## 国内外研究现状

### 1.2.1 FADEC系统软件概述

从上世纪70年代FADEC开始出现，国外已经在军用发动机F100和民用发动机JT9D上对系统的性能、使用和维护指标进行了验证。经过30多年的发展，FADEC系统已经成为目前航空发动机的主要控制系统[8]，随着我国成功研制出某型号发动机的控制系统，说明我国在发动机控制系统的研制上具有了一定实力。

目前FADEC软件系统主要包括信号采集和处理模块，核心机控制模块，系统监控模块，信号输出模块，冗余管理模块和支撑软件模块等六部分。其模块图如图1所示。



图 1 FADEC软件系统功能模块图

可以看到，FADEC软件系统由多个子模块构成，涉及到数据的采集和处理，系统控制等等方面的功能，软件系统复杂。随着我国航空领域技术的不断发展，对航空发动机的要求也越来越高，随之而来的是航空发动机控制系统软件的需求越来越复杂。

### 1.2.2 软件需求追踪技术

可追溯性作为航空发动机适航符合性审查的目标之一，其重要性早已软件业界认可。软件能力成熟度集成模型(Capability Maturity Model Integration, CMMI)对需求管理提出的明确的指导性的方法，目前已逐渐被软件工程领域所采用，需求追踪作为软件需求管理的重要组成部分，其重要性也是不言而喻。

需求追踪技术主要分为两种类型：静态需求追踪技术和动态需求追踪技术。静态需求追踪技术通常是由手工构建和维护追踪链，常用的方法有：需求追踪矩阵、交叉引用和追踪图。动态追踪技术是为了以自动或半自动化技术帮助开发人员建立和维护可追踪链，这个过程中各种软件制品被看作静态文本文件，根据文本间的关系来构建可追踪链。

#### 1.2.1.1 静态需求追踪技术

静态需求追踪方法优点易于构建，是传统需求追踪中主要使用的方法，但由于只能静态构造，在大型项目或软件需求变更频繁的项目中，这种方法却难以维护和扩展。

需求追踪矩阵（Requirement Traceability Matrix，RTM）是展现和表示需求间追踪信息的最简单方法，它的主要的表现方式是通过矩阵或者表格来表示两个或者多个软件制品间的追踪关系，需要注意的是，软件需求间存在的关系不仅仅是“一对一”的关系，还有“一对多”，“多对多”的关系。一个软件需求追踪矩阵实例如图2所示。



图 2 需求追踪矩阵示意图

追踪图与追踪矩阵类似，它可以用图形的表现形式把用户自定义的对象和关系，将软件需求、测试用例、及源代码等等软件制品表述出来，这种方法直观，灵活，易于理解。

交叉引用则是通过对相互关联的软件制品（实体），将其中一个实体的关联属性中给出另一个相关实体的引用的方式表示。可以在需求文档之间建立追踪关系，需求文档包括需求规格文档、需求说明文档等，对于发动机控制软件，还有系统需求、高层需求和低层需求等不同文档。这种方法表示追踪关系比较直观，方便实际使用。只适用于需求文档间关系的建立与处理是它的一个主要缺点[9]。

#### 1.2.2.2 动态需求追踪技术

为了解决在人工构建并维护软件需求的可追踪关系时费时费力，容易出错，难以维护的问题，有些学者开始尝试自动化或者半自动话的构建和维护需求间的追踪关系。但直到第13届需求工程(Requirements engineering, RE)大会上，“动态需求追踪”这个概念才首次由J. Cleland-Huang提出。

在构建可追踪链的过程中，学者们从不同角度提出了方法来自动构建需求间的追踪关系，主要有基于事件的追踪技术（Event-Based Traceability，EBT），基于场景的追踪技术（Scenario-Based Traceability，SBT），基于程序运行时（Runtime-Based Tracebility，RBT）及基于信息检索技术(Information-Based，IR)[7]等。由于软件在设计、开发和维护过程中产生的生命周期数据大部分为文本信息，而需求之间的关系可以认为是文本间的关系。在这种情况下，基于信息检索技术的动态的需求追踪算法中有广泛应用，基于信息检索的需求追踪算法经典模型[11]如图3所示。

软件工程师

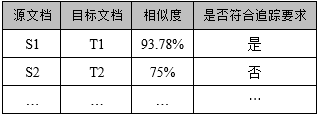
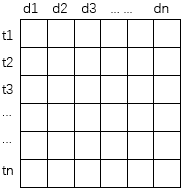
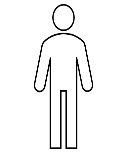
文本向量

结果分析

选择文件

软件制品

文本处理



相似度计算

图 3 基于信息检索的需求追踪算法经典模型

这种方法的主要思想是通过比较文档间的相似度确定需求间可追踪性关系的，从图3可以看到，基于信息检索算法的动态需求追踪算法流程为：首先，需要对软件制品数据作为文档进行预处理，并建立文本的向量表示；然后通过文本相似度算法来计算由软件工程师所选定的查询文件和候选文件间的相似度，最后，通过将文本相似度与事先设定好的阙值去比较，如果相似度足够大，则说明两个文件间具有可追踪关系。显然如何计算文本相似度是需求追踪算法的一个关键问题。

Dag等人[12]首先提出通过使用文本分析的方法来判断需求间的追踪关系，它们通过自定义一些衡量标准，主要通过比较文本间单词的关系计算文本相似度，最后通过调整阙值，来找到最优结果，并取得了一定的结果，并指出了这种方法很有发展前景。同年，G. Antoniol等人[13]使用基于概率模型(Probability Model, PM)和向量空间模型的需求追踪方法为源代码和文档间的追溯性关系，并在自己的数据集上取得了很好的效果。目前的很多基于信息检索的需求追踪算法都是以他们所做的工作为基础的。

Huffman 等人[15]在向量空间模型算法的基础上，有两点改进，一个是手工构建关键词库，另一个是使用简单词典，将一些有关联的词对应起来，起到了扩展查询的作用，最后她们将这种算法应用于NASA的项目，通过与手工和单纯的向量空间算法进行了对比，有较好的效果提升。在这篇文章中她们还整理了NASA项目的需求子集，将高层需求、低层需求和追踪关系整理出来提供给其他研究者继续进行研究。

随后，很多学者通过不同的方法对文本进行表示，从而计算相似度来更快和获取更高质量的需求追踪链的方法，隐语义模型(LSI)和主题模型(LDA)[15,16]被应用于需求追踪，建立需求、软件制品和测试用例之间的跟踪关系，他们通过将具有相同意义的词语映射到同一空间以降低文本向量的维数，以此来降低运算复杂度并获得较高质量的可追踪链。直到现在这些方法已经成为需求追踪领域比较成熟技术[17]。

一些学者在此基础上，进行了改进，将蚁群算法，加入过滤器，本体[18-20]等方法与传统的文本相似度算法相结合，用于提升结果精度。到现在通过改进这种基于统计的信息检索方法所获得的提升已经很小。已经有学者尝试将机器学习和深度学习技术应用于这一领域。

David等人[21]通过用机器学习分类器的方法应用于需求追踪算法，并通过将不同的信息检索算法与多个分类器进行组合，给出了在不同数据集上的最佳实践。它指出可以使用分类器来对追踪链接进行分类，从而不需要在不同的数据集上设置阙值。这说明可以将机器学习方法应用于需求追踪的动态构建中。

而Jin G等人[17]则是将目前流行的word embedddings计算和深度学习技术应用于需求追踪领域。他们将文本用词向量特征表示，并通过循环神经网络（Recurrent Neural Networks，RNN）训练并预测文本间的相似度，他们的实验效果很好，但是它需要一个高质量的领域本体和大量被人工验证过的可追踪链接来进行训练，因此这种方法并不适用与较小的数据集，不具有普遍性。

除此之外，研究者还提供了一些基于信息检索技术的工具，如处理非结构化文本的NETRO，可拓展的实验工具TraceLab，候选链接聚类工具TraCter和支持投票的FaceTrace[22-25]等工具，来帮助学者进行研究。