其他研究提出了许多对其进行开发或审定的方法，但由于软件的规模、特点、要求等的不同，此项工作涉及到的管理软件，开发计划、配置计划等，在开发中都会有很大的差异， 针对Do178c的针对于复杂的开发环境、开发工具等， 对于工具的使用和过程的控制方法等，过大的差异和过于复杂的过程，使得 对配置管理过程的审定只能依靠专家人工的方法。

本研究为了解决此项问题，只专注于 审定标准中 规定好的 目标的满足和审定，对于具体研发过程中的工具、流程规定等， 部分可由专家直接通过 经过验证的 工具进行审定，部分经过预处理后进行验证， 而研发方在制定开发计划时，也可在于审定人员沟通时，参照在此模型的基础上 制定软件配置管理计划的总思路，对于其他需要增加或者考虑的部分，再另行规定，单独审定。

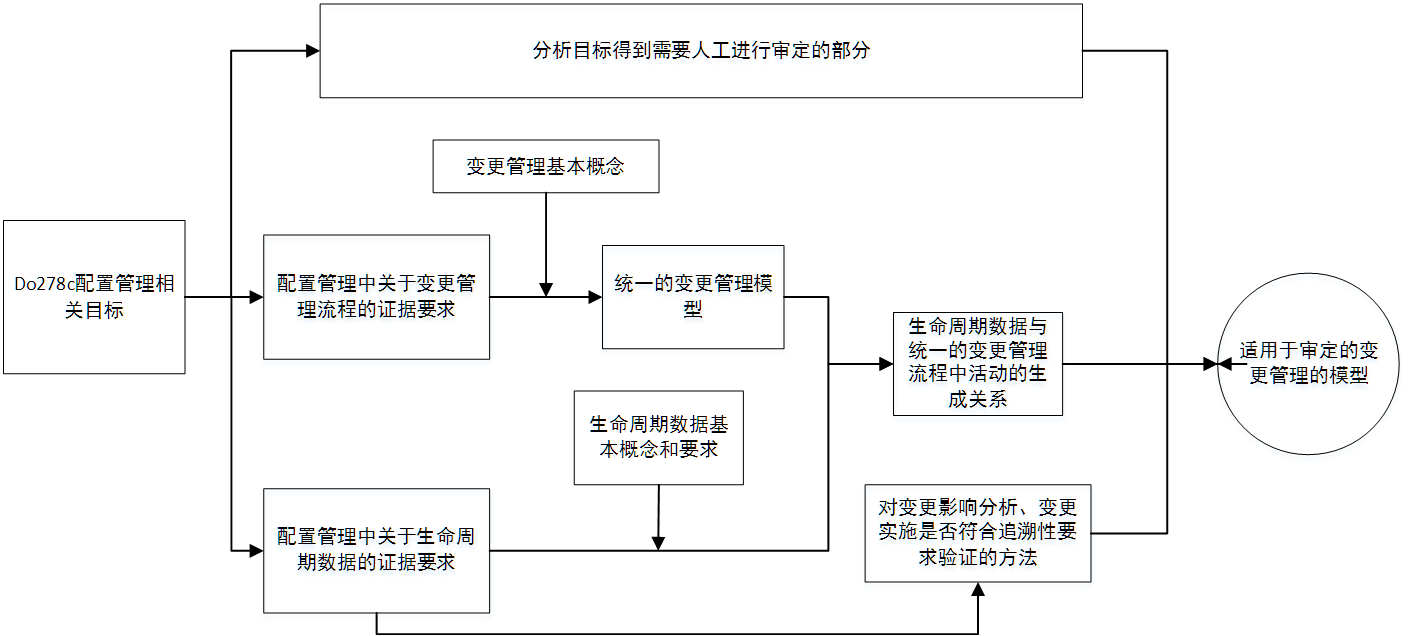
有的只专注于研究开发过程中如何符合标准，有的对于审定方法的研究主要还是比较理论的层次， 如 具体项目配置管理软件的设计， 针对项目无关的目标验证模型（也是针对于开发的）等

# 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：

标识构型项（A-8.1）

建立基线和可追溯性（A-8.2）

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）

建立归档、检索和发布机制（A-8.4）

建立软件加载控制机制（A-8.5）

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：

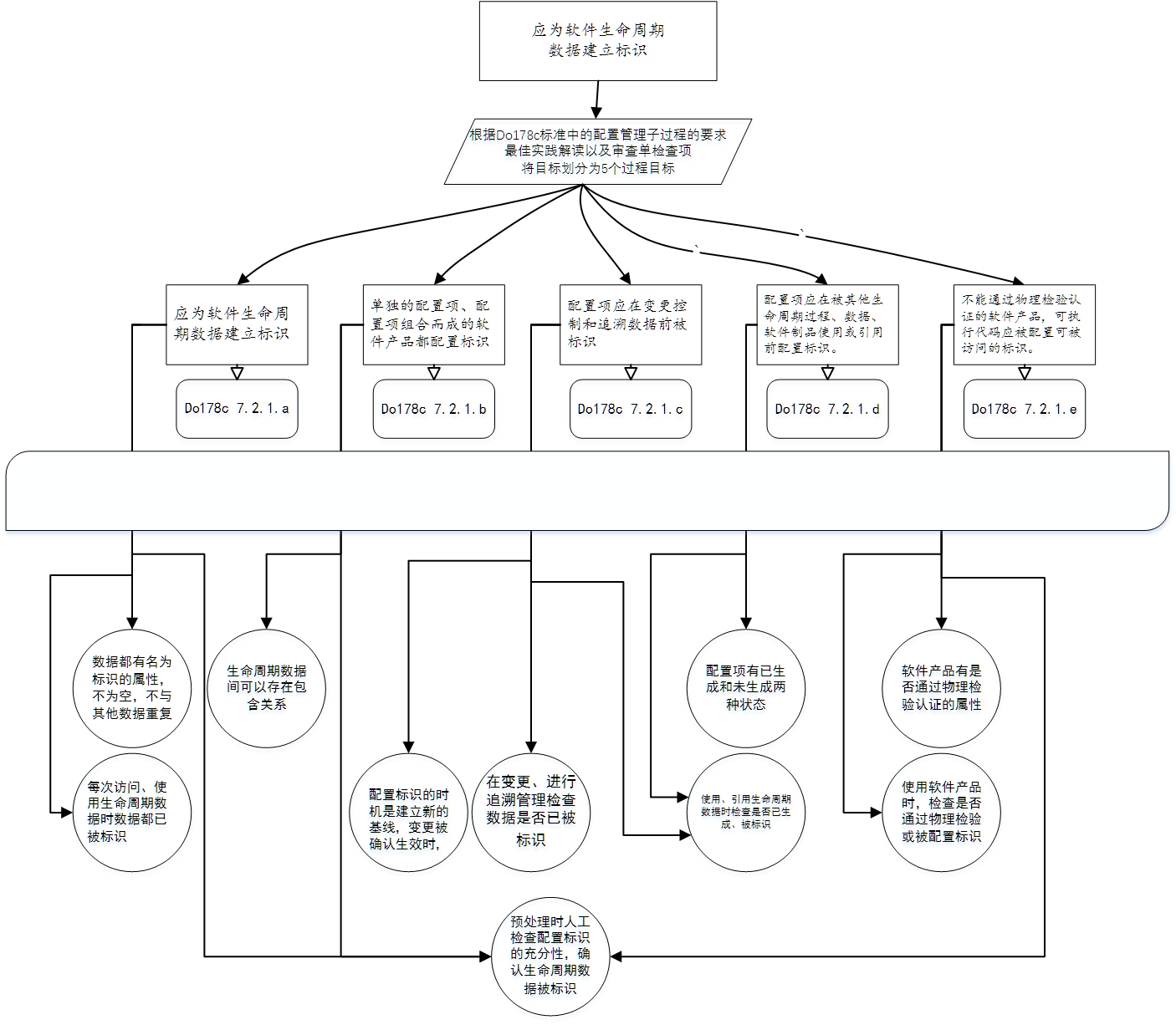
1在数据初始化和论证过程中需要人工辅助的证据要求。

2.主要对生命周期数据自身相关的证据要求。

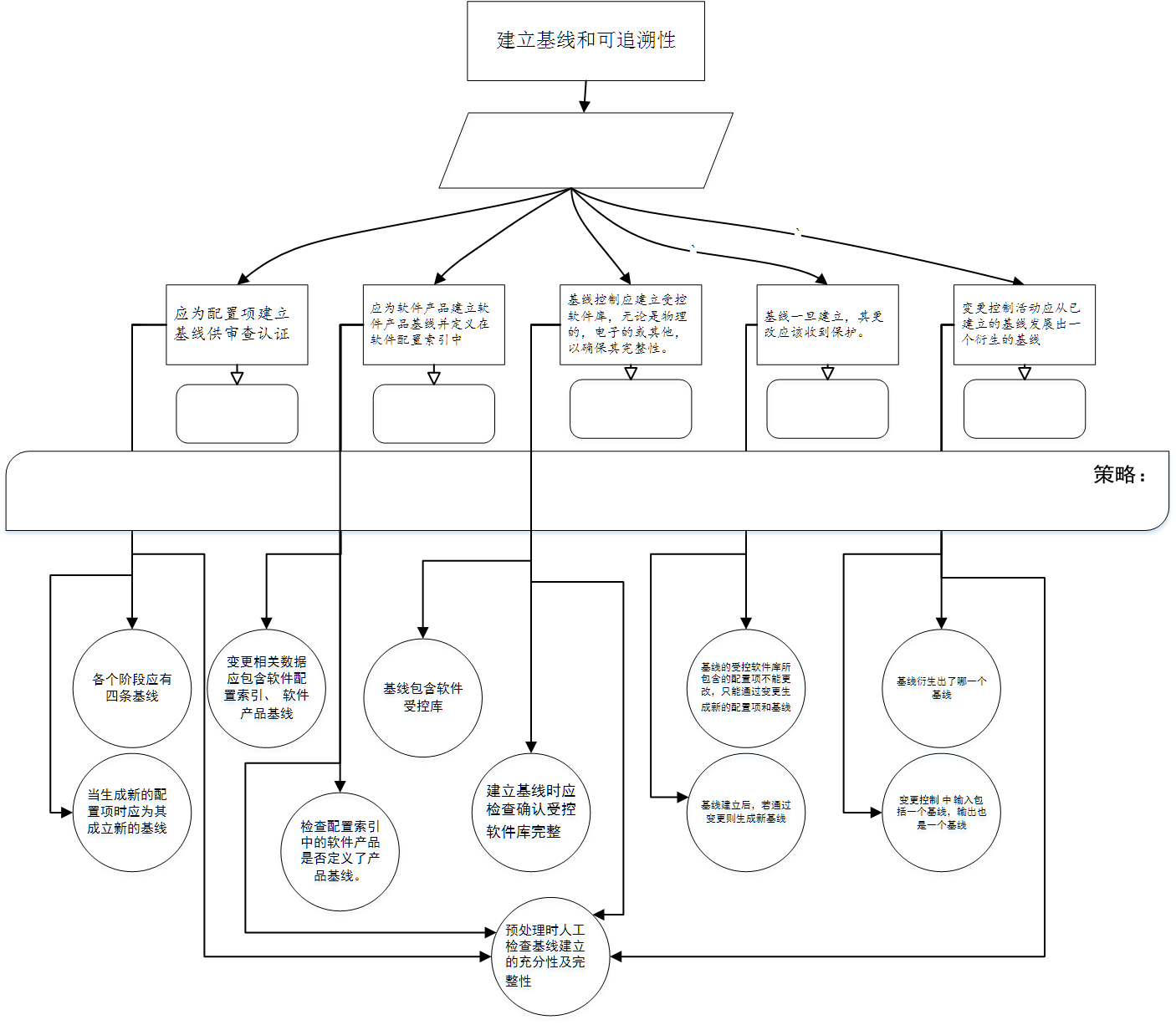
3与变更管理的流程要求相关的证据要求。

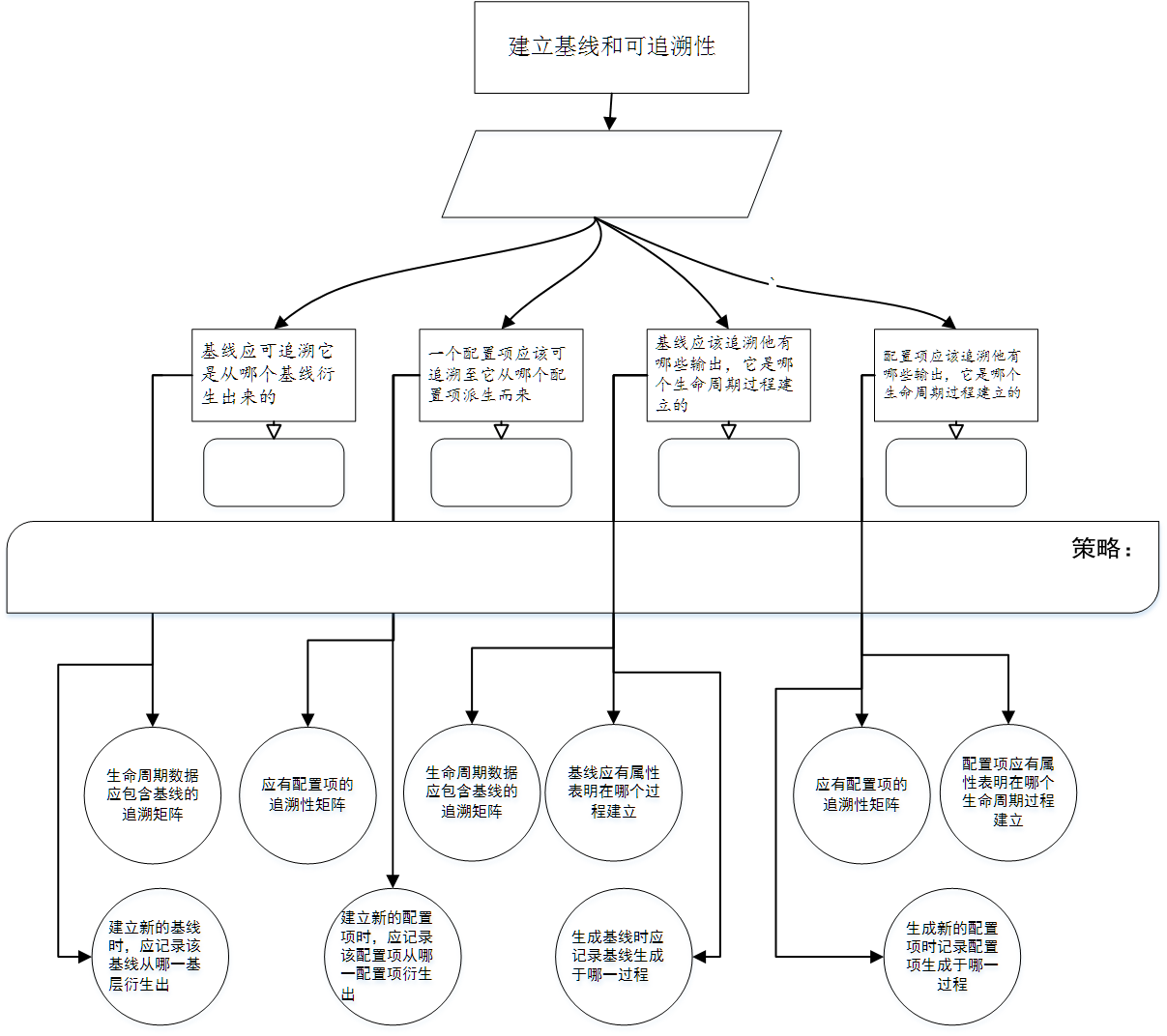
#### 3.2.1 对目标 A-8.1使用GSN方法进行分析，

具体如图（）

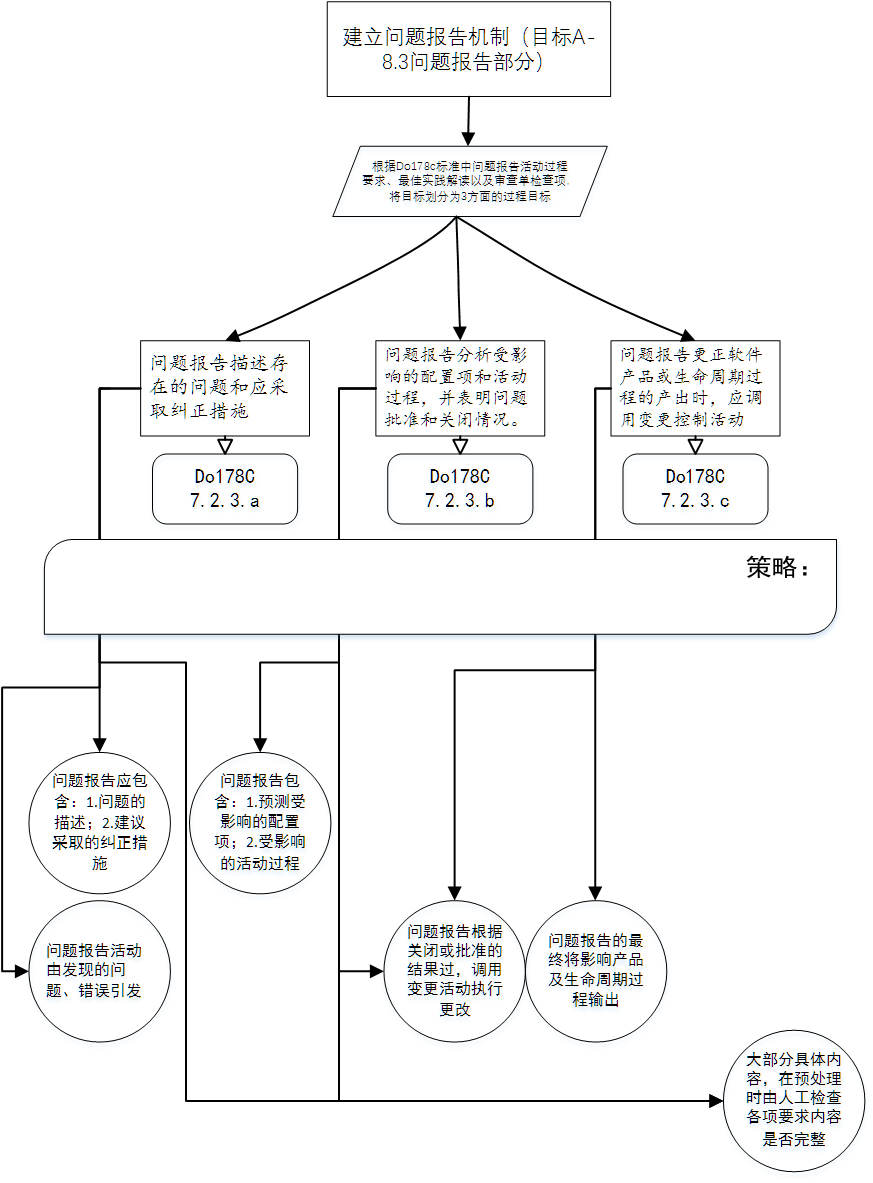


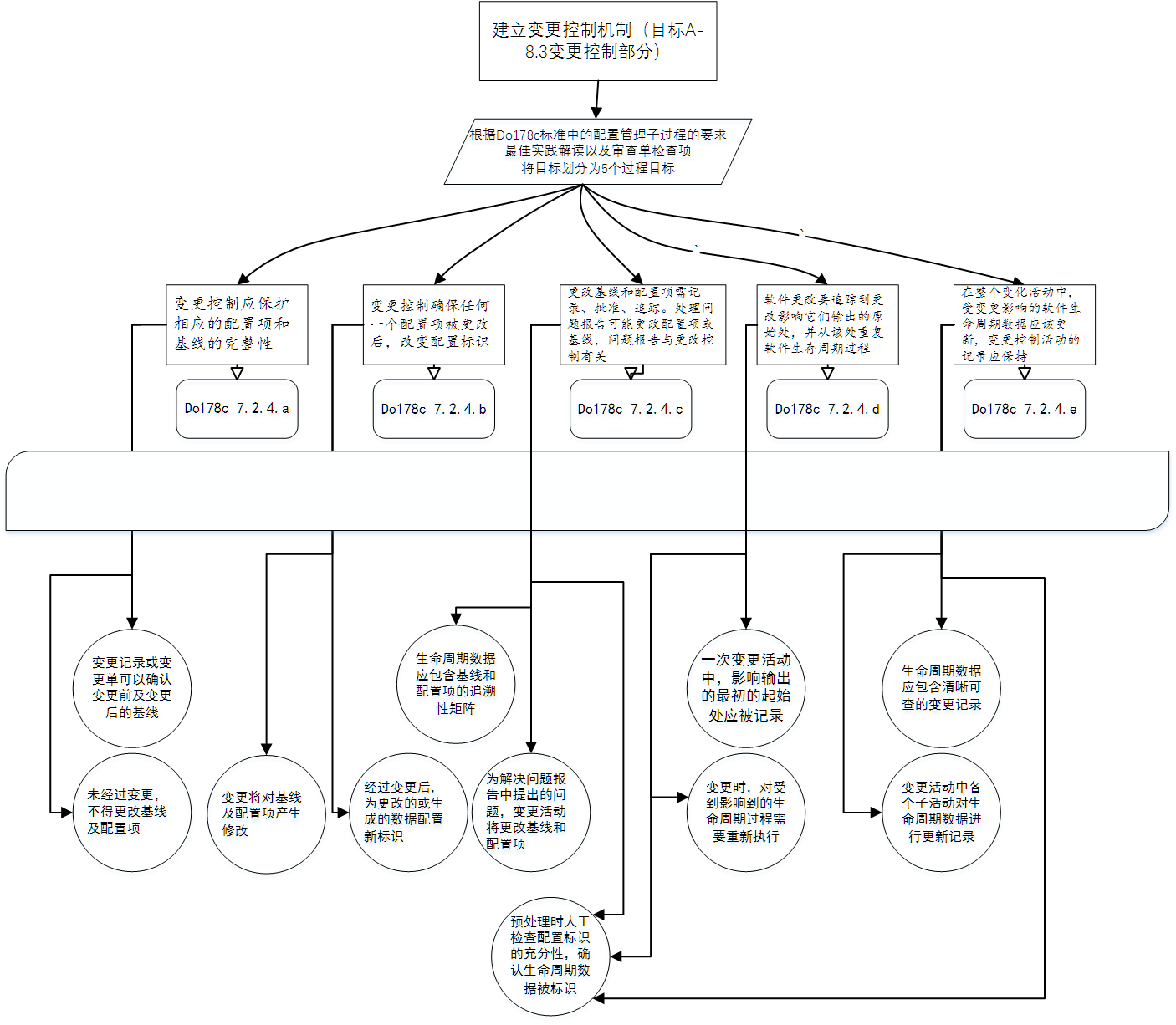
#### 3.2.2 对目标 A-8.2使用GSN方法进行分析，

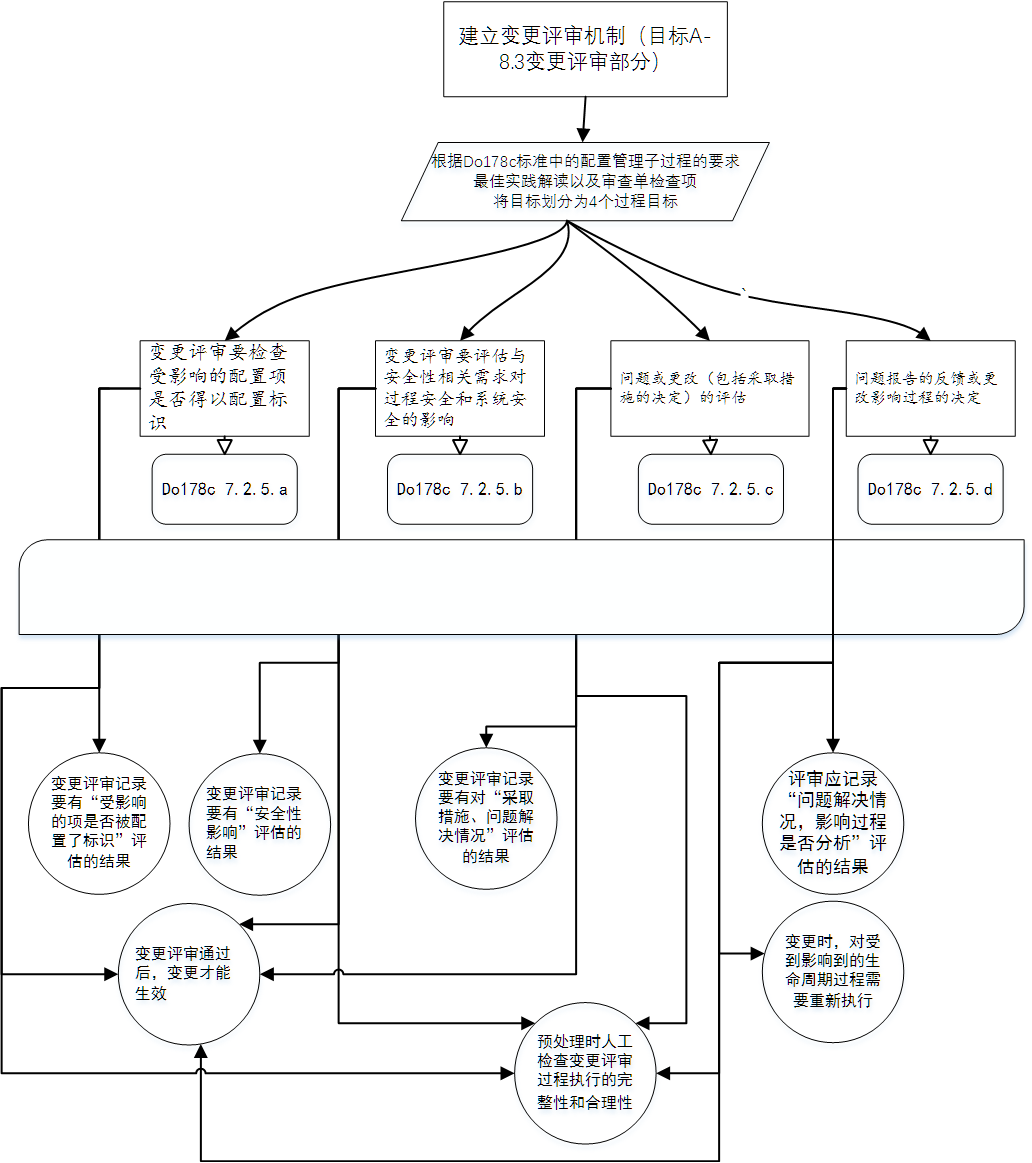




#### 3.2.3 对目标 A-8.3使用GSN方法进行分析，







#### 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.变更管理流程审定相关的关键数据。

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.配置管理纪实发布的信息等

具体数据项及数据元分类及需求见表（）。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 数据项 | 数据元内容 | 数据分类 |
| 1 | 软件配置管理记录 | 1.软件配置记录的标识;  2.条目化的变更记录，每此变更记录包含以下信息：a.变更申请时间;b.变更审批的时间;c.变更审批的结果（批准或驳回）;d.变更批准或驳回的时间;e.变更完成或终止时间;f.变更的最终状态;g.被变更基线的标识;h.变更完成后生成基线的标识 | 变更活动全流程相关关键数据 |
| 2 | 基 线 | 1.基线的标识;2.基线的受控配置项库;3.基线建立的时间;4.与其他基线间的衍生关系。 | 变更流程相关关键数据 |
| 3 | 问题报告 | 1.数据项标识;2.创建时间;3.问题描述  4.变更影响分析;5.受影响的配置项或数据元的标识;6.软件发生错误的起始配置项（若是需求变更则是被变更的需求的标识）;7.建议的变更和行动 ;8.问题报告提出者 |
| 4 | 变更请求 | 1.数据项标识;2.变更请求提出时间;3.变更影响分析;4.受影响的配置项或数据元标识;5.软件发生错误的起始配置项（若是需求变更则是被变更的需求的标识）6.建议采取的变更和行动;7.变更请求提出者 |
| 5 | 变更审批单 | 1.数据项标识;2.对变更的批准或拒绝意见;3.变更审批或拒绝的时间;4.负责审批的人员 |
| 6 | 变更评审单 | 1.变更评审单标识;2.评审的最终结果;3.评审生效时间;4.评审负责人员 |
| 7 | 软件配置状态纪实 | 已软件配置状态纪实报告的形式，应包括：  1.配置项标识;2.基线列表;3.配置项列表;4.问题报告列表;5.变更申请、审核、评审记录列表 |
| 10 | 高级需求 | 1.配置项标识;  2.需求数据具体类型;  3.条目化的数据内容 | 需求或需求派生  配置项数据 |
| 11 | 低级需求 |
| 13 | 源代码 |
| 14 | 测试用例 | 1.配置项标识;  2.需求派生数据具体类型;  3.条目化的数据内容 | 需求或需求派生  配置项数据 |
| 15 | 测试规程 |
| 16 | 目标代码 |
| 17 | 可执行目标代码 |
| 18 | 测试结果 |
| 19 | 测试用例与高层需求的追踪数据 | 1.配置项标识;  2.所属追溯性数据的类型;  3.条目化的数据内容 | 追溯性相关  配置项数据 |
| 20 | 底层需求与高层需求的追踪数据 |
| 23 | 源代码与底层需求的追踪数据 |
| 24 | 测试用例与底层需求的追踪数据 |
| 26 | 测试规程与测试用例的追踪数据 |
| 27 | 测试结果与测试规程的追踪数据 |
| 29 | 目标代码与源代码的追踪分析数据 |
| 30 | 其他受控生命周期数据 | 1.配置项标识;2.生成数据项时所处的生命周期过程;3.配置项间的追溯关系 | 其他配置项数据 |

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，提出统一变更流程。

根据3.1节对Do178c的分析结果，特别是对变更流程相关证据的需求作为依据，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，考虑了两方面的因素。第一，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。第二，该流程应该尽量简洁。作为审定用流程，应面向实际开发工的，应在要求其符合标准目标的基础上，不对开发流程提出与标准中目标无关的要求和活动。审定用流程过于复杂，会导致对符合do178c标准要求，但与该复杂流程不兼容的研发流程无法进行审定；第三，该流程强调的是过程而非方法，与使用何种变更管理工具无关。在开发过程中，配置管理或变更管理使用的工具种类较多，各有特点，过多的工具要求必然给开发造成更多困难且不是必要的，只要流程相符即可。

统一的变更模型的流程图如图（）所示：

一个图

按照变更子流程的特点，本文将变更流程分成了5个阶段，a.变更申请阶段 b.变更评估阶段 c.变更实施阶段 d.变更评审阶段 e.变更完成阶段 f.状态统计阶段。整个流程共9种子活动，分11步进行。

1．问题报告或需求变更报告。按照变更原因的不同，可以将变更分为两类，即问题引起的变更和需求变化引起的变更。若变更因问题而提出的，则由相关负责人展开问题报告活动，填写问题报告单；若变更时因为需求改变引起的，则由相关负责人进行需求变更的分析。

2．变更申请。变更申请人根据问题报告或需求变更报告，提出变更申请。。

3．变更审批。由审批负责人员对变更申请的内容进行审核，若审核通过，则开始执行变更；若审核不通过时，则重新申请，或中止变更，关闭申请和问题报告。

4．基线管理。根据变更审批明确的要被变更的原基线的标识，获得该基线的受控配置项的标识，该基线状态根据配置管理计划是否需要被checkout，取决于是否允许同时对同一基线开始两次以上的变更，一般来说应是不允许的。

5．配置项：通过基线受控配置项库中配置项的标识，获得配置项数据。并将此配置项checkout。

6．数据开发过程。由开发人员执行数据的更改。需要说明的是，根据最佳实践关于建立阶段基线的要求文献（），在开发阶段，完成一个子阶段后，需要建立核查基线、评审基线以及审查基线，因此开发活动过程中也会建立基线，但这些基线是在处理人角色发生变化时建立的，不是由变更建立，与变更生成的基线不同，此处不对其生成过程进行具体说明。

7.配置标识。为新生成的数据配置标识，包括开发过程生成的阶段基线和基线的受控数据项。对于阶段基线，虽与变更生成的基线不同，但由于可能作为未来变更活动的初始基线，因此要将这些基线及基线中新生成的数据项作为基线类型的数据。因此实际在第6步数据开发过程中，也需进行配置标识和基线管理活动。

8.变更评审。再由变更委员会进行变更评审，若评审通过，则建立新的基线，若不通过，则返回第6步数据开发过程重新执行。

9.基线管理：变更评审通过后，确定了基线的受控数据项库，建立新的基线。

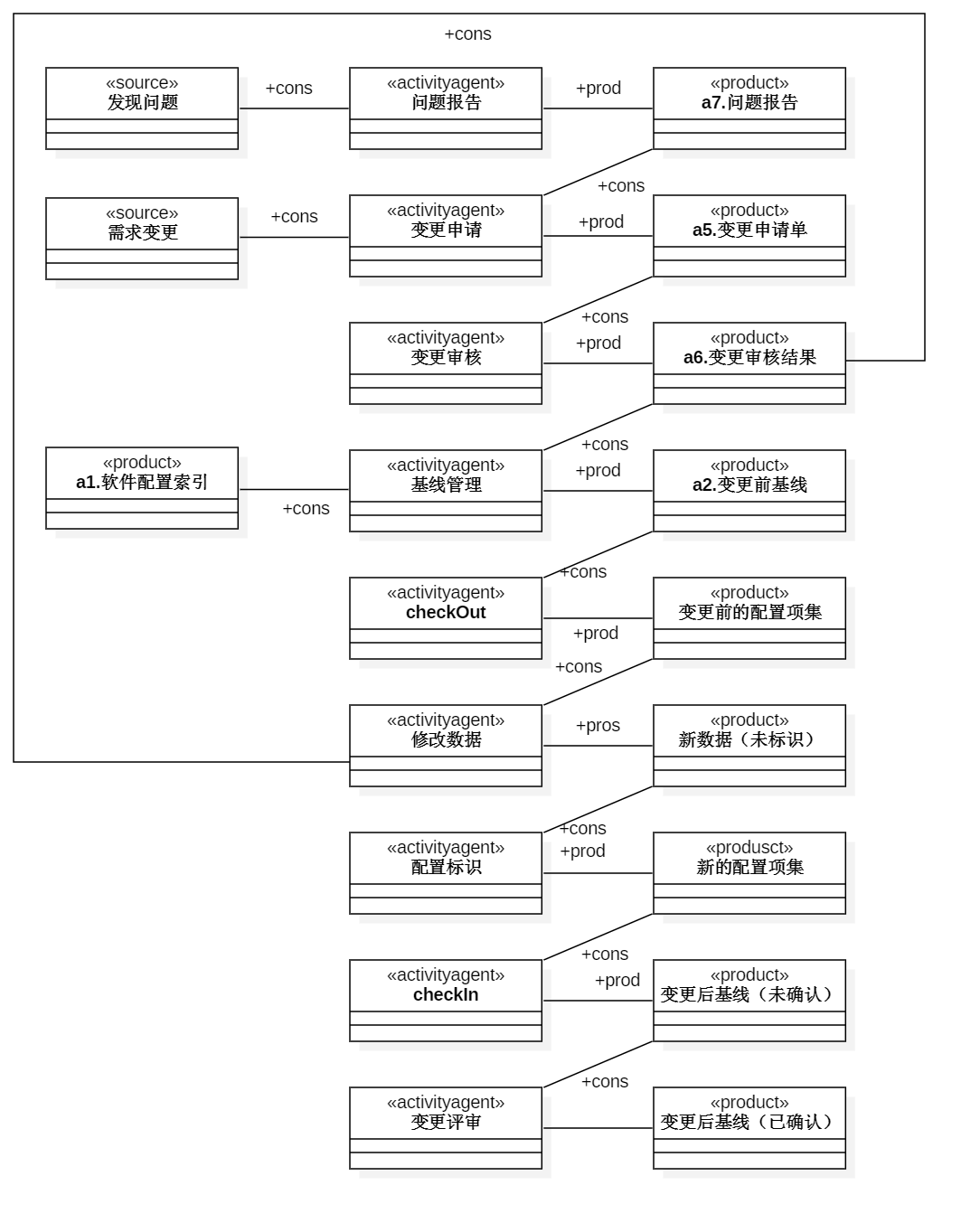
10.配置标识：为建立的新基线配置标识。同时基线已建立成功并配置了标识，将第4、5步检出(checkout)的基线和配置项进行检入(checkIn)。

11.配置状态纪实统计：将新基线加入基线列表，新配置项加入配置项列表，问题报告加入问题报告列表，变更申请、审核、评审记录加入变更报告列表。

### 3.5 生命周期数据和变更流程间的输入输出关系

在章节3.3、3.4中，我们确定了变更管理中生命周期数据项、数据元，提出了统一的适用于审定的统一的变更管理模型。适航审定的基本思想，是通过对研发阶段的生命周期数据进行审查，判断研发过程是否满足了Do178C标准的要求，因此，本节确定变更流程中生命周期数据与变更子活动间的输入输出关系。

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。Cons代表consume，表明活动的输入，prod代表produce，表明活动输出。中间一列为活动代理类，代表了活动的子流程，source类 引起变更的原因，product代表了生命周期数据，子活动流程自上而下依次执行。



数据与流程关系图

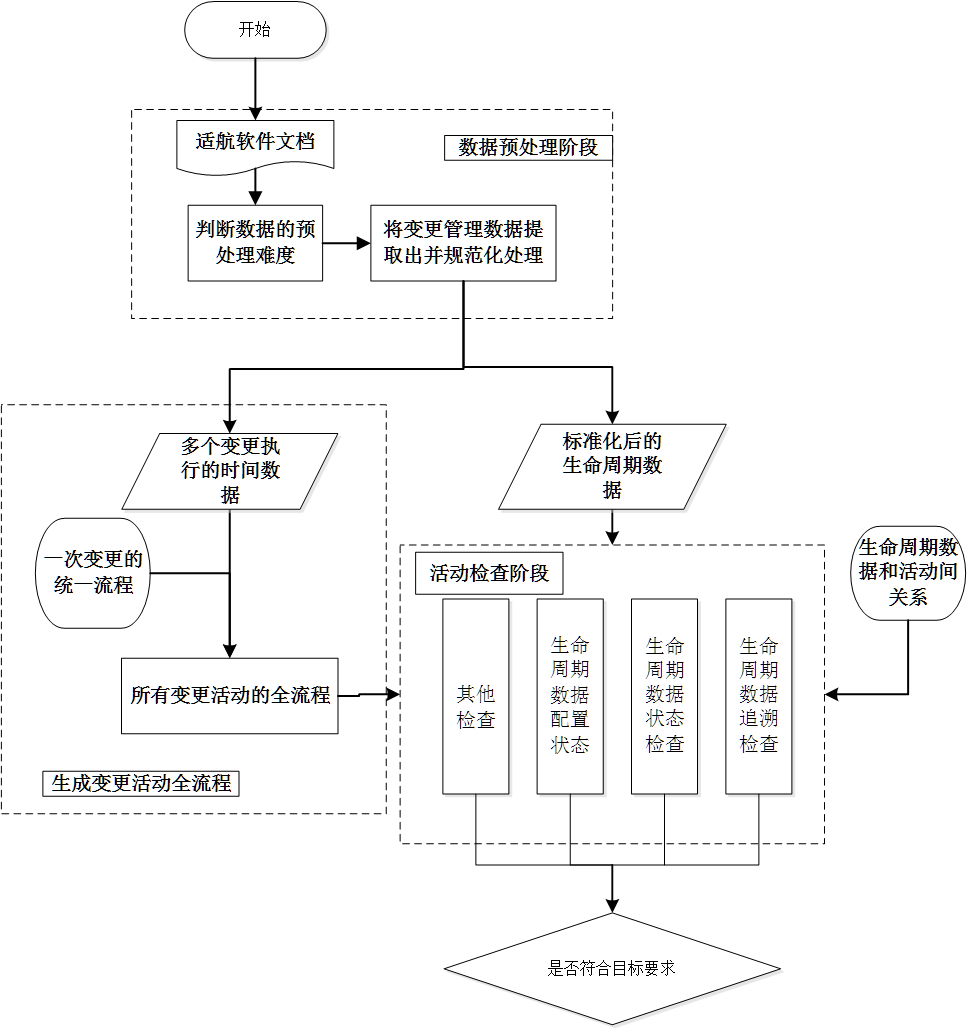
# 第四章 适航领域变更管理审定模型

本章以第三章的研究成果为基础，提出了适航领域变更管理审定模型。首先介绍了变更管理审定模型的整体框架SSSR，然后对使用审定模型三大部分及其应考虑的问题和具体实施方法步骤进行了详细描述。

## 4.1 审定模型

本文的第三章研究了变更管理生命周期数据项数据元要求、适用于审定的统一的变更管理流程以及生命周期数据与变更流程子活动间的输入输出关系，我们以此为基础，提出了适航领域变更管理审定模型。按照审定将审定分为了几个部分。分别是人工审查，流程审查和变更内容审查。人工审查是对需要人工参与进行审查部分的界定与说明；流程审查部分通过将生命周期数据代入变更活动全流程，按照统一变更流程要求、生命周期数据格式要求、输入输出关系要求，对流程相关的数据符合性要求进行审查；变更内容审查则是通过重建需求相关数据项追溯链接，对变更影响分析的准确性和变更实施时正确定进行审查。

审定模型的框架如图（）所示。



适航领域变更管理审查模型

## 人工审查

人工审查部分包含预处理时审查与监督审查两部分。

在预处理时，审查人员根据研发方提供的生命周期数据形式、精细程度、处理难度等，对生命周期数据进行初步判断，确定此生命周期数据的基本格式和要求是否符合适航要求，若符合，则对该原始生命周期数据进行预处理，得到标准化后的数据，同时也对预处理时需考虑部分进行人工审定。该部分审查的内容包括各项生命周期数据的充分性、完整性等资料预处理之后无法进行审定的内容。

另外，由于适航审定是一个极为复杂的工作，对于能够由机器审查的部分，机器审查的同时也需要人工参与进行监督审查。

## 流程审查

### 4.3.1 变更活动全流程

在上一章中，我们确定了统一的变更流程，将单次变更中的生命周期数据代入到统一的流程中，即可重现此次变更的具体流程。通过重现每次的变更流程，我们可以对变更过程中的先后关系、协作关系、数据的传递关系及资源的共享关系等是否符合要求进行验证。但如果仅进行此项验证，则忽略了多个变更间存在的这些关系，因此还需构建整个变更活动的全流程。

构建全流程的基础是配置管理记录，在生命周期数据中我们已经确定配置管理记录至少应提供变更的时间信息、关键标识项的标识以及变更的最终结果。因此，各次变更相应的问题报告、变更申请和变更审核结果、变更前及变更后的基线标识等可以由配置管理记录取得。Do178c并没有明确规定配置管理记录的具体内容及形式，在制定配置管理计划时，可以自行确定易于记录和使用的管理方式。

基于这些信息，我们可以将每个变更按照变更执行的顺序，放置于统一的时间线上，从而对所有的变更活动按照研发工作中变更的时间节点排序，得到了变更活动的全流程。代入相应的变更相关生命周期数据，多个变更的相互关系即可通过流程重现得到验证。

### 4.3.2 变更子活动审查

由上一节的方法，我们已经得到了变更活动的全流程。本章介绍的是通过将生命周期数据代入到各个子流程中，模拟研发过程对变更流程进行重现，对流程相关的生命周期数据是否符合DO178c的相关目标进行审查的方法。

在重现变更流程时，我们以开发时各项步骤的具体活动为依据，规定相应步骤的子活动在重现过程时的活动模拟和数据验证。

活动模拟的任务是模拟研制过程，并据此改变数据项相应状态，为数据验证提供依据。我们规定，审查时为数据项设置相应的状态属性，通过检查状态属性应能够确定相应问题报告或变更申请单在此时间点上是否已提出，变更审核单是否已审核完毕，各数据项是否已生成、已标识或已检出。

数据检查验证通过对相应数据项的状态属性进行检查，判定此过程中的生命周期数据是否存在矛盾、冲突、缺失等情况。例如，标识为C1变更的批准生成了配置项n1 ,变更完成时，会将配置项n2状态改变为已生成、已标识，而标识为C2的变更，对基线 B1 进行变更，基线B1的被控库中包括配置项n1，此时变更C2会检查配置项n1是否已生成、已标识。若变更C1的完成时间早于变更C2的批准时间，则此项检查无问题；若变更C1的完成时间晚于变更C2的批准时间，变更C2检查n1的状态时为未生成、未标识，无法使用该配置项，表明通过此生命周期数据重现的流程是有问题的，也就是说研发过程中的变更管理流程存在问题。审查流程时各步骤应对生命周期数据进行的活动模拟和数据验证见表（）。

*子活动过程开始*

*1.首先对变更所处的状态进行检查*

*2.然后（生命周期数据的类型由统一的变更流程来确定，具体哪一条记录则根据变更管理记录确定）确定了哪些生命周期数据是要检入该过程的，通过配置管理记录从大量的生命周期数据库中获得此子活动应输入的生命周期数据（获得的同时也对该数据的存在性进行检查）；*

*3.然后对数据项的状态进行检查；*

*活动实施*

*4.接着根据活动的任务，对输入的生命周期数据的状态进行相应的修改，对于会生成新的配置项或基线的开发活动，生成的新的配置项通过读取变更管理记录得到，修改的同时也对此项子活动生成的数据的存在性进行检查，并确定该活动生成的新的生命周期数据的状态。*

*5.根据子活动的类型和活动的结果重新修改变更所处的状态，并确定进入哪个下一子活动流程*

*进入下一子活动，从第1步重新开始。*

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：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 子活动名称 | 子活动的阶段 | 检查 | 状态操作 |
| 1 | 问题报告 | 申请阶段 | 1. 该变更应是变更的起始，无上一子活动 2. 2.输入的数据项应为问题或需求变更； 3. 输出的数据项应为问题报告；   3.问题报告的要素（上面某章节已确定）应齐全4.该问题报告状态应为尚未被提出；  5.问题报告提出者要有相应权限 6.问题报告的提出时间应与时间顺序相符 | 问题报告状态改变为已提出 |
| 2 | 变更申请 | 申请阶段 | 1.上一子活动应该是问题报告或变更审批未通过 2.输入的数据项应为问题报告（或审批未通过意见）； 输出的数据项应为变更申请单； 3.该变更申请单的要素（上面某章节已确定）应齐全 4.该变更申请单状态应为尚未被提出； 5.变更申请的提出者要有相应权限  6.变更申请的提交时间应与时间顺序相符 | 变更申请单状态改变为已提出 |
| 3 | 变更审批 | 评估阶段 | 1.该变更的上一子活动应该是变更申请 2.输入的数据项应为变更申请单  输出的数据项应为变更审批单（或变更审批结果） 3.变更审批单要素（上面某章节已确定）应齐全  4.该变更审批单状态为审批未完成 6.有明确的变更审批结果 7.变更审批人员要有相应权限。  5.变更审批时间应在时间顺序相符 | 变更审批单状态改变为审批完成 记录该变更审批结果（通过或不通过） |
| 4 | 基线管理 | 执行阶段 | 1.上一子活动应该是变更审批且审批通过，或变更评审且评审未通过;2.输入的数据项应为变更审批通过的意见和变更审批确定的准备变更基线的标识;3.输出的数据项应为准备变更的基线配置项；4.该基线的要素（上面某章节已确定）应齐全 5.该基线状态应为已建立状态；6.该基线应未处于检出状态 | 准备变更基线的状态改变为检出 |
| 5 | 配置项检出（checkout) | 执行阶段 | 1.上一子活动应该是基线管理;2.输入的数据项应为准备变更的基线受控数据项的标识;3.输出的数据项应为准备变更的基线的受控配置项的全集；4.该基线受控库的所有配置项的要素（上面某章节已确定）应齐全 5.受控配置项的状态应未处于检出状态 | 从基线受控库得到的受控配置项状态改变为检出 |
| 6 | 开发过程，执行数据的更改 | 执行阶段 | 1.该变更的上一子活动应该是配置项检出;2.输入的数据项应为准备变更的基线受控数据项的全集 3.输出的数据项为变更后的基线生成的数据项全集，同时还有此过程开发中建立的阶段基线及其生成的数据项全集 4.基线的状态应为尚未建立，新生成数据项的状态为未生成,未标识， 5.执行变更的开发人员符合权限要求 | 变更后的基线、阶段基线的新生成数据项的状态改变为已生成，未标识 |
| 7 | 配置标识 | 执行阶段 | 1.上一子活动应该是开发过程;2.输入的数据项应为开发中建立的阶段基线及其新生成数据项的全集，以及变更后基线新生成数据项的全集（均未标识） 3.输出的数据项为已经标识过的输入数据项 4.这些数据项状态应为尚未被标识 5.执行变更的开发人员符合权限要求 | 阶段基线及其新生成数据项的全集，以及变更后基线新生成数据项的全集状态改变为已标识 |
| 8 | 变更评审 | 评审阶段 | 1.上一子活动应该是配置标识;2.输入的数据项应为开发后基线的受控配置项内容 3.输出的数据项为变更评审单 3.该变更评审单的要素（上面某章节已确定）应齐全 4.变更评审单的状态应为尚未完成； 5.变更评审单的提出者要有相应权限  6.变更评审单提交时间应与时间顺序相符 | 1.变更评审单的状态改变为已完成； 2.记录该变更评审的意见（通过或不通过） |
| 9 | 基线管理 | 完成阶段 | 1.该变更的上一子活动应该变更评审且评审通过;2.输入的数据项应为开发后建立的新基线的受控配置项的全集 3.输出的数据项为开发后建立的新基线 3.该基线的要素（上面某章节已确定）应齐全 4.该基线的状态应为尚未建立 ，未标识；5.建立基线的人员要有相应权限 | 变更后的基线状态改变为已建立，未标识 |
| 10 | 配置标识 | 完成阶段 | 1.上一子活动应该为基线管理;2.输入的数据项应为开发后建立的新基线（为标识） 3.输出的数据项为开发后建立的新基线（已标识） 3.该基线的要素（上面某章节已确定）应齐全 4.该基线的状态应为已建立 ，未标识；5.配置标识的人员要有相应权限 6.基线标识配置时间应与时间顺序相符 | 变更后的基线状态改变为已建立，已标识 |
| 11 | 配置状态纪实 | 记录阶段 | 1.上一子活动应该为配置标识;2.输入的数据项应为变更过程中生成的配置项。基线、问题报告、变更申请、审核、评审记录 3.输出的数据项为配置状态纪实报告 4.配置状态纪实报告要素应齐全 5.本次变更配置状态纪实状态为未完成 5.配置状态纪实的人员要有相应权限 6.配置状态纪实报告的内容应与配置项库中数据项生成状态相符 | 本次变更配置状态纪实的状态改变为已完成 |

## 内容审查

在一次变更中，除了流程相关证据要求外，变更影响分析、变更新生成的配置项是否符合追溯性要求也是需进行审定的重要方面。对此要求的验证需要对相关配置项的具体内容，即3.4节中需求或需求派生配置项数据与追溯性相关配置项数据进行分析。

### 4.4.1验证方法

内容审查主要是对变更影响分析和变更执行新生成的数据项追溯性进行验证审查，对内容进行该项检查的相关生命周期数据为 3.4节中需求或需求派生配置项数据与追溯性相关配置项数据。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，通过文本相似度计算的方法计算需求或需求派生配置项之间的相似度，并进行选择排序，确定各个层次需求间的追溯关系，将确定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之间的相似度，并进行选择排序，以此为依据判断生成的配置项是否满足追溯性关系，来确定变更执行是否正确。

通过图+说明 表述清楚

下节介绍使用文本相似度加选择排序重建追溯矩阵的方法。

### 4.2 需求追溯矩阵建立方法

获得生命周期数据项之间的追溯性的流程如图（）所示，包括3个阶段：

流程图

（1）数据预处理阶段。包括对数据项文本的预处理和词向量的训练。具体在第五章实验部分进行说明。

（2）文本相似度计算阶段。预处理完成之后，将待建立跟踪关系的追溯相关数据项文本映射为词向量，，然后把每个数据项文本看作信息检索中的一个查询语句。计算每一个查询语句与其他数据项的文本相似度，根据相似度排序，得到了该查询的排序列表

（3）最后，将排序列表引入学习排序算法模型中，利用机器学习分类算法将排序后的列表进行二次处理，最终得到相关跟踪链接。

接下来主要介绍文本语义相似度计算和学习排序模型中的改进部分。

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。