面向轨交控制软件需求模型的量纲分析方法

王 尚[1)] 冯劲草[1)] 诸嘉逸[1)] 黄怿豪[1)] 郑寒月[1)]
徐想容[1)] 缪炜恺[1)] 张 翔[2)] 蒲戈光[1),3)]

[1)](上海市高可信计算重点实验室 上海 200062)
[2)](军事科学院国防科技创新研究院 北京 100000)
[3)](上海工业控制系统安全功能型平台 上海 200062)

摘 要 嵌入式控制软件是当前诸多控制系统的核心部件.各类安全攸关系统,例如轨道交通系统、航空航天系统和核电控制系统等,其内嵌的控制软件的功能是否正确、安全直接关系到人们的生命与财产安全.经过长期的研究和实践,学术界和工业界都意识到系统需求作为软件工程生命周期的起始点,是确保软件质量的根本途径.因此,如何从源头上保障软件开发流程的正确性一直以来是软件工程领域重要的研究课题.然而,在工业界的软件开发中,系统需求的正确性只能靠人工审查来保证,人工审查自然语言系统需求的方法不但低效,而且无法确认需求的正确性,仍缺少一种对需求进行确认的形式化建模分析方法.需求确认是保障所构建的形式化规约质量的重要工程活动,而量纲分析作为需求确认中的一个重要手段,可以检查出因量纲的错误定义而产生的潜在缺陷,减少因量纲使用不当所造成的错误.基于此,我们针对轨道交通领域的特点,提出了一种基于形式化工程方法的需求建模与量纲分析研究,结合软件需求工程的基本原理,从原始需求出发逐步完成需求规约的构建,并给出了需求规约确认的其中一种方法,以确认软件需求规约准确、充分地描述人们对软件期望的功能,为该领域的需求分析提供保障.本文的主要贡献是:(1)提出了一种嵌入式控制软件需求建模的形式化工程方法;(2)提出了一种列车控制领域专用的需求描述语言 ATPRDL(Automatic Train Protection Requirement Description Language),并对相应的需求文档进行建模;(3)针对嵌入式控制软件的特点,提出了一种领域专用的面向需求层的量纲分析方法,并构建了领域适用的量纲系统;(4)开发了一款工具用于支撑该方法的实际应用,使工程师可以快捷地进行量纲分析.通过应用本文提出的方法,我们对真实的轨道交通列车控制软件的需求进行了实验,实验结果表明此方法具有良好的可用性,并在实际的列车控制软件开发项目中,展现了优于工业界传统手工审查方法的缺陷发现能力和时间效率.

关键词 软件工程;嵌入式软件;形式化方法;需求建模;量纲分析;需求确认
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2020.02152

A Dimensional Analysis Method for the Requirements Model of Railway Control Software

WANG Shang[1)] FENG Jin-Cao[1)] ZHU Jia-Yi[1)] HUANG Yi-Hao[1)] ZHENG Han-Yue[1)]
XU Xiang-Rong[1)] MIAO Wei-Kai[1)] ZHANG Xiang[2)] PU Ge-Guang[1),3)]

[1)](*Shanghai Key Laboratory of Trustworthy Computing*, *Shanghai* 200062)
[2)](*National Defense Science and Technology Innovation Institute*, *Academy of Military Sciences*, *Beijing* 100000)
[3)](*Shanghai Trusted Industrial Control Platform Co.*, *Ltd*, *Shanghai* 200062)

Abstract Embedded control software is the kernel component of today's control systems. For the safety-critical control systems, such as the railway system, aerospace control system and

收稿日期:2019-12-24;在线发布日期:2020-04-13.本课题得到国家自然科学基金(61872144)、国家自然科学基金青年基金(61402178)资助.
王 尚,硕士,中国计算机学会(CCF)会员,主要研究方向为需求工程、形式化工程方法.E-mail:466168225@qq.com.**冯劲草**,博士,中国计算机学会(CCF)会员,主要研究方向为需求工程、形式化工程方法.**诸嘉逸**,博士,中国计算机学会(CCF)会员,主要研究方向为深度神经网络的验证.**黄怿豪**,博士,中国计算机学会(CCF)会员,主要研究方向为需求工程、形式化工程方法.**郑寒月**,博士,中国计算机学会(CCF)会员,主要研究方向为需求工程、自然语言处理.**徐想容**,硕士,主要研究方向为自然语言处理.**缪炜恺**,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为形式化工程方法.**张 翔**(通信作者),硕士,助理研究员,主要研究方向为软件质量保障.E-mail:jarchers@163.com.**蒲戈光**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为程序分析、软件验证.

nuclear power control system, the function and safety of the control software embedded in them are directly related to the safety of people's lives and property. After long-term research and practice, both academia and industry realized that system requirements are the starting point of the software engineering life cycle and the fundamental way to ensure software quality. Therefore, how to ensure the correctness of software development process from the source has been an important research topic in the field of software engineering. However, in the software development of industry, the correctness of system requirements can only be guaranteed by manual review. Manual review of system requirements described in natural language is not only inefficient, but also unable to ensure the correctness of requirements. There is still a lack of a formal modeling analysis method to validate requirements. Requirements validation is an important engineering activity to ensure the quality of the formalized specifications constructed. As an important method of requirements validation, dimensional analysis can check out potential defects caused by wrong definition of dimension and reduce errors caused by improper use of dimensions. At present, the dimensional analysis in computer systems is mostly based on the code level and not applied in the requirements field, while some hidden errors in safety-critical embedded control systems are mostly caused by the wrong use of dimensions in requirements, which leads to software development errors from the source. Therefore, we propose a requirements modeling and dimension analysis method by according to the characteristics of the railway, which is based on formal engineering method. First, the requirements engineer converts the original requirements specifications into the requirements specifications described using the domain-specific requirements description language, then automatically extracts the requirements model using the tool. Finally, carries out the subsequent dimensional analysis on the basis of the requirements model. This work combines with the basic principle of software requirements engineering, starting from the original requirements gradually build complete specifications. In this paper, one of the methods of requirements specifications validation is given to confirm the accuracy of software requirements specifications and to fully describe the expected functions of software, so as to guarantee the requirements analysis in the field. The main contributions of this paper are listing as follows: (1) Propose a formal engineering method for requirement modeling of embedded control software. (2) Propose a special requirement description language, ATPRDL(Automatic Train Protection Requirement Description Language) for the field of train control, and models the corresponding requirements specifications. (3) According to the characteristics of embedded control software, we proposed a domain-specific dimensional analysis method of requirements oriented layer, and construct a domain-specific dimension system. (4) We developed a tool to support the practical application of the method, enabling engineers to conduct dimensional analysis quickly. By applying the method proposed in this paper, we experiment on the requirements of the real railway control software. The experimental results show that this method has a good usability. In the actual train control software development project, the defect detection ability and time efficiency are shown to be better than the traditional manual review methods.

Keywords　software engineering; embedded software; formal method; requirements modeling; dimensional analysis; requirements validation

1　引　言

嵌入式控制软件是各种机械设备控制系统的重要组成部分，在很多领域中都有广泛的应用. 在军事装备领域，嵌入式控制软件应用于无人机、导弹等武器装备；在工业制造领域，嵌入式控制软件应用于数控机床等加工设备；在人类生活领域，嵌入式控制软

件应用于高速列车、航空飞机等交通工具.这些设备和工具能否正常使用依赖于设备内部的嵌入式控制软件是否能正常运行.如果嵌入式控制软件没有按照预期的设计进行工作,即使是发生很小的错误,也将引起严重的人员伤亡和财产损失.因此,我们有必要尽最大努力去保障这些嵌入式控制软件功能的正确性和安全性.

在经典的软件工程理论中,V 模型[1]广为人知,嵌入式控制软件的开发也遵守 V 模型的开发流程与规范.随着软件工程在工业实践中的逐步发展,工业界越来越多的人认识到系统需求的正确性是后续开发流程中的重中之重.系统需求作为控制软件开发流程的起始点和测试的终点,其重要性不言而喻.如果需求文档中原本就存在一个隐藏的缺陷,那么根据存在隐藏缺陷的系统需求来开发出的控制软件的安全性是很难保障的.尽管后续有很多验证方法可以大概率地检查出需求中的缺陷,但花费的代价往往是令人无法接受的.更为严重的是,后续的测试也无法保障能够发现需求中本身就存在的所有缺陷.因此,从源头上保证系统需求的正确性,是保障嵌入式控制软件正确性的重要基石.

量纲指的是将一个物理导出量用若干个基本量的乘方之积表示出来的表达式.量纲分析是指通过识别基本量和计量单位,并进行跟踪计算或比较的方式来分析不同物理量之间的关系的一种方法.在计算机程序中,量纲分析可以用来验证算法的语义一致性[2].

在嵌入式系统中,系统需求所描述的软件功能与物理环境息息相关,因此,对系统需求进行量纲分析是一种保障其正确性的有效方法.量纲的错误定义一般很难发现,然而往往由于对量纲的不重视造成了巨大的损失,如美国宇航局的火星气候探测者号任务失败的主要原因就是量纲的错误使用,其飞行系统软件使用公制单位牛顿计算推进器动力,而地面人员输入的方向校正量和推进器参数则使用英制单位磅力,从而导致探测器进入大气层的高度有误,最终瓦解碎裂.在实践中发现,量纲的正确性并不能保障系统需求是正确的,但是量纲的错误一定代表系统需求的不正确性.因此,构建一种能有效地确认系统需求中量纲正确性的分析方法,对于工业界来说有着重要意义.

对于控制软件而言,使用自然语言书写的需求文档难以进行量纲分析,因为在实际撰写系统需求的过程中,往往都是分模块化的多人合作完成的,在撰写需求时为了保障需求的一致性,往往采用自然语言来进行系统需求的撰写,以方便需求工程师之间的沟通及需求文档的版本迭代.然而,每个人对系统中各个变量的物理含义的理解不尽相同,这就造成了在系统需求中会人为的添加一些量纲错误.因此,为了对系统需求进行量纲分析,使用领域专用的需求描述语言来精确化地描述系统需求是不可缺少的一步.

我们与专注于轨道交通控制系统集成商卡斯柯信号有限公司合作,提出了一种面向嵌入式控制软件需求模型的量纲分析方法,具体研究内容包括:(1)以形式化需求描述语言为基础,建立系统需求模型;(2)提供一种基于模型的量纲分析方法,辅助需求工程师进行需求撰写与审查;(3)开发相应的工具并在编译阶段对系统需求进行量纲分析.

本文第 2 节主要介绍需求领域中量纲分析的相关现状;第 3 节介绍量纲分析方法的整体框架;第 4~7 节详细描述基于形式化工程方法的关键技术;第 8 节介绍在工业界中实际运用该方法的实验结果;最后是对本文的总结及未来可研究方向.

2 国内外相关研究

目前,在轨交领域已经有一些工作是运用形式化的方法对列车控制软件进行需求的确认[3-5],但是这些工作并未涉足对量纲的分析.工业界常在 SCADE 平台①[6]上使用状态图来描述需求,并用形式化方法来验证模型的一致性.但 SCADE 本身偏重于设计,而非描述需求之间的互相关系,其无法对需求中的量纲进行有效的分析,只能针对模型中的变量进行类型验证.Liu 教授提出用需求规范语言来描述需求[7],精化需求的方法使得需求的可分析性得到了进一步的改善,但是对于量纲系统的分析不是该方法的重点.

形式化建模是在安全攸关领域中根据诸多工业界的标准所推荐的需求描述技术,也是学术界推崇的精确描述软件需求的方法.然而,形式化建模方法对于不同领域更多地以定制化的形式出现,为特定领域软件需求描述予以支持.形式化语言有 Z[8]、B[9]方法和 Event-B[10]等,均在软件工程领域确认了较大的影响力.但是,在需求领域中的量纲分析并没有给出可行的方案.

① SCADE. http://scade.sharewarejunction.com/

经典的量纲分析方法在程序语言中的应用可以追溯到 1978 年[11]，在那之后，各种程序语言上的量纲分析被逐渐提出[12]，其中最为出名的是 Ada 的量纲分析[13]. 之后，在各类程序语言上的量纲分析陆续被开发出来[14-16]. Osprey[17] 提出了一种优美的解决量纲分析的方法，通过对源程序添加单位标注后生成约束方程，最后通过高斯消去来对方程进行求解.

目前在计算机系统中的量纲分析大多是面向代码层的量纲分析，并没有应用在需求层面，而在安全攸关的嵌入式控制软件中，需求中量纲的错误使用会导致软件开发从源头上就出现错误. 量纲分析应用在需求领域的难点在于：

(1) 控制软件中调用大量的具有实际物理含义的计算公式，牵扯到很多不同的量纲；

(2) 控制软件中变量之间的互相调用非常普遍，并且交互关系错综复杂，使得人工检查量纲的工作量陡增；

(3) 需求工程师撰写需求之时难以保证其理解的量纲与实际存在的量纲的一致性；

(4) 重视程度不够，因量纲引发的错误都被归咎于笔误，事实上需求中量纲的错误会直接导致软件开发中后续流程的错误，在未来修复的成本较大；

(5) 控制软件中的量纲系统与实际物理环境中使用的国际单位量纲系统并不完全等同，在软件中使用的量纲系统往往是物理量纲系统的变种，工程师在使用时自身未察觉量纲系统的变换；

(6) 现有的量纲分析基本上是基于代码层面的量纲分析，而没有在需求层面对量纲进行分析.

综上所述，目前尚没有成熟的可以直接用在轨道交通需求领域的量纲分析方法，而在需求文档的确认阶段进行量纲分析是至关重要的. 基于此，我们为了解决由于量纲错误所导致的一系列问题，提出了一种基于需求模型的量纲分析方法，同时开发了相对应的工具运用于实际.

3　方法框架

从软件工程的角度来说，需求作为软件工程开发流程的第一步，其正确与否直接影响最终软件的正确性，因此，从源头上保证需求的正确性是至关重要的. 而目前在工业界中，需求文档的模糊性和二义性是导致很多软件项目最终无法满足用户需求的主要原因. 针对这一现状，我们以自然语言撰写的需求文档为出发点，提出一种领域专用的需求描述语言 ATPRDL，需求工程师按照 ATPRDL 的语法规范对系统需求进行精确的描述，得到形式化需求文档，在严格的数学基础上进行软件开发，以获得更好的软件性能.

我们所提出的方法由需求建模和量纲分析两个模块组成，此方法框架如图 1 所示. 需求工程师将原始的需求文档转化为使用 ATPRDL 语言描述的形式化需求文档后，使用工具自动抽取出需求模型，在需求模型的基础上进行后续的量纲分析.

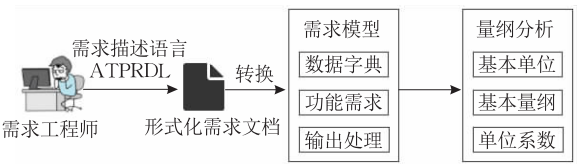

图 1　量纲分析方法框架

在形式化需求文档撰写完成后，抽取形式化需求文档，建立与需求描述相一致的形式化需求模型，以帮助需求工程师进一步明确需求，促进他们对需求的理解，本文后续的章节中将详细地介绍此需求模型. 在和轨道交通领域工程师合作交流的过程中，我们发现软件开发人员主要关心的是在需求模型构建完成后的问题，即抽取出的需求模型是否能准确地反映用户对软件功能的期望. 而在软件需求阶段，需求工程师最关心的是涉及到安全攸关功能的计算问题，比如在计算任务中相关变量的单位转换的正确性等. 然而，这些问题在实际嵌入式控制软件的运行中非常重要.

因此，我们提出了一种基于需求模型的量纲分析方法，从而帮助需求工程师更准确、更快速地进行需求的分析验证.

4　量纲分析概述

4.1　量纲的定义

在详细介绍基于需求模型的量纲分析方法之前，我们首先对量纲的定义进行解释. 国际单位制是一种通用的量纲系统，包含了 7 个基本单位和两个无量纲单位，从而派生出所有其它单位[18]. 在轨道交通领域的需求文档中所用的单位不多，且拥有其独特的量纲应用. 在 C 语言中进行的量纲分析方法 Osprey 虽然获得了巨大的成功，但是在特定领域内的适用性有待提高. TADA (Template meta programming bAsed Dimensional Analysis) 是

针对性的提出了优化方法，从而应用在 C++语言中的量纲分析[19]. TADA 带来的更大的启发则是我们可以针对特定领域来构建满足此领域所需要的量纲系统来进行量纲分析.

物理学中，将一个物理导出量用若干个基本单位的乘方之积表示出来的表达式，称为该物理量的量纲[20]. 量纲系统[21]中的几个最重要的元素为：基本单位(Base Unit)、量纲(Dimension)、基本量纲(Base Dimension)、单位系数(Unit Factor)和无量纲系数(Dimensionlessnumber). 其中，基本单位为整个量纲系统的基本组成单元，如表示时间的基本单位秒(s). 量纲均由基本单位的幂次方的乘积组成，如 $mm\times s^{-2}$(毫米每秒的平方). 基本量纲则为量纲系统中变量赋值使用的单位，如 mm/s. 单位系数为不同单位之间在转换时所使用的常量，如 1 s=1000 ms 中，单位系数为 1000. 无量纲系数指的是量纲运算只参与数值的计算而不参与量纲计算的系数，如 5 mm×3.24 中的 3.24 即为无量纲系数.

在轨道交通控制软件的需求文档中，我们定义了 5 个用于变量赋值的最基本的量纲，而其中使用的最基本单位只有 3 个，分别为表示长度的单位 mm，表示时间长短的单位 ms，表示周期长度的单位 cycle.

结合 ATPRDL 语言和量纲系统的特点，我们针对此领域内的量纲进行了基本的定义：

Dimension System::=(Base Unit, Dimension, Base Dimension, Unit Factor, Dimensionlessnumber)；

Base Unit::={mm, ms, cycle}；

Dimension::= $mm^p\times ms^q\times cycle^r$(其中 p,q,r 都为整数)；

Base Dimension::={mm/s, mm/s^2, μm, ms, cycle}；

Unit Factor::={mm_To_μm, μm_To_mm, s_To_ms, ms_To_s, cycle_To_ms}；

Dimensionlessnumber::=R(R 为实数).

基于此量纲系统，我们可以抽取 ATPRDL 语言中的功能需求模块进行量纲分析.

4.2 量纲分析与类型检查

虽说量纲分析与类型检查此两项工作的形式有所相同，但两者对于我们实际要解决的问题有着不同的意义. 本小节我们将对量纲分析和类型检查做出简单的对比分析.

类型检查主要是为了判断变量或者参数的实际类型和声明的类型是否匹配. 它由类型检查器完成，该检查器验证构造类型(如常量、变量、数组、列表、对象)是否与其使用上下文中所预期的相匹配. 类型是数据的一种属性，包括整型、浮点型、布尔型、字符型等. 例如，表达式 A+B=C，“+”运算符两边的 A 和 B 都是整型，那么 C 也必须是整型. 量纲分析主要是判断基本数量(如长度、质量、时间和电荷)和度量单位(如英里 vs. 公里或磅 vs. 公斤)，通过确定和跟踪基本数量和度量单位来分析不同物理量之间的关系.

量纲分析多用于物理和数学领域[22]，虽然类型检查和量纲分析的形式比较雷同，前者检查数据的属性，后者分析物理量的属性，但两个问题并不相干，如同程序员不去过度关心数据的单位、物理学家不过度区分数据的类型是整型或浮点型等，类型检查和量纲分析可以说是他们为各自领域的特有问题特意想出来的方法. 此外，类型检查和量纲分析的具体操作在细节上也有所不同. 例如，类型检查有很多规则，int 和 double 可以进行乘除运算，有些语言 int 和 char 不能相加运算，而量纲分析除了不同量纲之间不能相加，其余没有过多的限制. 除此之外，类型检查的类型是固定的，而量纲分析可以通过乘除运算来引入新的量纲.

所以，在此篇论文的课题研究中，我们认为类型检查和量纲分析是两个不同的问题，后续我们会做类型检查的工作并与量纲分析做出对比.

5 需求描述语言 ATPRDL 概述

5.1 ATPRDL 需求描述语言

在轨道交通领域，需求工程师使用自然语言撰写需求文档的初衷是使需求有更好的可读性. 然而对于不同背景的系统开发人员而言，列车控制领域的许多专有名词过于陌生. 他们较难理解这些专有名词在自然语言需求文档中表达的含义，且由于自然语言的二义性，不同背景的工程师在阅读需求时会有不同的理解，使得实现的系统发生偏差. 因此，需提供精确的需求描述语言，比如使用基于数学定义的形式化需求规格说明. 经典的形式化规约语言如 Z 与 B 等人建立的软件需求模型采用基于一阶谓词逻辑和集合论的数学符号来描述软件需求. 由于嵌入式控制软件日益复杂，参与开发的人员背景多元，对于软件开发者来说，单纯以数学符号书写、以文本形式展现的形式化规约对于普通软件工程人员而言难度过大，缺乏可读性. 而 UML 和 AADL 等图形化符号又因缺乏严格的形式化语义难以胜任

精确化需求描述.

为了解决自然语言需求文档可能导致的问题，首先需设计一种列车控制系统领域专用的、无二义性的、可靠的需求描述语言，帮助轨交公司对其需求文档进行形式化建模.

通过对轨交领域卡斯柯信号有限公司列车控制软件需求文档的调研，我们发现轨交控制软件是一种典型的嵌入式控制系统，具备以下特征：

(1) 周期驱动. 列车控制软件是一种不会终止的反应式系统，在系统运行时，处于不同的状态会执行不同的计算任务；

(2) 面向计算. 列车控制软件的核心是各部件根据实时物理环境执行特定的计算任务，如自动列车防护模块执行的计算任务为计算当前列车所允许的最大行驶速度；

(3) 状态机驱动. 列车在不同的场景下需要进行状态的切换，从而完成各种满足需求的控制效果.

结合列车控制软件的特征，我们设计了一种列车控制领域专用的轻量级的形式化需求描述语言 ATPRDL. 为了符合工程师的使用习惯，我们设计的 ATPRDL 是在 Python 语言的基础上，经过一些语法和语义上的修改而提出的一种新的语言. 使用 ATPRDL 描述的需求文档，可以精确的描述需求，从而消除原本自然语言需求文档的二义性.

5.2　ATPRDL 语法

在详细介绍形式化规约的关键技术之前，首先给出本文所提出的形式化需求描述语言 ATPRDL 的语法定义.

ATPRDL 描述的列车自动防护软件的需求模型由数据字典(DataDict)和功能需求(FunctionalRequirement)两部分组成. 其中，ATPRDL 的 BNF (Backus-Naur Form 巴克斯-诺尔范式)定义如下：

ATPRDL ::= (DataDict, FunctionalRequirement)；

DataDict ::= (Name, VariableName, Type, Unit, Value)；

FunctionalRequirment ::= (Title, NatureLanguageDescription, DefStmt | StateStmt)；

DefStmt ::= (FunctionName, PreCondition, CFG, ReturnStmt, PostCondition)；

StateStmt ::= (PreCondition, CFG, PostCondition)；

CFG ::= CompoundStmt；

PreCondition ::= ExpressionStmt(布尔表达式)；

CompoundStmt ::= CompoundStmt | if_stmt | for_stmt | assignment_stmt | continue_stmt；

ReturnStmt ::= "return" ExpressionName；

PostCondition ::= ExpressionStmt.

(1) 数据字典存储需求文档中出现的所有变量，每个变量都包含有中文名称、变量名、类型名、单位及默认值.

(2) 功能需求中包含了需求文档所有需要详细描述的功能需求，是需求文档的核心部分. 功能需求包含需求编号(Title)、自然语言描述的需求(NatureLanguage)、计算任务语句(DefStmt)或状态迁移语句(StateStmt).

(3) DefStmt 为列车控制软件的计算任务语句. 该语句包含任务名称(Function)、执行该计算任务所需要满足的前置条件(PreCondition)、计算任务的控制流图(CFG)、返回语句(ReturnStmt)以及完成任务后需要满足的后置条件(PostCondition). 由于在列车控制软件中所有计算的起始点和终止点均为信号变量的取值和计算，该语句与 python 中的函数定义不同的是此处声明的类型是控制系统中所使用的信号量，如"def BMbeaconReadAge(k):"表示 BMbeaconReadAge 这个信号量的计算任务，变量名后面括号中的 k 代表列车当前运行周期 k 的取值. 因为轨道交通信号系统中最核心的部分为关于信号的描述，且每个条目的需求文档均阐述的是各个信号变量的控制逻辑. 因此，我们使用 python 中类似定义函数的方法来定义使用的每个变量，函数体部分的复合语句详细地阐述具体的控制逻辑.

(4) StateStmt 为列车控制软件的状态迁移语句. 执行该迁移条件所需要满足的前置条件(PreCondition)，详细的迁移条件的控制流图(CFG)和完成任务后需要满足的后置条件(PostCondition). 与 DefStmt 不同的地方在于控制流逻辑的计算结果不需要返回，若满足迁移条件并经 PostCondition 验证后，会直接发生迁移操作.

(5) CFG 为控制流图，规定了系统在每个周期需要执行的计算任务，由复合语句(CompoundStmt)模型构成. 复合语句中包括复合语句、分支语句或简单语句. if_stmt、for_stmt、assignment_stmt 等为 python 中的基本操作语句，此处不再赘述. 详细的控制流图示例见第 6.2 节需求模型的构建部分.

5.3　ATPRDL 语义

在介绍形式化需求描述语言 ATPRDL 的语法定义之后，本小节将对 ATPRDL 的语义规则进行介绍. 首先，我们使用($ATPRDL, s, k, pc, l, t$)来表示操作语义.

(1) $ATPRDL$ 表示模型.

(2) s 表示系统当前的状态.

(3) k 表示系统当前所处的周期.

(4) $pc\in\{N\cup\{Start,Exit\}\}$是程序计数器,用于标记控制流的执行情况. N 是控制流中的位置标记,*Start* 和 *Exit* 分别代表单个计算任务的开始和结束.每个计算任务都是为了进行特定的计算而存在,由在实际运行中需要参与计算的具体任务的算法构成.

(5) $l\in\{Begin,Execution,End\}$记录系统在每个周期中处于哪个步骤. *Begin* 代表一个周期的开始,*Execution* 表示准备执行周期任务,*End* 代表一个周期的结束.周期作为一个基本时间单位,根据当前所处的特定状态,周期性地执行给定的计算任务.

(6) $t\in\{0,1\}$表示程序是否处于最后一个周期. 0 表示不是最后一个周期,1 表示是最后一个周期.

然后,我们定义该系统的语义规则.首先是系统进入(Enter),其语义如式(1)所示:

$$\frac{\perp}{\begin{array}{l}(ATPRDL,s,k,Start,Begin,t)\rightarrow\\(ATPRDL,s',k,pc',Execution,t)\end{array}} \tag{1}$$

当程序从第一个周期开始执行时,系统状态从当前的状态 s 转变为新状态 s'.因为系统周期在 *Enter* 中没有改变,所以当前周期 k 保持不变.由于系统开始执行,程序计数器改变为 pc',并开始记录程序执行位置.系统的步进信号从"*Begin*"转变为"*Execution*",t 保持不变.

系统执行(Execution)表示系统的执行过程,其语义如式(2)所示:

$$\frac{pre\_condition=\text{True},\ pc\neq Exit}{\begin{array}{l}(ATPRDL,s,k,pc,Execution,t)\rightarrow\\(ATPRDL,s',k,pc',Execution,t)\end{array}} \tag{2}$$

程序计数器 pc 的语义如式(3)所示:

$$\begin{cases}pc=Start\ \text{at the beginning of a computation task}\\pc'=Exit\ \text{at the end of a computation task}\end{cases} \tag{3}$$

在每个周期中,系统都要确保在执行计算任务过程中的前提条件得到满足.系统的状态和程序计数器 pc 在执行计算任务期间会发生变化,而周期 k、t 和步进信号保持不变.需要注意的一点是,在计算任务开始时 pc 的值为 *Start*,而在计算任务结束时 pc 的值变为 *Exit*.

系统完成一项计算任务并开始另一项计算任务的过程称为 Continue,其语义如式(4)所示:

$$\frac{pre\_condition=\text{True},\ l\neq End}{\begin{array}{l}(ATPRDL,s,k,Exit,Execution,t)\rightarrow\\(ATPRDL,s',k,Start,Execution,t)\end{array}} \tag{4}$$

当前提条件满足且程序未结束时,系统继续执行.计算任务之间的转换会更改系统的状态和程序计数器 pc 的值.系统状态从 s 转变为 s',pc 从上一个计算任务的 *Exit* 状态转变为新任务的 *Start* 状态.由于系统仍在执行且周期 k 保持不变,所以其它项保持不变.

在完成一个周期执行后,如果此时程序不处于最后一个周期,即 $t\neq1$ 时,并且满足后置条件,则程序将进入下一个周期.这个过程称为 Repeat,其语义如式(5)所示:

$$\frac{post\_condition=\text{True},\ t\neq1}{\begin{array}{l}(ATPRDL,s,k,Exit,End,0)\rightarrow\\(ATPRDL,s',k+1,Start,Begin,t)\end{array}} \tag{5}$$

在进行重复操作后,系统可以进入下一个周期.同时,系统状态转变为 s',周期计数器在先前值的基础上增加 1,系统退出先前的计算任务,并在新的周期中开始新的计算任务. *Start* 表示单个计算任务的开始,*Begin* 表示一个周期的开始,因此,pc 和 l 的值分别改变为 *Start* 和 *Begin*.

6 需求模型生成

6.1 抽取形式化需求文档

在轨道交通控制软件需求撰写的初期,需求工程师对于系统整体功能的设计及模块的划分尚不明晰,且需要不同系统组别的人互通有无.因此,前期修改的频率是极其高的,需求文档迭代的次数也非常多.在这样的情况下,尽管自然语言描述的系统需求容易产生模糊性和二义性,需求工程师仍会使用自然语言来撰写需求文档,以此来简单并快捷地表述自己的想法,从而导致在后期进行系统设计、实现和测试的阶段带来严重的问题.

基于自然语言需求的量纲分析是困难且难以实现的.如图 2 所示,在功能需求部分中用自然语言描述的需求片段,往往是工业界用来描述需求的方式,而用此需求描述方式是难以进行量纲分析的,但是对于经过形式化规约的需求是可以进行量纲分析的.通过应用我们提出的形式化工程方法,需求工程师可以采用 ATPRDL 将需求文档完全形式化,即通过 ATPRDL 将每个功能需求进行详细阐述,并将约束条件等予以形式化定义,从而解决对自然语言需求进行量纲分析的难题.在进行需求规约的过程中,需求工程师对需求会有更深层次的理解,对于一些有歧义的需求,在此过程中会逐渐显露.例如,

图 2 展示了系统需求中某个功能需求的形式化描述. 图中的“else：return None”片段考虑了 CoreId 没有取值的情况下 OtherCoreId 的情况，但是数据字典中并没有对 None 变量进行量纲的定义. 在量纲分析中，未对变量进行量纲的定义与无量纲系数的含义是不同的.

数据字典	变量名	变量注释	量纲	类型	…
	CoreID	车头号	无量纲系数	Enum	
	OtherCoreId	远端车头号	无量纲系数	Enum	
	END_1	常量值	无量纲系数	int	
	END_2	常量值	无量纲系数	int	

```
功能需求
  [System-Requirement-0001]
  远端车头号为一个特定信号变量，与车头号为互斥取值.
  PreConditon：CoreId is not None
  def OtherCoreId：
  if (CoreId is END_1)：
        return END_2
  elif (CoreId is END_2)：
        return END_1
  else：
        return None
  PostConditon：CoreId！=OtherCoreId
```

图 2　需求规约

下面我们以一个示例来说明从自然语言需求中抽取形式化需求文档的过程. 图 3 是卡斯柯信号有限公司初始的自然语言需求示例，第 1 行为该列车控制系统需求的标签，第 2 行至第 8 行为自然语言需求描述. 该需求示例对列车运行状态是滑动状态“SLIDING”还是靠站状态“COASTING”进行监测和处理，使用自然语言对需求进行描述无法做到那么精确.

[iTC_CC_ATP-SwRS-0100]
ATP software shall use the over-estimation model for trainmovement. The maximum and minimum train motion shall overestimate based on different state as follows：
COASTING. There is not sliding effect during on train coasting or motoring, so ATP need not to overestimate train motion.
SLIDING. If train slides or slips excessively, ATP shall consider odometer motion untrustworthy.

图 3　自然语言需求示例

通过对列车控制系统自然语言需求文档进行逻辑上的抽取，并映射于物理量，可以得到对应的半形式化需求. 半形式化需求与自然语言需求相比具有更清晰的逻辑结构，是向更精确的形式化需求过渡的中间产物，且该转化过程的代价在可控范围内. 图 4 是由图 3 转化而来的半形式化需求示例，第 1 行为该需求的标签，第 2 行至第 5 行为半形式化需求描述.

[iTC_CC_ATP-SwRS-0100]
The state transfers from “**COASTING**” to “**SLIDING**” when：
VariableA is in **COASTING** state and
VariableB is set to be flase and
VariableC is less than VariableD

图 4　半形式化需求示例

形式化需求描述如图 5 所示，该图由图 4 进一步转化得到.

```
[iTC_CC_ATP-SwRS-0100]
/*
COASTING. There is not sliding effect during on train coasting or
motoring, so ATP need not to overestimate train motion.
SLIDING. If train slides or slips excessively, ATP shall consider
odometer motion untrustworthy.
*/
if(VariableA==COASTING
   and VariableB(k)==False
   and VariableC(k)<VariableD(k))：
   VariableA(k)=SLIDING
```

图 5　形式化需求示例

图 5 所示的列车控制系统形式化需求示例包含三部分：需求标签、需求注释和需求主体.

第一部分是需求标签，指这条需求片段在该列车控制系统需求中的编号，即“[iTC_CC_ATP-SwRS-0100]”. 需求标签不可以重复，一个需求标签对应于一条特定需求，这为获取每一个特定需求模型提供便利.

第二部分是需求注释，这一部分是对需求主体的部分变量做出的解释，为原始的自然语言需求，方便工程师能够理解设计者的确切表述. 需求注释部分在建立模型的过程中不进行处理.

第三部分则是使用 ATPRDL 语言建立的需求主体部分. ATPRDL 语法基于 Python 语法，有着简单直观的阅读性.

6.2　需求模型的构建

在使用 ATPRDL 语言撰写完形式化需求文档后，我们使用第三方开源语法分析器 Antlr 来读取已定义好的形式化需求语法，根据语法和词法规则自动生成语法和词法分析器. 我们将形式化需求描述的主体部分作为输入，Antlr 生成的语法和词法分析器会按照 ATPRDL 中定义的语法层级和结构抽取出关键信息，并将这些信息构建为抽象语法树. 若输入的形式化需求描述部分不符合 ATPRDL 的语法规范，词法和语法分析器会在构建出错的语法树的相应结点记录错误信息. 然后，通过递归遍历的方法自顶向下搜索树节点. 当搜索到叶子结点时，将该结点存储的信息构建为自定义的需求模型，然后

自底向上完成需求模型的耦合，每次根据当前节点的不同种类，将依次搜索到的树节点构建为不同类别的需求模型. 当回到抽象语法树的根节点，即整棵抽象语法树都经过一次搜索后，完成对 ATPRDL 模型的建立.

需求模型的构建流程如图 6 所示. 构建完成的 ATPRDL 需求模型由两部分组成：数据字典(Data-Dictionary)，由数百条具体的功能需求组成的 ATP(Automatic Train Protection)列车自动防护需求描述文档.

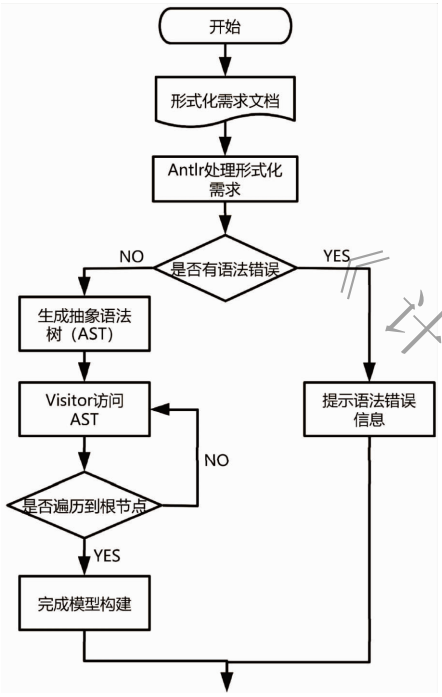

图 6 需求建模流程

ATPRDL 需求模型是一个面向列车自动防护的需求模型，如图 7 所示. 其中，数据字典模块是人工构建形式化需求文档之时构建完成的，包含整个需求文档中被定义的变量、存储了变量名、变量别名、类型、量纲等信息. ATP 列车自动防护模块包含了功能需求计算和输出处理两部分. 功能需求计算是整个模型的计算核心，其周期性的获取外部数据(如列车自动驾驶系统传递的信息)，并按照一定的逻辑来调用一系列的需求条目进行计算，最终产生的结果传递给输出处理部分. 输出处理是一个状态机，根据计算的结果与当前周期所处的状态来判断输出处理是否发生迁移等. 最后，我们基于此需求模型进行后续的量纲分析.

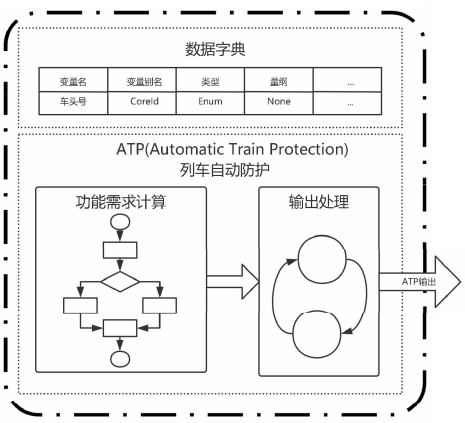

图 7 ATPRDL 模型图

7 基于需求模型的量纲分析

基于需求模型的量纲分析发生在编译阶段，在对 ATPRDL 编译完成且生成需求模型后，开始对需求文档进行量纲分析，最终得到量纲分析的错误列表. 量纲分析的算法如算法 1 所示.

算法 1. 量纲分析.

```
输入：FR：FunctionRequirmentSet
输出：ERR：ErrorList
INITIAL：ERR←∅
FUNCTION DimensionAnalysis(FR)
  FOR ALL fr∈FR DO
        CHECK(fr,ERR)
  END FOR
  RETURN ERR
END FUNCTION
FUNCTION CHECK(fr)
  IF FR.operater=='+' or FR.operater=='-' or
     FR.operater=='=' or FR.operater=='>' or
     FR.operater=='<' or FR.operater=='>=' or
     FR.operater=='<=' or FR.operater=='==' or
     FR.operater=='!=':
    IF (FR.leftOperand is dimensionless and
       FR.rightOperand is cycle) or
       (FR.leftOperand is cycle and
       FR.rightOperand is dimensionless)
          CONTINUE
    ELSE IF
          DimensionAnalysis(FR.leftOperand)!
          DimensionAnalysis(FR.rightOperand)
          ERR.add(FR)
          BREAK
    END IF
    Dimention←FR.leftOperand
  ELSE IF FR.operater =='×' or FR.operater =='/':
```

```
        Dimention ← DimensionAnalysis(FR.leftOperand) calculate DimensionAnalysis(FR.rightOperand)
    ELSE IF FR.operater == 'return':
        IF DimensionAnalysis(FR.leftOperand) != DimensionAnalysis(FR.FuncName)
            ERR.add(FR)
            BREAK
        END IF
        Dimention = FR.leftOperand
    ELSE IF FR.operater is exponentialFunc or FR.operater is logarithmicFunc or FR.operater is circularFunc
        IF DimensionAnalysis(FR.leftOperand) is not dimensionless
            ERR.add(FR)
            BREAK
        END IF
        Dimension = dimensionless
    END IF
    RETURN Dimention
END FUNCTION
```

按照量纲分析规则对每一个操作符两边的量纲进行单独的运算或比较. 在算法 1 的 *CHECK* 函数中，即量纲分析的核心部分，需要满足一些基本的和额外的规则，现阐述如下：

(1)“+”、“−”运算符两边变量的量纲必须相等才能进行计算；

(2)“×”、“/”运算符两边变量的量纲进行计算，产生新的变量量纲；

(3) 量纲单位为“cycle”的变量在进行“+”、“−”这两种运算时，在规则(1)的基础上，“cycle”还可以与无量纲单位进行计算；

(4)指数函数、对数函数和三角函数的因变量必须为无量纲系数才能进行计算；

(5)“return”语句替换为赋值语句进行计算；

(6)“=”、“>”、“<”、“>=”、“<=”、“==”、“!=”，当遇到以上运算符时，需要对运算符两边的量纲单位进行检查；

(7)当无法进行计算或检查结果不一致时，则终止检查并记录错误信息.

根据定义的量纲系统，按照量纲分析规则对全部的需求条目进行检查，其本质上是判断需求文档中的赋值语句、布尔表达式语句和多项式运算语句中的二元操作符的左操作数和右操作数之间的量纲是否可以按照上述给定规则进行计算. 从最小的二元操作符来保障量纲的正确性，即可保障所有语句的量纲的正确性.

在与卡斯柯信号有限公司实际的合作过程中，我们开发了相应的工具来验证了量纲分析，下面以一个例子来说明方法的可行性，如图 8 所示.

变量名	单位	…
MaxSpeed	mm/s	
LimitSpeed	μm/s	
MaxWheelMotion	mm	
WithoutMotionAvailable	mm	
ATP_CYCLE_TIME_MS	ms	

```
[System-Requirement-0002]
def MaxSpeed:
    …
LimitSpeed=(MaxWheelMotion+WithoutMotionAvailable)/
                ATP_CYCLE_TIME_MS
…
```

图 8　量纲分析举例

在使用 ATPRDL 语言将自然语言需求文档转化为形式化需求文档后，我们列举出其中的一条需求条目，如图 8 所示. 按照本章节提出的量纲分析方法对赋值语句“ LimitSpeed=(MaxWheelMotion+WithoutMotionAvailable)/ATP_CYCLE_TIME_MS”进行分析，其分析流程如下所示：

(1)根据量纲分析规则(1)，对“+”操作符两边的变量进行量纲分析，MaxWheelMotion 的单位为 mm，WithoutMotionAvailable 的单位为 mm，操作符两边变量的量纲相等，所以可进行加法计算. 由算法 1 可知，(MaxWheelMotion+WithoutMotionAvailable)的量纲为 mm；

(2)根据量纲分析规则(2)，对“/”操作符两边的变量进行量纲分析，(MaxWheelMotion+WithoutMotionAvailable)的量纲在第(1)步计算为 mm，ATP_CYCLE_TIME_MS 的量纲为 ms，经过“/”操作符后，产生新的变量量纲. 由算法 1 可知，(MaxWheelMotion+WithoutMotionAvailable)/ATP_CYCLE_TIME_MS 的量纲是 mm/ms；

(3)对“=”操作符两边的变量进行量纲分析可知，LimitSpeed 的量纲为 μm/s，(MaxWheelMotion+WithoutMotionAvailable)/ATP_CYCLE_TIME_MS 的量纲是 mm/ms. μm/s 与 mm/ms 不相等，因此，不可进行计算，记录此条错误并中断该需求条目的量纲分析.

8　实验与分析

为了使需求分析便于使用，我们开发了一个包含需求撰写及需求分析于一体的工具 Prema(Precise Requirement Editing, Modeling and Analysis)，在工

具的背后提供了一套可以完整描述轨道交通领域的需求描述语言，使用该工具可无障碍的撰写轨道交通领域的需求.此原型工具的开发得到了卡斯柯信号有限公司的大力支持，现已投入公司内部使用，以不改变工程师现有的开发习惯为出发点，提供实用性功能. Prema 的前端如图 9 所示. 用户可以在前端撰写需求，撰写完成后可以进行编译，并同步进行相对应的分析操作，最后形成各种视图供用户对需求进行确认和验证.

由于本文介绍的是基于需求模型的量纲分析，所以此处便不再展示工具的其它需求分析功能. 如图 10 所示，点击 Prema 工具的“Demensional Analysis”功能，选择量纲数据字典示例，便可对需求文档进行量纲分析. 量纲分析的结果示意图如图 11 所示.

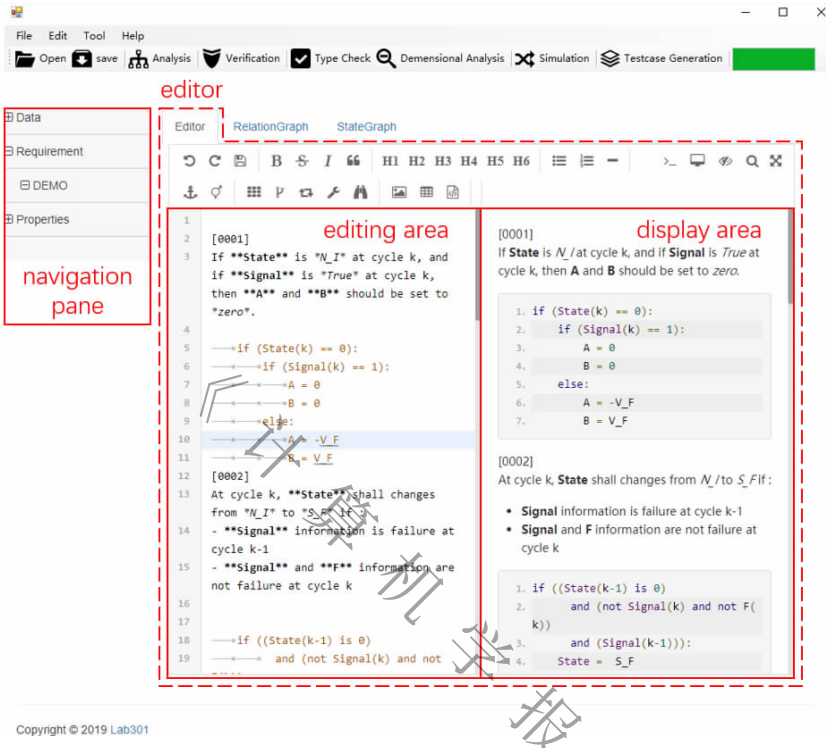

图 9 工具主界面

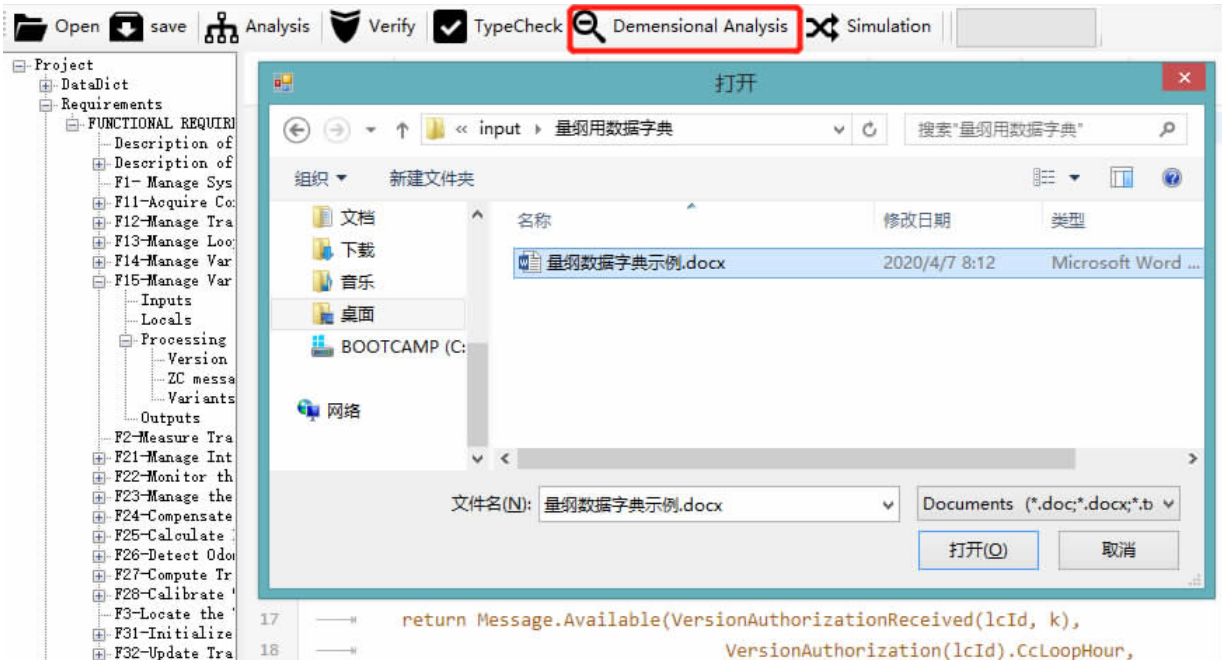

图 10 量纲分析

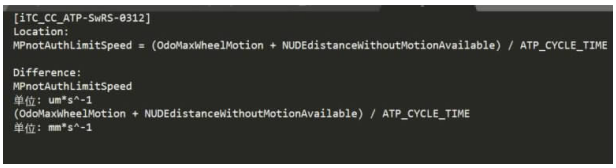

图 11　量纲分析结果示意图

在实际的实验中，我们针对轨交领域卡斯柯信号有限公司真实的 ATP 软件的某版需求文档进行了量纲分析. 输入文件由两部分组成，一部分是包含 455 条功能需求的需求文档，另一部分是包含 2776 条需求变量构成的数据字典. 在编译的过程中发现了 127 条语法错误，最终生成的需求模型中包含了 300 条功能需求和 28 条输出处理需求. 针对需求模型中的 300 条功能需求进行了量纲分析，发现了 27 条量纲错误信息. 后续通过人工根据报错结果进行比对，我们发现大部分的错误是因为系统预定义的一些常量值没有给定单位所导致. 通过更改数据字典再次进行分析，仅发现一条实际存在的量纲错误.

通过对量纲分析的结果进行归类总结，我们发现四种错误类型是造成轨道交通控制软件系统需求中量纲错误的主要原因，如下所示：

(1) 在需求文档中添加新的变量时，未更新数据字典；

(2) 变量之间单位转换的错误；

(3) 数组调用过程中数组索引值的错误；

(4) 基本单位 cycle 的错误使用.

针对以上四种错误类型，我们通过举例分析真实的错误案例分别对其进行解释说明.

错误类型(1)举例：

```
[iTC_CC_ATP-SwRS-0001]
def BMbeaconReadAge(k):
  if(Initialization
    ……
  else:
    return BMbeaconReadAge(k−1)+1
```

通过进行量纲分析，我们发现在需求文档中添加新的变量而未更新数据字典的错误情况有百余条. 如上述需求所示，[iTC_CC_ATP-SwRS-0001]为其中一条未完整定义变量量纲的需求条目，在添加一个新的变量 BMbeaconReadAge 时，没有更新数据字典，即没有此变量相对应的量纲，此时，就会导致错误类型(1)的出现.

错误类型(2)举例：

```
[iTC_CC_ATP-SwRS-0002]
def MSpeed:
    Lspeed=(MaxWheelMotion+MotionAvailable)/
           ATP_CYCLE_TIME_MS
```

在撰写需求时，如通常使用 mm 作为长度单位，而某个特殊变量使用 μm 作为长度单位，从而会忽略单位之间的转换，导致出现变量间单位转换的错误. 如上述需求条目[iTC_CC_ATP-SwRS-0002]所示，变量 MaxWheelMotion 的单位是 mm，变量 MotionAvailable 的单位是 μm，在进行 MaxWheelMotion+MotionAvailable 计算的时候，由于没有进行单位之间的转换，导致错误类型(2)的出现.

错误类型(3)举例：

```
[iTC_CC_ATP-SwRS-0003]
MinCogCalibrationMeasured= CaliMinRatio * ATPsetting. MeterCaliMaxMinCalibration[1][| CalibrationMeasurementStartPositionMin(k) − CogPositionAfterTopLoc (k) | − ATPsetting.OdoCaliCogCounterMin]
MaxCogCalibrationMeasured= CaliMaxRatio * ATPsetting.MeterCaliMaxMinCalibration[0][| CalibrationMeasurementStartPositionMax(k)−CogPositionBeforeTopLoc(k)| − ATPsetting. OdoCaliCogCounterMin]
```

如上述需求条目[iTC_CC_ATP-SwRS-0003]所示，此需求示例为一个典型的、在数组调用过程中出现的数组索引部分的错误. 通过查阅数据字典可以得知：数组 MeterCaliMaxMinCalibration[0…1][0…CALI_TA-BLE_LENGTH]的量纲为 mikron，而此数组的其中一个索引 | CalibrationMeasurementStartPositionMin(k)−CogPositionAfterTopLoc(k)| − ATPsetting.OdoCaliCogCounterMin 代表的含义是齿数，所计算出的齿数的量纲也为 mikron，由于在数组的调用过程中，索引部分的值必须是无量纲系数，带有量纲的变量不可作为索引值. 因此，此处导致错误类型(3)的出现.

错误类型(4)举例：

```
[iTC_CC_ATP-SwRS-0004]
def BMbeaconVariantValue(lineSection,VarIndex,k):
    if( BmbeaconReadAge (k) > ATPsetting. VariantsBMfullValidityTime):
    return False
    else:
        ……
```

基本单位 cycle 表示执行的周期数，但是在少部分需求中它被当作时间单位来使用，并被错误地当作系统执行一个周期所花费的时间参与计

算.在需求条目[iTC_CC_ATP-SwRS-0004]中,变量BMbeaconReadAge用来表示读取BM信标到当前的时间,单位为ms,而变量VariantsBMfullValidityTime的单位为cycle,在此计算过程中cycle被当作时间单位来使用,从而导致错误类型(4)的发生.

以上为我们通过大量的实验总结出的四种错误类型.对于那些经验相对不丰富的需求工程师来说这些错误信息是较难发现的,对于精通此领域的专家来说,他们可以发现绝大部分的错误信息,但需要付出较大的时间代价和较多的人力资源,也不能确保能找出所有的量纲错误.

9 结 论

针对轨交领域控制软件需求的安全性,本文展开了一种基于形式化工程方法的需求建模与量纲分析研究.通过和卡斯柯信号有限公司合作的过程中,我们发现工业界往往由于需求中的量纲错误从而造成巨大的损失.所以,构建一种能有效地确认需求中量纲正确性的分析方法,对于工业界来说有着重要意义.然而,目前量纲分析的工作主要面向代码层面,使用编程语言来撰写需求在实际的工业应用中很难得到推广.

我们的工作则是为了解决工业界中真实存在的问题,通过运用形式化工程方法,将自然语言描述的需求转换为形式化规约的需求,从而解决在自然语言需求层面进行量纲分析的难题.实验结果表明该方法具有良好的可用性,量纲的分析可以使需求文档的数据字典有更加准确的定义,并在实际的轨道交通开发项目中显著提高了需求文档的准确性,且较大程度地增加了需求工程师对需求文档的信心,证明了量纲分析在需求审查阶段是一个从不同角度来保障需求正确性和准确性的方法.但是该量纲系统的定义并不能适用于所有的需求文档,后续会通过配置文件的方式来供需求工程师修改基本单位和基本量纲,使他们可以构建属于自己的量纲系统.

此项工作主要是对需求文档进行量纲的分析,未来我们将针对需求的正确性进行深层次的验证,进一步围绕需求模型展开.针对已经建立好的需求模型,通过静态分析、动态分析和模型检查等手段来提高需求文档的正确性和准确性.具体来说,我们将尝试和探索如何在工业界上实际使用多种形式化方法,探索何种方法更能被工业界所接受,尽可能减少在需求撰写过程中由于人为因素所造成的潜在错误,并会在此过程中开发相对应的工具,以协助需求工程师高效率、高精确地对需求进行分析与验证.

致 谢 感谢FMAC审稿人的推荐,他们认真负责的工作态度使得这篇论文有机会投入《计算机学报》!感谢《计算机学报》编辑的严谨认真,为本文的组织逻辑等部分提供了严谨的帮助!

参 考 文 献

[1] Mathur S, Malik S. Advancements in the V-model. International Journal of Computer Applications, 2010, 1(12): 30-35

[2] Shen W, Davis T, Lin D K J, et al. Dimensional analysis and its applications in statistics. Journal of Quality Technology, 2014, 46(3): 185-198

[3] Ahmad E, Dong Y W, Larson B, et al. Behavior modeling and verification of movement authority scenario of Chinese train control system using AADL. Science China Information Sciences, 2015, 58(11): 1-20

[4] Miao W, Pu G, Yao Y, et al. Automated requirements validation for ATP software via specification review and testing //Proceedings of the International Conference on Formal Methods. Tokyo, Japan, 2016: 26-40

[5] Issad M, Rauzy A, Kloul L. A contribution to safety analysis of railway CBTC systems using Scola//Proceedings of the European Safety and Reliability Conference. Zurich, Switzerland, 2015: 459-467

[6] Colaco J L, Pagano B, Pouzet M. SCADE 6: A formal language for embedded critical software development// Proceedings of the International Symposium on Theoretical Aspects of Software Engineering. Nice, France, 2017: 1-11

[7] Liu S. Formal engineering for industrial software development—An introduction to the SOFL specification language and method//Proceedings of the International Conference on Formal Engineering Methods. Seattle, USA, 2004: 7-8

[8] Smith G. The Object-Z Specification Language. Kluwer: Kluwer Academic Publisher, 2000

[9] Behm P, Benoit P, Faivre A, et al. Meteor: A successful application of B in a large project//Proceedings of the Wold Congress on Formal Methods in the Development of Computing Systems. Lecture Notes in Computer Science. Toulouse, France, 1999: 369-387

[10] Abrial J R. Modeling in Event-B: System and Software Engineering. UK: Cambridge University Press, 2013

[11] Karr M, Loveman D B, et al. Incorporation of units into programming languages. Communications of the ACM, 1978, 21(5): 385-391

[12] Rowlett R. How many? A dictionary of units of measurement. US: University of North Carolina, 2005

[13] Rogers P. Dimensional analysis in Ada. ACM SIGAda Ada Letters, 1988, 8(5): 92-100

[14] Rosu G, Feng C. Certifying measurement unit safety policy// Proceedings of the IEEE International Conference on Automated Software Engineering. Montreal, Canada, 2003: 304-309

[15] Allen E, Chase D, Luchangco V, et al. Object-oriented units of measurement//Proceedings of the ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications. 2004: 384

[16] Higashimori M, Harada M, Yuya M, et al. Dimensional analysis based design on tracing type legged robots//Proceedings of the IEEE International Conference on Robotics and Automation. Barcelona, Spain, 2005: 3744-3749

[17] Jiang L, Su Z. Osprey: A practical type system for validating dimensional unit correctness of C programs//Proceedings of the IEEE Computer Society. Shanghai, China, 2006: 262-271

[18] ISO. International Standard ISO 31/0: General Principles Concerning Quantities, Units and Symbols, Second Edition. Switzerland: ISO, 1981

[19] Luo Jing-Li, Du Jian-Ge. A dimensional detection method based on template metaprogramming. Modern Electronic Technology, 2009, 32(4): 80-85(in Chinese)
(罗京丽，杜建革. 一种基于模板元编程的量纲检测方法. 现代电子技术，2009，32(4)：80-85)

[20] Grein C, Kazakov D A, Wilson F. A survey of physical unit handing techniques in Ada//Proceedings of the Ada-Europe International Conference on Reliable Software Technologies. Toulouse, France, 2003: 258-270

[21] Wang Hong-Wei. On the application of dimension method in physics. China Science and Education Innovation Guide, 2007, (14): 71(in Chinese)
(王宏伟. 浅谈物理学中量纲法的应用. 中国科教创新导刊，2007，(14)：71)

[22] Mahoney J F, Yeralan S. Dimensional analysis//Proceedings of the 29th International Conference on Flexible Automation and Intelligent Manufacturing. Limerick, Ireland, 2019: 694-701

WANG Shang, M. S. candidate. Her research interests include formal engineering and requirements engineering.

FENG Jin-Cao, Ph. D. candidate. His research interests include formal engineering and requirements engineering.

ZHU Jia-Yi, Ph. D. candidate. His research interest is the verification of deep neural networks.

HUANG Yi-Hao, Ph. D. candidate. His research interests include formal engineering and requirements engineering.

ZHENG Han-Yue, Ph. D. candidate. Her research interests include natural language processing and requirements engineering.

XU Xiang-Rong, M. S. candidate. Her research interest is natural language processing.

MIAO Wei-Kai, Ph. D., associate professor. His research interest is formal engineering methods.

ZHANG Xiang, M. S., assistant professor. His research interest is software quality assurance.

PU Ge-Guang, Ph. D., professor. His research interests include program analysis and software validation.

Background

As the embedded control software is usually deployed in the security system, its correctness and security have been paid much attention by academia and industry for a long time. In order to ensure the quality of embedded control software, we focus on how to provide a set of efficient formal engineering methods to help industrial software developers to establish accurate requirements specifications, and models and confirm them. However, in the current field of embedded control software development, the application of formal methods still faces many practical difficulties, and a large number of deep problems need to be solved. In view of the characteristics of railway field, there is still a lack of effective engineering methods based on formal theory to guide the requirement analysts to complete the construction of formal model step by step from the original requirement, as well as to effectively ensure the completeness and accuracy when people depicting their expectations of software functions. Therefore, in this paper we address the above issue using a formal engineering model based on requirements modeling and dimensional analysis. A more accurate definition of the data dictionary of the requirements specifications can be achieved by analyzing the requirements specifications, thus significantly improving the accuracy of the requirements specifications in the actual railway development project. However, the definition of the dimensional system is not applicable to all requirements specifications. Subsequently, by providing the requirements engineers with configuration files to modify the basic units and dimensions, so that they can build their own dimensional system.

In the future, we will conduct a deep verification of the correctness of the requirements, and further improve the performance of our requirements model. For the established requirements model, the correctness and accuracy of the requirements specifications are improved by means of static analysis, dynamic analysis and model check. Specifically, we will try and explore how to actually apply a variety of formal methods in the industry, explore which methods are more acceptable to the industry, and minimize the human and subjective factors in requirements writing to reduce the potential defects and errors of requirements and improve modeling efficiency. Finally, the embedded control software formalized method and tool chain which is more systematic and accepted by industry engineers can provide guarantee for the quality of embedded control software.

This research work in this paper is supported by the National Natural Science Foundation of China under Grant No. 61872144, and the Young Scientists Fund of the National Natural Science Foundation of China under Grant No. 61402178.