由于变更管理中，必然涉及到了基线管理、标识配置、所以建立变更管理的模型。基线的建立是基于变更的，而进行标识配置是为了基线管理和变更管理服务。

包含了配置标识活动以及基线管理活动的，以变更管理为中心的配置管理模型。能够反映配置管理过程的关键技术环节

变更控制是软件配置管理的核心，它通过创建产品基线，在产品的整个生存周期中控制它的发布和变更，实在整个软件生周期中控制度软件的变化

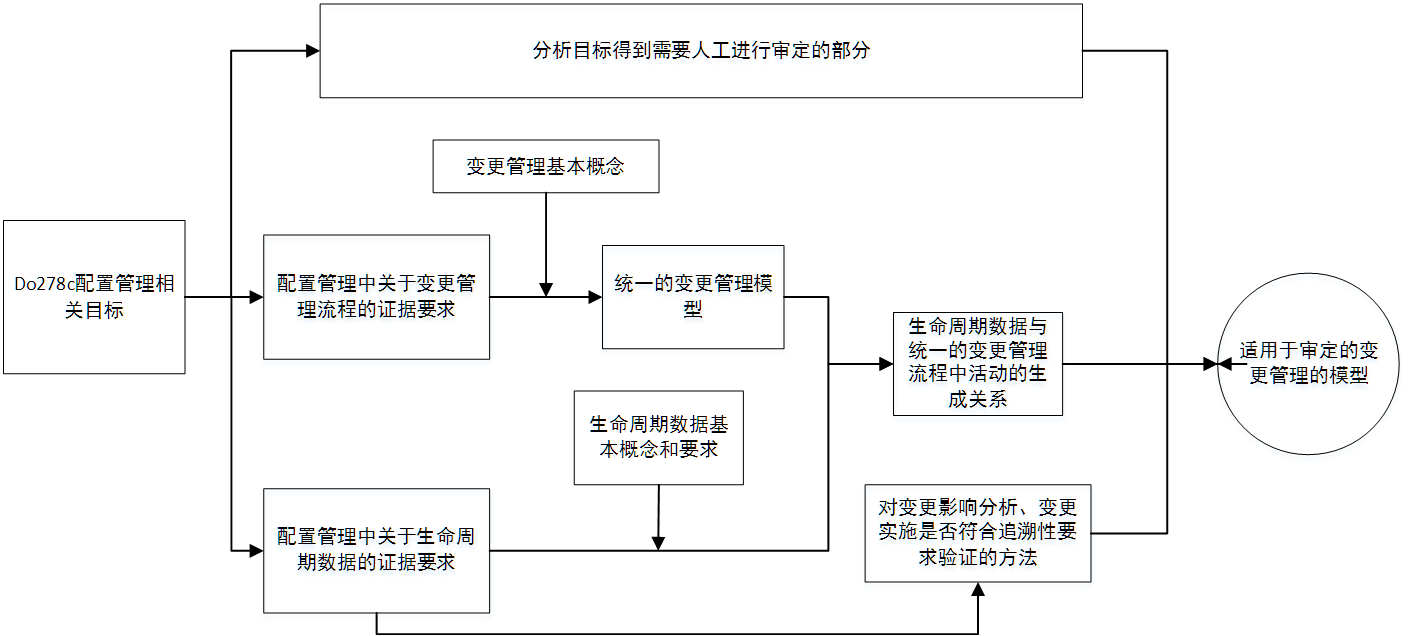
研发过程中为审定过程提供需要的生命周期数据，而我国在机载软件研发审定方面的实际情况是，审定基本靠专业人员经验，研发为了进度，人员对标准理解不多，研发主要还是按照传统软件的研制流程，最后生成的生命周期数据的形式与可审定的程度参差不齐，未来必然越来越规范化，但是一个逐步的过程，另外，由于研发方要考虑的到具体的设计等必然要精细的多，而审定工作不可能将每次审定一个软件都编写一次，故必然应该只专注于标准提到的统一性的要求，必须抽取出标准中共同的目标。

# 第三章 基于DO-178C标准研究，建立 适航领域配置管理过程审定模型

本章首先对使用GSN图符对D0178c中配置管理过程部分目标进行了分析，接下来提出了变更管理的审查中需要的生命周期数据及格式要求，然后提出了统一的变更流程，并确定了生命周期数据和变更流程的输入输出关系。并根据上述研究建立适航领域配置管理过程审定模型

### 3.1 建立适航领域配置管理过程审定模型的流程

本研究提出了通过对DO178c目标进行分析，结合变更管理的基本概念，以及DO178c中有关生命周期数据的概念和规定，建立了统一的变更管理流程，并 确定了生命周期数据和变更流程的输入输出关系 ，最终得到适航领域配置管理过程审定模型，对配置管理过程自动审定模型建模流程如下图所示。



**首先使用GSN方法分析Do178c配置管理流程的相关目标，将能构最终满足DO178C目标的**

**使用GSN方法，**研究Do178c中配置管理过程的目标，提出了一种适合于审查的变更管理的模型，并确定了变更管理过程中关键生命周期数据及变更记录的需求，并根据以上研究，使用安全UML，建立符合适航标准的主要用于适航审查的UML模型。

第一步，使用Gsn方法，对Do178c中的目标进行分析，在分析过程中，结合

### 3.2 使用GSN方法对目标进行分析

GSN是一种可以明确说明任一论证过程中存在的元素机器相互关系的图形化论证符号语言。运用GSN语言来表述论证过程可以有助于形象、具体、合理地说明系统、运营肌组织的安全性能。运用GSN语言完成的论证模型主要是用来表述安全子目标是如何支持顶层的安全目标的，下一层的安全子目标又是如何支持该层安全子目标的。

这与在Do178c标准中，目标、过程、与数据间的关系是相似的，通过适当的变换，所以可以利用GSN，对do178c标准进行具体分析，找出相关目标的过程要求、子过程及存在的证据间的对应关系。

将Do178c标准所要求的目标对应于GSN中的安全目标，生命周期数据或数据元对应GSN中的安全证据，而标准中对过程要求目标可以作为论证策略的一部分。其中将Do178c提供的过程指导作为策略或是论点的一部分在分析过程中得到证明，能够降低对整个论证过程进行分析的复杂性

DO-178C标准附录A中，DO-178C对配置管理过程中要满足目标，在DO-178C附录A的表7中有详细的说明，包括6个目标，分别是：

1. 标识构型项（A-8.1）
2. 建立基线和可追溯性（A-8.2）
3. 建立问题报告，变更控制，变更评审和构型状态纪实机制（A-8.3）
4. 建立归档、检索和发布机制（A-8.4）
5. 建立软件加载控制机制（A-8.5）
6. 建立软件生命周期环境控制（A-8.6）

本文选取a-8.1 ,a-8.2 ,A-8.3 三个目标作为研究对象，依据标准 进行了详细的解读，确定了各个目标的论证结构（包含证据）。进行详细解读的依据包括 标准中对过程目标的解读，审查指南，具有审查经验的人员的指导等。

在对目标使用GsN方法进行分析时，使用了两种大的策略：

策略1.第一次将大目标分解时，主要以“标准、流程、经验、审查指南”为依据，将Do178c要求的配置管理过程目标按子过程相关要求分解为支持目标论证的多个子过程目标。

策略2：为了构建符合Do178c标准的变更管理模型，在对子过程的目标进行自上而下的论证时，以“标准、流程、经验、审查指南”为依据，分析得出为满足子过程目标需求的证据需求，并分类：

1在数据初始化和论证过程中需要人工辅助的证据要求。

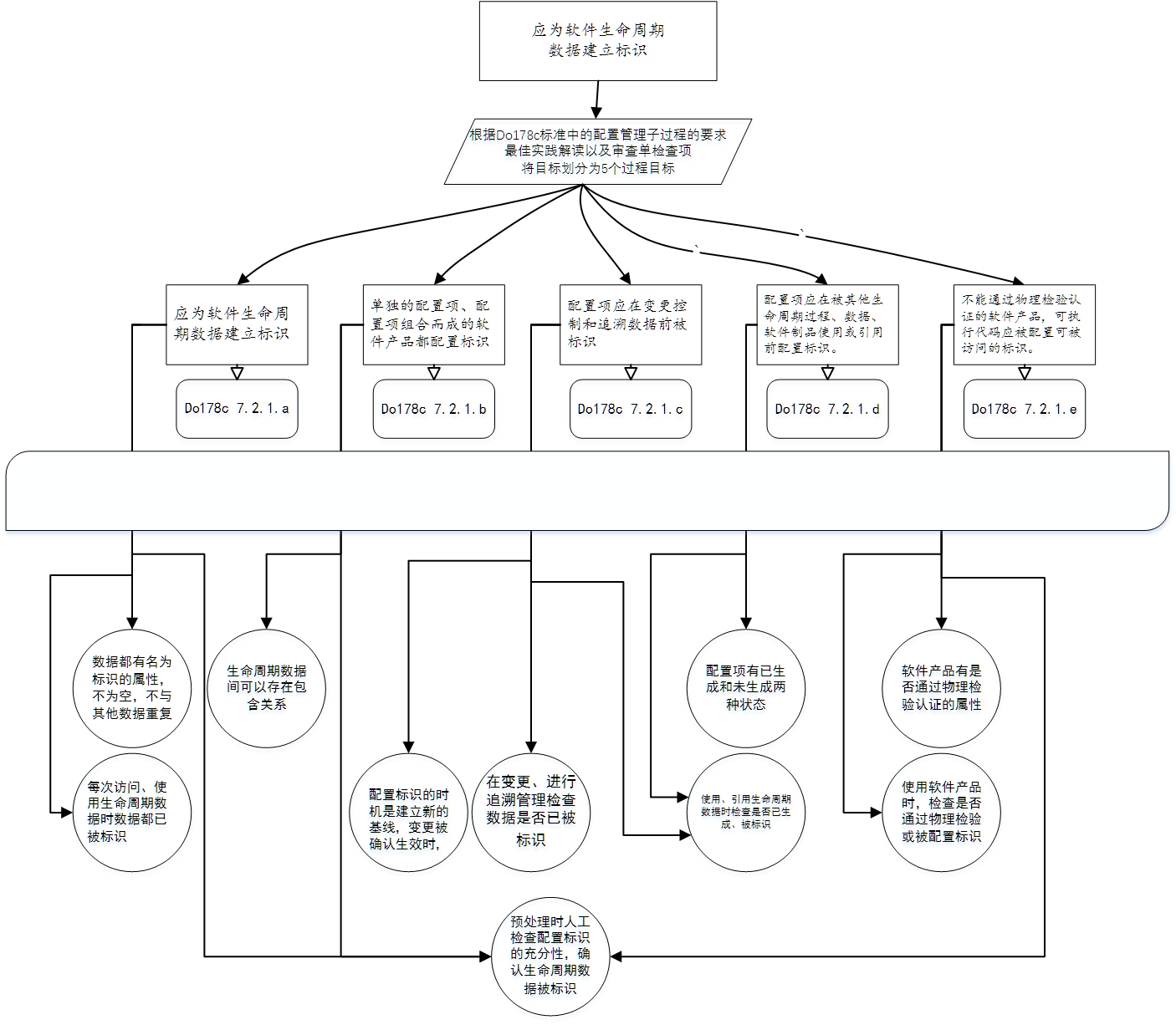
2.主要对生命周期数据自身相关的证据要求。

3与变更管理的流程要求相关的证据要求。

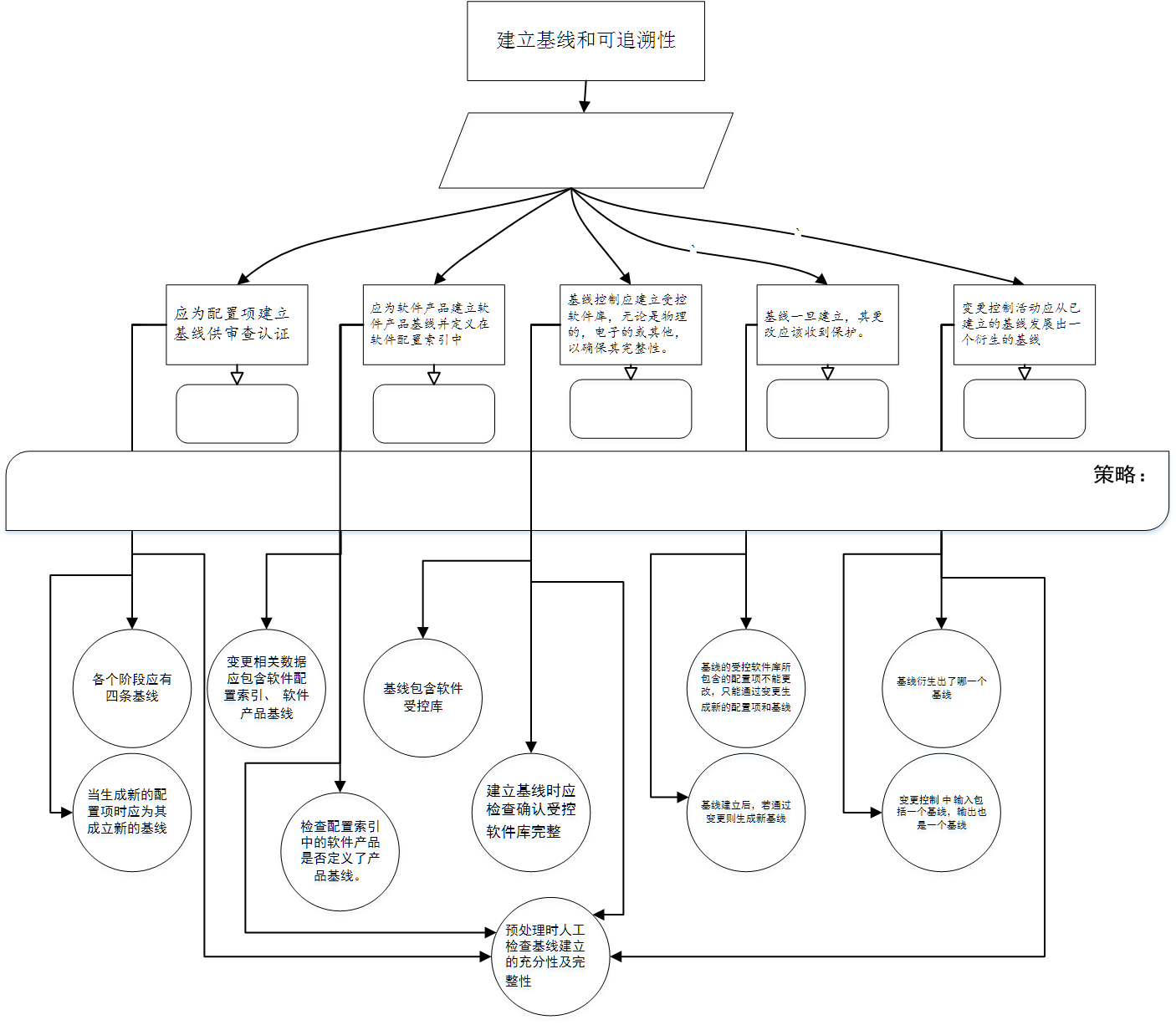
3.需要由生命周期数据的具体内容或由其他生命周期过程参与进行验证的目标。

#### 3.2.1 对目标 A-8.1使用GSN方法进行分析，

具体如图（）



#### 3.2.2 对目标 A-8.2使用GSN方法进行分析，



#### 3.2.3 对目标 A-8.3使用GSN方法进行分析，

目标分析的所有图。都弄在这里。

需要说明的部分晚一点再弄。

对于配置状态纪实，主要通过人工检查，结合本模型各个检查的结果判断。此项工作是使用本模型进行检查的基础，在数据预处理前完成。

#### 3.2.4 对分析后的目标进行总结归纳

对3.1节中对目标分析进行分析的结果，进行统一的分析和归纳，仍然按照分类，得出以下的证据性要求。

1. 人工进行审查的部分
2. 根据流程进行审查的部分
3. 根据配置项数据审查的部分

|  |  |
| --- | --- |
| 人工进行审查的部分 |  |
| 流程审查 |  |
| 配置项具体数据 |  |

### 3.3 生命周期数据及格式要求

生命周期数据、最佳实践提出数据项与数据元

#### 3.3.1标准要求的生命周期数据

在Do178C标准中，列举了22种生命周期数据（**错误!未找到引用源。**），按照标准要求，所有标准要求的软件生命周期数据都应该被标识。也就是说开发过程中至少应该将这22种生命周期作为配置管理的标识项。同时Do178C标准也提出了其它需要作为配置项进行管理的要求，因此22种配置项不是配置项的全集。

在实际的适航软件开发中，目前大部分的生命周期数据材料都是以文本的方式提供的，这使得直接使用该生命周期数据材料进行自动审定的工作是难以开展的，即使是由专家进行审定，也必然由于数据庞杂且不规范，提高了审定的难度，也降低了审定效率和效果。

|  |  |
| --- | --- |
| **数据项名称** | **数据项名称** |
| 软件方面的验证计划 | 可执行目标代码 |
| 软件开发计划 | 软件验证例子和程序 |
| 软件验证计划 | 软件验证结果 |
| 软件配置管理计划 | 软件生命周期环境配置索引 |
| 软件质量保证计划 | 软件配置索引 |
| 软件需求标准 | 问题报告 |
| 软件设计标准 | 软件配置管理记录 |
| 软件代码标准 | 软件质量保证（SQA）记录 |
| 软件需求数据 | 软件完成综述 |
| 设计说明 | 数据追踪 |
| 源代码 | 参数数据项文件 |

#### 3.3.2最佳实践提出的数据项与数据元

为了解决这个问题，根据最佳实践提出的数据项管理的要求，将管理的配置项分为了数据元与数据项，数据项实质是生命周期数据被标识为配置项后，对配置项进行分类的结果，是为了方便对生命周期数据进行管理而进行的细化。数据元则是更为喜欢的生命周期数据元素。最佳实践识别的数据项见表（）。

|  |  |
| --- | --- |
| 数据项编号、名称 | 数据项编号名称 |
| D-01 项目进度计划 | D-29软件需求数据（条目化） |
| D-02 分配到软件的系统需求 | D-30高层需求与系统需求的追踪数据 |
| D-03软件等级 | D-31软件架构（非条目化） |
| D-04项目估算数据 | D-32底层需求层次结构 |
| D-05项目软件生命周期 | D-33底层需求（条目化） |
| D-06项目管理计划 | D-34软件设计说明（非条目化） |
| D-07软件合格审定计划 | D-35底层需求与高层需求的追踪数据 |
| D-08软件开发计划 | D-36源代码 |
| D-09软件验证计划 | D-37源代码与底层需求的追踪数据 |
| D-10软件配置管理计划 | D-38测试用例 |
| D-11软件质量保证计划 | D-39测试用例与高层需求的追踪数据 |
| D-12软件需求标准 | D-40测试用例与底层需求的追踪数据 |
| D-13软件设计标准 | D-41高层需求测试覆盖数据 |
| D-14软件编码标准 | D-42底层需求测试覆盖数据 |
| D-15核查检查单 | D-43测试规程 |
| D-16核查记录 | D-44测试规程与测试用例的追踪数据 |
| D-17问题报告 | D-45软件生命周期环境 |
| D-18评审检查单 | D-46目标代码 |
| D-19评审报告 | D-47可执行目标代码 |
| D-20软件配置索引 | D-48编译、链接、加载日志 |
| D-21基线 | D-49测试结果 |
| D-22变更请求 | D-50测试结果与测试规程的追踪数据 |
| D-23软件配置状态报告 | D-51测试结构覆盖数据 |
| D-24软件配置管理记录 | D-52目标代码与源代码的追踪分析数据 |
| D-25软件生命周期环境配置索引 | D-53加载控制记录 |
| D-26软件质量保证记录 | D-54软件完成综述 |
| D-27高层需求层次结构 | D-55软件符合性评审报告 |
| D-28高层需求（条目化） |  |

表 1 最佳实践识别的数据项

#### 3.3.3 变更流程关键数据项及数据元

为了能够对变更流程中的适航符合性要求进行验证，本文根据上一小节的分析结果，对变更流程中需要使用到的数据项进行了筛选与分类，并对需要数据元级别数据信息的数据进行了数据元的确定与分类。

根据为了便于后期研究的使用（需求程度不同的生命周期数据，需要细化的数据元不同），

分类：1.配置管理记录相关等，确定用于多个变更执行时的相互顺序

2.变更管理流程相关的关键数据。

3.变更管理变更内容符合性进行审定的数据。

4.其他受控生命周期数据

5.配置管理的发布的信息等

具体数据项及数据元分类及需求见表（）。

|  |  |
| --- | --- |
| 数据项内容 | 原子性数据项 |
| a.1软件配置索引 | 软件配置索引数据 |
| a.2需要变更的基线 | a.2.1基线的受控软件库  a.2.2基线.pre  a.2.3基线.next |
| a.3变更完成后的基线 | a.3.1基线的受控软件库  a.3.2基线.pre |
| a.4配置项库，软件受控库 |  |
| a.5变更申请单 | a.5.1变更影响分析（包含受变化影响的生命周期数据）  a.5.2受影响的配置项或数据元列表（变更对象）  a.5.3软件发生错误的起始处（如果是需求变更则需求处）  a.5.4变更和行动 ） |
| a.6变更审批单 | a.6.1变更影响分析 （包含受变化影响的生命周期数据）  a.6.2受影响的配置项或数据元列表（变更对象）  a.6.3软件发生错误的起始处（如果是需求变更则需求处）  a.6.4变更和行动 ） |
| a.7问题报告 | a.7.1变更影响分析 （包含受变化影响的生命周期数据）  a.7.2受影响的配置项或数据元列表（变更对象）  a.7.3软件发生错误的起始处（如果是需求变更则需求处）  a.7.4变更和行动 ）  a.7.5问题描述  a.7.6问题报告的批准、关闭情况。 |
| 追溯矩阵 |  |

确定配置管理过程生命周期数据配置项或数据元级别的需求，能够为建模审定的模型提供对生命周期数据形式和内容的需求，而且对于形式随意，难以使用的研发方提供的生命周期数据，既不符合标准关于提供易于使用的生命周期数据的要求，更加难以应用于审定的工作，因此，研发方也可参照此数据项及数据元要求，对生命周期数据进行记录与归档整理时，尽量做好分类与标注工作，为审定提供便于使用的生命周期数据。

### 3.4 统一的变更流程

本节主要根据GSN分析目标所得出底层目标中对变更流程相关的目标要求，提出该统一变更流程。

根据对Do178c的分析结果，特别是对证据需求的作为依据，结合2.n节变更管理的基本概念和一般过程，提出一种用于适航审定的变更管理的流程，在确定该流程的过程中，主要考虑以下几点：

1. 该流程必须包含Do178c目标及过程中所涉及到的变更管理的流程要素及生命周期数据的需求，否则不符合标准要求。
2. 该流程应该尽量简洁。作为审定使用流程，理想情况应是对于实际开发的流程，既能使开发中实际使用的流程能够符合关键要素，又不对开发中提出太多于标准目标无关的流程。在开发过程中，存在的配置管理或变更管理的软件都不相同，过多的要求必然给开发造成更多困难，也可能导致无法对一些符合do178c要求，但由于审定用流程过于复杂而无法对其进行审定。

软件需求变更管理的研究与实践 赵海英 第三章 优化的统一变更管理

优化的统一变更管理 Rational的统一变更管理为基础（UCM）,同时考虑Do178C标准的相关目标和要求，加上其他促进变更管理的措施，建立的“优化的统一变更管理”

变更处理流程：1.变更评估处理流程 2.软件配置管理流程（变更实施流程）（分为两个部分，1.变更评估 问题报告、变更申请、变更审核 2. 新的基线的建立、配置项改变，、软件受控库的改变，更改的配置项的识别等，还涉及到可追溯性的判断、）

强调的是过程而非方法，不需要借助复杂的变更管理工具。

一、问题报告或需求变更，变更可以是由于出现了问题提出，也可能是有需求的变更引起的。

二、变更申请：

三、变更评估：

四、变更实施（配置管理）：

1.首先根据确定 需要重新执行哪个开发过程，每个开发过程，又要进行验证和核查，分别建立核查基线和评审基线。

2.先Check out，开始第一个开发过程，开发过程结束后，建立核查基线，进行核查，执行软件的验证及质量保证，核查通过后，建立评审基线，由软件的评审团和QA进行评审，评审通过后，建立本开发过程的审查基线，并进入下一个开发过程。

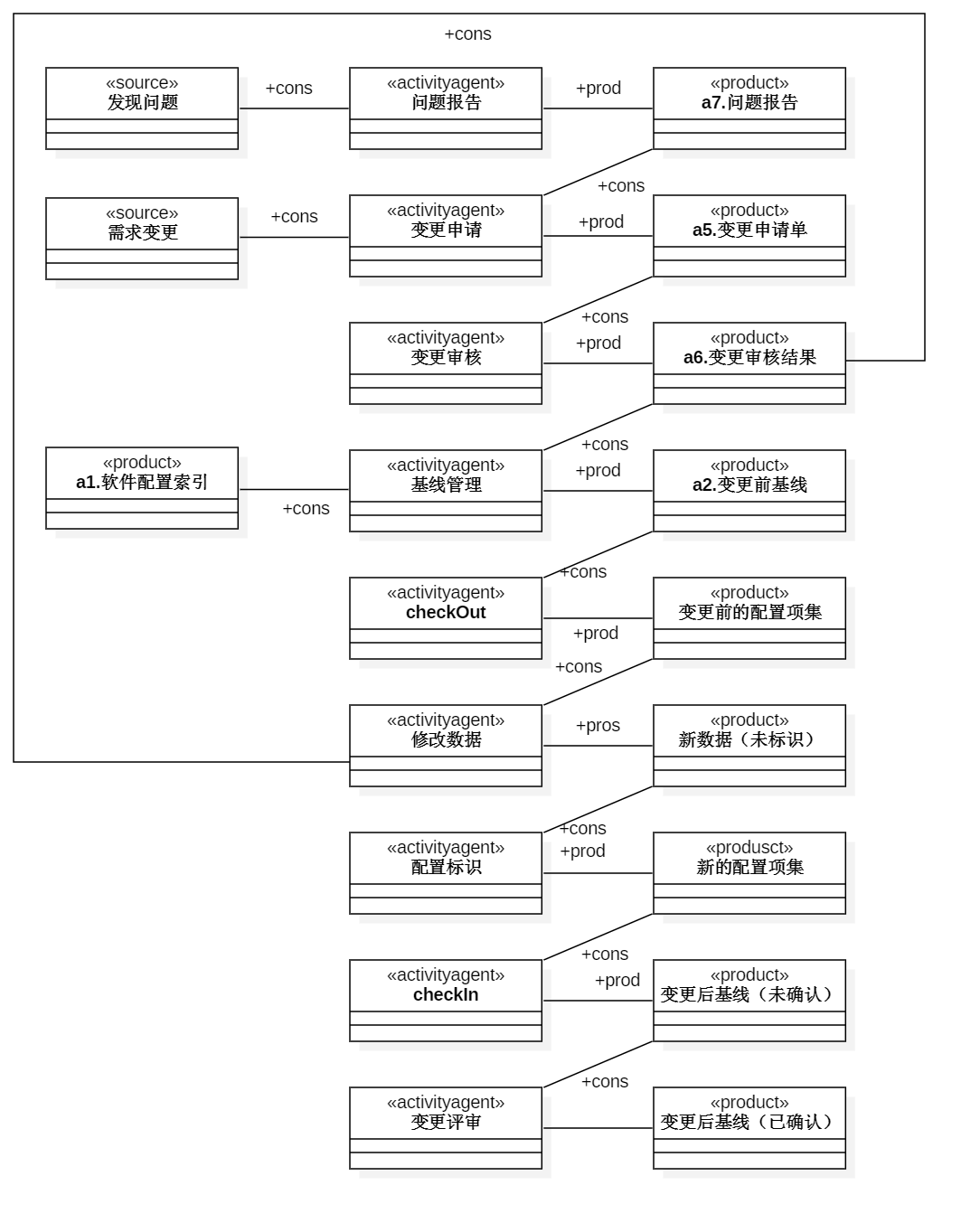
3.当所有开发过程完成，并且经过核查和评审合格后。变更实施阶段结束

五、变更核查：XX人员根据变更核查和评审报告，记忆是否达到变更目的，对此次变更进行评估。若通过，建立新的产品基线，变更完成，建立新的审查基线。若不通过，重新进入变更实施阶段。

统一的变更模型的流程图如图（）所示：

### 生命周期数据和变更流程间是输入输出关系

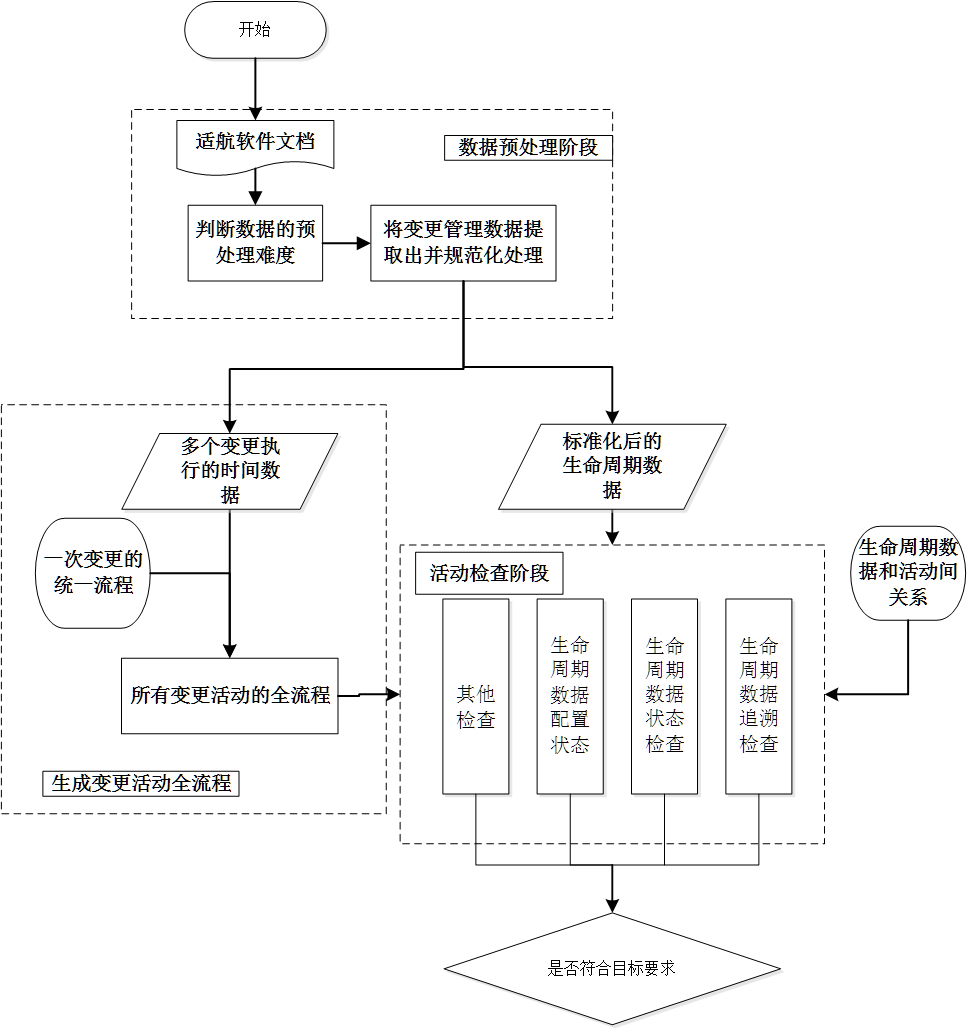
图() 展示的是各项生命周期数据和统一的变更流程间的输入输出关系



#### 数据与流程关系图

# 第四章 适航领域变更流程审定模型介绍说明

## 4.1 审定模型图



将需要达到的目标分为几类。1.追溯性。2。需满足的时间、人员、归属、已被验证、顺序关系、并发关系、 一致性等关系。先后关系、协作关系、数据的传递关系乃至资源的共享关系等。2.X还有多次变更间的关系，3.对于充分性的要求，无法进行判断，只能在初审（根据审定指南相关部分，由审查人员人工提前审查）。

## 人工审查（包含预处理部分和辅助审查部分）

 在预处理时，由审查人员根据研发方提供的生命周期数据的形式、精细程度、处理难度等情况，对生命周期数据进行初步审查（充分性等），确定此次审定的精细化层次，然后按照审查模型要求的生命周期的格式将原始生命周期数据进行处理，标准化该数据。

## 流程审查

### 4.3.1 变更活动全流程

统一的变更管理流程确定了单次变更的执行顺序，为了对各个多个变更的流程顺序进行确定，以变更批准时间、变更的完成时间为准，确定多个变更执行的顺序关系，由变更执行的顺序关系与单次变更的统一的变更流程，即可以确定 整个生命周期活动中所由变更活动的流程顺序。 同时 对 可能确定时间的各项活动流程的时间，进行时间是否有冲突问题的检查

### 4.3.2 变更子活动工作（检查）

确定每个活动应该进行哪些操作。办什么事儿，进行哪些检查。

生命周期过程从第一次变更的第一个活动开始，在通过每一个活动时，都根据研发过程中该活动所执行的任务，做几项工作。1.配置状态。已生成状态，2.检查生命周期数据是否齐全，生命周期数据的属性要求是否齐全，是否存在冲突、矛盾、缺失等 与生命周期数据的目标要求不符的情况。

DO-178C没有明确定义和解释活动之间的各种关联关系，例如，先后关系、协作关系、数据的传递关系及资源的共享关系等。

配置管理记录确定变更的顺序，

问题报告、变更申请和变更审核结果 可以由配置管理记录取得， 并可以得到 变更 前 变更后的基线及基线的受控配置项索引。

每一个过程，先检查输入的生命周期数据的 存在性和状态是否符合要求

然后对活动中生成的数据进行 已生成标注

改变状态的进行 状态更改操作

根据生命周期数据，决定下一个流程是什么。

全过程，对 执行人进行检查

对 数据项内容的合理性进行检查

过程需要的生命周期数据， 生成哪些，输出的生命周期数据

一次变更管理活动代理中各个活动过程应进行的操作和检查：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 检查哪些存在性（检查了流程顺序和生命周期数据存在与否） | 检查状态  （查看了流程的顺序） | 配置哪些状态（为检查流程顺序做准备） | 检查追溯性要求 | 检查时间性要求，若数据中有时间 | 检查其他特点 |  |
| 问题报告（需求变更分析） | 问题报告（各个要素是否齐全），问题报告的提出人是否符合权限 | | | | | | |
| 变更申请 | 变更申请各个要素是否齐全。包括变更申请人，（追溯性要求） | | | | | | |
| 变更审批 | 变更审批 各个要素是否齐全（要变更影响到哪条基线（得到准备进行变更的基线），变更的起始项，影响到哪些其他配置项），审批的结果。（追溯性要求）， | | | | | | |
| 基线管理 基线checkout | 检查需变更的基线状态是否为存在且已生成 | | | | | | |
| 配置项checkout | 从基线中获得配置项，改变基线状态为锁定，检查配置项是否存在，是否为已生成，改变配置项标识状态为已锁定。 | | | | | | |
| 开发过程，执行数据的更改 | 在生命周期数据中定位，此次变更建立的新基线（若变更成果），开发新已生成的配置项及其受控配置项库。 | | | | | | |
| 配置标识 | 新的配置项标识是否已存在，是否已为未生成状态，标识是否与其他标识重复，并将其标记为已生成状态。 | | | | | | |
| 新数据在开发库 | 对新生成的配置项进行追溯检查。 | | | | | | |
| 变更审核 | 变更审查的各个要素，变更审查的结果，审查人 | | | | | | |
| Check in |  | | | | | | |
| 建立新基线 | 检查生成的新基线是否在提供的生命周期数据中已存在，是否标记为未生成，并将其标记为已生成，）新基线的受控配置项库 | | | | | | |
| 基线管理 | 检查该基线是否已在配置项库中，将新生成的基线放入基线库，并改变状态。 | | | | | | |

## 变更内容审查

对变更影响项分析和新生成基线的审查方法

### 4.4.1方法说明

变更管理的关键因素中包括变更影响分析，应该属于变更理中的核心内容，然而对流程的分析无法对具体的配置项中的内容进行审定，来确定其是否符合相关目标要求。因此为了对需求变更影响分析、需求变更后新生成的配置项是否符合追溯性要求进行检查，本位提出了额使用文本相似度加选择排序重建追溯数据进行审定的方法，

该方法适用于对追溯性体现的较为明显的生命周期数据的变更进行验证审查，具体为 系统需求、高级需求、低级需求、代码等。

具体验证逻辑如下：

**1.变更前，判断变更影响分析的正确性。**

变更执行前，使用文本相似度计算的方法判断系统需求、高级需求、低级。。。间的相似度，并根据计算出的相似度，确定各个层次需求间的追溯关系，将该追溯关系与实际变更中被更改项、受影响项进行比较，初步判断变更影响的正确程度。若是提供了追溯矩阵，则可以用于对追溯矩阵进行检查。

**2.变更后，判断新生成配置项追溯性的正确性。**

变更执行后，计算被变更的需求与受变更影响新生成的配置项的相似度，得到新生成的配置项是否满足追溯性关系，来确定变更实施是否正确。

通过图+说明 表述清楚

### 4.4.2 需求追溯矩阵重建方法

下一章将提出利用文本相似度加选择排序 重新需求数据间的追溯性方法，并通过实验验证该方法的效能。

下一章介绍利用词向量计算相似度（）的方法及改进。

下一章对相似度算法进行验证

Wordvec是目前用于计算文本相似度的较为常用的方法，获得生命周期数据的追溯性的过程包括：一、训练词向量 二、获取易于处理的生命周期数据格式 三、将生命周期数据映射为词向量并计算相似度 四、根据设置的相似度阈值确定生命周期数据间的追溯关系。

对于变更前基线LinePre， P为全部系统需求的集合，HT为全部高级需求的集合，LT为全部低级需求的集合，对于一次由高级需求改变而引起的变更，经变更后，建立新的基线LineCe，新的系统需求集合为ST\_c,高级需求集合为HT\_c，低级需求集合为LT\_c。

该需求对应的配置项C1发生改变，该变更引起的其他配置项的集合为Q，经过变更后，在新的基线中，原配置项C1变更为配置项C1\_m,并生成了新的配置项集合Q\_m,

4.2利用词向量计算相似度的方法

利用 词向量Wordvec是目前用于计算文本相似度的较为常用的方法，获得生命周期数据的追溯性的过程包括：一、训练词向量 二、获取易于处理的生命周期数据格式 三、将生命周期数据映射为词向量并计算相似度 ##### 3.5 选择排序 四、根据设置的相似度阈值确定生命周期数据间的追溯关系。



对于语料库的获取，我们选取了维基百科数据集，并加以处理，获取专业性更强，适用于软件开发的用来训练词向量的语料库。并以此训练词向量作为文本相似度计算的向量使用。

然后通过预处理获取需要进行相似度计算的生命周期数据，并映射为词向量，输入到计算文本相似度算法中

然后，将信息检索中查询的概念与待建立跟踪关系的需求文本等价，即把每个需要建立跟踪链接的软件需求文本看作是信息检索中的一个查询语句。

算法对每一个查询语句进行文本相似度计算，最终每个查询都得到一个候选的排序列表，排在最前边的是与相应查询相关度最高的软件制品。

下面主要详细介绍对文本相似度计算算法的改进：

#### 4.4.3 文本语义相似度计算方法

本节基于TF-IDF方法权重计算与缺失词替代，提出一种提高文本相似度计算效果的方法。

##### 4.4.3.1基于tf-idf策略计算权重的关键词相似度加成算法

###### 4.3.1.1基于TF-idf策略计算权重

在计算文件相似度时，是先计算词与词之间的相似度，再计算出词与句子的相似度，最后再计算文本与文本间的相似度。但是不同的词能够代表文档信息的程度是不同的，如果将所有的词一视同仁，就忽略了关键词或词组对文档相似度影响更明显这一情况。因此，要对文档中的词计算特征权重，通过计算权重选出更能代表文档信息的词或词组。此处使用TF-IDF算法找出权重足够大的词汇TF（Term Frequency）是词频，如公式（）所示，指的是目标词在文档中出现次数的频率，tf与目标词能够表示文档信息的程度成正比，；IDF（Inverse Document Frequency）是逆文档频率，如公式（）所示，IDF与含有目标单词或词组的文档数量有关，用来表示目标词在其他文档中出现的频率，IDF值越大，代表目标单词在其他文档中出现的频率越小，则目标词或词组能更大程度的表示文档信息；最后将求TF与idf的乘积，如公式（）所示，得到该目标词的tf-idf值。

(3.1)

(3.2)

其中指目标词在文档中出现的频率，指单词在文档中的词频，表示文档中目标词出现的次数。指目标词在文档集合中的逆文档频率，表示文档总数，表示出现过目标单词的文档数，为了防止出现过目标单词的文档为零而导致分母为零，将该文档数加一处理。最后取对数处理。

###### 4.3.1.2 对关键信息词进行加权计算调整后相似度

使用tf-idf选出能够更大程度反映目标文档信息的关键词后，需要使用专门加权方法计算相似度，以合理使用关键词所携带信息。

当计算关键词与其他词的相似度时，若两词的相似度大于阈值，则进一步按照一定比例提高相似度，应满足 ，具体数值通过实验确定，若两词的相似度小于阈值，则按照一定比例降低相似度，应满足 ,具体数值通过实验确定。计算方法如公式（）所示。

代表经过加权计算后得到的词与词之间的相似度， 代表直接计算得出的词与词之间的相似度，为取值大于1的常数，的值决定了对相似度进行调整的幅度。当或时，应用此公式。

当时，不对相似度进行调整，直接使用作为词与词间最终相似度。

4.3.1.2.1以下为选出tfidf权重较大的单次或词组集合的步骤（意思不变，改，防相似）

在论文中，计算两个词之间的语义相似度我们首先将词表示为词向量的形式，然后使用cosine相似度计算，如公式(3.4)所示。

(3.4)

其中和分别表示词和的词向量，和分别表示两个词的词向量的长度。

合适的查询扩展方法和加权策略能够提升信息检索任务的性能**错误!未找到引用源。**。对某个文档进行查询扩展，是对文档中的各个单词或词组进行扩展，步骤如图1所示，描述如下：

（1）首先计算TFIDF权重，然后根据TFIDF进行排序，选择前的单词或词组进行扩展，组成集合，其中参数设置为经验值0.3。

（2）对以上选择出的前的单词或词组进行扩展，使用单词语义相似度计算公式，选择出相似度大于阈值的单词或词组，每个被扩展的词组成集合，形式化的表示如公式(3.5)所示。其中参数同样设置为经验值0.7。

(3.5)



图1 查询扩展流程图

##### 4.3.3.2对基于缺失词的处理策略

对于缺失词，建立词汇表，他的相似度，等于“相关词汇”的相似度的平均值。但计算词汇间的相似度时，先确定是否为同一词。

原来的策略：1.不考虑该词，2.统一替代为某一向量 3.？？？

对停用词的相似度计算方法

本文提出一种，由人工对缺失词 使用相关词 组合 进行替代计算的方法，具体为将 在计算某缺失此 A 与 另一词 C的相似度时，该相似度等于 A的替代词组“” 中每个词与c的相似度的平均值，

这样既可将缺失词也考虑到计算中去

缺失在计算相似度时，首先判断另一词是否为与该缺失词为同样的词，若相同，怎相似度为一，若不相同，再按照以上计算方法进行计算。

若某词汇属于缺失词，但 不在idf词库，直接使用向量表示

若某词汇术语缺失词，同时在idf词库，表示改缺失词较为重要，

##### 4.4.3.3改进的文本语义相似度算法

最后基于以上权重调整策略与缺失词处理方法，提出优化的文本相似度计算方法：

Ye X等人在**错误!未找到引用源。**提出了用来计算文本与文本间相似度的算法W2V，该方法计算文本语义相似度的步骤为：

1）通过计算两个词向量的cosine相似度得到词与词之间相似度，如公式（）所示；

(3.4)

其中和分别表示词和的词向量，和分别表示两个词的词向量的长度。

2）计算单个词与文本之间的相似度，如公式 （）所示，计算目标词与文本中所有词的相似度，取最大值即为目标词w与文本T的相似度。

(3.6)

3）计算文本与文本间的相似度。如公式(3.7)所示，算出文本中的每个词与文本的相似度，并求和，最后进行归一化处理，即得到文本与的相似度。

(3.7)

(3.8)

其中表示文本中与文本的相似度不为0的词的集合。

（1）在计算词与文本的相似度时，添加上查询扩展，于是重写了公式(3.6)，如公式(3.9)所示。添加上查询扩展后，同时还设置了参数来分配原词的相似度和扩展的词的相似度之间的权重，该值的取值范围为，在实际的计算过程中，该值在0.5到0.9之间，以0.01的步长增长，遍历整个范围，当效果最好时，设定值，因此参数根据数据集的不同而不同。如果当前单词没有扩展词，则使用与论文**错误!未找到引用源。**中相同的方法，即使用公式(3.6)计算。

(3.9)

其中表示查询扩展项，即当计算词与词之间的相似度时，如果该词具有扩展词，则扩展项使用词的扩展词集合中的词与词之间的相似度之和，然后做归一化处理，保证的值的范围在0到1之间。举个例子，单词“technique”的扩展词有“technology”、“method”和“approach”等，当计算单词“technique”与句子“The basic requirement of planes is safety.”的相似度时，不仅仅计算“technique”与句子的相似度，同样会计算其扩展词“technology”、“method”和“approach”与该句子的相似度，最后使用比例参数将两者结合起来。

下面是对以上算法的改进部分

（1）首先计算TFIDF权重，然后根据TFIDF进行排序，选择前的单词或词组，组成集合，其中参数设置为经验值0.3。

（2）改写计算词与词之间的相似度的计算方法，当两个词都不属于集合中时，算法不变，仍然按照公式（）计算，得到词与词之间的最终相似度，当两个词中有任一词属于集合时，计算出两词的相似度，并将

与设置好的阈值 、比较，

若，则

；

若或，则

其中设置为经验值0.3，设置为取经验值0.7，r设置为取经验值3。得出的作为最终的词与词间相似度，继续参与下一步的计算。

两词之间计算相似度的公式改写为公式（）。其中 a,b,c分别取经验值。。。。

对以上选择出的前的单词或词组进行扩展，使用单词语义相似度计算公式，选择出相似度大于阈值的单词或词组，每个被扩展的词组成集合，形式化的表示如公式(3.5)所示。其中参数同样设置为经验值0.7。

(3.5)

（3）对原算法的第二步进行改进。在w2v算法中，算法得出的相似度与两个文本的输入顺序是相关的，两个文本在算法中并不完全对称，差异出现在计算词与文本间相似度时，不同的文本顺序会得出不同的结果；另外，当短文本是长文本的子集时，若拆分的文本是短文本，长文本中不属于短文本部分的词将会被忽略，计算出的相似度为1，与实际情况明显不符。为了保证计算文本相似度时计算结果的一致性，同时为了避免特殊情况下的错误，在新的算法中，作出如下规定：

计算两个文本间的相似度时，设较短的文本为，较长的文本为，则必须拆分长文本,通过计算中每个词与短文本的相似度，获得文本间的相似度。

#### 4.4.4学习排序算法

3.3 学习排序算法

在以往的需求可追踪链接研究中，通常只使用文本相似度一个特征作为判断软件文档间追踪关系的依据。而在目前的文本检索，自然语言处理等领域，使用机器学习方法结合文本中的多个特征，如词频、文档逆频率，文本长度等来对文本进行检索、分类已经在得到验证。因此，本文将学习排序应用到需求追踪任务当中。在上一节提到，通过计算每一条查询语句与候选文本间的相似度，可以产生一个排序列表，这个排序列表可以作为学习排序的输入。下面小节中将介绍学习排序算法和选取的特征。

##### 4.4.4.1排序算法选择

在第二章中提到，按照训练样本的不同，学习排序算法可以分为单文档排序、文档对排序和文档列表排序三种方法。其中文档列表排序每次训练对象是整个候选列表，在数据量足够的情况下，拥有最好的效果；而单文档方法由于只考虑了单个文本的特征，效果不如文档对方法。因此，综合效果和实验数据集的规模，本文选择了相对合适的文档对排序算法。 本文根据任务特点选择了针对信息检索领域的排序算法IR SVM，该算法是Ranking SVM在信息检索领域的改进。

###### 3.3.1.1 Ranking SVM

Ranking SVM排序算法主要分为将文档对的排序问题转化为分类问题和使用SVM算法分类模型训练并求解两个步骤。

假设给定查询-候选文档列表，并用一个维的特征向量来表示候选文档，然后输出空间用与查询的相关级别用 来表示，表示文档相关级别的数量，并且相关性级别按照递增排列。本节使用第二章中提到的排序函数将特征计算排序分数， 可以表示为公式(3.6) ，“”表示文档将相对位置关系。

 (3.6)

Herbich等人[51]在2000年首次提出可以将文档对排序问题转化为分类问题。其转化的推导可以假设排序函数为线性函数(可以为任意函数)。表示为公式(3.7)：

 (3.7)

其中是权重向量，该公式表示排序函数为权重向量和特征向量的内积。结合公式(3.8)，可以得到：

 (3.8)

可以由公式(3.8)看到，对于任一文档对，候选文档中 表示文档在文档的靠前位置。可以通过将文本特征向量重新组合为，结合公式(3.7)则可以得到两者相对位置的判别函数。根据线性函数的特性，可以得到每一对文档对相对位置的标签。接下来，可以依次求出候选文档列表中任意一组文档对组合的向量，并为其重新打标签。本文用和分别代表文档对中的第一个和第二个文档特征，两者间先后关系的标签用来表示，形式化的表示如公式(3.9)。

 (3.9)

这样就可以把给定的排序列表作为训练集，构建一个新的带有文档对顺序标签的数据集，见公式(3.10)。

 (3.10)



图2 原始排序问题

这里可以举例说明，假设候选文档与查询文档中共有三个相关等级，现在有两个查询分别对应两个候选文档列表，根据公式(3.7)，每个查询会计算得到一个权重向量，可以用图15形象表示。而本文使用文档对排序方法 ，根据公式(3.10)将文本特征向量两两重新组合为：、和，并给他们打上相应的标签，然后就可以将排序问题转化为了分类问题，如图16所示。



图3 排序问题转化为分类问题

本节使用作为分类算法的训练数据集，使用支持向量机(Support Vector Machine, SVM)作为分类算法，+1表示，为正样本，反之，-1为负样本。这样排序问题转化为0-1分类问题，根据SVM求解过程可以得(3.11)：

(3.11)

在这里是松弛变量，是训练集中实例数量，是第二范式，把松弛变量代入公式(3.12)中，可以得到：

(3.12)

在这里，公式加和的第一项为合页损失函数，第二项是防止过拟合而加入的正则项。

###### 3.3.1.2 IR SVM

Ranking SVM算法作为学习排序模型的排序算法应用于信息检索领域时[43]，有两点不足：

(1) 在上一小节提到，Ranking SVM排序算法的主要思想是将排序问题转化为分类问题。它在训练的时候把不同级别差值文档组，如和，其中 表示与查询相关的等级，同等对待，假如在训练实例中，越多，那么最终得到的结果也会越偏向于。而在信息检索领域的问题中，最终的结果列表中，文档排名越靠前，即r值越高，它对最终检索的结果影响越大，显然Ranking SVM在这里有待改进。

(2) Ranking SVM还有一点没有考虑的因素是，不同查询语句所对应的查询语句对所得的候选文档同等对待，这显示是不合理的。在这里可以举例说明：仍然假设文件的相关等级分为三个等级级别，现有两个查询，每个查询所对应的候选文档如表4所示，从中可以看到，对于结果中可提供正样本数量为14，多于可提供的8个，而由于Ranking SVM对不同查询语句所对应的查询语句对所得的候选文档同等对待，这也会影响最终检索结果。应该调整算法，使文档在相关等级相同的情况下，正样本多的在训练过程中应该对最终结果影响大于。

表2 排序列表实例

|  |  |
| --- | --- |
| **查询** | **结果** |
|  | 文档相关等级分别为：r3, r2, r2, r1, r1 |
|  | 文档相关等级分别为：r3, r2, r2, r1, r1, r1, r1 |

由于Ranking SVM算法的以上两个不足点，本文使用了它在信息检索领域的改进算法IR SVM，通过对它的优化Ranking SVM的损失函数，将0-1分类问题转化为代价敏感的二分类问题。具体来说就是，对于同一查询下的不同文档对，相关级别越高在最终排序结果中的重要性越大，这时候，对该文档对赋予较大的损失权重；对不同查询语句下的候选文档列表，为列表条目越少的文档对数赋予较大的损失权重。基于以上IR SVM改进损失函数可以表示为公式(3.13)。

(3.13)

在这里表示第个文档对的相关等级，表示等级下的权重值，为第个文档对所对应的查询，表示查询对应的相关文档对的参数。和作为惩罚因子，分别用来弥补上述Ranking SVM两个方面的不足。

对于参数值的确定，使用的是一种启发式算法：首先选定一种评价指标，如MAP, NDCG等。对于每条查询语句或者文档，首先找到它的最佳排序列表，然后随机挑选其中一对文档，交换两者的位置，记录下指标的变化值，重复该过程，直到所有文档对交换完成，最后使用评价指标变化的平均值作为参数的值。对于参数，它的主要作用是平衡候选文档列表条目数目所带来的影响，可以用最大候选文档条目和当前查询候选文档条目表示，可由公式(3.14)表示。

(3.14)

其中，代每条查询所对应的候选文档数量。于是可以简单的计算出来。

###### 3.3.1.3 求解IR SVM

令时，公式(3.13)可以等价变化为二次规划问题，如公式(3.15)。

(3.15)

|  |
| --- |
| **算法 1**: 坐标下降法 |
| 初始化为0,其中，令  迭代直至算法收敛： |

本文使用坐标下降法(Dual coordinate descent method)求解对偶问题来间接求解公式(3.15)的对偶问题从而得到它的最优解，首先引入拉格朗日引入拉格朗日乘子，然后对其求极大极小问题后，可以得到它的对偶形式，如公式(3.16)所示。

(3.16)

坐标下降法每次选择一个进行优化，记为迭代指定次数或目标函数值变化量达到指定精度时停止，其过程如算法1所示。

##### 4.4.4.2 特征选择

在学习排序中，影响最后排序结果的另一个因素是文本特征的选择，在以往的研究工作中已经证明一些特征在信息检索领域的有效性。本文从常用的特征中选取五个特征作为排序算法输入。这五个特征中的前四个是依赖查询特征，这些特征强调了查询和候选文档间的关系，最后一个是不依赖查询特征，体现了候选文档本身的特点，主要作用是可以用来加快查询速度。

(1)文本相似度：表示查询查询文档与候选文档间语义的相似程度，可以由本文提出的相似度算法CWI计算得到。

(2)IDF之和：查询中词语在候选文档列表中的IDF值之和，它体现了查询在其对应的结果集的重要程度，可用公式(3.17)表示。

(3.17)

在这里代表查询文档中的单词，代表查询所对应的候选文档列表。

(3)广义Jaccard系数：通过文档向量表示查询文档和任一候选文档的上下文相似度，其计算方法由公式(3.18)所示。

(3.18)

这里代表查询文档的向量表示，是候选文档的向量表示，它们可以由公式(3.3)计算得出。

(4)缺失词数量：即在使用词向量表示文本时，缺失词的数量，体现缺失词语在查询中的重要程度。

(5)文本长度：表示查询候选文档列表的长度，学习排序中常用的不依赖查询特征，可以一定程度体现结果的有效程度。

## 4.5 本章小结

本章首先介绍了面向发动机适航的软件需求追踪算法的模型框架，然后介绍了基于词向量的文本相似度算法，算法包括了语义相似度的计算和缺失词语的动态处理，最后，介绍了机器学习排序将排序问题转化为0-1分类问题的原理，并给出了损失函数的优化和推导过程，结合当下的流行的机器学习技术，通过组合文本中的多个特征从而提升结果精度。