

软件体系结构分析与评价方法研究

杨 红, 杨德礼, 郑 志

(大连理工大学系统工程研究所, 大连 116023)

摘 要: 软件体系结构分析与评价的目的是为了识别体系结构设计中的潜在风险, 验证系统的质量需求在设计中是否得到了体现, 预测系统的质量并帮助开发人员进行设计决策。从软件体系结构的概念出发, 介绍并讨论了国内外有代表性的分析评价方法, 探讨了软件体系结构分析与评价方法研究中存在的问题和未来的发展方向。

关键词: 软件体系结构; 质量属性; 场景

Research on Software Architecture Analysis and Evaluation Methods

YANG Hong, YANG Deli, ZHENG Zhi

(System Engineering Research Institute, Dalian University of Technology, Dalian 116023)

【Abstract】 The purpose of software architecture (SA) analysis and evaluation is to identify the potential risks, verify that the quality requirements addressed in the design, predict the quality of software system and helps make proper architecture decision. Based on the concept of SA, this paper presents the recent representative analysis and evaluation methods, and then discusses some issues and promising tendency of the SA analysis and evaluation methods.

【Key words】 Software architecture; Quality attribute; Scenario

1 概述

自 20 世纪 90 年代初期以来, 由于对软件系统规模、复杂性和质量等方面需求的不断增长, 软件体系结构的研究受到了广泛的关注和重视。它将大型复杂软件系统的总体结构作为研究的对象, 认为系统中的计算元素和它们之间交互的高层组织是系统设计的一个关键方面。

大型的复杂软件系统开发所关注的主要问题之一就是质量, 在软件系统的早期设计阶段, 选择合适的体系结构对系统的许多关键质量(如可维护性、效率、可重用性等)起着决定性影响。因此, 尽早地按照质量需求评价一个体系的体系结构是非常重要的。软件体系结构分析与评价的目的是为了识别体系结构设计中的潜在风险, 验证系统的质量需求在设计中是否得到了体现, 预测系统的质量并帮助开发人员进行设计决策^[1]。具体如图 1 所示。

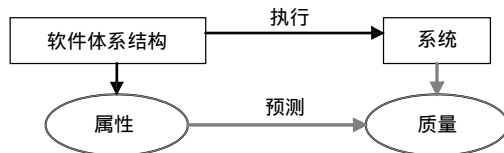


图 1 软件体系结构分析与评价

在评价方法中, 涉及到的主要术语定义如下:

(1) 软件体系结构 (Software Architecture, SA): 系统的一个或多个结构, 包括软件组件、这些组件的外部可见属性以及组件之间的相互关系。它以抽象的组件来表示可重用、可传递的系统, 系统由多个结构组成。

(2) 质量属性 (Quality Attributes): 一个组件或一个系统的非功能性特征。依照标准 ISO/IEC 9126-1, 共有 6 种特征: 功能性, 可靠性, 可用性, 有效性, 可维护性和可移植性, 并且它们又被分成若干子特征, 根据各个软件系统外部的可

见特征来定义这些属性。

(3) 场景 (Scenario): 对于风险承担者与系统的交互的简短描述。用来将那些模糊的质量属性需求描述转换为具体的易于理解的表述形式。

2 典型软件体系结构分析与评价方法比较

2.1 基于场景的体系结构分析方法 (SAAM)

SAAM 方法是 1993 年提出的第 1 个被广泛接受的体系结构分析评价方法。用于分析可修改性、可拓展性以及功能覆盖等质量属性。SAAM 方法采用大脑风暴技术构建场景, 体系结构的描述采用自然语言或其他形式表示。使用场景作为指定和评估质量属性的表述手段。将场景归类为直接场景和间接场景, 分别支持对体系结构的静态结构分析和动态分析。

SAAM 方法共分为场景开发、SA 描述、场景分类、单个场景评估、场景交互评估和总体评估 6 个评估活动, 如图 2 所示。

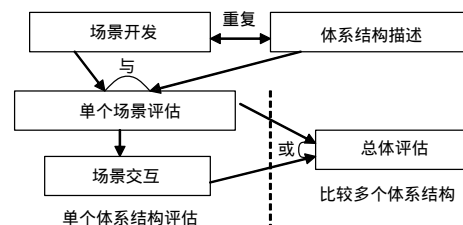


图 2 SAAM 的评估活动

基金项目: 国家自然科学基金资助重点项目(70031020)

作者简介: 杨 红(1977-), 女, 博士生, 主研方向: 适应性软件体系结构分析与评价方法; 杨德礼, 教授、博导; 郑 志, 博士生

收稿日期: 2006-03-13

E-mail: yh.Maggie@163.com

SAAM 以场景为中心, 简单易用, 但评估过程依赖专家经验, 仅仅是粗粒度的分析。没有对体系结构质量属性提供清晰的度量。

2.2 基于复杂场景的体系结构分析方法(SAAMCS)

SAAMCS 方法是基于 SAAM 方法的专门针对于风险评估研究的分析方法。该方法认为风险评估中最重要的影响因素是场景的复杂性。它在两方面对 SAAM 方法进行了扩展: 一是处理场景的方式, 二是评价场景冲突的位置。

2.3 针对进化和重用的体系结构分析方法(SAAMER)

SAAMER 方法是由 SAAM 方法扩展而来的, 主要研究进化和重用两个质量属性。该方法对于系统如何支持每一个质量目标、进化的风险级别及如何重用给出了建议。

SAAMER 分析方法揭示了一个系统关于目标和未来需求的重要性的敏感点, 使不同体系结构的选择工作更加便利。

2.4 体系结构权衡分析方法(ATAM)

ATAM 方法是基于 SAAM 方法的于 2000 年提出的针对性能、实用性(availability)、安全性和可修改性的分析方法。

ATAM 方法的分析评价是一个迭代过程, 包括 4 个活动: 体系结构描述及收集评估有关信息, 体系结构视图及场景实现, 质量属性模型建立与分析, 提供权衡报告。7 个阶段: 收集场景, 收集需求/约束/环境等信息, 描述体系结构视图, 解释场景, 属性细节分析, 定义敏感点, 定义权衡办法。如图 3 所示。

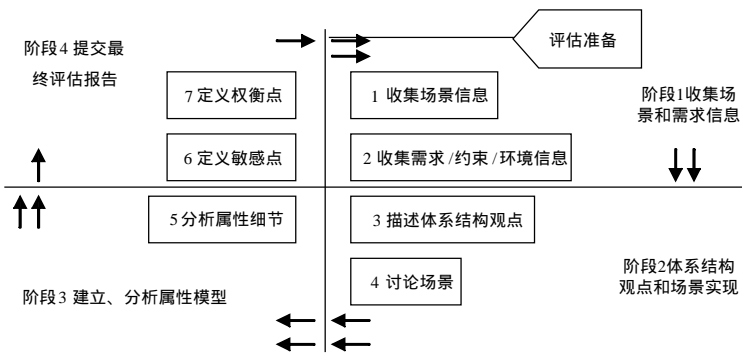


图 3 ATAM 的评估活动

ATAM 方法从不同角度探测系统的特性, 有助于提高场景的完整性并对体系结构的决策提供支持。分析质量属性之间的交互关系和依赖关系, 并探讨多个质量属性的权衡机制。该方法是被验证有效和广泛使用的一种方法, 但对质量属性并没有进行深入分析, 缺少定量的数据来支持分析的结果。

2.5 体系结构级别上的软件维护预测方法(ALPSM)

ALPSM 方法是 Bengtsson 和 Bosch 提出的在体系结构层次考察场景的影响, 评估软件系统可维护性的一种方法。该方法采用场景的变更, 描述一个具体的维护任务并用来分析体系结构, 对系统所需的维护性工作作出预测。输入和输出结果如图 4 所示。

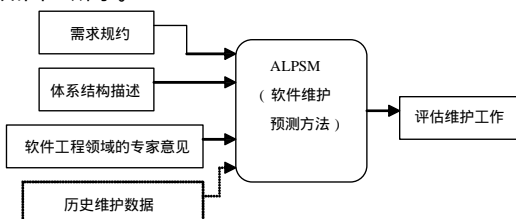


图 4 ALPSM 的输入和输出结果

ALPSM 方法提供了一个在设计期间重复地评估软件体系结构的度量工具。结合设计经验和历史数据对可维护性框架进行验证, 并且有效地引入变更, 预测系统的可维护性。方法的缺陷是具有不确定性。

2.6 基于场景的体系结构再工程方法(SBAR)

SBAR 方法是 Bengtsson 和 Bosch 提出的基于场景的软件体系结构再工程方法。该方法采用场景来评估软件体系结构的质量属性。通过改变设计提高不满足需求的质量属性。采用场景、仿真、数学建模和推理进行体系结构评价。提供 5 种类型的体系结构转变: 改变体系结构风格, 应用体系结构模式, 利用设计模式, 将质量需求转变为功能性需求和将质量需求分类。SBAR 方法有图 5 所示的 3 个主要活动: 将新的功能需求合并到体系结构中, 软件质量评估和体系结构转变。

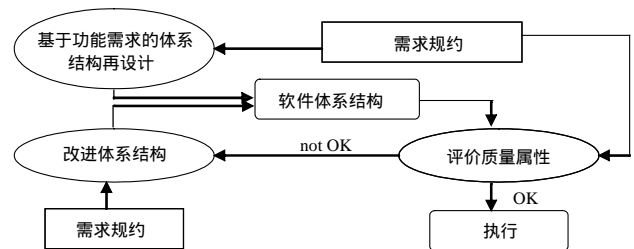


图 5 SBAR 的活动

2.7 体系结构级别上的可修改性分析方法(ALMA)

ALMA 方法是 Bengtsson 等人于 2004 年提出的基于预测的软件体系结构可修改性分析方法。该方法基于可维护性成本预测、风险评估和候选体系结构的比较, 通过对变更场景的构建、评价来进行可修改性的分析。包括确定目标、体系结构描述、发现并选择变更场景、评价场景和得出结论 5 个主要步骤。具体如图 6 所示。

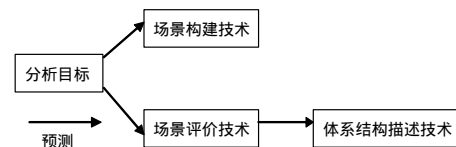


图 6 ALMA 的评估活动

ALMA 方法引入了定量的度量指标, 支持可维护性成本预测、风险评估和体系结构的选择多个角度评估体系结构的可修改性, 并提供了场景构建的停止准则。ALMA 方法的缺点是缺少对结果准确性的判断和风险评估完整性的判断。

2.8 主动设计评审方法(ARID)

ARID 方法是 2000 年 Clements 提出的一种将基于场景的方法和主动设计评审技术相结合的软件设计评审方法。一个完整的 ARID 过程分为 2 个阶段:

第 1 阶段准备工作分为定义评审专家、准备设计介绍、准备原始场景和准备评审 4 个步骤。

第 2 阶段评审工作分为介绍 ARID 方法、介绍设计、集体讨论和区分场景优先次序、执行评审和得出结论等 5 个步骤。

表 1 从方法的评估活动、目标、涉及的质量属性、体系结构的描述形式、分析评价技术、风险承担者、方法验证、知识的可重用情况以及相应的工具支持等几个方面评述国内外有代表性的体系结构分析与评价方法。

表 1 典型软件体系结构分析与评价方法比较

方法因素	SAAM	SAAMCS	SAAMER	ATAM	ALPSM	SBAR	ALMA	ARID
评估活动	6个	3个	4个	9个	6个	3个	5个	9个
目标	风险定义、合适性分析	预测相对的适应性风险评估	评估软件体系结构重用和进化	敏感点和权衡点分析	可维护性分析	按照需求的质量属性评价软件体系结构	变化冲突分析、预测维护工作	验证设计的生存能力
质量属性	主要是可修改性	适应性、可维护性	重用、进化	多个	可维护性	多个	可维护性	设计的合适性
体系结构描述	逻辑和模块化观点	微观和宏观的体系结构设计	静态、图示、动态的观点	过程、数据流、物理的和模块化的观点	没有特殊要求	一个被执行的体系结构	独立的体系结构描述和符号使用	在充分详细和合适的观点下进行初步设计
评价技术	基于场景的功能性和变化分析	场景(复杂场景)	信息模型、场景	问卷和度量相结合	场景	多个技术	依赖于分析目标	场景和活动设计评审相结合
风险承担者	所有的风险承担者	主要的风险承担者	设计者、管理者、最终用户	所有的风险承担者	设计者、分析者、评价者	体系结构设计者	不同的活动不同承担者	设计者
方法验证	多个领域	商业信息系统	转换系统	在不断的验证	血液透析系统	软件系统	不同领域	无
知识重用	无	无	无	知识库维护	无	无	历史数据	无
支持工具	部分	无	无	无	无	无	无	无

3 存在的问题和发展方向

软件体系结构是决定软件系统成败的重要因素之一。近几年来，软件体系结构的研究成为软件工程发展的一个热门研究领域，在软件体系结构分析与评价方法方面取得了一定的进展，提出了很多方法和支持工具，但是还存在一些问题，未来拥有广阔的发展空间。

(1)不同方法的结合

不同的体系结构分析与评价方法具有各自的特点和适用范围，将不同的方法进行结合，汲取不同技术的优点，从而获得更好的评估效果。例如将 SAAM 与 ALPSM 结合，使风险承担者参与促进他们之间的交流，加深对系统质量属性的了解；将 ATAM 方法和 CBAM (cost benefit analysis method) 方法结合，综合两种方法的特点，提供对经济风险的分析和折中；将基于场景的评估技术与其它的特定分析技术如实时的分析方法 RMA(rate monotonic analysis)相结合，都是很好的途径。

(2)度量的应用

目前的体系结构分析与评价方法大多采用基于场景的技术，为了更精确地分析体系结构的质量属性，绝大多数的研究者都认为采用度量在体系结构阶段评价属性是更加精确的技术之一。它包括质量属性的度量选择、度量的规模和一组度量方法。可以采用两种方法：一是适应现有的度量技术，如采用在设计和代码一级经验验证有效的面向对象指标，如动态复杂性、动态耦合等。面向对象的适应性度量对于软件可维护性预测是很有效的，因为度量所需数据只能从源代码中搜集，而在体系结构阶段，还没有开发原型系统也不存在源代码。因此可采用第 2 种方法，根据体系结构的特性，定义和验证新的度量指标，目前一些学者正在做这方面的工作。GQM (goal-question-metric) 是一个用于定义新的度量的很好的技术。它的主要活动是根据目的、观点和环境定义目标，确定与目标相关的代表属性的问卷，就提出的每个问题给出解答。这里的目的是与体系结构评价分析和最终产品的质量预测相关的。观点主要依赖于评估的目标和与之密切相关的评价人员的角色(开发人员、用户、管理人员和维护人员)。环境是指体系结构是被作为一个中间设计产品还是作为自身的最终产品对待。

(3)风险评估的研究。

体系结构评价的目的就是分析体系结构，定义潜在的风险，在软件系统正式开发前预测它的质量。关于潜在的风险定义，上述的评价方法基本都是利用改变场景和场景交互来揭示体系结构中的潜在问题域。当评价一个系统对场景的响应时，可修改的程度可以用来表示可度量的风险。场景的复杂性也是风险评估的一个重要因素。需要进行的改变和领域专家的经验也可作为系统为了进化或重用支持风险标准的系统可修改性程度。风险决策利用可探测的场景进行优化。降低潜在的风险也可通过分析属性交互获得。可采用可重复的分析方法将风

险降低到可接受的标准。高风险可通过仿真、模型或原型系统进行分析。QFD (Quality Function Deployment) 是一个可以被考虑采用的技术。它被 SAAMER 和 SAEM 方法采用。这种技术在形式化表示内部质量属性与质量特征或子特征之间的关系的的过程中是非常有用的，因为这个过程必须被作为特定的应用领域、开发过程和体系结构描述语言来研究。

(4)方法中的重用因素

体系结构分析与评价方法中有很多因素是可以重用的，类似系统中经常出现的场景、多次发现的同种类型的风险、质量属性刻画以及评估时提出的问题等，专家的经验、现有知识库都是值得考虑重用的因素。使用这些可重用的因素，能够提高评估工作的效率。例如 ATAM 方法使用 ABAS 来实现重用。ABAS 能够为体系结构设计师提供基于对复发问题的解决方案及使用这些方案时的已知困难预先封装的分析和提问的集合，有助于重用与某个重复出现的体系结构方法相关的分析结果和反复出现的风险、敏感点和权衡点。

(5)适应性研究

随着软件规模和复杂度的增加，对软件的适应性提出了很高的要求。适应性与软件的扩展性、修改性、动态特性有密切关系。适应性是指当软件生存环境发生变化时软件实体的结构或行为可以随之改变并满足新环境需要的特性。作为体系结构质量特征的一个方面，适应性具有一定的特殊性，目前对适应性体系结构分析与评价的研究还很不完善，缺少定性和定量的度量指标和系统的评价方法。环境变化是产生适应性问题的根源，而变化的不可预知性增大了适应性软件设计的困难，研究软件生存环境和生存环境系统，分析其中的可变因素，将与适应性相关的特征映射成的质量属性度量指标，是进行适应性研究的一个可采用的途径。

4 结论

软件体系结构是决定软件系统成败的重要因素之一。在软件体系结构阶段进行分析与评价，识别体系结构设计中的潜在风险，验证系统的质量需求在设计中是否得到了体现，预测系统的质量，能够有效地提高软件系统的成功率。近几年来，在软件体系结构分析与评价方法方面取得了一定的进展，提出了很多方法和支持工具，但是还存在一些问题，需要不断完善。

(下转第 70 页)