

文章编号: 1672-2892(2012)04-0519-05

数字化工程中面向系统重用的信息提取

周在华^a, 曹云峰^{a,b}, 庄丽葵^{a,b}, 王彪^b

(南京航空航天大学 a.自动化学院; b.高新技术研究院, 江苏 南京 210016)

摘 要: 系统重用技术作为数字化工程的一部分, 具有减少重复劳动的特点。为数字化工程实现缩短设计周期, 降低开发成本和提高产品质量提供了重要支持。面向对象系统中的类模型、状态模型和交互模型在系统重用方面起着重要作用, 为了提取被重用系统中的结构和功能信息, 以类图、状态图和序列图为例, 研究其数据在系统存储文件.sbs 包中的组织结构, 并且采用 VC 工具对一组实例图中的信息进行提取。结果表明, 通过该存储文件获得的信息完全符合原设计意图, 这为管理重用信息提供了数据支持。

关键词: 数字化工程; 系统重用; 信息提取; 统一建模语言

中图分类号: TN92; TP319

文献标识码: A

Information extraction in digital project for system reusability

ZHOU Zai-hua^a, CAO Yun-feng^{a,b}, ZHUANG Li-kui^{a,b}, WANG Biao^b

(a.College of Automation Engineering; b.Academy of Frontier Science, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu 210016, China)

Abstract: As a part of digital project, reuse system can provide important support for digital project by reducing wasteful duplication of efforts so as to shorten design cycles, cut development costs and improve product quality. The class model, state model and interaction model of object-oriented system play important roles in the system reuse. In order to extract the structure and function information from the reuse system, class diagrams, state diagrams and sequence diagrams are taken as examples to study the structure of data in the system files(.sbs) package. VC tools are used to extract information from a set of instance graphs. The results indicate that, the information extracted from the storage file meets the original design requirements, which provides information support for managing reuse data.

Key words: digital project; system reuse; information extraction; Unified Modeling Language

“飞机数字化工程”是我国航空工业系统为了推动和促进数字化设计技术在飞机设计制造领域的应用而启动的一项重要工程, 其目的是通过应用数字化设计技术改变传统设计方法存在的设计周期长、质量低、花费大等缺点^[1]。缩短新产品设计周期最有效的方法是重用已有系统的部分功能。有数据显示, 在零部件产品开发中, 约 40% 是重用过去的零部件; 另外 40% 是对以往的相似件稍作修改得到的; 而只有 20% 是新设计的零件^[2]。系统重用可以减少不必要的重复劳动, 降低开发成本, 缩短开发时间, 极大地提高产品的竞争力。将系统重用应用到数字化设计中无疑会促进数字化设计所提出的缩短设计周期, 降低开发成本和提高产品质量等要求的实现。

数字化设计当前还处于探索阶段, 主要任务是针对特定的设计对象搭建协同设计环境。文献[3]针对飞控系统设计搭建了一个集 Rhapsody, Matlab/Simulink 等多种成熟软件为一体的虚拟样机开发平台。文献[1]在该平台上设计开发了一个飞控系统。航天二院李伯虎院士团队搭建了一个基于 HLA(High Level Architecture)总线多领域协同仿真平台^[4]。目前, 系统重用正从认知、表达、本体论及管理工具等多方面开展研究。Su 研究了在存贮和检索中应用神经网络和专家系统^[5], Clausing 等人总结了面向重用的策略管理工具设计^[6], 美国国防部的高性能知识库(High Performance Knowledge Base, HPKB)将本体论应用于知识重用^[7]。

本文以数字化设计环境中系统描述文件为载体, 研究其中的描述语言, 根据其语言结构提取系统数据, 为后续的系统重用提供数据支持。

1 数字化设计中的系统重用

1.1 数字化设计环境介绍

飞机设计是一个涉及机械、电子、软件、光学等多领域的极具复杂性的设计过程。鉴于此,飞机设计环境中必然要包括能进行多领域协同设计的工具。本课题组以飞控软件为设计对象,搭建了一个包括 Rhapsody, VC, Simulink 等软件的数字化设计环境(见图 1),该环境以 Rhapsody 为核心,并与其他工程软件实现无缝链接,组成了一个集需求分析、功能设计、结果实现、系统集成和整体验证的全过程设计环境^[1,3]。该系统开发环境以面向对象的模型驱动开发思想为指导,采用标准的模型驱动开发语言(Unified Modeling Language, UML),并能自动生成 C, C++, java 等代码,保证模型和代码的一致性。

1.2 三种模型简介

面向对象系统需要使用 3 种模型——类模型、状态模型和交互模型对系统从静态和动态两方面进行建模^[8]。

类模型:用于对系统的静态结构进行建模。其代表图是类图,描述类的属性、操作以及与其他类的关系。

状态模型:对系统时序的和行为的控制过程进行建模。其代表图是状态图,描述系统中某个对象类所具有的状态、事件以及在事件的激励下状态所进行的转换。

交互模型:对系统中对象间的协作进行建模。其代表图是用例图、交互图和活动图^[9]。用例图描述参与者与系统间进行的交互;序列图描述对象间基于时间顺序的交互;活动图描述系统行为执行步骤的控制流。

3 种模型在系统中的作用,简单来说,系统运行就是利用数据结构(类模型),按时间(或事件)的操作顺序(状态模型),在对象间传递数据的控制过程(交互模型)^[8]。

3 种模型几乎囊括了面向对象系统的全部信息,获取 3 种模型的信息即可理解系统的主要功能,这样就为重用系统部分功能找到了数据源。

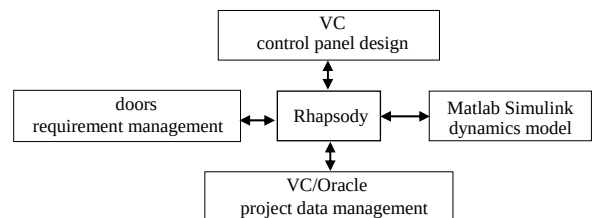


Fig.1 Architecture of digital design environment
图 1 数字化设计环境体系架构

1.3 系统重用

系统重用,就是在开展新系统设计时,参考或应用已有的、相似的系统部分功能的过程。重用的前提是存在相似的系统,同时,还要能够全面地、准确地了解该系统的全部或部分功能。由于该系统已事先存储在介质中,这就需要通过数据管理系统提取系统的相关信息。鉴于该设计环境采用面向对象的开发思想,所以获取系统的 3 种模型信息便成为实现系统重用的重要前提。

2 模型信息提取

为了从重用角度分析被重用系统的信息,如,查询某个类与多少个类有数据交互,需要利用管理系统获得被重用系统的信息。鉴于 3 种模型信息几乎构成了系统的全部,提取 3 种模型信息便成为目前研究的主要内容。本节分别以类图、状态图和序列图为例,研究 3 种模型信息在存储介质中的组织结构。

2.1 系统存储单元

该数字化设计环境以 Rhapsody 软件为核心,其他软件与其无缝链接,所有设计数据都统一在 Rhapsody 的存储文件中。设计数据以包为单位进行组织,包中可以包括若干个包和若干个类,每个包都有一个存储文件(*.sbs),详细记录本包内的数据结构,包括类的描述和子包的描述。

数据在包文件中以树形结构组织,并由‘{’和‘}’分层显示。*.sbs 文件第 1 层主要结构如下:

```
{ ISubsystem
- _ownerHandle = {IHandle}           //上层包信息
- _id = GUID 58bc6f34-d910-4787-a10e;
- _name = "ArchitecturalDesignPkg";
- Declaratives = {IRPYRawContainer}    //图信息
- Classes = {IRPYRawContainer}         //类信息
}
```

2.2 类图存储结构

类的所有信息都存储在-Classes={ }内。此处从类属性、类操作和类关系 3 个方面分析其存储结构。

1) 类属性

属性的信息存储在-Classes={ }内的-Attrs={ }中, 一个{IAttribute}结构描述一个属性信息, 包括类型、名称和访问权限, 其相应字段如下:

```
IAttribute/_name = "controlMode";    记录属性名称;
IAttribute/_typeOf/_name = "int";    记录属性类型;
IAttribute/_protection = iPublic;    记录属性访问权限;
```

如上信息描述的内容为: Public int controlMode;

2) 类操作

操作信息存储在-Classes={ }内的-Operations={ }中, 一个{ IPrimitiveOperation}结构描述一个操作信息, 包括名称、访问权限和执行的动作, 其相应字段如下:

```
IPrimitiveOperation/_name = "Print";    记录操作的名称;
IPrimitiveOperation/_returnType/_name = "void";    记录操作的返回值类型;
IPrimitiveOperation/_protection = iPrivate;    记录操作的访问权限;
IPrimitiveOperation/_itsBody/_bodyData = Printf("Hello World!");    记录操作的动作;
```

如上信息描述的内容为: iPrivate void Print() { Printf("Hello World!"); }

3) 类关系

类之间的关系分为关联、泛化和依赖 3 种, 字段相应取值见表 1。

表 1 关联、泛化和依赖关系与对应字段的取值

Table1 Class relations and correlative values

class relation	generalization	dependency	association
value	inheritances	dependencies	associations

其中关联又分为 n 元关联、导向关联、组合聚集和共享聚集等, 各字段相应取值见表 2。

表 2 关联关系与字段取值

Table2 Association and correlative values

association	n-ary association	directed association	aggregation	composition
- _m2Class	"IAssociation End"	""	—	—
- _linkType	Assoc	Assoc	Aggreg	Composition

2.3 状态图存储结构

状态图中的所有信息都存储在-Classes={ }内的-StateCharts={IRPYRawContainer 结构中。其中, 需要关注的是状态、事件以及在事件的激励下状态发生的转换。

1) 状态的存储结构

状态的信息在-States={IRPYRawContainer 结构中, 一个{IState}结构描述一个状态信息, 其重要字段如下:

```
IState /_id=GUID 6b8d66f5-c75a-44ec-bcf6-4f8a4847f9a1;    记录状态在工程中的唯一标识;
IState /_name = "S1";    记录状态的名称;
```

2) 事件及转换的存储结构

事件作为转换条件与转换共同组织, 在-States={IRPYRawContainer 内的-Transitions={IRPYRawContainer 中, 一个{ITransition}结构描述一个状态转移, 包括源状态、目的状态和激励事件, 其相应字段如下:

```
ITransition/_itsLabel/_itsTrigger/_body = "evStart";    记录激励的事件;
ITransition/_itsTarget = GUID 2db6a45b-3e54-458b-90bb-9070ae91aa12;    记录转移的目标状态;
ITransition/_itsSource = GUID 6b8d66f5-c75a-44ec-bcf6-4f8a4847f9a1;    记录转移的源状态;
```

如上信息表示: (2db6...)标识的状态在事件 evStart 激励下转移到(6b8d...)标识的状态。

2.4 序列图的存储结构

序列图中需要关注的是交互的对象和交互的消息。由于序列图描述多个对象的交互, 所以其是隶属于“包”而非“类”。其信息组织在-Declaratives={IRPYRawContainer}中, 如 2.1 节.sbs 包结构代码第 5 行所示。而每个序列图则由 Declaratives 内的一个{IMSC}结构描述。

1) 对象的存储结构

对象信息组织在 {IMSC/ m_pICollaboration 内的 {IClassifierRole} 中, 一个 {IClassifierRole} 结构描述一个对象的信息, 包括对象名称和对应的类, 其相应字段如下:

IclassifierRole/ _id = GUID 1538274e-f07d-45f7-9837-ffffb577fbd5;

记录序列图中对象的标识;

IclassifierRole/ _name = "c_reg";

记录对象的名称;

IclassifierRole/ m_pBase/ _id = GUID 03a00c10-1d96-4e58-bc65-de90;

记录对象类的标识;

2) 消息传递的存储结构

由于序列图中以时间顺序传递消息, 所以存储结构也是以时间顺序组织消息, 包括消息的发送者和接收者。其信息组织在 {IMSC/ m_pICollaboration 内的 Messages 中, 一个 {IMessage} 结构描述一个消息信息, 包括发送端、接收端和传递的消息, 其相应字段如下:

IMessage/ m_pReceiver/ _id = GUID ed61d944-287d-4c6f-ac03-9fc5536a35b5;

记录接收对象标识;

IMessage/ m_pSender/ _id = GUID 2ac20702-bba4-4ddc-8dce-3443466ee4ca;

记录发送对象标识;

IMessage/ m_pFormalMessage/ _id = GUID a0416dae-759c-42fb-8b25-865dc01f058e;

记录消息标识;

3 信息提取结果

本文以系统中的 3 个图——类图、状态图和序列图为例, 通过 VC 工具提取信息, 验证从系统的信息存储文件.sbs 包中提取 3 种模型的信息的过程。

3.1 提取类图信息

这里以飞控系统的组成为例, 设计了相应的类图, 其结构如图 2 所示。

根据 2.2 节提供的类图信息提取方式, 得到的结果如图 3 所示。

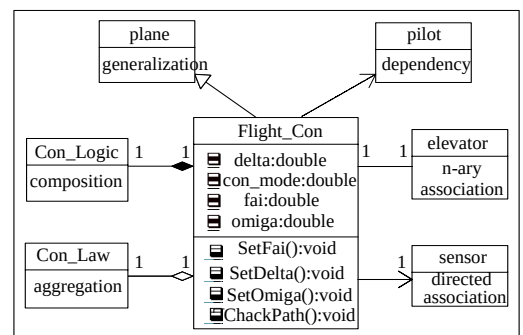


Fig.2 Relations between classes of flight control system

图 2 飞行控制系统类图组成关系

Visib	Type	Name
iPublic	double	delta
iPublic	double	con_mode
iPublic	double	fai
iPublic	double	omiga

attributes from FCC

Visib	Return	Name
iPublic	void	SetFai
iPublic	void	SetDelta
iPublic	void	SetOmiga
iPublic	void	ChackPath

operations from FCC

C_name	C_relation
Con_Logic	Composition
Con_Law	Aggregation
Sensor	D_association
Elevator	n_Association
Plane	Generalization
Pilot	Dependency

classes' relation from FCC

Fig.3 Extraction of attributes, operations and classes' relation from Flight Control Classes

图 3 飞控类中的属性、操作和类关系提取结果

通过对比可知, 提取的信息与原类图所要表达的设计意图相符。这样可以从存储文件中提取出系统的组成单位——类的全部结构信息, 包括类属性、类操作和所有的类关系。这样可保证系统静态信息的完整性。

3.2 提取状态图信息

下面以飞控类从起飞到着陆的状态转换为例, 从其中提取信息, 其结构如图 4~图 5 所示。

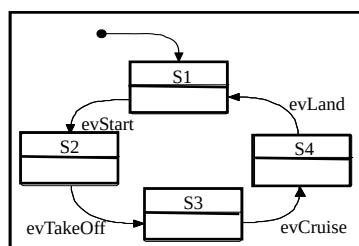


Fig.4 Statechart of Flight Control Classes

图 4 飞控类状态图

S_state	Event	D_state
(sour)	(NULL)	S1
S1	evStart	S2
S2	evTakeOff	S3
S3	evCruise	S4
S4	evLand	S1

Fig.5 Extraction of statechart

图 5 状态图信息提取结果

由提取结果可知, 从存储文件中可以获得相应类的所有状态、可以响应的事件及状态在事件的激励下发生的转移。这样可能获得相应类动态行为功能, 便于提取系统动态信息。

3.3 提取序列图信息

下面以控制逻辑对象和飞控对象针对飞机起飞时的逻辑控制过程为例,从中提取主要信息,其结构如图6~图7所示。

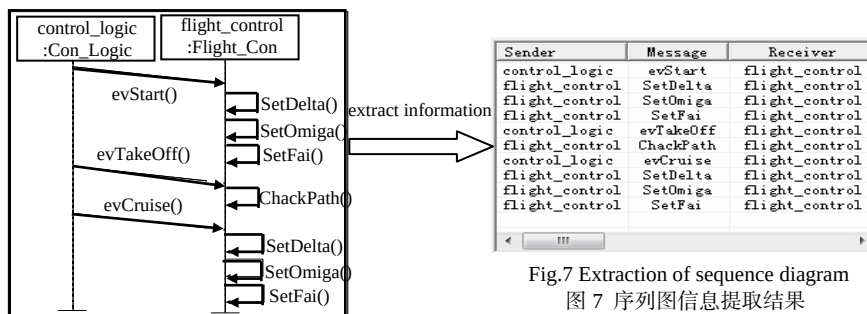


Fig.6 Sequence diagram between F_C and C_L

图6 飞控类与控制逻辑类的序列图

Fig.7 Extraction of sequence diagram

图7 序列图信息提取结果

从提取结果可知,从存储文件中可以提取原序列图所有表达的内容。提取的内容包括发生交互的类及在类之间传递的消息。从数量上看,类之间传递消息越多说明两个类联系得越紧密,这种紧密程度在一定程度上决定着相应类被重用的可能性。

4 结论

作为数字化设计的重要组成部分,系统重用需要对存储介质中的相似系统进行重用。对于被重用的系统,为了便于从重用角度理解系统信息,需要利用管理系统分析被重用系统的结构、功能信息,而基于UML的面向对象系统,其结构、功能信息由3种模型描述,所以提取3种模型信息即为提取系统信息。本文分别以3种模型的代表图为例,通过VC工具提取其中主要的信息,根据验证,其提取结果完全符合UML的设计意图,这表明通过对.sbs存储文件分析可以了解被重用系统的结构、功能信息,为进一步实现系统重用提供了前提。

参考文献:

- [1] 刘兴华,曹云峰. 支持模型驱动开发的飞控系统数字样机设计[J]. 计算机集成制造系统, 2011,17(1):23-29.
- [2] 王玉,邢渊,阮雪榆. 机械产品设计重用策略研究[J]. 机械工程学报, 2002,38(5):145-148.
- [3] 何火军,曹云峰,刘兴华. 无人机飞行控制系统虚拟样机平台[J]. 应用科学学报, 2009,27(6):637-643.
- [4] 邸彦强,李伯虎,柴旭东,等. 多学科虚拟样机协同建模与仿真平台及其关键技术研究[J]. 计算机集成制造系统, 2005,11(7):901-908.
- [5] Su D. Application of Expert System and Artificial Neural Network for Design Knowledge Retrieval[C]// Proceedings of Engineering Design Conference'98 on Design Reuse. London:Professional Engineering Publishing Limited, 1998:135-139.
- [6] Clausing D P, Andrade R. Strategic Reusability[C]// Proceedings of Engineering Design Conference'98 on Design Reuse. London: Professional Engineering Publishing Limited, 1998:98-101.
- [7] 孔繁胜. 知识库系统原理[M]. 杭州:浙江大学出版社, 2000.
- [8] Michael Blaha, James Rumbaugh. UML面向对象建模与设计[M]. 2版. 车皓阳,杨眉,译. 北京:人民邮电出版社, 2006.
- [9] 薛振伟,吴志杰,杨德成. UML2.0对模型驱动架构的支持[J]. 信息与电子工程, 2007,5(4):296-300.

作者简介:



周在华(1987-),男,黑龙江省齐齐哈尔市人,在读硕士研究生,主要从事数字化技术研究.
email:zhouzaihua_8@126.com.

曹云峰(1964-),男,浙江省湖州市人,教授,博士生导师,主要从事飞行控制、计算机视觉和数字化设计与虚拟样机研究。

王彪(1975-),男,辽宁省铁岭市人,副教授,硕士生导师,主要从事飞行控制、计算机视觉、精确制导等研究。

庄丽葵(1963-),女,南京市人,助理研究员,副工程师,主要从事无人机飞行控制与导航研究。