

基于WF StateMachine的UML状态图动态构建与测试

孔令东

(盐城工学院信息工程学院, 江苏 盐城 224051)

摘 要: 采用UML分析与设计的业务信息系统, 业务流程经过层层抽象迭代, 缺乏一种透明的业务流程实现。WF提供了可视化的业务过程编程模型, 便于实现业务流程自动化, 在对比分析WF State Machine和UML状态图的基础上, 研究从UML状态图到WF State Machine业务流程映射关系, 选取UML中典型状态图, 依据一定的命名转换规则, 实现了从UML状态图分析设计到WF状态机业务过程可视化的构建, 完成了动态测试。

关键词: WF; State Machine; UML; 状态图

中图分类号: TP311.51 **文献标识码:** A

Dynamic Construction and Testing of UML State Diagrams Based on WF State Machine

KONG Lingdong

(School of Information Technology, Yancheng Institute of Technology, Yancheng 224051, China)

Abstract: There is a lack of transparent implementation for the business process systems analyzed and designed on the basis of UML when the business process is iterated one after another. WF provides a visualized business process programming model so as to automatize business process. By comparing and studying WF State Machine and UML state diagram, selecting typical state diagrams in UML and based on certain denominating principles, the visualized construction of business process between WF State Machine and UML state diagram is realized and the dynamic testing is completed.

Keywords: WF; State Machine; UML; state diagram

1 引言(Introduction)

Windows Workflow Foundation(WF)是微软推出的一个可扩展框架, 提供了工作流定义的图形化的工具和界面。WF可将业务逻辑作为规则或条件来实现, 作为工作流的基本组成。WF中状态机(State Machine)工作流提供的建模类型, 可以对事件驱动方式的工作流进行建模, State Machine活动包含状态和组成的状态机逻辑转换。可以为状态驱动的业务过程提供可视化的建模实现过程^[1-3]。UML提供业务过程建模的分析和设计工具, 用于企业信息系统的可视化业务流程描述与处理^[4,5], 在此过程中, 经历了的层层业务抽象, 最后到业务的实现, 一旦业务流程需要修改, 需要不断迭代完成, 缺少一种从业务建模到实现的透明转换过程, 实现所见即所得。UML中也包含了State Machine的语义描述, 便于对离散事件驱动的动态行为建模^[6-8]。

本文在研究UML典型状态图(State Diagram)样例的基础上, 结合WF的可视化实现, 探索从UML状态图建模到WF

State Machine编程模型的无缝转换, 快速构建便于测试的应用程序。

2 映射关系及典型示例(Mapping relationship and typical example)

2.1 从UML状态图到WF State Machine的对应关系分析

UML和WF在状态机的可视化建模及实现上具有紧密的关联性, 从相关概念的组成和含义几乎都一一对应, 这一点保证了无缝的转换。如表1所示WF和UML的State Machine相关组成及含义对比, 可以看出WF和UML中的State Machine都是事件驱动的建模, 相当于所有状态的容器。相比较而已, 在业务系统中, UML中偏重业务流程的分析设计, 而WF中更偏重基于工作流的业务实现。两者的State(状态)都可置于State Machine之中。两者的Transition(转换)和FinalState(终止状态)的概念也基本相同。有了相对应的组成, 这两个之间在可视化的建模中就有了相对应的可视化图形符号。

表1 WF和UML的State Machine相关组成对比

Tab.1 Comparison of compositions between WF and UML State Machine		
概念	WF	UML
State machine	StateMachine活动将工作流作为状态机来建模	用于对离散事件驱动的行为建模
State	state活动可以放置到StateMachine	一个状态用于对满足一定条件的StateMachine执行过程建模
Transition	表示两个状态之间的转换	表示对象的原状态在满足特定条件进入目标状态的过程
FinalState	用于创建终止状态机实例的state	用于表示一个特定的状态，表示在一个封闭区域行为的终止

如表2所示，WF和UML State Machine的映射关系及图形表示，两者都包括了开始状态、状态、状态转换、结束状态等基本符号组成，从符号的表达到信息的展示，都可以无缝对应转换，从而保证了可视化建模从分析到实现的一致性。表中基于两者的基本符号，初步构建了基本图形结构，可以看出，从形式语义上没有任何信息丢失，主要的区别在于UML中是静态的分析、设计，WF中可以直接动态的实现，在下面的基于WF的UML状态图构建中会详细分析。

表2 WF和UML State Machine的映射关系及图形表示

Tab.2 Mapping relationship and graphical representation between WF and UML State Machine		
符号类别	WF	UML
开始状态		
状态		
状态转换		
结束状态		
基本图形结构		

2.2 典型的UML状态图示例

为更好的说明基于WF的UML状态图构建和实现，选取了Rational Rose中典型的状态图，以便于深入的转换说明。如图1所示机器人自动转换状态图，在这个示例中，显示了机器人在转换过程中涉及的各种状态，包含了典型的开始状态，结束状态、嵌入状态、自身转换状态、历史状态，以及

各种动作活动(入口动作、出口动作等)、监护条件和触发等。

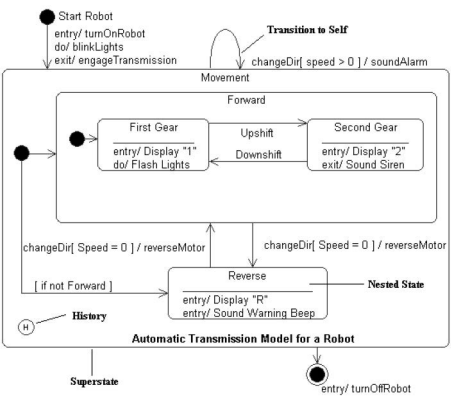


图1 机器人自动转换状态图

Fig.1 State diagram of automatic transmission for a robot

3 转换构建(Convert construction)

依据前面的映射关系和命名规则，如图2所示，基于WF的UML机器人自动转换状态图顶层映射模型，采用分层逐步迭代的思想，清晰地实现了顶层模型的映射。由图中可以看出，UML典型示例中的Start、Movement、FinalState、nestedstates等状态，以及相关的变迁、入口动作、监护条件都信息无损的映射到WF的StateMachine_Automatic Transmission Model for a Robot上。其中UML中Movement状态的Transition to Self变迁映射成toSelf变迁，从Movement到FinalState的变迁映射成toFinal变迁。由于UML中Movement状态是一个复合状态，需进一步的分层构建。

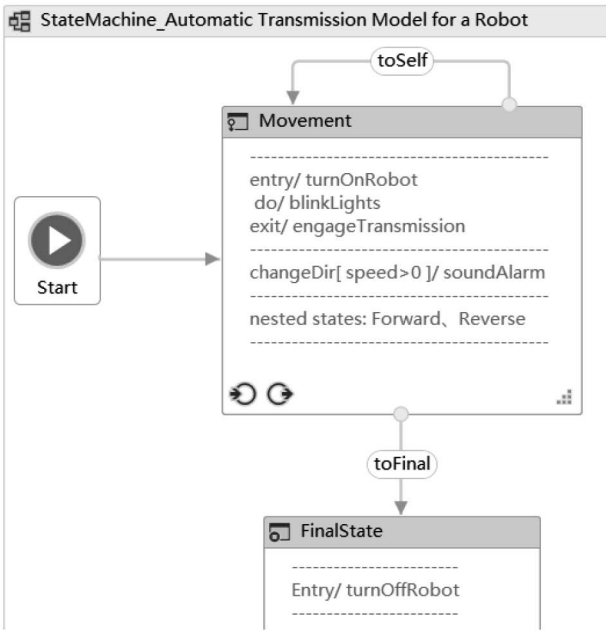


图2 基于WF的机器人自动转换状态图顶层映射模型

Fig.2 Top mapping model of state diagram of automatic transmission for a robot based on WF

如图3所示StateMachine_Movement细化模型, UML中的Movement复合状态, 被构建映射为StateMachine_Movement状态机, 嵌入的Start、History、Forward、Reverse置于其内, 相关的变迁、入口动作、监护条件都信息无损的映射。由于UML中Forward状态也是一个复合状态, 也需要进一步的分层构建。

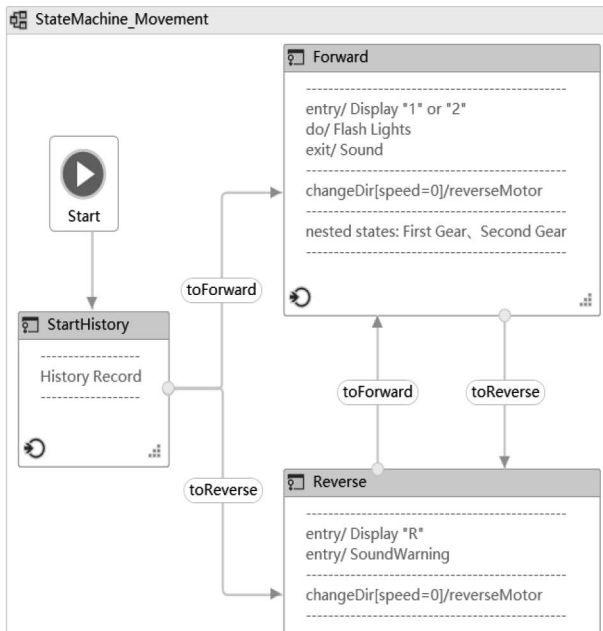


图3 StateMachine_Movement细化模型

Fig.3 Refined model of StateMachine_Movement

如图4StateMachine_Forward细化模型, UML中Forward复合状态被映射为StateMachine_Movement状态机, 嵌入的First Gear和Second Gear状态置于其内。

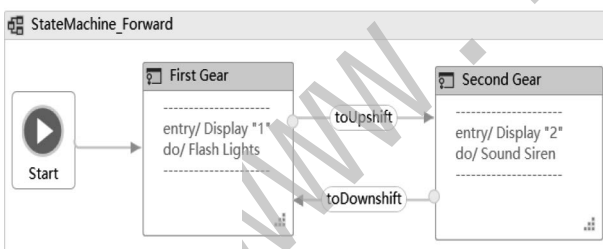


图4 StateMachine_Forward细化模型

Fig.4 Refined model of StateMachine_Forward

4 动态测试(Dynamic test)

在业务系统中, UML主要实现系统静态分析与设计, 经过如图2—图4所示的映射, 静态的UML状态图被映射成WF的状态机可视化动态建模, 所见即所得, 如图5基于WF的UML机器人自动转换状态图动态运行测试, 构建的WF模型在工作流的引擎牵引下就可以实现动态的运行测试, 一定程度上实现了业务过程的自动化, 实现了业务流程从UML分析设计到无缝实现。

```
turnOnRobot
doBlinkLight
engageTransmission
自我的转换速度还为零
转换到MoveMentNestedState
速度大于零, 将启动
forward 启动
upshift
```

图5 基于WF的机器人自动转换状态图动态测试

Fig.5 Dynamic test of state diagram of automatic transmission for a robot based on WF

5 结论(Conclusion)

对于复杂的系统, 从业务流程的分析设计到实现总是存在着信息的丢失和不一致性, 探索一种从业务分析到实现的无缝转换是必要的。本文在分析UML状态图和WF状态机结合优势及相关概念和符号表示的基础上, 选取Rational Rose中典型的状态图, 深入研究了基于WF的UML状态图的构建实现, 从UML状态图中简单状态之间的转换、入口动作, 到复合状态的映射, 显式地将业务逻辑映射为一定的规则或条件, 构建了基于WF的状态机可视化动态模型, 基于微软工作流引擎, 实现了状态驱动的业务过程建模自动化测试, 探索了从UML可视化建模分析、设计到WF可视化实现的透明转换具有实际的意义。

参考文献(References)

- [1] The programming model,samples,and tools of the Windows Workflow Foundation(WF).<https://docs.microsoft.com/en-us/dotnet/framework/windows-workflow-foundation/index>.
- [2] 钱维平,王坚.基于WF状态机工作流的日常工作管理系统的设计与实现[J].机电产品开发与创新,2015,28(1):81-83.
- [3] 冯倩,董丽丽.基于WF状态机工作流的研究与应用[J].电脑知识与技术,2008,4(8):2263-2265.
- [4] [美]Grady Booch,James Rumbaugh,Ivar Jacobson.UML用户指南(第二版)[M].北京:人民邮电出版社,2013:5-20.
- [5] Object Management Group:Unified Modeling Language.formal/2017-12-05,OMG UML(2017).
- [6] Alexander Knapp and Till Mossakowski.UML Interactions Meet State Machine—An Institutional Approach.7th Conference on Algebra and Coalgebra in Computer Science(CALCO 2017),2017,15:1-15.

(下转第7页)