软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(1):27−46 [doi: 10.13328/j.cnki.jos.007157] [CSTR: 32375.14.jos.007157]

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

嵌入式系统的需求描述综述*

陈小红[1], 刘少彬[1], 金 芝[2,3]

[1](上海市高可信计算重点实验室 (华东师范大学), 上海 200062)

[2](高可信软件技术教育部重点实验室 (北京大学), 北京 100871)

[3](北京大学 计算机学院, 北京 100871)

通信作者: 金芝, E-mail: zhijin@pku.edu.cn

摘 要: 随着嵌入式系统的广泛应用, 其需求正变得越来越复杂, 需求分析成为嵌入式系统开发的关键阶段, 如何准确地建模和描述需求成为首要问题. 系统地调研嵌入式系统的需求描述, 并进行全面的比较分析, 以便更深入地理解嵌入式系统需求的核心关注点. 首先采用系统化文献综述方法, 对 1979 年 1 月–2023 年 11 月间发表的相关文献进行识别、筛选、汇总和分析. 通过自动检索和滚雪球等检索过程, 筛选出 150 篇与主题密切相关的文献, 力求文献综述的全面性. 其次, 从需求描述关注点、需求描述维度、需求分析要素等方面, 分析现有嵌入式需求描述语言的表达能力. 最后, 总结现有嵌入式系统软件需求描述所面临的挑战, 并针对嵌入式软件智能合成任务, 提出对嵌入式系统需求描述方法表达能力的要求.

关键词: 嵌入式系统; 需求描述; 需求描述语言; 需求分析; 系统需求

中图法分类号: TP311

中文引用格式: 陈小红, 刘少彬, 金芝. 嵌入式系统的需求描述综述. 软件学报, 2025, 36(1): 27–46. http://www.jos.org.cn/1000-9825/7157.htm

英文引用格式: Chen XH, Liu SB, Jin Z. Survey on Requirements Description of Embedded System. Ruan Jian Xue Bao/Journal of Software, 2025, 36(1): 27–46 (in Chinese). http://www.jos.org.cn/1000-9825/7157.htm

Survey on Requirements Description of Embedded System

CHEN Xiao-Hong[1], LIU Shao-Bin[1], JIN Zhi[2,3]

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

[2](Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

[3](School of Computer Science, Peking University, Beijing 100871, China)

Abstract: As embedded systems are widely applied, their requirements are becoming increasingly complex, making requirements analysis a critical stage in embedded system development. How to correctly describe and model requirements has become a primary issue. This study systematically investigates the current requirements descriptions of embedded systems and conducts a comprehensive comparative analysis to deepen the understanding of the core concerns of embedded system requirements. The study first applies the systematic literature review method to identify, retrieve, summarize, and analyze the relevant literature published between January 1979 and November 2023. Through the automatic retrieval and snowball processes, 150 papers closely related to the topic are finally selected for the comprehensiveness of the review. The study analyzes the existing capabilities of embedded requirements description languages from their description concerns, description contents, requirements analysis elements, etc. Finally, it summarizes the challenges to the current requirements descriptions. Moreover, aiming at the task of intelligent synthesis of embedded software, it puts forward the need for the expressive ability of embedded system requirement description languages.

Key words: embedded system; requirements description; requirements description language; requirements analysis; system requirements

* 基金项目: 国家自然科学基金 (62192731, 62272166, 62192730)
收稿时间: 2023-09-04; 修改时间: 2023-10-19; 采用时间: 2024-01-04; jos 在线出版时间: 2024-05-15
CNKI 网络首发时间: 2024-05-18

嵌入式系统已经渗透进人类生活的方方面面, 广泛应用于汽车电子、航空航天、轨道交通、医疗设备和个人移动设备等各个领域. 伴随而来的是日益增长的需求复杂性. 每个嵌入式系统都融合了大量的功能需求和实时、安全、可靠等非功能需求. 以高档汽车为例, 其内部就拥有超过 2 000 个基于软件的功能[1]; 而一个航空嵌入式系统, 也有多达 139 个功能[2]. 此外, 嵌入式系统的需求复杂度还体现在接入设备数量的急剧增长上. 据 Mumtaz 等人在文献 [3] 中指出, 未来交通、智慧城市和工厂等领域无线接入的设备数量将达到百亿级别.

对复杂软件系统开发而言, 需求阶段是最容易出错而且出错代价最高的阶段, 对嵌入式软件系统而言更是如此. Broy 等人发现, 嵌入式领域中有超过 50% 的错误发生在系统交付时, 而且与需求的错误理解有关[4]. Naumchev 等人指出导致软件灾难的需求问题可以分为两类, 一类是对需求的错误理解、更新和实现, 另一类是需求的不完整、不一致和未满足用户需要[5]. 准确地描述需求成为嵌入式系统开发的关键任务. 相较于传统的信息系统软件, 嵌入式系统需求有其特殊性. 首先, 嵌入式系统由软硬件组成, 其软件通过各类传感设备和作用设备与物理世界直接交互, 并按照系统控制逻辑在资源有限的硬件平台上运行. 因此, 其需求不仅包含软硬件行为, 还涵盖了对性能、精确度、可靠性等的约束, 其中一些是强制性需求. 这就要求我们在需求描述时, 准确地捕捉和表达这些约束. 其次, 嵌入式系统开发始于任务意图, 需求分析将任务意图逐步细化, 从系统级需求到软硬件相关需求, 需求分析需要将相互交织的需求清晰地解耦成相对简单的操作. 需求的不同层次之间的关系成为需求分析的重要内容. 如何表达各种层次需求则成为首要问题.

从 20 世纪 70 年代起, 研究者就开始关注嵌入式系统需求描述的问题, 提出了一系列需求描述语言, 从不同的视角设计了其语言表达能力. 在 20 世纪 80 年代, Davis 等人[6]和 Melhart 等人[7]分别综述了当时主流的嵌入式需求描述语言, 阐述了这些语言各自的背景、特征和应用等, 但其成文年代距今较为久远, 涵盖的语言较少, 并且当时的计算机和嵌入式技术远没有现如今成熟和发达, 连接的设备也很少. 随着嵌入式系统复杂度的与日俱增, 其需求不但错综复杂, 而且分散在软硬件开发的不同阶段, 因而又出现了不少新的嵌入式系统需求描述方法. 本文通过对这些已有工作的全面调研、比较与分析, 希望能深入理解嵌入式系统需求的核心关注点, 以及现有嵌入式需求描述语言的能力, 理清由于设备共享导致的嵌入式系统需求交织, 提炼需求描述的关键成分, 为提出具有一定通用性并能捕捉分散需求的需求描述以及需求分析打下基础.

本文采用系统化文献综述方法, 对 1979 年 1 月–2023 年 11 月发表的相关文献进行了检索、筛查, 最终从 3 442 篇文献中选出了 150 篇论文作为研究对象. 由于各文献中研究对象和术语存在差异, 同时考虑到嵌入式系统涉及的需求与其系统结构密切相关, 为了能统一表达各种类型、各层次需求的含义, 并便于进行比较分析, 我们在开始调研之前, 首先给出了一般嵌入式系统的参考结构, 作为调研分析的基础. 接下来从需求建模关注点、需求描述维度、需求分析要素等方面, 分析了现有嵌入式需求描述语言的表达能力. 最后, 讨论现有嵌入式系统软件需求描述面临的挑战, 并面向研究热点——嵌入式系统智能合成任务, 提出了其对嵌入式系统需求描述语言的表达能力要求.

本文第 1 节给出嵌入式系统的参考结构. 第 2 节给出所采用的系统文献综述方法, 并给出研究文献的总体情况. 第 3 节阐述嵌入式系统需求的不同描述类型. 第 4 节综述现有嵌入式需求描述语言的能力. 第 5 节讨论现有嵌入式系统软件需求描述的挑战, 结合嵌入式软件智能合成任务讨论需求描述语言表达能力要求. 最后, 第 6 节总结全文.

1 嵌入式系统参考结构

为了能在一个统一的视角下综述嵌入式系统需求描述, 我们给出了嵌入式系统的参考结构, 如图 1 所示. 其中, 嵌入式系统被视为具备特定功能的计算软硬件综合体, 软件作为协调系统设备完成嵌入式系统设计意图的控制器, 硬件设备主要包含各种传感器、执行器等系统可以管理和调度的设备. 控制器通过系统设备与外部环境进行通信, 这些外部环境包括超级管理员、外部软件系统、自然环境、物理环境等.

复杂嵌入式系统通常有多个软件控制器, 我们区分系统控制器和一组子系统控制器. 其中, 系统控制器可以接

收来自超级管理员的外部控制指令, 监视每个子系统控制器的状态, 并对子系统控制器进行控制协调. 子系统控制器则负责协调相应的传感器和控制器, 获取外部环境信息并实施控制操作. 传感器可以获取外部环境属性, 除了看作为外部设备, 也看成是系统接口. 执行器可以将软件控制器指令作用在外部环境上.

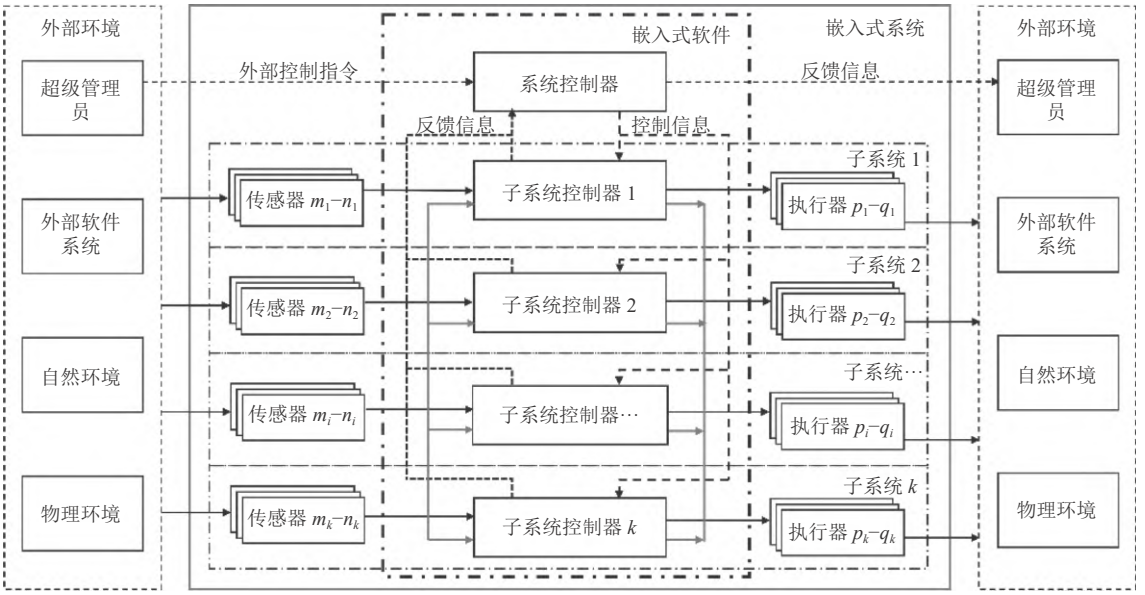

图 1 嵌入式系统参考结构

对上述嵌入式系统参考结构中不同组件的期望和要求, 会得到不同类型的需求描述. 这些类型包括系统任务意图、系统能力需求、软件能力需求、硬件需求以及软件设计约束. 其中, 系统任务意图表达了系统的目标, 由于嵌入式系统将要作用到外部环境上, 因此, 遵循基于环境建模的需求工程理念[8], 系统任务意图通常反映在对预期外部环境变化的响应上. 例如, 在航天领域中, 太阳搜索控制系统的任务意图是感知自然环境中太阳的位置以及物理环境中当前卫星的星体角度, 并调整自身的星体角以实现对日巡航. 在这个例子中, 太阳和星体都构成了外部环境. 系统能力需求描述了系统必须具备的能力, 这些能力是为了实现系统任务意图所必须满足的具体条件或要求. 具体来说, 由于嵌入式系统将要通过系统设备与外部环境进行交互, 其能力建立在嵌入式系统与外部环境的输入和输出关系之上, 即系统设备与外界环境的输入输出上. 例如太阳搜索控制系统中, 系统通过传感器陀螺、太阳敏感器测量卫星速度和角度, 感知太阳可见信息等.

软件能力需求则是指软件所需要具备的调度、管理、控制系统设备的能力, 以便满足系统能力需求. 它常表现为嵌入式软件控制器与系统设备之间的输入、输出及其关系上, 也被称为嵌入式软件需求, 包括各种功能需求和非功能需求. 例如太阳搜索控制软件需要向陀螺和太阳敏感器等设备发送电源控制指令和数据通信指令、接收并处理测量数据等, 这些都是功能需求; 太阳搜索控制软件向推力器写入控制命令的周期间隔必须小于 120 ms 等, 则属于非功能需求. 软件设计约束则是对软件控制器内部结构的组织和设计的约束条件, 例如太阳搜索控制软件中控制模式的计算方法对软件内部模块设计的约束. 硬件需求则是针对嵌入式硬件设备的约束, 可分为系统设备需求和软件运行平台需求. 系统设备需求主要涉及各种传感器和执行器, 例如传感器陀螺能测量卫星角速度, 传感器的每次数据采集必须能在 90 ms 内完成, 执行器的重启时间不能超过 500 ms 等约束; 软件运行平台需求指的是对支撑嵌入式系统软件运行的硬件平台的要求, 例如内存容量不低于 128 MB、存储空间不小于 1 GB、处理器每秒必须处理 500 万条指令等.

上述 5 种不同类型的需求描述, 实际上也是不同需求开发阶段的产物. 通常的需求开发流程为: 首先通过系统目标分析, 研究预期的外部环境变化, 得出系统的任务意图. 然后经过系统能力分析, 将任务意图具体化为对系统

与外部环境的输入输出关系, 导出系统能力需求. 接下来进行软硬件划分, 通过对软件与外部设备的交互以及对外部设备和平台的要求进行分析, 获取软件能力需求和硬件能力需求, 最后可以对软件的初步设计阶段得出软件设计约束. 总之, 嵌入式系统的软件需求规约是在这些过程中不断从对外部环境期望到软件控制器行为推导的过程, 其中涉及不同类型需求的精化、分解、推导与演化等.

2 文献综述方法

我们采用 Kitchenham 的指南[9]开展系统文献综述. 本节首先给出研究问题, 接下来介绍数据采集遵循的策略和流程, 最后介绍研究文献的总体情况, 包括出版年份、涉及的软件开发阶段和领域.

2.1 研究问题

本文的目的是深入理解嵌入式系统需求的核心关注点以及需求描述语言能力, 为此设计了两个研究问题.

RQ1: 嵌入式系统常见描述中涉及的需求类型有哪些?

RQ2: 现有嵌入式系统需求描述语言的能力如何?

2.2 文献收集

本节描述系统文献综述中的数据收集过程, 包括文献检索、文献筛选、滚雪球等. 本文的 3 个作者在自动和人工检索以及滚雪球之后, 最终从 3 442 篇文献中选出了 150 篇文献作为研究集合.

(1) 文献检索

我们首先考虑与需求工程、嵌入式系统、嵌入式软件、需求描述相关的关键字, 在搜索中又发现过程控制系统、实时系统也有很多是嵌入式系统, 因此又增加了相关的关键字. 最后确定的文献检索中文关键字为“(嵌入式系统软件 + 嵌入式系统 + 嵌入式软件 + 过程控制系统 + 实时控制系统) * (需求描述 + 需求规约 + 需求规范 + 需求建模 + 系统描述 + 系统规约 + 系统规范 + 系统建模 + 需求标准 + 需求格式 + 需求文档)”, 英文关键字为“(“embedded system software” OR “embedded system” OR “embedded software” OR “process control system” OR “real time control system”) AND (“requirements description” OR “requirements specification” OR “requirements modeling” OR “system description” OR “system specification” OR “system modeling” OR “requirements standard” OR “requirements format” OR “requirements documentation”)”.

检索使用的文献数据库包括 Web of Science、IEEE Xplore、ACM Digital Library、Science Direct、 Springer、Engineering Village 与中国知网. 在上述文献数据库中进行检索, 共得到 3 442 篇文献. 其中 Springer 搜到的文献最多, 为 2 443 篇, 来自中国知网的检索结果最少, 为 13 篇, 具体的搜索结果如图 2 所示. 由于数据差距比较大, 我们采用了非等比例纵坐标进行了显示.

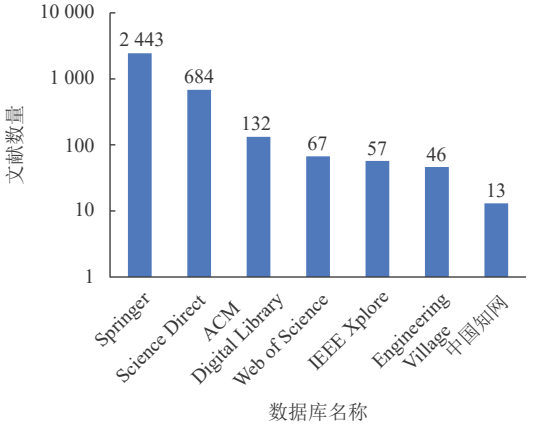

图 2 数据库检索结果

(2) 文献筛选

我们对检索出的文献进行筛选, 文献纳入和排除标准如下. 注意, 纳入符合任一纳入标准的文献, 排除符合任何一项排除标准的文献. 文献纳入标准 (include criteria, IC) 如下.

● IC1: 文献涉及了嵌入式系统相关的需求描述. 在这一阶段, 我们的目标是最大限度地扩大文献范围, 以确保研究的完整性.

● IC2: 文献把通用的需求描述和规约的语言、方法、过程成功应用到嵌入式系统真实案例上.

文献排除标准 (exclude criteria, EC) 如下.

● EC1: 无法获得电子版全文的文献.

● EC2: 用英语、中文以外的语言撰写的文献.

● EC3: 没有经过同行评审的文献.

● EC4: 存在更加完整的文献, 即同一研究有多篇文献, 仅纳入最完整的文献.

具体筛选过程如下. 首先对文献的标题、关键字、摘要进行筛选, 以确定哪些文献符合纳入/排除标准, 然后通过阅读全文进行筛选. 在分配任务时, 我们确保每篇文献的每个选择阶段都由至少两位作者完成. 存在争议的文献要先由 3 位作者进行评估, 然后再通过讨论达成共识. 按照上述标准进行筛选后, 得到 106 篇文献. 再使用滚雪球方法对这些文献的参考文献进行筛选和汇总, 最后选出 150 篇文献.

2.3 文献总体情况

本节简要介绍这 150 篇论文的总体情况.

出版年份: 图 3 显示了这些文献按照出版年份的分布情况. 统计数据显示, 从 1980 年开始, 论文发表逐年增多, 说明嵌入式系统的需求相关工作在逐步得到关注.

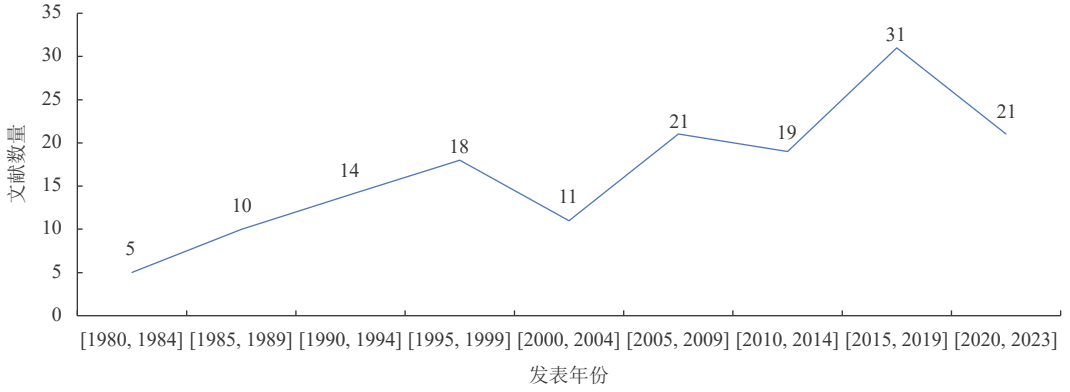

图 3 文献发表年份统计

软件开发阶段: 图 4 显示了所收集的文献按软件开发的不同阶段的分布情况. 在所选的 150 篇论文中, 11 篇涉及需求获取, 39 篇论述了需求建模与分析, 17 篇探讨了需求规约, 40 篇研究了需求验证, 10 篇涵盖了需求管理. 另外, 其中 11 篇论文探讨了系统建模, 22 篇关注了系统设计. 值得注意的是, 有些论文并不局限于研究一个阶段. 例如, 有 23 篇论文同时涉及需求建模与分析和需求规约两个阶段, 32 篇论文同时关注了需求规约和需求验证阶段, 10 篇论文既研究了需求规约又涉及系统建模阶段, 而 18 篇论文则同时包含了需求规约和系统设计两个阶段的研究. 这些论文所覆盖的不同阶段, 揭示了软件需求规约并非一蹴而就, 而是一个不断精细化的过程. 此外, 嵌入式系统的需求与传统需求有所不同, 它同时涉及系统设计阶段的研究.

应用领域: 图 5 显示了文献按应用领域的分布情况. 其中, 涉及航空航天领域的文献数量最多, 为 27 篇, 其次是汽车电子、轨道交通、医疗设备等领域. 除了这些安全攸关领域之外, 个人及家用设备也达到 8 篇. 其他领域还包括工业控制、军事装备等. 这说明嵌入式系统已经进入人民生活的方方面面.

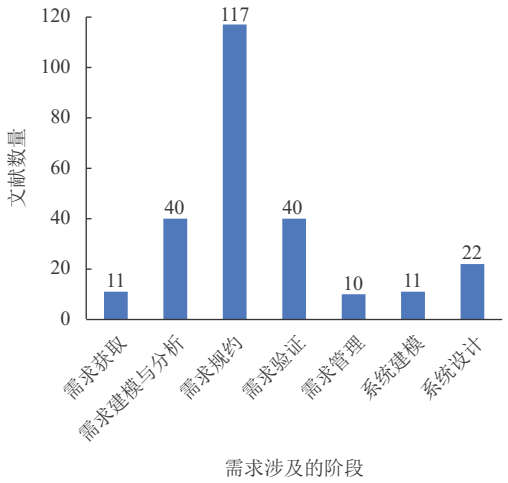

图 4 涉及的不同阶段统计

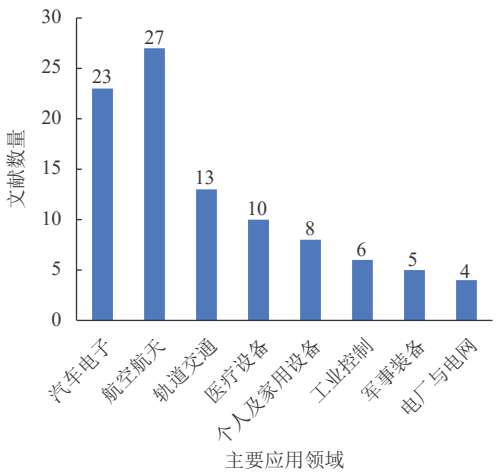

图 5 主要应用领域统计

3 嵌入式系统需求描述类型

表 1 总结了搜索到的文献中涉及的嵌入式系统需求类型, 包括系统任务意图、系统能力需求层、软件能力需求、硬件需求和软件设计约束, 它们涉及各种不同层次. 其中系统任务意图和系统能力需求的描述层次比较高, 但现有工作比较少. 大部分的工作集中在对软件能力需求的描述上, 这是嵌入式软件需求规约中必不可少的基本组成部分. 软件能力需求中功能需求和非功能需求都很重要. 描述硬件需求的研究工作较少, 另外也有较多的研究工作涉及了软件设计约束.

表 1 嵌入式系统涉及的需求类型

需求类型		文献
系统任务意图		[10–13]
系统能力需求		[10,11,14–20]
软件能力需求	功能需求	[2,5,10–12,14–18,21–97]
	非功能需求	[2,5,10,11,14–18,20–30,32,34–41,43,56,63,64,66–70,72–78,82–84,87,95,98–122]
硬件需求	系统设备需求	[13,14,16,19,20]
	软件运行平台约束	[18,22,84]
软件设计约束		[11,16,21,23,24,33,40,53,60,83,84,97,123–125]

需要注意的是, 这些不同类型的需求之间有着紧密的关系. 从实现的角度来看, 系统能力需求要满足系统任务意图, 软件能力需求和硬件需求要满足系统能力需求. 由于这样的关系, 有许多工作涉及需求描述的多个层次. 例如, SCR (software cost reduction) 方法[14−16]、ECSAM (embedded computer system analysis and modeling) 方法[11]同时包含系统能力需求和软件能力需求两个层次的描述; SDL (specification description language)[22]、SOFL (structured-object-based-formal language)[23,126,127]和 Hume[40] 都含有软件能力需求和软件设计约束的描述; Pereira 等人的研究则涉及了系统能力需求、软件能力需求和硬件需求[19]. 然而, 目前尚未有研究涉及从系统任务意图到低层次需求的精细化过程. 接下来, 我们将分别阐述这些需求类型的现有表达形式.

3.1 系统任务意图和系统能力需求

系统任务意图描述了系统任务要达成的高层目标. 例如, Leveson 等人将飞机防碰撞系统的任务意图描述为“消除飞机相距太近的危险”或者“最小化飞机相距太近情形的数量”[10]. 这里的飞机就是外部环境. Pereira 等人将意图定义为由利益干系人定义的一组系统目标[19], 如“服务更多的旅客”. 类似地使用目标模型表达意图的还有 Ponsard 等人[12] 以及 Braun 等人的工作[13].

系统能力需求是对系统功能的描述[20]. 由于嵌入式系统包含了软硬件, 软件要通过传感器等设备感知外部环境的信息, 通过执行器等设备对外部环境施加控制, 因此, 系统能力需求常表现为对系统设备与外部环境的交互或者从外部环境的输入和输出之间的关系. 例如在 SCR 方法中, 系统能力需求表现为对外部环境的监视变量 (monitored variable) 与控制变量 (controlled variable) 之间的关系. 其中, 监视变量即系统通过感知器感知到的输入, 控制变量为系统通过作用器控制的输出, 该关系可由数学公式进行描述. Leveson 等人把外部环境中的可控部分定义为过程 (process), 将系统需求表示为外部环境的输入与系统输出之间的关系或者函数 (function)[10]. 王飞等人有类似的描述[18]. Zave 等人[32] 和 Lavi 等人[11] 则将系统需求表示为设备与外部交互环境之间的交互. Lavi 等人将系统需求表示为系统的任务序列[11], 例如“当在受保护区域有入侵者时, 警报响了”.

3.2 软件能力需求

软件能力需求包括了软件的功能和非功能需求. 一般来说, 软件功能需求描述的是软件的输入和输出之间的关系. 嵌入式软件通过传感器和执行器与外部环境交互, 其输入和输出都是通过传感器和执行器得到. 很多研究都关注于输入事件和输出事件之间的关系. 例如一些研究使用状态机来表达的输入和输出事件关系[10,11,22,25−27,29,30]、EBS (event-based specification language)[33]中则通过事件关系描述语言表达这种关系[33], 而在 RSL (requirements statement language)[21], 输入和输出事件的关系则通过刺激-响应路径 (stimulus-response path) 来表达[21]. 另一些研究则关注输入数据和输出数据之间的关系. 例如, PAISLey (process-oriented applicative and interpretable specification language)[32]、ESML (extended system modeling language)[31]、SOFL 等的研究, 都涉及输入与输出数据的转换. 在 ASLAN[36]中, 输入输出变量之间要满足的性质和约束以及 SCR 和 SPARDL (spacecraft requirement description language)[41] 中, 都探讨了输入变量与输出变量之间的计算关系或者函数关系.

嵌入式软件的非功能需求很多, 主要包括时间相关需求、可靠性、健壮性、人身安全、信息安全等. 时间相关需求最常见[10,14,21,22,25,26,32,34,37,38,40,111], 它有多种叫法, 如时间需求、实时需求、时间约束、性能等. 按照对时间要求的严格程度, 时间相关需求可分为 3 类. 第 1 类是对时间序列的要求[128], 第 2 类则是对时间段、时刻的量化要求[32], 可以用最大响应时间 (响应时间不能超过某个值)、最小响应时间 (响应时间不能小于某个值)、平均响应时间、固定值等表达. 第 3 类则是对实时性的要求, 如 RTRSM* (real-time requirements specification model*)[29]、RT-FRORL (real-time frame and rule oriented requirements language)[35]、RTASLAN[37] 等中的有些约束是可以被违反的, 属于软实时. 有些是强硬要求必须实现的, 属于硬实时.

人身安全需求 (safety requirement) 和信息安全需求 (security) 也很重要. 人身安全需求指防止系统伤害人身安全或者财产安全的措施[76,98,104,106,108,109,112,114,116,117], 一般用系统不允许做什么表达[129], 描述关于安全要素存在的系统特定声明, 同时与一些质量衡量标准的最低或最高要求阈值相结合[130]. 还有些工作规定了系统必须处于怎样的安全程度. 如 Medikonda 等人进一步地把人身安全需求分为了 3 种类型: 重要安全需求、纯安全需求和安全约束[131]. 信息安全需求[20,118,119,122] 考虑软件系统可能面临的攻击, 如泄露个人数据或者允许攻击者获得对车辆的未经授权的控制, 并引入为应对各种破坏或者窃取系统信息和数据的威胁而应该采取的应对措施.

除此之外, 其他常见非功能需求还包括可靠性与健壮性. 可靠性指的是系统完成特定功能的能力或概率. 例如, PAISLey 采用概率来衡量可靠性, 它把函数的值域分为成功和失败两部分, 并利用一个随机变量、相关分布信息或一个固定值来表示两者的概率. 健壮性 (robustness) 通常指的是系统在面对非法或错误输入、意外的环境变化时, 依然能正常运行的能力. 嵌入式系统通常需要对异常事件 (如资源失效、不正确的输入等) 提供响应策略[10,14,40,78], 以确保系统的稳定运行.

3.3 硬件需求和软件设计约束

硬件是指用于处理、储存、传输计算机程序或者数据的物理设备[19]. 硬件包括如传感器、执行器这样的外部设备, 同时也包含了软件的运行平台. 由于外设和软件运行平台有不同的要求, 硬件需求又分为对外设的需求和软件运行平台的约束. 外设需求应该提供对外设特征的要求[14,16], 常包括外设的功能、用户的交互、硬件特征 (温度范围、湿度范围、电池)、动作按钮、精确度、内存规格、响应时间等[20].

软件运行平台的约束是功能设计时对处理器、内存和数据存储空间等运行平台的要求. 它们在嵌入式系统中均为有限的资源, 因此必须对它们进行描述. 很多工作都对它们进行了明确表达. 例如 Hume 提出了一个空间开销

模型, 用于预测程序的堆栈和空间使用上限; MARTE (modeling and analysis of real-time and embedded systems)[132] 包含许多与资源有关的模板属性用于表达资源限制.

软件设计约束一般包括对硬件设备的接口需求、软件结构、编程语言、开发标准要求、保密要求、可维护性、易用性等要求[84]. 接口需求详细阐述了每个功能在接口层面的要求, 这经常用于描述硬件接口的输入和输出. 一般来说, 接口需求还会包含精确度、范围、时间要求等属性. 在这里, 精确度指系统输出数据的准确程度, 也就是描述了输出数据值与理想值之间能够被接受的误差[14]. 在现有研究工作中, 对软件内部结构的设计研究较为深入. 例如, SDL 采用了块 (block) 和过程 (process) 等结构化概念来描述经过设计得到的系统内部结构. SYSREM 方法[21] 则将功能分解后的子功能分配给各系统组件, 同时完成了组件接口的设计. 此外, SYSREM 方法还提出分布式计算设计系统 (DCDS), 并详细展示了模块的设计过程.

4 现有的嵌入式系统需求描述语言

经过深入调研, 我们总结出 20 种具有代表性的需求描述语言, 如表 2 所示. 这些语言仅涵盖了 3 种需求描述类型, 分别是系统能力需求、软件能力需求与软件设计约束. 其中, 系统能力需求是描述层次最高的一种. 大多数工作都集中在软件能力需求描述上. 相较而言, 涉及软件设计约束的描述则比较少.

表 2 嵌入式系统的需求描述语言

类型	描述语言	描述关注点	描述维度	需求分析要素	非功能需求
系统能力需求	SCR[14,16]	影响系统自身行为和系统控制的环境属性	监视变量与控制变量的关系	—	精确度、时间
软件能力需求	Statecharts[26], Modechart[27], RTRSM*[29], Stateflow[30]	需要自身系统响应的外部环境事件、信号和刺激等	输入和输出序列、条件、动作和时间约束	超状态表示, AND-分解和 XOR-分解	时间
	SDL[22]		将输入和当前状态映射为输出和更新状态	层次化表示, 分解	时间
	RTRL[25]		输入事件和输出事件的对应关系	系统分解为相互独立的模块	时间
	EBS[33]		在接口上发生的事件或消息及其关系	—	时序
	SCR[14−16]	被输入设备监测到的现实世界属性和输出设备控制的属性	在不同模式下输出变量与输入变量的关系	使用模式组织系统状态	性能、异常处理
	PAISLey[32]	自身系统控制的物理对象、人与其他数字系统	一组异步交互过程的状态变迁之间的计算	层次化	性能、可靠性
	FRORL/RT-FRORL[34,35,133]	现实世界对象之间的交互、它们可能的变更、约束与假设	一组改变现实世界对象交互的规则	分解对象的活动, 逐步精化	实时
	RSML[10]	现实世界中被控过程可以被操纵和控制的变量	输入输出序列	超状态, AND-分解	精确度、时间、异常处理、接口
	SPARDL[41], Giotto[39]	现实世界中被控设备的响应及响应时间	模式转换与周期驱动的计算任务	模式含有多个任务, 超状态	时间
	ASLAN/RT-ASLAN[36,37], ASTRAL[38]	能够引起自身系统状态变迁的事件	一组改变自身系统状态的变迁	多层次, 精化	实时
	RSL[21]	将要在自身系统中处理的外部环境信息	输入数据和输出数据及其处理过程	功能分解	性能、精确度
	Hume[40]			3层表示	时间性、异常处理
	GCSR[78]	与外部环境共享的资源与其通信的信道	通信事件或者时间和资源消耗动作序列	复合的资源和事件分解	实时、资源受限性、异常处理
软件设计约束	SDL[22]	—	系统内部结构设计	—	—

在系统能力需求上只有 SCR, 而在软件设计约束上只有 SDL. 值得注意的是, 这两个语言在软件能力层都有涉及. 但在不同的需求层面上, 它们的关注点又各有侧重. 在系统能力需求层面, SCR 主要关注影响自身系统行为的可测量环境属性以及系统所能控制的环境属性, 其中, 可测量的环境属性称为监视变量, 而系统所控制的环境属性为控制变量. 系统能力需求则表现为这些环境属性或变量之间的关系, 系统必须保证这些关系的可实现性. 同时, SCR 也描述了在测量监视变量和计算控制变量时的精确程度, 为了实现可接受的系统行为, 输入和输出设备必须以足够高的精度和足够小的时间延迟来测量被监视的属性并设置被控制的属性. 在软件设计约束的描述上, SDL 描述语言关注于系统内部结构或模块的组织和设计. 例如, SDL 使用了块的概念来描绘系统结构, 块是分层的, 可以被不断分解或者嵌套, 并且还包含各种子结构概念, 用于描述复杂的结构.

接下来, 本节将从描述关注点、描述维度、需求分析要素以及非功能需求这 4 个维度, 对表达最多的软件能力描述语言进行详细阐述.

4.1 描述关注点

描述关注点是语言看待世界的角度, 它与语言如何看待自身系统的作用有密切关系. 不同的语言其描述关注点不尽相同. 有些语言将自身系统视作响应式系统, 自身系统通过对外部环境的事件、信号或者刺激做出响应, 因此它们的关注点就是外部环境发出的事件、信号或者刺激. 如 Statecharts[26] 通过事件驱动持续地对激励做出响应. 类似地, Stateflow[30] 也描述软件如何对信号、事件和基于时间的条件做出反应. 其他类似的语言还包括 Modechart[27]、RTRSM*、SDL、RTRL (real-time requirements language)[25] 和 EBS.

有的语言认为软件将要观察世界、改变世界. 如 SCR 关注从输入设备中监测到的现实世界属性, 如气压高度、雷达测量到的地面上高度等, 以及由输出设备控制的现实世界属性, 如平视显示器上飞行轨迹标记的坐标、雷达天线转向指令和转向信号等. 类似地, PAISLey 关注计算机系统环境中对分布式和连续 (通过离散模拟) 现象的建模, 其环境模型可以包括输入/输出设备、自身系统控制的物理对象、人与其他数字系统的通信链路等等. FRORL (frame-and-rule oriented requirements language)[34] 和 RT-FRORL 则关注现实世界对象之间的交互, 关注现实应用领域的对象、它们可能的变更、约束以及对这个世界的假设. RSML (requirements state machine language)[10] 面向过程控制系统, 关注的是待控制过程中可以被操纵和控制的变量. SPARDL 和 Giotto[39] 则关注被控设备的响应和响应时间.

其他的关注点总结如下. 如 ASLAN 和 RTASLAN 以及 ASTRAL (ASLAN based TRIO assertion language)[38] 将自身系统视作处于一组状态中, 它们关注能够引起自身系统状态变迁的事件. RSL 以及 Hume 则认为自身系统就是要来处理数据的, 因此, 它们关注系统中要处理的外部环境消息. GCSR (graphical communicating shared resources)[78] 将实时系统视为一组通信组件, 这些组件在有限的串行共享资源集上执行, 并通过通信信道与组件同步. 它关注与外界环境共享的资源及其通信信道.

总之, 从当前的描述关注点来看, 仅关注自身系统状态的变化或者仅将自身视为对外界刺激响应者, 将难以解决伴随着嵌入式系统规模增大所带来的问题. 随着设备数量的增加, 外部刺激事件和系统状态数据也将日益增多, 这将使用户难以理解并准确表达其需求. 而软件将要观察世界、改造世界的观点将要得到加强, 它们可以有效地将用户的需求限定在对这些外部世界的调度和控制上, 通过改造外部世界定义软件自身, 从而让用户能够明确地表达出他们的实际需求.

4.2 描述维度

由于描述关注点不同, 语言的描述维度也会不同. 很多语言从对信号、事件和时间条件做出的反应来描述软件需求, 它们常采用状态机或者状态变迁图为基础来表示. 如 Statecharts 将软件行为视为包含输入和输出、条件、动作、时间约束等要素的集合, 扩展了普通的状态转移图, 引入了层次化和并发状态, 来表达需求. 类似的还包括 RTRSM*、Stateflow. 只不过 Stateflow 使用 Statecharts 的状态机表示方法和流程图进行描述, 还提供了状态迁移表和真值表. Modechart 将软件行为视为可能运行在某个模式下执行某些特定动作的序列, 它更关注与动作的开始和完成相关的时间约束. 类似地表达内容还包括 SDL. SDL 没有使用常见的状态转移图的形式, 而是用一种更突

出消息的接收和发送的语法形式, 但最后得到的描述结果的含义仍然与状态转移图相同.

有些语言描述了输入与输出的序列, 尽管它们的具体表达形式可能略有不同. 例如, EBS 将软件行为视为接口上发生的事件或消息及其相互关系. 它定义了 3 种事件关系: 时间顺序、并发和使能关系, 并使用这 3 种事件关系的符号以及一阶谓词逻辑来描述软件的行为. RTRL 将软件行为看作是输入事件与输出事件的对应关系, 其中输出不仅与输入信号集合有关, 还与这些输入的到达顺序有关. RSML 则使用黑盒方式来描述控制器的行为, 这也就描述了控制器的输入和输出序列. 随着被控制的过程状态的改变, 其行为也会随之改变, 其采用的表示方式是层次化和并发状态机.

属于这一类的方法还包括 SCR 和 PAISLey. SCR 将软件视作一组与输出数据项关联的功能, 每个输出数据项都是由一个功能进行赋值. 这些输入数据项和输出数据项都代表了与传感器和执行器的交互事件. 它使用表格表示模式、状态、输出之间的关系. PAISLey 的基本描述单位是功能, 包含状态空间和为每个状态定义后继状态的后继函数. 功能之间的异步交互过程由交换函数来定义. 它采用层次的数据流图和内嵌的控制状态机来表示功能调用.

有些语言将软件行为看作是一组改变现实世界对象交互的规则, 如 FRORL 和 RT-FRORL. 它们的描述主要包含对象和活动. 对象含有一些属性, 活动含有参与对象、前置条件、动作序列、替代流程等. 动作可以是其他活动或断言 (assertion). 活动和断言都使用一阶谓词逻辑进行表示.

有些语言将软件行为视作模式转换与周期驱动的计算任务. 如 SPARDL, 它使用模式和模式转换来组织任务, 在模式内还含有控制流图表达的一组计算模块. SPARDL 使用状态转移图的表达形式, 模式允许嵌套, 与 Statecharts 的层次化状态类似. 类似地还包括 Giotto, 但 Giotto 不关心任务内部的具体实现方式, 只对任务的外部交互进行表示.

有些语言将软件行为视作一组能够改变自身系统状态的变迁. 例如, ASLAN 和 RTASLAN. 它们使用状态和状态转换来组织对软件行为的描述. 这里的状态是指软件或自身系统内某一时刻状态变量的取值. 这种状态不适合用状态迁移图来表示, RT-ASLAN 使用基于一阶谓词逻辑的断言和不变式 (invariant) 对状态进行描述. ASTRAL 保留了 ASLAN 和 RT-ASLAN 中相同的表示, 其语义定义在 TRIO 逻辑上.

还有描述如 GCSR 将自身系统视为一组共享有限资源的通信过程, 软件的行为由一组执行步序列组成, 其中, 每个执行步表示了通信事件或者时间和资源消耗动作. 它使用图形表示这些符号, 其过程中定义了通信事件、事件或者资源消耗动作及其之间的关系. 它的语义定义为标签迁移系统或者将其转换为进程代数 ACSR.

总之, 软件需求描述的主要维度包括行为和数据. 大多数语言更侧重于对行为的描述, 主要关注其动作行为流程, 描述对信号、事件和事件条件的反应. 这些语言通常使用有限状态机、状态转换图、谓词逻辑等手段进行描述. 一些方法会采用层次化的并发状态机 (模式转迁图) 来描述复杂的行为. 然而, 也有少数几种语言, 如 RSL 和 Hume, 更专注于数据转换. 这些语言着重于处理系统内部的外部环境信息, 用输入数据和输出数据处理过程来描述功能, 并可能考虑使用层次结构来表示复杂的功能. 还有一些工作, 如 SPARDL, 既关注行为, 也关注数据, 它们既有模式图, 又有计算任务. 对于复杂的嵌入式系统来说, 其行为和数据都具有复杂性, 因此在其软件需求描述中, 需要将这两种维度的模型有机地结合起来.

4.3 需求分析要素

经过调研, 我们发现在不同的需求描述语言中, 存在着各自独特的描述元素, 以便于进行不同层次的需求分析. 如表 2 所示, 大多数语言都能够支持不同粒度级别的需求表达. 唯独 EBS, 它仅依赖于事件以及事件间的关系进行描述, 而没有自上而下的分析方式. 在需求分析中, 主要包含以下几种分析要素形式.

有很多语言提出了超状态 (superstate) 的概念, 对状态进行层次化的表示. 如 Statecharts, 它不仅提出了超状态, 还提出了对超状态进行“AND-分解”和“XOR-分解”, 允许状态的跨层迁移和并行关系的表示, 支持从高层到低层不断精化的行为描述. Statecharts 的这种机制被很多语言进行借鉴, 包括 Modechart、RTRSM*、Stateflow、以及 RSML 语言.

还有一些语言使用模式来组织任务. 如 SCR 方法指出, 在把系统输出数据表示为系统状态和输入的映射关系

后, 使用模式 (mode) 来组织这些映射关系, 其模式为不同的系统状态类. SPARDL 使用模式来组织任务, 其模式图包含两个层次, 高层次是模式与模式转换, 低层次是在每个模式内用控制流表示一组计算任务. Giotto 语言[39] 也类似. 不同之处在于, SPARDL 中的模式允许嵌套, 这与 Statecharts 中的超状态类似, Giotto 中的任务允许并发执行, 而 SPARDL 中的控制流是顺序执行.

有一些语言本身就分了很多层次, 并定义了其层次之间的关系. 如 ASLAN 语言, 其需求描述中包含了一组层次序列 (sequence of levels), 每层都是对系统数据类型的抽象视图. 顶层视图是对系统组成的非常抽象的模型, 还包括系统做什么 (状态变迁) 和系统必须满足的关键需求 (不变式、约束). 低一些的层次会增加更多的细节, 低层次实现了高层次的需求用实现 (implementation) 关系表示. ASTRAL 语言保留了 ASLAN 的层次化表示. Hume 语言支持 3 层表示, 最外层表示外部 (静态) 声明/元编程层, 中间层描述动态过程的静态布局, 以及内层将每个过程描述为从匹配输入的模式到产生输出的表达式的动态映射. RTRL 语言使用特征来表达需求, 将需求分解为一组特征, 而特征又包含很多独立实现的功能.

有的语言支持语言要素的分解. 例如, FRORL 语言用对象和活动的槽 (frame) 来表达需求模型, 其中的活动 (activity) 概念是抽象概念, 它将每个活动槽都表示为变量, 对同一个活动的不同实例就可以表示在同一个槽里. 它提供了机制将活动分解为多个子活动, 这些子活动之间可以是顺序或并行关系. 同时它还支持逐步精化 (step-wise refinement) 机制. PAISLey 规约是由一组功能定义组成, 功能的调用由层次化的数据流图和内嵌的控制表示. GCSR 中实体是基本概念, 它可以包含很多节点 (如资源、事件等) 和复合节点 (一组资源和事件), 可以分解为更低层次的实体. RSL 语言含有 AND 和 OR 节点, 将处理需求的条件进行分解. SDL 语言使用块 (block) 组织系统行为和结构, 这种块可以被不断分解为多个块, 直到一个块只含有过程为止. SDL 还提供多种子结构 (substructure) 概念: 块的子结构用于进一步地描述块的内部结构; 信道 (channel) 的子结构用于描述信道内的行为; 信号 (signal) 精化机制是对信号的精化, 目的是隐藏低层的信号细节获得高层次的抽象, 允许自上而下地对系统行为进行描述.

除此之外, 为了支持语言中的需求层次, 有些语言还有方法支撑. 如 RSL 语言所属的 SREM 方法[21], 支持把功能分解为一组低层次的功能. SCR 方法、PAISLey 也有方法支持. 这些方法提供了获取这些需求描述语言规约的过程, 这也是很重要的部分.

总之, 现有的语言基本都支持不同描述粒度的表达, 可能是相同或不同的语言要素, 支持不同粒度的需求描述, 使得需求可以从同类型的较高抽象层次的描述, 通过分解、精化等手段不断完善为较低层次的具体描述. 也有部分工作, 支持不同需求类型的需求粒度表达, 并建立了它们之间的精化或者分解关系. 有些语言还会提供分解和精化机制, 甚至是过程支持, 但分解和精化都依靠需求分析员和领域专家, 其质量严重依赖于人的经验. 特别是, 现在的分解基本上都是功能划分, 需求之间并不存在交织. 面向复杂嵌入式系统, 分解与精化都是需要的, 但如何提高效率和质量, 依然是待解决的问题.

4.4 支持的非功能需求

根据调研, 现在的软件需求描述语言中涉及的非功能需求主要包括时间相关需求 (时序、实时、延迟、性能、时间约束等)、精确度、异常处理和资源受限性.

涉及的比较多的是时间相关需求. 有些语言使用了定时器的概念进行延迟表示. 如 Statecharts 和 SDL, 它们都在功能需求描述的状态和变迁基础上, 使用定时器指定延迟时间, 在经过该时间后定时器产生特定的 (超时) 事件或信号, 进而导致状态的转移. 有的语言表达了周期性时间约束和与之相对的偶发性 (sporadic) 时间约束. 如 SCR 中的功能描述分为需求功能和周期功能, 与这两种时间约束相对应. 需求功能指明特定的触发事件, 周期功能指定启动和停止事件, 还有执行周期. Modechart 的时间约束与系统的特定模式和条件下动作的执行相关联, 偶发性时间约束表现为完成期限和间隔时间, 周期性的时间约束则表现为执行周期和完成期限. 类似地, RT-FRORL 中的周期型时间约束被表示为时间活动的一个简单的周期属性, 阵发性时间约束的表示则引入了对全局时钟变量的操作. RT-ASLAN 中的时间需求表现为状态迁移的顺序关系和迁移的周期性属性.

其他的时间相关表达如下. ESB 中的时序表现为定义了事件的时间顺序关系, 并糅合在功能描述之中. RSL 中的性能需求通过 R-net 中的特定数据处理路径来表达, 性能需求表示为特定路径的最大和最小响应时间. GCSR 中的实时需求与动作相关, 动作的执行花费一定的时间. ASTRAL 在 RT-ASLAN 的基础上, 提供了状态迁移的开始和结束时间的操作, 还为状态迁移增加了调度时间的需求. Giotto 和 SPARDL 中的时间需求首先表现为其描述的计算任务都是周期性的, SPARDL 中的模式带有周期性属性, Giotto 中的模式带有转换频率. SPARDL 还提供了时序谓词和时序控制流关系, 用于描述模式的迁移条件和控制模式内的计算任务. PAISLey 中的性能需求与描述其 process 行为的函数相关联. RSML 中的时间需求表现在状态迁移的守护条件 (guarding condition) 上, 表示为时间函数. RSML 描述了 3 个时间函数: 前一个时间点的变量值、过去某个点的条件真值和基于时间隐式生成的事件 (即相对于状态项的超时), 都建立在守护条件之上.

对于健壮性中的异常处理, SCR 把所有异常事件和系统的响应作为单独的部分进行记录, 分为 3 类: 资源 (设备) 故障、错误的输入数据和错误的内部数据. Statecharts 中的超状态和层次化状态机可以灵活的表示局部异常事件和全局异常事件及其处理, RSML 与其类似. GCSR 通过异常边 (exception edge) 明确对异常事件的处理, 允许状态的跨层次迁移, 与 Statecharts 类似.

其他的非功能需求表达如下. 精确度需求都关联在数据上, 例如 RSL 中的精确度需求关联到验证点处存储的数据上. GCSR 中资源需求与动作相关, 动作的执行消耗一组资源. PAISLey 中的可靠性需求用概率表示, 与描述其过程行为的函数相关联.

总之, 从非功能需求上来看, 软件能力需求描述中, 都包含功能需求. 其现有的非功能性需求描述中, 仅包含了一些能表示在功能基础之上的需求, 如时间、可靠性、异常处理、精确度. 对其他不能直接表示在功能之上的需求类型, 如人身安全等, 并没有涉及.

5 嵌入式系统的需求描述发展趋势分析

本节从复杂嵌入式系统的需求描述面临的挑战开始, 提出需要研究新的嵌入式软件需求描述语言, 并进一步提出针对嵌入式软件智能合成的需求描述的具体要求.

5.1 面向复杂嵌入式系统的需求描述挑战

嵌入式系统的发展是一个软件“嵌入”计算装置的“可嵌入”过程. 先是计算功能“嵌入”到应用对象, 之后随着多形态的网络接入, 嵌入式系统呈现出网络化的特征. 近年来, 嵌入式软件被集成到更多物理对象中, 形成各种各样的可嵌入数字化设备, 它们具备环境感知和自主交互的能力, 使计算装置深度“嵌入”到应用对象并“消失”在物理世界中, 促进人机物三元世界的深度融合. 这样的复杂嵌入式系统对软件需求描述带来了如下挑战.

首先, 复杂嵌入式需求描述不能仅涉及软件能力需求, 而应该描述嵌入式系统涉及的所有类型的需求, 即任务意图、系统需求、外设需求、运行平台约束以及软件设计约束. 嵌入式软件开发从任务意图开始, 其任务意图和外在交互环境密切相关, 任务意图和系统能力需求都是其来源. 嵌入式软件将通过各种各样的传感设备和作用设备直接置身于物理世界之中, 其系统设备需求也必须记录其中. 另外嵌入式软件需要按照特定策略并运行在特定的平台上, 其资源、性能等都要必须有约束. 软件设计约束也可能存在.

其次, 需要定义各需求层次的描述维度. 复杂嵌入式系统中涉及复杂的控制流程和计算流程, 包含行为和数据两个大的维度. 但在各不同的需求层次, 维度可能完全不同. 在任务意图层, 主要是针对外部环境的预期效果. 系统能力需求只要是与外部环境的输入输出序列. 而软件需求则是与系统设备的输入输出序列, 软件设计约束则包含了输入输出之间的计算公式等.

再次, 需要定义系统性的软件需求规约方法, 以建立从任务意图到软件需求 (包括功能需求与非功能需求) 不同层次之间的追踪关系, 应对复杂度不断增加的嵌入式系统需求变化. 在嵌入式系统中, 系统任务意图是软件开发的开始, 系统能力需求满足任务意图, 而软件能力需求和系统设备需求共同满足系统能力需求. 有了追踪关系, 就可以有效根据需求变化的原因, 进行软件需求的变化.

第四, 需要提供更高效的需求分析方法对需求进行分解和精化机制. 嵌入式系统中, 由于其软件的强设备依赖性, 使得软件需求的控制需求交织, 出现在不同需求中对同一设备的调度控制, 比如“自动关灯”和“人工关灯”两个需求都是要对灯光进行控制. 这种设备交织性使得原来的划分式需求分解 (无交织) 不再适用, 它们在分解中并没有关注设备的特性, 可能会出现分解后需求不一致性问题, 如一个需求可能要求开灯, 而另一个需求则在同时要求关灯. 另外, 随着嵌入式设备的增多和嵌入式系统复杂度的增加, 人工的需求分析将成为开发的瓶颈, 通过需要自动化的需求分析方法以提高效率.

综上所述, 复杂嵌入式系统需要设计新的需求描述语言. 在设计中, 需要针对嵌入式软件的特征, 根据嵌入式软件任务意图描述, 在各维度各层次需求分析基础上, 提取软件嵌入式语言的主要成分并定义语言的逻辑, 支持嵌入式软件需求的表达. 这是跨越嵌入式软件整个需求阶段的任务, 需要综合借鉴并扩展现有的需求工程方法, 结合有效的需求提取、建模、分析、仿真和验证等技术, 提出语言结构, 定义具有足够表达能力的结构化的嵌入式系统需求描述语言.

5.2 嵌入式软件智能合成带来的要求

目前软件智能合成已成为一种备受关注的软件自动化技术[134]. 软件智能合成是指在传统软件合成技术基础上, 采用机器学习等人工智能基础, 利用已有的代码知识自动合成满足用户意图的软件. 承载用户意图的软件需求规约为软件智能合成的基础, 其重要性不言而喻. 嵌入式系统的软件智能合成也将成为未来研究热点[135], 这给嵌入式系统软件需求规约的描述语言带来了新的要求.

首先, 从描述维度上来说, 它需要表达各种维度的需求规约, 包括行为、数据、约束等. 嵌入式软件与一般软件相同, 每段代码都具有控制流和数据流. 由于嵌入式软件大部分需要实时地调度系统设备, 在需求描述中必须加入对系统设备的描述, 如端口、响应时间、通信协议等. 不仅如此, 它还需要加上对性能、效率、安全、可靠等的约束, 由此才能合成符合设备调度需求的软件.

其次, 从描述粒度上来看, 软件需求最好能分解到一个比较小的粒度, 也称为原子需求. 这个粒度最好是能跟软件资产的功能保持在同一个描述粒度上, 或者是更小的粒度. 依然需要精化技术将需求从高抽象层次的描述完善为低层次的具体描述. 除此之外, 由于现在的嵌入式系统的高复杂性, 需求的分解或解耦必不可少. 与传统信息系统的软件分解——功能划分不同, 嵌入式系统的软件由于与硬件设备的密切相关, 其在分解时还需要考虑在不同的功能中可能要调用相同的设备, 存在设备的共享, 这将给相互交织的软件需求的解耦带来新的挑战.

再次, 从描述方案的角度来说, 一个完整的项目需求规约还要明确原子需求之间的依赖关系, 从而合成完整的项目代码. 由于原子需求之间设备的交织, 将可能带来控制依赖, 如需求“自动开灯”要求灯要先处于加电状态, 而需求“初始化”则会将使灯进入加电状态, 则“自动开灯”则控制依赖于“初始化”. 由于原子需求之间的数据共享, 可能会带来数据依赖, 即一个需求生产数据, 而另一个需求使用数据. 这些控制依赖和数据依赖可能表现为原子需求之间的顺序关系或者并发关系, 给后续的合成带来影响.

最后, 从描述形式上来说, 面向智能合成的软件需求描述语言必须是机器可理解的, 以方便后期的代码合成. 最好是形式化表达的, 具有严格的语法和精确的语义. 其软件需求规约能自动地转换成仿真模型或验证模型, 在合成软件之前进行需求的仿真模拟、确认和验证, 能自动生成最终合成软件的测试用例, 在合成之后进行软件测试. 经过多次的仿真、验证与测试, 保证合成代码的质量.

6 总 结

本文面向嵌入式系统的需求描述这一主题进行了系统文献综述, 提供了对嵌入式系统的需求描述类型现状的概览, 同时全面比较了现有的嵌入式系统需求描述语言的能力, 总结了复杂嵌入式系统需求描述面临的挑战, 预测了未来趋势, 并针对软件智能合成任务, 对嵌入式系统的需求描述语言的能力要求进行了讨论.

在进行综述调研的过程中, 我们的排除标准可能存在一定的偏颇, 这使得我们的综述论文可能并未涵盖所有相关领域. 例如, 我们并未包含使用其他语言撰写的参考文献. 然而, 由于大部分研究成果都有对应的英文版本, 这

并不会对我们对嵌入式需求描述类型和需求描述语言现状的调研造成实质性影响. 另外, 搜索过程可能会有一定的不稳定性, 搜索引擎有时会出现大量与搜索关键词无关的文献. 然而, 相关度较高的文献通常都会优先显示, 因此并不会对我们的研究结果产生影响.

目前大部分需求描述语言仅描述了软件能力需求, 没有描述各种可能的其他类型需求, 如系统任务意图、系统能力需求等. 面向未来的嵌入式软件智能合成, 新的需求描述语言描述从任务意图开始的各种类型的需求, 建立它们的追踪关系, 能分析设备共享带来的控制需求交织的问题, 通过解耦, 将需求描述在一个较合适的粒度, 以方便后续的基于软件资产的代码合成.

References:

[1] Broy M. Challenges in automotive software engineering. In: Proc. of the 28th Int'l Conf. on Software Engineering. Shanghai: ACM, 2006. 33–42. [doi: 10.1145/1134285.1134292]

[2] Feng JC, Miao WK, Zheng HY, Huang YH, Li JW, Wang Z, Su T, Gu B, Pu GG, Yang MF, He JF. FREPA: An automated and formal approach to requirement modeling and analysis in aircraft control domain. In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and the Symp. on the Foundations of Software Engineering. ACM, 2020. 1376–1386. [doi: 10.1145/3368089.3417047]

[3] Mumtaz S, Alsohaily A, Pang ZB, Rayes A, Tsang KF, Rodriguez J. Massive Internet of Things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation. IEEE Industrial Electronics Magazine, 2017, 11(1): 28–33. [doi: 10.1109/MIE.2016.2618724]

[4] Broy M, Stauner T. Requirements engineering for embedded systems. 1999. https://wwwbroy.in.tum.de/publ/papers/femsys_boesswet_1997_Conference.pdf#:~:text=In%20requirements%20engineering%20we%20describe%20the%20required%20behaviour,be%20as%20simple%2C%20abstract%2C%20and%20suggestive%20as%20possible

[5] Naumchev A, Meyer B, Mazzara M, Galinier F, Bruel JM, Ebersold S. Autoreq: Expressing and verifying embedded software requirements. arXiv:1710.02801v1, 2017.

[6] Davis AM. The design of a family of application-oriented requirements languages. Computer, 1982, 15(5): 21–28. [doi: 10.1109/MC.1982.1654021]

[7] Melhart BE. Specification languages for embedded systems: A survey. Technical Report, Irvine: UCI, 1988.

[8] Jin Z. Environment Modeling-based Requirements Engineering for Software Intensive Systems. Amsterdam: Elsevier, 2018. [doi: 10.1016/C2014-0-00030-5]

[9] Kitchenham B. Procedures for performing systematic reviews. Technical Report, 0400011T.1, Keele: Keele University, 2004. 1–26.

[10] Leveson NG, Heimdahl MPE, Hildreth H, Reese JD. Requirements specification for process-control systems. IEEE Trans. on Software Engineering, 1994, 20(9): 684–707. [doi: 10.1109/32.317428]

[11] Lavi JZ, Kudish J. Systems modeling & requirements specification using ECSAM: An analysis method for embedded & computer-based systems. Innovations in Systems and Software Engineering, 2005, 1(2): 100–115. [doi: 10.1007/s11334-005-0010-4]

[12] Ponsard C, Massonet P, Molderez JF, Rifaut A, van Lamsweerde A, van Tran H. Early verification and validation of mission critical systems. Formal Methods in System Design, 2007, 30(3): 233–247. [doi: 10.1007/s10703-006-0028-8]

[13] Braun P, Broy M, Houdek F, Kirchmayr M, Müller M, Penzenstadler B, Pohl K, Weyer T. Guiding requirements engineering for software-intensive embedded systems in the automotive industry. Computer Science-research and Development, 2014, 29(1): 21–43. [doi: 10.1007/s00450-010-0136-y]

[14] Heninger KL. Specifying software requirements for complex systems: New techniques and their application. IEEE Trans. on Software Engineering, 1980, SE-6(1): 2–13. [doi: 10.1109/TSE.1980.230208]

[15] Parnas DL, Madey J. Functional documents for computer systems. Science of Computer Programming, 1995, 25(1): 41–61. [doi: 10.1016/0167-6423(95)96871-J]

[16] Heitmeyer CL. Software cost reduction. In: Marciniak JJ, ed. Encyclopedia of Software Engineering. Hoboken: John Wiley & Sons, 2002. [doi: 10.1002/0471028959.sof307]

[17] Yoo J, Kim T, Cha S, Lee JS, Seong Son H. A formal software requirements specification method for digital nuclear plant protection systems. Journal of Systems and Software, 2005, 74(1): 73–83. [doi: 10.1016/j.jss.2003.10.018]

[18] Wang F, Yang ZB, Huang ZQ, Zhou Y, Liu CW, Zhang WB, Xue L, Xu JM. Approach for generating AADL model based on restricted natural language requirement template. Ruan Jian Xue Bao/Journal of Software, 2018, 29(8): 2350–2370 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5530.htm [doi: 10.13328/j.cnki.jos.005530]

[19] Pereira T, Sousa A, Silva R, Albuquerque D, Alencar F, Castro J. A metamodel to guide a requirements elicitation process for embedded systems. In: Proc. of the 11th Int'l Conf. on the Quality of Information and Communications Technology. Coimbra: IEEE, 2018. 101–109. [doi: 10.1109/QUATIC.2018.00023]

[20] Apvrille L, Li LW. Harmonizing safety, security and performance requirements in embedded systems. In: Proc. of the 2019 Design, Automation & Test in Europe Conf. & Exhibition (DATE). Florence: IEEE, 2019. 1631–1636. [doi: 10.23919/DATE.2019.8715124]

[21] Alford M. SREM at the age of eight; the distributed computing design system. Computer, 1985, 18(4): 36–46. [doi: 10.1109/MC.1985.1662863]

[22] Belina F, Hogrefe D. The CCITT-specification and description language SDL. Computer Networks and ISDN Systems, 1989, 16(4): 311–341. [doi: 10.1016/0169-7552(89)90078-0]

[23] Liu SY, Offutt AJ, Ho-Stuart C, Sun Y, Ohba M. SOFL: A formal engineering methodology for industrial applications. IEEE Trans. on Software Engineering, 1998, 24(1): 24–45. [doi: 10.1109/32.663996]

[24] Liu S, Asuka M, Komaya K, Nakamura Y. An approach to specifying and verifying safety-critical systems with practical formal method SOFL. In: Proc. of the 4th IEEE Int'l Conf. on Engineering of Complex Computer Systems. Monterey: IEEE, 1998. 100–114. [doi: 10.1109/ICECCS.1998.706660]

[25] Taylor B. A method for expressing the functional requirements of real-time systems. Annual Review in Automatic Programming, 1980, 10: 111–120. [doi: 10.1016/0066-4138(82)90015-5]

[26] Harel D. Statecharts: A visual formalism for complex systems. Science of Computer Programming, 1987, 8(3): 231–274. [doi: 10.1016/0167-6423(87)90035-9]

[27] Jahanian F, Mok AK. Modechart: A specification language for real-time systems. IEEE Trans. on Software Engineering, 1994, 20(12): 933–947. [doi: 10.1109/32.368134]

[28] Shu FD, Wu GQ, Wang M. Embedded real-time software-oriented requirements engineering environment—SREE. Computer Science, 2002, 29(4): 4–8, 14 (in Chinese with English abstract). [doi: 10.3969/j.issn.1002-137X.2002.04.002]

[29] Shu FD, Wu GQ, Li MS. An embedded real-time software oriented requirements specification language and checking methods. Ruan Jian Xue Bao/Journal of Software, 2004, 15(11): 1595–1606 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1595.htm

[30] The MathWorks. Stateflow® and Stateflow® Coder™ User's Guide. Natick: The MathWorks, 2009.

[31] Bruyn W, Jense R, Keskar D, Ward P. An extended systems modeling language (ESML). ACM SIGSOFT Software Engineering Notes, 1988, 13(1): 58–67. [doi: 10.1145/43857.43866]

[32] Zave P. An operational approach to requirements specification for embedded systems. IEEE Trans. on Software Engineering, 1982, SE-8(3): 250–269. [doi: 10.1109/TSE.1982.235254]

[33] Chen BS, Yeh RT. Formal specification and verification of distributed systems. IEEE Trans. on Software Engineering, 1983, SE-9(6): 710–722. [doi: 10.1109/TSE.1983.235434]

[34] Tsai JJP. A knowledge-based system for software design. IEEE Journal on Selected Areas in Communications, 1988, 6(5): 828–841. [doi: 10.1109/49.634]

[35] Tsai JJJP, Jang HC. A knowledge-based approach for the specification and analysis of real-time software systems. Int'l Journal on Artificial Intelligence Tools, 1992, 1(1): 1–35. [doi: 10.1142/S0218213092000119]

[36] Auernheimer B, Kemmerer RA. ASLAN User's Manual. Santa Barbara: University of California, 1985.

[37] Auernheimer B, Kemmerer RA. RT-ASLAN: A specification language for real-time systems. IEEE Trans. on Software Engineering, 1986, SE-12(9): 879–889. [doi: 10.1109/TSE.1986.6313044]

[38] Ghezzi C, Kemmerer RA. ASTRAL: An assertion language for specifying realtime systems. In: Proc. of the 3rd European Software Engineering Conf. Milan: Springer, 1991. 122–146. [doi: 10.1007/3540547428_46]

[39] Henzinger TA, Horowitz B, Kirsch CM. Giotto: A time-triggered language for embedded programming. In: Proc. of the 1st Int'l Workshop on Embedded Software. Tahoe City: Springer, 2001. 166–184. [doi: 10.1007/3-540-45449-7_12]

[40] Hammond K, Michaelson G. Hume: A domain-specific language for real-time embedded systems. In: Proc. of the 2nd Int'l Conf. on Generative Programming and Component Engineering. Erfurt: Springer, 2003. 37–56. [doi: 10.1007/978-3-540-39815-8_3]

[41] Gu B, Dong YW, Wang Z. Formal modeling approach for aerospace embedded software. Ruan Jian Xue Bao/Journal of Software, 2015, 26(2): 321–331 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4784.htm [doi: 10.13328/j.cnki.jos.004784]

[42] Miao WK, Pu GG, Yao YB, Su T, Bao DZ, Liu Y, Chen SH, Xiong KP. Automated requirements validation for ATP software via specification review and testing. In: Proc. of the 18th Int'l Conf. on Formal Engineering Methods. Tokyo: Springer, 2016. 26–40. [doi: 10.1007/978-3-319-47846-3_3]

[43] Feng JC. Detailed requirement of embedded system oriented formal modeling and analysis [Ph.D. Thesis]. Shanghai: East China Normal University, 2022 (in Chinese with English abstract). [doi: 10.27149/d.cnki.ghdsu.2022.004041]

[44] Wang S, Feng JC, Zhu JY, Huang YH, Zheng HY, Xu XR, Miao WK, Zhang X, Pu GG. A dimensional analysis method for the requirements model of railway control software. Chinese Journal of Computers, 2020, 43(11): 2152–2165 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2020.02152]

[45] Dietrich D, Atlee JM. A pattern for structuring the behavioural requirements of features of an embedded system. In: Proc. of the 3rd Int'l Workshop on Requirements Patterns (RePa). Rio De Janeiro: IEEE, 2013. 1–7. [doi: 10.1109/RePa.2013.6602664]

[46] Hoang TS, Snook C, Salehi A, Butler M, Ladenberger L. Validating and verifying the requirements and design of a haemodialysis machine using the rodin toolset. Science of Computer Programming, 2018, 158: 122–147. [doi: 10.1016/j.scico.2017.11.002]

[47] Park S. Software requirement specification based on a gray box for embedded systems: A case study of a mobile phone camera sensor controller. Computers, 2019, 8(1): 20. [doi: 10.3390/computers8010020]

[48] Ghazel M, Yang J, El-Koursi EM. A pattern-based method for refining and formalizing informal specifications in critical control systems. Journal of Innovation in Digital Ecosystems, 2015, 2(1–2): 32–44. [doi: 10.1016/j.jides.2015.11.001]

[49] Galinier F. A DSL for requirements in the context of a seamless approach. In: Proc. of the 33rd IEEE/ACM Int'l Conf. on Automated Software Engineering. Montpellier: IEEE, 2018. 932–935. [doi: 10.1145/3238147.3241538]

[50] Bujorianu MC, Bujorianu ML. An integrated specification framework for embedded systems. In: Proc. of the 5th IEEE Int'l Conf. on Software Engineering and Formal Methods (SEFM 2007). London: IEEE, 2007. 161–172. [doi: 10.1109/SEFM.2007.6]

[51] Lattemann F, Lehmann E. A methodological approach to the requirement specification of embedded systems. In: Proc. of the 1st IEEE Int'l Conf. on Formal Engineering Methods. Hiroshima: IEEE, 1997. 183–191. [doi: 10.1109/ICFEM.1997.630425]

[52] Maraninchi F, Rémond Y. Argos: An automaton-based synchronous language. Computer Languages, 2001, 27(1–3): 61–92. [doi: 10.1016/S0096-0551(01)00016-9]

[53] Roßkopf A, Tempelmeier T. Aspects of flight control software—A software engineering point of view. Control Engineering Practice, 2000, 8(6): 675–680. [doi: 10.1016/S0967-0661(00)00012-5]

[54] Fuchs NE, Schwertel U, Schwitter R. Attempto controlled English—Not just another logic specification language. In: Proc. of the 8th Int'l Workshop on Logic Programming Synthesis and Transformation. Manchester: Springer, 1999. 1–20. [doi: 10.1007/3-540-48958-4_1]

[55] Pettersson F, Ivarsson M, Öhman P. Automotive use case standard for embedded systems. ACM SIGSOFT Software Engineering Notes, 2005, 30(4): 1–6. [doi: 10.1145/1082983.1083193]

[56] Shan JH, Zhao HY, Wang JB, Wang RX, Ruan CL, Yao ZX. An extended TASM-based requirements modeling approach for real-time embedded software: An industrial case study. In: Proc. of the 15th National Software Application Conf. on Software Engineering and Methodology for Emerging Domains. Kunming: Springer, 2016. 19–34. [doi: 10.1007/978-981-10-3482-4_2]

[57] Jung H, Lee C, Kang SH, Kim S, Oh H, Ha S. Dynamic behavior specification and dynamic mapping for real-time embedded systems: Hopes approach. ACM Trans. on Embedded Computing Systems, 2014, 13(4s): 135. [doi: 10.1145/2584658]

[58] Post A, Menzel I, Podelski A. Applying restricted english grammar on automotive requirements—Does it work? A case study. In: Proc. of the 17th Int'l Working Conf. on Requirements Engineering: Foundation for Software Quality. Essen: Springer, 2011. 166–180. [doi: 10.1007/978-3-642-19858-8_17]

[59] Yue T, Briand LC, Labiche Y. A use case modeling approach to facilitate the transition towards analysis models: Concepts and empirical evaluation. In: Proc. of the 12th Int'l Conf. on Model Driven Engineering Languages and Systems. Denver: Springer, 2009. 484–498. [doi: 10.1007/978-3-642-04425-0_37]

[60] Stachtiari E, Mavridou A, Katsaros P, Bliudze S, Sifakis J. Early validation of system requirements and design through correctness-by-construction. Journal of Systems and Software, 2018, 145: 52–78. [doi: 10.1016/j.jss.2018.07.053]

[61] Mavin A, Wilkinson P, Harwood A, Novak M. Easy approach to requirements syntax (EARS). In: Proc. of the 17th IEEE Int'l Requirements Engineering Conf. Atlanta: IEEE, 2009. 317–322. [doi: 10.1109/RE.2009.9]

[62] Mav AM, Wilkinson P. Ten years of EARS. IEEE Software, 2019, 36(5): 10–14. [doi: 10.1109/MS.2019.2921164]

[63] Górski J. Formal specification of real time systems. Computer Physics Communications, 1988, 50(1–2): 71–88. [doi: 10.1016/0010-4655(88)90117-8]

[64] Al-Fedaghi S. High-level representation of time in diagrammatic specification. Procedia Computer Science, 2015, 62: 478–486. [doi: 10.1016/j.procs.2015.08.519]

[65] Denger C, Berry DM, Kamsties E. Higher quality requirements specifications through natural language patterns. In: Proc. of the 2003 Symp. on Security and Privacy. Herzlia: IEEE, 2003. 80–90. [doi: 10.1109/SWSTE.2003.1245428]

[66] Marques MRS, Siegert E, Brisolara L. Integrating UML, MARTE and sysML to improve requirements specification and traceability in

the embedded domain. In: Proc. of the 12th IEEE Int'l Conf. on Industrial Informatics (INDIN). Porto Alegre: IEEE, 2014. 176–181. [doi: 10.1109/INDIN.2014.6945504]

[67] Rashid M, Anwar MW, Azam F, Kashif M. Model-based requirements and properties specifications trends for early design verification of embedded systems. In: Proc. of the 11th System of Systems Engineering Conf. (SoSE). Kongsberg: IEEE, 2016. 1–7. [doi: 10.1109/SYSOSE.2016.7542917]

[68] Morzenti A, San Pietro P. Object-oriented logical specification of time-critical systems. ACM Trans. on Software Engineering and Methodology, 1994, 3(1): 56–98. [doi: 10.1145/174634.174636]

[69] Ravindran B, Edwards S. Palette: A reuse-oriented specification language for real-time systems. In: Proc. of the 6th Int'l Conf. on Software Reuse: Advances in Software Reusability. Vienna: Springer, 2000. 20–40. [doi: 10.1007/978-3-540-44995-9_2]

[70] Kang KC, Ko KI. PARTS: A temporal logic-based real-time software specification and verification method. In: Proc. of the 17th Int'l Conf. on Software Engineering. Seattle: IEEE, 1995. 169. [doi: 10.1145/225014.225030]

[71] Videira C, Da Silva AR. Patterns and metamodel for a natural-language-based requirements specification language. In: Proc. of the 17th Conf. on Advanced Information Systems Engineering. Porto: CAiSE, 2005.

[72] Mahmud N, Seceleanu C, Ljungkrantz O. ReSA: An ontology-based requirement specification language tailored to automotive systems. In: Proc. of the 10th IEEE Int'l Symp. on Industrial Embedded Systems (SIES). Siegen: IEEE, 2015. 1–10. [doi: 10.1109/SIES.2015.7185035]

[73] Mahmud N, Seceleanu C, Ljungkrantz O. Specification and semantic analysis of embedded systems requirements: From description logic to temporal logic. In: Proc. of the 15th Int'l Conf. on Software Engineering and Formal Methods. Trento: Springer, 2017. 332–348. [doi: 10.1007/978-3-319-66197-1_21]

[74] Welch LR, Ravindran B, Shirazi BA, Bruggeman C. Specification and modeling of dynamic, distributed real-time systems. In: Proc. of the 19th IEEE Real-time Systems Symp. Madrid: IEEE, 1998. 72–81. [doi: 10.1109/REAL.1998.739732]

[75] Ebert C. Specifying, designing and rapid prototyping computer systems with structured Petri nets. In: Frey HH, ed. Safety of Computer Control Systems 1992. Amsterdam: Elsevier, 1992. 19–24. [doi: 10.1016/B978-0-08-041893-3.50008-2]

[76] Damm W, Hungar H, Josko B, Peikenkamp T, Stierand I. Using contract-based component specifications for virtual integration testing and architecture design. In: Proc. of the 2011 Design, Automation & Test in Europe. Grenoble: IEEE, 2011. 1–6. [doi: 10.1109/DATE.2011.5763167]

[77] Camilli M, Gargantini A, Scandurra P. Zone-based formal specification and timing analysis of real-time self-adaptive systems. Science of Computer Programming, 2018, 159: 28–57. [doi: 10.1016/j.scico.2018.03.002]

[78] Ben-Abdallah H, Lee I, Kim YS. The integrated specification and analysis of functional, temporal, and resource requirements. In: Proc. of the 3rd IEEE Int'l Symp. on Requirements Engineering. Annapolis: IEEE, 1997. 198–209. [doi: 10.1109/ISRE.1997.566870]

[79] Doering D, Pereira CE, Denes P, Joseph J. A model driven engineering approach based on aspects for high speed scientific X-rays cameras. In: Proc. of the 16th IEEE Int'l Symp. on Object/Component/Service-oriented Real-time Distributed Computing (ISORC 2013). Paderborn: IEEE, 2013. 1–8. [doi: 10.1109/ISORC.2013.6913190]

[80] Dutertre B, Stavridou V. Formal requirements analysis of an avionics control system. IEEE Trans. on Software Engineering, 1997, 23(5): 267–278. [doi: 10.1109/32.588520]

[81] Faulk S, Brackett J, Ward P, Kirby J. The core method for real-time requirements. IEEE Software, 1992, 9(5): 22–33. [doi: 10.1109/52.156894]

[82] Ferrante O, Passerone R, Ferrari A, Mangeruca L, Sofronis C. BCL: A compositional contract language for embedded systems. In: Proc. of the 2014 IEEE Emerging Technology and Factory Automation (ETFA). Barcelona: IEEE, 2014. 1–6. [doi: 10.1109/ETFA.2014.7005353]

[83] Pierce RH, Ayache S, Ward R, Stevens J, Clifton H, Galle J. Capturing and verifying performance requirements for hard real time systems. In: Proc. of the 1997 Int'l Conf. on Reliable Software Technologies. London: Springer, 1997. 137–148. [doi: 10.1007/3-540-63114-3_13]

[84] Du G, Lin J. Real-time embedded software requirements description framework exploration. Quality and Reliability, 2008, (1): 47–50.

[85] Goldsack SJ, Finkelstein ACW. Requirements engineering for real-time systems. Software Engineering Journal, 1991, 6(3): 101–115. [doi: 10.1049/sej.1991.0014]

[86] Ravn AP, Rischel H, Hansen KM. Specifying and verifying requirements of real-time systems. IEEE Trans. on Software Engineering, 1993, 19(1): 41–55. [doi: 10.1109/32.210306]

[87] Ribeiro FGC, Misra S, Soares MS. Application of an extended sysML requirements diagram to model real-time control systems. In: Proc. of the 13th Int'l Conf. on Computational Science and Its Applications. Ho Chi Minh City: Springer, 2013. 70–81. [doi: 10.1007/

978-3-642-39646-5_6]

[88] Saiedian H, Kumarakulasingam P, Anan M. Scenario-based requirements analysis techniques for real-time software systems: A comparative evaluation. Requirements Engineering, 2005, 10(1): 22–33. [doi: 10.1007/s00766-004-0192-6]

[89] Laouadi MA, Mokhati F, Seridi-Bouchelaghem H. A novel formal specification approach for real time multi-agent system functional requirements. In: Proc. of the 8th German Conf. on Multiagent System Technologies. Leipzig: Springer, 2010. 15–27. [doi: 10.1007/978-3-642-16178-0_4]

[90] Siegl S, Hielscher KS, German R. Model based requirements analysis and testing of automotive systems with timed usage models. In: Proc. of the 18th IEEE Int'l Requirements Engineering Conf. Sydney: IEEE, 2010. 345–350. [doi: 10.1109/RE.2010.49]

[91] Zhou JL, Lu Y, Lundqvist K, Lönn H, Karlsson D, Liwång B. Towards feature-oriented requirements validation for automotive systems. In: Proc. of the 22nd Int'l Requirements Engineering Conf. Karlskrona: IEEE, 2014. 428–436. [doi: 10.1109/RE.2014.6912294]

[92] Zhu XN. A formal model for service-based behavior specification using stream-based I/O tables. In: Proc. of the 10th Int'l Workshop on Formal Aspects of Component Software. Nanchang: Springer, 2014. 369–383. [doi: 10.1007/978-3-319-07602-7_22]

[93] Sinha R, Patil S, Pang C, Vyatkin V, Dowdeswell B. Requirements engineering of industrial automation systems: Adapting the cesar requirements meta model for safety-critical smart grid software. In: Proc. of the 41st Annual Conf. of the IEEE Industrial Electronics Society. Yokohama: IEEE, 2015. 002172–002177. [doi: 10.1109/IECON.2015.7392423]

[94] Li R, Ma SL, Yao WT. Ontology-based requirements generation for credibility validation of safety-critical system. In: Proc. of the 2015 Int'l Conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. Liverpool: IEEE, 2015. 849–854. [doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.126]

[95] Ribeiro FGC, Pereira CE, Rettberg A, Soares MS. Model-based requirements specification of real-time systems with UML, sysML and MARTE. Software & Systems Modeling, 2018, 17(1): 343–361. [doi: 10.1007/s10270-016-0525-1]

[96] Westman J, Nyberg M. Providing tool support for specifying safety-critical systems by enforcing syntactic contract conditions. Requirements Engineering, 2019, 24(2): 231–256. [doi: 10.1007/s00766-017-0286-6]

[97] Blouin D, Giese H. Combining requirements, use case maps and AADL models for safety-critical systems design. In: Proc. of the 42th Euromicro Conf. on Software Engineering and Advanced Applications (SEAA). Limassol: IEEE, 2016. 266–274. [doi: 10.1109/SEAA.2016.15]

[98] Fu RR, Bao XH, Zhao TD. Generic safety requirements description templates for the embedded software. In: Proc. of the 9th Int'l Conf. on Communication Software and Networks (ICCSN). Guangzhou: IEEE, 2017. 1477–1481. [doi: 10.1109/ICCSN.2017.8230353]

[99] Chen XH, Han L, Liu J, Sun HY. Using safety requirement patterns to elicit requirements for railway interlocking systems. In: Proc. of the 24th Int'l Requirements Engineering Conf. Workshops (REW). Beijing: IEEE, 2016. 296–303. [doi: 10.1109/REW.2016.055]

[100] Hu ZM, Huang LT, Zhao YX. Hybrid modeling language for aerospace model software. Aerospace Control and Application, 2021, 47(2): 25–31 (in Chinese with English abstract). [doi: 10.3969/j.issn.1674-1579.2021.02.004]

[101] Feyerabend K, Josko B. A visual formalism for real time requirement specifications. In: Proc. of the 4th Int'l AMAST Workshop on Aspects of Real-time Systems and Concurrent and Distributed Software. Palma: Springer, 1997. 156–168. [doi: 10.1007/3-540-63010-4_11]

[102] Roop PS, Sowmya A. Hidden time model for specification and verification of embedded systems. In: Proc. of the 10th EUROMICRO Workshop on Real-time Systems. Berlin: IEEE, 1998. 98–105. [doi: 10.1109/EMWRTS.1998.685073]

[103] Khan AM, Mallet F, Rashid M. Natural interpretation of UML/MARTE diagrams for system requirements specification. In: Proc. of the 11th IEEE Symp. on Industrial Embedded Systems (SIES). Krakow: IEEE, 2016. 1–6. [doi: 10.1109/SIES.2016.7509429]

[104] Kirner TG, Davis AM. Nonfunctional requirements of real-time systems. Advances in Computers, 1996, 42: 1–37. [doi: 10.1016/S0065-2458(08)60483-0]

[105] Kirner TG, Davis AM. Requirements specification of real-time systems: Temporal parameters and timing-constraints. Information and Software Technology, 1996, 38(12): 735–741. [doi: 10.1016/0950-5849(96)01104-4]

[106] Martins LEG, Gorschek T. Requirements engineering for safety-critical systems: A systematic literature review. Information and Software Technology, 2016, 75: 71–89. [doi: 10.1016/j.infsof.2016.04.002]

[107] Lee HK, Lee WJ, Chae HS, Kwon YR. Specification and analysis of timing requirements for real-time systems in the CBD approach. Real-time Systems, 2007, 36(1–2): 135–158. [doi: 10.1007/s11241-007-9017-2]

[108] Wu X, Liu C, Xia QX. Safety requirements modeling based on RUCM. In: Proc. of the 2014 Computers, Communications and IT Applications Conf. Beijing: IEEE, 2014. 217–222. [doi: 10.1109/ComComAp.2014.7017199]

[109] Han L, Liu J, Zhou TL, Sun JF, Chen XH. Safety requirements specification and verification for railway interlocking systems. In: Proc.

of the 40th Annual Computer Software and Applications Conf. (COMPSAC). Atlanta: IEEE, 2016. 335–340. [doi: 10.1109/COMPSAC.2016.182]

[110] Petters S, Muth A, Kolloch T, Hopfner T, Fischer F, Farber G. The REAR framework for emulation and analysis of embedded hard real-time systems. In: Proc. of the 10th IEEE Int'l Workshop on Rapid System Prototyping. Shortening the Path from Specification to Prototype. Clearwater: IEEE, 1999. 100–107. [doi: 10.1109/IWRSP.1999.779038]

[111] De Lemos R, Saeed A, Anderson T. Analysis of timeliness requirements in safety-critical systems. In: Proc. of the 1992 Int'l Symp. on Formal Techniques in Real-time and Fault-tolerant Systems. Berlin: Springer, 1992. 171–192. [doi: 10.5555/646842.706607]

[112] Aoyama M, Yoshino A. AORE (aspect-oriented requirements engineering) methodology for automotive software product lines. In: Proc. of the 15th Asia-Pacific Software Engineering Conf. Beijing: IEEE, 2008. 203–210. [doi: 10.1109/APSEC.2008.59]

[113] Freitas EP, Wehrmeister MA, Pereira CE, Wagner FR, Silva Jr ET, Carvalho FC. Using aspect-oriented concepts in the requirements analysis of distributed real-time embedded systems. In: Proc. of the 2007 Working Conf. on Embedded System Design: Topics, Techniques and Trends. Irvine: Springer, 2007. 221–230. [doi: 10.1007/978-0-387-72258-0_19]

[114] De Lemos R, Saeed A, Anderson T. A train set as a case study for the requirements analysis of safety-critical systems. The Computer Journal, 1992, 35(1): 30–40. [doi: 10.1093/comjnl/35.1.30]

[115] Tjell S, Fernandes JM. Expressing environment assumptions and real-time requirements for a distributed embedded system with shared variables. In: Proc. of the 2008 IFIP Working Conf. on Distributed and Parallel Embedded Systems. Milano: Springer, 2008. 79–88. [doi: 10.1007/978-0-387-09661-2_8]

[116] Martins LEG, De Oliveira T. A case study using a protocol to derive safety functional requirements from fault tree analysis. In: Proc. of the 22nd Int'l Requirements Engineering Conf. (RE). Karlskrona: IEEE, 2014. 412–419. [doi: 10.1109/RE.2014.6912292]

[117] Hansen KM, Ravn AP, Stavridou V. From safety analysis to software requirements. IEEE Trans. on Software Engineering, 1998, 24(7): 573–584. [doi: 10.1109/32.708570]

[118] Markose S, Liu XQ, McMillin B. A systematic framework for structured object-oriented security requirements analysis in embedded systems. In: Proc. of the 2008 Int'l Conf. on Embedded and Ubiquitous Computing. Shanghai: IEEE, 2008. 75–81. [doi: 10.1109/EUC.2008.92]

[119] Roudier Y, Idrees MS, Apvrille L. Towards the model-driven engineering of security requirements for embedded systems. In: Proc. of the 3rd Int'l Workshop on Model-driven Requirements Engineering (MoDRE). Rio de Janeiro: IEEE, 2013. 55–64. [doi: 10.1109/MoDRE.2013.6597264]

[120] Saeed A, De Lemos R, Anderson T. An approach for the risk analysis of safety specifications. In: Proc. of the 9th IEEE Annual Conf. on Computer Assurance. Gaithersburg: IEEE, 1994. 209–221. [doi: 10.1109/CMPASS.1994.318451]

[121] Sunindyo W, Melik-Merkumians M, Moser T, Biffl S. Enforcing safety requirements for industrial automation systems at runtime position paper. In: Proc. of the 2nd Int'l Workshop on Requirements@Run.Time. Trento: IEEE, 2011. 37–42. [doi: 10.1109/ReRunTime.2011.6046246]

[122] Zafar S, Dromey RG. Integrating safety and security requirements into design of an embedded system. In: Proc. of the 12th Asia-Pacific Software Engineering Conf. Taipei: IEEE, 2005. 8. [doi: 10.1109/APSEC.2005.75]

[123] Colaço JL, Pagano B, Pouzet M. SCADE 6: A formal language for embedded critical software development. In: Proc. of the 2017 Int'l Symp. on Theoretical Aspects of Software Engineering (TASE). Sophia Antipolis: IEEE, 2017. 1–11. [doi: 10.1109/TASE.2017.8285623]

[124] Radjenovic A, Paige R. Architecture description languages for high-integrity real-time systems. IEEE Software, 2006, 23(2): 71–79. [doi: 10.1109/MS.2006.36]

[125] Dajsuren Y, Van Den Brand M, Serebrenik A, Huisman R. Automotive ADLS: A study on enforcing consistency through multiple architectural levels. In: Proc. of the 8th Int'l ACM SIGSOFT Conf. on Quality of Software Architectures. Bertinoro: ACM, 2012. 71–80. [doi: 10.1145/2304696.2304710]

[126] Liu SY, Asuka M, Komaya K, Nakamura Y. Applying SOFL to specify a railway crossing controller for industry. In: Proc. of the 2nd IEEE Workshop on Industrial Strength Formal Specification Techniques. Boca Raton: IEEE, 1998. 16–27. [doi: 10.1109/WIFT.1998.766294]

[127] Wang JC, Liu SY, Qi Y, Hou D. Developing an insulin pump system using the SOFL method. In: Proc. of the 14th Asia-Pacific Software Engineering Conf. Nagoya: IEEE, 2007. 334–341. [doi: 10.1109/ASPEC.2007.31]

[128] Chen XH, Liu J, Mallet F, Jin Z. Modeling timing requirements in problem frames using CCSL. In: Proc. of the 18th Asia-Pacific Software Engineering Conf. Ho Chi Minh City: IEEE, 2011. 381–388. [doi: 10.1109/APSEC.2011.30]

[129] Chen XH, Liu QQ, Mallet F, Li Q, Cai SB, Jin Z. Formally verifying consistency of sequence diagrams for safety critical systems. Science of Computer Programming, 2022, 216: 102777. [doi: 10.1016/j.scico.2022.102777]

[130] Vilela J, Castro J, Martins LEG, Gorschek T. Integration between requirements engineering and safety analysis: A systematic literature review. Journal of Systems and Software, 2017, 125: 68–92. [doi: 10.1016/j.jss.2016.11.031]

[131] Medikonda BS, Panchumarthy SR. A framework for software safety in safety-critical systems. ACM SIGSOFT Software Engineering Notes, 2009, 34(2): 1–9. [doi: 10.1145/1507195.1507207]

[132] Selić B, Gérard S. Modeling and Analysis of Real-time and Embedded Systems with UML and MARTE: Developing Cyber-physical Systems. Amsterdam: Elsevier, 2014. [doi: 10.1016/C2012-0-13536-5]

[133] Tsai JJP, Liu AL. Experience on knowledge-based software engineering: A logic-based requirements language and its industrial applications. Journal of Systems and Software, 2009, 82(10): 1578–1587. [doi: 10.1016/j.jss.2009.03.019]

[134] Gu B, Yu B, Dong XG, Li XF, Zhong RM, Yang MF. Intelligent program synthesis techniques: Literature review. Ruan Jian Xue Bao/Journal of Software, 2021, 32(5): 1373–1384 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6200.htm [doi: 10.13328/j.cnki.jos.006200]

[135] Yang MF, Gu B, Duan ZH, Jin Z, Zhan NJ, Dong YW, Tian C, Li G, Dong XG, Li XF. Intelligent program synthesis framework and key scientific problems for embedded software. Chinese Space Science and Technology, 2022, 42(4): 1–7 (in Chinese with English abstract). [doi: 10.16708/j.cnki.1000-758X.2022.0046]

附中文参考文献:

[18] 王飞, 杨志斌, 黄志球, 周勇, 刘承威, 章文炳, 薛垒, 许金淼. 基于限定自然语言需求模板的 AADL 模型生成方法. 软件学报, 2018, 29(8): 2350–2370. http://www.jos.org.cn/1000-9825/5530.htm [doi: 10.13328/j.cnki.jos.005530]

[28] 舒风笛, 毋国庆, 王敏. 面向嵌入式实时软件系统需求工程环境——SREE. 计算机科学, 2002, 29(4): 4–8, 14. [doi: 10.3969/j.issn.1002-137X.2002.04.002]

[29] 舒风笛, 毋国庆, 李明树. 面向嵌入式实时软件的需求规约语言及检测方法. 软件学报, 2004, 15(11): 1595–1606. http://www.jos.org.cn/1000-9825/15/1595.htm

[41] 顾斌, 董云卫, 王政. 面向航天嵌入式软件的形式化建模方法. 软件学报, 2015, 26(2): 321–331. http://www.jos.org.cn/1000-9825/4784.htm [doi: 10.13328/j.cnki.jos.004784]

[43] 冯劲草. 面向嵌入式设计需求的形式化建模与验证 [博士学位论文]. 上海: 华东师范大学, 2022. [doi: 10.27149/d.cnki.ghdsu.2022.004041]

[44] 王尚, 冯劲草, 诸嘉逸, 黄怿豪, 郑寒月, 徐想容, 缪炜恺, 张翔, 蒲戈光. 面向轨交控制软件需求模型的量纲分析方法. 计算机学报, 2020, 43(11): 2152–2165. [doi: 10.11897/SP.J.1016.2020.02152]

[84] 杜杠, 林佳. 实时嵌入式软件需求描述框架探索. 质量与可靠性, 2008, (1): 47–50.

[100] 胡指铭, 黄丽桃, 赵涌鑫. 面向航天型号软件的混成建模语言研究. 空间控制技术与应用, 2021, 47(2): 25–31. [doi: 10.3969/j.issn.1674-1579.2021.02.004]

[134] 顾斌, 于波, 董晓刚, 李晓锋, 钟睿明, 杨孟飞. 程序智能合成技术研究进展. 软件学报, 2021, 32(5): 1373–1384. http://www.jos.org.cn/1000-9825/6200.htm [doi: 10.13328/j.cnki.jos.006200]

[135] 杨孟飞, 顾斌, 段振华, 金芝, 詹乃军, 董云卫, 田聪, 李戈, 董晓刚, 李晓锋. 嵌入式软件智能合成框架及关键科学问题. 中国空间科学技术, 2022, 42(4): 1–7. [doi: 10.16708/j.cnki.1000-758X.2022.0046]

陈小红(1982－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为需求工程, 形式化方法, 安全攸关系统.

金芝(1962－), 女, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为需求工程, 知识工程.

刘少彬(1998－), 男, 硕士生, 主要研究领域为需求工程.