软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(9):4013−4037 [doi: 10.13328/j.cnki.jos.007126]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

关于安全案例论证构建的综述*

陈泽众, 邓玉欣

(上海市高可信计算重点实验室 (华东师范大学), 上海 200062)

通信作者: 邓玉欣, E-mail: yxdeng@sei.ecnu.edu.cn

摘　要: 安全案例提供清晰、全面和可靠的论据, 说明系统在特定环境下的操作满足可接受的安全性. 在受监管的安全攸关领域, 如汽车、航空和核能等领域, 认证机构通常要求系统经过严格的安全评估程序, 以确保其符合一个或多个安全标准. 在系统开发中应用安全案例是一种新兴的技术手段, 以结构化和全面的方式表达安全攸关系统的安全属性. 对安全案例的 4 个基本构建步骤: 确定目标、收集证据、构建论证和评估安全案例, 进行简要介绍. 然后聚焦于构建论证这一关键步骤, 详细介绍现有的 8 种安全案例表达形式, 包括目标结构符号 (GSN)、声明-论点-证据 (CAE)、结构化安全案例元模型 (SACM) 等, 并分析了它们的优缺点. 由于安全案例所需材料的显著复杂性, 软件工具通常被用作构建和评估安全案例的实用方法. 比较 7 种用于安全案例开发和评估的工具, 包括 astah system safety、gsn2x、NOR-STA、Socrates、ASCE、D-Case Editor 和 AdvoCATE. 此外, 还深入探讨了安全案例构建中所面临的多重挑战, 这些挑战包括数据的可靠性和完整性、复杂性和不确定性的管理、监管和标准的不一致、人因工程、技术的快速发展以及团队和跨学科合作 6 个方面. 最后, 展望安全案例的未来研究方向, 揭示其潜在应用和研究问题.

关键词: 安全案例; 系统安全; 论证构建; 目标结构符号; 安全案例工具

中图法分类号: TP311

中文引用格式: 陈泽众, 邓玉欣. 关于安全案例论证构建的综述. 软件学报, 2024, 35(9): 4013–4037. http://www.jos.org.cn/1000-9825/7126.htm

英文引用格式: Chen ZZ, Deng YX. Survey on Construction of Safety Case Arguments. Ruan Jian Xue Bao/Journal of Software, 2024, 35(9): 4013–4037 (in Chinese). http://www.jos.org.cn/1000-9825/7126.htm

Survey on Construction of Safety Case Arguments

CHEN Ze-Zhong, DENG Yu-Xin

(Shanghai Key Laboratory of Trustworthy Computing (East China Normal University), Shanghai 200062, China)

Abstract: Safety cases provide clear, comprehensive, and reliable arguments which mean that a system's operation under a specific environment meets acceptable safety levels. In safety-critical sectors subject to regulations such as automotive, aviation, and nuclear industries, certification authorities often require the system to undergo a rigorous safety assessment process and thus demonstrate that the system complies with one or more safety standards. The safety case utilization in system development is an emerging technical means to express the safety attributes of safety-critical systems in a structured and comprehensive way. This study briefly introduces the four basic steps of building a safety case, including determining the goal, gathering evidence, constructing arguments, and evaluating the case, and then focuses on the key step of constructing arguments. Meanwhile, eight existing forms of safety case expressions are introduced in detail, containing goal structuring notation (GSN), claim-argument-evidence (CAE), and structured assurance case metamodel (SACM), with their strengths and weaknesses analyzed. Given the significant complexity of the materials required for safety cases, software tools are often adopted as practical methods for constructing and evaluating safety cases. Additionally, seven tools for developing and evaluating safety cases are compared, including astah system safety, gsn2x, NOR-STA, Socrates, ASCE, D-Case Editor, and AdvoCATE.

* 基金项目: 国家自然科学基金 (61832015, 62072176)
本文由"形式化方法与应用"专题特约编辑曹钦翔副教授、宋富研究员、詹乃军研究员推荐.
收稿时间: 2023-09-10; 修改时间: 2023-10-30; 采用时间: 2023-12-13; jos 在线出版时间: 2024-01-05
CNKI 网络首发时间: 2024-03-15

Furthermore, this study delves into multiple challenges in building safety cases. These challenges include data reliability and integrity, complexity and uncertainty management, inconsistencies in regulations and standards, human factor engineering, rapid technological advancements, and challenges in team and interdisciplinary collaboration. Finally, a prospect is provided for the future development of safety cases to reveal their potential utilization and relevant research problems.

Key words: safety case; system safety; argument construction; goal structuring notation; safety case tool

1 引 言

安全案例 (safety case 或 assurance case) 是在安全攸关系统中用于证明一个系统在特定场景下的安全性和可靠性的结构化论证 (argumentation). 这些案例通常涉及系统设计、开发和维护的各个方面. 它们旨在确保一个系统满足安全和可靠性要求, 以便在实际操作中达到预期的性能. 安全案例的理论起源可以追溯到逻辑推理领域, 英国哲学家 Toulmin 在其 1958 年的著作《论证的用途》[1]中已经引入类似的思想. 之后由于复杂行业的快速发展和新型自动化技术的引入, 人类开始面临前所未有的技术风险, 这导致安全案例概念的产生[2]. 其实践应用的普及受到 1988 年 Piper Alpha 石油平台事故的影响[3].

如今, 安全案例在许多领域都有关键的应用, 尤其是在对安全性、可靠性和合规性有着高标准要求的行业, 例如以下所列的代表性应用领域.

航空航天工业[4,5]: 由于对安全性的高度要求, 航空航天工程采用安全案例来对飞机[6]、卫星[7]和飞行器[8]系统的安全性和可靠性进行验证和保证.

铁路行业[9−12]: 安全案例在此行业中被用于验证铁路系统 (如信号系统、列车控制和运行设备) 的安全性和可靠性, 从而降低事故风险并确保旅客和工作人员的安全.

汽车工业[13,14]: 随着自动驾驶[15]技术的发展, 安全案例已被用于论证自动驾驶系统的安全性和可靠性.

医疗设备[16]: 医疗设备 (例如输液泵[17], 心脏起搏器[18]等) 相关制造商利用安全案例对产品的设计、制造和使用过程进行论证, 来保证其产品的安全性和合规性.

核能工业[19−21]: 鉴于核能领域对安全性和合规性的严格要求, 安全案例被用于评估核电站、核设施和核材料管理系统的安全性.

石油化工行业[22−24]: 在石油、天然气以及化工行业中, 安全案例被用于评估和保证全过程的安全性和可靠性, 以预防重大事故, 避免环境灾难, 保护工人和环境的安全.

军事和国防领域[25]: 在对系统安全性要求极高的军事和国防领域, 安全案例被用于评估武器系统、通信系统和防御系统的安全性和可靠性.

金融和银行业[26]: 金融和银行业利用安全案例来验证金融交易系统的安全性和合规性, 保障金融数据和交易的安全.

安全管理与规范制定[27]: 在制定安全管理与规范, 例如网络安全监管[27]、校园防灾[28]和疫情防控[29]政策等, 安全案例起到评估风险、设计并确认控制措施、提供安全证据以及推动持续改进的角色, 从而保证系统的安全和有效风险管理.

尽管大量文献强调了基于论证的方法是提高软件系统可信度的有效手段[30], 实际上由于安全、保密和敏感性等因素, 完整的安全案例很少被公开. 为了解决这一问题, 本文精选了一批代表性的论文. 这些论文包括了从传统汽车[14]到自动驾驶汽车[15], 以及飞机[6]、无人机[8]、小型卫星系统[7]、网络安全[27]和心脏起搏器[18]多个应用场景的较完整安全案例.

值得注意的是, 存在多个国际功能安全标准, 如软件系统标准 (ISO/IEC 15026[31])、电气电子产品标准 (IEC 61508[32])、道路车辆标准 (ISO 26262[33])、铁路信号标准 (EN 50657[34])、航空电子软件标准 (DO-178C[35]) 和英国国防标准 (DS00-56[36]), 这些标准不仅为开发安全关键系统提供了明确的指导, 还特别推崇了使用安全案例来证明系统安全性. 例如, 在汽车安全领域, ISO 26262 被广泛应用, 并明确规定了使用安全案例来证明系统安全性. 一篇关于传统汽车的论文[14]详细描述了如何将 ISO 26262 标准融入安全案例中. 根据 EN 50129 标准, 铁路车辆的制造

商需要通过一个安全案例来证明他们的车辆对于预定用途是安全的[37]. DO-178C 标准的指导文件很少直接说明如何实现安全目标, 这个文档的安全案例是隐含的. 美国航空航天局 (NASA) 的论文[38]聚焦于识别 DO-178C 指导文件中暗示的特定论证, 将隐含的论证转向明确、具体化的安全案例.

安全案例旨在表达一个清晰、全面和可靠的论点, 即系统在特定环境下的操作满足可接受的安全性[39]. 安全案例是一种交流思想和信息的工具, 通常是向第三方 (如监管机构) 传达内容. 为了令人信服地做到这一点, 它必须尽可能地清晰. 安全案例所指的系统可以是任何对象, 例如一个管道网络、软件配置、一套操作程序, 这一概念并不局限于对传统工程“设计”的考虑. 绝对安全是一个无法实现的目标, 安全案例的存在是为了说服别人, 证明系统是足够安全的 (存在可容忍风险的可接受的安全). 安全的论证需要考虑前提, 如果以不适当或意外的方式使用, 几乎任何系统都可能是不安全的, 例如争论传统房屋砖块的安全性[40]. 因此, 安全案例的部分工作是定义安全的背景或特定环境. 安全案例由 3 个主要元素组成, 分别是目标、论点和证据, 这 3 个要素之间的关系如图 1 所示.

安全论证是为了表达证据和目标之间的逻辑关系. 然而, 在一些存在潜在安全隐患的项目中, 安全论证的重要性可能被低估, 其价值在项目审查和安全性讨论中往往被忽视. 常见的情况是, 项目文档中可能会包含大量的支持性证据, 如影响分析表和故障树[42], 但它们与安全目标之间的关系却少有明确阐述, 使读者不得不去猜测潜在的、隐含的论点. 对于安全案例来说, 论点和证据是两个关键组成部分, 它们必须同步考虑. 如果论点缺乏证据的支持, 它就会缺乏说服力; 如果证据缺乏论点的解释, 我们可能无法确定是否 (或如何) 满足了安全目标.

创建安全案例的过程分为 4 个基本步骤: 确定目标、收集证据、构建论证和评估安全案例[43]. 如图 2 所示, 这些步骤构建了安全案例的基本框架, 为安全工程师和项目经理提供了方向.

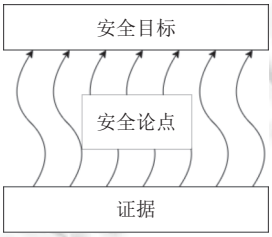

图 1 安全案例的主要组成部分[41]

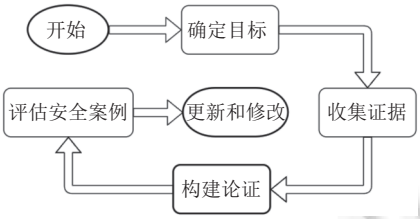

图 2 创建安全案例的 4 个基本步骤

步骤 1. 确定目标, 是安全案例创建的初始阶段. 在此阶段, 需要明确安全案例的目标, 这可能包括满足特定的安全标准、证明系统的安全性或满足客户的安全需求. 这个阶段应该明确地描述安全目标, 为后续步骤提供指导.

步骤 2. 收集证据, 是支持安全论证的基础. 证据可以来自各种来源, 如设计文档、测试结果、历史数据等. 有效的证据应该是可信的、相关的, 并能充分支持安全论证.

步骤 3. 构建论证, 是一个涉及将证据和目标通过逻辑关系链接起来的过程, 以形成一个完整的安全论证. 这一步骤通常需要使用如 GSN[44]、CAE[19]等安全案例表示方法, 以便以清晰、结构化的方式展示论证结构. 相关论文[45,46]提供了创建论证的 5 种常用思路.

步骤 4. 评估安全案例, 是确认其有效性的关键步骤. 这一步骤需要检查安全案例的完整性、一致性和可信度, 以确保其满足安全目标.

值得注意的是, 这 4 个步骤并不是一次性完成的, 而是在整个项目开发过程中反复进行的. 随着项目的发展和需求的变化, 安全案例可能需要进行更新和修改. 此外, 为了确保安全案例的质量和效果, 这 4 个步骤也需要团队之间的良好协作.

本文全面审查了超过 50 篇相关文献, 这其中不仅包括学术论文和技术报告, 还涵盖了国际标准和网站信息. 我们详细地分析了 8 种常用的安全案例表现形式, 探讨了它们的发展历程、优缺点, 以及各自适用的场景. 同时, 我们还评估了 7 款用于构建和评价安全案例的专业工具, 每款工具都是基于至少 4 个关键指标来进行评估的, 如运行环境、功能性、易用性和获取性. 此外, 基于对 15 个以上实际案例的综合分析, 本文深入地探讨了在安全案

例构建中可能遇到的 6 大挑战, 并分析了导致这些挑战出现的原因, 同时也提供了潜在的解决方案.

本文主要聚焦于创建安全案例的 4 个基本步骤中的第 3 步——构建论证. 第 2 节详细介绍现有的安全案例表现形式, 对其主要特征和适用情况进行深入的分析. 第 3 节列出并比较用于安全案例开发和评估的工具, 这为实际应用提供了参考. 第 4 节探讨安全案例构建中的各种挑战, 分析产生的原因并给出潜在的解决方案. 第 5 节展望安全案例的未来研究方向, 揭示其发展潜力和研究重点. 第 6 节对全文的内容进行总结, 梳理主要观点和研究结果.

2 安全案例的表现形式

现有的安全论证结构包括自然语言、表格结构、断言结构、可追溯性矩阵、贝叶斯可信度网络、目标结构符号 (GSN)[41]、声明-论点-证据 (CAE)[19]、结构化安全案例元模型 (SACM)[47]. 当前最常用的是 GSN 和 CAE 这两种形式.

2.1 自然语言

在许多现有的安全案例中, 自然语言常被用于构建和表达安全论证, 如图 3 所示的片段. 这种形式的论证通常清晰地阐述了安全需求 (例如纵深防御原则 P65) 在系统中的实现方式, 并提供了支持低级声明的证据的引用. 当然, 以结构化的方式通过文本表达安全论证可以是有效的[39].

然而, 当把自然语言作为唯一的表达方式时, 尤其是在处理复杂的安全论证时, 可能会引发一些问题. 图 4 展示了一个真实的工业安全案例片段, 其中体现了这类问题的一些例证.

> 纵深防御原则 (P65) 已在本系统中通过提供以下内容得到解决:
> 1. 在危险源和环境之间有多个物理屏障 (见第 X 节)
> 2. 存在防止这些屏障被破坏的保护系统，并减轻屏障被破坏的影响 (见第 Y 节)

图 3 自然语言形式的安全论点片段[39]

> 对于与警告有关的危险, 假想 [7] 中第 3.4 节表示没有发生设备故障时提出的警告被记录下来. 特别是关于第 5.7 节中的危险 17, 对于测试操作, 需要引入操作限制以防止危险, 同时进—步收集数据以确定问题的程度.

图 4 不清晰自然语言形式的安全论点片段[39]

首要问题是, 自然语言描述的表达可能并不清晰. 由于不是所有构建安全案例的工程师都能够撰写出结构清晰、易于理解的自然语言, 因此可能出现表达混乱、安全论证结构模糊的情况. 在自然语言描述中, 安全案例作为证据的集成器, 通常需要进行交叉引用. 然而, 过多的交叉引用可能会干扰主要论证的连贯性.

使用自然语言在安全案例中开发、确认和维护安全论证的最大问题, 是确保所有参与者对论证有相同的理解. 若无法达成对论证的清晰和共同理解, 管理安全案例可能会成为一项低效且定义模糊的任务. 因此, 寻求清晰、结构化的论证表达方式成为确保安全案例有效性的重要任务.

2.2 表格结构

用于表述安全论证的表格结构最初是在欧盟环境计划资助的 SHIP 项目[48]中提出, 该项目主要针对工业危害. 表格结构的安全案例后来也被收录在 DS 0055[49]的附件 E 中.

如表 1 所示 (源自文献 [49]), 表格结构主要分为 3 个部分来表达论证: (1) 目标: 论证的总体目标; (2) 论证: 对支持目标的论证类型的简要描述; (3) 证据/假设: 支持论证的证据或假设.

表 1 表格结构的安全案例示例

目标	论证	证据/假设
软件实现没有问题	形式化证明指定安全属性 形式化证明代码按规范实现	● 这个设计很简单, 经得起检验 ● 证明工具是正确的 ● 编译器生成正确的代码 ● 高质量的验证与确认过程 ● 测试结果
软件可靠性超出系统要求	可靠性可以在模拟操作条件下评估	● 统计检验结果

这种表格结构提供了一个简单的方法来构建论证. 相对于自然语言的描述, 表格结构更清晰地描绘了论证的各个组成部分. 然而, 在单一的表格结构中, 只能表示论证分解的两个步骤, 即从目标到论证, 以及从论证到证据. 对于更复杂的论证, 可能会涉及多个层次的论证和子论证, 这就需要在“论证”栏目中尝试表达多层次的论证结构, 或者使用多个表格来表达较低层次的论证分解. 在后一种情况下, “证据”栏目可能会转变为一个论证的子表格. 由此可能导致论证的清晰性或流畅性受损.

值得注意的是, 虽然表格结构的安全案例为构建论证提供了框架, 但对于如何在每一列中表达信息, 几乎没有提供具体的指导[39].

2.3 断言结构

断言结构出现在 DS 0055[49]的附件 H 中, 用于为 SHOLIS (船舶直升机操作仪表系统) 项目的开发过程提供安全论证. 如图 5 (取自文献 [49] 的附件 H) 所示, 断言结构由一个总断言以及通过 AND 和 OR 门连接的若干子断言构成.

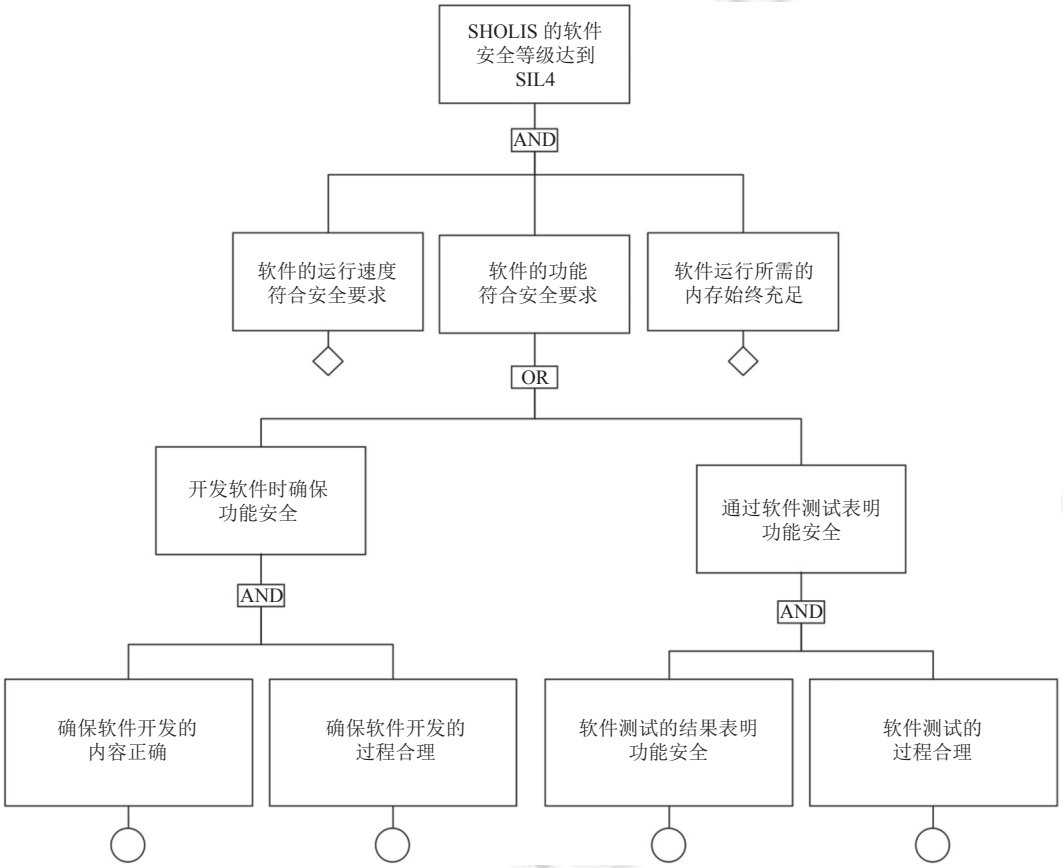

图 5 断言结构的安全论点案例[49]

在断言结构中, AND 门代表一个联结关系, 即所有子断言都必须为真, 才能证实上一级的断言为真. 相反, OR 门代表一个选择关系, 即子断言中只要有任何一个为真, 就足以确认上一级的断言为真. 这两种逻辑门提供了构建复杂安全论证的灵活性. 断言被逐级分解, 直到到达基本断言 (由所附的圆圈表示) 或未开发的断言. 基本断言是由证据支持的, 但证据的作用并没有以图形的方式表示出来.

在图 5 的根节点中, 安全完整性等级 4 (safety integrity level 4, SIL4)[50]是工业过程安全领域中用于评估系统可靠性和安全性的一种级别. SIL 分为 4 个等级, SIL1–SIL4, 等级越高, 系统的安全性能和可靠性越好. SIL 等级是通过对系统进行风险分析和评估而确定的, 通常应用于工业自动化、化工、石油天然气等行业的安全仪表系统

(SIS). SIL4 级别的系统具有非常高的安全要求, 其设计、验证和维护成本相对较高, 因此只有在极端情况下才需要这个等级. 相比之下, SIL1–SIL3 级别的系统可以满足大部分工业应用场景的安全需求.

断言结构可视为目标结构符号 (GSN) 的简化版本. 除了简单的 AND 和 OR 组合外, 断言结构并未提供表示论证策略的方式, 也没有用图形方式表示证据、论证理由和论证背景.

2.4 可追溯性矩阵

可追溯性矩阵是一种工具, 用于表示设计特性 (如断言、需求、目标等) 如何与一系列其他需求相互关联. 在需求工程和安全领域, 可追溯性矩阵已经得到广泛的应用[51]. 如图 6 所示, 给出了一个可追溯性矩阵的示例 (取自文献 [39]).

设计特性	需求									
	REQ1	REQ2	REQ3	REQ4	REQ5	REQ6	REQ7	REQ8	REQ9	REQ10
冗余通道设计		■	■			■	■	■	■	
异常处理设计		■		■	■			■		■
独立的监控设备					■	■				■
设计简洁	■			■						■
形式化证明软件	■	■	■							

图 6 可追溯性矩阵示例[39]

在图 6 中, 我们可以看到高级需求 (列在矩阵顶部) 是如何与低级设计特性 (列在矩阵左侧) 关联的. 矩阵中的黑色方块表示某个设计特性与特定的需求相关, 如果没有黑色方块则表示不相关.

然而, 尽管可追溯性矩阵清楚地展示了设计特性和需求之间的关系, 但它们在表达分解层次上有所限制, 一次只能表示一个层次. 因此, 可能需要多个矩阵来表示声明的深度分解. 此外, 它们不能表示更低级别特性的安全问题, 也没有提供证明存在于较高层次和较低层次声明之间关系的方法.

2.5 贝叶斯可信度网络

贝叶斯可信度网络[39], 也称为因果概率网络, 是一种图形网络, 用于表示变量之间的概率因果关系. 如图 7 所示, 这是一个贝叶斯可信度网络的示例. 在图 7 中, 节点表示事实, 箭头则表示事实之间的因果关系.

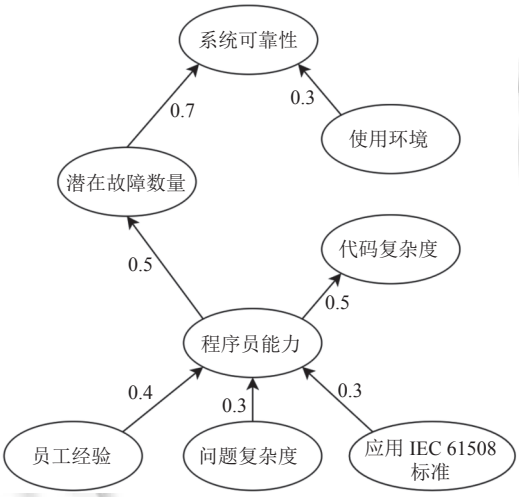

图 7 利用过程和产品证据预测可靠性的贝叶斯可信度网络实例[39]

在贝叶斯可信度网络中, 条件概率用于明确表示不同事实之间的依赖关系和可信度. 例如, 图 7 中的例子揭示了程序员的能力与其经验、所处理的问题的复杂性, 以及是否使用了软件标准 IEC 61508[32,52]之间的关系. 条件概

率用于表示程序员能力取决于这些因素的程度.

贝叶斯可信度网络可以用于推导出与系统安全性相关的定量声明, 例如总体可靠性. 它的优点在于可以基于不确定或部分的数据预测事实的可信度. 其主要缺点在于确定表示事实之间因果关系程度的条件概率是一项相当主观的任务. 然而, 如果事实是可观察的属性, 有更多的数据可用, 并且条件概率可以随时间的推移提高, 那么使用贝叶斯可信度网络来构建安全案例是一个可行的选择.

贝叶斯可信度网络提供了一种表示安全论证和断言之间关系的方法[53]. 然而, 作为一种可视化表示, 它只能隐含地传达安全论证的概率值, 无法表示推理原因等信息. 例如, 在图 7 中, 贝叶斯可信度网络并没有明确提出断言, 节点只被标记为名词或短语, 大部分的可信度信息被包含在与箭头和节点相关的条件概率中. 相较于纯粹的论证方法 (如第 2.6 节将介绍的 GSN), 贝叶斯可信度网络的优点在于它能够建立定性和定量安全属性之间的因果关系. 并且, 它提供的证据 (如故障树) 可以用于支持安全论证中的定量主张.

2.6 目标结构符号 (GSN)

目标结构符号 (GSN) 最初在 ASAM-II 项目[54,55]中被开发, 用于表示安全案例. 该项目由约克大学联同英国航空航天公司、劳埃德船级社和劳斯莱斯公司共同领导, 其旨在为安全案例的开发提供一种结构化的方法和全面的工具支持.

目前, GSN 的标准化信息和相关指导可在 SCSC 网站[44]上获得. GSN 信息由 GSN 标准工作组 (GSN_SWG) 创建和维护, 该工作组是 SCSC 的安全案例工作组 (ACWG) 的一个分支. 该网站的主要目标在于传播与 GSN 相关的信息和资源.

最初被称为目标层次结构的 GSN, 是一种图形化展示安全论证结构的方法. GSN 由节点元素和关系元素组成, 其中主要的节点类型如图 8(a) 所示, 关系类型则如图 8(b) 所示.

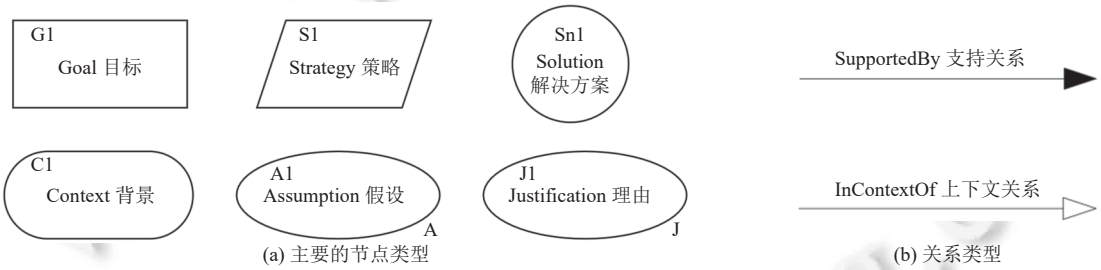

图 8 GSN 元素的图形符号

目标结构符号 (GSN) 中的结点元素主要分为 6 种类型, 分别是目标、策略、解决方案、背景、假设和理由.

(1) 目标 (goal): 表示为矩形, 目标是论证中的一种主张, 通常由子目标、策略或解决方案支持. 目标描述关于系统特性、性能或安全性等方面的断言.

(2) 策略 (strategy): 表示为平行四边形, 策略描述目标与其支持目标之间的推理关系. 策略明确如何通过支持目标、解决方案或其他证据来满足上级目标.

(3) 解决方案 (solution): 表示为圆形, 解决方案提供对证据项目的引用. 证据可以是实验数据、分析报告或其他支持论证的材料.

(4) 背景 (context): 表示为跑道形, 背景提供关于目标或策略的附加信息. 背景有助于定义术语、概念或系统的操作环境, 以确保论证在正确的场景下理解.

(5) 假设 (assumption): 表示为带有字母“A”的椭圆形, 假设是一种不能立即证实的陈述. 假设用于表示不确定性、论证范围的限制或某些主张另外处理的情况.

(6) 理由 (justification): 表示为带有字母“J”的椭圆形, 理由是用于解释采用特定策略的陈述. 理由可以提供对标准要求的响应, 或解释为什么选择特定方法来支持主张.

关系元素主要有两种类型, 分别是支持关系和上下文关系.

(1) 支持关系 (SupportedBy): 用实心箭头线表示, 记录元素间的支持关系. 支持关系表明一个目标、策略或解决方案如何支持上级目标. 允许的连接包括: 目标到目标, 目标到策略, 目标到解决方案, 以及策略到目标.

(2) 上下文关系 (InContextOf): 用空心箭头线表示, 声明了一个上下文关系. 上下文关系表明一个目标或策略与背景、假设或理由之间的联系. 允许的连接包括: 目标到背景, 目标到假设, 目标到理由, 策略到背景, 策略到假设, 以及策略到理由.

后文图 9 展示了一个目标结构符号 (GSN) 的实例, 该实例描述了对控制系统的安全论证. 其中, 该论证考虑了所有已识别的危险已经被消除或充分减轻, 且明确表示了控制系统中的软件开发已经根据所涉及的危险等级开发到适当的安全完整性等级 (SIL)[44].

目标结构符号 (GSN) 是目前最广泛使用的安全论证方法之一, 其主要优点如下.

(1) 提供了明确的安全论证逻辑流程表示, 通过目标、策略等之间的定向“解决”关系.

(2) 明确表述了证据的作用, 通过解决方案符号.

(3) 明确标注了论证所依据的理由, 通过理由符号.

然而, 目标结构符号也存在一些缺点. 它并未提供明确的指导关于如何构建目标结构. 每个节点的描述内容都是非形式化的自然语言, 对于目标的拆分方法没有提供完备性和正确性的证明. 因此, 安全工程师在应用此方法时可能会觉得困难. 这也意味着在如何使用这种符号上存在着很大的差异.

2.7 声明-论点-证据 (CAE)

声明-论点-证据 (claims-arguments-evidence, CAE) 是一种由 Adelard 公司开发的简单有效的安全论证表述和交流方法, 它在 ASCE 软件中是一个重要的表示法[46]. CAE 方法将整个论证结构化为 3 类元素: 声明、论点和证据, 如图 10 所示.

(1) 声明 (claim): 声明是论证中的陈述, 可以被评估为真或假. 每个声明由若干子声明、论点或证据支持. 声明中可能包含额外的背景材料, 例如定义使用的术语和范围.

(2) 论点 (argument): 论点描述了支持声明的论证方法. 这个元素是可选的, 但一般建议包含, 以便阐释如何满足父级声明的方式. 然而, 如果支持声明的方法非常直观或被预期的受众所理解, 可以直接连接父子声明, 而省略论点.

(3) 证据 (evidence): 证据指引用支持声明或论点的材料. 一般情况下, 证据节点会概述并链接到包含相关证据的报告. ASCE 提供了一系列工具以便于链接、管理和跟踪证据的变化.

CAE 框架通过整合声明、论点和证据, 使工程师和利益相关者能够清晰地展示和评估系统的安全性. 同时, 这也有助于识别安全论证中的缺陷或不足, 进而对系统设计进行改进以确保满足安全需求.

后文图 11 展示了一个 CAE 的实例. 总声明表示“这款产品对环境友好”, 它由两个子声明——“该产品使用可再生材料制成”和“该产品的生产过程产生的碳排放量低”支持. 这两个子声明分别从产品材料和生产过程两个角度论证总声明, 每个子声明都有一系列证据支撑. 通过这样的论证和证据, 我们得出了结论: 这款产品对环境友好.

CAE 和 GSN 都是表达和沟通安全论证的方法. 虽然它们在一些方面具有相似性, 但在其他方面也有所不同.

(1) 灵活性: GSN 提供了更丰富的元素类型 (6 种), 使得其在表示复杂的安全论证上更具灵活性. 而 CAE 只有 3 种基本元素 (声明、论点和证据), 在某些情况下可能显得不够灵活.

(2) 标准化程度: GSN 已经发展成为一个广泛使用的安全论证表示法, 并被纳入了一些国际标准[56], 因此, GSN 被视为具有更高的标准化程度. 虽然 CAE 也得到了广泛应用, 但它可能没有 GSN 那样被广泛接受.

(3) 工具支持: GSN 具有丰富的工具支持[57]. 这些工具提供了强大的功能, 如图形编辑、模型验证和报告生成. 尽管 CAE 也可以在一些工具中使用, 但其工具支持可能不如 GSN 完备.

值得注意的是, CAE 的限制并不意味着它在特定场景下就不适用. 实际上, CAE 的简单性和直接性在某些情况下可能是优点, 使其更适合用于表达安全论证. 在选择表示法时, 应考虑项目需求、团队的熟悉程度以及各种表示法的优点和局限.

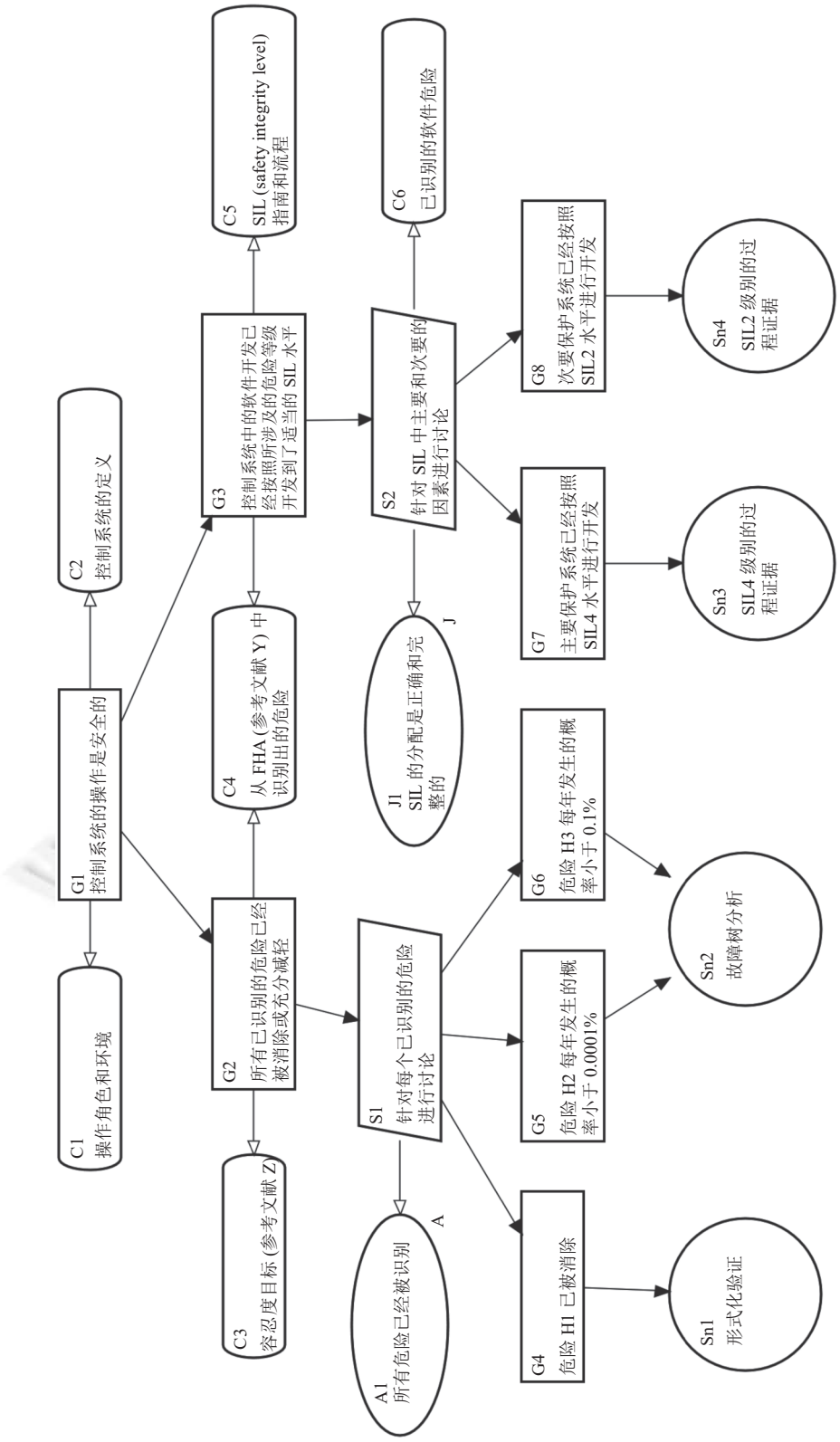

图 9 目标结构符号 (GSN) 示例[44]

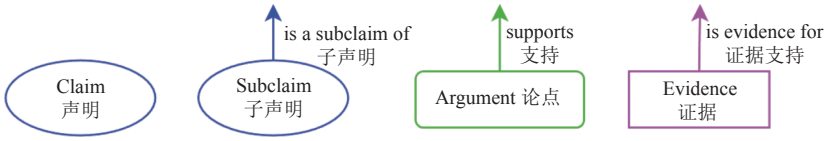

图 10 CAE 元素的图形符号

2.8 结构化安全案例元模型 (SACM)

结构化安全案例元模型 (structured assurance case metamodel, SACM)[47,58]是由对象管理小组 (OMG) 制定的标准[59]. 该标准是一种用于表示和分析系统安全案例的方法. 这些案例的目标是证明系统在特定环境中满足其安全性或其他关键属性的要求. SACM 为表达这些论证提供了一种方式, 以支持系统开发过程中的关键决策. SACM 由 5 个组件构成, 如图 12 所示.

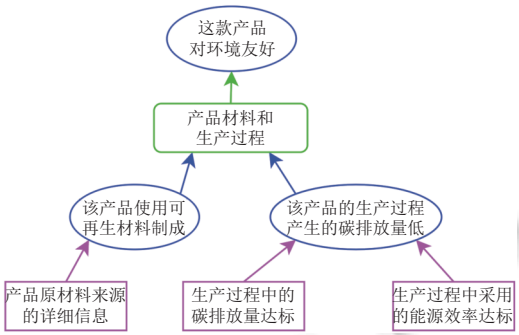

图 11 一个声明-论点-证据 (CAE) 的例子

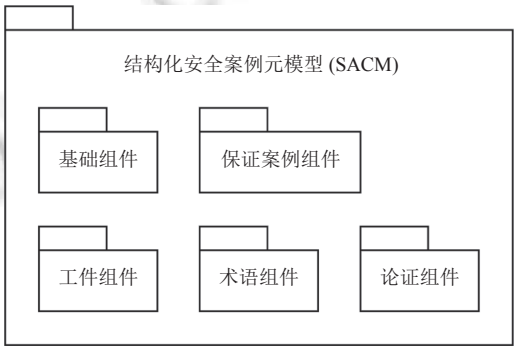

图 12 结构化安全案例元模型的组成部分[47]

SACM 定义了一种基于图形的语言, 用于创建和表达关于系统属性的结构化论证. SACM 的核心组件如下.

(1) 基础组件 (base): 为其他组件提供基本的结构和元素.

(2) 安全案例组件 (assurance case): 用于定义和组织安全案例的结构.

(3) 工件组件 (artifact): 记录与安全案例相关的工件和活动信息.

(4) 术语组件 (terminology): 定义控制词汇, 以更精确地描述论证过程.

(5) 论证组件 (argumentation): 建立关于系统属性的结构化论证.

尽管 SACM 提供了一种详细和全面的方式来表示和分析安全案例, 但其在实践中的应用尚未广泛推广. 一方面, 由于 SACM 相比其他方法更为复杂, 对于简单的安全案例, 可能不需要使用所有功能. 另一方面, 学习和掌握 SACM 可能需要更多的时间, 因为它涉及多个组件和概念. 然而, 已有部分研究人员和公司开发了工具和技术来支持 SACM 的使用, 如建模工具[60]、自动推理引擎和验证方法[61]. 这些应用和工具的发展反映出 SACM 在支持复杂系统的安全性和可靠性方面的潜力. 随着相关研究和实践的不断进步, SACM 有可能会继续改进和扩展, 以满足未来系统保证的需求.

2.9 表现形式总结

本节介绍了安全论证结构的表现形式, 包括自然语言、表格结构、断言结构、可追溯性矩阵、贝叶斯可信度网络、目标结构符号 (GSN)、声明-论点-证据 (CAE) 和结构化安全案例元模型 (SACM). 表 2 总结了这些表现形式的优缺点.

上述安全案例的表现形式中, 自然语言形式是最传统的表现方法, 表格结构弥补了自然语言结构模糊的缺点, 断言结构解决了表格结构论证层数少的问题. 目标结构符号 (GSN) 和声明-论点-证据 (CAE) 在断言结构的基础上补充了证据、论证理由和论证背景等辅助论证信息. 结构化安全案例元模型 (SACM) 进一步添加术语、工件、活动信息等内容. 可追溯性矩阵从需求与设计特性的关系方面进行论证. 贝叶斯可信度网络可以定量地计算论证目

标的可信度. 在选择安全案例表示法时, 应考虑项目需求、团队熟悉程度以及表示法的优缺点选择一个合适的安全案例表示法进行论证.

表 2 安全案例表现形式优缺点总结

表现形式	优点	缺点
自然语言	入门容易	论证结构模糊、容易产生歧义
表格结构	论点描绘清楚	不适用于多层论证
断言结构	适用于多层论证	没有证据、论证理由和论证背景
可追溯性矩阵	需求与设计的关系表示清楚	只能表示高级别的设计特性、只能表示一层
贝叶斯可信度网络	可以定量预测可信度	初始数据设定人为主观、不能表达推理原因
目标结构符号 (GSN)	支持多层论证、标准化程度高、支持的工具多	构建困难
声明-论点-证据 (CAE)	简洁清晰、支持多层论证	部分场景不够灵活、支持的工具少
结构化安全案例元模型 (SACM)	论证内容全面、标准化程度高	表现形式复杂、支持的工具少

3 安全案例的工具

由于安全案例所需材料的显著复杂性, 软件工具经常被用作构建和评估安全案例的有效手段. Maksimov 等人在文献 [57] 提供了一份包含 37 个安全案例工具的综合清单, 并对其功能进行了比较. 文献 [62] 介绍了包含在 10 个安全案例软件工具中的各种安全案例评估特征的调查. 目标结构符号 (GSN) 标准化信息网站[63]罗列了 7 个优秀的可以构建 GSN 的工具.

在本节中, 我们将介绍 GSN (goal structuring notation) 标准化信息网站上列出的 GSN 工具 (astah system safety、gsn2x、NOR-STA、Socrates、ASCE、D-Case Editor 和 AdvoCATE). 值得注意的是, 这些工具不仅支持 GSN, 部分工具还支持 CAE (claims-arguments-evidence)、SACM (structured assurance case metamodel) 等其他符号. 这些工具以多种形式呈现, 包括桌面工具、在线平台、插件和命令行工具. 此外, 它们在可获取性和成本方面也有所不同, 有些是免费且容易获取的, 有些则是付费的商业工具, 还有一些是在有限范围内使用且难以获取的工具. 为了更好地了解这些工具, 接下来我们将对它们的特点和功能进行细致的比较, 如表 3 所示. 通过对这些工具的综合评估, 我们旨在为读者提供一个全面的了解, 以便在实际应用中选择最适合自己需求的工具.

表 3 安全案例工具列表

工具名称	工具类型	支持的符号	是否开源	入门难度
astah system safety[64]	桌面工具	GSN、SysML、STPA、SCDL	否	*
gsn2x[65]	命令行工具	GSN	是	**
NOR-STA[66]	在线平台	GSN	否	**
Socrates[67]	在线平台	GSN、EA	否	**
ASCE[68]	桌面工具	GSN、CAE、SACM	否	***
D-Case Editor[69]	Eclipse插件、浏览器	GSN、D-Case	是	***
AdvoCATE[70]	Eclipse插件	GSN	否	*****

3.1 astah system safety

astah system safety[64]是一款综合性的系统安全建模工具, 特别适用于安全攸关系统的建模和分析, 如图 13 所示. 该工具在汽车领域表现优秀, 并适用于航空航天、铁路、国防、机器人以及医疗保健等行业. astah system safety 为那些有兴趣采用模型驱动系统工程 (MBSE) 的公司和工程师提供了一种有效的方法来评估和分析安全攸关系统.

astah system safety 将多种建模语言和技术整合在一个工具中, 包括 SysML、STAMP/STPA、GSN/D-Case 和 ASAM SCDL. SysML 是一种通用的建模语言, 专为系统工程而设计, 基于统一建模语言 (UML) 并对其进行了扩展, 以更好地满足复杂系统的建模需求. STAMP/STPA 是一种基于系统理论的事故因果模型和分析技术, 由学者 Ishimatsu 及麻省理工学院的 Leveson 教授开发[71]. GSN/D-Case 分别是用于可视化安全论证和描述系统可靠性与

依赖性的表示法. ASAM SCDL 则是一种半形式化的表示法, 用于描述符合 ISO 26262 标准的安全体系结构. astah system safety 将这 4 种建模语言和技术很好地整合在一个工具中, 使工程师能够在一个工具中完成系统安全评估和分析, 从而提高工作效率并降低开发成本.

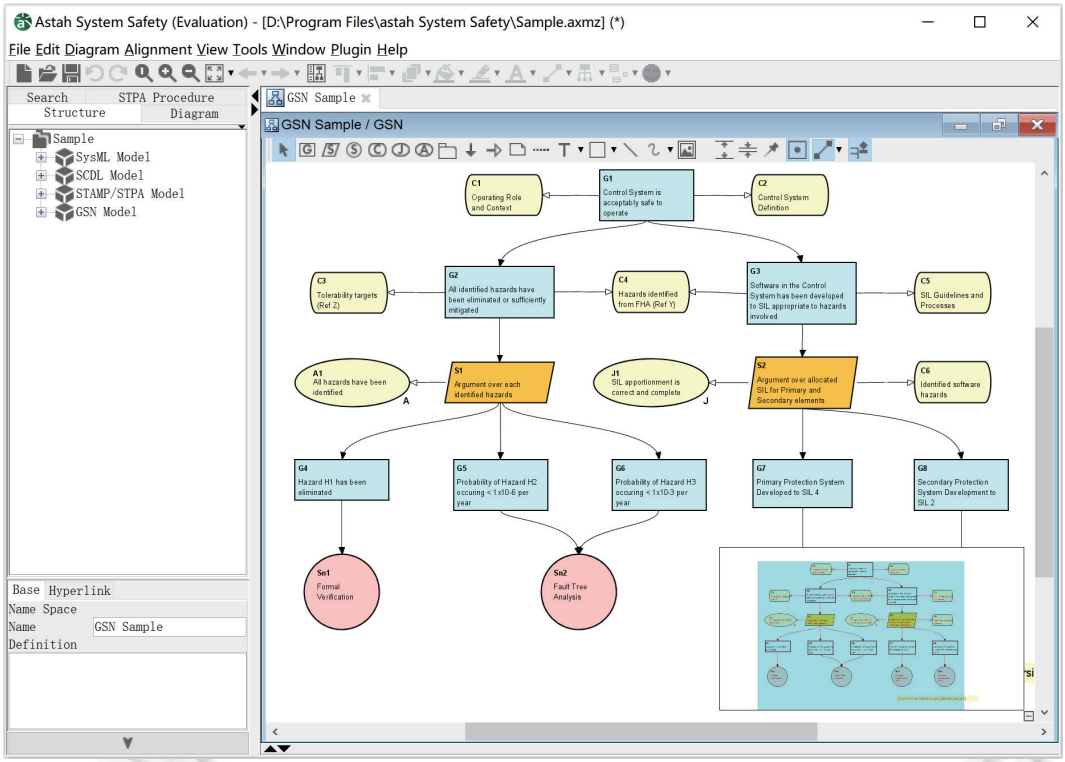

图 13 astah system safety 工具界面

尽管 astah system safety 在系统安全建模和分析方面表现出较强的综合能力, 但在某些方面仍然存在一些局限性. 首先, astah system safety 的学习曲线相对较陡峭, 尤其是对于初次接触系统建模和安全评估的工程师. 因此, 这可能会影响到新用户的工作效率. 其次, astah system safety 的许可费用可能会对预算有限的团队和个人用户造成一定的经济负担. 此外, 该工具在实时协作功能方面略显不足, 可能影响团队间的沟通和协同工作效率. 与其他工具的兼容性也是一个潜在的问题, 可能需要额外的时间和努力进行数据转换和适配. 在处理大型和复杂的系统模型时, astah system safety 可能对计算资源的要求较高, 这在计算资源有限的环境中可能会导致性能下降. 此外, 虽然该工具定期发布更新和新功能, 但用户可能需要等待新版本发布以解决一些问题或获取新的功能. 最后, 相较于一些更为流行的系统工程工具, astah system safety 的社区支持可能相对较弱, 这可能会在寻求帮助和解决问题时带来一定的挑战. 尽管如此, astah system safety 仍然是一款有价值的工具, 对于许多安全攸关系统的开发团队来说, 可以在一个集成环境中进行系统安全评估和分析.

3.2 gsn2x

gsn2x[65]是一款开源软件, 旨在以简洁易用、最小依赖、易于集成和符合 GSN 标准的方式, 实现将以 YAML 格式编写的 GSN 模型转换为可缩放的矢量图像 (SVG). 该工具以其自动错误检查功能, 保证了 GSN 图的正确性和一致性, 同时支持模块化扩展和自定义 CSS 样式表, 强化了其灵活性和用户体验. gsn2x 的一些实用特性, 包括布局控制、图层功能、完整视图和架构视图的生成、证据列表的创建, 以及可定制的模块信息, 都进一步提升了用户在创建和维护 GSN 论证时的效率和体验. 尽管 gsn2x 并未支持所有的 GSN 标准扩展, 但其核心功能已能满

足大部分用户需求, 且易于集成到持续集成环境中, 实现自动化生成和验证 GSN 图. 因此, gsn2x 凭借其简洁易用、功能丰富和灵活可定制的特性, 已成为资源有限环境下快速构建和分享安全论证的理想工具.

在本文中, 我们将提供一个 gsn2x 使用的实例, 以便更好地描述其处理基本 GSN 结构的方式. 这个例子将涉及一个名为 example.yaml 的 YAML 文件, 这个文件的内容如图 14 所示.

```yaml
G1:
  text: 目标
  supportedBy: [S1]
  inContextOf: [A1]

S1:
  text: 分解策略
  supportedBy: [G2,G3]

A1:
  text: 假设 1

G2:
  text: 子目标 1
  inContextOf: [J1]
  undeveloped: true

G3:
  text: 子目标 2
  supportedBy: [Sn1]
  inContextOf: [J1,C1]

Sn1:
  text: 解决方案 1
  url: https://github.com/jonasthewolf/gsn2x

J1:
  text: 理由 1

C1:
  text: 背景 1
```

图 14 描述 GSN 的 YAML 文本

此实例包含了对 GSN 的 6 种元素类型和 2 种关系类型的定义, 其中 G2 是一个待开发的目标. 为了使用 gsn2x 将此 YAML 文件转化为 SVG 格式的 GSN 图, 需要在命令行中执行 gsn2x example.yaml 指令.

运行此命令后, 将在当前目录下生成一个名为 example.svg 的文件, 如图 15 所示, 其中呈现了采用 GSN 标准形式的论证结构. 这个 SVG 文件可以在任何支持 SVG 格式的图像查看器或编辑器中查看和修改.

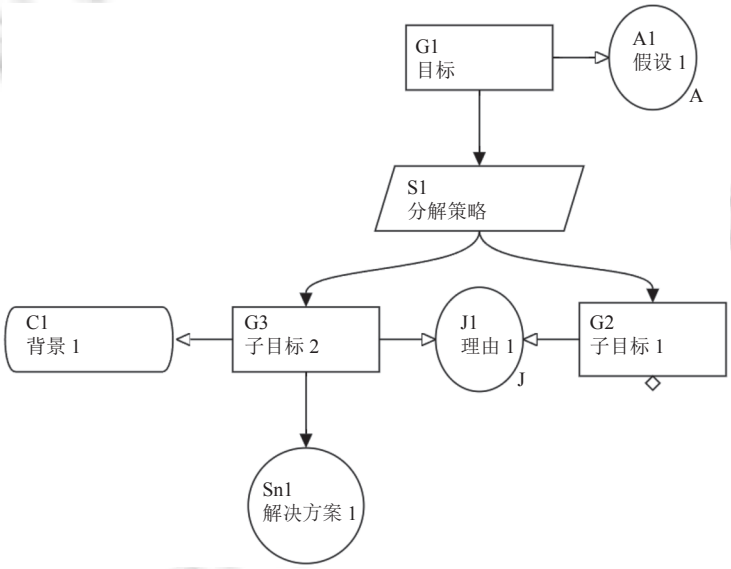

图 15 gsn2x 案例生成的图形

尽管 gsn2x 作为一款轻量级的开源软件在创建和维护 GSN 论证方面具有诸多优势, 但它在某些方面仍然存在一些局限性. 首先, 由于 gsn2x 仅支持 YAML 格式作为输入文件, 这可能会对那些习惯于使用其他格式 (如 XML、JSON 等) 的用户带来一定的学习成本. 其次, gsn2x 目前仅支持 SVG 作为输出图像格式, 这可能限制了

某些用户对输出结果的进一步编辑和操作. 此外, 虽然 gsn2x 提供了一定程度的布局控制, 但它可能无法满足所有用户对图像布局和美观的高度要求. 在某些情况下, 用户可能需要手动调整生成的 SVG 图像以实现期望的视觉效果. 同时, gsn2x 作为一个命令行工具可能在易用性方面略逊于图形界面, 这可能导致新用户在上手时遇到一定的困难. 尽管 gsn2x 具有输入验证功能, 但其错误检测和提示可能不如一些商业 GSN 工具那么详细和准确. 这可能会导致用户在解决问题和修复错误时花费更多的时间. 最后, 由于 gsn2x 是一个开源项目, 其技术支持和功能更新可能不如商业软件那么及时和完善. 在某些情况下, 用户可能需要自行解决问题或等待开源社区的响应. 总的来说, 虽然 gsn2x 存在一些不足之处, 但其在快速构建和分享安全论证方面的优势仍使其成为一个有价值的工具.

3.3 NOR-STA

NOR-STA[66]是一个全面的网络化安全案例支持工具, 专门为处理复杂系统的安全和可靠性保证而设计, 具有创建、编辑、评估和管理安全案例的全套功能, 且支持在线协作和权限控制. NOR-STA 遵循核心 GSN 标准, 部分实现了模块化和论证模式扩展, 并且其论证元模型遵循了 OMG SACM 和 ISO/IEC 15026 标准. NOR-STA 的模块化功能可以帮助用户将大型论证分解, 方便在团队和组织中分配工作, 实现权限单独分配. 此外, NOR-STA 提供了证据集成功能, 能够直接从各种知识库中引用证据, 并监控证据变化, 提供系统化的变更管理. 其评估功能集成了论证评估过程, 并支持评审员和第三方评估员的专用角色. NOR-STA 还具有丰富的报告功能, 支持使用颜色比例进行评估可视化, 并可将报告导出为多种格式, 同时支持利用 XML 数据和自定义报告模板创建定制报告内容. 因此, NOR-STA 凭借其全面的功能集、在线协作支持以及适应不同行业和组织的安全案例管理需求的能力, 已成为一款强大的 GSN 工具, 适用于处理复杂的系统安全和可靠性保证问题.

在 NOR-STA 工具中, 安全案例的编辑采用了树形表的形式, 如图 16 所示, 并且可以进一步转化为标准的 GSN 形式, 如图 17 所示.

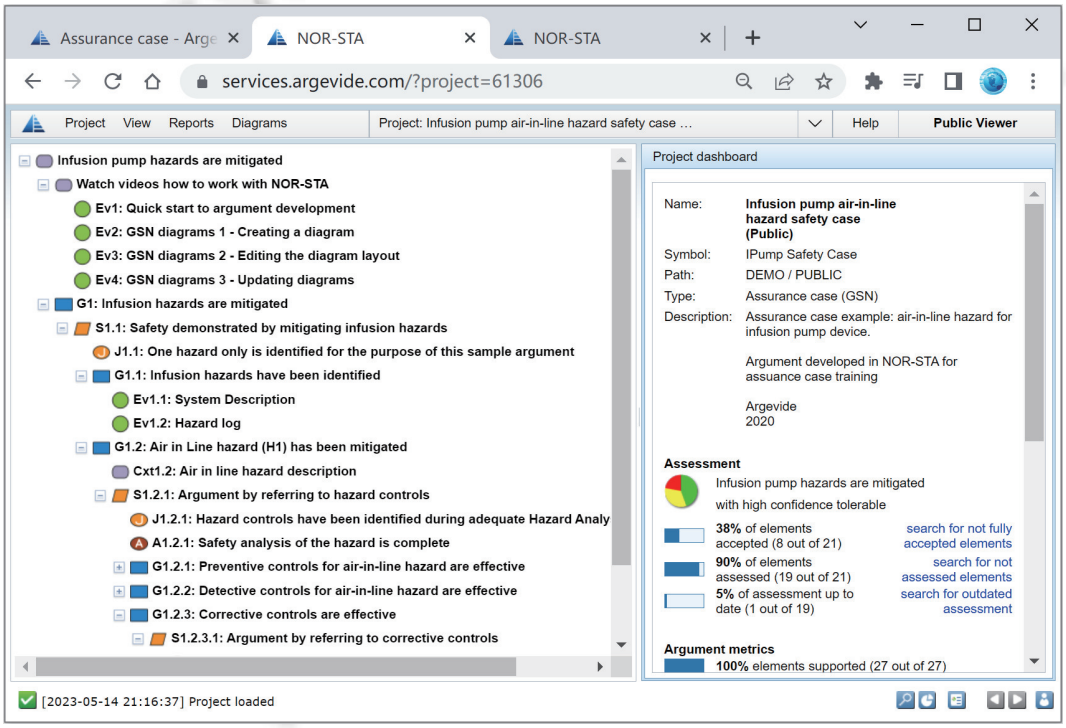

图 16 NOR-STA 工具页面中编辑安全案例界面

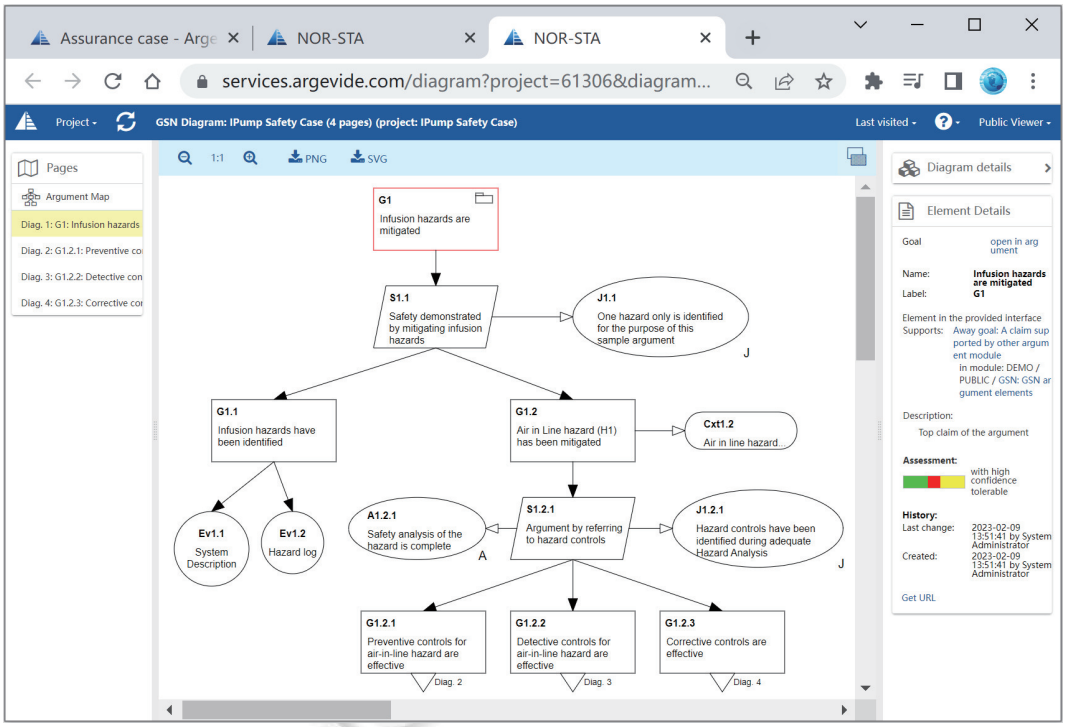

图 17 NOR-STA 工具页面中查看标准 GSN 形式界面

虽然 NOR-STA 在处理复杂系统安全和依赖性保证问题方面具有很强的能力, 但在某些方面仍然存在一定的局限性. 首先, NOR-STA 的学习成本和配置复杂性相对较高, 对于初次接触 GSN 或拥有较小规模需求的用户, 可能会显得过于复杂. 其次, NOR-STA 的许多高级功能依赖于与其他企业级知识库的集成, 对于没有这些知识库或者使用其他文档管理系统的用户, 可能无法充分利用其全部功能. 此外, 尽管支持在线协作和权限控制, 但 NOR-STA 可能在灵活性和可扩展性方面不如一些云原生 GSN 工具. 在评估功能方面, 虽然提供多种评估方法, 但这些方法可能无法完全满足所有用户的需求, 特定情况下可能需要自定义评估过程. 最后, 考虑到 NOR-STA 主要针对企业用户, 其许可和定价可能对个人用户和小型组织来说相对昂贵, 导致这些用户寻求更便宜或免费的 GSN 工具替代方案.

3.4 Socrates

Socrates[67]是一款专为关键系统开发而设计的协同安全案例管理平台, 它的独特之处在于其设计初衷就强调了协作, 使得团队中的每个成员都能为安全案例做出贡献. 这款平台由具有丰富经验的 Critical Systems Labs 开发, 他们在关键系统领域拥有超过 60 年的经验, 他们将这些知识和经验融入了 Socrates 之中. Socrates 也考虑到了安全案例中可能包含的敏感信息, 因此提供了部署在组织内部的 IT 基础设施的解决方案, 以确保数据完全在用户的控制范围内. 此外, 它还支持最新的论证技术和符号, 包括目标结构表示法 (GSN) 和消除论证法 (EA).

消除论证法 (EA) 是一种在安全案例中评估和推理的方法, 其目标是识别和消除不确定性、风险和漏洞, 从而建立对系统的信心. 与目标结构表示法 (GSN) 这种建设性方法相比, EA 更注重发现系统中的潜在问题并消除这些问题. EA 的主要优点是, 它鼓励团队积极发现和解决问题, 而不是仅关注证明系统的安全性. 此外, EA 可以更好地适应系统的变化, 因为它允许团队在系统开发过程中不断识别和解决新的问题.

在功能方面, Socrates 提供了协作多用户环境、多视图编辑、讨论与问题管理、工件管理、图表导出、论证导入与导出、程序化访问、灵活托管以及用户管理等多种功能, 可满足多样化的需求. 总体来说, Socrates 是一款强大的安全案例管理平台, 适用于关键系统的开发和管理. Socrates 工具提供了一种树形表的形式用于编辑安全案例, 如图 18 所示, 并且能够将其转化为标准的 GSN 形式. 与 NOR-STA 不同的是, 尽管 Socrates 无法显示完整的

GSN, 但它能从树形表中的特定节点跳转到对应的 GSN 片段进行显示, 如图 19 所示. 此外, Socrates 工具的运行速度相较于 NOR-STA 具有明显的优势.

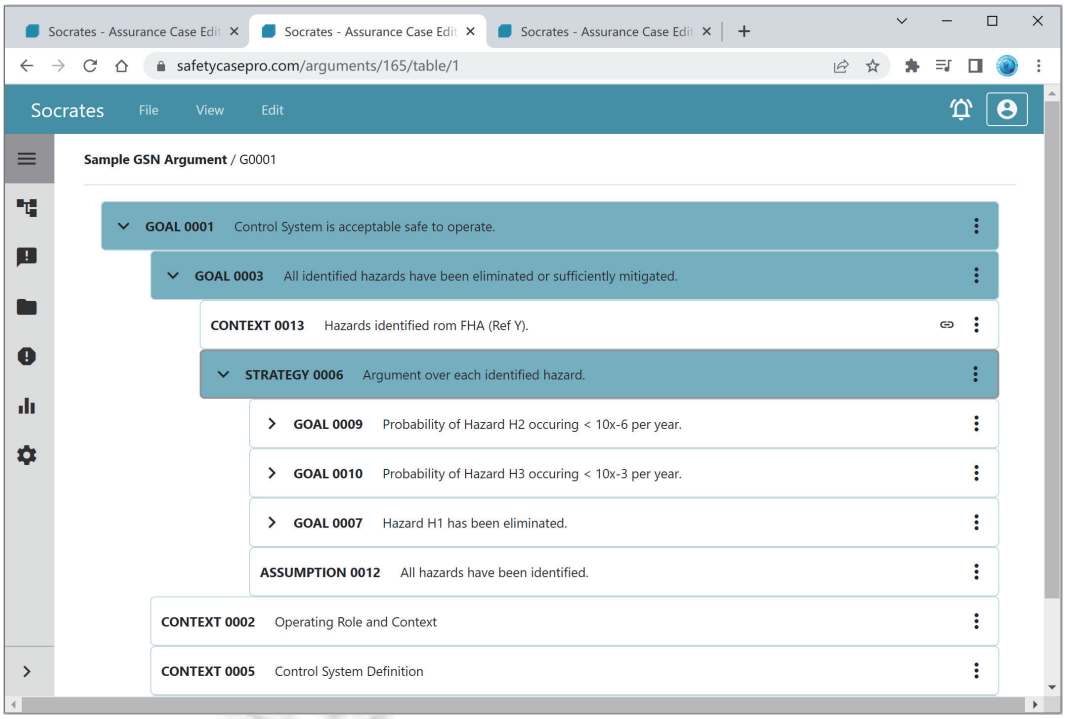

图 18 Socrates 工具页面中编辑安全案例界面

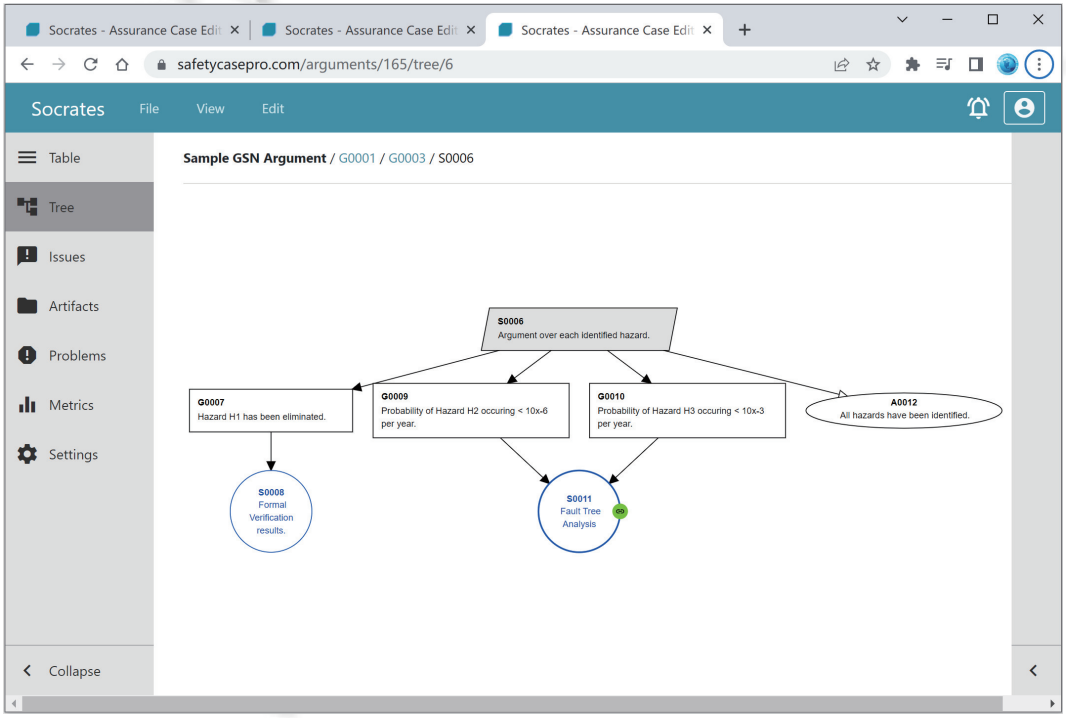

图 19 Socrates 工具页面中查看 GSN 片段界面

在评估 Socrates 作为一款关键系统开发的协同安全案例管理平台时, 我们也需要关注其潜在的局限性. 首先, 尽管 Socrates 提供了丰富的功能, 但初学者可能需要一定时间来熟悉其操作方式和功能. 此外, 由于 Socrates 专为关键系统开发而设计, 其在非关键系统项目中的适用性可能受限. 作为基于 Web 的在线平台, Socrates 需要稳定的网络连接以保证良好的用户体验, 因此在网络不稳定的情况下, 使用效果可能会受到影响. 从经济角度来看, Socrates 的价格可能对一些小型团队或个人用户构成使用障碍. 最后, 尽管 Socrates 提供了 Web API 以实现与其他工具的集成, 但在与特定系统工程工具集成时仍可能面临兼容性挑战. 因此, 在考虑采用 Socrates 时, 需要充分评估这些潜在缺点, 以确保其能满足项目需求.

3.5 ASCE

ASCE (assurance and safety case environment)[68]是一款被广泛采用的商业软件, 专用于创建和管理安全案例. 通过表达结构化的论证和相关证据, ASCE 能够降低项目和系统风险, 使得安全信息的管理和法规合规性证明更为简单、高效和经济. 在其第 5 代版本中, ASCE 全面支持了最新的 GSN 社区标准[44]. 此外, ASCE 还根据现代化框架 Assurance 2.0 更新了 CAE 功能, 以应对持续有效且及时的保障以及技术创新 (如机器学习) 带来的挑战. 在 ASCE 中应用该框架将有助于采用更严谨、更有说服力的方法来应对下一代安全案例, 支持创新和持续增量保障.

ASCE 的显著特点包括提高安全案例开发与管理的可见性, 通过图形化表示整个安全论证, 快速发现需要进一步关注的问题. 同时, ASCE 也能有效减轻文档管理的负担, 确保引用最新、最相关或最合适的内容, 并突出显示底层文档的变化, 使用户能够轻松评估这些变化对安全案例的影响. 此外, ASCE 还提高了生产力, 支持跨组织协作, 改善企业记忆, 并降低采用成本. ASCE 与各种文档和证据库集成, 并可轻松整合流程. Adelard 为 ASCE 提供了完整的培训课程, 以满足初级和高级用户的需求. 总的来说, ASCE 是一款强大而实用的安全案例工具, 能够有效地满足安全案例的创建和管理需求.

尽管 ASCE 作为一款广泛采用的商业软件, 在创建和管理安全案例方面具有显著优势, 但根据文献 [45], 该工具在某些方面仍存在局限性. 首先, ASCE 的初始投资和许可成本可能较高, 尤其对于个人用户和小型组织来说, 可能会显得相对昂贵. 其次, ASCE 可能需要一定的学习曲线, 特别是对于那些没有 GSN 或安全案例背景的用户, 这可能增加了培训和学习成本. 此外, 尽管 ASCE 与许多文档和证据库集成良好, 但其兼容性可能受限于特定的文件格式或知识库系统, 对于使用其他文件管理工具的用户可能会存在一定的挑战. 在协作方面, 虽然 ASCE 支持跨组织协作, 但可能不如一些云原生 GSN 工具在实时协作和访问控制方面灵活. 最后, ASCE 的定制能力可能有限, 尤其是在评估方法和可视化方面, 这可能导致用户在特定场景下需要进行额外的定制开发. 总之, 文献指出 ASCE 在成本、学习曲线、兼容性、协作灵活性和定制能力方面的局限性可能对部分用户的使用体验产生影响.

3.6 D-Case Editor

D-Case[69]是一种基于 GSN 的扩展方法, 其研究团队作为 DEOS 项目[72]的一部分, 强调了可靠性 (dependability) 的方面, 致力于开发可靠的嵌入式系统. 该方法尝试将安全论证与系统生命周期中的相关文档 (如评估报告、会议记录和各种规范文件等) 联系起来, 以支持论证的共识和讨论. D-Case 与 GSN 都是用于表示和组织安全案例的方法, 但它们的侧重领域有所不同: GSN 主要关注目标结构表示法, 而 D-Case 则更侧重于可靠性和系统生命周期相关的文档.

D-Case Editor 是一款用于创建和编辑 D-Case 和 GSN 的图形编辑器, 它作为 Eclipse 插件实现, 基于 Eclipse GMF 框架. 这款编辑器的主要特点包括支持 GSN、提供 GSN 模式库功能、原型类型检查功能以及目标系统的监控等. 另一款基于 Web 浏览器的 D-Case 编辑器——D-Case Weaver, 则提供了在线创建、编辑和管理 D-Case 和 GSN 图的功能. 其功能包括创建 GSN 图并添加、修改、删除节点和链接, 还有将 D-Case 的子树模块化并添加到上级 D-Case 中, 向节点添加、修改或删除 D-Script 信息, 生成与 D-Case Editor 兼容的 XML 格式的 D-Case 等. D-Case Editor 和 D-Case Weaver 共同支持 DEOS 过程和 DEOS 架构, 为开发人员提供了灵活的 D-Case 和 GSN 图编辑和管理工具. 这两款工具相辅相成, 帮助用户在不同环境下更有效地处理和管理 D-Case 和 GSN 图.

尽管 D-Case Editor 和 D-Case Weaver 为处理和管理 D-Case 和 GSN 图提供了灵活的工具, 但是这两款工具存

在一些局限性. 首先, 它们的学习曲线可能对于缺乏 GSN 或依赖性背景的用户来说相对较陡, 这可能导致用户需要花费更多的时间和精力来熟练掌握这些工具. 其次, 由于 D-Case Editor 作为 Eclipse 插件实现, 它的使用受限于 Eclipse 环境, 这可能影响到习惯于其他集成开发环境的用户. 此外, D-Case Editor 和 D-Case Weaver 在可扩展性和集成性方面可能相对较弱, 与其他案例管理工具或证据库的集成可能需要额外的开发工作. 尽管 D-Case Weaver 提供了在线编辑和管理 D-Case 和 GSN 图的便利, 但其性能和响应速度可能较慢, 尤其是在处理大型论证时. 最后, 这两款工具可能缺乏强大的社区支持和持续软件更新, 这导致用户在遇到问题时难以寻求解决方案, 并可能影响到工具在新技术和标准面前的适应性. 综上所述, 虽然 D-Case Editor 和 D-Case Weaver 具有一定的优势, 但它们在学习曲线、环境依赖、可扩展性、集成性、在线编辑器性能以及社区支持和软件更新方面的局限性需要用户充分考虑.

3.7 AdvoCATE

AdvoCATE (assurance case automation toolset)[70]是美国国家航空航天局 (NASA) 开发的一款支持 GSN 的工具, 其适用性广泛, 包括核电、道路与铁路交通、国防、医疗设备等安全关键领域, 尤其在航空系统中有显著应用. AdvoCATE 基于形式化的基础, 强调在安全案例的开发与管理过程中自动化生产的重要性. AdvoCATE 的特色包括支持 GSN、手动创建和编辑使用 GSN 的安全论证, 并提供了用户可自定义的元数据. 此外, 它还支持使用模块和层次结构对论证进行组织, 与形式化方法集成, 并能将手动创建的和自动生成的安全案例片段组装在一起. AdvoCATE 还能通过参数实例化论证模式实现论证的半自动创建, 并能计算各种论证指标, 提供了逻辑查询功能. AdvoCATE 2.0 是一款基于 Eclipse 的应用程序, 相比于 AdvoCATE 1.0, 其覆盖了更广泛的论证活动. 除了支持创建安全案例, AdvoCATE 2.0 还能帮助用户更广泛地组织项目保障活动, 基于一个集成的安全模型, 将危险分析、需求、结构化论证、屏障模型 (蝴蝶结图) 和验证工件相结合. AdvoCATE 1.0 的所有功能都可以在 AdvoCATE 2.0 中使用, 部分功能甚至经过了全新的重构. 综上所述, AdvoCATE 作为一款功能强大的 GSN 工具, 其在安全关键领域的广泛应用性, 以及其对开发和管理安全案例的有效性, 都提高了系统的安全性和可靠性.

尽管 AdvoCATE 是一款功能强大且适用于广泛安全关键领域的 GSN 工具, 但它仍存在一些缺点. 首先, AdvoCATE 的学习曲线可能对于新手用户较为陡峭, 这意味着掌握该工具可能需要额外的时间和培训资源. 此外, 作为基于 Eclipse 的应用程序, AdvoCATE 2.0 的使用局限于 Eclipse 环境, 这可能对习惯使用其他集成开发环境的用户造成不便. 一个潜在的缺点, AdvoCATE 在与其他工具和证据库的集成方面可能存在局限性, 这导致用户在实际应用过程中需要进行额外的配置和集成工作. 另一个潜在的缺点是 AdvoCATE 的可定制性和扩展性相对较弱, 可能无法完全满足特定行业或组织的需求. 最后, 由于 AdvoCATE 是由美国国家航空航天局 (NASA) 开发的工具, 其社区支持和软件更新可能受到限制, 这直接影响用户在解决问题和适应新技术或标准方面的能力. 总之, 尽管 AdvoCATE 在安全关键领域具有优势, 但用户在选择使用该工具时应充分考虑其学习曲线、环境依赖、集成能力、可定制性以及社区支持和软件更新等方面的局限性.

3.8 安全案例工具总结

本节介绍了安全案例的 7 个创建工具, 包括 astah system safety、gsn2x、NOR-STA、Socrates、ASCE、 D-Case Editor 和 AdvoCATE. 表 4 总结了这些工具的优缺点.

表 4 安全案例工具优缺点总结

工具名称	优点	缺点
astah system safety	支持多种建模语言、安装简单、操作直观	学习曲线陡峭、实时协作能力较弱
gsn2x	开源免费、支持纯文本创建	没有图形化操作界面
NOR-STA	适合大型项目、多人合作开发	运行速度慢、操作复杂
Socrates	适合大型项目、多人合作开发	不能显示完整图形、操作复杂
ASCE	支持最新的安全案例技术	试用申请周期长
D-Case Editor	开源免费	缺乏持续软件更新
AdvoCATE	评估功能强、支持参数化创建案例	难以获取使用授权、学习曲线陡峭

在研究了各种安全案例工具之后, 我们发现 astah system safety 和 ASCE 是功能丰富的商业桌面工具, 它们均提供试用许可, 前者的试用期为 42 天 (申请后几分钟内可获得), 后者则在提交相关项目材料后提供 30 天的企业试用许可或 12 个月的学术试用许可 (通常需要花费几天的时间来完成申请过程).

而 D-Case Editor 和 AdvoCATE 则是基于 Eclipse 环境的非商业桌面工具. D-Case Editor 相对易于获取且完全免费, 但缺乏持续的软件更新; 而 AdvoCATE 由美国航空航天局 (NASA) 开发, 具有强大的功能并公开了大量文档, 但获取软件包较为困难.

NOR-STA 和 Socrates 是基于 Web 的在线平台, 它们支持多人协作, 非常适合应用于大型项目. 在性能方面, Socrates 的运行速度优于 NOR-STA, 但 Socrates 不能显示完整的 GSN 图形.

此外, gsn2x 是一款轻量级的开源软件, 用户可以通过 YAML 文本描述 GSN, 然后生成 SVG 格式的图形.

总的来说, 选择合适的安全案例工具需要综合考虑项目需求、团队熟悉程度以及工具的优缺点等多个因素.

4 安全案例构建中的挑战

在前面的章节中, 我们详细探讨了安全案例的不同表现形式和常用工具, 这为安全案例的构建和评估提供了基础性的理解. 然而, 即使是最先进的工具和方法也不能完全消除在实际应用中所面临的多种挑战. 这些挑战不仅来自技术层面, 还涉及管理、人因工程、监管环境以及多学科合作等多个方面.

本节旨在深入探讨这些挑战, 为读者提供一个全面的视角来理解安全案例构建的复杂性. 我们将从数据的可靠性和完整性、复杂性和不确定性的管理、监管和标准的不一致、人因工程、技术的快速发展, 以及团队和跨学科合作等几个关键方面进行分析, 试图给出针对这些问题的潜在解决方案.

这些挑战的全面理解不仅对安全案例的有效构建至关重要, 而且能够提供关于如何优化和改进现有安全管理实践的有价值的洞见. 因此, 本节旨在作为一个桥梁, 连接理论和实践, 以助于构建更为全面和可靠的安全案例.

4.1 数据的可靠性和完整性

● 挑战: 构建任何可靠的安全案例的基石是其依赖的证据质量. 这些证据通常以多种形式出现, 例如测试数据、历史记录和第三方评估. 然而, 在动态和复杂系统的环境中, 确保这些数据集的可靠性和完整性是一项复杂的任务.

● 原因: 这一挑战的复杂性来自多个因素. 例如, 数据源可能是异构的, 分布在不同的平台或地理位置上, 或者甚至由多个利益相关方拥有. 这样的多样性通常使审计和验证过程变得复杂. 现代系统的动态性 (组件可以被更新、替换、甚至失败) 为维护数据完整性增加了另一层困难.

● 潜在解决方案

(1) 数据审计和质量检查: 必须实施定期的审计和质量检查. 自动化工具在执行这些检查方面可以发挥关键作用, 从而标记需要进一步调查的异常或不一致.

(2) 加密和访问控制: 应通过加密和严格的访问控制措施来保护数据, 以防止未经授权的篡改.

(3) 来源跟踪: 为所有数据实施严格的来源跟踪和标记机制, 确保每一条信息的来源都是已知和经过验证的. 这对来自第三方或开源存储库的数据尤为重要.

通过系统地应用这些解决方案, 可以有效地缓解与安全案例中数据可靠性和完整性有关的挑战, 从而提高安全评估的健壮性和可信度.

4.2 复杂性和不确定性的管理

● 挑战: 现代复杂系统, 特别是在自动驾驶和医疗领域系统中, 通常具有高度的复杂性和内在的不确定性. 这些因素使得构建全面和可信的安全案例更加困难.

● 原因: 对这种复杂性和不确定性进行管理和量化是一项极具挑战性的任务. 首先, 复杂性通常意味着系统有多个交互的组件, 它们可能以复杂和不可预测的方式行为. 其次, 不确定性可能来自多个源, 包括但不限于环境因素、用户行为和系统本身的内在变异性. 解决这些问题通常需要应用高级数学模型和算法, 例如概率论和不确定

性量化, 这不仅需要高度专业的技术知识, 还需要大量的计算资源.

● 潜在解决方案

(1) 模型验证: 使用高级数学和统计方法, 如蒙特卡洛模拟[73]或贝叶斯网络[74], 来建立和验证系统模型.

(2) 敏感性分析: 在进行安全案例分析时, 包括对不确定性和敏感性的全面分析, 以识别可能影响系统安全性的关键变量.

(3) 适应性设计: 构建能够适应不确定条件的安全控制和响应机制. 例如, 使用自适应控制算法来调整系统在面临不确定输入或条件时的行为.

通过综合应用这些策略, 我们有可能更有效地管理系统的复杂性和不确定性, 从而提高安全案例的可靠性和可信度.

4.3 监管和标准的不一致

● 挑战: 在全球化日益增强的当代, 企业和组织常常面临多重监管环境. 不同的行业和国家有其独特的系统安全标准和监管规定. 这一多样性不仅加剧了构建统一、高效安全案例的复杂性, 也可能导致相互矛盾或冲突的要求.

● 原因: 这种复杂性主要来自各种监管体系和标准的不一致性. 在全球化的背景下, 特别是跨国公司, 可能需要遵循不仅是一个国家或一个行业的规定. 每个规定都有其特定的要求和约束, 满足所有这些要求往往需要付出额外的努力和资源.

● 潜在解决方案

(1) 跨行业合作: 积极与多个行业和监管机构沟通和合作, 以期达到一个统一或至少是相容的安全标准.

(2) 可定制的安全模板: 设计和创建灵活、可定制的安全案例模板, 以便在不同的监管环境中进行快速调整和应用.

(3) 合规性检查工具: 开发或引入自动化工具, 用于检查和验证安全案例是否满足各种不同的国际和行业标准, 从而提高其适应性和灵活性.

通过这些潜在解决方案, 可以一定程度上缓解因监管和标准不一致而带来的挑战, 提高安全案例构建的可行性和有效性.

4.4 人因工程

● 挑战: 人因工程在构建安全案例时扮演着至关重要的角色. 在很多系统, 特别是高度交互式和高度依赖用户输入的系统 (如医疗、交通、工业控制等) 中, 人的行为和决策往往是影响系统安全性的关键因素.

● 原因: 这一挑战的根本原因在于人类行为的多变性和不可预测性. 人们的决策受到多种因素的影响, 包括但不限于个人经验、情感、认知偏见等, 这些因素在传统的安全模型和分析方法中往往被忽略或简化. 由于这些复杂性, 构建一个全面且可信赖的安全案例将变得更加困难.

● 潜在解决方案

(1) 行为建模: 为更准确地评估人因工程对系统安全的影响, 应该将人类因素和行为模型集成到安全案例中. 这可能涉及使用心理学、社会学和认知科学等多学科的研究成果.

(2) 用户培训和教育: 通过对系统用户进行持续的安全意识培训, 可以在一定程度上减少因误操作或不当决策导致的安全问题.

(3) 人机界面优化: 设计直观且易于理解的用户界面, 以减少操作错误和提高系统的整体安全性.

通过这些潜在解决方案, 我们不仅可以提高安全案例的质量, 还可以更全面地考虑到人的行为和决策在系统安全中的角色, 从而提供更为全面和可靠的安全评估.

4.5 技术的快速发展

● 挑战: 在当前的技术环境中, 新技术如人工智能、物联网、区块链等正在迅速地改变我们的生活和工作方式. 这些技术不仅带来了便利, 还引入了新的安全需求和挑战. 因此, 在这种环境下构建一个全面和可靠的安全案例变得尤为重要.

● 原因: 技术快速发展的核心挑战在于其不断变化的性质, 这直接影响了安全案例的有效性. 传统的安全模型

可能很难适应或预测新技术可能带来的风险, 因此必须不断地更新和修改安全案例以保持其相关性和有效性.

● 潜在解决方案

(1) 持续更新: 一种可能的解决方案是确保安全案例具有一定的灵活性, 使其可以容易地进行更新和修改以适应新技术和环境. 这可能需要一个持续的审查和改进过程.

(2) 前瞻性评估: 对新技术进行早期的安全性评估, 以便更好地了解潜在的风险和挑战. 这种早期评估可以帮助组织更加有针对性地准备和规划.

(3) 灵活的架构设计: 考虑到新技术的不断出现, 设计一个具有模块化和可插拔组件的安全案例架构会是非常有用的. 这样不仅可以方便地添加新的安全措施, 还可以在不影响整体安全性的情况下, 容易地移除或替换过时或无效的措施.

通过这些潜在的解决方案, 安全案例将能更有效地适应快速发展的技术环境, 从而提供一个更为全面和可靠的安全评估.

4.6 团队和跨学科合作

● 挑战: 在构建安全案例的过程中, 往往需要各种学科和专业领域的专家共同参与. 这不仅涉及技术层面的挑战, 还包括如何实现有效的组织协作和沟通, 以确保安全案例的全面性和可靠性.

● 原因: 跨学科合作的挑战主要源于两个方面: 一是不同学科和专业领域的知识和方法论可能存在巨大的差异, 这可能导致沟通和理解的难度增加; 二是组织和沟通的复杂性, 特别是在大型和复杂项目中, 团队成员需要清晰地了解自己的角色和责任, 以及如何与其他团队或部门有效地协作.

● 潜在解决方案

(1) 团队培训和发展: 为了促进有效的跨学科合作, 组织应投资于团队培训和团队建设活动. 这可以包括跨学科的培训课程, 以及专门针对项目管理和沟通技巧的培训.

(2) 通信平台: 使用高效的项目管理和通信工具可以极大地提高跨团队协作的效率. 这样的工具应支持实时更新和多方参与, 以确保信息流通的及时性和准确性.

(3) 明确角色和责任: 在项目开始阶段, 应明确每个团队成员的角色和责任. 这不仅有助于减少误解和冲突, 还可以确保每个团队成员都能明确自己在整个项目中的位置和任务.

通过实施这些潜在解决方案, 跨学科团队可以更有效地协作, 从而提高安全案例构建的质量和可靠性.

本节从多个角度讨论了构建全面和可靠安全案例所面临的挑战, 包括数据的可靠性和完整性、复杂性和不确定性的管理、监管和标准的不一致、人因工程、技术的快速发展, 以及团队和跨学科合作. 这些挑战不仅反映了现代系统安全需求的多样性和复杂性, 也突显了需要综合多学科知识和技术来应对这些挑战.

虽然每个挑战都有其特定的解决建议, 但值得注意的是, 这些解决方案往往需要跨学科和跨部门的紧密合作才能实施成功. 因此, 组织和个体不仅需要关注单一的技术或管理问题, 还需要采取一种更全面、更协同的方法来构建和维护安全案例.

通过对这些挑战及其潜在解决方案的深入分析, 我们可以更好地理解和评估安全案例的复杂性和多维性. 这为进一步的研究和实践提供了有价值的洞见, 也为提高现有安全案例的质量和效率铺平了道路. 总体而言, 构建一个全面和可靠的安全案例是一个需要多方共同努力和持续改进的过程, 而对这一过程的深入理解无疑将有助于我们在未来更好地保障系统和信息的安全.

5 未来研究方向

随着技术的发展和安全需求的日益提高, 安全案例的重要性将继续增加. 我们期待未来的研究能够进一步深入理解和发展安全案例的理论和实践, 包括如下几个方面.

(1) 自动化和半自动化的安全案例构建. 目前, 安全案例的构建主要依赖于人工, 这在一定程度上限制了其效率和一致性. 未来的研究可以探索如何利用自然语言处理、机器学习等技术自动或半自动地构建安全案例.

(2) 安全案例的可视化和交互性. 安全案例的表现形式对于理解和沟通安全案例至关重要. 未来的研究可以研究如何通过更先进的可视化和交互设计来提高安全案例的可理解性和可操作性.

(3) 安全案例的标准化. 尽管已经有了 GSN、CAE、SACM 等标准表现形式, 但在实践中, 不同的组织和项目可能会有不同的标准和方法. 未来的研究可以进一步探索和推广安全案例的标准化, 以促进安全案例在不同场景的通用性和互操作性.

(4) 安全案例的评估和验证. 目前, 安全案例的评估主要依赖于人工, 这可能会受到主观性的影响. 未来的研究可以探索如何通过定量或半定量的方法来评估和验证安全案例, 以提高其公正性和准确性.

(5) 安全案例在新领域的应用. 随着新技术的发展, 例如自动驾驶、区块链、人工智能等, 安全案例的应用场景也将继续扩大. 未来的研究可以探索如何将安全案例应用到这些新的领域, 以应对新的安全挑战.

(6) 开发更通用的安全案例工具. 尽管当前已有一些工具支持安全案例的构建和评估, 但这些工具可能并不满足所有场景的需求或缺乏某些功能. 未来的研究可以着重于设计和开发更通用、更灵活且易于集成的安全案例工具, 以适应不同的应用场景和需求, 同时能够与其他工具和平台进行有效的互操作.

总的来说, 安全案例是一个重要而复杂的研究领域, 它需要多学科的知识和技能, 包括系统工程、风险管理、计算机科学、人工智能等. 我们期待未来的研究能够进一步推动安全案例的发展, 以更好地保障我们的生活和工作.

6 总 结

本文对安全案例的基本概念、构建过程、表现形式和工具进行了深入的分析和讨论. 首先, 我们理解了安全案例的定义和目的, 即通过系统地收集、组织和评估证据来证明系统的安全性. 然后, 我们介绍了创建安全案例的 4 个步骤: 确定目标、收集证据、构建论证和评估安全案例. 这 4 步是创建安全案例的基本框架, 但在实际操作中, 这些步骤可能会反复进行以适应项目的发展和需求变化. 在介绍了如何创建安全案例之后, 我们讨论了 8 种安全案例的表现形式, 包括自然语言、表格结构、断言结构、可追溯性矩阵、贝叶斯可信度网络、目标结构符号 (GSN)、声明-论点-证据 (CAE) 和结构化安全案例元模型 (SACM). 每种表现形式都有其特点和适用场景, 因此, 在选择表现形式时, 需要根据项目需求和团队熟悉程度进行权衡. 最后, 我们介绍了 7 种安全案例工具, 包括 astah system safety、gsn2x、NOR-STA、Socrates、ASCE、D-Case Editor 和 AdvoCATE. 这些工具各具特色, 适用于不同的应用场景. 在选择工具时, 也需要根据项目需求、团队熟悉程度以及工具的优缺点进行选择. 此外, 本文还深入探讨了安全案例构建中面临的多重挑战. 这些挑战包括但不限于数据的可靠性和完整性、复杂性和不确定性的管理、与监管和标准的一致性、人因工程的影响、技术的快速发展以及团队和跨学科合作的复杂性. 这些挑战不仅增加了安全案例构建的复杂性, 还提出了新的研究和应用问题, 值得进一步的探究和解决.

References:

[1] Toulmin SE. The Uses of Argument. Cambridge: Cambridge University Press, 2003. 1–247.

[2] Cleland G, Sujan MA, Habli I, Medhurst J. Using safety cases in industry and healthcare. London: The Health Foundation, 2012. https://www.health.org.uk/publications/using-safety-cases-in-industry-and-healthcare

[3] Sklyar V, Kharchenko V. Assurance case for safety and security implementation: A survey of applications. Int'l Journal of Computing, 2020, 19(4): 610–619. [doi: 10.47839/ijc.19.4.1995]

[4] Bate IJ, Burns A, Kelly TP, McDermid JA. Building a preliminary safety case: An example from aerospace. In: Proc. of the 1997 Australian Workshop on Industrial Experience with Safety Critical Systems and Software. Sydney, 1997. 1–10. https://pure.york.ac.uk/portal/en/publications/building-a-preliminary-safety-case-an-example-from-aerospace

[5] Rushby J, Xu XD, Rangarajan M, Weaver TL. Understanding and evaluating assurance cases. Technical Report 20160000772, Hampton: NASA Langley Research Center, 2015.

[6] Graydon PJ, Knight JC, Strunk EA. Assurance based development of critical systems. In: Proc. of the 37th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks. Edinburgh: IEEE, 2007. 347–357. [doi: 10.1109/DSN.2007.17]

[7] Austin R, Mahadevan N, Sierawski BD, Karsai G, Witulski AF, Evans J. A CubeSat-payload radiation-reliability assurance case using goal structuring notation. In: Proc. of the 2017 Annual Reliability and Maintainability Symp. Orlando: IEEE, 2017. 1–8. [doi: 10.1109/

RAM.2017.7889672]

[8] Vierhauser M, Bayley S, Wyngaard J, Xiong WD, Cheng JH, Huseman J, Lutz R, Cleland-Huang J. Interlocking safety cases for unmanned autonomous systems in shared airspaces. IEEE Trans. on Software Engineering, 2021, 47(5): 899–918. [doi: 10.1109/TSE.2019.2907595]

[9] Niu R, Tang T. Application of safety argument in safety assurance system for railway signalling development. Journal of the China Railway Society, 2014, 36(4): 54–59 (in Chinese with English abstract). [doi: 10.3969/j.issn.1001-8360.2014.04.010]

[10] Xu ZJ, Wang Q. Application of GSN safety demonstration method in change management of product safety case. Control and Information Technology, 2020(2): 95–99 (in Chinese with English abstract). [doi: 10.13889/j.issn.2096-5427.2020.02.018]

[11] Medhurst J, Embrey D. Safety case use in the railway industry. In: Supplements to: Using Safety Cases in Industry and Healthcare. London: The Health Foundation, 2012.

[12] Beugin J, Legrand C, Marais J, Berbineau M, El-Koursi EM. Safety appraisal of GNSS-based localization systems used in train spacing control. IEEE Access, 2018, 6: 9898–9916. [doi: 10.1109/ACCESS.2018.2807127]

[13] Griessnig G, Schnellbach A. Development of the 2nd Edition of the ISO 26262. In: Proc. of the 24th European Conf. on Software Process Improvement. Ostrava: Springer, 2017. 535–546. [doi: 10.1007/978-3-319-64218-5_44]

[14] Palin R, Habli I. Assurance of automotive safety—A safety case approach. In: Proc. of the 29th Int'l Conf. on Computer Safety, Reliability, and Security. Vienna: Springer, 2010. 82–96. [doi: 10.1007/978-3-642-15651-9_7]

[15] Bourbouh H, Farrell M, Mavridou A, Sljivo I, Brat G, Dennis LA, Fisher M. Integrating formal verification and assurance: An inspection rover case study. In: Proc. of the 13th NASA Formal Methods. Springer, 2021. 53–71. [doi: 10.1007/978-3-030-76384-8_4]

[16] Bloomfield R, Chozos N, Cleland G, Adelard LLP. Safety case use within the medical devices industry. In: Supplement to: Using Safety Cases in Industry and Healthcare. London: The Health Foundation, 2012.

[17] Larson BR, Hatcliff J, Chalin P. Open source patient-controlled analgesic pump requirements documentation. In: Proc. of the 5th Int'l Workshop on Software Engineering in Health Care. San Francisco: IEEE, 2013. 28–34. [doi: 10.1109/SEHC.2013.6602474]

[18] Jee E, Lee I, Sokolsky O. Assurance cases in model-driven development of the pacemaker software. In: Proc. of the 4th Int'l Symp. on Leveraging Applications of Formal Methods, Verification and Validation. Heraklion: Springer, 2010. 343–356. [doi: 10.1007/978-3-642-16561-0_33]

[19] Bloomfield R, Bishop P. Safety and assurance cases: Past, present and possible future—An Adelard perspective. In: Dale C, Anderson T, eds. Making Systems Safer. London: Springer, 2009. 51–67. [doi: 10.1007/978-1-84996-086-1_4]

[20] Leveson NG. The use of safety cases in certification and regulation. Technical Report, ESD-WP-2011-13, Engineering Systems Division, Massachusetts Institute of Technology, 2011. https://dspace.mit.edu/handle/1721.1/102833

[21] Wassyng A, Maibaum T, Lawford M, Bherer H. Software certification: Is there a case against safety cases? In: Proc. of the 16th Monterey Workshop: Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems. Redmond: Springer, 2011. 206–227. [doi: 10.1007/978-3-642-21292-5_12]

[22] Henderson J. Safety case use in the petrochemical industry. In: Supplements to: Using Safety Cases in Industry and Healthcare. London: The Health Foundation, 2012.

[23] Baram MS. Preventing accidents in offshore oil and gas operations: The U.S. approach and some contrasting features of the Norwegian approach. Technical Report, 09-43, School of Law, Boston University, 2010.

[24] Mendes PAS, Hall J, Matos S, Silvestre B. Reforming Brazil's offshore oil and gas safety regulatory framework: Lessons from Norway, the United Kingdom and the United States. Energy Policy, 2014, 74: 443–453. [doi: 10.1016/j.enpol.2014.08.014]

[25] Kelly T. Safety case use in the defence industry. In: Supplements to: Using Safety Cases in Industry and Healthcare. London: The Health Foundation, 2012. 19–23.

[26] Duncan B, Whittington M. Compliance with standards, assurance and audit: Does this equal security? In: Proc. of the 7th Int'l Conf. on Security of Information and Networks. Glasgow: ACM, 2014. 77–84. [doi: 10.1145/2659651.2659711]

[27] Bloomfield R, Bishop P, Butler E, Netkachova K. Using an assurance case framework to develop security strategy and policies. In: Proc. of the 2017 Int'l Conf. on Computer Safety, Reliability, and Security. Trento: Springer, 2017. 27–38. [doi: 10.1007/978-3-319-66284-8_3]

[28] Widowati E, Sutomo AH, Istiono W. Are elementary schools ready for disaster preparedness and safety? E3S Web of Conf., 2021, 317: 01087. [doi: 10.1051/e3sconf/202131701087]

[29] Habli I, Alexander R, Hawkins R, Sujan M, McDermid J, Picardi C, Lawton T. Enhancing COVID-19 decision-making by creating an assurance case for simulation models. arXiv:2005.08381, 2020.

[30] National Research Council. Software for Dependable Systems: Sufficient Evidence? Washington: National Academies Press, 2007. [doi:

10.17226/11923]
[31] ISO/IEC 1502-2:2011 Systems and software engineering-systems and software assurance-part 2: Assurance case. 2011. https://www.iso.org/standard/52926.html
[32] IEC 61508:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems. 2010. https://webstore.iec.ch/publication/22273
[33] ISO 26262-1:2018 Road vehicles-functional safety part 1: Vocabulary. 2018. https://www.iso.org/standard/68383.html
[34] EN 50657 Railway applications—Rolling stock applications—Software on board rolling stock. 2017. https://verifysoft.com/en_EN_50657_Railway_Applications.html
[35] DO-178C Software considerations in airborne systems and equipment certification. 2011. https://www.do178.org/
[36] Menon CI, Hawkins R, McDermid J. Defence standard 00–56 issue 4: Towards evidence-based safety standards. In: Dale C, Anderson T, eds. Safety-critical Systems: Problems, Process and Practice. London: Springer, 2009. 223–243. [doi: 10.1007/978-1-84882-349-5_15]
[37] Gallina B, Gómez-Martínez E, Earle CB. Deriving safety case fragments for assessing MBASafe's compliance with EN 50128. In: Proc. of the 16th Int'l Conf. on Software Process Improvement and Capability Determination. Dublin: Springer, 2016. 3–16. [doi: 10.1007/978-3-319-38980-6_1]
[38] Holloway CM. Making the implicit explicit: Towards an assurance case for DO-178C. Technical Report, NF1676L-16361, NASA, 2013. https://ntrs.nasa.gov/citations/20140002745
[39] Kelly T. Arguing safety: A systematic approach to managing safety cases [Ph.D. Thesis]. Heslington: University of York, 1998.
[40] Kelly TP. A systematic approach to safety case management. Technical Paper, 2004-01-1779, SAE Int'l. 2004. [doi: 10.4271/2004-01-1779]
[41] Kelly T, Weaver R. The goal structuring notation—A safety argument notation. In: Proc. of the 2004 Dependable Systems and Networks Workshop on Assurance Cases. 2004. https://www.researchgate.net/profile/Tim-Kelly-10/publication/228990118_The_goal_structuring_notation-a_safety_argument_notation/links/00b7d51b58537a2fef000000/The-goal-structuring-notation-a-safety-argument-notation.pdf
[42] Fithri P, Riva NA, Susanti L, Yuliandra B. Safety analysis at weaving department of PT. X Bogor using failure mode and effect analysis (FMEA) and fault tree analysis (FTA). In: Proc. of the 5th Int'l Conf. on Industrial Engineering and Applications. Singapore: IEEE, 2018. 382–385. [doi: 10.1109/IEA.2018.8387129]
[43] Bloomfield R, Rushby J. Assurance 2.0: A manifesto. arXiv:2004.10474, 2020.
[44] The Assurance Case Working Group (ACWG). Goal structuring notation community standard version 3. 2021. https://scsc.uk/r141C:1?t=1
[45] Netkachova K, Netkachov O, Bloomfield R. Tool support for assurance case building blocks. In: Proc. of the 2015 Int'l Conf. on Computer Safety, Reliability, and Security. Springer, 2015. 62–71. [doi: 10.1007/978-3-319-24249-1_6]
[46] Bloomfield RE, Netkachova K. Building Blocks for Assurance Cases. In: Proc. of the 25th Int'l Symp. on Software Reliability Engineering Workshops. IEEE Computer Society, 2014. 186–191. [doi: 10.1109/ISSREW.2014.72]
[47] Wei R, Kelly TP, Dai XT, Zhao S, Hawkins R. Model based system assurance using the structured assurance case metamodel. Journal of Systems and Software, 2019, 154: 211–233. [doi: 10.1016/j.jss.2019.05.013]
[48] Bishop PG, Bloomfield RE. The ship safety case approach: A combination of system and software methods. In: Proc. of the 12th Annual CSR Workshop, Safety and Reliability of Software Based Systems. London: Springer, 1997. 107–121. [doi:10.1007/978-1-4471-0921-1_4]
[49] Standard D. Requirements for safety related software in defence equipment part 2: Guidance. Ministry of Defence, 1997. https://www.software-supportability.org/Docs/00-55_Part_2.pdf
[50] Shu YD, Zhao JS. A simplified Markov-based approach for safety integrity level verification. Journal of Loss Prevention in the Process Industries, 2014, 29: 262–266. [doi: 10.1016/j.jlp.2014.03.013]
[51] Madan M, Dave M, Tandon A. Need and usage of traceability matrix for managing requirements. Int'l Journal of Engineering Research, 2016, 5(8): 666–668. [doi: 10.17950/ijer/v5s8/805]
[52] ASCAS Manual. The adelard safety case development (ASCAD) manual. 1998. https://www.adelard.com/resources/ascad-manual/
[53] Fenton N. The role of measurement in software safety assessment. In: Proc. of the 12th Annual CSR Workshop, Safety and Reliability of Software Based Systems. London: Springer, 1997. 217–248. [doi: 10.1007/978-1-4471-0921-1_11]
[54] Wilson SP, Kelly TP, McDermid JA. Safety case development: Current practice, future prospects. In: Proc. of the 12th Annual CSR Workshop, Safety and Reliability of Software Based Systems. London: Springer, 1997. 135–156. [doi: 10.1007/978-1-4471-0921-1_6]
[55] Wilson SP, McDermid JA. Integrated analysis of complex safety critical systems. The Computer Journal, 1995, 38(10): 765–776. [doi: 10.

1093/comjnl/38.10.765]
[56] Support for GSN and ISO 15026 assurance cases. 2022. https://www.argevide.com/2022-06_release_7_8/
[57] Maksimov M, Fung NLS, Kokaly S, Chechik M. Two decades of assurance case tools: A survey. In: Proc. of the 2018 Int'l Conf. on Computer Safety, Reliability, and Security. Västerås: Springer, 2018. 49–59. [doi: 10.1007/978-3-319-99229-7_6]
[58] Selviandro N. Assurance case pattern using SACM notation. In: Proc. of the 9th Int'l Conf. on Information and Communication Technology. Yogyakarta: IEEE, 2021. 494–499. [doi: 10.1109/ICoICT52021.2021.9527483]
[59] Structured assurance case metamodel (SACM). 2020. https://www.omg.org/spec/SACM/2.1/PDF
[60] Sutopo RA, Selviandro N, Wulandari GS. Analysis and implementation of Web-based graphic editor for structured assurance case metamodel notation. In: Proc. of the 1st Int'l Conf. on Software Engineering and Information Technology. Bandung: IEEE, 2022. 222–227. [doi: 10.1109/ICoSEIT55604.2022.10029970]
[61] Nemouchi Y, Foster S, Gleirscher M, Kelly T. Isabelle/SACM: Computer-assisted assurance cases with integrated formal methods. In: Proc. of the 15th Int'l Conf. on Integrated Formal Methods. Bergen: Springer, 2019. 379–398. [doi: 10.1007/978-3-030-34968-4_21]
[62] Maksimov M, Kokaly S, Chechik M. A survey of tool-supported assurance case assessment techniques. ACM Computing Surveys, 2020, 52(5): 101. [doi: 10.1145/3342481]
[63] Goal structuring notation tools. 2022. https://scsc.uk/gsn?page=gsn%206tools
[64] astah. Download astah software. 2023. https://astah.net/downloads/
[65] gsn2x. Tool to create graphical representations of goal structuring notations from YAML. 2023. https://github.com/jonasthewolf/gsn2x
[66] Argevide. Develop assurance case online with NOR-STA. 2023. https://www.argevide.com/assurance-case/
[67] Socrates. 2023. https://criticalsystemslabs.com/socrates/
[68] Adelard. ASCE software overview. 2023. https://www.adelard.com/asce/
[69] DEOS. D-Case Editor—A typed assurance case editor. 2023. https://www.jst.go.jp/crest/crest-os/tech/D-CaseEditor/index-e.html
[70] NTRS. AdvoCATE user guide. 2023. https://ntrs.nasa.gov/citations/20220009664
[71] Ishimatsu T, Leveson NG, Thomas J, Katahira M, Miyamoto Y, Nakao H. Modeling and hazard analysis using STPA. Int'l Association for the Advancement of Space Safety, 2010.
[72] DEOS. Welcome to dependability engineering for open systems. 2023. https://www.jst.go.jp/crest/crest-os/osddeos/index-e.html
[73] Harrison RL. Introduction to Monte Carlo simulation. AIP Conf. Proc., 2010, 1204(1): 17–21. [doi: 10.1063/1.3295638]
[74] Heckerman D. A tutorial on learning with Bayesian networks. In: Holmes DE, Jain LC, eds. Innovations in Bayesian Networks: Theory and Applications. Berlin, Heidelberg: Springer, 2008. 33–82. [doi: 10.1007/978-3-540-85066-3_3]

附中文参考文献:

[9] 牛儒, 唐涛. 安全论证方法及其在铁路信号开发安全保障中的应用. 铁道学报, 2014, 36(4): 54–59. [doi: 10.3969/j.issn.1001-8360.2014.04.010]
[10] 徐征捷, 王奇. GSN 安全论证方法在产品安全案例变更管理中的应用. 控制与信息技术, 2020(2): 95–99. [doi: 10.13889/j.issn.2096-5427.2020.02.018]

陈泽众(1995－), 男, 博士生, CCF 学生会员, 主要研究领域为软件工程, 形式化方法.

邓玉欣(1978－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为并发理论, 程序语义, 形式化验证, 量子计算.