基于设计文档的组件提取研究

刘丽娟 杨 一 张 胤

（国家计算机网络应急技术处理协调中心上海分中心 上海 201315）

摘 要 为了从目标系统有效提取构件，达到软件复用的目的，文章提出了一种基于设计文档的组件提取技术。该方法通过借助计算机辅助软件工程工具设计目标系统，针对该设计文档提取概念组件及系统的体系结构，并使用可扩展标记语言元素和特征元匹配度验证提取结果的准确性，为软件复用提供了依据。实验通过两个实例：产品资源管理平台和转专业系统，说明该方法的应用，其中文件上传概念组件和数据转储概念组件得到提取和验证，并成功复用实现系统功能。结果表明，该方法不仅提供可重用的构件框架，而且提高了软件复用的成功率。

关键词 组件提取；概念组件；语义分析；软件复用

中图分类号 TP 311 **文献标志码** A

Component Extraction Based on Design Documents

LIU Lijuan YANG Yi ZHANG Yin

(*Shanghai Branch*, *Computer Network Emergency Response Technical Team/Coordination Center of China*, *Shanghai* 201315, *China*)

Abstract In order to extract components from the target system effectively and achieve the goal of software reuse, a component extraction method based on design documents was proposed in this paper. Computer aided software engineering tools were used to design the target system, then concept components and system architecture were extracted from this design documents. After the accuracy of extract result was verified by extensible markup language element and matching degree of characteristic element, foundation for software reuse was provided in this way. The application of this method was illustrated through two practical applications: product resource management platform and change-major student system. During this process, concept components of fileUpload and dataDump were extracted and verified, and components which match these concept components were reused successfully for the system. The result shows that the proposed method not only provides a framework of reusable components, but also improves the success rate of software reuse.

Keywords component extraction; concept component; semantic analysis; software reuse

收稿日期：2015-06-21 **修回日期**：2015-07-15

作者简介：刘丽娟（通讯作者），硕士，研究方向为软件工程、数据挖掘，E-mail：liulijuan123job@126.com；杨一，硕士，研究方向为计算机网络、知识发现；张胤，学士，研究方向为信息处理、语义分析。

1 引　言

20 世纪 60 年代以来的“软件危机”，让人们明白了软件批量生产和提高质量的重要性。软件复用是提高软件生产力和质量的一种重要技术：通过吸收和借鉴以往项目开发的知识与积累的经验，既充分利用了已有开发成果，又消除了开发过程的重复性工作，提高软件开发效率和软件质量，大大减轻维护工作。

然而软件复用过程并不顺利，具体存在以下三个问题：(1)系统需要哪些构件；(2)构件以何种形式存在，提供了什么功能和接口；(3)如何建立映射关系，查找匹配构件。现有工作中，复用过程较多地关注构件检索与装配方面，即问题(3)；而对于问题(1)和(2)，如何提取构件，往往缺少深入的研究。

可复用构件的提取直接影响着软件复用的效率。现阶段，常用的组件提取方法有领域分析法[1]和面向服务分析与设计(Service Oriented Analysis，SOA)方法[2]。领域分析法借鉴知识，较依赖于领域专家来决策组件的切分，应用不具备普遍性；SOA 法尽管能从底层对复用构件进行分割，但要求有中立的接口，组件切分粒度较难把握；此外，Caldiera 等[3]提出的软件度量学方法对组件进行权值评估，其结果具有较大的主观性。这些方法灵活性上受到很大限制，造成复用成功率较低，都未能从根本上给出通用的组件提取实际方案。

目前计算机辅助软件工程(Computer Aided Software Engineering，CASE)工具应用广泛，用于对系统开发的规范设计，如 Rational Rose、Powerdesigner。本文提出了一种基于设计文档的组件提取方法。遵循基于构件的软件开发思想，首先借助 CASE 工具描述目标系统，在统一建模语言(Unified Modeling Language，UML)图基础上提取概念组件及系统体系结构；其次用可扩展标记语言(Extensible Markup Language，XML)元素和特征元匹配度指标衡量语义分析的准确性。

本文第 2 节介绍基于构件的软件开发的基本概念；第 3 节介绍了基于设计文档得到概念组件的方法；第 4 节阐述了使用 XML 元素和特征元匹配度指标进行验证的过程；第 5 节结合实例阐明该方法的应用；第 6 节总结全文并给出下一步的工作设想。

2 基于构件的软件开发

基于构件的软件开发[4]能够提高软件复用效率。它包含以下几个阶段的生命周期：第一阶段为需求获取，采用仿真办法描述客观世界的人工系统；第二阶段是领域分析，分析客观系统，设计出逻辑系统；第三阶段为系统集成，找到合适构件类，将其生成实例，用过程控制语言描述出系统中的各子系统，生成各种输入输出构件实例等；最后集成系统，通过实际运行不断修改，直到用户完全满意为止。

构件是指在软件系统中可独立部署、接口由契约指定且多由第三方提供的可组装软件实体。它具有相对独立功能，可以明确辨识，和语境有明显依赖关系。一般构件的内涵包含如下三部分：

(1)构件实体：包括代码、服务和系统三方面内容；

(2)构件文档：包括需求规约、系统构架、设计文档和测试案例等各种具有复用价值的软件资源；

(3)对软件构件的要求：分为对外部接口、内部封装(支持复用)的要求，可提供技术文档的支持。

可复用性是软件构件的本质特征，是评价构件质量好坏的重要因素。组件准确提取是查找与组装的前提，提取过程主要是针对目标系统(待开发的应用系统)，目的是得到概念组件(满足业

务需求的组件)，提取是为了成功的复用。提取包括系统分解与度量两个主要步骤，二者在实施过程中相互依存。系统分解需要一种构件识别手段，度量用于验证分解结果。

3 基于设计文档的概念组件提取

3.1 设计文档

CASE 是一套规范化的工具，包含图形工具、代码生成器等。CASE 工具覆盖软件开发生命周期各阶段，能够加快开发速度，保证软件的质量。目前主流的 CASE 语言规范是 UML[5]，UML 用于系统可视化建模，是对产品进行说明、可视化和文档编制的一种标准语言。使用 UML 描述系统，一方面，形成了可视化标准图对系统规范建模，帮助全面理解系统；另一方面，CASE 中的组件图有助于提取概念组件。设计文档主要指 CASE 工具用于系统建模的文档。

3.2 抽取概念组件

概念组件是指设计文档中描述的组件，要求符合业务需求[6]。

借助 CASE 工具 ArgoUML 建模，能够得到存储为 XML 格式的 UML 图。以产品资源管理平台为目标系统，该平台通过聚合产品资源(文档、安装文件等)，以达到管理产品的目的。其中文件系统是平台核心子系统，提供产品资源文件的上传、格式转换和下载等功能。实现功能包括文件管理(fileManage)、文件上传(fileUpload)、用户信息管理(userManaInfo)、登录(login)等。图 1 组件图中，fileManage 与 userItemInfo 组件都依赖 fileUpload 组件；buyHandle 组件与 userItemInfo 组件相互依赖，并依次与 userManaInfo、login 组件形成了层次的依赖关系。

为了准确理解并分割出复用构件，遵循组件划分原则[7]：(1)类与其他类的交互方式；(2)服务

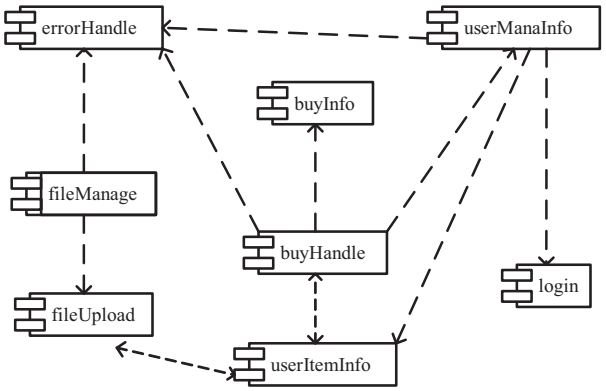

图 1 产品资源平台组件图

Fig. 1 Component diagram for education resources platform

类单独划分；(3)类与其他类的依赖。概念组件从组件名称(name)、输入参数(Input para)、输出参数(Output para)、返回值(Return value)、提供接口(Interface)等方面进行描述。借助 ArgoUML 工具将 CASE 设计文档保存为 XML 格式，并解析 XML 文档，对照功能描述、接口等抽取概念组件。

以 fileManage 类为例，其 Upload、Transfer 接口因涉及到上传功能，据此原则将其划分到 fileUpload 组件类。如图 2 所示，fileUpload 组件包含 id、size 等属性及 Upload、FileShow、FileTransfer 等方法。

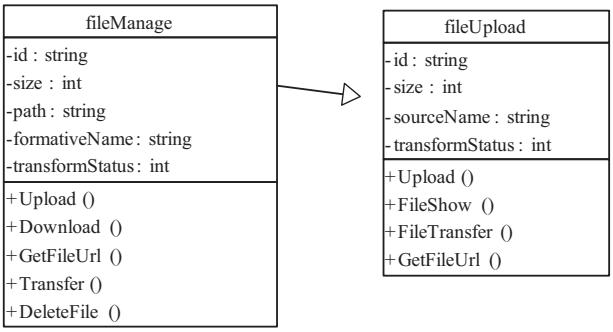

图 2 FileUpload 组件的类图

Fig. 2 Class diagram for FileUpload component

3.3 抽取体系结构

体系结构表明概念组件之间存在的关系。两个概念组件间可建立组装关系，也可不具备组装关系[8]。在 ArgoUML 生成的 XML 文档(UML

图）中，包含了组件的语义特征：name 标识和 incoming 标识。

（1）name 标识

XML 文档会为 UML 图中的每个类自动分配一个 name 标识，作为划分 XML 文档中的标识对象的特征。例如：在 XML 文档中，name = 'manage_item' 表明这是一个 manage_item 名称的标识对象，代码中 visibility、isSpecification 分别表明可见性与规格属性，具体如图 3 所示。

```
<UML:Package xmi.id =
'-34--52--37--127--27024b68:13deca58cad:-
8000:0000000000000AEE'
name = 'FileUpload' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false'>
```

图 3　XML 文档中的 “name” 标识

Fig. 3　“name” in XML documentation

（2）incoming 标识

XML 文档中，</UML:StateVertex.incoming> 标识反映了元素 A 与 B 依赖特征。例如：在 XML 片断中（图 4），体现了 manage_item 与 add_item 依赖关系。

```
name = 'manage_item' isSpecification = 'false'>
<UML:Transition xmi.idref =
'-34--52--37--127--27024b68:13deca58cad:-
8000:0000000000000ADE'/>
</UML:StateVertex.incoming>
</UML:SimpleState>
<UML:SimpleState xmi.id =
'-34--52--37--127--27024b68:13deca58cad:-
8000:0000000000000ADA'
name = 'add_item' isSpecification = 'false'>
```

图 4　XML 文档中的 “incoming” 标识

Fig. 4　“incoming” in XML documentation

根据组件的语义特征，定义六元组<C_i, t, p, r, f, C_j>元素，用以刻画目标系统的体系结构，意义如下：

<C_1, t, p, r, f, C_2>元素：当满足 t 条件时，表明 XML 文档（UML 图）中时序图部分的概念组件 C_1 传递 p 参数给概念组件 C_2，具有 r 联系标识；f 代表组件的连接状态，f 为 next，表示 C_2 仍有后继组件联系；f 为 end，表明 C_2 已经是终点。

由于体系结构刻画的是概念组件间的调用依赖关系[9]，所以重点考察 XML 文档的时序图部分。为了得到 XML 文档中的体系结构，本文提出基于上下文约束的 UML 序列图分析法进行分析，步骤如下：

（1）记概念组件集合为 C=$\{C_1, C_2,\cdots, C_k\}$；记 XML 文档中具有 name 标识的对象集合为 X=$\{X_1, X_2,\cdots, X_n\}$，X_n 为第 n 个对象的标识符；记体系结构集合为 G，初始 G 为空集；

（2）遍历概念组件集合 C，在集合 X 中，如果查找到标识对象 X_n 的 name 标识与概念组件 C_i 的名称 cName 一致，并存在从概念组件 C_i 到 C_j 的 r 联系，组件间传递的参数是 p，则记为<C_i, t, p, r, next, C_j>，将这个六元组加入集合 G 中，同时在 C 中删除组件 C_i；如果不存在从概念组件 C_i 开始的 incoming 对象标识，则在 C 中删除组件 C_i；

（3）重复步骤（2），直到集合 C 为空，得到体系结构集合 G。

上述过程的体系结构的抽取流程如图 5 所示。最终得到包含六元组的体系结构集合 G。

本文提出的方法，将概念组件作为构件复用的目标，赋予了概念组件在体系结构下的领域含义，为后续的系统构建、软件复用提供依据。

4　抽取结果验证

在以上设计文档的分析基础上，接着对其进行语义验证，以衡量语义分析的准确性，提高复用成功率。

4.1　可扩展标记语言元素验证

XML 元素验证是指通过解析 XML 文档，

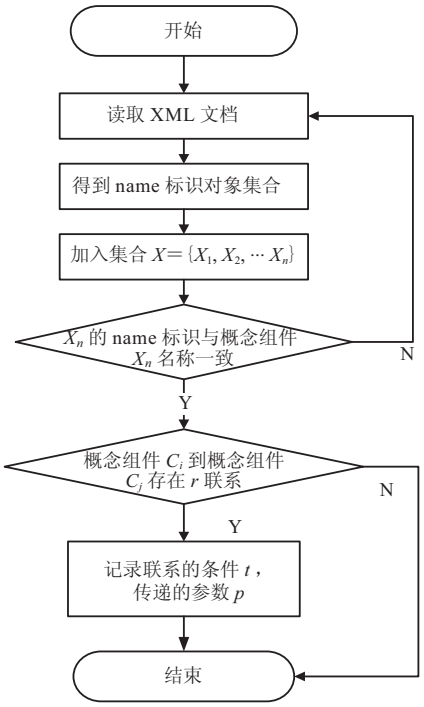

图 5 分析体系结构的流程

Fig. 5 Process of system architecture analysis

验证对应组件符号的 XML 元素。考虑组件图在 ArgoUML 工具下转换成为 XML 文档，提取得到的概念组件同样用 XML 语言进行描述，所以二者具有共同的语言基础[10]。XML 元素代表了组件的特征。这样，通过编写 XMLparser 解析器[11]解析 XML 文档，运用 matcher 迭代进行匹配，在包含组件名称的字符串中判断 XML 元素与组件符号是否一致对应。解析器 XMLparser 的部分代码如下：

```
for(int i = 1; i<str.length(); i++) {
Matcher matcher = pattern.matcher(str.
substring(i, i+1));
if(matcher.find()) {
string = string + str.substring(index, i)+"\n";
index = i;
}}
String = string + str.substring(index);
String[] strArray = string.split("\n");
```

4.2 特征元匹配度

特征元是属性、方法、功能的集合。以属性为例，包含构件类型(cType)、参数类型(pType)等基本元素；以方法为例，包含前继(pre)、返回值(returnType)等基本元素；以功能为例，包含功能描述(funcDes)、对象格式(objectFormat)基本元素。表 1 给出了属性、方法、功能的语法和语义特征元素。

表 1 属性、方法和功能的特征元素

Table 1 Characteristic element of property, method and function

分类	属性	方法	功能
语法	cType	pre	funcDes
	access	post	objectFormat
	pType	returnType	
	readMethod		
	writeMethod		
语义	cName		

本文用匹配度衡量元素的联系程度，分为精确匹配和部分匹配两种情况。

(1)精确匹配：上述元素中，若两个属性所有的元素都匹配成功，则匹配度为 1；

(2)部分匹配：若两个属性中部分元素匹配成功，则匹配度由公式(1)计算：

$$M_k=\prod_j M_F p_j \times(\sum_{k=1}^{n} M_Y p_k \div n) \qquad (1)$$

其中，M_F 表示基本元素的语法匹配度；M_Y 表示基本元素的语义匹配度；p_j 代表属性中的语法元素；p_k 代表属性中的语义元素；n 为语义元素的个数。

这样得到概念组件的各部分特征元匹配度。

按照各个组件图中的组件重要性，分配权值 W_i，由公式(2)计算整体概念组件的匹配度 M_A：

$$M_A=\sum_{i=1}^{n}\sum_{k=1}^{n} W_i M_k \qquad (2)$$

最后将 M_A 与阈值比较，如果 M_A 在阈值范围内，则通过验证，说明语义分析的概念组件结果

符合设计者的要求；如果 M_A 不在阈值范围内，则结束，需要重新进行组件提取。根据统计学规律，当阈值取 0.7(推荐阈值)[12]时，匹配度结果与设计预期较为符合，因此阈值一般采用 0.7。

5　系统实例

实验通过两个实际系统说明基于设计文档的组件提取方法的应用。实际系统为产品资源管理平台和转专业系统。产品资源管理平台的文件子系统中，需文件入库和批量导出操作，含文件格式转换功能。转专业平台中，涉及课程信息批量录入，需打印学生信息的操作。综合以上实际业务，提取概念组件 fileUpload 和数据转储 dataDump，分别如表 2 和表 3 所示。

表 2　fileUpload 概念组件

Table 2　Concept component of FileUpload

名称	fileUpload
提供的功能	upload file、transfer file、show file、get file's url
输入参数	id、size、sourceName、transformStatus
返回值	void
类型	业务逻辑型
接口	public Boolean Upload(String id, String url)
	public void FileShow(String id)
	public boolean FileTransfer(String id, String sourceName)
依赖	FileTransfer

表 3　dataDump 概念组件

Table 3　Concept component of DataDump

名称	dataDump
提供的功能	批量存储，批量打印
输入参数	id, url
返回值	void
类型	业务逻辑型
接口	public Boolean export(String id, String url)
	public void import(String id, String url)
依赖	无

经过分析，文件子系统的体系结构如图 6 所示，fileManage 包含 fileUpload 和 fileCategory，概念组件 fileUpload 依赖 fileTransfer 组件。

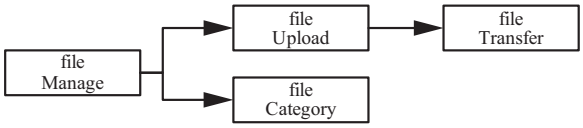

图 6　文件子系统体系结构

Fig. 6　System architecture of file subsystem

提取概念组件和分析体系结构后，接下来对语义分析结果进行验证：首先对 XML 元素进行验证，得到结果{fileManage，fileUpload，fileTransfer}，结果与组件图(图 2)中的组件名称一致，符合 XML 元素验证；然后计算各个概念组件的特征元匹配度，如表 4 所示。

表 4　各个概念组件特征元素的匹配度

Table 4　Matching degree of each concept component by its Characteristic element

分类	fileManange	fileUpload	fileCategory	fileTransfer
语法	0.84	0.86	0.85	0.83
语义	0.82	0.91	0.83	0.71
整体	0.81	0.88	0.82	0.75

结果表明，上述概念组件(fileManage、fileUpload、fileCategory、fileTransfer)的整体特征元匹配度与阈值 0.7(这里取推荐阈值)进行比较，大于 0.7，验证得到了通过。该方法的实际结果为下一步软件复用提供基础，反映了设计者的意图。

在接下来的软件复用中，运用本文提出的方法得到两个概念组件的实际需求后，通过在构件库中(如上海构件库)进行查找并结合自行开发的方式，获得相应的实体构件，进行系统装配。由于接口设计规范且具有良好通用性[13]，最终，两个系统都成功实现了文件入库和数据存储的功能，重用了文件上传、数据转储。整个过程提供可重用的构件框架，消除了软件开发的重复劳动，同时，构件接口相对独立，避免重复编码可能引入的错误，从整体上提高了软件复用效率。

为比较本文方法的效果，将本文方法与领域专家方法进行对比。考虑 SOA 方法针对大数量的组件分析不具备可操作性，软件度量方法采用标准不一，难以通用，因此不予比较。分别运用两种方法，对 10 个、20 个、…、100 个构件进行提取，统计复用成功率情况，结果如图 7 所示。由图 7 可以看到，本文方法整体的复用成功率都较领域专家方法高。领域专家方法由于有一定的主观性，因此装配复用效果不理想。而本文方法提供通用的可重用构件框架，成功率较高，基本在 0.6 左右，更符合实际需求。

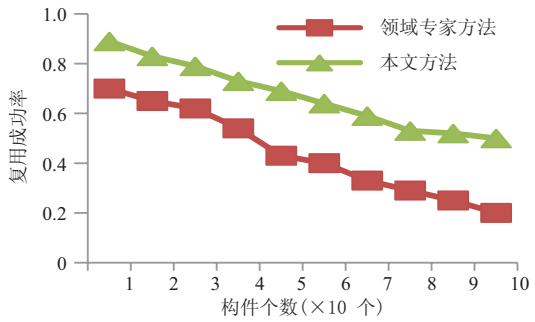

图 7 两种方法下的复用成功率情况

Fig. 7 Reuse success rate of two methods

6 结束语

基于构件的软件开发是软件复用关注的重点。本文提出了基于设计文档的组件提取方法，该方法将概念组件作为软件复用目标，建立概念组件及体系结构分析模型。较前人的组件提取研究工作，本文的方法摒弃了主观性评估，增强了可操作性与通用性。本文的主要贡献在于：(1)提出一套完善系统的组件提取方法；(2)提出一种完备的体系结构分析法(基于上下文约束的 UML 序列图分析法)；(3)提出一种针对概念组件的通用验证方法，采用 XML 元素验证和特征元匹配度对提取结果进行衡量；(4)构建一套规范的组件描述框架；(5)通过与领域专家方法进行比较，证明本文方法具备较高的复用成功率。

下一步的工作将关注复用过程的后期，考虑如何实现目标系统到构件检索的准确定位，加强对构件装配的适应性，以便高效地实现软件复用。

参 考 文 献

[1] Wang D, Huang N, Ye M. Reliability anaysis of component-based software based on relationships of components [C] // IEEE International Conference on Web Services, 2008: 814-815.

[2] Meilicke C, Stuckenschmidt H. Analyzing mapping extraction approaches [C] // Proceedings of ISWC'2007 Workshop on Ontology Matching, 2007: 25-36.

[3] Caldiera G, Basili VR. Identifying and qualifying reusable software components [J]. Computer, 1991, 24(2): 61-70.

[4] Farrugia J. Model-theoretic semantics for the web [C] // Proceedings of the 12th International Conference on World Wide Web, 2003: 29-38.

[5] He KQ, Jiang H, He F. Extended UML with role modeling [J]. Wuhan University Journal of Natural Science, 2001, 6(1-2): 175-182.

[6] Chen XH, Liu J, Liu ZM. Requirements monitoring for internetware: an interaction based approach [J]. Science China Information Sciences, 2013, 56(8): 1-15.

[7] Wei B, Jin Z, Zowghi D, et al. Implementation decision making for internetware driven by quality requirements [J]. Science China Information Sciences, 2014, 57(7): 1-19.

[8] Korthaus A, Schwind M, Seedorf S. Leveraging semantic web technologies for business component specification [J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2007, 5(2): 130-141.

[9] Zhang XY, Zheng L, Sun C. The research of the component-based software engineering [C] // The Sixth International Conference on Information Technology: New Generations, 2009: 1590-1591.

[10] Liu L, Yang C, Wang JM, et al. Requirements model driven adaption and evolution of internetware [J]. Science China Information Sciences, 2014, 57(6): 1-19.

[11] 叶品菊, 胡远望. MIS 系统可复用软部件研究与设计 [J]. 价值工程, 2014, 33(36): 215-216.

[12] Li Y, Sun KW, Yang J, et al. Model-based system configuration approach for internetware [J]. Science China Information Sciences, 2013, 56(8): 1-20.

[13] 邵飞, 彭蓉. 一种基于领域知识的非功能需求建模辅助方法 [J]. 计算机学报, 2013, 36(1): 39-54.