计算机科学
COMPUTER SCIENCE

一种基于变量依赖关系的需求模型耦合分析方法

尹伟, 窦霖, 高忠杰, 王立松, 孙倩

引用本文

尹伟, 窦霖, 高忠杰, 王立松, 孙倩. 一种基于变量依赖关系的需求模型耦合分析方法[J]. 计算机科学, 2025, 52(7): 58-68.

YIN Wei, DOU Lin, GAO Zhongjie, WANG Lisong, SUN Qian. Method for Coupling Analysis of Requirements Models Based on Variable Dependency Relationships [J]. Computer Science, 2025, 52(7): 58-68.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

基于深度强化学习的四旋翼无人机自主控制方法

Autonomous Control Algorithm for Quadrotor Based on Deep Reinforcement Learning

计算机科学, 2023, 50(11A): 220900257-7. https://doi.org/10.11896/jsjkx.220900257

一种结构关系一致的对比聚类方法

Contrastive Clustering with Consistent Structural Relations

计算机科学, 2023, 50(9): 123-129. https://doi.org/10.11896/jsjkx.220700288

一种文本-图像增强的多模态知识图谱嵌入方法

Multimodal Knowledge Graph Embedding with Text-Image Enhancement

计算机科学, 2023, 50(8): 163-169. https://doi.org/10.11896/jsjkx.220700216

基于图像数据耦合识别的输电线路安全风险评估方法

Image Recognition Method of Transmission Line Safety Risk Assessment Based on MultidimensionalData Coupling

计算机科学, 2023, 50(6A): 220500032-6. https://doi.org/10.11896/jsjkx.220500032

基于多核CPU的无锁并行Semi-naive算法

Lock-free Parallel Semi-naive Algorithm Based on Multi-core CPU

计算机科学, 2023, 50(6): 29-35. https://doi.org/10.11896/jsjkx.220800050

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241000092

一种基于变量依赖关系的需求模型耦合分析方法

尹 伟[1] 窦 霖[2] 高忠杰[1] 王立松[3] 孙 倩[1]
1 中国航空无线电电子研究所 上海 200241
2 南京中兴公司 南京 210000
3 南京航空航天大学计算机科学与技术学院 南京 211106

摘 要 机载软件是典型的安全关键软件,其开发和验证过程在航空业内受到严格控制。机载软件的复杂性和多样性使得需求分析成为关键的研究内容,特别是在需求验证阶段,需要关注系统部件之间的交互方式以及变量之间的依赖关系是否满足预期。对此,提出了一种面向机载软件需求的基于VRM模型的耦合分析方法,定义了需求中变量之间的依赖关系,通过度量指标来衡量系统部件之间的数据耦合和控制耦合。为了弥补基于需求层面的耦合分析技术的不足,使用VRM模型作为形式化需求模型,在系统需求层次进行建模和分析,有效支撑了DO-178C对数据耦合、控制耦合和软件部件的相关要求。同时,提出了基于变量间层次依赖关系的耦合度量方法,将变量间的关系定义为n叉依赖树的结构,并使用一系列算法对变量划分权重。通过构造变量矩阵、需求变量依赖树等方法对耦合度进行度量,形成了基于变量依赖关系的数据耦合和控制耦合分析原型。该研究突破了基于需求层面的耦合分析技术,为复杂系统的设计提供了帮助,提高了机载软件开发的质量和可靠性。

关键词:VRM模型;数据耦合;控制耦合;需求变量依赖树
中图分类号 TP301;V19

Method for Coupling Analysis of Requirements Models Based on Variable Dependency Relationships

YIN Wei[1],DOU Lin[2],GAO Zhongjie[1],WANG Lisong[3] and SUN Qian[1]
1 China Aeronautical Radio Electronics Research Institute,Shanghai 200241,China
2 Zhongxing Telecom Equipment,Nanjing 210000,China
3 College of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China

Abstract The airborne software is a typical safety-critical software,and its development and verification process is strictly regulated in the aviation industry. The complexity and diversity of airborne software make requirements analysis a crucial area of research. Particularly during the requirements verification phase,it is essential to focus on the interaction patterns between system components and whether the dependencies between variables meet expectations. Therefore,This paper proposes a coupling analysis method based on the VRM model for airborne software requirements,which defines the dependencies between variables in the requirements,and measures the data coupling and control coupling between system components through metrics. In order to solve the deficiencies of the coupling analysis technique based on the requirement level,this paper uses the VRM model as a formal requirement model to model and analyze the system requirements at the system requirement level,which effectively supports the relevant requirements of DO-178C for data coupling,control coupling and software components. Secondly,this paper proposes a coupling measurement method based on the hierarchical dependency relationship between variables,defines the relationship between variables as an n-forked dependency tree structure,and uses a series of algorithms to classify the weights of the variables to measure the coupling degree by constructing variable matrices,demand variable dependency trees,etc.,which forms a prototype for analyzing the data and control coupling based on the dependency relationship of the variables. The research in this paper breaks through the coupling analysis technology based on the requirement level,which provides help for the design of complex systems and improves the quality and reliability of airborne software development.

Keywords VRM modeling,Data coupling,Control coupling,Demand variable dependency trees

1 引言

随着航空技术的不断发展,机载软件在飞行系统中扮演着越来越重要的角色。当前,国内机载软件的研发主要基于RTCA/DO-178C标准。机载软件系统的复杂性和多样性使得需求分析成为关键的研究内容,特别是需求模型中的耦合

到稿日期:2024-10-18 返修日期:2025-03-13
基金项目:工信部民机预研项目(YY-23171-ZX)
This work was supported by the Civil Aircraft Pre-Research Program(YY-23171-ZX).
通信作者:尹伟(yinw008@avic.com)

性对系统的性能和安全性产生着重要影响。同时，耦合关系也是衡量软件体系结构可靠性的重要指标，对于安全关键的机载软件，数据耦合和控制耦合分析尤为重要。如果来自其他软件部件的数据和控制信息处理不当，可能会导致错误的行为，给耦合的软件部件造成操作问题，从而使它们处于异常模式。因此，在机载软件研制过程中，需要考虑软件的数据耦合和控制耦合特性，并对其进行合理约束，以确保交互和依赖关系的正确性和覆盖范围的完整性。

数据耦合和控制耦合是复杂软件设计中常见的情况，目前国外对软件耦合关系的分析与测试有较多深入的研究。DO-178C[1]标准对高安全等级软件(DAL A/B/C)提出了数据耦合和控制耦合分析的要求，来确保数据传递和控制信号传递的正确性、安全性和完整性。CAST-19[2]描述了数据耦合与控制耦合分析的目的，即度量集成测试的充分性，以证明软件部件能够以设计者的预期方式影响其他软件部件，且不能以任何非预期的方式影响其他软件部件。Hellhake 等[3]通过研究不同组件及接口的耦合度量与汽车系统集成测试中故障分布的关联，提出耦合度量与故障倾向间存在正相关关系。Meyers[4]提出了一种自动测试生成器(ATG)，它可以解析飞机电子设备操作软件的计算机可执行设计模型，以识别设计模型中功能模块之间的数据耦合和控制耦合，并生成一系列测试条件来测试操作软件是否满足数据耦合和控制耦合。Rierson 等[5]从 DO-178C 的角度将控制耦合的概念与软件工程的控制和内容耦合的概念相关联，而数据耦合的思想可以与数据、标记、公共和外部耦合相关联。Maia 等[6]介绍了一种自底向上的验证策略，使用不带存根的被调用函数，生成了一个增量集成，从而减少了检查函数间耦合的工作量。Estrada 等[7]提出了基于模型的设计和代码自动生成的嵌入式控制系统开发方法，使工程师能够使用可执行模型作为设计、模拟以及软件验证和确认的基础来创建高级嵌入式软件系统。Chen 等[8]从需求描述关注点、需求描述维度、需求分析要素等方面，分析现有嵌入式需求描述语言的表达能力，考虑采用智能方法对嵌入式系统需求描述方法表达能力的要求。Priyalakshmi 和 Latha 介绍了一种基于耦合和内聚的软件重用性评估方法，对 Python 包中模块之间的导入关系和模块内部的函数调用关系进行静态分析，使用 Page Rank 算法对模块耦合得分进行加权，并根据这些得分计算包的重用性得分[9]。Abdurazik 和 Offutt 利用耦合度测量来估计桩代码的编写难度，并打破类之间的依赖循环，解决了类集成和测试顺序(CITO)问题[10]。Hellhake 等分析了软件组件之间的耦合度和接口复杂度与系统集成测试中发现的故障分布之间的关系，使用了皮尔逊相关系数和斯皮尔曼等级相关系数来评估不同耦合度度量与故障分布之间的线性和非线性关系[11]。Priyalakshmi 等[12]通过将 Python 包中所有模块的耦合度分数和内聚度分数累加，来评估一个包的可复用性分数。Chhabra 和 Parashar 提出了一种基于数据聚类的方法，用于预测类的重用性模式和耦合行为。该方法使用 N 维向量空间模型和基于权重的文档聚类技术，对 Java 应用程序中的类进行聚类，并采用基本的 K 均值聚类算法和余弦相似度度量来寻找具有相似特征的类簇，然后对每个簇进行评价和排名[13]。Poshyvanyk 等使用了基于信息检索的耦合度量方法来进行影响分析。该方法采用潜在语义索引(LSI)作为信息检索技术，被应用于面向对象软件系统的上下文中来测量不同类之间的概念耦合度[14]。Zhang 等[15]通过在 Coq 定理证明器中使用并发精化分离逻辑(CSL-R)框架，对某抢占式微内核操作系统的互斥量模块进行了代码级形式化建模和功能正确性验证，扩展了验证框架以表达用户假设条件，并发现了代码中存在的问题。Bavota 等[16]探讨了不同类型的耦合度量(结构、动态、语义和逻辑)如何反映开发者对类之间耦合的感知，并使用一种基于信息检索技术的语义耦合度量(CCBC)进行比较。Sherriff 等[17]提出了一种基于历史变更记录的影响分析方法，其利用奇异值分解(SVD)生成文件的关联簇，然后根据风险识别和优先级确定系统修改可能影响的文件。Hu 等[18]提出了一种基于四变量模型的系统安全性建模与分析方法，旨在为安全关键系统工程领域提供一种有效的系统安全性分析和验证方法。未来研究方向包括：耦合度量在代码审查、代码异味检测和变更预测等领域的应用；以及如何将不同类型的耦合度量方法结合起来，以更全面地评估软件耦合[19]。

通过这些研究，可以看到耦合分析在软件工程领域的广泛应用和重要性。从软件设计、开发、集成、测试到维护和演化，耦合度量和优化方法都可以为提高软件质量、可维护性和可重用性等提供有力的支持。但是，以上对软件系统的耦合分析大多数是基于代码层级的。而在软件需求层面，如果在需求定义时发生错误，后续的代码编写和软件测试阶段会遇到更多的问题，并且更难以发现和处理，需要不断回溯甚至修改需求。对于高安全性和高可靠性的机载软件来说，需求定义的错误对整个生命周期的软件设计和开发影响巨大。此外，DO-178C 中所有的验证活动都是基于需求的，基于代码结构的耦合分析难以满足适航局的要求。VRM(Variable Relationship Model)是形式化变量关系模型的简称，是一套用于系统需求建模与分析的算法框架[20]。该框架将系统需求以表格的形式表达，通过描述需求的状态变量以及行为函数，使得需求模型具有状态机的特点，能够很好地描述诸如航空电子软件系统、飞行制导系统领域的需求特征。本文提出了一种基于 VRM 的需求模型耦合分析方法，该方法基于需求的模型，通过定义需求中变量间的依赖关系以及构建变量矩阵和需求变量依赖树，在系统需求层次上描述所需的系统行为及软件行为。该方法指明系统需求以及系统的输入和输出硬件接口，对软件需求进行限制，包括系统与外界的交互行为以及系统内部的状态、模式变化的行为特征描述等，弥补了基于需求层面的耦合分析技术的不足。同时，本文采用层次依赖关系对变量划分权重，以度量系统部件间的数据耦合和控制耦合。这一研究突破性地改进了需求层面的耦合分析技术，为复杂系统的设计提供了有益帮助，并有效提高了机载软件开发的质量和可靠性。

2　基于 VRM 模型的耦合定义

DO-178C 标准要求开发人员对软件部件之间的数据耦合和控制耦合进行分析，以确保软件系统满足航空电子设备的安全要求，并给出了以下相关定义。

1) 数据耦合：指一个软件部件依赖于非其完全控制的数据，即部件之间通过数据传递产生关联。

2）控制耦合：指一个部件对另一个部件执行流程的影响程度，例如通过控制标志、状态切换等方式干预其他部件的运行逻辑。

3）部件：指一个独立的部件、部件组合、子组件或单元，其在系统中执行特定功能。

本文使用 VRM 模型作为形式化需求模型，能够很好地满足 DO-178C 的工程要求。通过对原始需求进行解析，可以得到耦合分析所需的输入变量、输出变量、中间变量和模式类，将这些变量分布到 VRM 模型部件中，VRM 模型部件通过条件表、事件表和模式转换表之间交换变量的值来描述部件之间的交互和依赖。

2.1 VRM 形式化模型

形式化变量关系模型 VRM 在四变量 FVM（1978 年由 Parns 等提出，是一种用于指明需求的方法）模型的基础上，引入了另外 4 个模型元素概念：模式类、中间变量、条件和事件。模式类可以理解为一个状态机，其状态称为系统模式，其转换由事件触发。中间变量是输入变量、模式或其他中间变量的任何函数关系映射之后形成的变量，它的存在是为了更好地描述系统的需求层次。条件是关于系统状态的具体取值或范围。当系统实体更改为特定的值，或者从特定范围内变更到范围之外时，就会发生事件。本文将中间变量、模式类与监督变量、受控变量、输入变量、输出变量统称为系统中状态变量。表 1 列出了 VRM 模型中的各个模型元素的具体定义。

表 1 VRM 模型元素

Table 1 VRM model elements

VRM 元素	定义
监督变量	系统观察到的并对系统行为做出回应的变量
控制变量	系统用于控制外部环境并做出变化的变量
输入变量	软件读入的变量，此变量由输入设备转换监控变量所得
输出变量	软件读出的变量，此变量由输出设备转换控制变量所得
中间变量	由于实际的系统结构一般比较庞大，采用层次结构可以更好地对整体系统进行建模和分析，中间变量位于输入变量和输出变量之间
模式	说明事物结构的主观理性形式
状态变量	由监督变量、受控变量、输入变量、输出变量、中间变量和模式类组成的变量
条件	评估为 true 或 false 的布尔表达式
事件	在条件值更改时发生，每个事件发生都是瞬时的和原子的
表格函数	以二维表格的形式描述状态变量之间关系的数学函数

VRM 模型是由状态变量、变量类型、变量值域、条件、事件、表格函数组成的六元组。其中，VRM 模型中的表格函数包括条件表函数、事件表函数和模式转换表函数三大类，且每一类都有相应的形式化语义定义。VRM 形式化需求模型各个状态变量之间的依赖关系如图 1 所示。

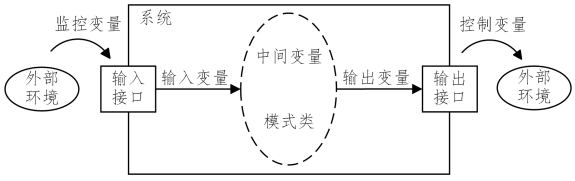


图 1 状态变量依赖关系

Fig. 1 State variable dependencies

2.2 VRM 模型部件

软件部件（也称为模块）代表了软件系统的一部分，而不是整个系统。如果一个机载软件仅由一个模块组成，软件程序或许足以正常运行且无异常行为。然而，鉴于嵌入式机载软件具有规模庞大与复杂性高等特点，一个仅由独立模块组成的系统将导致故障率激增并降低维护效率。因此，依据系统功能将其划分为不同部件是必要的，这些部件相互交互并依赖，以满足软件需求。构建模块或部件是进行数据耦合和控制耦合分析的基础，而耦合分析的目的是为这些部件的交互和依赖关系提供度量和保障。下面将介绍基于 VRM 模型的软件部件的相关定义。

VRM 模型由一组模型部件的集合组成，可表示为 $CPT=\{cpt_1,cpt_2,\cdots,cpt_n\}$。如图 2 所示，模型部件包含状态变量和表格函数，其中状态变量由输入变量 IV（Input Variables）、输出变量 OV（Output Variables）、中间变量 TV（Intermediate Variables）和模式类 MC（Mode Class）组成；表格函数由条件表（Condition Table）、事件表（Event Table）和模式转换表（Mode Transition Table）组成。表格函数通常是需求的表现形式，函数通过状态变量的不同组合与取值来满足需求的定义。VRM 模型通过条件表和事件表描述软件系统的数据流，通过模式转换表描述系统的控制流。部件之间的数据耦合通过条件表和事件表中状态变量的交互来确定，控制耦合通过模式转换表中状态变量的交互来确定。

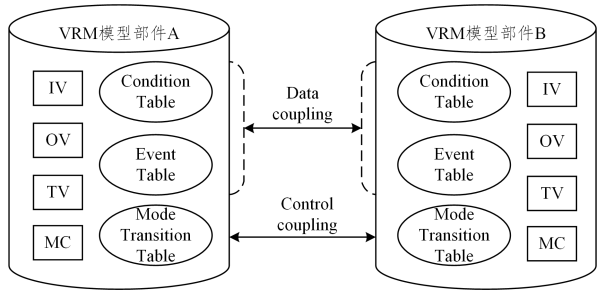


图 2 VRM 模型部件交互图

Fig. 2 VRM model component interaction diagram

2.3 VRM 数据耦合定义

数据耦合表明了不同模块或部件之间通过共享数据信息实现相互通信和交互。在机载软件系统中，不同的部件可能会需要共享某些数据，以完成系统功能的协作，这些共享的数据信息可以是传感器读数、状态信息、命令参数等。

VRM 模型部件通过条件表、事件表中状态变量的组合和取值来描述模型部件之间数据的交互和依赖，这种依赖关系即为数据耦合。例如，在飞行制导系统中，有一条原始需求如图 3 所示，该需求定义了何时关闭飞行指导仪，将其从主飞行显示器上移除。

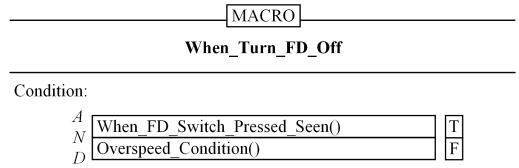


图 3 数据耦合需求实例

Fig. 3 Example of data coupling requirements

2.4 VRM 控制耦合定义

控制耦合表明了不同模块或组件之间通过共享控制信息实现相互调用和协作。在机载软件系统中，不同的部件可能

需要共享控制逻辑，以协调各自的行为并实现系统功能，这些共享的控制信息可以是命令、状态信息、调度信息等。

VRM 模型部件之间的控制耦合通过模式转换表中状态变量的交互来确定。如图 4 所示，模式 ROLL 表示飞机是否处于滚动状态，它有 Cleared 和 Selected 两种模式，用于说明 ROLL 状态是选中还是非选中。在对这条需求规范化后，得到了 4 个状态变量：mcROLL，tSelectROLL，tDeselectROLL 和 mcIsThisSideActive。模式类 mcROLL 由 Cleared 状态转变为 Selected 状态的条件是：tSelectROLL = TRUE 并且 mcIsThisSideActive = TRUE；mcROLL 由 Selected 状态转变为 Cleared 状态的条件是：tDeselectROLL = TRUE 并且 mcIsThisSideActive = TRUE。

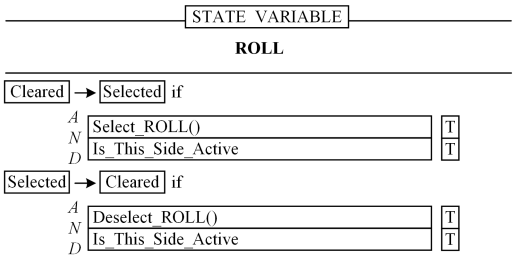


图 4　控制耦合需求实例

Fig. 4　Example of control coupling requirements

其中 mcROLL，tSelectROLL 和 tDeselectROLL 属于 Flight Modes 部件，mcIsThisSideActive 属于 Independent Mode 部件，这两个部件通过该需求模型形成关联，mcROLL 的值会因为 tSelectROLL，tDeselectROLL 和 mcIsThisSideActive 的数值变化而改变，进而实现对飞机滚动状态的控制。本文将这种控制流的依赖关系定义为部件之间的控制耦合。

3　分层次的耦合类型

系统由多个部件组成，而每个部件又包含多个变量。为了准确定义它们之间的依赖关系，可以考虑不同层次的耦合度。首先是部件内部的耦合度，它描述了部件内部变量之间的依赖程度。其次是部件之间的耦合度，指的是一个部件的变量对属于另一个部件的变量的依赖程度。最后是部件与系统之间的耦合度，它表示一个部件中的变量对整个系统中所有部件的变量的依赖程度。

通过定义和计算不同层次的耦合度，可以更好地理解系统的结构和功能。如表 2 所列，某个系统中有 3 个部件和相对应的 5 个变量，其中 1 表示行变量依赖于列变量，0 表示没有。下面将介绍在这样一个简单的系统中，该如何定义和度量部件内部、部件之间和部件与系统的耦合。

表 2　耦合关系示意表

Table 2　Schematic coupling relationship

系统		部件 1		部件 2		部件 3
		变量 A	变量 B	变量 C	变量 D	变量 E
部件 1	变量 A	0	1	1	0	1
	变量 B	0	0	1	1	1
部件 2	变量 C	1	0	0	0	0
	变量 D	0	0	1	0	0
部件 3	变量 E	0	0	1	1	0

3.1　部件内部的耦合

部件内部的数据耦合和控制耦合是评估部件复杂性和内聚性的重要指标：数据依赖关系，即组件内部各个功能之间如何通过数据交换实现相互作用；控制依赖关系，指的是部件内部的控制逻辑和行为模式之间的关联程度，即部件内部各个功能如何通过控制结构协同工作。通过分析部件内部的耦合情况，可以识别出关键变量、强耦合的变量组合，有助于理解部件的结构和行为特点。

在本文中，部件内部的耦合度量方法是将一个部件内部的变量筛选出来，只考虑这些变量之间的依赖关系。如在表 3 中，部件内部的耦合可以理解为：在部件 1 中，变量 A 依赖于变量 B 和变量 C，变量 B 又依赖于变量 C。那么部件 1 内部的耦合度即为变量 A 对变量 B、变量 C 的依赖度与变量 B 对变量 C 的依赖度之和。

1）部件内部数据耦合分析：关注部件内部功能之间数据流。例如，在机载软件中，一个部件是姿态控制功能模块，通过分析该部件内部的输入变量、输出变量和中间变量之间的依赖关系，可以确定不同功能之间的数据耦合程度，更好地理解该部件的数据交互方式和数据传递路径。

2）部件内部控制耦合分析：关注部件内部的控制结构和行为模式。如一个部件是自动驾驶功能模块，通过分析部件内部的控制结构和控制流，可以确定不同功能之间的控制依赖关系，进而理解各个功能是如何协同工作以实现自动驾驶功能。

3.2　部件之间的耦合

部件与部件之间的耦合分析关注的是不同部件之间的数据交互和控制逻辑关系。需要注意的是，部件之间的耦合关系是单向的，即部件 1 对部件 2 的耦合度与部件 2 对部件 1 的耦合度是不同的。部件之间的耦合可以揭示不同部件之间的相互依赖和协作方式，为系统设计和集成提供帮助。

部件之间的耦合度量方法是将系统中的任意两个部件单独列出来，找出他们之间的变量依赖关系。如在表 2 中，部件 1 的变量 A 分别依赖部件 2 的变量 C 和部件 3 的变量 E，部件 1 的变量 B 又依赖于部件 2 的变量 D。那么部件 1 对部件 2 的耦合度即为属于部件 1 的变量对属于部件 2 的变量的依赖度之和，部件 1 对部件 3 的耦合度同理。

1）部件之间的数据耦合分析：识别部件之间的数据流，包括输入、输出和中间变量。如一个部件是姿态控制功能模块，另一个部件是导航功能模块，通过分析两个部件之间的数据依赖关系，可以了解导航数据传递给姿态控制后会产生什么影响。

2）部件之间的控制耦合分析：识别不同部件之间的控制流，即模式类。如一个部件是姿态控制功能模块，另一个部件是引擎控制功能模块，分析这两个部件中的控制依赖，可以了解引擎控制与姿态控制之间的相互作用。

3.3　部件与系统的耦合

部件与系统之间的耦合分析涉及评估部件对整个系统的数据交互和控制逻辑关系。通过分析部件与系统之间的数据耦合和控制耦合，可以更好地理解部件在整个系统中的作用和影响，识别出关键部件。

部件与系统的耦合度量方法是将系统中的一个部件与所有部件都进行比较，包括这个部件内部的耦合和部件与部件之间的耦合。如在表 2 中，部件 1 内部的耦合度为 c_1，部件 1 与部件 2、部件 3 的耦合度分别为 c_2 和 c_3，那么部件 1 与系统的耦合度即为 $c_1+c_2+c_3$。

1）部件与系统的数据耦合分析：考虑部件与系统之间的数据交互关系。如在飞行控制系统中，部件可以是各个功能模块，而系统是整个飞行控制系统。通过分析各个部件与系统之间的数据流动和交换，可以确定各个部件在系统中的数据依赖关系。

2）部件与系统的控制耦合分析：研究部件的控制行为以及系统对部件的控制方式。如部件是各个功能模块，系统是飞行控制系统。通过分析控制信号的传递和处理，可以确定部件与系统之间的控制依赖关系和协作方式。

4 变量间的依赖关系与权重划分

本文重点研究系统变量之间的依赖关系。当一个软件部件修改了某个数据信息或控制信息后，与其耦合的部件也会受到影响，因为它们都使用同一份数据。这种依赖性可能会由于部件中的一个小变化，引起整个系统的连锁反应。变量间的依赖关系可以是直接的，也可以是间接的。直接依赖意味着部件之间直接共享数据或控制，间接依赖则是通过一个或多个中间部件来共享数据或控制。本文通过 VRM 模型筛选需求中的状态变量来建立模型，将变量之间的依赖关系表示为 n 叉树的结构，并为每个变量分配权重，通过度量指标来衡量软件部件之间的数据耦合和控制耦合。

4.1 变量间的层次依赖

需求变量依赖树是一种描述变量之间层次依赖关系的 n 叉树结构，它将系统需求中的各个变量按照其依赖关系组织起来，以便对变量之间的直接依赖与间接依赖进行分析和度量。在需求变量依赖树中，每个节点代表一个变量（即前文的状态变量），节点之间的边表示变量之间的依赖关系。对于树中的每一个节点来说，父节点变量与其子节点变量构成一条需求，父节点变量依赖于它的子节点变量，其含义是：当某一父节点变量的所有子节点变量组成的条件成立时，该父节点变量所在的需求得到满足。以下将父节点变量定义为需求变量，其子节点变量定义为条件变量。需求的阶数根据需求变量在 n 叉树中的层数划分，依赖树中的需求数目与前三阶的变量节点数一致。树中的节点既是自己所在阶数的需求变量，又是上一层父节点变量的条件变量。这种需求变量与条件变量的依赖关系，构成了需求变量依赖树模型。

图 5 给出了一个构造需求变量依赖树的示例，需求变量 A 与条件变量 B，C 构成第一条需求，$B-D$ 为第二条需求，$C-E-F$ 为第三条需求，以此类推。根据 n 叉树的层数划分需求的阶数，第一层为一阶需求，也称为根需求，它的具象化表示为：当条件变量 B 与 C 同时满足时，A 变量所处的需求得以实现。类似地，二阶需求共有两条：一是基于 B 节点变量的需求（简称需求 B），二是基于 C 节点变量的需求（简称需求 C）。需求 B 的实现条件为 D 变量满足相应条件，而需求 C 则要求 E 变量与 F 变量同时满足相应条件。

需求变量依赖树中的需求数目与前三阶的变量节点数一致，树中的节点既是自己所在阶数的需求变量，又是上一层父节点变量的条件变量，这种需求与条件的依赖关系构成了需求变量依赖树模型。本文只研究到 4 阶需求，再往下的需求依赖度相对于根需求来说很小，所产生的耦合效果也可大致忽略。

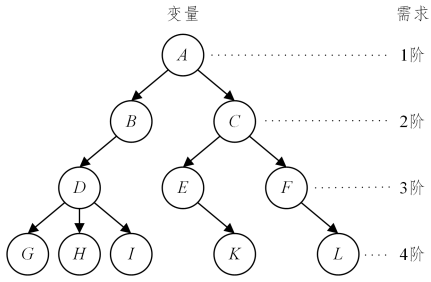


图 5 需求变量依赖树

Fig. 5 Requirement variable dependency tree

4.2 变量的权重划分

复杂系统通常具有高阶、多回路和非线性反馈结构，其特点包括：系统部件数量众多，结构和层次复杂，规模庞大。例如，现代机载软件的代码规模通常达到百万或千万行级别。复杂系统的运行方式复杂，涉及人机交互以及与环境的相互作用。航空电子系统是典型的复杂系统，在基于 VRM 模型的耦合度分析中，不能将所有状态变量的权重设为相同值。首先，人工定义出变量的初始权重系数，再根据变量在每条需求中出现的次数计算出变量标准差权重系数。最重要的是，根据需求变量依赖树模型将变量划分为不同层级，并为每层的变量设置不同的权重系数。结合以上 3 种方法，可以综合计算出变量的权重大小。

4.2.1 人工定义系数

人工定义系数（Manual Coefficient，MC）是在 VRM 建模时由人工进行定义的，用于表示变量在系统中的重要程度。在 VRM 模型中，人工定义系数可以通过以下方式进行定义。首先，根据需求，将状态变量划分为输入变量、输出变量、中间变量和模式类。然后，针对每个状态变量，根据其在系统中的重要程度，给出一个人工定义的系数。这个系数可以根据经验和专业知识进行设置，如对于输入变量和输出变量，系数可以根据其对系统性能的影响进行设置；对于中间变量，系数可以根据其在系统中的重要程度进行设置；对于模式类，系数可以根据其对系统控制的影响进行设置。最终，根据人工定义系数，可以计算出变量的标准差系数，用于表示变量在系统中的重要程度。

以航电系统为例，控制操纵杆的变量权值应设为高值，因为这是飞行员操纵飞行器的关键部件。相反，显示舱内温度的变量权值应设为较低值，因为即使温度显示出现误差，驾驶员仍可以通过其他途径估计大致的温度值。在具体实现过程中，可以将每个变量的权重初始值设为 1，然后根据建模人员对系统功能的深入了解，识别系统中的关键部件。对于重要的系统变量，可以给予较高的初始权值，从而更好地反映其在

整个系统中的作用和影响。这种权值分配有助于更好地理解系统中各部件的重要性，并为后续分析提供基础。

4.2.2　变量标准差系数

在对原始需求进行 VRM 模型建模后，得到了 n 个状态变量，这些变量分布在 m 条需求之中。需求与变量之间呈现出一对一、一对多或多对多的对应关系，如图 6 所示。

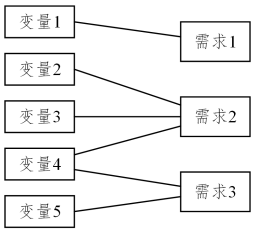


图 6　需求与变量关系示意图

Fig. 6　Schematic representation of the relationship between demand and variables

为了准确评估变量在系统中的重要性，仔细观察了需求中变量的出现次数。结果发现，若某个变量在需求中出现次数较多，其对系统的影响力和重要性往往相对较大。尤其当多个部件都涉及到该变量时，在耦合分析中，该变量的重要性显得尤为突出。鉴于此，本文提出了一种基于变量在需求中出现次数的权重定义方法，即变量的标准差系数（Standard Coefficient，SC）。标准差系数被认为是划分权重的一个关键参考指标，其计算式如下：

$$SC=\sqrt{X_i \div \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{N}}} \tag{1}$$

其中，$SC(i)$为变量 i 的标准差系数，X_i 表示变量 i 在所有需求中出现的次数，μ 为所有变量出现次数的平均值，N 是所有需求中变量的总个数。通过这种计算方法，首先得到需求中变量数目的标准差，再根据每个变量的出现次数的不同，对出现数目多的变量给定一个较高的标准差权重系数。

4.2.3　层级权重系数

本文针对 n 叉树中不同层级之间变量的依赖程度，为每一层的变量赋予相同的权重系数。在需求变量依赖树模型中，提出了一种加权策略，对不同层级的变量节点进行加权处理。简言之，权重系数与层级深度呈负相关，而与本层中变量节点数量呈正相关。根据权重系数，可以计算出根节点变量与其他阶数变量之间的耦合度，具体算法包括以下两个部分。

1）距离度：在需求变量依赖树中，变量节点距离根节点越近，对根需求的影响就越大，因为它们与根节点之间的依赖关系更为紧密，这种紧密的依赖关系也意味着这些节点之间的耦合度更高；反之，距离根节点越远的变量节点影响越小，因为它们与根节点之间的依赖关系相对较弱，这也意味着它们之间的耦合度相对较低。因此，在考虑需求的满足时，必须充分考虑变量之间的依赖关系和距离根节点的距离。具体计算式如下：

$$dep(i)=\frac{depth-F(i)+2}{depth} \tag{2}$$

其中，$dep(i)$表示节点 i 与根节点变量的距离度，$depth$ 为 n 叉树的整体深度，$F(i)$为依赖树中节点 i 的阶数。

2）复杂度：需求的满足涉及多个条件变量的相互作用，它们在依赖树中被视为节点，而树的层级则代表了变量的层次关系。节点在依赖树中依赖于其子节点，同时影响其父节点，导致某层级的复杂度随着变量节点数量的增加而增大，其对上一层级的影响也相应增大。在实际需求中，节点增加会导致条件变量数量的增加，从而增加满足需求的难度。针对高复杂度层级，通过增加相应的权重系数，可以更好地满足需求。具体计算式如下：

$$cmp(i)=\frac{L(F(i))}{\sum_{n}^{depth}L(n)} \tag{3}$$

其中，$cmp(i)$表示节点 i 所在阶数的复杂度，$F(i)$为依赖树中节点 i 的阶数，$L(n)$为计算依赖树中第 n 阶节点个数的函数。

综上以上两点，需求变量依赖树中变量节点 t 的层级权重系数（Hierarchical Coefficient，HC）可以通过式（4）设计，它涉及计算节点的距离度和复杂度，并将它们与树的阶数结合。

$$HC=(dep(i)+cmp(i))\times\frac{1}{2^{F(i)-1}} \tag{4}$$

其中，$HC(t)$表示节点 t 的权重系数，$dep(t)$表示节点 t 的距离度，$cmp(t)$表示节点 t 的复杂度，$F(i)$为依赖树中节点 i 的阶数。

上述方法可以为需求变量依赖树中的每个变量节点赋予一个合适的权重系数，从而在耦合分析中更好地体现出各变量节点之间的关联程度。此外，这种加权策略也有助于在需求分析过程中发现关键变量，更加准确地反映变量节点之间的依赖关系和耦合度。

5　数据耦合和控制耦合度量

上文详细介绍了基于 VRM 模型的数据耦合和控制耦合类型，其中数据耦合的度量与输入变量、输出变量和中间变量相关，控制耦合的度量与模式类相关。下面介绍耦合度的计算方法，共分为 6 个步骤。

5.1　构造部件变量关系矩阵

在对原始需求进行 VRM 模型建模之后，得到了部件与变量之间的对应关系。为了便于研究与分析，采用矩阵的形式来表示这种关系。如图 7 所示，部件变量矩阵是一个二维矩阵，前两行与前两列呈对称关系。矩阵的第一行（列）表示部件，第二行（列）则表示隶属于该部件的变量，部件与变量是一对一或一对多的关系。

在矩阵中，将行变量定义为需求变量，列变量定义为条件变量，需求变量的值会根据条件变量的改变而不同。如在一条需求中：当满足以下条件时，变量 A 的值应设置为 x：变量 $B=y$& 变量 $C=z$（A,B,C,x,y,z 依据实际需求而定）。这条需求中的变量 A 即为需求变量，变量 B 和变量 C 为条件变量。矩阵中行与列交叉的数值代表某一条需求中变量的关系。具体来说，以部件 i（行）的变量作为需求变量，部件 j（列）的变量作为条件变量，行列的交叉值 n 表示在这条需求中条件变量出现的次数，次数越多则表明需求变量对条件变量的依赖度越大。通过构建部件变量关系矩阵，初步展示部件之间的耦合度，同时揭示不同部件之间变量的依赖关系。

系统 system		部件1			部件2	部件3	
		变量A	变量B	变量C	变量D	变量E	变量F
部件1	变量A	0	1	1	2	1	0
	变量B	2	0	0	1	0	0
	变量C	0	0	0	0	0	1
部件2	变量D	0	3	0	0	1	0
部件3	变量E	1	0	0	1	0	0
	变量F	0	0	0	1	0	0

图 7 部件变量关系矩阵

Fig. 7 Component variable relationship matrix

5.2 构造部件变量关系矩阵

上一步构造的部件变量关系矩阵展示了变量的直接依赖关系，但间接依赖关系需要用需求变量依赖树的形式层次化地展示。在系统所给的需求中，每一条需求都有自己的 n 叉树，树的结构已在上文中展示。n 叉树的建立需要在 VRM 模型建模完成之后，首先需要分析需求以识别变量间的关系，根据需求确定哪些变量作为需求变量，哪些变量作为条件变量；然后初始化树结构，创建一个空的 n 叉树结构，将根节点设置为一阶需求变量(根需求)，该需求变量是整个树结构的起点；接着添加子节点，对于每个需求变量，将与之相关的条件变量添加到依赖树中，这些子节点可以是多个，表示 n 叉树的 n 个子节点；最后递归地构建子树，如果子节点也有自己的条件变量，递归地为这些子节点创建子树，继续添加子节点，直到满足所有需求为止。

5.3 构造部件变量耦合矩阵

为了更有效地量化这种耦合关系，在部件变量关系矩阵的基础上，对变量之间的依赖关系进行了权重化处理。具体而言，将每个变量之间的依赖度转换为相应的权重值，并将这些权重值填充到矩阵中，从而得到一个新的矩阵，即部件变量耦合矩阵。矩阵中的元素表示某个部件与某个变量之间的耦合程度，这个耦合程度是通过考虑变量之间的直接依赖关系和间接依赖关系计算得到的。具体来说，对于每个部件中的变量，需要找到其依赖的相关变量，然后将这些变量的权重填入该部件所对应的行和列中。在填充的过程中，如果两个变量之间的依赖度为 0，则该位置的权重为 0。例如，部件 A 中的变量 x 依赖于部件 B 中的变量 y 和变量 z，且变量 y 和变量 z 对于变量 x 的相对权重大小分别为 w_1 和 w_2，那么可以在矩阵中变量 x 所在的这一行与变量 y 和变量 z 对应的列填入相应的权重值 w_1 和 w_2。部件变量耦合矩阵算法如算法 1 所示。

算法 1 Component Coupling Algorithm

输入：VRM 模型

输出：部件耦合矩阵

```
1. Initialize CVmatrix
2.     set CVmatrix[0][j]=CVmatrix[i][0]=component
3.     set CVmatrix[1][j]=CVmatrix[i][1]=variable
4. Build requirement variable dependency tree
5.     Define function build_tree(root,children_vals)
6.     for val in children_vals do
7.         child=build_tree(val[0],val[1:])
8.         root.children.append(child)
9.     end for
10.    root=build_tree(root_val,children_vals)
11. Calculate variable weights
12.    MC=vrm.getValue(weight)
13.    mean=sum / len(values)
14.    Initialize squared_diff_sum as 0
15.    for value in values do
16.        squared_diff=(value - mean) * (value - mean)
17.        squared_diff_sum += squared_diff
18.    end for
19.    SC=math.sqrt(squared_diff_sum / len(values))
20.    HC=(depth-O(t))/depth * (sum(O(t)node)/ sum(node)/
       pow(2,depth-1))
21.    Define Weight(n→m) as MC * SC * HC
22. Calculate data coupling and control coupling
23.    dataCoupling=sum(CVmatrix(input) + CVmatrix(output)+
       CVmatrix(intermediate))
24.    controlCoupling=sum(CVmatrix(modeClass))
25. set CVmatrix as CCmatrix
26.    for variable in CCmatrix do
27.        dataCoupling→CCmatrix
28.        controlCoupling→CCmatrix
29.    end for
30. return CCmatrix
```

5.4 变量权重计算

变量权重的计算共分 3 步：首先，人工定义出变量的初始权重系数；然后，根据变量在每条需求中出现的次数，计算出变量标准差权重系数；最后，根据需求变量依赖树模型将变量划分为不同层级，并为其设置不同的权重系数。

综合 3 个权重系数，可以得到需求变量依赖树中变量之间依赖度的大小，即耦合度。具体计算式如下：

$$Weight(i)=MC(i)\times SC(i)\times HC(i) \tag{5}$$

其中，$Weight(i)$为 i 节点对于根节点的权重大小，$MC(i)$为节点 i 的人工定义系数，$SC(i)$为节点 i 的变量标准差系数，$HC(i)$为节点 i 的层级权重系数。

5.5 计算耦合度

耦合计算分为数据耦合和控制耦合，其中数据耦合度通过输入变量、输出变量和中间变量实现信息交换的程度，控制耦合度则是指两个部件之间通过模式类实现控制逻辑和行为模式交互的程度。

5.5.1 数据耦合度

数据耦合度可以通过式(6)计算：

$$DC(x,y)=\sum_{\forall i\in x\cap(iv,ov,tv\in y)} W(iv)+W(ov)+W(tv) \tag{6}$$

其中，i 表示属于 x 的状态变量（包含输入变量、输出变量、中间变量和模式类）；$W(iv)$，$W(ov)$ 和 $W(tv)$ 分别表示 y 的输入变量 iv、输出变量 ov 和中间变量 tv 对于变量 i 的权重值。

1）部件内部的数据耦合：对于系统中的一个任意部件 A，其内部数据耦合度可通过将式(6)中的 x 和 y 均设定为部件 A 来推导，部件 A 内部的数据耦合度，属于部件 A 的所有变量，对其自身输入变量、输出变量和中间变量的依赖权重之和。

2）部件之间的数据耦合：对于任意两个部件 A 和 B，其数据耦合度可通过将式(6)中的 x 设定为部件 A、y 设定为部件 B 进行计算。部件 A 对部件 B 的数据耦合度，即为属于部件 A 的所有变量，依赖部件 B 的输入变量、输出变量和中间变量的权重之和。

3）部件与系统的数据耦合：对于任意的部件 A，其与系统之间的数据耦合度可通过将式(6)中的 x 设定为部件 A、y 设定为系统(system)进行计算。部件 A 对系统 system 的数据耦合度，即为属于部件 A 的所有变量，依赖系统中所有部件的输入变量、输出变量和中间变量的权重之和。

5.5.2　控制耦合度

控制耦合度可以通过式(7)计算：

$$CC(x,y)=\sum_{\forall i\in x\cap(mc\in y)}W(mc) \tag{7}$$

其中，i 表示属于 x 的状态变量，$W(mc)$ 表示 y 的模式类 mc 对于变量 i 的权重值。

1）部件内部的控制耦合：对于系统中的一个任意部件 A，其内部控制耦合可通过将式(7)中的 x 和 y 均设为部件 A 来计算。部件 A 内部的控制耦合度，即为属于部件 A 的所有变量，依赖部件 A 自己的模式类的权重之和。

2）部件之间的控制耦合：对于任意两个部件 A 和 B，其控制耦合度可通过将式(7)中的 x 设定为部件 A、y 设定为部件计算得出。部件 A 对部件 B 的控制耦合度，即为属于部件 A 的所有变量，依赖部件 B 的模式类的权重之和。

3）部件与系统的控制耦合：对于任意的部件 A，其与系统之间的控制耦合度可通过将式(7)中的 x 设定为部件 A、y 设定为系统(system)进行计算。部件 A 对系统 system 的控制耦合度，即为属于部件 A 的所有变量，依赖系统中所有部件的模式类的权重之和。

5.6　识别关键变量和关键部件

找出系统的关键变量和关键部件，是识别系统中潜在风险点的关键步骤。为了实现这一目标，使用排序筛选算法来评估变量和部件的关键程度。通过调整筛选的百分比阈值，可以控制关键变量和关键部件的数量，以适应不同的系统需求和分析目的。以下将阈值设为 30%。

5.6.1　关键变量

关键变量的识别可以通过分析部件变量耦合矩阵来实现。该方法首先对矩阵中每个变量的权重值进行降序排序，然后筛选出权重值最大的前 30%作为关键变量，较高的权重值表示该变量在部件和系统中具有较大的影响程度。以下是具体的算法步骤：

1）输入：部件变量耦合矩阵（$\boldsymbol{M}$），百分比阈值(30%)。

2）根据矩阵 $\boldsymbol{M}$，计算每个变量的权重值。

3）对变量权重值进行降序排序。

4）将权重值最大的前 30%的变量作为关键变量。

5.6.2　关键部件

关键部件的识别也通过分析部件变量耦合矩阵来实现。首先计算出矩阵中每个部件的变量权重值之和，将其作为部件在系统中的权重值；然后对所有部件的权重值进行降序排序，筛选出前 30%作为关键部件。以下是具体的算法步骤：

1）输入：部件变量耦合矩阵（$\boldsymbol{M}$），百分比阈值(30%)。

2）计算部件 x 的变量权重值之和，$x\in\boldsymbol{M}$。

3）重复步骤 2），得到矩阵 $\boldsymbol{M}$ 中每个部件的权重值。

4）对部件权重值进行降序排序。

5）将权重值最大的前 30%的变量作为关键部件。

5.7　算法梳理及复杂性分析

整体流程包含以下步骤：

1）VRM 需求建模；

2）构造部件变量关系矩阵；

3）建立需求变量依赖树；

4）变量权重计算；

5）构造部件变量耦合矩阵；

6）数据耦合和控制耦合度量。

根据上述步骤，算法的复杂度分析如下。

1）需求变量依赖树的构造：在构建变量依赖树时，算法需遍历所有变量并为其分配依赖关系，时间复杂度为 $O(n\times m)$，其中 n 是变量总数，m 是变量之间的依赖关系数量。

2）变量权重计算：3 种权重（MC，SC，HC）的计算分别为常数时间操作（MC）、标准差计算（SC，复杂度为 $O(n)$），以及递归层级遍历（HC，复杂度为 $O(d)$，其中 d 为树的深度）。

3）耦合矩阵的构建与运算：变量之间耦合度的计算需要遍历矩阵元素，复杂度为 $O(k^2)$，其中 k 是变量数量。

6　实例研究

图 8 给出了一个基于 VRM 模型对机载软件系统数据耦合分析的实例——飞行制导系统（FGS）的模型部件。对飞行制导系统进行 VRM 模型建模后，得到了与数据耦合相关的输入变量、输出变量和中间变量共计 218 个，与控制耦合相关的模式类共计 22 个。由于飞行制导系统过于庞大，本文仅以系统中的一条需求为例，展示基于 VRM 模型的数据耦合和控制耦合分析方法。

如表 3 所列的几条需求中，需求 A 作为 1 阶需求，它的定义为：模式通知的状态由关闭切换为开启的条件为：当前为飞行员飞行的一侧且开启飞行模式并显示在 PFD 上；模式通知的状态由开启切换为关闭的条件为：当前为飞行员飞行的一侧且关闭飞行模式并在 PFD 上不显示。而对于需求 A 来说，当前是否为飞行侧这个条件依赖于需求 B，开启飞行模式并显示在 PFD 上依赖于需求 C，关闭飞行模式并在 PFD 上不显示依赖于需求 D。也就是说，1 阶需求 A 的条件依赖于2 阶需求 B、需求 C 和需求 D。同理可以从表里看出，需

求 B 的条件又依赖于 3 阶需求 E、需求 F 和需求 G。以此类推，可以得到以 mcModes 为一阶需求的需求与变量的依赖关系。

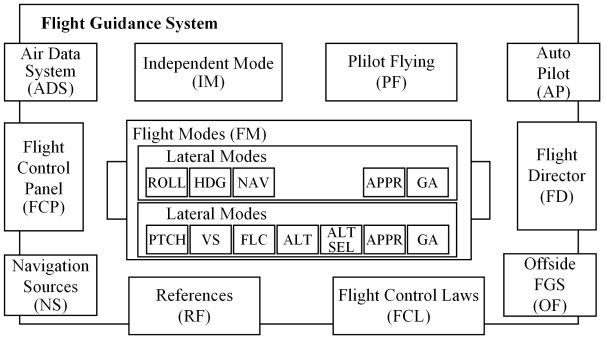


图 8 飞行制导系统部件

Fig. 8 Flight guidance system components

为了更清晰地理解变量的依赖关系，对以上需求进行 VRM 模型建模后，整理出了如表 4 所列的 19 个变量与它们属于的部件，并根据变量在需求中出现的次数，计算出变量权重的标准差系数 *SC*。同时为了方便演示，将所有变量的人工定义系数 *MC* 设置为 1。之后需要建立需求变量依赖树。变量间的依赖关系如图 9 所示。1 阶需求的变量 mcModes 依赖于 2 阶需求的变量 mcIsThisSideActive，tTurnModesOn 和 tTurnModesOff，2 阶需求的变量 tTurnModesOn 又依赖于 3 阶需求的变量 mcIndependentMode，mcPilotFlying 和 tOffsideModes。依次往下，构成了以 mcModes 为根需求的需求变量依赖树，从中可以很直观地看出根需求变量与其他变量的依赖关系。

mcModes 的需求变量依赖树中共有 25 个变量节点，如表 5 所列。通过计算每个变量与根节点变量的距离度 dpt 和树中每一层节点的复杂度 cmp，可以得到每个阶数的层级权重系数 HC。

最后，构造部件变量耦合矩阵，得到每个变量对其他变量的依赖度，并将属于不同部件之间的变量依赖度累加起来，计算出变量与变量、变量与部件和部件与部件之间的耦合度值。如表 6 所列，部件 Flight Modes 的变量 mcModes 对其他部件的数据耦合计算方法为：将 mcModes 依赖的输入变量、输出变量和中间变量的耦合度值累加；部件 Flight Modes 的变量 mcModes 对其他部件的控制耦合计算方法为：将 mcModes 依赖的模式类的耦合度值累加。

当计算出 Flight Modes 组件中所有变量与变量耦合矩阵中所有变量的耦合度后，可进一步推导得出 Flight Modes 组件内部、该组件与其他任意组件之间，以及 Flight Modes 组件与 FGS 系统之间的耦合度指标。

表 3 tTurnFDOff 需求依赖表

Table 3 tTurnFDOff demand dependency table

编号	阶数	需求内容
A	1	mcModes Off→On : if AND mcIsThisSideActive = T AND tTurnModesOn = T On→Off : if AND mcIsThisSideActive = T AND tTurnModesOff = T
B	2	mcIsThisSideActive = TRUE : if OR AND PreviousStep(mcIndependentMode)=Off F T T AND mcPilotFlying = THIS SIDE$_{LEFT}$ T F AND mcOffsideModes = On F mcIsThisSideActive = FALSE : if AND Previous Step(mcIndependentMode) = Off T AND mcPilotFlying = THIS SIDE$_{LEFT}$ F AND mcOffsideModes = On T
C	2	tTurnModesOn = TRUE : if OR AND mcOnsideFD = ON T AND mcOffsideFD = ON T AND tIsAPEngaged T
D	2	tTurnModesOn = TRUE : if AND mcOnsideFD = Off T AND mcOffsideFD = Off T AND mcAP = Disengaged T
E	3	mcIndependentMode := On : if tIndependentModeCondition T := Off : if tIndependentModeCondition F
F	3	mcPilotFlying LEFT→RIGHT : if tTransferSwitchPressedSeen T
		RIGHT→LEFT : if tTransferSwitchPressedSeen T
G	3	tOffsideModes = TRUE : if ipOffsideModesOn T
H	3	mcOnsideFD Off→On : if tTurnFDOn T On→Off : if tTurnFDOff F
I	3	mcOffsideFD := Off : if ipOffsideFDOn F := On : if ipOffsideFDOn T
J	3	mcIsAPEngaged := TRUE : if AP = Engaged T
K	3	mcOnsideFD Off→On : if tTurnFDOn T On→Off : if tTurnFDOff F
L	3	mcOffsideFD := Off : if ipOffsideFDOn F := On : if ipOffsideFDOn T
M	3	mcAP Disengaged→Engaged : if tEngageAP T Engaged→Disengaged : if tDisengageAP T

表 4　变量标准差系数

Table 4　Variable standard deviation coefficients

阶数	变量	部件	数量	*SC*
1	mcModes	FM	15	2.395
2	mcIsThisSideActive	IM	14	2.313
2	tTurnModesOn	FM	1	0.618
2	tTurnModesOff	FM	1	0.618
3	mcIndependentMode	IM	2	0.874
3	mcPilotFlying	PF	1	0.618
3	tOffsideModes	OF	4	1.236
3	mcOnsideFD	FD	2	0.874
3	mcOffsideFD	OF	4	1.236
3	mcIsAPEngaged	AP	4	1.236
3	mcAP	AP	4	1.236
4	tIndependentModeCondition	IM	1	0.618
4	tTransferSwitchPressedSeen	FCP	1	0.618
4	ipOffsideModesOn	OF	2	0.874
4	tTurnFDOn	FD	1	0.618
4	tTurnFDOff	FD	1	0.618
4	ipOffsideFDOn	OF	2	0.874
4	tEngageAP	AP	1	0.618
4	tDisengageAP	AP	1	0.618

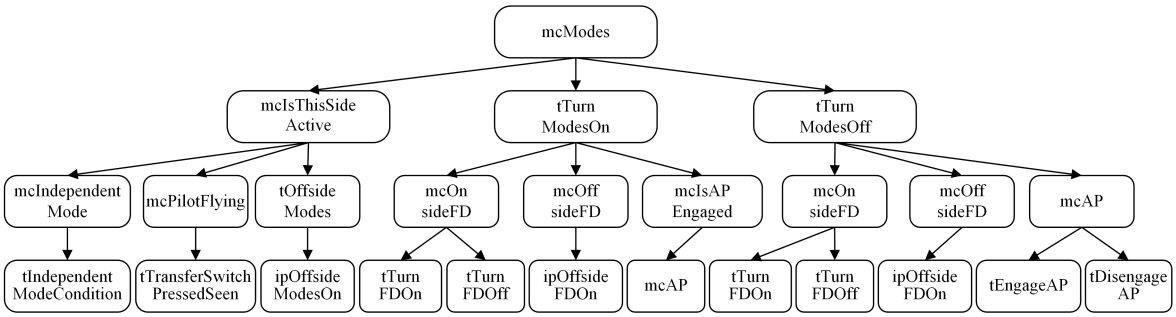


图 9　mcModes 需求变量依赖树

Fig. 9　mcModes demand variable dependency tree

表 5　变量层级权重系数

Table 5　Weight coefficients for variable hierarchies

阶数	距离度 *dpt*	复杂度 *cmp*	层级权重系数 *HC*
2	1.00	0.12	1.120
3	0.75	0.36	0.555
4	0.5	0.48	0.245

表 6　tTurnFDOff 变量耦合度

Table 6　tTurnFDOff variable coupling

部件	变量	依赖度	数据耦合	控制耦合
IM	mcIsThisSideActive	2.59	0.151	2.985
	mcIndependentMode	0.485		
	tIndependentModeCondition	0.151		
FM	tTurnModesOn	0.692	1.384	0
	tTurnModesOff	0.692		
PF	mcPilotFlying	0.342	0	0.342
FD	mcOnsideFD	0.485	0.302	0.485
	tTurnFDOn	0.151		
	tTurnFDOff	0.151		
OF	mcOffsideFD	0.685	1.113	0.685
	ipOffsideModesOn	0.214		
	ipOffsideFDOn	0.214		
	tOffsideModes	0.685		
AP	mcIsAPEngaged	0.685	0.302	1.672
	mcAP	0.987		
	tEngageAP	0.151		
	tDisengageAP	0.151		
FCP	tTransferSwitchPressedSeen	0.151	0.151	0

结束语　本文介绍了一种基于 VRM 模型的面向机载软件需求的耦合分析方法，该方法通过定义变量之间的依赖关系和度量指标来评估系统部件之间的数据耦合和控制耦合。为了弥补基于代码层级的耦合分析技术的不足，采用 VRM 模型作为形式化需求模型，对基于 DO-178C 的数据耦合、控制耦合和部件进行了定义。

其次，本文提出了基于 n 叉依赖树的变量层次依赖关系，并对不同变量划分权重，形成了基于变量依赖关系的数据耦合和控制耦合分析原型。该方法的局限性有两点：1）未充分考虑变量的时间序列依赖，难以处理动态需求的变化；2）在处理极大规模的复杂系统时，变量依赖树可能因节点过多使计算复杂度和存储成本上升。该方法的先进性在于：1）首次在 VRM 中引入变量依赖树和权重划分机制，支持多达 4 层的依赖分析；2）在需求阶段应用该方法，能有效减少研发后期潜在问题未及时暴露的问题溯源和修复成本；3）该方法不仅适用于航空飞行制导领域，也适用于其他非航空的控制类。

在研究中，以飞行制导系统的需求案例为对象，对本文提出的耦合分析方法的可行性进行了验证。然而，该方法仍存在一定局限性，例如在分层建模、变量时序条件定义及耦合算法优化等方面有待完善。未来，可通过开展更深层次的实验持续优化该方法，研发一套符合 DO-178C 标准的耦合分析工具，并将其实际应用于工程需求场景。该方法未来的优化

方向有3个方面。

1)自动化权重定义:使用机器学习技术(如深度学习或决策树)从历史项目数据中自动学习变量权重,减少人工干预。

2)时序模型扩展:引入时序逻辑或动态变量关联分析方法,将时间维度纳入变量依赖关系建模。

3)分布式计算支持:在处理超大规模系统时,利用分布式计算框架(如Hadoop或Spark)优化变量依赖树构造及权重计算流程。

参 考 文 献

[1] RTCA. Software considerations in airborne systems and equipment certification: DO-178C [S]. Washington, DC: RTCA, Inc., 2011.

[2] CAST. Clarification of Structural Coverage Analyses of Data Coupling and Control Coupling: CAST-19-2004 [S]. USA: CAST, 2004.

[3] HELLHAKE D, BOGNER J, SCHMID T, et al. Towards using coupling measures to guide black-box integration testing in component-based systems[J]. Software Testing, Verification and Reliability, 2022, 32(4): e1811.

[4] MEYERS S C. Automatic generation of data coupling and control coupling test conditions: US10120785B2[P]. 2018.

[5] RIERSO N, LEANN A. Developing safety-critical software: a practical guide for aviation software and DO-178c Compliance [M]. Taylor & Francis, 2013.

[6] MAIA T, SOUZA M. A Practical Methodology for DO-178C Data and Control Coupling Objective Compliance[C]// Proceedingsof the International Conference on Software Engineering Research and Practice (SERP). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp). 2018: 236-240.

[7] ESTRADA R G, SASAKI G, DILLABER E. Best practices for developing DO-178 compliant software using Model-Based Design[C]// AIAA Infotech@ Aerospace (I@ A) Conference. 2013: 4566.

[8] CHEN X H, LIU S B, JIN Z. Survey on Requirements Description of Embedded System[J]. Ruan Jian Xue Bao/Journal of Software, 2025, 36(1): 27-46.

[9] ABDURAZIK A, OFFUTT J. Using coupling-based weights for the class integration and test order problem[J]. The Computer Journal, 2009, 52(5): 557-570.

[10] HELLHAKE D, BOGNER J, SCHMID T, et al. Towards using coupling measures to guide black-box integration testing in component-based systems[J]. Software Testing, Verification and Reliability, 2022, 32(4): e1811.

[11] KAGDI H, GETHERS M, POSHYVANYK D, et al. Blending conceptual and evolutionary couplings to support change impact analysis in source code[C]// 2010 17th Working Conference on Reverse Engineering. IEEE, 2010: 119-128.

[12] PRIYALAKSHMI G, LATHA R. Evaluation of software reusability based on coupling and cohesion[J]. International Journal of Software Engineering and Knowledge Engineering, 2018, 28(10): 1455-1485.

[13] POSHYVANYK D, MARCUS A, FERENC R, et al. Using information retrieval based coupling measures for impact analysis [J]. Empirical Software Engineering, 2009, 14: 5-32.

[14] CZIBULA I G, CZIBULA G, MIHOLCA D L, et al. An aggregated coupling measure for the analysis of object-oriented software systems[J]. Journal of Systems and Software, 2019, 148: 1-20.

[15] ZHANG L Y, LI X M, SHI Z P, et al. Formal Verification of Functional Correctness for Mutexes in Microkernel[J]. Ruan Jian Xue Bao/Journal of Software, 2024, 35(9): 4179-4192.

[16] BAVOTA G, DIT B, OLIVETO R, et al. An empirical study on the developers' perception of software coupling[C]// 2013 35th International Conference on Software Engineering (ICSE). IEEE, 2013: 692-701.

[17] SHERRIFF M, WILLIAMS L. Empirical software change impact analysis using singular value decomposition[C]// 2008 1st International Conference on Software Testing, Verification, and Validation. IEEE, 2008: 268-277.

[18] HU J, SHI J J, ZHEN C, et al. System Safety Modeling and Analysis Method Based on Four-variable Model[J]. Computer Science, 2016, 43(11): 193-199.

[19] FREGNAN E, BAUM T, PALOMBA F, et al. A survey on software coupling relations and tools[J]. Information and Software Technology, 2019, 107: 159-178.

[20] HU J, LYU J R, WANG L S, et al. Case Study on Formal Modeling and Analysis of Airborne Software Requirements[J]. Ruan Jian Xue Bao/Journal of Software, 2022, 33(5): 1652-1673.



YIN Wei, born in 1979, Ph. D, professorate senior engineer. His main research interests include control science and engineering, avionics, software engineering and artificial intelligence.

(责任编辑:柯颖)