绪论素材：

目前有多种论证机载软件满足DO-178C标准的方法，使用比较广泛的是基于证据的目标满足性论证，即针对DO-178C标准中规定的目标和相关要求，构建相应的目标满足性论证模型，并通过对软件开发过程及其制品数据的分析，提取论证模型中所需证据，并据此来论证软件开发过程及其最终产品是否满足标准中的各项目标，即符合标准中规定的各项目标和相关目标。这种基于证据的论证方法不仅适用于标准满足性的论证，同样也适用于软件安全性论证。

3. 针对不同项目建立基于证据的论证模型时，应当避免重新阅读DO-178C标准和重新建立论证模型。但是，目前尚缺少一种有效的方案，将DO-178C中项目无关的论证结构提取出来，在建立不同项目的论证模型时，有效利用项目无关的论证结构，提高论证结构的复用性。

### 安全案例

验证系统及其开发过程是否符合相应的安全标准要求评估人员验证一系列安全目标是否得到满足[[[1]](#endnote-1)]。目前，基于证据的安全论证过程是一种被广泛采用的论证过程。为了能够系统化地构建基于证据的安全论证过程，Tim Kelly提出了使用安全案例（Safety Case）来描述证据到安全目标的推理过程。一个安全案例包括三个基本要素：安全目标，安全证据，安全论证结构。其关系是：评估人员使用安全证据对安全目标进行论证[[[2]](#endnote-2)]。安全案例的结构关系如图2所示。

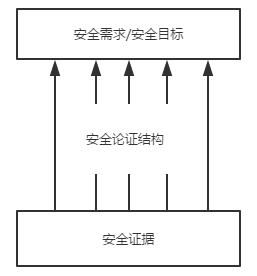


图 2 安全案例

安全目标（Safety Objective）描述了系统及其开发过程中应当满足的安全相关要求，也可以叫做安全需求。安全证据（Safety Evidence）是一类制品或者数据信息，这类信息能够为论证系统是否满足安全目标提供证据和支撑[34]。

### 2.2.2 目标结构化表示法GSN（Goal Structuring Notation）

GSN[8]由英国约克大学的Tim Kelly教授首次提出。GSN可以层次化的描述安全目标的论证结构[8]。GSN标准[24]中定义了GSN图形符号及其含义。表2给出了GSN基本的节点类图符（包括辅助论证符号和主论证符号），并解释了其含义。表3给出了GSN的基本连接图符，并解释了其含义。

表 2 GSN图形符号含义

|  |  |
| --- | --- |
| Goal | **目标**  用矩形表示，代表了安全论证中的目标（也可以是子目标） |
| Strategy | **策略**  用平行四边形代表，描述了目标到其分解的子目标的推理的本质，或者推理的规则 |
|  | **解决方案**  用圆形来表示，代表了某个证据或者某几个证据项的引用 |
| Context | **上下文**  用圆角矩形来表示，代表了上下文。这个可以作为一个引用来表示上下文相关的信息，也可以是一个论述 |
| J | **说明**  在椭圆形下方有一个大写字母J表示论证的某个目标对象是可信的 | |
| A | **假设**  在椭圆形下方有一个大写字母A表示一个可能的外部条件，这很有可能作为说明策略的合理性的依据 | |

**表2 GSN图形符号含义（续）**

|  |  |
| --- | --- |
|  | **未开发的实体**  表示该论证过程尚未开发出来，这个可以应用与目标，也可以应用于策略 |
|  | **尚未开发的目标**  代表了一个目标在开发过程中尚未开发出来 |

表 3 GSN连接符号含义

|  |  |
| --- | --- |
|  | **推理**  实心箭头表示一种推理关系，用来表示目标，解决方案，策略之间的关系 |
|  | **支撑论证**  空心箭头表示一种上下文附属关系 |

使用GSN建立安全案例时，使用上面的符号就可以完全实现需要表达的信息。其中，目标、策略、解决方案是主论证符号。上下文、说明、假设是辅助论证符号，未开发的实体的使用方法如表2的“未开发实体”表示符号所示。此外，实心箭头连接主论证符号，空心箭头表达辅助论证符号和主论证符号之间的关系。

### 安全案例

验证系统及其开发过程是否符合相应的安全标准要求评估人员验证一系列安全目标是否得到满足[[[3]](#endnote-3)]。目前，基于证据的安全论证过程是一种被广泛采用的论证过程。为了能够系统化地构建基于证据的安全论证过程，Tim Kelly提出了使用安全案例（Safety Case）来描述证据到安全目标的推理过程。一个安全案例包括三个基本要素：安全目标，安全证据，安全论证结构。其关系是：评估人员使用安全证据对安全目标进行论证[[[4]](#endnote-4)]。安全案例的结构关系如图2所示。

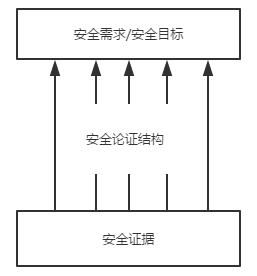


图 2 安全案例

安全目标（Safety Objective）描述了系统及其开发过程中应当满足的安全相关要求，也可以叫做安全需求。安全证据（Safety Evidence）是一类制品或者数据信息，这类信息能够为论证系统是否满足安全目标提供证据和支撑[34]。

第三章 **基于DO-178C的目标的研究 建立符合适航标准的变更管理uML**

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。

**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。

## 目标满足性论证模式的建模方法

目标满足性论证模式的建模流程如图5所示。

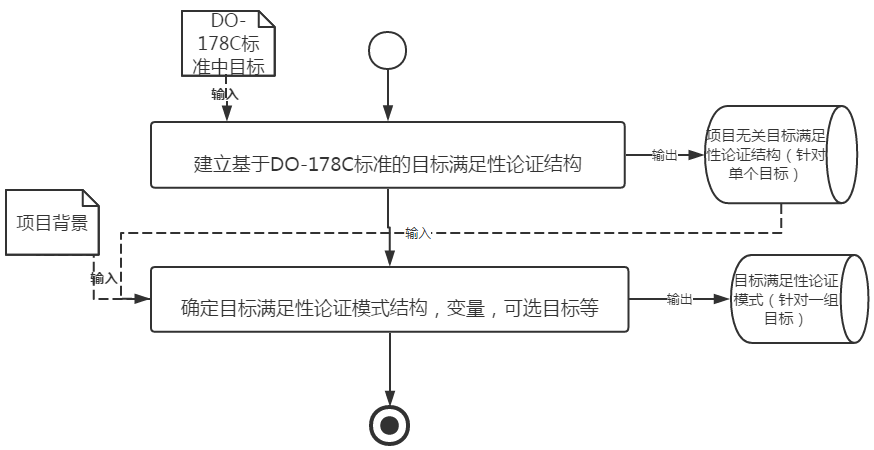


图 5 目标满足性论证模式建模流程（此处图形预计要做成整个第三章的思路）

第一步，基于对DO-178C标准中规定的各项目标的分析和理解，对DO-178C标准中配置管理相关目标进行项目无关的目标满足性论证。

针对每个目标，依据标准中对该目标的定义和描述，以及对需要执行的活动及其输入和输出的说明。以及最佳实践、审查指南等权威的依据，确定建立论证结构的基本步骤如下所述：

1. 确定目标及其支持活动：针对需要论证的目标，确定该论证对应DO-178C标准中需要执行的活动。

2. 基于活动确定子目标：针对目标中定义需要执行的活动，依据活动的具体内容，及该活动与其余活动间的关联关系，确定目标论证结构中的子目标和分解策略，同时添加关联标准章节等辅助论证符号支撑相应子目标的论证。

3. 基于目标输入及相关要求确定必需的证据：参考目标中规定的输出，及其相关联的数据，依据标准的描述，确定哪些输出为目标满足性论证结构的证据。**（此处输出应为生命周期数据）**同时依据标准中对输出数据的描述，结合需要论证的子目标，定义证据的约束条件，同时确定证据之间的关联。有的子目标不能直接使用目标规定的输出进行论证，这时依据目标的实际情况确定其需要的证据数据，参考标准中对该数据的描述，以及需要论证的子目标对证据的结构要求，确定证据的约束条件。

第二步，根据第一步的研究结果。确定统一的变更管理流程。统一的变更管理流程包含流程和生命周期数据两方面。

第三步，借鉴安全UML的内容，建立符合安全UML的变更管理模型。

1. 使用GSN方法对目标进行分析

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。

标识构型项

1.应为软件生命周期数据建立标识。

2.要为每个可单独分开的配置项建立标识，也要对配置项组合而成的软件产品建立标识。

3.配置项应该在进行变更控制和追溯数据之前就被标识。

4.配置项应该在被其他生命周期过程、其他生命周期数据、其他软件制品或软件使用过引用前就配置标识。

5. 软件产品认证不能通过物理检验(如零件号牌检验)确定，则可执行目标代码应包含可由机载系统或设备的其他部分访问的配置标识。这可能适用于现场加载的软件。

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，2.转化为模型可验证的小目标。

策略2：为了构建符合Do178c标准的变更管理模型，将二层目标考虑分为 类：1在数据初始化时由人工确认的目标。2由单次变更管理的模型构架能够验证的目标。由多个变更整体利用变更模型进行全过程 来验证的目标。3.由生命周期数据间的 4.需要由生命周期数据的具体内容或有其他生命周期过程验证的目标。

如图 N.

1，要建立。2分的，总的 3.变更和追溯前。4.使用前。5 软件产品的代码的标识。

！！：配置管理模型的每个数据都必须有标识，且整个流程无错误。 2.在对生命周期数据进行格式化的过程中对此进行审查。3。对于未能建立标识的，只能主要由人工进行审查，对于已建立标识的，通过变更管理的流程来确定是否有问题。

定义标识确立的时间（如果可能）， 配置项属于哪个生命周期过程，标识是否已被建立，标识是否重复。

根据分析结果，为了进行建模，事实上要确定三个方面的内容，一是审定用模型的变更流程，二是关键的生命周期数据及其属性，三是判断符合目标的其他证据。

此三项对应着三种底层符合性目标验证的方法。根据分析，可以将“证据”分为几类。1.难以自动识别或在只涉及配置管理的模型中难以识别，由人工识别。2.通过模型本身的逻辑来确保该证据符合的3.需要在模型以外提供其他简单验证机制来确定的。3.1 变更了的配置项之间的文本相似性是否符合要求。

1. 流程

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，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，主要考虑一下几点：

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。

变更管理流程图

变更处理工作流程图

问题报告:

变更申请：

变更评审： 通过 或 不通过

变更实施, check out check in

变更核查

确认 生命周期数据

生命周期数据的架构

生命周期项

生命周期的属性项（根据生命周期间的关系确定）

生命周期数据的追溯，生命周期的输出 矩阵

是否挂起 在数据中使用

是否已存在

能追溯矩阵

二、生命周期数据

|  |  |
| --- | --- |
| 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，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 |  |

每一项都应有标识。

三、其他底层目标要求

大部分需要由人工完成，或者需要由其他生命周期活动的过程来判断。

提出一种判断变更是否合理的判断方法。

基于以上的变更流程、生命周期数据及属性，以及底层目标要求，建立审定用变更管理的模型

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， 作为一项基于标准的工作，某种程度上以后的相关工作人员都应参照此项的情况，

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

对生命周期数据以及与流程间的关系进行明确。从而建立uml模型

由于是服务于审查的变更管理模型，在建模时考虑以下一些情况：将生命周期数据的状态分成了已生成和未生成。 用例图中包含了审查方，在生命周期数据被使用时进入介入式的审查。

每个生命周期数据的状态。

我是分界线，下面是素材，上面是草稿

if the software product idntification cannot be determined by physical examination(for example,part number plate examination),then the executable object code should contain configuration identifacation which can be accessed by other parts of the airborne system or equipment. this may be applicable for field-loaded software

7.2.1.c配置项应该在变更控制和可追溯性数据记录的实施之前进行配置标识

目标1 主要通过人工判定。一方面，对于配置表示的完全性的研究不是机器能完成的。二方面，在目前的情况下，现阶段的实际开发中，还做不到所有配置项的完全数据元化，只能根据实际情况由审查人员确定软件配置标识的全面性的程度。

可用的已收集的素材

。

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）

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、 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。

涉及到配置管理的数据项和数据元

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

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。

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

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。

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？

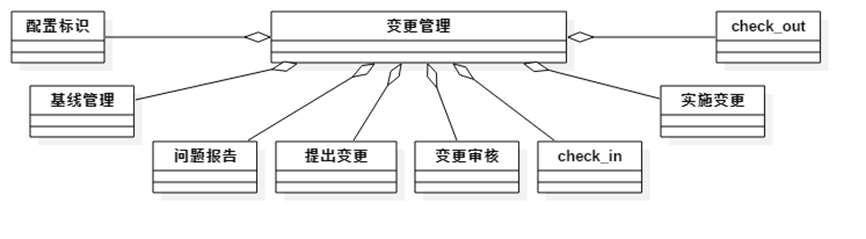


图 5 变更管理与其子活动间的聚集关系

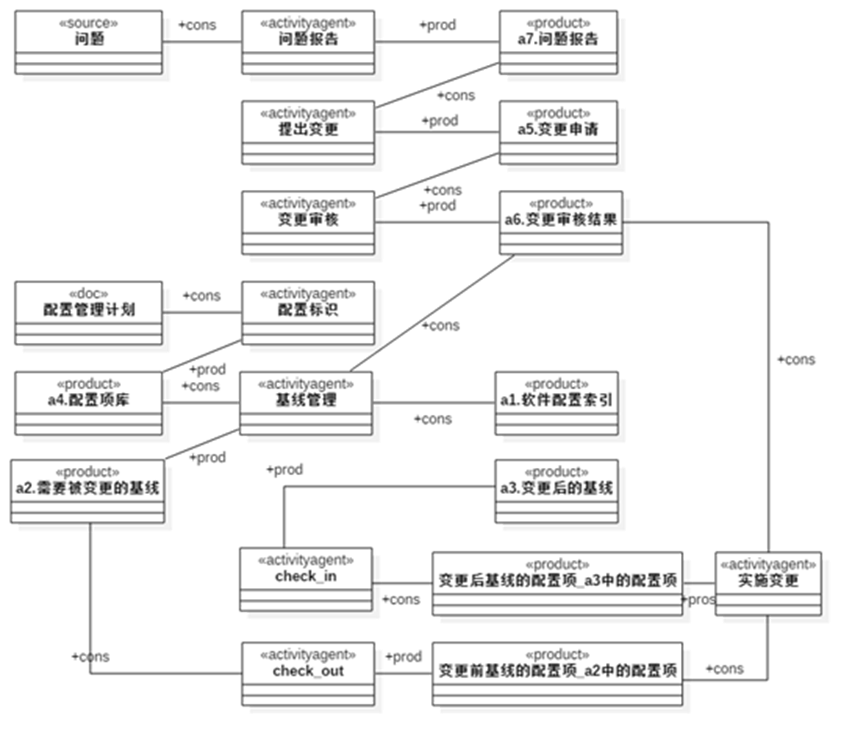


图 6 变更管理中子活动与产品间的输入输出关系

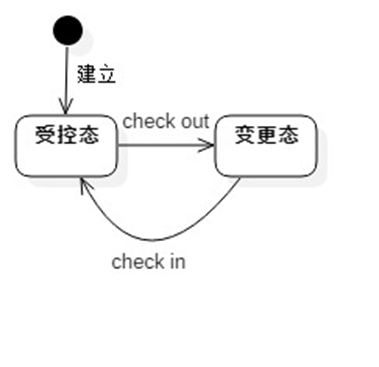


图 8 基线的状态迁移图

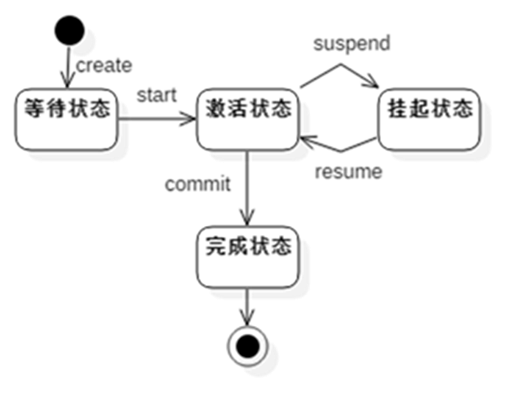
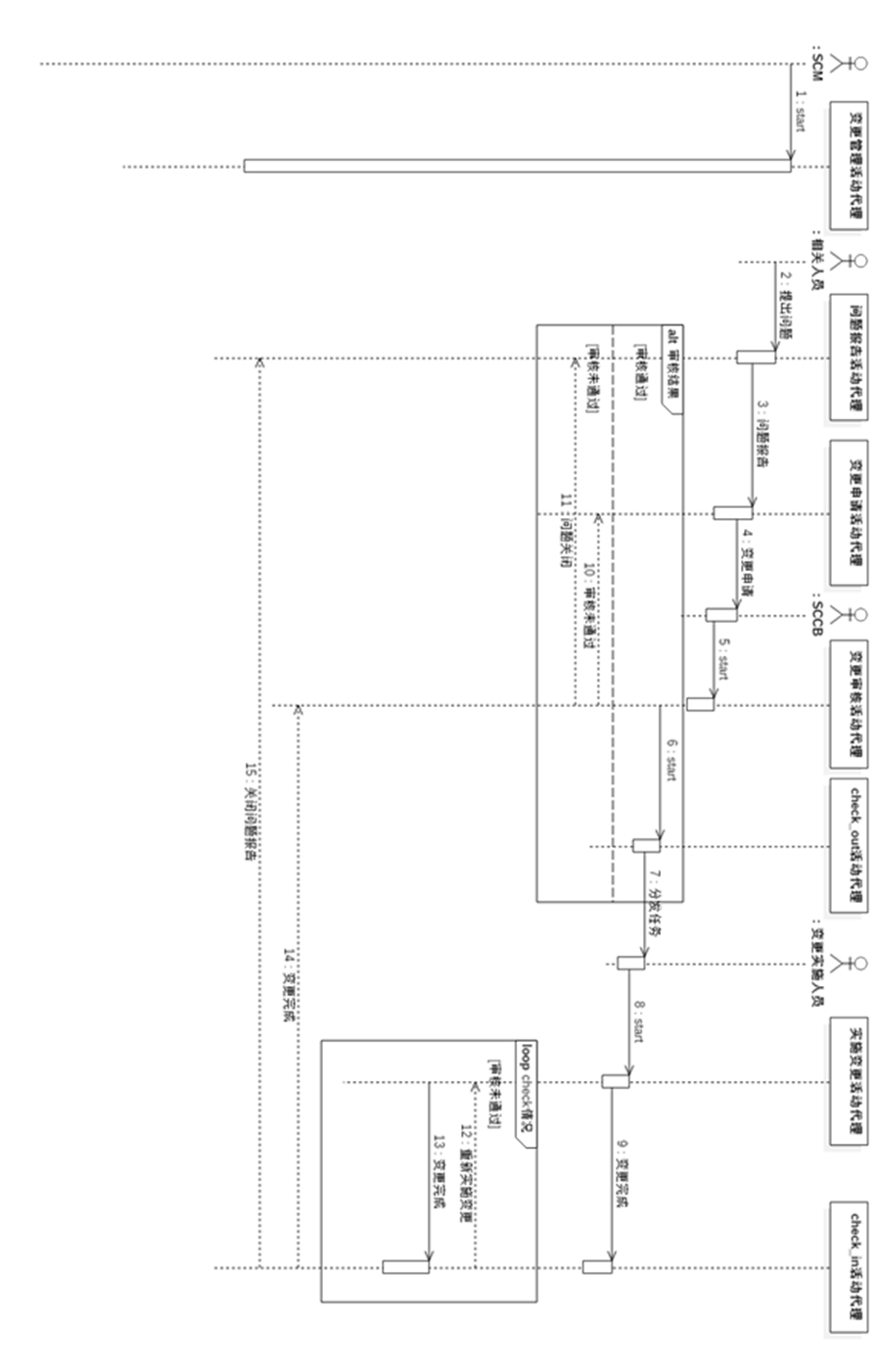


图 9 活动代理的状态迁移图



1. S. Niar, J. de la Vara, A. Melzi, G. Taglaferri, and F. Belmonte. Safety Evidence

   Traceability: Problem Analysis and Model[A]. Requirement Engineering: Foundation for Software Quality[C]. 2014: 309-324 [↑](#endnote-ref-1)
2. R. Weaver, G. Despotou, T. Kelly, and J. McDermid. Combining Software Evidence: Arguments and Assurance[A]. England: SIGSOFT Software, 2004: 152-160 [↑](#endnote-ref-2)
3. S. Niar, J. de la Vara, A. Melzi, G. Taglaferri, and F. Belmonte. Safety Evidence Traceability: Problem Analysis and Model[A]. Requirement Engineering: Foundation for Software Quality[C]. 2014: 309-324 [↑](#endnote-ref-3)
4. R. Weaver, G. Despotou, T. Kelly, and J. McDermid. Combining Software Evidence: Arguments and Assurance[A]. England: SIGSOFT Software, 2004: 152-160 [↑](#endnote-ref-4)