1. 第二章： 留在弄完其他的再弄，但是需要的时间也可能短不了。 不确定多久

第三章： 一整块儿，今晚优先完成 2天 2月27日晚，进行到2点半，将本章内容至少完成80%. 今天3月3日，要将该部分完成

第四章： 1对于整个框架怎么框进去 1天 该部分已基本完成

2对于相似度算法如何改进， 1天

3还有把选择排序？机器学习？的内容加进去 2天

第五章： 根据第三章内容的修改，把第五实验做出相应修改。 1天

整体文章理顺，1-2天。

目前仍需完成的任务：

* + 第一章绪论和第二章相关技术 共计约15000字，需要找材料 并撰写 暂定留10个小时 2天（可能不够）
  + 第三章 Do178c研究 任务量仍然比较大，需要自己写 时间不好预测 尽量多留一点 暂定 15个小时 3天
  + 第四章 算法改进部分 算法改进需要补充 缺失词部分 4个小时 1天
    - * 实验验证部分 实验部分（语料库生成1小时、数据预处理1小时、实验结果分析2小时）需要改 说法 共4小时 1天
  + 第四章怎样串 比较麻烦 需要自己想自己写 有难度 5小时 1天

**暂定每天至少5个小时，且必须完成最少的任务**

**a.2019年2月24日星期日 改写 算法验证部分**

**第五章 要尽量往前提**

* 一周之后，找赵腾，具体看怎么处理

2412 绪论前自己写的文字 主要是需要写摘要和翻译英文

7283 绪论 2天 全天 10个西红柿

1. 6000 相关理论和技术研究 2天 全天 8个西红柿
2. 6600 模型的建立 3天 工作日 20个西红柿
3. 4.1 文本相似度算法的设计 1972 1+1天 工作日 5个西红柿

4.2 算法的验证 2238 5296 1+1天 工作日 20个西红柿

4.3 怎么实现功能 1天 工作日

1. 设计与实现 4288 带设计 20 天
2. 对目标进行分解，呈列出要实现的“小目标”，再根据与生命周期数据间的关系
3. 根据Do178c规定的生命周期数据类型，参考Do178c实验给出的数据元，
4. 确认进行建模使用的方法：安全UML的精简版本
5. 选择安全UML作为**建模的方法，**（使用到了哪些相互间关系）

**二、配置管理过程相关的元数据，**

三、分析配置管理目标、过程、证据间的关系，（利用GSN）,

介绍：

对于航空机载软件的研制和适航审定，FAA确定了Do-178c、Order 8110.49chg1标准，其中，从适航符合性角度，Do-178c是基础标准，它定义了开发过程需要满足的与相应安全等级相适应的目标；Order8110.49 chg1则属于操作标准，定义了如何实际操作来审查一个项目的研制过程是否满足Do-178c的相应目标要求。Do-178c标准从软件安全性角度，按照机载软件的安全级别对开发过程和活动提出了一系列目标要求；后者则从符合性审查角度规定了具体的审查阶段、活动、涉及的数据和相应的审查规则等。

## 配置管理过程建模方法的选取

由于所建立模型的过程、数据与目标间的复杂关系，选择一种适用于此次建模的方法是一个难点。

由于过程建模是对软件过程活动的再工程,涉及到软件产品的开发与维护、软件项目管理、过程管理与过程改善等各个方面,涉及到过程活动、角色、产品、资源和约束等各种过程成分,还涉及到建模所用的形式化方法,加之软件过程本身具有的复杂性,使得构造过程模型的方法也是多种多样的。

* 以**过程所涉及的各种成分**(如活动、角色、产品、资源和约束等)为出发点来考虑过程建模。主要有四种不同方法：

1. 以活动为中心的建模方法
2. 以角色为中心的建模方法
3. 以产品为中心的建模方法
4. 基于过程模板的建模方法

* 以软件过程模型侧重于描述**功能目标**还是侧重于描述**活动行为**。主要考虑过程模型是否支持软件过程的动态性进行分类

1. 面向活动的过程建模方法
2. 面向目标的建模方法
3. 面向活动和面向目标的建模方法的结合。

* 按照过程建模所采用的不同形式化方法的分类。

1. 过程程序设计方法
2. 多功能分解方法
3. 基于计划的建模方法
4. 基于Petri网的建模方法
5. 基于规则的建模方法
6. 基于逻辑的建模方法
7. 基于知识的建模方法
8. 混合风格的建模方法

基于建模目的以及配置管理过程特点的考虑，本研究所准备采用基于过程模板、面向活动和面向目标结合、基于Petri网与UML集合的建模方法，并在研究过程中跟进实际情况加以考虑变更或改动。

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，并且识别了以下数据项（表 3）：

|  |  |
| --- | --- |
| 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） |  |

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。

在Do178c标准的71个目标中，以下6个是与配置管理相对应的目标。

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）

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，能够减轻对整个论证过程进行分析的复杂性。

|  |  |
| --- | --- |
| 数据项内容 | 原子性数据项 |
| a.1软件配置索引 |  |
| 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 |  |
| 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。 |

软件质量评估的方法，主要是针对研发方的，对于审定方的，间“证据”论文，（大多都没有太强的实用性）

选择建模的方法，实际情况，国外的研发审定工作1.目前证据管理的情况 2.国外成立的专门进行研发审定的研究、统一工作的传统，其中，1论文提到了一种安全UML，该UML的目的是， 某种程度上以后的相关工作人员都应参照此项的情况，

但此UML专注于研发，同时考虑到了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。

考虑这两点因素，本人此次建模参考安全UML标准，特别是n.n章对各个对象间关系的分类、精简与总结；同时在数据项的选择上，设定为可粗可细，（不归这里管），选取了实际工作中最关键的相关的要素，结合标准规定的数据项要素， 在软件的实际设计开发过程中，为能够适应不同格式标准情况的生命周期数据，可以选择审定项的细致程度，当然可由该工具审定的结果也根据生命周期数据情况的不同而不同。对于无法标准化或由于复杂程度只能有专家进行分析的部分，展现为易于得到的文本以利于审查人员管理证据

相关的要素包括\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，来源于\_\_\_\_\_\_\_\_\_\_\_\_

分析各个要素之间的关系,通过Do178c标准的原文、实践、咨询相关专家人员。

将需要达到的目标分为Ji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、 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。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）。

建立 对 配置管理进行审定的 软件工作的UML图

对于标准中对可追溯性的要求，提出了一种。。。的方法，

对追溯性进行判定的方法研究，方法是利用 改进软件工程领域 文本搜索的 算法。

分析 存在哪些特点：在变更管理或是基线改变中，关键字加权？，缺失词处理？

变更：主要是需求变更，也可能是错误引起的变更 所以，在变更阶段主要是需求追溯性的改变。

对于一些难以判定的，留给人工判定，

而系统则是可以提供（可选择的数据项）的追溯性的计算，审查人员根据追溯性计算 的结果，进行进一步的检查。

涉及到配置管理的数据项和数据元

1.

涉及到追溯性的

每次变更（两条基线间中的关键要素：变更申请时间，数据项，变更申请人，变更审核人员，基线，受控软件库，基线间的衍生关系，配置项的追溯关系？ 变更应记录、批准、并进行跟踪，变更的源头。

建模和设计软件的关系：

1. 若是建变更管理的模型。通过最后的数据进行验证。变更管理中的要求怎么弄，变更管理是有很多步骤的，（验证） （执行）

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

1. 统一的变更管理流程
2. 分析do178c目标，
3. 根据1、2挑选出 需要的配置项。 2、3考虑顺序。 3应是在1、2的基础上，2，可能需要根据3进行微调
4. 方法：用安全UML对统一的变更流程进行建模，并分析do178c要求的证据应在哪一步进行验证，并用安全UML表示出来。

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。

变更申请阶段：变更申请编号，变更描述，变更原因，不实施的后果 变更申请单

变更评估阶段：影响和风险评估 变更评估单

变更审批阶段：批准签名，批准日期。 变更审批结果（单？）

（三个共同要有的变更对配置项的影响。）哪些配置项

变更行动阶段的属性: 变更负责人。变更 开始执行和完成时间期限。

配置管理阶段：

变更执行完成后：变更核查日期 核查结果

变更管理流程图 放在第二章

时序图 有了，基本可以用

变更流程 基本确定

涉及的配置项 一会儿进一步明确

状态迁移图， 感觉好像没有用

用例图 需要重新画

变更管理中子活动与产品间的输入输出关系 改一改可以用，有助于理解

变更管理及子活动聚集关系图 说明用

活动代理的聚集关系 写程序时用

配置项进行分类（归总），结合 安全UML分析目标 未完成

UML图需要做几个？

提出文本语义相似度计算的方法，用于对变更管理中，对变更是否合理进行一定程度的判断，

对于由于需求的改变而进行的变更，通过对相似度的的计算来判断变更的合理性。

对于需求变化而引起的变更是否基本符合变更管理中对于追溯性的要求。具体就是检查

例如 变更的起始项是 A1，经过更改新生成里a2, 受到影响的配置项包括 b1,b2,b3,b4等等，变更管理完成后，在新建立的基线中，b1,b2.,b3,b4….已不在该基线中，而该基线中应包含新生成的配置项 c1,c2,c3,c4…

1. 计算a1与b…..的相似度
2. 计算a1 与c。。。的相似度 是否存在相似度极低的情况，根据实际情况设置阈值，对于相似度低于该阈值的方法进行人工审查

下面主要详细介绍对文本相似度计算算法的改进：

## 四、文本语义相似度计算

本节主要提出一种基于TF-idf算法的权重调整与缺失词处理方法计算提高文本相似度计算效果方法。

## 4.1,基于td的加权方法与策略

## 4.1.1基于TF-idf选择加权词

（tf-idf的介绍与计算方法）

为了区分句子中的单词的重要程度，我们选择对句子中的单词进行加权处理。本文使用TF-IDF的策略来计算，如公式(3.3)所示。TF（Term Frequency，词频）表示某个词在当前文档中的出现频率，如公式(3.1)所示；IDF（Inverse Document Frequency，逆文档频率）与含有某个单词或短语的文档数量有关，并且包含该单词或短语的文档数量越少，则该值越大，即表示当包含该单词或短语的文档数越少时，该词在当前文档中所占的权重越高，如公式（3.2）所示。当一个词或短语在一个文档中出现的频率较高，而在其他文档中出现频率较低时，则可以认为这个词或短语具有较强的类别区分能力，即能够比其他词更有代表当前文档的能力。

(3.1)

(3.2)

(3.3)

其中指单词在文档中的词频，分子表示在文档中该单词出现的次数，分母表示文档中单词的总数。指单词或短语在文档集合中的逆文档频率，分子表示文档总数，分母表示出现过此单词或短语的文档数，一般为了防止分母为0，会给该文档数加一，最后对比值取对数。

Tf idf 是指哪些文档的数量？应该是手上的生命周期数据文档数，因为

在当文档数量较少时，计算得到的tfidf值并不准确，在适航领域，

可能在软件需求相关文档尚未充足的情况下建立需求跟踪关系，因此此时需要进行数据补充。使用当前软件生命周期阶段已有的文档进行数据扩充，能够在一定程度上缓解tfidf计算不准确的问题。

### 

## 4.1.2对选出的权重高的词库集合的加权计算方法

### 4.1.2.1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[42]。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12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12 查询扩展流程图

### 加权处理策略

1. 方法：在0.5左右进行测算，看哪个个效果最好，在计算词与词之间的相似度时，若其中一方为 idf词，则计算出的相似度根据公式进行加成。

对于权重较大的词，若该词与被对比的文本相似对较大（大于0.5），则在计算时，将该词与被对比的句子的相似对继续加大。若该词与被对比的文本相似对较小，则在计算时，将该词与被对比的句子的相似对继续减小。

词与词？词与句子？控制词与词的比较方便。

公式

基于缺失词相似度的计算方法的改进。

4.2

### 4.2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，

### 4.3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：

在论文[16]中，文本语义相似度的计算有三个步骤：1）计算词之间的语义相似度；2）计算词与文本之间的语义相似度；3）计算文本之间的语义相似度。词与词的相似度计算如公式(3.4)所示，词与文本的相似度计算即为词与文本中所有词相似度的最大值，如公式(3.6)所示。文本与的相似度为文本中的每个词与文本的相似度的和，然后做归一化处理，如公式(3.7)所示。

(3.6)

(3.7)

(3.8)

其中表示文本中与文本的语义相似度不为0的单词或词组集合。

以上即为论文[16]中提出的方法，我们称之为W2V方法。相对于该方法，我们做出如下改进：

（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[16]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。

（2）在计算文本之间的相似度时，原方法的公式是不对称的，即对于同样的一对文本和文本，不同的输入顺序会产生不同的结果。为了保证一致性，以及结果与语句长短的无关性，本方法增加了一个限制，即在输入时使文本的长度长于文本。增加该限制主要有两个原因：第一是使得该公式是对称的，当计算任意一对文本的相似度时，使得结果与二者的前后顺序无关；第二是当较短文本是较长文本的子集时，避免较长文本中的其他词汇被忽视。第二个原因是一种特殊情况，举个例子：文本和文本分别是“Airworthiness is necessary.”、“This document emphasizes that airworthiness is necessary.”，在使用公式(3.9)计算时，文本中的每个词在文本中都出现了，因此在使用公式(3.7)计算词与文本的相似度时，文本中的每个词与文本中的词的相似度的最大值均为1，此时文本中的其他词被忽略掉了。为了避免这种情况，限制两个文本的前后顺序是重要的。