

# 基于复用的软件构架评估方法及其在嫦娥工程中的应用

胡智新<sup>1,2</sup>, 李春来<sup>1</sup>, 欧阳自远<sup>1</sup>

1. 中国科学院国家天文台, 北京 100012  
2. 中国科学院研究生院, 北京 100039

**摘要:** 针对目前软件构架评估方法未考虑复用构架评估知识的局限性, 提出一种新的、基于构架评估知识复用的软件构架评估模式(包括软件构架评估方法元模型和应用框架)。运用一致的评估元模型, 建立高层模型, 辅助构架评估决策; 并在应用框架支持下, 系统地复用评估历史数据, 将 SAEM 的活动整合为一个系统的、可复用的、可管理的过程。该方法已成功应用于中国探月工程地面应用系统的软件构架评估, 降低了深空探测航天复杂系统的研制风险。

**关键词:** 软件构架; 软件构架评估; 软件构架评估模式; 场景。

中图分类号: TP301

文章编号: 1000-2383(2006)03-0384-05

收稿日期: 2005-12-28

## Reuse-Based Software Architecture Evaluation Methods

HU Zhi-xin<sup>1,2</sup>, LI Chun-lai<sup>1</sup>, OUYANG Zi-yuan<sup>1</sup>

1. National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China  
2. Graduate School, Chinese Academy of Sciences, Beijing 100039, China

**Abstract:** The limit of the current scenario-based software architecture evaluation (SAE) is not taken into account the knowledge reuse of SAE. A new reused-based software architecture evaluation pattern, which includes meta-model and application framework, is proposed in this paper for integrating the SAE into a systematic and reusable procedure. Exploiting the meta-model and the application framework of SAE can make decision of SAE in high-level and reuse the historic data of SAE. Finally, this pattern is successfully applied and validated in the software architecture of the ground system for data, research and application of lunar exploration program of China. The pattern reduces the research risk of deep-space exploration complex system.

**Key words:** software architecture; software architecture evaluation; software architecture evaluation patterns; scenario.

## 0 引言

软件构架 (software architecture, SA) (Kruchten, 1995; IEEE, 1998) 是软件系统最初设计决策的体现和风险承担者交流的手段, 决定着软件系统所有的质量属性和项目结构。软件构架评估 (evaluating software architecture, ESA) (Li and Henry, 1993) 是在项目的竞争需求和构架决策映射到代码之前对决策进行公布和评估, 判断 SA 是

否满足项目所期望的质量属性要求, 将决策文档化, 是保证软件质量的必要条件。成功的软件产品开发和演化依赖于 SA 的设计或选择是否恰当。因此, 尽早评估 SA 至关重要。

ESA 分为提问技术 (如调查问卷、检查表、场景等)、度量技术 (如仿真、试验等) 和混合技术 (提问技术和度量技术相结合) (Abowd *et al.*, 1996; Bass *et al.*, 1998)。Parnas and Weiss (1985) 提出了 ESA 的基础, Carnegie Mellon 的软件工程研究

表 1 基于场景的软件构架评估方法对比分析

Table 1 Analysis result of scenario-based software architecture evaluation methods

	SAAM	ATAM	ARID	CBAM	SPE	QAW
涉及的质量属性	可修改性、变化性和功能	所有质量属性	设计方法的适宜性	成本和收益	性能	所有质量属性
分析对象	构架文档	构架方法和样式;构架文档	组件的接口规范	关键属性、ATAM 评估结果	构架文档、性能需求文档	多个尚未完成的构架文档
评估时机	在组件已经分配到各个模块中以后	在构架设计方法选定后的软件开发各阶段	系统开发早期或概念阶段	构架设计方法已定,且 ATAM 评估已完成	已确定组件的性能约束条件时	多个软件构架完成之前
采用方法	头脑风暴,场景分析、演练	头脑风暴,构架分析,确定敏感点、权衡点	头脑风暴	量化收益和成本值	构建、调整和验证性能模型,量化静态分析场景	求精质量属性,比较构架优劣
复杂度	简单	复杂	简单	较复杂	复杂	较复杂
支持复用	否	否	否	否	否	否

所 (software engineering institute, SEI) 则将软件构架评估引入软件开发周期,针对不同的焦点,提出一系列基于场景的软件构架评估方法 (software architecture evaluation methods, SAEM)。目前,广泛应用于软件工程实践中的主流 SAEM 有软件构架分析方法 (software architecture analysis method, SAAM)、构架权衡分析方法 (architecture tradeoff analysis method, ATAM)、成本—收益分析方法 (cost benefit analysis method, CBAM)、中间设计的主动评审 (active reviews for intermediate designs, ARID)、质量属性专题讨论法 (quality attribute workshop, QAW) 以及软件性能工程 (software performance engineering, SPE) (Kazman *et al.*, 1994, 1996, 2001; Abowd *et al.*, 1996; IEEE, 1998; Kazman, 1998; Smith and Woodside, 1999; Clement, 2000; Williams and Smith, 2003; Clement *et al.*, 2002; Barbacci, 2003a, b; Moore *et al.*, 2003)。

本文分析基于场景的主流 SAEM, 存在未从复用的角度考虑软件构架评估知识的管理和应用的局限性, 缺乏 SAEM 元模型、应用框架支持 ESA 的高层模型辅助决策和评估知识的复用。结合中国探月工程地面系统工程实践, 本文提出一种新的、基于构架知识复用的软件构架评估模式——软件构架评估方法元模型和软件构架评估应用框架 (以下简称“评估元模型”和“应用框架”)。目的: (1) 提出一致的评估元模型, 建立高层模型, 辅助构架评估决策, 并指导复用构架评估数据; (2) 在应用框架支持下, 系统地复用评估历史数据, 将 SAEM 的活动整合为一个系统的、可复用的、可管理的过程。提高 ESA 效率和

质量, 降低风险和成本。

## 1 基于场景的软件构架评估方法对比

本节分析广泛应用于软件工程实践的、基于场景的主流 SAEM (表 1)。由于未考虑软件构架知识的系统性复用, 基于场景的 SAEM 存在以下局限性: (1) ESA 缺乏高层模型 (high-level model) 辅助决策。ESA 是一个风险度很高的软件过程, 不同的 SAEM 关注焦点、方法、适用状况、应用时机和结果精确度等存在较大差异, 需要软件组织构建高层模型进行辅助决策: 采用何种 SAEM, 何时实施, 以及如何复用构架评估知识和历史数据; (2) 缺乏基于评估知识复用的 SAEM 元模型和应用框架。目前的 SAEM 只是关注于单一 SA 的评估, 只是方法的重用, 缺乏从软件过程性管理的整体性考虑。此外, 构架评估是一个人员密集型过程, 评估效果过多依赖于评估人员素质。因此需建立基于评估知识复用的、一致的 SAEM 元模型和应用框架支持 ESA。

## 2 软件构架评估元模型和应用框架

由于软件构架评估过程是基于组件开发的基本活动, 多数软件开发组织开发、维护的产品, 在特定领域的构架评估存在共性, 因此, 针对目前 SAEM 未从复用的角度考虑软件构架知识的系统性复用, 软件组织不能以构架评估的知识和历史数据指导以后的构架评估过程的局限性, 本文提出一种新的、基于构架评估知识复用的软件构架评估模式——评估

元模型和应用框架,将 SAEM 的活动整合为一个系统的、可复用的、可管理的过程. 利用评估元模型和应用框架支持 ESA 高层决策,复用构架评估历史数据,提高 ESA 的效率与质量.

### 2.1 软件构架评估元模型

在分析基于场景的主流 SAEM 的基础之上,本文抽象 SAEM 的基本要素,构建新的评估元模型(图 1). 目的:组织 ESA 各要素,指导软件组织根据待评估 SA 的特点、资源约束等因素,在评估应用框架的基础上,在评估过程中参考可复用历史数据库(以下简称“历史数据库”)的应用实例(场景、模型、构架决策或风险问题等),决策采用何种 SAEM、何时实施(图 2),并在评估结束后增加新的数据至历史数据库,支持其演化. 通过复用 ESA 知识和历史数据,提高构架评估质量和效率,降低决策的风险和代价. 评估元模型的应用方法和步骤见 3.3 节.

### 2.2 软件构架评估应用框架

从基于组件化的软件过程管理考虑,本文提出基于历史数据复用的评估应用框架,将过程管理、数据复用和技术支持整合(图 3,斜线阴影部分是目前 SAEM 的范围). 目的:ESA 在统一的应用框架内,基于评估元模型,与管理机制融合,运用 SAEM 库、可复用历史数据库(以下简称“历史数据库”)和工具集,提供统一平台,支持面向特定领域的、大粒度的构架评估历史数据复用,实现构架评估的工程一体化.

通过评估应用框架和评估元模型的支持,复用了构架评估历史数据,特别是复用已有的、经实践验证的构架决策,可提高构架评估的质量和效率,降低风险和代价,提高评估结果的可信度.

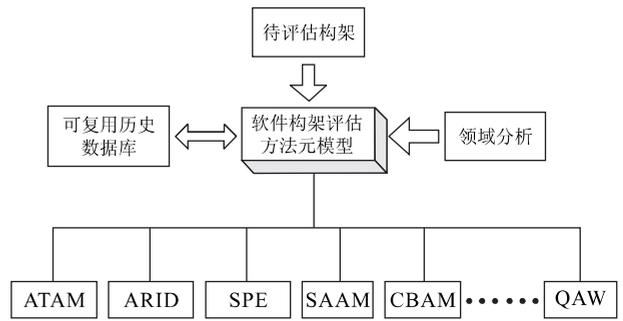


图 2 软件构架评估元模型应用概念图

Fig. 2 Application of meta-model of SAEM

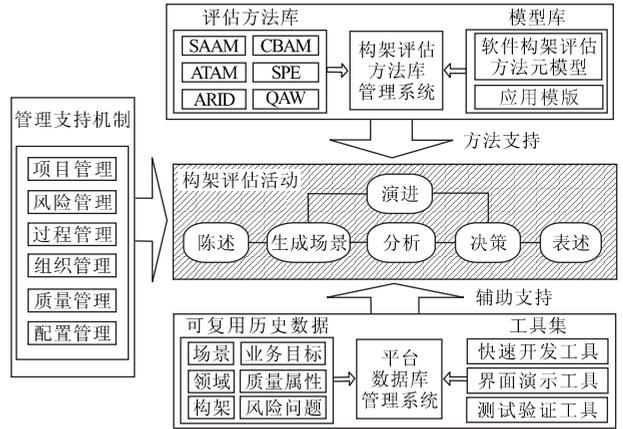


图 3 软件构架评估应用框架的过程模型图

Fig. 3 Process model for application framework of software architecture evaluation

评估应用框架的主要应用步骤:

(1)分析. 根据待评估 SA 的业务目标和领域特性,通过领域分析和评估元模型,结合历史数据库,分析其质量属性和拟采用的构架方法,在项目管理的指导下选择 SAEM.

(2)实施. 基于评估元模型,通过工具集支持,复用历史评估数据,实施选定的构架评估方法:①在生成场景时,基于评估元模型,从历史数据库中,提取与待评估构架质量属性相关的场景为种子场景,在此基础上扩展或裁剪,生成待评估构架的场景;②在工具集的支持下应用步骤①中生成的场景,结合历史数据库的场景验证案例,验证待评估构架方法是否满足质量属性要求,运用风险管理识别、分析风险问题,并与历史数据库对照;③决策. 根据验证结果和风险问题(构架对关键质量属性的满足程度、重要场景的排序等),参考历史数据库的构架决策历史数据,进行构架评估决策. 如果发现构架方法不能满足业务目标,或风险度过高,需调整构架方案并返回步

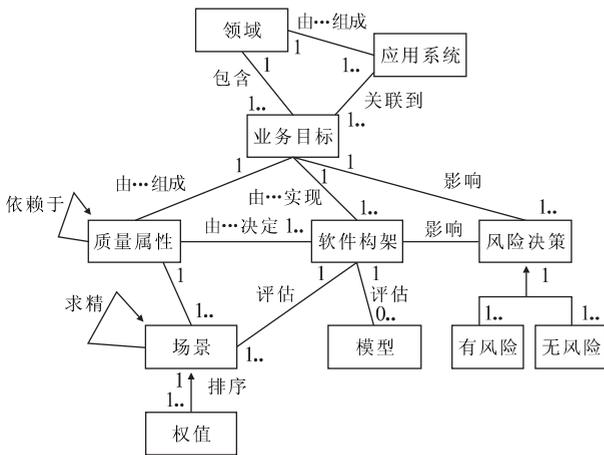


图 1 软件构架评估元模型

Fig. 1 Meta-model of SAEM

表 2 标注了优先级的 GSDAS 效用树子集  
Table 2 Prioritized utility tree subset of GSDAS

第 2 层 质量属性	第 3 层质量属性求精	第 4 层质量属性场景	重要性	难度	累加和
性能	P1:0 级数据处理时间为毫秒级	P1.1:0 级数据的处理时间<0.1 s	H	H	(H,H)
	P3:准实时处理和显示快视图像	P3.1:处理和显示快视图像时间<5 s	L	H	(L,H)
	Ra1:系统过载或崩溃数据不丢失	Ra1.1:双机实时切换,恢复故障机<5 min	H	H	(H,H)
可靠性/ 可用性	Ra4:数据处理、存储的故障时间	Ra4.1:数据处理、存储或备份引起的故障时间每周<1 h. 处理机、网络及存贮资源的使用、I/O 通道的吞吐量的余量>30%	H	M	(H,M)
	可维护性	M1:对一个子系统的更改不要求改变其他子系统	M1.1:对数据管理子系统的修改不会影响其他子系统及数据的查找、存储	M	H
M5:对操作系统、数据库、COTS 产品的升级时间缩短 50%		M5.1.1 天完成 AIX 操作系统的升级	M	M	(M,M)
		M5.2:1 天完成应用软件包的升级	M	L	(M,L)
可演化性	E1:系统支持 50 个大规模数据请求的输出	E1.1:系统能够每天处理 100 个不同的数据输出请求(>1 G 数据/每个请求)	M	M	(M,M)

骤(1)重新评估.

(3)演进. 形成构架评估最终报告. 基于评估元模型,提取评估过程中产生的数据(包括业务目标、质量属性、构架方法、场景、测试案例、模型、风险问题、构架决策和最终报告)至历史数据库,对历史数据库进行配置管理,维护其完整性和一致性,促进其演进.

在实施过程中,同时实施项目管理、质量管理和过程管理,控制评估过程、质量、进度、成本和决策.

### 3 案例应用

本节介绍在中国探月工程的地面应用系统(ground segment for data and application system, GSDAS)研制中,利用本文提出的软件构架评估方法模式成功进行了 GSDAS 的 ESA,降低了深空探测航天复杂系统的研制风险(评估过程的细节情况由于保密缘由进行了更改和省略).

#### 3.1 项目背景

嫦娥工程(2003~2007)是中国研制和发射月球探测器开展月球科学探测,是对月球表面的地貌、地形、地质构造、元素分布、物质类型与月表环境进行全球性、整体性与综合性的探测,加深对月球、地一月系和太阳系起源与演化的认识,对有开发利用前景的月球能源与资源的分布和规律进行探测与研究. GSDAS 是嫦娥工程五大系统之一. GSDAS 负责嫦娥工程科学目标;嫦娥 1 号卫星(CE-1)的业务运行及有效载荷的科学探测计划制定;协作处理海量的月球地质、地理、遥感、空间等科学数据,组织科学研究等任务.

#### 3.2 分析阶段

首先,建立评估小组,提出日程安排,做好准备工作;其次,评估负责人介绍评估元模型和应用框架的方法、过程和技巧;项目决策者介绍 GSDAS 的上下文和系统开发的主要业务目标、关键质量属性(集中于性能、可靠性、可操作性和可维护性);构架师对 SA 做出描述(包括三级分层结构、客户机/服务器方法、管道/过滤器方法). 应用领域分析和评估元模型,结合历史数据库,分析 GSDAS 的质量属性和拟采用的构架方法,此次 SAEM 选择 ATAM.

#### 3.3 实施阶段

(1)基于评估元模型,从历史数据库中提取与待评估 SA 质量属性相关的场景为种子场景,根据关键质量属性在此基础上扩展或裁剪,生成 GSDAS 的质量效用树(utility tree)(表 2),设定场景优先级. 本文对场景采用 9 级权值,把审查的关注重点放在了(H,H)上,其次是(H,M)、(M,H)、(M,M),其余不考虑.

(2)在工具集的支持下应用步骤(1)中生成的场景,结合历史数据库的场景验证案例,描述并分析高优先级场景的 SA 设计方法,确定出 SA 的有风险决策、无风险决策、敏感点和权衡点,运用风险管理识别、分析风险问题,并与历史数据库对照. 验证 GSDAS 的构架方法是否满足质量属性要求. 其后,评估人员集体讨论形成更大的场景集合,通过投票表决确定这些合成场景的优先级,并使用合成场景再次分析 GSDAS 的构架方法,未产生以前步骤中没有发现的高优先级场景.

(3)决策阶段. 根据步骤(2)的验证结果,结合历史数据库,判定 GSDAS 采用的 SA 可满足其业务

目标,确定了 GSDAS 的 5 个权衡点并录入文档,形成评估最终报告,评估小组向 GSDAS 风险承担者表述评估结果。

### 3.4 演进阶段

基于评估元模型,提取 GSDAS 评估过程中产生的数据至历史数据库,对历史数据库进行配置管理,维护其完整性和一致性,支持其演进。

## 4 总结

与传统的 ESA 相比,本文提出新的、基于构架评估知识复用的软件构架评估方法模式——软件构架评估元模型和应用框架,在统一平台支持下,有效复用了面向特定领域的、大粒度的构架评估要素的历史数据,特别是复用已有的、经实践验证的构架决策数据,提高了构架评估的质量、效率和可信度,降低了风险和代价。

从细化软件构架评估模式的应用前景来看,未来的研究方向和突破点为:(1)增强评估应用框架的灵活性、稳定性和可扩展性,更好地支持复用;(2)集成软件构架评估模式于软件产品线开发中,优化软件组织的资源。

## References

- Abowd, G., Bass, L., Clements, P., et al., 1996. Recommended best industrial practices for system architecture evaluation. Technique Report, CMU/SEI-96-TR-025.
- Barbacci, M., 2003a. Using the architecture tradeoff analysis method (ATAM) to evaluate the software architecture for a product line of avionics systems: A case study. Technique Report, CMU/SEI-2003-TN-012, SEI, Carnegie Mellon University. [http://www.sei.cmu.edu/publications/documents/03\\_reports/03tn012.html](http://www.sei.cmu.edu/publications/documents/03_reports/03tn012.html).
- Barbacci, M., 2003b. Quality attribute workshops (QAWs). Third Edition. Technique Report, CMU/SEI-2003-TR-016, SEI, Carnegie Mellon University. [http://www.sei.cmu.edu/publications/documents/03\\_reports/03tr016.html](http://www.sei.cmu.edu/publications/documents/03_reports/03tr016.html).
- Bass, L., Clement, P., Kazman, R., 1998. Software architecture in practice. Addison-Wesley, Reading, MA.
- Clement, P., 2000. Active review for intermediate designs. Technique Report, CMU/SEI-2000-TN-009, SEI, Carnegie Mellon University. [http://www.sei.cmu.edu/publications/documents/00\\_reports/00tn009.html](http://www.sei.cmu.edu/publications/documents/00_reports/00tn009.html).
- Clement, P., Kazman, R., Kelein, M., 2002. Evaluating software architectures: Methods and case studies. Addison-Wesley, MA.
- IEEE, 1998. IEEE glossary of software engineering terminology, 610. 12-1990.
- Kazman, R., Abowd, G., Bass, L., et al., 1994. SAAM: A method for quality through formal technical review. In: Proceedings of the 16th International Conference on Software Engineering, Sorrento, Italy, May, 113-122.
- Kazman, R., Abowd, G., Bass, L., et al., 1996. Scenario-based analysis of software architecture. *IEEE Software*, 13(6): 47-55.
- Kazman, R., 1998. The architecture tradeoff analysis method. In: Proceedings of the Fourth International Conference on Engineering of Complex Computer Systems (ICECCS98).
- Kazman, R., Asundi, J., Klein, M., 2001. Quantifying the costs and benefits of architectural decisions. In: Proceedings of the 23rd International Conference on Software Engineering (ICSE 23), Toronto, Canada, May, 297-306.
- Kruchten, P. B., 1995. The 4+1 view model of architecture. *IEEE Software*, 12(6): 42-50.
- Li, W., Henry, S., 1993. Object-oriented metrics that predict maintainability. *Systems and Software*, 23(2): 111-122.
- Moore, M., Kazman, R., Klein, M., et al., 2003. Quantifying the value of architecture design decisions: Lessons from the field. In: Proceedings of the 25th International Conference on Software Engineering (ICSE 25), Portland, Oregon, May.
- Parnas, D. L., Weiss, D., 1985. Active design review: Principles and practices. In: Proceedings of the 18th International Conference on Software Engineering.
- Smith, C. U., Woodside, M., 1999. Performance validation at early stages of software development. *The Journal of Systems and Software*. <http://www.perfeng.com/papers/smitwood.pdf>.
- Williams, L. G., Smith, C. U., 2003. PASASM: A method for the performance assessment of software architecture. In: Proceedings of the Workshop on Software and Performance (WOSP2002), Rome, Italy, July.