

技术流程和研发活动的结构化

赵献民

沈阳飞机设计研究所, 沈阳 110035

摘要 参考热力学第二定律(熵增原理)剖析了系统工程的本质, 分析了对研发活动的不同认识及组织方式, 进而揭示出不同组织方式在运行效果方面产生差异的原理。提出了结构化定义和组织研发活动的方法, 以元文件体系和集成文件体系的形式说明系统工程在工程实践中的实施。

关键词 系统工程; 主体系统; 研发活动

系统工程要研究的对象不仅是研发活动及其组织管理, 还包括与系统工程相关的基本科学原理、实施研发活动的主体、方法论本身的完整性及相关性等; 需要用系统工程思想理解和研究系统工程、研究系统工程的应用。本文阐述对上述内容的一些思考。

1 对系统工程的认识

1.1 热力学第二定律

热力学第二定律即熵增加原理, 揭示了自然界的一个普遍规律, 可以帮助理解系统工程。熵即系统的混乱程度。非孤立系统的熵减小需要外部能量投入。假如, 家里的房间不打扫, 会越来越乱。要想让房间重新变得整洁, 就需要花费精力打扫。打扫房间所花费的人的精力, 对于需要打扫的房间来说就是外部能量注入。

如果任由一个系统自由发展, 系统的熵必然持续增大。如果要控制系统的熵, 必须以能量注入为代价。系统工程就是这样的一套方法: 它力争以最小的能量代价控制人造系统的熵, 并使其发展聚焦于既定的目标。系统工程的生命周期管理本质目的之一也是

为了控制熵, 使项目不至于陷入难以控制的混乱状态。

1.2 康威定律的魔咒

系统工程中所说的系统可以说是客体系统, 它是提供产品或服务为目的而人为构造的相互关联的元素的集合。与客体系统相对应的是主体系统, 它是构造客体系统的主体, 是人或者组织(企业)。

对康威定律^[1]可以理解为, 作为客体系统的产品或服务, 其架构是主体系统架构的反映。康威定律提醒我们, 要研究如何更有效地构造一个客体系统, 必须同时研究需要什么样的主体系统。因为能量代价是要由主体系统承担的, 而且主体系统的缺陷必然反映在客体系统上。

1.3 系统工程推进工作现状

要评估系统工程推进工作的现状, 首先要选择评价现状的尺子, 也就是要考察系统工程工作的目的和评价标准。

INCOSE 系统工程手册^[2]提出“系统工程以提供满足用户需求的高质量产品为目的”, 这显然是站在用户视角的观点。如果站在主体系统(企业)视角, 应该是“通过提升研发质量和效率、有效控制风险, 以提升企业的核心竞争力为目的”。

收稿日期: 2018-03-22; 修回日期: 2018-05-08

作者简介: 赵献民, 研究员, 研究方向为航空电子系统设计, 电子信箱: zxmlx@126.com

引用格式: 赵献民. 技术流程和研发活动的结构化[J]. 科技导报, 2018, 36(20): 67-73. doi: 10.3981/j.issn.1000-7857.2018.20.010

不考虑代价谈高质量产品是不现实的。企业生存发展是提供高质量产品的一个前提。系统工程有助于使企业(主体系统)的利益与用户的利益达成统一。

基于以上观点,提出系统工程工作效果的如下评价标准。

- 1) 是否有效提升了研发质量。广义的质量包括产品的竞争力。
- 2) 是否有效提升了研发效率。
- 3) 是否有效控制了研发风险。

采用以上标准,可以评估一个单位或系统工程的工作效果。有些单位的现状也许不尽如人意。

考虑一下这其中的原因,也许包括以下方面。

- 1) 科研团队对系统工程掌握得不够,需要更多的培训。
- 2) 工具链不完整,如需求管理工具、系统建模工具、仿真工具等。
- 3) 用户不按照系统工程的正确方法管理项目。

4) 首个应用系统工程的项目承受双重压力,难以应付。企业发展总是从简单系统研发到复杂系统研发,从单一项目到多项目并行。发展到一定阶段,终于意识到原来的方法再也应付不下去了,希望依靠系统工程解决发展中的矛盾。可是推进系统工程是有代价的,包括经济代价和员工的精力代价。在研发方面的压力没有减小的情况下,推进系统工程的代价就是额外代价,从哪个项目开始试行系统工程,这经常是企业难以决策的一个问题。企业要迈过这个门槛非常不容易,这就是所谓的双重压力。

- 5) 组织架构和 workflows 不匹配。

除了上面提到的,也许还有很多原因。笔者认为,关键原因也许是另外两个:方法论体系不完整;主体系统有缺陷。

2 什么是研发活动

系统工程是方法论的局部而非必要的整体。不能对方法论采取断章取义的“拿来主义”,应该全面了解先进的方法论体系,按需剪裁和扩展,聚焦于解决实际问题。另外,需要解决主体系统的缺陷。

为了说清楚这两个问题,需要从最基本的概念谈起。INCOSE 系统工程手册中说系统工程是“……结构化的研发流程”,那么流程是什么?也许可以说流程是

对活动的有序组织。活动又是什么,也就是说,对于系统研发来说,研发活动是什么?

研发活动的本质如图1所示,是在问题空间认识问题、描述问题和确认问题;在方案空间分析问题、传递问题和解决问题。

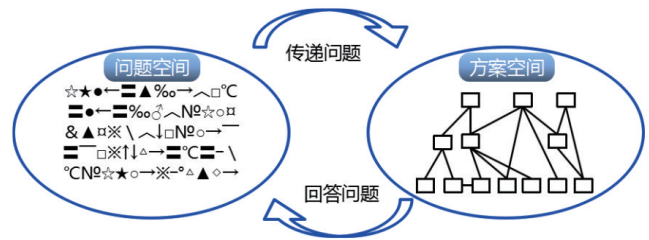


图1 问题空间和解决方案空间

事实上对什么是研发活动却有不同的观点,或者说存在另外的现实。现实之一是把部门的业务包(图2)作为研发活动。所说的部门在某些科研机构中也称为专业,两者在本质上是一样的。在项目执行过程中,工作计划总是按照部门逐层分解下达。事实上所谓的计划通常是模糊的,不能清晰完整地描述问题空间。

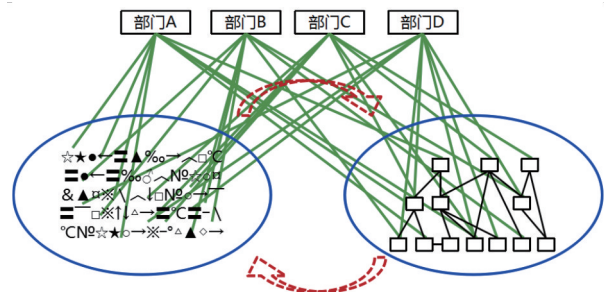


图2 将部门的业务包作为研发活动

用户的真正需要是服务,而服务主要由客体系统的功能承载和体现。将部门的业务包作为研发活动的问题在于,客体系统的功能会被部门之间的界面割裂。部门之间的业务关系复杂、业务低效,难以界定对问题空间中的问题的责任主体。

现实之二是将系统元素(SE)的描述包作为研发活动,如图3所示。图3解决方案空间椭圆中的方框表示逐级细分的系统元素。提供给用户的客体系统是一个整体,这个整体承载着用户所需要的功能。功能往往是有多个系统元素共同支撑的。将系统元素的描述包作为研发活动,功能(问题)会被系统元素之间的界面割裂。研发过程中在系统元素之间传递数据的工作量巨大,衰减和畸变严重。此处所说的传递数据的一种

形式是传递需求使用的规范。

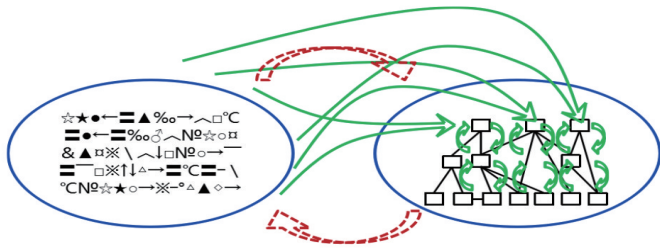


图3 将系统元素的描述包作为研发活动

分析了上面两种定义研发活动的情况后,可以考虑一个问题:有没有一种理想的情况,将问题空间和解决方案空间中的问题划分为可管理的小块,如图4所示,将每个小块作为一项科研活动,为每个小块配置人力资源。这也许是最简单高效的方式。因为每个小块都直接面向服务或者说对应具体的功能,且块之间的关系是可以依靠一套方法给出明确定义。

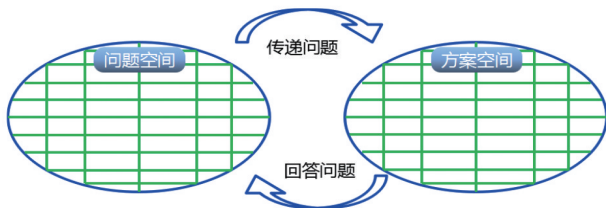


图4 将空间划分为“可管理的小块”

可是,如何将问题空间和解决方案空间划分为可管理的小块呢?在讨论这个问题之前,先讨论两个概念:面向过程和面向对象。

将科研流程视为科研活动以及科研活动的顺序,本质上是面向过程的方法。当科研活动非常复杂时,面向过程的方法是不适用的,而应该增加面向对象的方法。对于这个认识,《大象》一书中有几段非常精彩的如下陈述^[9]。

- 1) 面向过程的分析方法是找到过程的起点,然后顺藤摸瓜,分析每一个部分,直至达到过程的终点。
- 2) 将世界视为过程的这个方法本身蕴含着一个前提假设,即这个过程是稳定的……只可惜我们这个世界从来都不是一成不变的。
- 3) 并非面向过程的方法不正确,只是它应对复杂事物时面临了太多的困难。
- 4) 面向对象的方法与面向过程的方法根本的不同,是不再把世界看作是一个紧密关联的系统,而是看

成一些相互独立的对象。每个对象都只与有限的其他对象有关系,在分析对象时需要考虑的信息量就大大减少。

接下来看看认知域的4个象限(图5)。已知是有限的,仅仅是一个小角落;未知是无限的,是事物的绝大部分,事实上第三象限是无边界的。面向过程的方法基于这样一个世界观假设的前提,即认为事物之间有确定的因果关系,其关系和行为是稳定的、已知的或可知的(即可预测的)。但事实并非如此。当所面对的系统越来越复杂、越来越接近真实的世界时,就会发现面向过程的方法已经无法应付了。面向过程的方法试图将过程和关系定义清楚,并期望系统按照确定的过程和关系运行。但现实世界并非如此,过程和关系往往是不稳定和不确定。面向对象的流程比面向过程的流程更适应不确定和不稳定的关系。

不知道存在 但可知的	知道存在 且已知的
不知道存在 且不可知的	知道存在 但未知的

图5 认知域的4个象限

对于软件领域所说的过程和对象大家都是熟知的,对于科研活动来说对象是什么?结合对问题空间和解决方案空间的认识,可以说科研活动的对象有3种形态:问题、活动和文件,如图6所示。

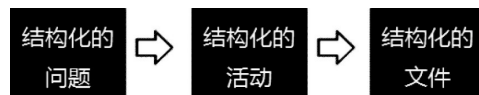


图6 研发活动的对象

系统工程定义了研发流程(图7),但没有定义研发活动及其关系,没有解决研发活动的结构化定义问题,

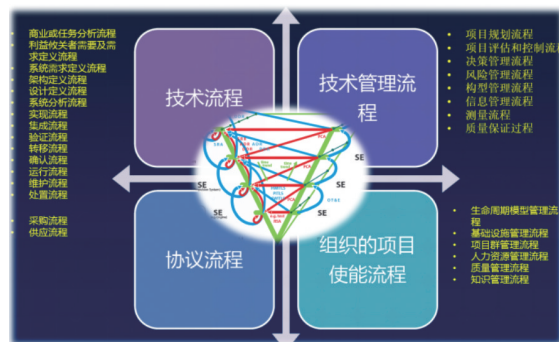


图7 结构化的流程

这也许是它落地困难的重要原因。

我们尝试通过面向对象的方法将研发活动作结构化定义,为系统工程提供必要的支持。结构化的研发活动对应结构化的设计文件体系、结构化的设计文件体系支持系统工程落地。

3 研发活动的结构化

结构化指在对事物按正交维度分类和描述基础上,用一般性方法解决问题的方法。其优势在于以下两点。

- 1) 按正交维度分类(图8)的优点是分类完整且无冗余。
- 2) 一般性方法是适用于特定类别问题的通用解决方法,而不是对特定问题的方法。与结构化方法相对应的非结构化则是针对特定事物用特殊方法解决问题的方法。

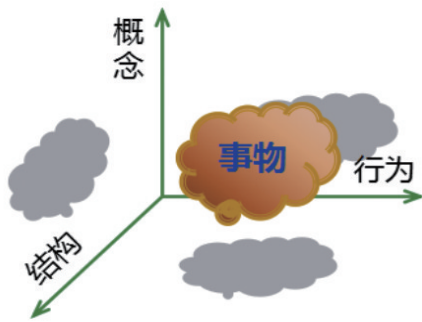


图8 对事物分类的维度

结构化的优点如下。

- 1) 因为问题类别可穷举解决方法也可穷举且易于复用。
- 2) 描述事物的数据(广义)易维护。
- 3) 描述事物的数据(广义)可追溯。

结构化的基础是对事物的分类方法。对事物的分类方法是结构化最重要的基础。

美国国防部体系结构框架(DoDAF)可以支持对研发活动的结构化定义,如图9所示。DoDAF中有几句非常重要的描述,引用如下。

- 1) DoDAF 为文档设计和开发决策提供一种结构化的方法,为系统工程提供支持^[4]。
- 2) DoDAF2.0 为系统工程过程提供广泛的体系结构支持。

3) DoDAF 模型从利益攸关者视角将问题空间分解为可管理的小块,并进一步定义为DoDAF模型。

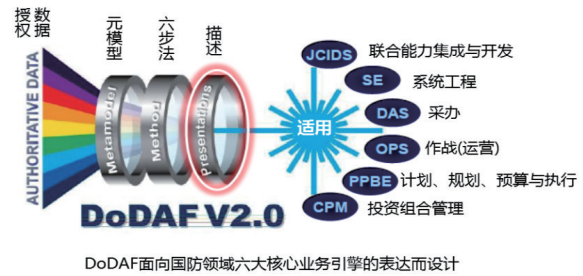


图9 美国国防部体系结构框架

概念、结构、行为、关系是对事物分类的4个基本维度(图10)。DoDAF定义的8个视角52个模型是一种细分维度(图11)^[5],可以对问题空间作进一步细分,细分为可管理的小块。让人们可以在52维空间里上看、下看、左看、右看……原来每个视图都很简单。

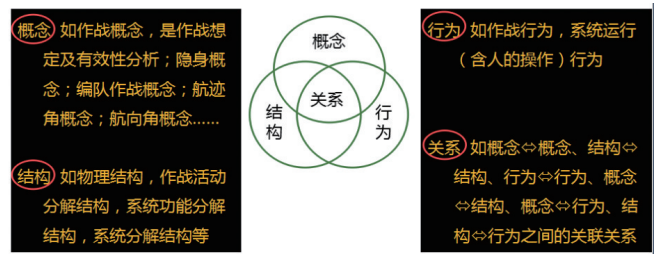


图10 对事物分类的维度

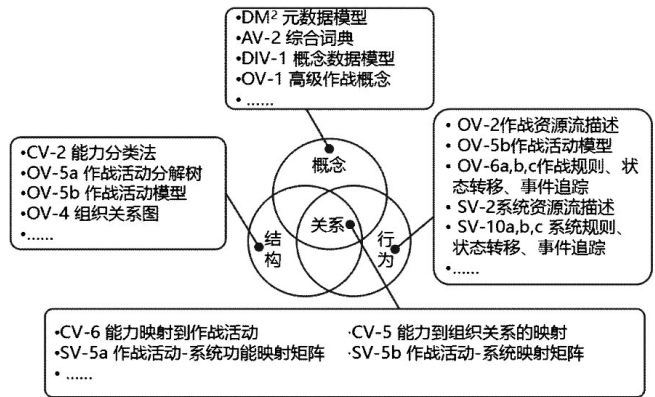


图11 4维与52维

以 DoDAF 支持系统工程,两者共同构成了更完整的方法论体系。

再回到前面的问题,DoDAF 能把问题分解为可管理的小块吗?是怎么做到的?图12就是尝试对DoDAF模型对应的问题的小块的理解。而图13是与图12对

应的DoDAF模型较正式的描述。需要说明的是,图12和图13是笔者基于自己的业务特点所作的理解和剪裁,不必深究其正确性。尤其是各模型中的关联线,笔者本人也认为不是绝对的。实际上,笔者认为DoDAF

模型之间的关系恰恰体现了并行流程、而不是完全串行流程的特点。大型项目往往需要几百、几千人多年协作才能完成,如此复杂系统研发工作,不可能是完全是串行的。

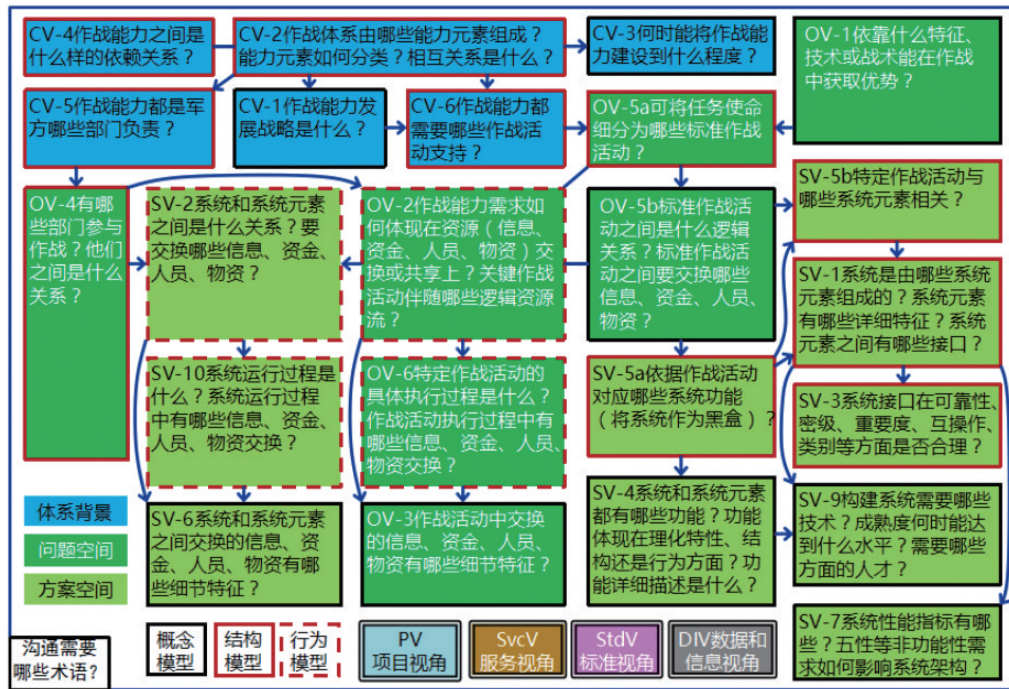


图12 问题的小块

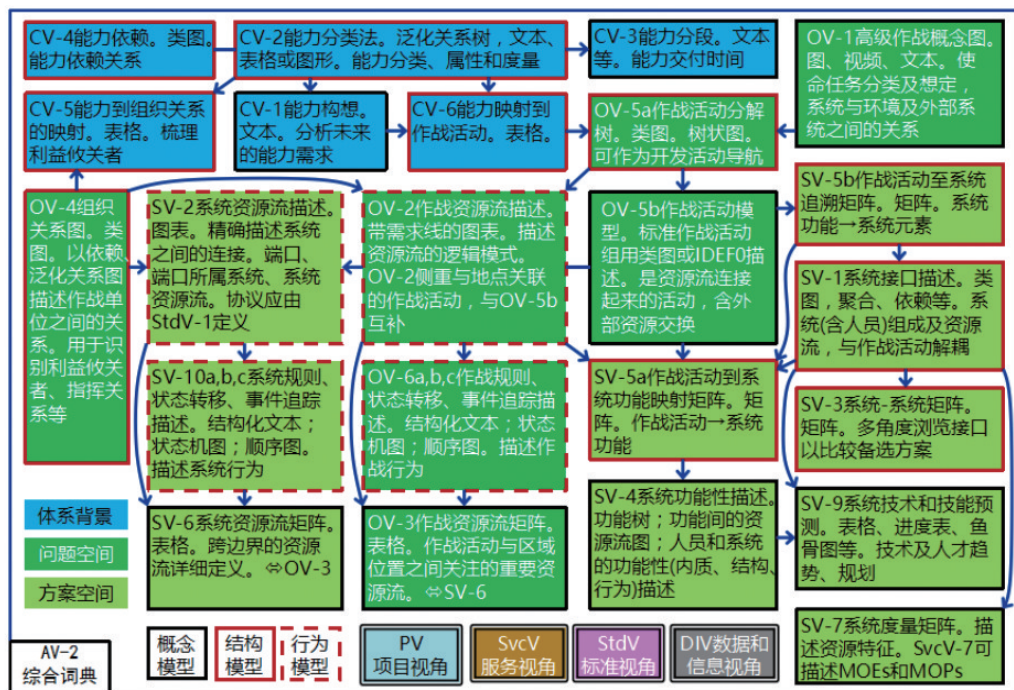


图13 DoDAF模型(剪裁)

4 元文件和集成文件

为了进一步说清楚工程实践中如何操作,笔者提出了两个术语:元文件和集成文件。

对于问题空间中的问题和解决方案空间中的问题,按照结构化方法将这些问题划分为可管理的小块,对每一小块的描述就是一份元文件。

在系统研制过程中,为了特定利益攸关者使用方便,将元文件打包并作整体描述的文件称为集成文件。例如为了开展方案评审,提供给评审专家的方案报告是集成文件;为了向下层分包商传递系统元素的需求,提供给相应分包商的规范也是集成文件。用集合的思想理解元文件和集成文件之间的关系,如图 14 所示。

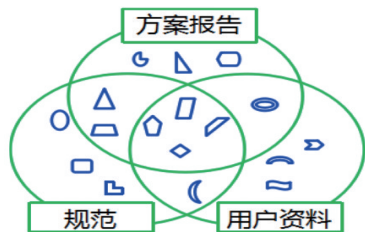


图 14 元文件与集成文件的集合

每个 DoDAF 模型可以对应一类元文件,对应问题的小块。元文件全集是对项目定义的完整数据集;集成文件通过引用元文件的形式建立元文件与集成文件之间的追溯关系。一份元文件可以被多份集成文件引用。也就是元文件一处定义,多次使用。

设计活动和元文件依靠 DoDAF 的各模型组织和描述。结构化的设计活动划分是结构化的元文件体系的基础。每份元文件均说明对应哪个 DoDAF 模型,甚至文件的编号可以带有 DoDAF 模型名称,这样就确定了该元文件在整个文件体系中的地位和作用。

以 3 棵树(图 15)作为导航地图组织大部分元文件如下。

- 1) 作战活动分解树 OV-5a。
- 2) 系统功能分解树 SV-4。
- 3) 系统元素分解树 SV-1。

设计文件是设计活动的交付物。结构化的设计文件体系等同于结构化的设计活动。构建结构化的设计文件体系,实际上也是构建结构化的设计流程。

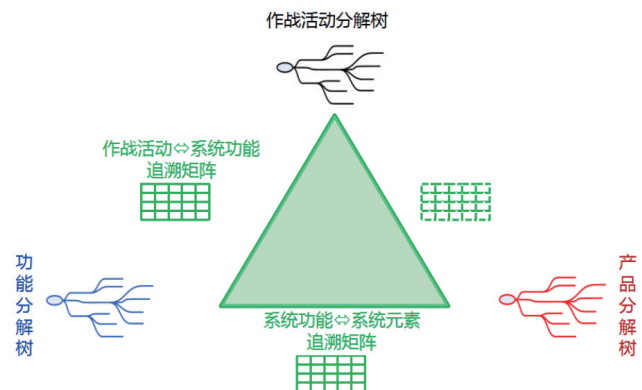


图 15 组织元文件的 3 棵树及映射矩阵

这样做的好处是,每个设计员可专注于如下 5 个活动和内容。

- 1) 特定的设计活动。
- 2) 特定的问题。
- 3) 特定类型的设计文件。
- 4) 特定的表达方式。
- 5) 特定的技术。

设计员容易上手。工作成果可以非常方便地重复利用。既利于技术进步,又利于人才成长。对于多项目并行情况下提升工作效率更是意义重大。

元文件一次定义,多次使用。结构化的设计文件是可维护且方便维护的,可以方便地组合为集成文件(如组合形成方案报告、分发给成品厂所的规范、用户资料等)。文件更改的影响范围非常清晰,易于传递变更,易于控制状态。

对应基于模型的系统工程,元文件其实就是模型。

将项目分解为面向对象的研发活动,其积极意义在于将人或团队的职责部署到问题上,而不是将界面不清的任务部署到部门和人上;不是将产品开发责任部署到部门和人上。

5 结论

主体系统的研发活动架构会在基本层面上影响系统工程的应用,决定其应用效果。对对象(问题、活动和文件)作结构化分类,是改造主体系统研发活动架构的基础。面向对象配置人力资源(本质上是配置人员的职责而不是配置人员),以此改造研发活动架构。这样做既利于项目推进,又利于技术积累,并有利于应对多项目并行。

参考文献 (References)

- [1] 刘刚. 核心系统架构转型助力金融科技创新[J]. 中国金融电脑, 2017(10): 41-44.
- [2] 国际系统工程师协会(INCOSE). 系统工程手册系统生命周期流程和活动指南[M]. 张新国, 译. 北京: 机械工业出版社, 2017: 31.
- [3] 谭云杰. 大象[M]. 北京: 中国水利水电出版社, 2009: 3-6.
- [4] The Department of Defense architecture framework(DoDAF)version2.0[S]. Washington DC: US Department of Defense, 2009: 31
- [5] 梁振兴, 沈艳丽. 体系结构设计方法的发展及应用[M]. 北京: 国防工业出版社, 2012: 172-239.

Structuralization of technical processes and research and development activities

ZHAO Xianmin

Shenyang Aircraft Design Institute, Shenyang 110035, China

Abstract This article, referring to the entropy increase theory, dissects the nature of system engineering, analyzes the different understanding of research and development activities and organization, reveals the different organization way to produce the principle of difference in terms of running effect. Based on this, a method for structuring and organizing research and development activities is put forward, which is implemented in engineering practice in the form of a meta-file system and an integrated file system.

Keywords system engineering; main system; research and development activities ●



(编辑 徐丽娇)