

基于控制流的盒图动态建模与测试

孔令东

(盐城工学院信息工程学院, 江苏 盐城 224051)

✉ikld@outlook.com



摘 要: 为解决传统盒图静态过程描述缺少动态逻辑关系验证问题, 研究工作流程中控制流的可视化建模与实现; 通过分析盒图和控制流的转换关系, 将静态的盒图无缝地转换成可视化的动态控制流模型, 完成从盒图静态过程描述到控制流的可视化动态建模、实现、测试和验证; 实现快速构建复杂业务流程的可视化分析模型并提前验证, 解决了复杂业务过程模型在编码之前的逻辑验证, 在实际的建模中, 逻辑关系清晰, 便于验证。

关键词: 工作流; 盒图; 控制流; 动态建模与测试

中图分类号: TP311.51 **文献标识码:** A

Dynamic Modeling and Testing of N-S Diagram based on Control Flow

KONG Lingdong

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

✉ikld@outlook.com

Abstract: This paper studies visualization modeling and implementation of control flow as there is no dynamic logical relation verification in static process description of traditional N-S (Nassi Shneiderman) Diagram. Static N-S diagram is seamlessly transformed into visual dynamic model of control flow by analyzing the relationship between N-S diagram and control flow. Visual dynamic modeling, implementation, testing and verification method are completed from N-S static process description to control flow. The visual analysis model of complex business process can be quickly built and verified in advance. This study realizes the dynamic logic verification of complex business process model before coding. In actual modeling, the logical relationship is clear and easy to verify.

Keywords: workflow; N-S diagram; control flow; dynamic modeling and testing

1 引言(Introduction)

模块或者算法的详细设计, 描述了业务逻辑实现的方法和步骤, 其结果基本上决定了最终的程序代码的质量。N-S图也称为盒图, 作为过程详细设计的工具, 在算法和程序之间充当桥梁的角色, 可以指明控制流程、处理功能、数据组织及实现细节, 从而保证程序代码的正确性^[1-3]。然而, 对于复杂的业务逻辑, 在代码实现之前仅仅使用N-S图的静态过程描述, 无法快速完成逻辑关系的动态测试和验证, 期望寻求一种简明、快速、可视化的动态建模与测试方案。

WF(Workflow Foundation)是一种可视化的工作流编程引擎, 其过程控制流C-F(Control Flow), 可以直观地描述复杂的业务逻辑过程, 实现对流程控制的可视化建模。借助WF的过程工作流, 可以实现从N-S到C-F的无缝转换, 通过建模转

换不仅可以可视化地展示模块的业务逻辑, 而且可以动态地完成测试以及实现^[4-8]。如图1本文技术路线所示, 在算法分析到程序实现之间, 探索一种从N-S静态过程描述到C-F的可视化动态建模、测试和验证方式, 解决复杂业务过程模型在编码之前的逻辑验证, 实现快速构建复杂业务流程的可视化分析模型并提前验证, 具有实际意义。

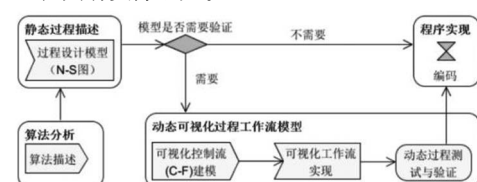


图1 技术路线

Fig.1 Technology roadmap

2 N-S图和控制流的转换(Transform between N-S diagram and control flow)

如图2N-S和C-F控制流之间的结构转换图所示, 图中我们典型地给出了顺序、选择、循环结构的图形表示和逻辑转换, 可以看出, N-S图的基本结构和C-F控制流在可视化静态建模上具有一一转换对应关系, 逻辑实现具有一致性。如图2(b)中所示, 对于选择条件结构, N-S图只能静态地描述条件判断任务, 而C-F控制流在显式地描述静态逻辑判断结构的同时, 可以方便地借助WF引擎, 动态的运行给出的“Condition”或参数, 快速的验证逻辑关系和算法; 同时, 在实际的算法描述中C-F控制流的规范性和可视化, 也带来建模的便利性和清晰性, 尤其对于较为复杂的算法和逻辑关系分析, 更具备优势。N-S优势在于静态的过程设计和描述, C-F的控制流可以在可视化描述业务过程的同时, 进一步的快速实现动态测试验证, 这将在后面的建模与集成应用中详细描述。

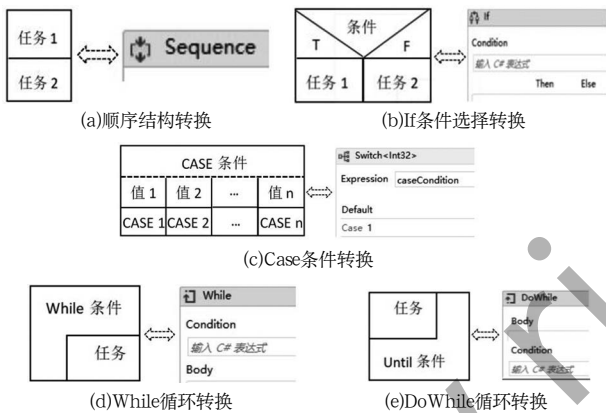


图2 N-S和C-F之间的结构转换图

Fig.2 Structure transform diagram between N-S and C-F

3 可视化建模与实现(Visualization modeling and implementing)

3.1 顺序结构的建模与实现

如图3所示C-F的顺序结构建模与实现, 在Sequence控制流中显式地完成了Sequence_Activity_1、Sequence_Activity_2、Sequence_Activity_3顺序过程建模, 其中Sequence_Activity_1给出了相关的顺序说明, Sequence_Activity_2活动任务中给出了可以动态测试的变量variable1, 可以完成模型的动态业务逻辑测试, Sequence_Activity_3显式地给出了测试的输出结果。在实际的测试中, 变量的输入可以方便的设置变量类型和作用域, 可视化的模型一目了然, 且逻辑关系清晰, 我们会在随后的集成测试中给出测试结果。相比较N-S图中的静态描述和直接的编码实现, C-F图给了我们快速可视化的逻辑展示。

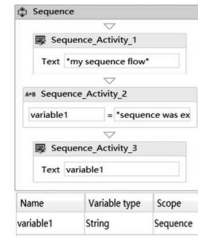


图3 C-F的顺序结构

Fig.3 Sequential structure of C-F

3.2 选择结构的建模与实现

如图4所示C-F的选择结构。其中4(a)图为IF选择结构, 判断条件“Condition”可以动态的设置, 根据判断的条件可以显式地执行“Then”和“Else”业务逻辑, 同时, 对于每一个“Then”和“Else”分支都可以显式地嵌入更复杂的业务逻辑或算法分析, 非常容易扩展。图4(b)为Switch-Case结构, 可以根据“Expression”条件表达式, 显式地确定“Case”执行的对应任务, 对于每个“Case”也同样可以嵌入复杂的过程描述。在细节之处, 我们可以看到C-F控制流中也默认给出了“Default”分支。在实际的建模中, 对于Case结构, C-F控制流以纵向分布, 多判断的扩展远比N-S图的横向布局清晰。



图4 C-F的选择结构

Fig.4 Select structure of C-F

3.3 循环结构的建模与实现

如图5所示C-F的循环结构, 其中图5(a)为While循环结构, 图5(b)为DoWhile循环结构, 可以直观显式地完成N-S的While循环条件和Until循环条件的建模, 同时可以根据不同的参数和变量完成动态测试, 我们在后面的集成动态测试中给出案例分析。此外WF控制流中也给出了ForEach的建模方法。



图5 C-F的循环结构

Fig.5 Loop structure of C-F

4 集成动态测试(Integrated dynamic testing)

前面我们分析了C-F可视化建模的基本结构, 下面将以一个典型的登录测试过程完成测试应用。主要关注集成应用的可视化便捷性、动态测试的参数、作用域和执行的过程。更为方便的是对于复杂业务逻辑, 每一个控制流都

可以作为集成的组件，直接进行可视化再集成。如图6控制流的集成应用所示，集成使用了C-F控制流的基本结构，最外围是一个顺序结构(Sequence)，在此内部顺序地完成“inputPassword”变量的赋值和If结构；进入If结构中可视化地给出了“inputPassword==password”判断条件，可以根据动态测试的需要，灵活的修改判断条件，条件满足输出“welcome! ”，否则进入Else任务；在Else任务中，又采用了一个Sequence结构用于嵌入复杂的DoWhile循环结构和顺序执行最后的提示信息“please enter next day! ”；在嵌入的DoWhile循环结构中顺序的执行判断密码输入的次數“inputCounter = inputCounter +1”语句，用于计算输入密码的次數和频率，并顺序地给出每次输入的提示信息。至此，一个显式的N-S图静态过程设计被可视化地建模完成，根据模型的逻辑结构可以静态地分析逻辑关系，但对于复杂的业务流程显然是不够的，于是我们试图引入动态的测试和分析。



图6 控制流的集成应用

Fig.6 Integration application of control flow

如图7所示C-F组件动态测试，借助WF的可视化实现，可以很方便地实现集成动态测试。在图7(a)中给出了项目集成的更高层级的集成组件单元，将C-F中的过程流程封装成可以进一步可视化建模的组件，相当于一个功能黑盒参与更复杂的业务过程测试。图7(b)所示采用组件化的可视化建模，图中的Sequence_Workflow、If_Condition_Activity、Case_Condition_Activity、While_Activity等不再是基本的结构，而是可以包含复杂业务逻辑的组件功能单元，比如可以是一个疫情状态检查的组件“YiQingStateChecking_Activity”或者完成订单任务的“Order_Activity”等等，都可以在可视化建模完成后进行集成封装。可视化的动态集成测试，需要方便的可视化的动态参数修改，在图7(d)中，不仅可以动态的设置参数的变量类型和值，也可以随时设置并查看变量的作用域，比如password和inputPassword的作用域在Sequence基本结构中，而inputCounter变量的作用域在DoWhile基本结构中，同时还可以可视化的输入或导入动态的参数和变量，可以方便地实现可视化的动态测试。图7(c)给出了集成动态测试的结果，每一个过程和组件执行都显式可见，可以有效地帮助复杂业务逻辑的验证和测试，达到了预期的效果。



(a)组件单元

(b)组件化建模

(c)动态测试结果

名称	变量类型	范围	默认值
inputCounter	Int32	DoWhile	0
password	String	Sequence	"pp"
inputPassword	String	Sequence	输入 C# 表达式

(d)动态变量和参数

图7 C-F组件动态测试

Fig.7 Dynamic testing of C-F component

5 结论(Conclusion)

通过N-S和C-F可视化建模分析，我们将静态的N-S图无缝地转换成动态可视化的C-F图，显式地实现参数的输入、输出和处理过程，实现了业务逻辑和算法的正确性和有效性验证。对于较为复杂的业务流程，提供快速可视化的清晰验证，可以及时发现算法分析的错误。解决了传统N-S图的静态过程描述，缺少动态可视化的逻辑关系验证问题，在N-S图和编码之间起到桥梁作用。在实际的建模实现过程，快速清晰、高效，易于理解和应用，具有实际意义。

参考文献(References)

- [1] 张海藩,牟永敏.软件工程导论(第6版)[M].北京:清华大学出版社, 2013:117-128.
- [2] 邓德祥.结构化控制流程图(N-S图)的改进与模块程序设计的设想[J].计算机研究与发展,1995,32(12):21-26.
- [3] 刘正余,苏守宝,杨洋,等.N-S流程图分析及其在算法描述中的应用[J].井冈山大学学报(自然科学版),2010,31(06):70-72;81.
- [4] Yongliang Li, Huaqing Zhang, Hongbo Ju, et al. Visual simulation of quantitative thinning in a Chinese fir plantation based on workflow foundation[J]. Applied Engineering in Agriculture, 2015,31(3):339-349.
- [5] Microsoft. WindowsWorkflowFoundation[EB/OL]. <https://docs.microsoft.com/en-us/dotnet/framework/windows-workflow-foundation>, 2017-03-30.
- [6] Microsoft. ProceduralWorkflows[EB/OL]. <https://docs.microsoft.com/en-us/dotnet/framework/windows-workflow-foundation/procedural-workflows#controlling-execution-flow>, 2017-03-30.
- [7] 何隆.基于WF的工作流框架的设计和应用[D].重庆大学,2015.
- [8] 钟委明.基于WF可视化 workflow 编辑器的设计与实现[D].中山大学,2012.

作者简介:

孔令东(1973-),男,博士,讲师.研究领域:软件工程和 workflow 技术.