大地提高实时嵌入式软件的开发效率和项目可管理性。

Ilogix公司的Rhapsody工具便是一种支持实时UML标准的嵌入式系统软件工具，它基于内置的OXF框架实现从系统的分析、设计到代码自动生成的开发过程自动化。OXF框架是面向对象和跨RTOS平台的，它很好地将UML的概念移植到实时嵌入式系统中来，为嵌入式系统的开发提供了清晰的结构以及可复用的软件模块，能够切实提高实时嵌入式软件的开发效率和可维护性。

版权所有：UML软件工程组织