

# 基于 UML 时序图的系统运行时耦合度量方法

杜防汛, 黄志球, 杨俊, 张玉宏

(南京航空航天大学信息科学与技术学院, 南京 210016)

**摘要:** 面向对象耦合性是软件质量重要的内部属性, 该文分析了现有基于静态分析的耦合度量, 以及其因继承和多态的使用而产生对运行时耦合度量的不足。提出了以用于系统动态建模的 UML 时序图为基础, 定义了反映系统运行时的耦合度量指标, 并通过实例验证了耦合指标定义的可操作性和实用性。

**关键词:** 统一建模语言; 时序图; 运行时; 耦合度量

## System's Run-time Coupling Measure Based on UML Sequence Diagram

DU Fangxun, HUANG Zhiqiu, YANG Jun, ZHANG Yuhong

(Information Science and Technology Institute, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

**【Abstract】** Object-oriented coupling is one of the important internal attributes of software quality. This paper firstly analyses the limitations of the existing static coupling measurement because of using inheritance and polymorphism, then defines the coupling metrics describing system run-time coupling, which based on UML sequence diagram which used to dynamic modeling, and validates the operability and practicability of this definition by a simple example.

**【Key words】** Unified modeling language; Sequence diagram; Run-time; Coupling measure

在面向对象的度量研究中, 作为与软件质量密切相关的软件内部属性之一的耦合性, 其度量研究自然成了软件度量领域的一个研究热点。国内外相关研究者提出了 30 多种关于耦合性的度量<sup>[1-3]</sup>, 但绝大部分度量都是建立在对源码或设计作静态分析的基础上。随着面向对象分析与设计应用的深入, 特别是继承和多态的频繁使用, 原来基于静态分析的耦合度量, 因为大量继承和动态绑定的发生, 已不能很好反映系统运行时对象间耦合形态。因此, 作为现有基于静态分析耦合度量的一个补充, 很有必要提出一套反映系统运行时耦合形态的耦合度量。

系统运行时对象间动态交互信息的获取最直接的方法就是执行目标代码, 在此过程中收集相关交互信息, 然而此过程花费较大, 复杂度高。随着 UML 的广泛应用, 特别是 UML 中用于系统动态建模的时序图(Sequence Diagram), 它描述了按时间顺序排列的对象交互作用, 包括交互作用中的对象和交换的消息序列。因此, 本文在分析 UML 时序图中对象间动态交互信息的基础上, 提出了一种反映系统运行时类间耦合形态的一种度量方法。

### 1 相关研究

面向对象的设计中, 类是最基本的模块, 耦合主要指类间相互关联的紧密程度。高质量的软件设计, 一个重要原则就是要遵循“低耦合原则”。对耦合性的度量不但可以掌握由于耦合性引起的软件设计问题, 还可以为软件的某些外部质量属性的度量提供依据。到目前为止, 许多研究者针对耦合性提出了一系列度量指标: 较早的Chidamber和Kemerert<sup>[1]</sup>从单个类与外界耦合的角度, 提出了对象间耦合度量(Coupling Between Object Classes, CBO)和类间响应度量(Response For

a Class, RFC); MOOD<sup>[2]</sup>度量集是从描述系统整体耦合度的角度, 提出耦合因子度量(Coupling Factor, COF); Briand<sup>[3]</sup>等人提出了统一耦合度量框架, 并在此基础上定义了相关耦合度量集。以上耦合度量研究, 均是在对源码或设计(如UML类图)作静态分析, 得到类间耦合的度量。

随着面向对象分析与设计应用的不断深入, 特别是在一些大型的项目开发中, 如一些开源项目、框架应用等, 其设计中越来越多地使用继承和多态特性, 提高了系统内部重用及可维护性。然而现有基于静态分析的耦合度量, 因为系统运行时大量继承和动态绑定的发生, 已不能很好地反映系统实际运行的耦合形态。

图 1 说明继承的使用对运行时耦合影响, 图中包含一个类图和时序图, 类图中描述了父类 A' 和 B' 及各自实现的方法 mA() 和 mB(), 且在方法 mA() 中调用方法 mB(), 类 A、B 分别继承了各自的父类 A'、B'。时序图中描述系统运行时的一个场景, 类 A 的实例对象 a 调用了类 B 的实例对象 b 的方法 mB()。由以上分析可知, 以静态分析类图角度看, 类 A' 和 B' 存在类间耦合, 类 A 与类 B 并不产生耦合; 而从时序图分析可知, 在实际运行时, 因 A 的实例对象 a 与 B 的实例对象 b 间存在方法调用, 因此类 A 和类 B 存在运行时类间耦合。

因此, 作为现有基于源码或设计静态分析的耦合度量的补充, 本文讨论了在分析 UML 时序图中对象间动态交互信

**作者简介:** 杜防汛(1982 - ), 男, 硕士生, 主研方向: 软件度量与质量评价与保障; 黄志球, 博导、教授; 杨俊, 博士生; 张玉宏, 硕士生

**收稿日期:** 2005-11-03

**E-mail:** dufangxun@nuaa.edu.cn

息的基础上，提出了一种反映系统运行时类间耦合的度量方法，更好地反映系统实际运行的耦合形态。

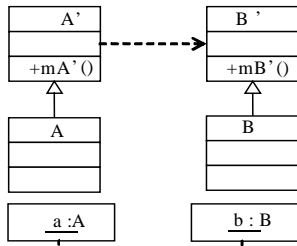


图1 类图与时序图

## 2 基于时序图的运行时耦合度量

统一建模语言(UML)是一种直观的、可视化的建模语言，被OMG组织采纳为面向对象分析和设计的标准建模语言，得到了广泛的应用。时序图是UML中用来显示系统中运行时对象间的交互，它提供了一种按照时间顺序对给定场景中相互协作对象之间消息传递的视图。时序图包含两个基本的元素：交互的对象和对象间交互的消息序列，也是下面定义运行时类间耦合的基础。

### 2.1 集合定义

在定义基于时序图的动态耦合之前，首先按照时序图的基本元素定义下面的集合：

$S = \{SD_1, SD_2, \dots, SD_n\}$ ， $S$ 表示系统所有时序图集合，其中 $SD_i$ 为第 $i$ 个时序图；

$C = \{C_1, C_2, \dots, C_n\}$ ， $S$ 表示系统时序图中所有实例化对象所属类的集合；

$M = \{m_1, m_2, \dots, m_n\}$ ， $M$ 表示系统所有定义的实现所有方法集合；

$O = \{SD_i: ObjId \mid \forall SD_i \in S\}$ ， $O$ 表示系统所有时序图中所有实例化的对象集合，其中 $SD_i: ObjId$ ， $SD_i$ 为对象所属的时序图， $ObjId$ 为对象名称；

$Msg = \{SD_i: MsgId \mid \forall SD_i \in S\}$ ， $Msg$ 表示系统所有时序图中对象间交互的消息序列集合，其中 $SD_i: MsgId$ ， $SD_i$ 为交互消息所属的时序图， $MsgId$ 为对象间交互的消息名称。

### 2.2 关系定义

在定义以上集合的基础上，下面引入集合间关系运算的定义，作为下一步定义耦合度量的基础。

$$R_{OC} = \{(o_i, C_i) \mid \forall o_i \in O, \exists C_i \in C, o_i \in \text{instance}(C_i)\}$$

其中， $o_i$ 是 $C_i$ 的实例化对象， $R_{OC}$ 定义了时序图中实例化的对象和所对应的类之间的关系；

$$R_{OO} = \{(o_i, m_i, msg, o_j, m_j \mid \forall o_i, o_j \in O, m_i, m_j \in M, msg \in Msg, i \neq j)\}$$

$R_{OO}$ 定义了时序图中两个对象间的消息传递，消息 $msg$ 是从时序图中源对象 $o_i$ 的方法 $m_i$ ，调用目标对象 $o_j$ 的方法 $m_j$ 。

### 2.3 耦合定义

在定义以上UML时序图各基本元素集合以及各集合间关系的基础上，下面定义基于时序图的类间耦合度量。按照耦合产生的因素可分类间耦合为：类间动态消息耦合，类间

方法调用耦合，类间聚合耦合。按照Briand等人提出的静态耦合框架<sup>[3]</sup>，耦合按方向(direction of coupling)又可分为导入耦合和导出耦合，导入耦合是指一个类的调用其他类的方法或属性产生的耦合称为该类的导入耦合，反之若类的方法或属性被其他类调用而产生的耦合称为该类的导出耦合。

**定义1** 类间动态消息耦合(Coupling of Dynamic Message between Classes)

类间动态消息耦合描述了在时序图中系统类的实例化对象间因消息传递而产生的类间耦合。它聚合了类在时序图中所有的实例化对象，与其他类的实例化对象间动态交互消息的总和。下面给出导入、导出的动态消息耦合度量定义。

导入消息耦合(Import Coupling of Dynamic Message)：

$$IC_{DM}(C_i) = \{(m_i, C_i, msg, m_j, C_j) \mid \forall (o_i, C_i) \in R_{OC}, \exists (o_j, C_j) \in R_{OC}, msg \in Msg, C_i \neq C_j \wedge (o_i, m_i, msg, o_j, m_j) \in R_{OO}\}$$

导出消息耦合(Export Coupling of Dynamic Message)：

$$EC_{DM}(C_i) = \{(m_j, C_j, msg, m_i, C_i) \mid \forall (o_i, C_i) \in R_{OC}, \exists (o_j, C_j) \in R_{OC}, msg \in Msg, C_i \neq C_j \wedge (o_j, m_j, msg, o_i, m_i) \in R_{OO}\}$$

**定义2** 类间方法调用耦合(Coupling of Methods Invocation between Classes)

类间方法调用耦合描述了在时序图中的系统类的实例化对象间因方法调用而产生的类间耦合。它聚合类在时序图中所有实例化对象，调用其他类实例化对象的方法数的总和，一个方法调用过程可以包含多个交互消息，下面是导入、导出方法调用耦合度量定义。

导入方法调用耦合(Import Coupling of Methods Invocation)：

$$IC_{MI}(C_i) = \{(m_i, C_i, m_j, C_j) \mid \forall (o_i, C_i) \in R_{OC}, \exists (o_j, C_j) \in R_{OC}, msg \in Msg, C_i \neq C_j \wedge (o_i, m_i, msg, o_j, m_j) \in R_{OO}\}$$

导出方法调用耦合(Export Coupling of Methods Invocation)：

$$EC_{MI}(C_i) = \{(m_j, C_j, m_i, C_i) \mid \forall (o_i, C_i) \in R_{OC}, \exists (o_j, C_j) \in R_{OC}, msg \in Msg, C_i \neq C_j \wedge (o_j, m_j, msg, o_i, m_i) \in R_{OO}\}$$

**定义3** 类间聚合耦合(Coupling of Classes Aggregated)

类间聚合耦合是聚合了时序图中所有类的实例化对象间交互的类对(Coupling Pair of Classes)的集合，对象间交互包括了动态消息和方法调用，下面是导入、导出类间聚合耦合度量定义。

导入类间聚合耦合(Import Coupling of Classes Aggregated)：

$$IC_{CA}(C_i) = \{(m_i, C_i, C_j) \mid \forall (o_i, C_i) \in R_{OC}, \exists (o_j, C_j) \in R_{OC}, msg \in Msg, C_i \neq C_j \wedge (o_i, m_i, msg, o_j, m_j) \in R_{OO}\}$$

导出类间聚合耦合(Export Coupling of Classes Aggregated)：

$$EC_{CA}(C_i) = \{(m_j, C_j, C_i) \mid \forall (o_i, C_i) \in R_{OC}, \exists (o_j, C_j) \in R_{OC}, msg \in Msg, C_i \neq C_j \wedge (o_j, m_j, msg, o_i, m_i) \in R_{OO}\}$$

## 2.4 耦合度量的应用

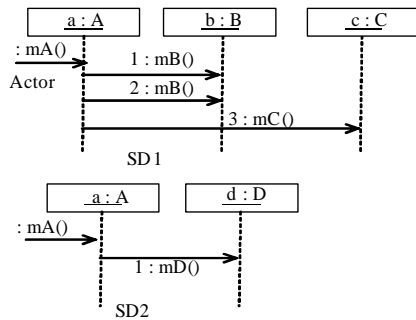


图 2 时序图 SD1 和 SD2

表 1 时序图 SD1 和 SD2 的耦合度量集

导入耦合	IC_DM(A)	{(mA, A, SD1:1, mB, B), (mA, A, SD1:2, mB, B), (mA, A, SD1:3, mC, C), (mA, A, SD2:1, mD, D)}
	IC_MI(A)	{(mA, A, mB, B), (mA, A, mC, C), (mA, A, mD, D)}
	IC_CA(A)	{(mA, A, B), (mA, A, C), (mA, A, D)}
导出耦合	EC_DM(B)	{(mA, A, SD1:1, mB, B), (mA, A, SD1:2, mB, B)}
	EC_MI(B)	{(mA, A, mB, B)}
	EC_CA(B)	{(mA, A, B)}
	EC_DM(C)	{(mA, A, SD1:2, mC, C)}
	EC_MI(C)	{(mA, A, mC, C)}
	EC_CA(C)	{(mA, A, C)}
	EC_DM(D)	{(mA, A, SD2:1, mD, D)}
	EC_MI(D)	{(mA, A, mD, D)}
EC_CA(D)	{(mA, A, D)}	

下面介绍通过对一个简单应用场景下的时序图的分析，得到以上定义的耦合度量的结果。图 2 所示描述了系统在一

(上接第 44 页)

在验证过程中发现不少网页在文本中包含了要找的关键词，却在元数据注解中没有包含该关键词。通过数据检验可以找到这些网页，并将其提交给系统管理员，由系统管理员决定是否在这些文档中加入关键词。

数据集成不是新问题，而是在 XML 出现以前就已经被广泛研究的一个领域<sup>[4]</sup>。虽然 XML 可以在很大程度上减轻对异构数据进行集成的工作量，但是还不能够独立完成相关数据的集成工作。数据集成的关键因素在于需要有一种将数据内容与语义进行关联的语言，W3C 也意识到这一点。资源描述框架 (Resource Description Framework, RDF) 和 Web 语义工作组都在致力于寻找 Web 数据的语义表示方法。

作者在 Web 数据挖掘中通过两个步骤实现数据的集成。首先爬行者将多个原始 HTML 文档中的所需数据分别析取下来，从每个析取器得到的输出 XML 文档都是最终输出文档的一部分。然后将这些输出片断进行连接，并将其传送到负责整合数据的 XSLT 过滤器，最终形成一个完整的 XML 文档。

## 3 结论和展望

本文介绍了 Web 数据挖掘的相关理论知识，并提出了一

个应用场景下的动态运行的时序图 SD1 和 SD2，为了更好地说明耦合度量，对于各个类的方法，类间交互消息等详细信息都只作简单表示。表 1 是在对图 2 的分析基础上得到该时序图运行时的耦合度量结果。

## 3 结束语

本文在描述系统动态建模的 UML 时序图的基础上，定义了类间的消息级、方法级和类级的系统运行时耦合度量指标，并通过一个 UML 时序图的实例验证耦合指标定义的可操作性和实用性，很好地反映了系统运行时的耦合形态。目前，基于 UML 的系统分析与设计的应用越来越广泛，特别是 UML 的动态建模方法的使用，如 UML 时序图。因此，在本文提出的系统运行时的耦合度量的基础上，建立一套基于 UML 动态模型的自动化耦合度量工具，动态分析系统运行时耦合度，应为下一步研究工作的重点。

## 参考文献

- Chidamber S, Kemerer C F. A Metrics Suite for Object-oriented Design[J]. IEEE Trans. on Software Eng., 1994, 20(6): 476-493.
- Fernando B A, Miguel G, Rita E. Toward the Design Quality Evaluation of Object-oriented Software Systems[C]. Proceedings of the 5<sup>th</sup> International Conference on Software Quality, 1995.
- Briand L C, Daly J, Wüst J. A Unified Framework for Coupling Measurement in Object-oriented Systems[J]. IEEE Trans. on Software Eng., 1999, 25(1): 91-121.
- Briand L C, Labiche Y. A UML-based Approach to System Testing[J]. Software and Systems Modeling, 2002, 1(1): 10-42.
- Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language Users Guide[M]. Addison-Wesley, 1998.
- 张 静. 面向对象系统耦合性度量研究与应用[D]. 南京: 南京航空航天大学, 2005.
- 冀振燕. UML 系统分析设计与应用案例[M]. 北京: 人民邮电出版社, 2003.

种基于 XML 的实现方法，通过对 Web 页面的一系列处理，最终得到所需的数据。

下一步的工作主要集中在以下几个方面：(1)如何优化数据析取处理过程，以便能够更加快捷、方便地获取所需数据；(2)利用自动或半自动工具建立和维护域定义规则，处理丢失或冲突的数据；(3)如何从多个 Web 站点进行数据的分类和汇总。

## 参考文献

- Kosala R, Blockeel H. Web Mining Research: A Survey[J]. ACM SIGKDD, 2000, (7): 311-321.
- Myllymaki J. Effective Web Data Extraction with Standard XML Technologies[C]. Proceedings of the 10<sup>th</sup> International Conference on World Wide Web. New York: ACM Press, 2001: 689-696.
- Liu Ling, Pu Calton, Han Wei. XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources[C]. Proc. of International Conference on Data Engineering, San Diego, California, 2000-02.
- Han Jiawei, Chang Kevin. Data Mining for Web Intelligence, Volume 35[M]. Los Alamitos: IEEE Computer Society Press, 2002: 64-70.