****

|  |
| --- |
| 二零一二年五月 |

**硕士研究生学位论文**

**题目：复杂产品协同设计工作流工具**

**的设计与实现**

The Design and Implementation of Workflow Tools for Complex products Collaborative design

|  |  |
| --- | --- |
| 姓 名： | **刘 轻 舟** |
| 专 业： | **计算机软件与理论** |
| 研究方向： | **软件工业化生产与技术** |
| 学 号： | **10498300** |
| 指导老师： | **张世琨研究员** |

个 人 声 明

在张世琨等老师的指导下，本文确为本人独立完成，并愿意承担因违反学术规范而应该承担的全部责任。

签字:

年 月 日

**摘要**

伴随产品制造业的不断发展，先进的产品制造技术日益涌现，同时也呈现出许多新的挑战，其中尤为突出的难题体现在以下几点：设计团队的扩充以致地域不断分散，设计过程愈发复杂(呈现并发与迭代的特点)，过程信息量越来越庞大(参数数量与版本众多)，设计工具的多样化。因此使得协同的重要性变得越来越突出，而本文针对这些问题，对复杂产品协同设计进行了比较深入的研究，同时设计并实现了一套协同设计工作流模型与工具。

本文把协同设计定义为在计算机的支持的系统环境中，异地分布的成员围绕一个设计项目，承担相应的部分设计任务，遵循相关的信息交换和协同机制，借助相关设计工具例如CAD,CAE等，并行交互地执行迭代式的设计，共同完成一项设计目标。

本文首先针对复杂产品协同设计及协同设计过程，对复杂产品协同设计工作流工具的需求进行了总结与分析。根据明确的需求，本文对协同设计工作流所采用的关键技术进行了研究探讨，包括常见工作流技术（业务工作流与科学工作流），任务分解的工作分解结构(Work breakdown Structure)WBS技术，协同关系描述的DSM（Design Structure Matix）技术，然后采用这些技术对协同设计工作流进行了建模，包括协同设计层级模型以及协同设计工作流模型。

接下来的设计与实现部分把工具主要分成建模工具与引擎两个模块进行设计，建模工具进行工作流可视化建模与展示，引擎负责流程的启动，运行以及监控等流程全生命周期的管理，使其可以满足协同的需求。

最后使用J2EE与Flex技术开发了整个工具。并且采用了一个飞机设计的实例来验证整个工具正确性。

**关键词：协同设计 工作流 设计结构矩阵DSM 工作分解结构WBS**

**Abstract**

Accompanied by the continuous development of product manufacturing, many advanced manufacturing technologies have been increasingly emerging. However new problems have also presented, especially the problems which are reflected in the followings:

1. The design teams have been more dispersed.
2. The design process becomes more and more complexas the characters of concurrent and iterative.
3. The increasingly large amount of information in the process (large

 number of parameters and versions).

1. The diversification of the design tools.

Thus the importance of collaboration becomes more apparent, to resolve these problems, we have in-depth research on the collaborative design of

complex products, and then design and implement a collaborative design

workflow models and tools.

The paper **defined** Collaborative Design as such :In the computer support

 system environment, each geographically distributed member   in the design project affords a corresponding part of the design task, who follows the

related information exchange and collaborative mechanisms, and uses design tools such as CAD,CAE to implement iterative design interactively together to accomplish a design goal.

The next, we make  a summary and analysis of complex product coll-

-aborative design, then we get the needs of complex product collaborative design workflow tools.

According to the definitely needs ,we have research on the mainly  technologies that can be used in the collaborative design workflow, including the common workflow technology (business workflow and scientific workflow), work breakdown structure of the task decomposition ( WBS)  technology, collaborative relationship describing technology DSM(Design Structure Matrix), and then using these technologies in the collaborative design workflow modeling.

The following part designs and implements the whole tools including a modeling tool and an engine tool. A designer can use the modeling tool for visual workflow modeling and display .The engine is responsible for the

process’ management, include starting, running and monitoring a process’s life cycle .

Finally we developed the entire tools by using the technologies of J2EE and Flex, and then used an instance of an aircraft design to verify the correctness of the entire tools.

**Key words:** **Collaborative design|** **Workflow| Design Structure Matrix**|**Work Breakdown Structure**

**目录**

[1 绪论 7](#_Toc325451631)

[1.1 协同设计背景 7](#_Toc325451632)

[1.1.1 复杂产品设计特点 7](#_Toc325451633)

[1.1.2 协同设计提出及问题 8](#_Toc325451634)

[1.2 国内外相关研究 9](#_Toc325451635)

[1.2.1 国外相关研究 9](#_Toc325451636)

[1.2.2 国内相关研究 11](#_Toc325451637)

[1.2.3 小结 12](#_Toc325451638)

[1.3 论文解决的问题 12](#_Toc325451639)

[1.4 论文组织结构 13](#_Toc325451640)

[2 工作流相关技术 14](#_Toc325451641)

[2.1 工作流技术 14](#_Toc325451642)

[2.1.1 传统业务工作流 14](#_Toc325451643)

[2.1.2 科学工作流 16](#_Toc325451644)

[2.2 工作分解结构 18](#_Toc325451645)

[2.3 设计结构矩阵 19](#_Toc325451646)

[2.4 本章小结 21](#_Toc325451647)

[3 协同设计工作流模型 22](#_Toc325451648)

[3.1 任务分解建模 22](#_Toc325451649)

[3.2 工作流程建模 24](#_Toc325451650)

[3.3 模型总结 30](#_Toc325451651)

[4 协同设计工作流工具的设计与实现 32](#_Toc325451652)

[4.1 系统需求 32](#_Toc325451653)

[4.2 系统设计 36](#_Toc325451654)

[4.2.1 总体设计 36](#_Toc325451655)

[4.2.2 算法设计 41](#_Toc325451656)

[4.2.3 数据库模型 45](#_Toc325451657)

[4.2.4 接口定义 47](#_Toc325451658)

[4.3 系统工具 47](#_Toc325451659)

[4.3.1 任务工具 48](#_Toc325451660)

[4.3.2 流程工具 49](#_Toc325451661)

[4.4 实例验证 51](#_Toc325451662)

[4.4.1 开发平台与工具 51](#_Toc325451663)

[4.4.2 运行实例 51](#_Toc325451664)

[5 总结与展望 61](#_Toc325451665)

[5.1 主要工作总结 61](#_Toc325451666)

[5.2 下一步工作 61](#_Toc325451667)

[参考文献 62](#_Toc325451668)

[作者就读期间科研成果简介 64](#_Toc325451669)

[北京大学学位论文原创性声明和使用授权说明 65](#_Toc325451670)

[致谢 66](#_Toc325451671)

**图表目录**

[Figure 1 国内航天总体设计模式 8](#_Toc325463271)

[Figure 2 复杂产品研制生命周期与特性 9](#_Toc325463272)

[Figure 3协同设计屋 15](#_Toc325463273)

[Figure 4 工作流模型 17](#_Toc325463274)

[Figure 5 各种工作流建模比较 18](#_Toc325463275)

[Figure 6 布尔DSM示例 22](#_Toc325463276)

[Figure 7 流程关系 23](#_Toc325463277)

[Figure 8 复杂流程示例 23](#_Toc325463278)

[Figure 9 3层分解模式 24](#_Toc325463279)

[Figure 10 WBS的3层模型 25](#_Toc325463280)

[Figure 11 流程示例 26](#_Toc325463281)

[Figure 12 状态转移视图 27](#_Toc325463282)

[Figure 13 DSM矩阵表示 28](#_Toc325463283)

[Figure 14 DSM矩阵表示<一> 28](#_Toc325463284)

[Figure 15 DSM矩阵表示<二> 28](#_Toc325463285)

[Figure 16 层级交互模式 29](#_Toc325463286)

[Figure 17 DSM表示的层级交互 29](#_Toc325463287)

[Figure 18 并发任务交互模式 30](#_Toc325463288)

[Figure 19 写操作 31](#_Toc325463289)

[Figure 20 读冲突 31](#_Toc325463290)

[Figure 21 读写冲突消解 31](#_Toc325463291)

[Figure 22 死锁问题 32](#_Toc325463292)

[Figure 23协同设计模型 33](#_Toc325463293)

[Figure 24 总体流程 34](#_Toc325463294)

[Figure 25 工具总体功能 35](#_Toc325463295)

[Figure 26 总体用况图 36](#_Toc325463296)

[Figure 27 逻辑架构图 37](#_Toc325463297)

[Figure 28 功能结构树 38](#_Toc325463298)

[Figure 30 各模块相互关系 39](#_Toc325463299)

[Figure 29 建模工具类图 40](#_Toc325463300)

[Figure 31 建模活动图 41](#_Toc325463301)

[Figure 32 任务定义活动图 41](#_Toc325463302)

[Figure 33 任务执行与审核 42](#_Toc325463303)

[Figure 34 变更申请 42](#_Toc325463304)

[Figure 41 WBS分解数据库表定义 43](#_Toc325463306)

[Figure 43 约束定义数据库表 44](#_Toc325463308)

[Figure 44 版本基线数据库表 44](#_Toc325463309)

[Figure 45数据库表 45](#_Toc325463310)

[Figure 36 工作流引擎状态转移 46](#_Toc325463311)

[Figure 37变更影响分析流程 47](#_Toc325463312)

[Figure 38变更影响分析算法 47](#_Toc325463313)

[Figure 39 多方案比较算法流程 48](#_Toc325463314)

[Figure 40 构造DSM树 48](#_Toc325463315)

[Figure 46 工作集成机制 50](#_Toc325463316)

[Figure 47 WBS元素 50](#_Toc325463317)

[Figure 48 WBS品字图示例 51](#_Toc325463318)

[Figure 49 WBS树形图示例 51](#_Toc325463319)

[Figure 50 DSM工具类图 52](#_Toc325463320)

[Figure 51融合参数的DSM 53](#_Toc325463321)

[Figure 52 飞机总体设计过程 54](#_Toc325463322)

[Figure 53 WBS树形图 55](#_Toc325463323)

[Figure 54 WBS品字图 55](#_Toc325463324)

[Figure 55 任务定义 56](#_Toc325463325)

[Figure 56 参数定义与绑定 56](#_Toc325463326)

[Figure 57 指定执行人 56](#_Toc325463327)

[Figure 58 定义约束 57](#_Toc325463328)

[Figure 59 约束定义列表 57](#_Toc325463329)

[Figure 60 DSM视图 58](#_Toc325463330)

[Figure 61 DSM视图说明 58](#_Toc325463331)

[Figure 62 流程视图 59](#_Toc325463332)

[Figure 63 DSM模型的XML表示 59](#_Toc325463333)

[Figure 64 执行任务列表 60](#_Toc325463334)

[Figure 65 调用计算工具计算 60](#_Toc325463335)

[Figure 66 变更影响分析 61](#_Toc325463336)

[Figure 67 任务执行状态监控 61](#_Toc325463337)

[Figure 68 流程总体监控 62](#_Toc325463338)

[Figure 69 多方案比较 62](#_Toc325463339)

[Figure 70 毯式图法 63](#_Toc325463340)

[Figure 71 确定总体参数 63](#_Toc325463341)

# 绪论

## 协同设计背景

### 复杂产品设计特点

一般来讲，在复杂产品研发的全生命周期中，需要历经需求定义、技术方案定义、部件分系统制造、系统集成、系统试验评估、系统交付等几个阶段**[1]**。

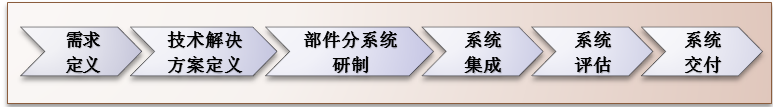


Figure 1 国内航天总体设计模式

这个是国内大部分航空航天项目的产品研制生命周期，一个型号项目首先由专项办决策层制定项目总体规划，然后将规划分解为高层计划分配到各个项目办公室，每个项目办公室作为控制层负责一个子项目的生命周期管理。然后有承研单位作为执行层对每个子项目进行执行设计。

在全生命周期中，各个阶段呈现的问题如下图所示：

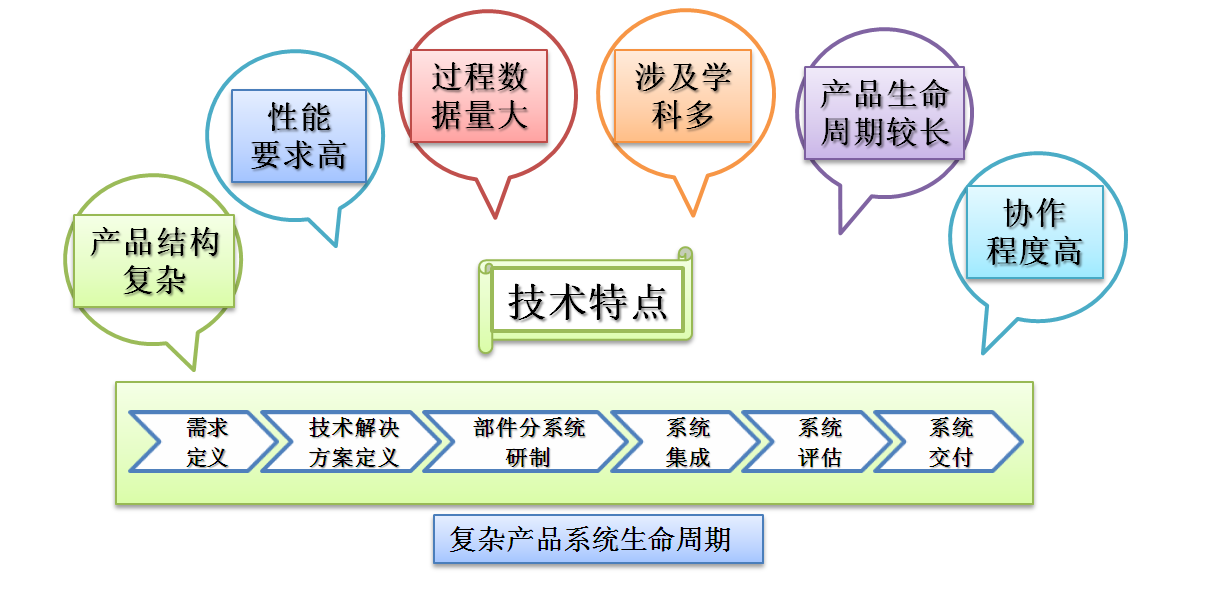


Figure 2 复杂产品研制生命周期与特性

以航空为例，航空复杂产品比如波音飞机以及空客，具有几何结构复杂、性能要求高、过程数据量大版本多、涉及学科多、制造工艺复杂、研制周期长等特点。同时过程中涉及了气动、结构、强度、传热等多个学科专业以及领域，因此需要相关的团队紧密配合，才能让复杂产品的研制过程更加高质高效低成本。复杂产品的多地域和多学科研制特点反映了产品开发的协作要求高，分布合作的多单位构成了协同工作的**异地**特点；复杂产品开发各部分工作具有的关联性和各自工作局部开展具有时序上的**异步**并发性。这些异地、异步的特点对产品研制过程的协同和信息共享都提出了更高的要求，迫切需要从方法、环境、协同模式等方面研究产品协同设计中的关键问题，以缩短产品的研制周期**[2]**。

### 协同设计提出及问题

协同设计的理念最开始可以追溯到1984年。Computer Supported Cooperative Work （CSCW）即计算机支持的协同工作：在计算机技术支持的环境下，一个群体协同完成一项共同的任务[3]。CSCW的涵盖面过于广泛，涉及一切人机相关的群体协同工作领域。而协同设计则主要面向的是产品设计领域。

结合上述过程中呈现的问题，以及相关的研究，我们将协同设计定义为**在计算机的支持的系统环境中，异地分布的成员围绕一个设计项目，承担相应的部分设计任务，遵循相关的信息交换和协同机制，借助相关设计工具例如CAD,CAE等，并行交互地执行迭代式的设计，共同完成一项设计目标。**

伴随产品制造业的不断发展，先进的产品制造技术的日益涌现，协同的重要性越来越被广为认知。不仅体现在需要打破组织内部严格的团队界限，强调多团队的有效协同，而且要求在产品设计过程中不同地点或不同企业的合作伙伴实现资源、信息、过程以及应用的有效协同，甚至是共享知识和决策的协同。协同设计过程既要完成信息的共享，也要完成过程的驱动和控制，对多样化的设计工具集成也有要求。归纳一下协同设计的问题与特性，主要有以下几个方面**[4]**：

**异地性**：复杂产品协同设计团队一般都是由异地分布的设计小组构成，他们需要通过网络来进行信息的交互与共享，协同完成设计任务。

**并发性**：为了协同设计的效率，设计任务要求比较高的并发，尤其是相互独立的设计任务，具有良好并发性的协同设计平台，是高效率产品设计的保证。

**冲突性**：设计任务之间由于存在输入输出的约束，所以在设计过程当中由于信息交互的延迟以及不对称，协同过程的冲突在所难免，因此协同设计需要解决冲突的问题。

**集成性**：各单位采购了很多建模、设计、分析、仿真、优化工具，大多数在独立使用，需要进行衔接，达到最优效果。

**参数化**：在实际的设计过程当中，设计师之间的信息交互，尤其是产品的特征描述，主要通过参数化描述的信息单元进行传递，例如产品的几何特征、功能特征等信息，同时和常见的CAD设计软件相互兼容。参数化设计过程是构思设计要求、提出产品功能、给出相应的结构和几何约束，最后形成设计对象，不断修改，直到满足设计要求的过程。

本文提及的协同设计也正是从以上几点问题出发，进行了进一步深入的研究。

## 国内外相关研究

### 国外相关研究

**计算机支持的协同工作**

1984年由美国[麻省理工学院](http://www.baidu.com/baidu.php?url=RnfK00aCgVNuQSVKN-qBKpAQ05OwQqy79hOLzYXnemFDTNkijPpnWOg5ASlfH3-1Msn1arUEEBxAfaJGNZsZcOmsGapRK_RRcs7wsV8i-jQX9ND_L3QCgdIog35J.7D_imcMOfhpPqjw4g_ECECrkC6sw5S9nYUlABgECsnkIjvgnU8VqdZC4U8W-EzoDkAeRe0q5QgzLp-9h9m3e8M__R0.U1Yk0ZDqHR-R0Zfqs8M2CTjV8pHwGqil0A-V5HcsP0KL5fKM5gN8nHD0Iybqmh7GuZN_UfKspyfqnfKWpyfqn0KVIjYk0AVG5H00TMfqrjD10AN3TjY0uy-b5HDkrHf0Uy-b5H00mhbqn0KVm1YdPWRLn1n1n6KkTA-b5H00TyPGujYs0ZFMIA7M5H00mLFW5HRzPWb1)（MIT）的Irene Greif和Digital Equipment Corporation (DEC)公司的PauCashman两位研究人员领导了一个20余人的工作组，他们首次提出来了Computer Supported Cooperative Work CSCW (“计算机支持的协同工作”)这一概念。[5]

CSCW的定义如下：“地域分散的一个群体在计算机与网络支持的环境中协同完成某项共同的任务。”[6]由于计算机技术以及网络技术的兴起，CSCW应运而生，CSCW技术发展至今，产生了各类应用系统，主要包括：远程教育、电子商务、电子政务中的决策与协调、远程医疗应用等。可见计算机支持的协同工作应用范围已很广泛。

由于CSCW的出现，处于不同空间的人们可以在计算机以及网络提供的环境中，进行更加高效以及共享的工作。可以说CSCW大大减少了人们空间上的隔阂以及信息交流的阻碍。

**并行工程**

另一方面，在时间维度上，制造业传统的设计模式以串行设计为主。传统的串行迭代的模式，最典型的瀑布式的设计方法，是按[产品生命周期](http://wiki.mbalib.com/wiki/%E4%BA%A7%E5%93%81%E5%AF%BF%E5%91%BD%E5%91%A8%E6%9C%9F)的各个过程顺序执行的，一旦在使用阶段发现问题，必须重新在前面各阶段中找原因加以解决，带来很多的重复劳动，效率往往比较低下。人们为了提高效率，又在不断的尝试和创新中找到了新的模式。

1988年美国防御分析研究所（Institute of Defense Analyze, IDA）完整地提出了并行工程的概念: “集成的、并行地设计产品及相关过程的系统方法。”[7]目前并行工程在美国、德国、日本等一些国家中已得到广泛应用，其领域包括汽车、飞机、计算机、机械、电子等行业。

并行工程主要研究的内容是[8]：

1.建立能够描述产品生命周期包括需求定义、设计、仿真、制造等所有阶段的统一模型。

2.对具备与产品设计并行交互的能力，对产品或零件进行可制造性评价并反馈结果的CAPP进行研究。

3. 研究面向产品生命周期各/某环节的设计(Design for X)。

4.实现对企业内的各类应用进行信息、功能以及过程集成的并行工程集成框架。

5.冲突消解：消解协同过程中产生的资源访问、信息共享等冲突。

从并行工程的关注点以及解决的问题来看，已经非常适用于复杂产品的协同设计过程当中了，但是针对国内的协同设计尤其是军工领域的产品协同设计又有着新的问题与难点需要研究及解决。

**国外协同设计相关工具**

国外在协同设计方面的研究历史比较长，其中比较著名有Kepler，Isight等科学工作流工具以及Pro/Engineer和Solid Edge的商业设计软件。

**Kepler[9]**

Kepler是一个由一系列面向应用的研究项目开发而成的开源科学工作流引擎。美国加利福尼亚大学伯克利分校的相关研究单位，在Ptolemy II系统的基础上建立了Kepler系统。它是一个以数据流驱动的工作流架构，并以参与者为导向实现科学计算。参与者可以从输入流中获取需要的参数再将参数信息写入输出流中。于是一组参与者就可以通过引入一个从输出到输入的映射有线地联结在一起。

**iSight[10]**

iSight是美国Engineous Software 公司开发的多学科设计优化框架。iSight可以将设计师在设计中所需要的各种工具组装成一个设计流程，然后将方案设计与评价过程完全自动化。iSight自动控制各个设计工具的执行和交互，在运行过程中将数据从一个工具传到另一个工具。最后，iSight充分利用过程自动化，运用相关搜索算法确定最优的设计参数。

上述两个系统主要针对科学计算实现的自动化科学工作流系统，均是以参数流驱动，支持不同地区的团队进行协同。

**Pro/Engineer[11]**

**Pro/Engineer**是美国参数技术公司（PTC)旗下的CAD/CAM/CAE一体化的三维软件，它第一个提出了[参数化设计](http://baike.baidu.com/view/725815.htm)的概念，采用了单一数据库来解决特征的相关性问题。Pro/Engineer的基于特征方式，能够将设计至生产全过程集成到一起，实现并行工程设计。

**Solid Edge[12]**

Solid Edge是有UGS(后来被西门子收购)公司开发的一块参数化特征实体造型软件。它的特点是提供了用户体验极佳的可视化设计方式，支持并行装配存取控制，能够让多个工程师可以工作于同一个设计项目，从而进行协同设计。

这两个软件主要侧重于CAD软件的参数化设计与协同的集成，解决工具与协作平台的集成问题，主要偏向设计，协同支持程度不如[9][10]中提及的两个系统。

### 国内相关研究

国内对于计算机支持的协同工作的研究始于90年代中期，研究工作主要集中在理论方面，对于具体的协同设计的应用研究较少，至今还没有形成一个完整的商品化的协同设计制造软件。

在理论研究方面国内相关研究单位关于协同设计的主要研究成果有**[13]**：

国内于1992 年开始了并行工程的预研阶段，由863/CIMS 年度计划和[国家自然科学基金](http://baike.baidu.com/view/1314657.htm)资助了一些相关技术的研究课题；在1993 年，863/CIMS 组织[清华大学](http://baike.baidu.com/view/1563.htm)、北航、上海交大在内的等204 所高校和单位组成了一个并行工程可行性论证小组，在 计算机集成制造系统(Computer Integrated Manufacturing Systems , CIMS )工程的基础上开展并行工程的攻关研究。

1998年开始，研究成果在航天领域得到了运用，比如[西安](http://baike.baidu.com/view/2155.htm)飞机工业集团有限公司在已有的各种CAD/CAE/CAM软件的基础上，开发一套支持飞机内装饰并行设计的系统工具。最后取得了显著效益：研制成本降低20%+，研制周期缩短33%，尤其是设计过程变更，减少了 60%+，。

尽管国内的相关的研究也做了很多，但是能够很好的解决复杂产品协同问题的协同设计工作尚处于初步阶段，国内软件支撑也不够完善，基于参数驱动工作流的协同设计模式依然需要进行大量以及深入的研究。

### 小结

总之，国内外对于计算机支持的协同设计系统的应用和研究还是存在如下问题：

**标准化程度低。**不同厂商生产的协同设计平台都有自己独立的一套流程体系，工作流语言，API函数。平台集成难度很高。各个平台的设计过程又涉及了不同的设计工具（CAD、CAE工具等），这些工具与平台的集成也很欠缺。

**信息共享不够。**基于CSCW技术，特别是实时多媒体通信和各种辅助支持工具（例如CHAT、电子白板、应用程序共享等）的协同设计系统较多的考虑了支持实时协同设计的能力，但是缺少对于信息共享，特别是动态的工程设计信息的充分利用。

**迭代设计过程支持不好。**协同设计过程最大的特点就是迭代，在设计某个新产品的时候，某个参数需要经过多轮的计算迭代才能获得比较理想的结果，当前协同设计系统对这个过程的支持还比较薄弱。

## 论文解决的问题

在研究过程当中，我们针对协同设计的特点，总结出了我们当前关注的协同设计的问题：

1. 设计工具集成

大量CAD、CAE以及PDM 工具形成“信息孤岛”，需要进行有效的集成；

1. 团队协同设计

满足人员管理上的地域、层级以及保密等限制，为了节省沟通成本，提高设计效率；

1. 迭代设计支持

为了进行快速的进行设计迭代，需要对多方案设计和比较的良好支持；

1. 过程数据管理

过程中产生的参数数量以及版本过多，需要对过程数据的有效管理。

我们针对其特性，提出了协同设计平台的设想来解决上述的典型问题，可以用一个图来描述如下：

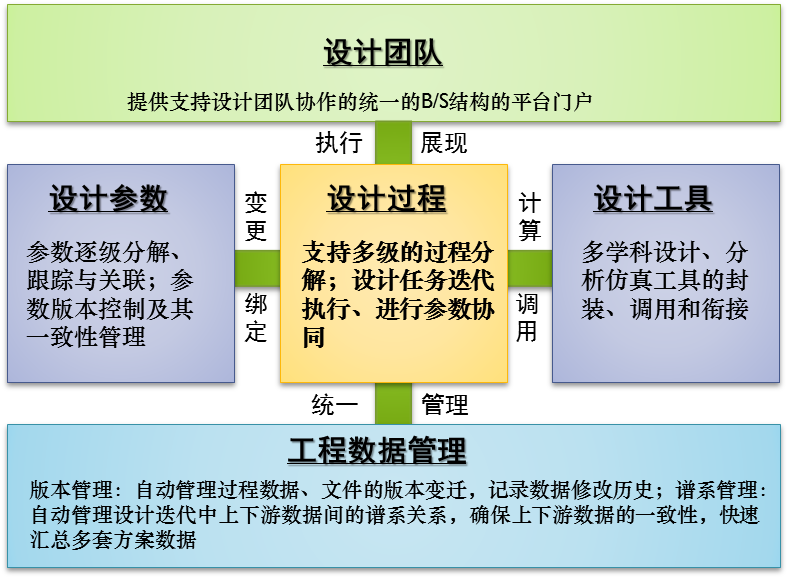


Figure 3协同设计屋

其中**设计过程**为核心，负责整合设计团队、设计任务（包括设计参数与设计工具），同时进行设计流程的管理：主要有设计方案的配置，设计任务协同与迭代执行，工程数据的读写，设计工具的调用等工作。

## 论文组织结构

本文共有五章来构成，论文的具体篇章组织如下：

第一章绪论

阐述了课题的研究背景及相关概念、国内外研究现状、课题的研究内容与解决问题及论文组织结构。

第二章协同设计设计工作流相关技术

介绍了当前的主流的工作流技术包括传统业务工作流和科学工作流，并介绍了接下来建模过程当中所用到的工作分解结构以及设计结构矩阵。

第三章协同设计工作流模型

本章主要是建立了基于工作分解结构WBS的协同设计层级模型以及基于DSM的协同设计工作流模型。

第四章协同设计工作流工具的设计与实现

进行协同设计工作流工具的需求描述，包括用例图，模块图，功能图，流程图等；根据需求进行工具进行总体设计与详细设计包括三个最关键的实现算法。最后使用一个飞机设计实例进行工作流工具的运行测试，对结果进行分析。

第五章结论与展望

对论文及本人所做工作进行总结，并对有待进一步完善的研究内容进行了展望。

# 工作流相关技术

## 工作流技术

按照国际工作流管理联盟（Workflow Management Coalition， WFMC）的定义[16]，工作流是指“在计算机支持下，实现业务过程的自动化”。其中文档或者过程信息遵循一组已定义的规则集合，以任务的形式，从一个参与者传递到另一个参与者执行。发展至今，常见工作流类型主要是传统业务工作流与科学计算工作流。

### 传统业务工作流

业务工作流始于80 年代办公自动化领域即所谓的“无纸化办公”[17]。在1993 年，国际工作流管理联盟WFMC的成立标志着工作流技术开始进入相对成熟的阶段。

由WFMC 在业务工作流BFW的体系架构、标准术语及API等方面制定了一系列相关标准。WFMC发布的工作流系统抽象模型主要如下图所示[18]：

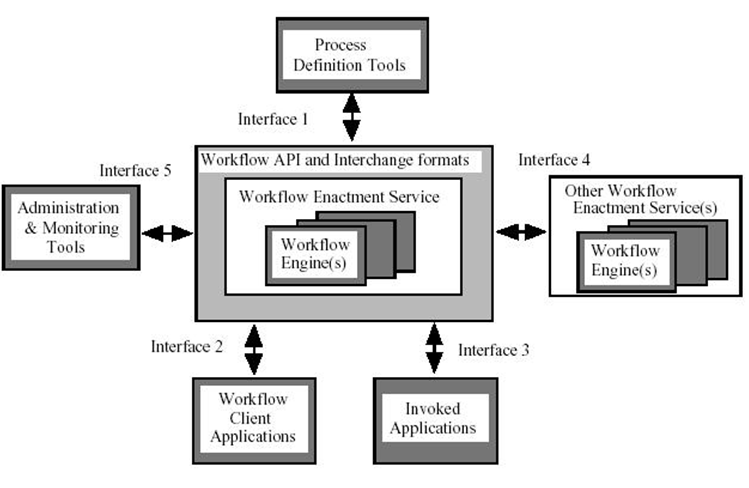
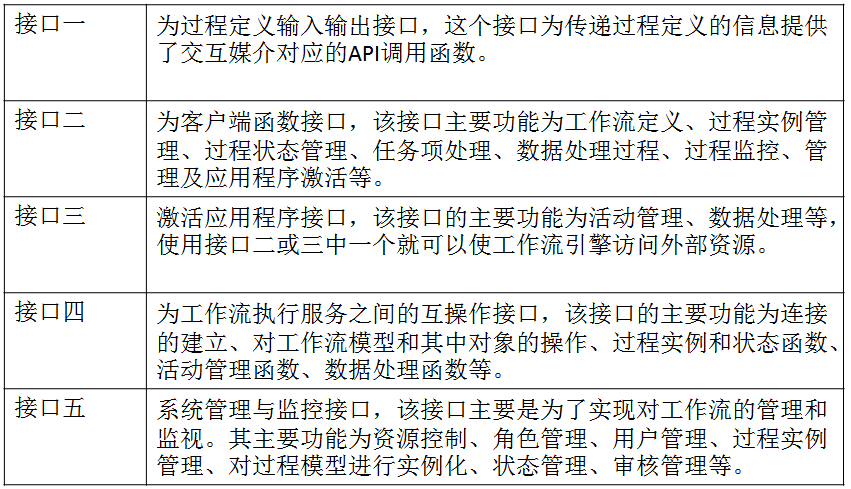


Figure 4 工作流模型



常见工作流控制模式：

|  |  |
| --- | --- |
| **基本控制模式** | |
| 顺序控制 | 任务顺序执行 |
| 并行分叉 | 任务并行执行 |
| 同步 | 同步两个并行任务 |
| 排他选择 | 多路选择一个执行 |
| 简单合并 | 合并多个可选路径 |

对于工作流建模理论基础的研究，目前还没有统一的建模方法标准。从现有的研究工作来看，主要有流程图、Pert、IDEFX、Petri网、DSM等建模方法[18]。

下表是常见的工作流建模工具比较：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 流程图 | PERT | IDEFX | Petri网 | DSM |
| 过程描述能力 | 好 | 一般 | 较好 | 好 | 好 |
| 功能描述能力 | 弱 | 弱 | 好 | 弱 | 弱 |
| 组织描述能力 | 较好 | 较好 | 弱 | 较好 | 好 |
| 资源描述能力 | 较好 | 较好 | 弱 | 较好 | 好 |
| 信息描述能力 | 较好 | 较好 | 好 | 较好 | 若 |
| 产品结构集成能力 | 弱 | 弱 | 弱 | 弱 | 好 |
| 迭代分析能力 | 弱 | 弱 | 弱 | 一般 | 好 |
| 过程分析能力 | 一般 | 一般 | 一般 | 一般 | 好 |
| 计算机化能力 | 好 | 好 | 好 | 好 | 好 |

Figure 5 各种工作流建模比较

根据协同设计的特性，再结合产品结构设计的特殊性，包括结构表示，设计过程迭代等等，我们基于每个描述方式的各自特点发现DSM是其中最适合产品协同设计过程建模的工具，因此我们后面会更多提及到对迭代过程支持最好的DSM模型。

### 科学工作流

科学工作流（Science Work Flow SWF） 起源于传统的业务工作流技术 (Business Workflow, 简称 BWF)，随着网格计算的发展，已经为大规模的科学计算提供的新的想法与更广阔的空间，但是与此同时，科研过程中的数据分析过程也日益复杂，同时计算过程又涉及了更多地组织以及不同领域的科学家进行协同作。为了提高科研效率，减少科研成本，同时应对越来越来大规模科学计算的挑战，迫切需要一种类似业务流程自动化的流程系统来对复杂的科学计算流程进行管理与控制。

科学工作流技术就在这种背景下应运而生，它作为一种促进和保障协同开发的技术路线为科学家提供一种科学计算流程定义和自动运行的支撑平台。[19]

对于科学工作流的定义有很多，比如：

在 GridFlow 系统中Guan 等人将 SWF 定义为[20]：“根据对应的规则集合在选定的数据上进行科学计算”。这个定义比较简洁，但体现了科学工作流的用途与基础。

在Liu 等人在提出的科学工作流管理系统参考模型 VIEW(VIsual sciEntific Workflow management system)中将 SWF 定义为[21]：

“科学工作流SWF通过自动或半自动化方法来帮助进行科学实验的过程，通过数据流将工作流任务组合在一起来完成数据处理和分析，并达到加速科学发现的目标”。此定义较为详细的将科学工作流的特征进行了总结，数据流驱动，自动化运行等。

目前国际上常见的SWFMS[22]有：Triana、Pesasus、Kepler、ASKAIDN、Java Cog Kit、P-GRADE和GridFlow等，这些系统也就是基于业务工作流(Business Workflow，BWF)的基础上，再针对各自的领域特征开发出来的工作流系统。

在SWF中需要对工作流中的任务以及任务间的逻辑关系进行表示，其描述方法大体可以分为基于有向图、基于Petri网以及基于UML 三大类。

科学工作流技术与业务工作流技术相比，最大的不同在于：业务工作流是控制流驱动，每一步的自动执行与活动的状态等条件相关联，活动的激活依赖于前置活动的执行结果，业务工作流的主要作用是使得业务流程例如评审、审核更加高效便捷；科学工作流是数据流驱动，决定每一步的执行的主导是输入数据的约束条件是否得到满足，一旦满足则活动自动执行，且支持任务迭代，从而使得科学计算更加高效准确。所以SWF起源于业务工作流，又与BWF不同。SWF面向数据，并具有数据集成、计算集成、分析集成的特点。与BWF相比，SWF系统具有如下一些特点。

|  |  |  |
| --- | --- | --- |
|  | 传统业务工作流BWF | 科学工作流SWF |
| 驱动 | BWF面向控制 | SWF面向数据 |
| 应用领域 | BWF复杂程度与规模 | 跨领域、多学科和大规模数据量的特征 |
| 执行过程 | 较少对执行过程中的动态变化因素提供支持 | 需要支持动态流程设计与修改 |
| 数据管理 | 没有特别的数据要求 | 需提供数据与工作流流程的来源管理 |



## 工作分解结构

工作分解结构work breakdown structure (WBS), 主要是指把一个复杂的工作通过使用一定的原则，将其分解细化为更细粒度的工作，可以极大的把一项工作具体到一个可控范围，也能明确定义某个复杂的工作。

*A work breakdown structure (WBS), in*[*project management*](http://en.wikipedia.org/wiki/Project_management)*and*[*systems engineering*](http://en.wikipedia.org/wiki/Systems_engineering)*, is a deliverable oriented decomposition of a project into smaller components. It defines and groups a*[*project*](http://en.wikipedia.org/wiki/Project)*'s discrete work elements in a way that helps organize and define the total work scope of the project.[23]*

在WBS设计原则中，公认的最重要的原则被称为100%原则.[[24]](http://en.wikipedia.org/wiki/Work_breakdown_structure#cite_note-12)

原文定义如下：The 100% rule states that the WBS includes 100% of the work defined by the project scope and captures all [deliverables](http://en.wikipedia.org/wiki/Deliverable) – internal, external, interim – in terms of the work to be completed, including project management.

100%原则是说WBS是包含项目范围内定义的100%的工作的，并获取所有的交付物，包括内部的，外部的以及项目过程中的各个阶段包含项目管理在内。

这条原则详细阐述说明如下：

(1)WBS覆盖了项目工作的全部范围。

(2)所有的可交付成果或输出产品都在WBS中得到表示。

(3)每一级元素的总和都代表了次高级元素工作的百分之百。

(4)每一个元素中的工作都等于其下属元素工作的总和

(5)细分应该是有逻辑性的，并要反映产品、服务或结果的本质特征。

(6)项目利益相关者应该参与WBS开发。

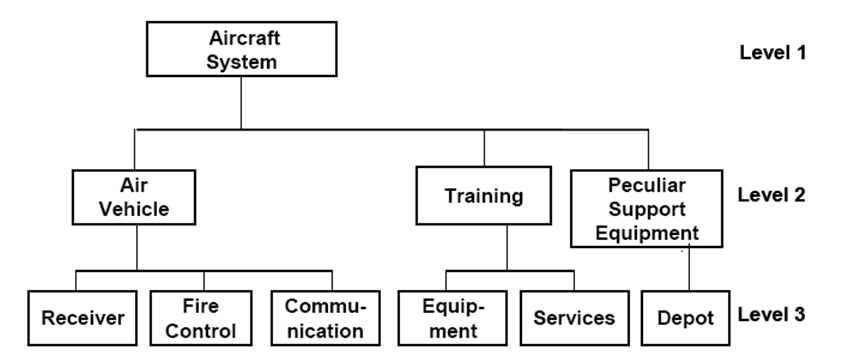
(7)在项目利益相关者参与WBS建立后，WBS应该成为基准

(8)WBS应集中于项目输出或可交付成果，它不是一个组织图，也不是一个进度计划或资源表。

(9).最低级WBS元素应该是在活动以上的那一级

(10).最低级WBS元素应该允许对项目管理的足够的控制和可见度。

航空领域的WBS示例[25]



一个简单的三层航空系统的WBS模型，包括航空器(飞机)、训练、特定的支持设备；各个部分再进行3层的细化分解。

在文章里，我们用到了WBS作为项目分解工具对协同设计项目进行了层次划分以及设计任务的分解细化。

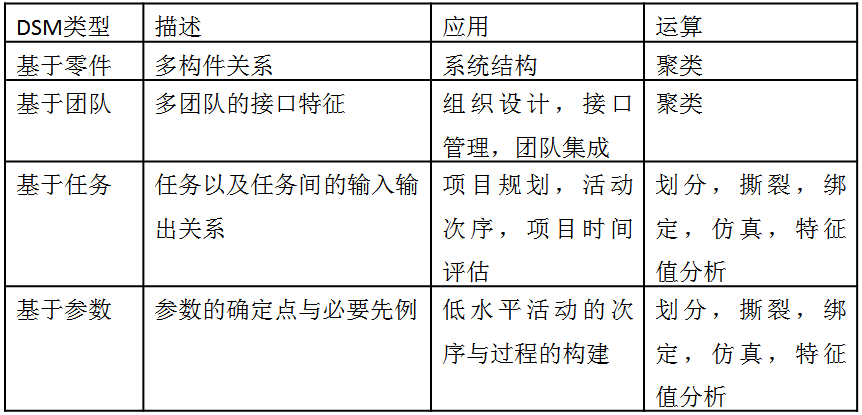
## 设计结构矩阵

设计结构矩阵（Design Structure matrix）简称DSM，最早是由美国的Steward博士提出的，用于对产品开发过程进行规划和分析的矩阵工具。[26]

常见DSM一般根据领域划分为四大类，包括基于零件DSM、基于团队DSM、基于任务DSM以及基于参数DSM [27]，它们分别以零件、团队、任务以及参数作为模型的行列元素。

1. 基于零件DSM描述的是零部件之间的关联关系，体现的是产品结构的特性。
2. 基于团队DSM描述团队间的信息流动，基于信息流建立对应的组织模型。
3. 基于任务DSM通过任务之间的输入输出关系来对过程进行建模与分析。
4. 基于参数DSM根据系统参数之间的依赖关系对系统进行建模与分析。

各个DSM 模型分类与特征如下表所示。



根据协同设计的特征，以产品设计任务为主，描述任务间输入输出的关系，我们主要采用的是基于任务的DSM建模。

DSM是由有向图发展而来的，可以用矩阵的形式表示。行列元素的排列顺序相同，下图列出了A,B,C,D 4个任务，我们用最简单的布尔型DSM来表示他们的关联关系。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** |  | **False** | **False** | **True** |
| **B** | **False** |  | **True** | **False** |
| **C** | **False** | **True** |  | **False** |
| **D** | **False** | **False** | **False** |  |

Figure 6 布尔DSM示例

对角线上方出现True则说明两个活动间有正向连接：例如B->C，如果为False则没有正向联系;相应的如下对角线下方出现True，则说明二者间有反向联系，在协同设计中用来表示反馈关系。如果两个活动存在正向与反向联系，那么这两个活动就存在协同设计中一种重要并且复杂的关系：耦合。如上图中B->C与B<-C构成的耦合关系。

对于协同设计任务，任务之间的最基本的关系有如下3种：顺序关系、并行关系、耦合关系。

如下图所示：



Figure 7 流程关系

由这3个关系将各个任务关联起来形成了复杂了协同工作流程。



Figure 8 复杂流程示例

## 本章小结

这一章主要是介绍了协同设计工作流相关的一些关键技术。包括业务工作流技术BWF与科学工作流技术SWF、工作分解结构技术WBS与设计结构矩阵技术DSM。

传统业务工作流面向业务流程，例如办公自动化中的审批流程，表单或者文档审核的自动流转，是以控制流为驱动的过程自动化技术。它解决了流程自动化执行与控制的问题，并有一套比较成熟的模型体系，而BWF针对业务流程，以控制流驱动，无法满足科学计算中迭代计算的核心问题，因此相关研究人员提出了科学工作流SWF，专门用来解决科学计算的问题。SWF又结合了网格计算的特性，是基于数据驱动的自动化流程。协同设计工作流和SWF的一致之处也在于此，是基于参数驱动的流程，需要支持迭代设计。因此从这个关键要素来看，科学工作流更适用于协同设计过程。

工作分解结构WBS技术与设计结构矩阵DSM技术由于各自自身的特点，分别在协同设计的任务分解，层级模型建立和任务依赖关联关系定义，流程模型建立上起到了核心作用。将在下一章作详细介绍。

# 协同设计工作流模型

一个复杂产品的设计的基本单元是设计任务，而设计任务来自项目WBS。系统分解设计的主要工具是工作分解结构（WBS，Work Breakdown Structures）。

## 任务分解建模

为了清楚的描述复杂产品设计上下级以及同级之间的交互过程，我们在此处使用了WBS技术，并进行了一定的扩展。

在复杂产品设计过程尤其在总体设计过程当中，一般都是将项目进行3层分解如下图：



Figure 9 3层分解模式

决策层的工作主要是进行产品需求目标确定与建立，制定顶层的**WBS顶层分解结构**。管理层主要是承担各个产品部件或者子系统设计，他们负责组织团队、细化WBS、实施产品设计、监控任务执行、汇总执行结果。执行层属于各个具体设计任务的执行，具体到每个设计团队甚至设计人员。

决策层的确定一个产品的设计方案后，将工程分解为各个子结构给承担的下级部门或者单位，下级单位为他们承担的部分进行进一步的细化，设计过程中可能出现设计的多方案问题，因此Parsec提出了Concept的概念来表示一种具体方案。

而在执行层，我们借鉴科学工作流的思想，采用数据驱动的模式，进行协同设计工作流模型建模。

由于存在多方案设计的问题，我们基于Parsec的分解模式，对于Concept层进行了WBS分解的扩展。

**Parsec的wbs分解方式[28]：**

PARSEC (Preliminary Analysis of Revolutionary Space Exploration Concepts) CEE(Collaborative Engineering Environment):协同工程环境这个平台给出的WBS的三层模式

基于WBS的3层分解模型：



Figure 10 WBS的3层模型

Project：表示一个复杂产品设计项目。有Architecture结构树构成。

Architecture:表示一个复杂产品设计分解出来的各个模块，所有的Architecture构成一个WBS树，是WBS第二层的主体。

Concept:其实相当于一套由于产品结构配置的不同导致的新的设计方案。

Task :表示设计每个Architecture的设计流程的任务。在这里特指设计任务，主要是指设计是调用计算工具或者子流程将输入参数转换为输出参数的活动。

实现多方案的模式主要是通过Concept层来实现。

总体设计由高层策划好各个分系统，各个分系统进行部件的WBS分解，形成了Architecture树，这一层一般属于分系统以及部件层，产品结构需要统一无二义，在每个叶子结点的Architecture的实现，采用多方案的Concept来实现。

## 工作流程建模

为了解决问题需要一个合理的模型来描述工作流程。

一个简单的流程如下图表示



Figure 11 流程示例

协同设计流程模型要素主要构成有：

**任务（Task）**：对于每个Architecture进行设计任务分解与定制。其中任务也可能是个子流程，在此处任务是获取一套输入参数调用计算工具或者子流程经过计算后生成一套输出参数，推送给下级关联的任务。

任务有输入参数集合，调用的计算工具或者子流程，输出参数集合以及执行角色组成。进行的主要操作时获得输入参数，调用计算工具或者子流程，推送输出参数给下级任务。

**连接 （Connector** **）：** 每两个活动间的关系叫做连接。连接上传递的内容是每个活动与下级活动关联的输出参数与输入参数映射以及在传递过程中调用对应的约束进行合法性检测。

成员Validator用来检测传入参数是否具备数据方面的启动条件，检测内容包括数据类型、任务输入数据约束、数据是否到位等工作，调用Validation Check()方法，如果返回为True则输入参数将被下级任务接受。

**约束（Constraint）**：表达参数的约束公式，建立在输入参数和输出参数基础上，与**Connector**关联**。**

**角色（Role）：**角色是组织中具有完成特定活动能力的参与者的逻辑表示。业务中的每个活动都要有角色与之相对应，并且需要属于相应角色的资源来完成。参与者可能会属于一个或者多个角色，反之亦然。

我们将协同设计工作流作如下定义：

**定义**：五元组N=(T,C, O, R，S) 其中

T表示任务集合**T**asks，C表示**连接集合C**onnectors, S表示约束集合C**o**nstraints，R表示执行角色集合**R**oles，S表示任务状态集合States。

令t∈T表示流程中的任务，T={t[1],t[2]…t[n]},

c ∈C表示流程中的连接如果t[i]->t[j] 则表示为c[i][j]=(t[i],t[j]),在DSM矩阵当中我们令c[i][j]=1;表示t[i]与t[j]存在参数关联。

o∈O表示每个任务相关约束，令o[i]表示对应任务t[i]的所有约束集合集合大小为k，s[i][j]表示对应任务t[i]的第j条约束，j∈[1,k]。当且仅当 对于任意j∈[1,k] 都有o[i][j]==true ，则t[i]就处于激活状态。

r∈R表示流程中的角色 **R**={r[1],r[2],…r[n]},其中r[i][j]表示参与第i个任务的第j个角色。

s∈R表示流程中每个任务所处状态，s[i]表示第i个任务的状态，任务状态集合包括{Waiting，Active，Executing，Executed}表示待激活，满足约束激活待执行，执行中与执行完毕4个基本状态。状态转移视图如下：



Figure 12 状态转移视图

为了描述工作流控制数据，我们可以采用设计结构(design structure matrix，DSM)作为描述工具。这是因为产品并行设计过程的重要特点就是**迭代和耦合**。

接下来我们开始进行协同设计工作流建模：假设设计流程活动由n个任务组成，表示为t[i] (i=1,2,…,n)，每个活动都有相应的输入和输出，活动之间通过参数约束建立联系，我们根据活动间的联系映射为一个n\*n的DSM矩阵如下



Figure 13 DSM矩阵表示

在矩阵中，矩阵维数代表设计任务的个数，元素t[i][j]表示活动间的关联关系，在流程中的具体含义是：对于t[i][j]如果i==j，由于任务i关联自己是无意义的，对角线上的元素我们可以不用考虑；如果t[i][j]==1且i≠j表示t[i] 向t[j]传递了输出参数，连接通过约束建立了t[i]输出与t[j]输入的联系。如果t[i][j]==[0] 且i≠j表示从t[i]到t[j]并未有关联。

顺序关系、并行关系、耦合关系我们可以用矩阵表示为：



Figure 14 DSM矩阵表示<一>

举个简单的包含三种关联的流程例子如下

Figure 15 DSM矩阵表示<二>

所以由DSM矩阵可以等价的表示出由此3种基本关系构成的协同工作流程。

**交互模式**

**1.上下级任务交互**

复杂产品设计过程中上下级分好后，落实在执行层。每个执行层初始任务的激活都是来自于输入参数的获取，同样包括上级对下级任务和流程的监控和分析，以及下级任务和流程向上级任务的汇总和统计，下列模型表示上下级参数如何通过参数推送形式实现的上下级设计活动的迭代并发流程。



Figure 16 层级交互模式

采用DSM表示以上关系：

Figure 17 DSM表示的层级交互

左上角的矩阵是上层的DSM，右下角红色矩阵是下层任务DSM，他们呢分别表示流程内部的逻辑关系，右上角蓝色3\*3的矩阵表示的是参数从上层下发到下层，比如从t2->t11，而左下角蓝色3\*3的矩阵表示参数从下层汇总到上层比如从t13->t3。

2.**同级任务交互**

在实际产品设计中，执行层的设计师的交互需求是最多的，也就是同级任务交互，协同设计过程当中不同设计小组之间存在信息的相关性，之间必然有参数协同的需求。接下来我们来处理这个问题。



Figure 18 并发任务交互模式

上图描述的是两个同级任务在并发执行过程中的参数交互，用相应的DSM表示如右图。

在图中最直接的并发问题就是参数的同步，体现在读写冲突。 如果不处理这个情况将引发数据不一致。我们解决这个问题的方法一个任务激活的条件是输入参数全部就位，不然就处于待激活状态。某个任务正在进行读取操作的参数。



Figure 19 写操作

向数据空间写入一个参数Param,生成版本V1, 由于Tto对Tfrom的Param参数有关联，因此Tto接下来读取Param参数的时候获取版本V1.如下图所示：



Figure 20 读冲突

如果Tto读取Param的时候，Tfrom对Param参数又进行了修改接下来就会发生读写冲突。解决方法：在两个任务的作用域范围内，对参数Param生成新的版本V2,不影响Tto的读取操作，并且将Tfrom的操作发布给Tto。这样就能避免读写冲突问题。而对于Tto来说输入参数多了一种可选方案，因此在不同参数版本的组合下，才会出现最后的多套设计方案。同时



Figure 21 读写冲突消解

涉及读写操作的还可能出现的一个问题就是死锁。如下图所示，在编制设计任务过程中，如果Tto与Tfrom发生耦合，且Tto的输入Param1来自Tfrom,而Tfrom得Param1又需要Tto的Param2来生成的话，那么流程将会造成**死锁**。也就是说在这种情况下，任何一个任务都无法激活。



Figure 22 死锁问题

遇到这种情况，从实际设计过程上说，一般会让某个节点作为“前置节点”，例如上图中的Tfrom，然后自动给予一个参数历史经验数值，使得流程可以向下进行，这个是迭代执行的特点。

使用DSM来进行流程建模，目前看来是实践过程中比较好的一个选择，首先DSM可以很直观的反应产品设计过程中的设计结构，对DSM进行进一步扩展不但可以表达任务与任务之间的关联关系，甚至可以表达出层次的任务与参数，参数与参数的关联关系。这个在第四章我们的工具扩展里做了实现。同时DSM很适合进行迭代计算，由于迭代计算其实就是参数与任务产生新的版本，而每次新版本的产生必然涉及到变更，采用DSM可以很方便的进行变更传播影响分析，从而能够很直观的了解变更发生后的波及范围以及产生的代价，从而能够给设计师一个很直观的参考数据来决定是否执行变更。另外由于DSM矩阵形式使得工作流程可以用二维数组进行存储，而使得一些流程的变换分析涉及的计算变得更加方便和简单。

## 模型总结

这一章的主要工作是对协同设计进行建模，包括上下级协同，同级协作的协同设计总体模型，以及描述协同设计过程的工作流模型。第一步采用的是WBS技术实现的协同设计3层分解，获得工作流建模的基本任务单元，第一层是决策层，节点负责项目总体规划与目标；第二层是管理层，负责产品分解后的各个子系统与部件的设计过程的管控；第三层是执行层，执行产品设计流程，主要角色是设计师。接下来工作流模型在第一步的WBS基础上，采用DSM技术来描述产品设计流程，包括建立每个步骤的任务以及参数依赖，定义对应的迭代设计任务，以及节点任务之间的信息交互。结合这两个步骤就形成了我们完整的协同设计模型。



Figure 23协同设计模型

从上图来看各设计单元都有相应的设计任务、过程、数据以及与其它设计单元的交互需求。协同设计模型可以分为总体规划阶段和设计实施阶段，前者分析需求、设计总体参数、分解任务以及设计约束关联；后者按照具体设计指标，以一定设计流程完成设计任务。所以协同设计过程实际上可以看成一个树形的分层的总体规划结构和相应的约束关联，以及一套具体的设计流程和数据交互关系。从总体来看，虽然各设计模块的工作流在局部并发执行，但从全局来看依托树形规划结构可以整合成为一个大规模的工作流，以参数的传递为基础表示任务的开始和结束，并控制流程的进度。

# 协同设计工作流工具的设计与实现

## 系统需求

协同设计工作流旨在为产品设计过程提供一个团队协同、参数协同、过程协同和工具协同的统一工作平台，并为项目管理人员，总设计师和设计人员提供参数管理、流程管理、工具集成、数据管理和知识管理工具。针对实际需求，协同设计工作流工具的 总体流程如下：



Figure 24 总体流程

流程定义阶段进行任务分解，生成任务的WBS分解结构，定义每个任务的设计师、输入输出以及约束条件。根据任务的输入输出依赖关系生成DSM流程。部署到引擎工具。在设计执行阶段，引擎启动流程后生成对应的流程实例以及对应的任务列表，每个设计师执行自己的任务，调用对应的设计工具进行参数的计算；流程管理员可以对流程与任务状态进行监控，对设计某个里程碑阶段进行基线配置。总体设计师可以提取过程参数版本进行多方案比较。

总体功能结构图：



Figure 25 工具总体功能

|  |  |  |
| --- | --- | --- |
| 1 | 流程定义 | 1. 进行WBS分解 2. 任务定义包括输入输出参数关联 3. 进行参数约束定义。 4. 计算公式的定义 |
| 2 | 执行监控 | 1. 流程启动，任务执行并进行参数提交，检测相关约束，执行任务审核。 2. 进行变更影响分析，对即将发起的变更进行影响范围的检查与评价。 3. 进行流程监控，显示任务当前状态，显示总体流程和各级子流程的当前状态 4. 基于过程历史数据建立基线，作为下一步任务执行的标准。 5. 对不同版本的历史数据进行配置，形成多套执行方案并进行比较。 |
| 3 | 系统管理 | 1. 对基础数据进行管理包括参数类型、项目状态、项目类型、任务参数类型、任务状态、任务类型、参数单位等 2. 进行用户以及组织角色的管理，以及系统权限管理。 |

用况图描述如下：



Figure 26 总体用况图

建立好描述模型后，协同设计工作流工具的整体流程我们划分为流程定义与流程执行两个大阶段。

**流程定义**

工作流定义阶段的工作主要是为用户提供任务分解、约束定义、计算公式定义等功能，形成工程的任务树，并为各任务提供结构视图、流程视图和设计结构矩阵等多种展现视图。工程定义中定义的任务，是设计团队中任务分配的基本单位，也是设计人员执行任务实例的任务模版。

**流程执行**

根据由参数组织决定的任务执行顺序，执行工程定义子系统中定义的任务。在执行任务的过程中，用户能够在系统中方便的使用一些常用工具。任务执行完毕后，需要具有特定权限的角色对任务的执行过程进行审核，审核通过后方能执行后续任务。

工程执行部分需求包括：任务执行、任务审核、任务变更、变更影响分析，共4个子功能模块。

**流程监控**

监控流程的整体执行状况，显示出当前流程执行到哪些节点以及流程中传递了哪些参数。并且能够展示任务节点的详细状态，包括当前的各个参数的信息及历史版本。

**系统管理**

系统管理为系统管理员提供对系统的基本管理，包括基础数据管理、修改密码、用户管理、功能管理和角色管理。

## 系统设计

系统采用流行的B/S结构模式。系统的分析设计采用面向对象的技术总体设计。

### 总体设计

**1）系统逻辑架构图**



Figure 27 逻辑架构图

每个角色从客户端登录后，进入身份验证，权限验证以及页面功能组织的控制层，通过后进入到业务逻辑操作层进行相关的操作，其中每次数据相关操作都需要通过数据接口进行数据库的访问。

**2）功能结构树**：



Figure 28 功能结构树

各模块描述如下：

|  |  |
| --- | --- |
| **流程定义** | |
| WBS分解 | 能够完成对任务的WBS分解，并且将WBS显示在界面中。 |
| 输入输出定义 | 能够定义参数，包括参数的类型，单位等属性，提供管理参数的机制，包括增删改等功能；让用户能够把参数与对应的WBS任务关联起来。 |
| 约束定义 | 能够定义各种参数的约束以及参数之间的映射表达式。 |
| 计算公式定义 | 支持用户直接在表达式文本框中录入或者修改计算公式。 |
| **流程执行** | |
| 任务执行 | 显示任务相关信息，任务执行者根据设计要求使用相关工具执行任务。 |
| 约束检测 | 任务提交后调用对应的约束进行输入参数的约束检测，并返回检测结果。 |
| 任务变更 | 在任务执行提交过之后，如果需要再次提交，需要执行变更申请。 |
| 变更影响分析 | 审核人在审核变更申请的时候，可以查看变更发生后可能受到影响的任务。 |
| **流程监控** | |
| 流程状态监控模块 | 监控流程的总体执行状况以及各个任务节点任务的进展。 |
| 任务基线配置 | 在项目的某些里程碑阶段，将对应产品的设计进度进行基线配置，从历史版本库中选取一套参数作为当前基线。 |
| 多方案比较 | 提取不同版本且符合一致性的过程参数，形成多套可行方案，在一定规则下进行方案比较，比较结果可以直观展示给设计师。 |
| **系统管理** | |
| 基础数据管理 | 基础数据管理提供系统中所用到的一些基础数据类型的管理，包括参数类型、项目状态、项目类型、任务参数类型、任务状态、任务类型、参数单位等。 |
| 用户管理 | 提供组织机构树的建立、维护和增加、删除用户的功能 |
| 权限管理 | 主要对工具的各个功能模块进行权限分配的管理。 |
| 角色管理 | 定义组织角色树，与用户形成多对多的关系。 |

**3）处理流程与模块关系**



Figure 30 各模块相互关系

上图描述的是从建模工具到引擎工具，每个过程产生的主要生成物之间的转化关系。建模工具首先进行任务分解，生成任务的WBS分解结构，传递给任务定义模块，任务定义模块对每个任务的设计师、输入输出以及约束条件进行了详细定义。根据任务的输入输出依赖关系生成DSM流程。部署到引擎工具。引擎启动流程后生成对应的流程实例以及对应的任务列表，每个设计师登录自己的门户可以执行自己的任务实例，调用对应的设计工具进行参数的计算；流程管理员可以对流程与任务状态进行监控，对设计某个里程碑阶段进行基线配置。

**4）主要的工作流工具的类图设计如下**



Figure 29 建模工具类图

首先流程是可以分解为任务树的，而任务有可能是子流程类型。以任务为核心，每个任务的主体包括参与角色，连接以及配置。其中配置相当于综合了任务相关的输入参数、输出参数、约束、计算公式（表达式）以及调用工具。基于此图，我们设计开发了我们的建模工具。

**5）主要的业务流程操作：**

建模过程：**生成WBS，构建Architecture层**。



Figure 31 建模活动图

项目高层策划进行项目WBS分解，WBS的输入输出定义，节点负责人委派，构造层级约束，生成总体DSM以及参数依赖关系图等建模工作。

**细化任务进行流程建模：**



Figure 32 任务定义活动图

每个节点任务的负责人对所负责任务进行IO参数定义，工具绑定，设计师指定，审核人指定，生成对应执行流程与DSM，部署到引擎并启动。

接下来进入**引擎部分**的执行过程：

**任务执行及审核流程如下**：



Figure 33 任务执行与审核

这个流程描述了设计师登录以后可以查看引擎生成的任务列表，选择任务进行执行。执行后提交管理员审核，管理员审核通过后将参数发布以及生成对应参数基线的过程。

最后是变更流程：



Figure 34 变更申请

变更流程主要是由设计师发起变更申请，然后会对变更涉及的参数进行全局变更影响分析，审核员根据分析结构可以决定是否接受变更请求，如果接受则执行变更，把变更任务放入执行任务列表，说明此任务可以执行变更。后续流程同任务执行。

把这些流程综合起来就构成了下面这张大的总体流程视图

Figure 35 详细流程

### 数据库设计

接下来我们根据设计开始进行数据库设计，WBS分解阶段的数据库设计如下：



Figure 41 WBS分解数据库表定义

从顶层Project开始分解为Architecture,以及实现Architecture的流程Process，最后获得整个WBS分解树。

接下来的任务定义相关数据库表，主要是关联任务执行人与输入输出参数。

Figure 42 任务定义参数表定义

约束定义与参数关联关系



Figure 43 约束定义数据库表

迭代运行变更记录以及版本基线相关数据库表



Figure 44 版本基线数据库表

从WBS分解到任务定义、参数关联、约束定义、流程执行、变更等所有协同设计工作流基本过程的数据库表设计如下：



Figure 45数据库表

### 算法设计

此系统中最主要的算法是迭代执行算法，变更影响分析算法以及多方案比较算法。

**任务状态转移视图与迭代执行算法：**



Figure 36 工作流引擎状态转移

**任务节点执行执行算法描述如下：**

**待激活：**

指具有如下特征的任务节点：不存在一个输入参数完整的可用版本序列。即不管如何选择输入参数的可用版本，都无法构成一个输入参数完整的可用版本序列。此处改为用户驱动

**激活待执行：**存在一个输入参数完整的可用版本序列。{Input}

**执行审核通过:** Input条件成立，另外存在一个输出参数完整的可用版本序列{Input,Output}。

1. 如果节点所处状态为{**激活待执行、执行审核通过、变更审核未通过、变更审核通过**} ，增加输入变更参数的新版本(变更的参数通过外部约束找到影响的参数进行同步版本更新)，然后提示用户有新变更是否启用，如果启用将任务改为变更激活待执行，否则不改变输入参数版本以及关联输出参数；
2. 如果节点所处{**待激活**}那么增加输入变更参数的新版本，然后提示用户有新变更是否启用，如果启用将任务改为变更激活待执行，否则不改变输入参数版本以及关联输出参数；
3. 如果节点所处{ **执行任务审核中、变更审核中**} 那么增加输入变更参数的新版本，等待审核完成后，进行新变更的提示

反馈的处理采用消息机制，即存在反馈线的任务，提供一个反馈的消息功能，将反馈的参数以及反馈的内容发送到目标任务，目标任务根据情况进行相应的变更处理。

**变更影响分析算法**

变更影响分析发生在任务提出变更申请之时，相关设计是需要关注此次变更对整体结构带来的影响及波及范围。



Figure 37变更影响分析流程



Figure 38变更影响分析算法

算法过程描述：

1. 某个节点的输出参数Yo<pid,n>发生变更;
2. 找到所有以Yo<pid,n>为自变量的外部关联实例R集合;
3. 如果R为空集则到步骤9,否则到步骤3;
4. 找到外部关联R的应变量Yi；
5. 找到以Yi为输入的所有任务；
6. 进行变更广播：通知受影响的任务以及被波及的输入参数Yi
7. 在T内找到以Yi为自变量的内部关联R`
8. 获得所有受影响的应变量集合Yo`;
9. 回到步骤11。
10. 结束。

**多方案比较算法：**

多方案比较的主要形式是设计师选中需要比较的参数集合，然后指定某个或者某几个特定参数的版本，对参数版本库进行搜索，获得其他参数的有效版本序列，最后构造出选定参数集合的版本依赖树Tpar，对于Tpar上的每个从根节点到叶子结点的路径都是一套设计方案。对于每套设计方案，设定入选可行方案的要求，比如某些参数的范围约束，或者更为复杂的函数关系约束，筛选出可选方案，提供给设计师进行多方案比较。

总体流程描述如下：



Figure 39 多方案比较算法流程

递归构造方案树算法描述：

**令方案树为Tpar，参数依赖关系集合为DSM。**

Tpar是一个四元组集合，**元素为(P, p\_ver,P\_son, P\_son\_ver);**

*参数P,以及参数P的版本和参数子节点P\_son以及版本P\_son\_ver相关联。*

1. 为了使得方案树存在一个共同根节点，我们先构造参数DSM的虚拟根。找出DSM中没有任何输入的参数集合P0=(p[i1],p[i2]…),如果SizeOf(P0)>1,添加一个虚拟根节点p\_virtual,指向P0中的所有元素。



Figure 40 构造DSM树

否则这个根就是P0中的唯一元素。

这样我们就得到了单根的参数DSM树。

1. 定义参数指针p\_Itr = P\_virtual，根据DSM树进行递归查询。

递归函数定义：

//此函数用来构造多方案树。输入为一个参数节点。

|  |
| --- |
| Tpar(Pnow){  Array Pson = getSonOfp(Pnow);**//根据DSM找出p\_Itr的子节点集合Pson,进行遍历;**  Array P\_ver = getVerOfp(Pnow)**//获取到当前参数的所有有效版本**  For(i=0;i<SizeOf(Pson);i++){  p\_Itr = Pson[i];  for(j=0;j<SizeOf(P\_ver);j++){  **//调用数据世系接口获得对应当前参数节点某个版本的依赖参数的版本集合**  **//输入为当前参数以及版本，以及子参数。**  **Array P\_son\_ver = getPsonVer(Pnow, P\_ver [j], p\_Itr);**  Add(Tptr, Pnow ,P\_ver[j], p\_Itr ,P\_son\_ver);//为方案树添加节点  //递归搜索子节点  Tpar(p\_itr);  }  **}**  **}** |

1. 在方案树Tpar构造完毕后，我们可以定义方案筛选规则集合S,然后对每条方案进行筛选，遍历Tpar树获取符合筛选方案，呈现给设计师进行进一步深入比较。

### 接口定义

引擎工具的外部接口主要包括以下几个：

1. **初始化用户配置方案**

根据任务编号从参数库中读取对应任务的配置方案，包括输入参数集合，输入参数集合的有效版本序列。

1. **冲突检测分析**

需要对所读取的数据进行冲突检测，对正在写入的参数进行加锁操作，以防重复写入。

1. **新输出数据方案写入**

把新生成的参数值写入参数库，同时生成对应新版本

1. **多方案比较**

可以同时读取生成的多套配置方案，给用户进行多方案展示。

**与工具集成接口**

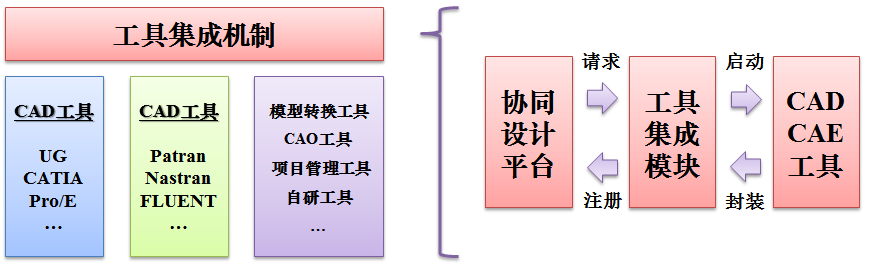


Figure 46 工作集成机制

工具集成机制如图所示，主要就是与常见的CAD工具集成数据集成，如何获取访问请求，启动工具然后调用工具借口进行参数的读写操作。

## 系统工具

用户通过使用建模工具可以直接进行可视化的工作流建模，建模工具主要分为如下几个部分：

### 任务工具

**——WBS工具**

WBS工具我们采用的是品字图与树形结构相结合的展示方式。

基本的WBS元素包括编号、分解名称、类型、上级节点以及基本信息。如下图所示：

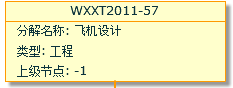


Figure 47 WBS元素

支持自由折叠和展开。



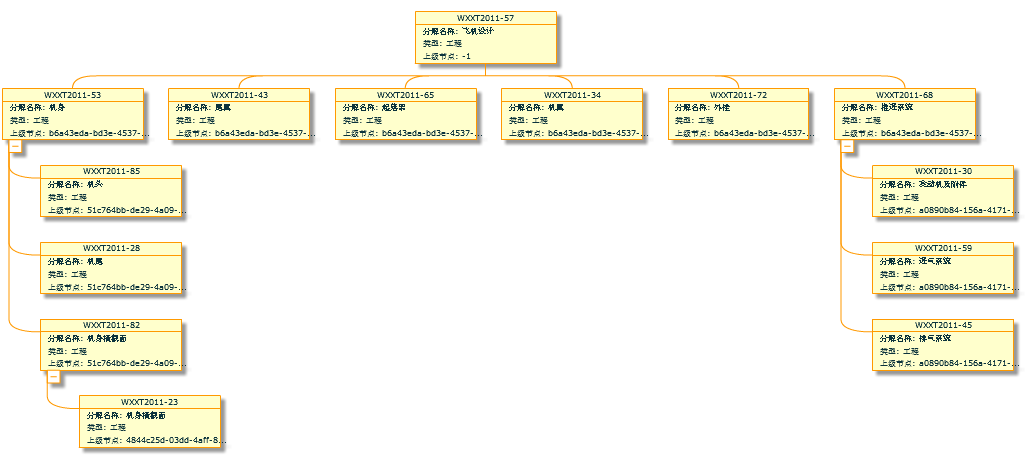


Figure 48 WBS品字图示例

与树形结构可以互相切换显示

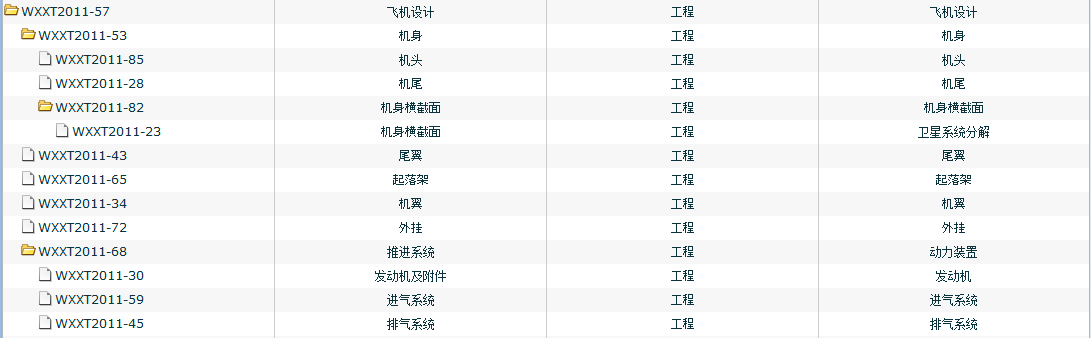


Figure 49 WBS树形图示例

WBS工具支持任务分解、任务定义以及执行角色定义等相应操作。

### 流程工具

**——DSM工具**

DSM工具主要用来显示任务间关联关系，以及作为流程的展现方式。



Figure 50 DSM工具类图

**对应流程的XML描述：**

|  |
| --- |
| <DSM>  <DSMNodes>  <DSMNode>  <Id></Id><Name></Name><Order></Order><Change></Change>  <FatherNode></FatherNode>  <InParams>  <DSMParam><Id></Id><Name></Name></DSMParam>  </InParams>  <OutParams>  <DSMParam><Id></Id><Name></Name></DSMParam>  </OutParams>  </DSMNode>  </DSMNodes>  <Mapping>  <MapItem>  <LeftParam></LeftParam><RightParam></RightParam>  <MapType>REF/FB<MapType>  </MapItem>  </Mapping>  </DSM> |

**常规 DSM的扩展——基于参数的DSM**

由于协同设计平台是基于参数驱动，所以为了更好的表达任务之间的关系我们将DSM的粒度细化至参数层，同时引入WBS的折叠展开，从而可以随意控制DSM的表达粒度。

基于参数DSM我们可以清晰和完整的看到两个关联任务之间的参数关联，且同样是以DSM的方式来表达。



Figure 51融合参数的DSM

1. 点击一个任务只展开到第二级，显示第二级的DSM图
2. 点击带+号的任务，可以做进一步的展开
3. 连接线通过参数与参数关联，如果任务未完全展开则关联至参数所属任务。
4. 根据二级展开生成等价的流程视图。

此工具可以反映任务与任务，任务与参数，参数与参数依赖关系。反映任务与任务关系的DSM用来进行流程定义；反映参数与任务关系的DSM用来展现参数在任务间的流动；反映参数与参数关系的DSM用来展现参数间的约束关联关系。

## 实例验证

### **开发平台与工具**

本系统采用J2EE最常见的MVC框架SSH(Struts+Spring+Hibernate)实现。

前端可视化部分采用的是ExtJs技术与Flex技术。

数据库采用的是关系数据库Oracle 10。

### 运行实例

接下来我们使用一个飞机设计的实例来进行工具验证。

飞机设计的基本流程是[29]：

 Figure 52 飞机总体设计过程

#### 设计流程定义

首先进行产品设计项目的WBS分解：

WBS结构树



Figure 53 WBS树形图

WBS的品字视图

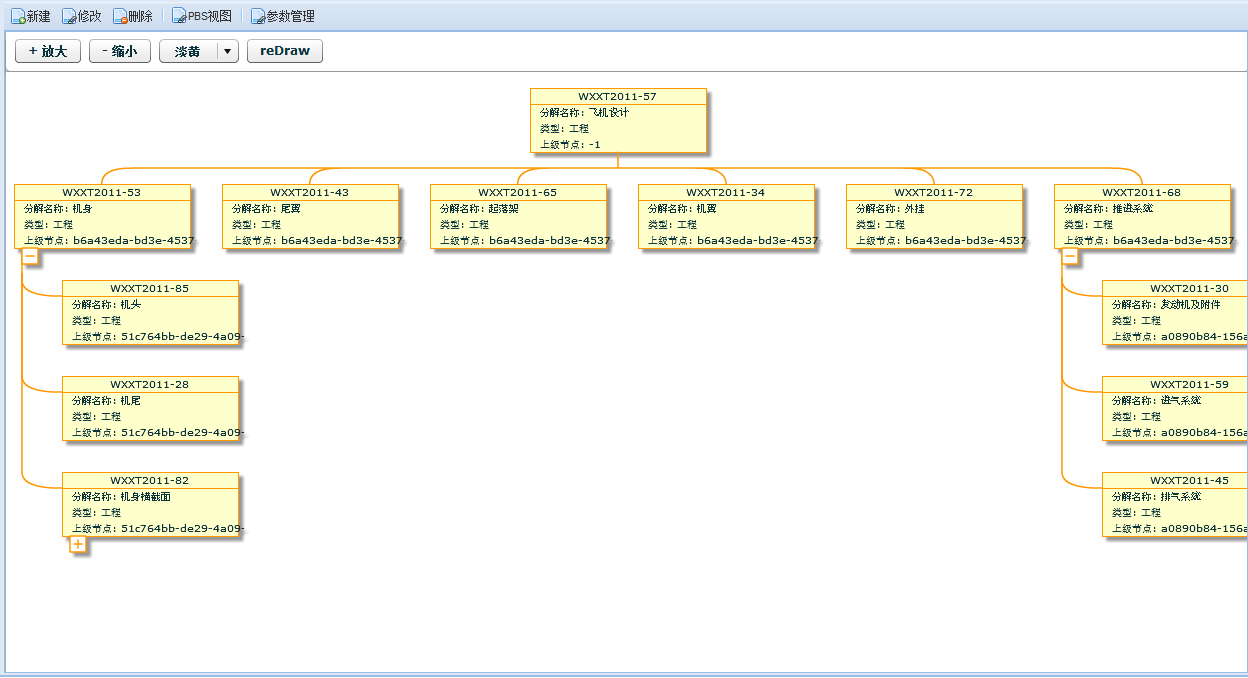


Figure 54 WBS品字图

完成WBS的定义后，再在系统中定义每个任务及其相应子任务，如下图所示：



Figure 55 任务定义

接下来为每个任务关联的输入输出参数。以总体参数估算为例。

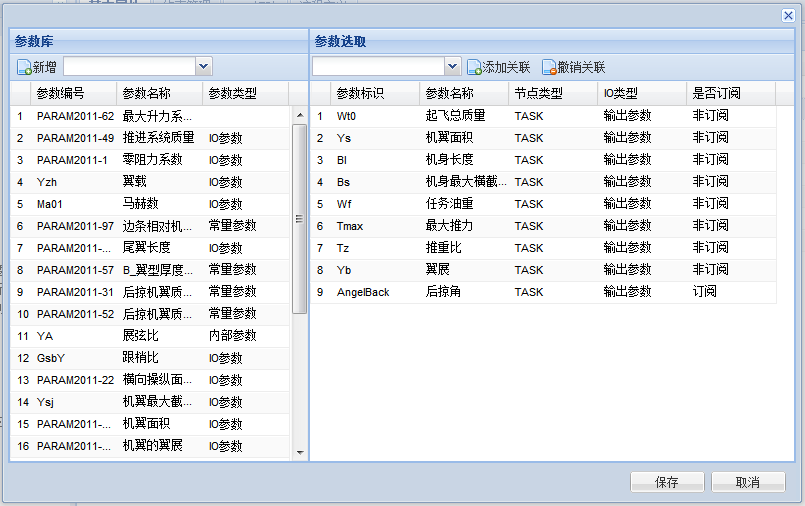


Figure 56 参数定义与绑定

指定任务执行人：

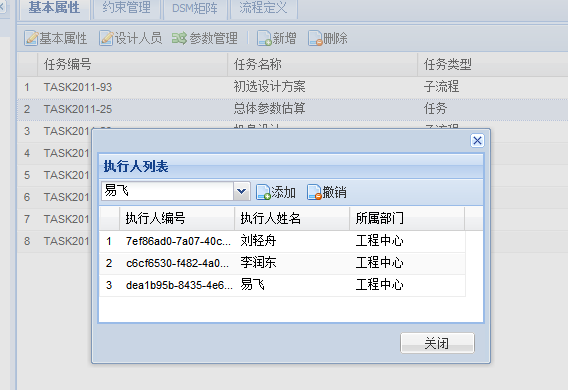


Figure 57 指定执行人

然后调用约束定义工具来定义各个任务的相关约束：

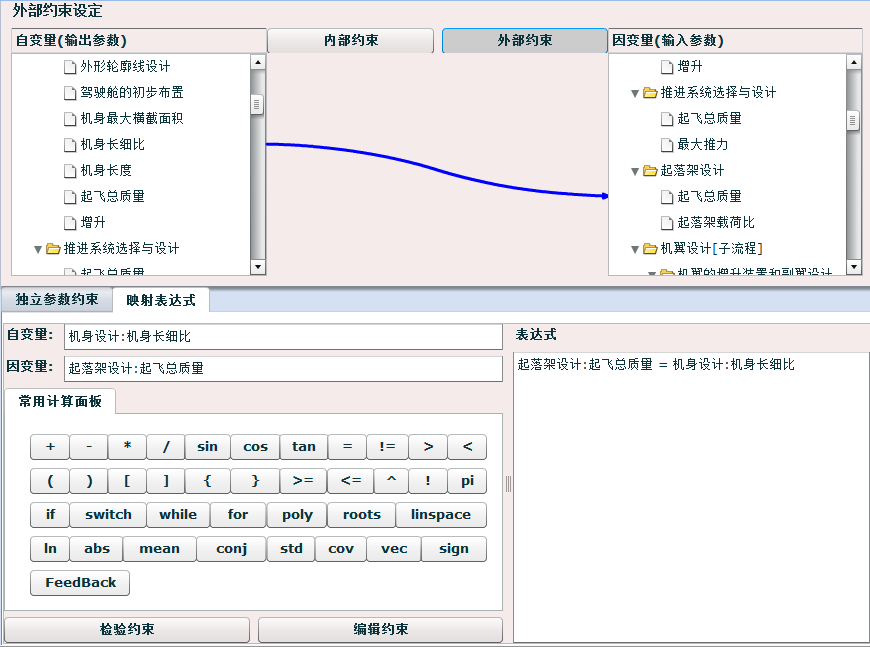


Figure 58 定义约束

定义好后所有的约束关联列表如下：

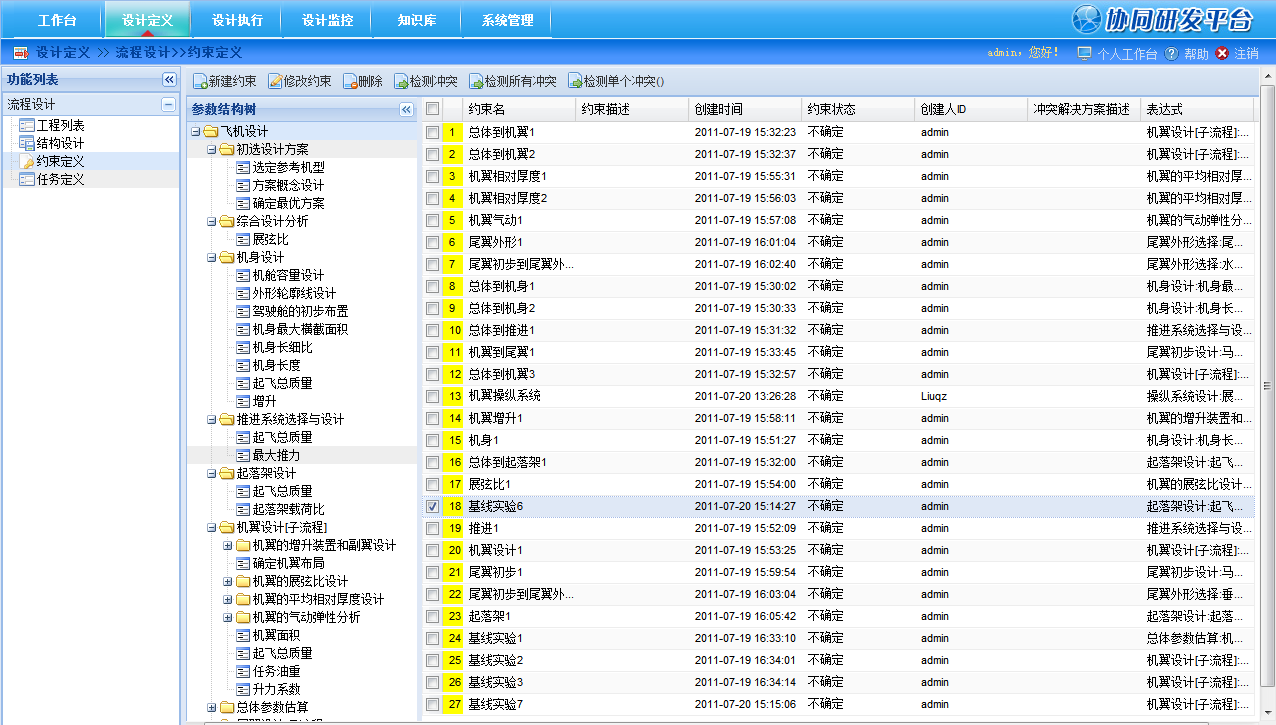


Figure 59 约束定义列表

接下来生成对应的DSM矩阵。

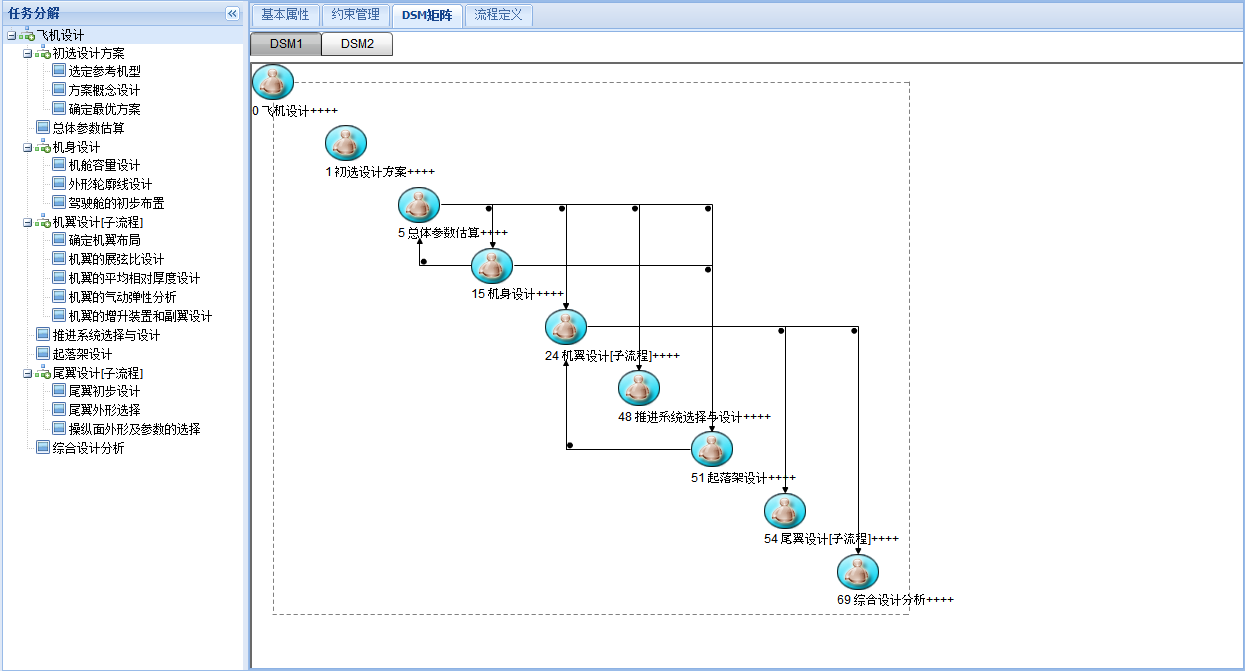


Figure 60 DSM视图

图中实线表示两个任务间存在输入与输出的关系。

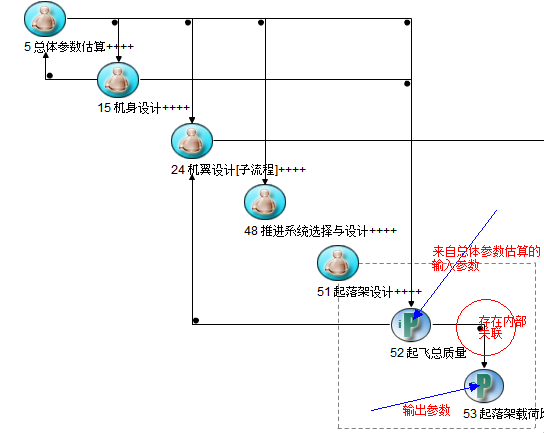


Figure 61 DSM视图说明

一个任务的参数来自其他任务的话可以在图中以实线箭头来表达。

根据DSM矩阵自动生成流程视图。

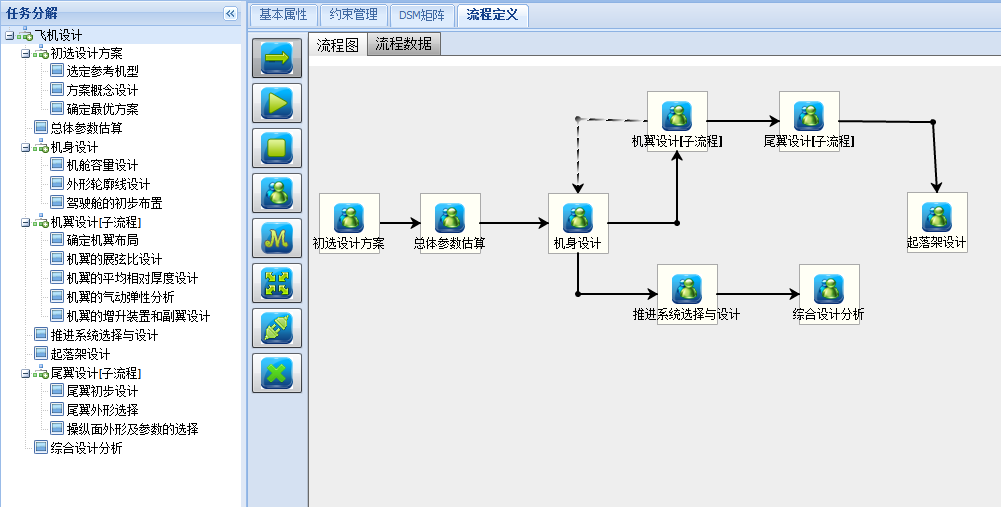


Figure 62 流程视图

相关XML表示：

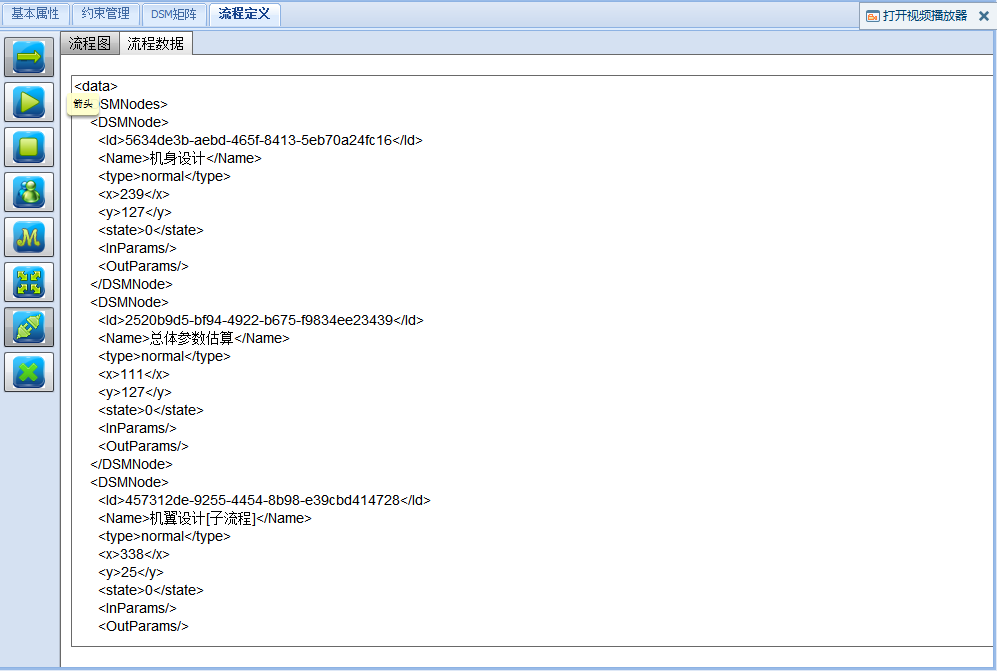


Figure 63 DSM模型的XML表示

至此流程定义的工作将告一段落。

#### 设计流程运行

启动设计流程后，会根据每个任务的执行人来分配设计任务。以设计师身份登录后，在任务执行界面可以看到自己每个任务的所处状态。选择已激活待执行的任务进行执行。其他任务由于输入参数还未获得，或者处于审核状态，因此无法执行。



Figure 64 执行任务列表

任务执行中，首选可以查看可选的输入配置方案，每个方案为一套可行的输入参数配置。方案由数据管理功能提供。每次计算新生成的输出参数一旦提交，便会等待变更审核，进入下个步骤。

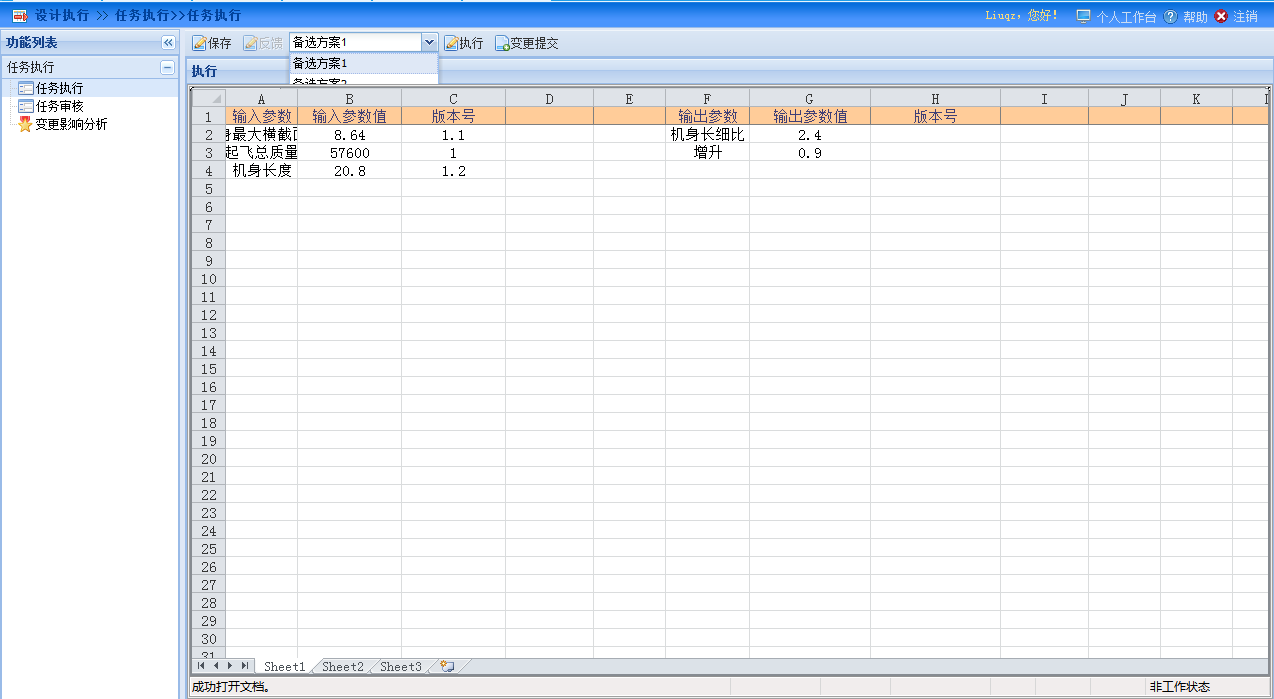


Figure 65 调用计算工具计算

任务提交审核后会以通知形式对下游进行广播，告知对应参数新版本已经产生。

**变更影响分析**

每次任务的执行，一旦参数发生变更，那么将对下游产生影响，变更影响分析就是用来对影响范围进行分析。受到变更波及的参数会用黄色来标注，而变更参数也可以用树形结构进行展开。

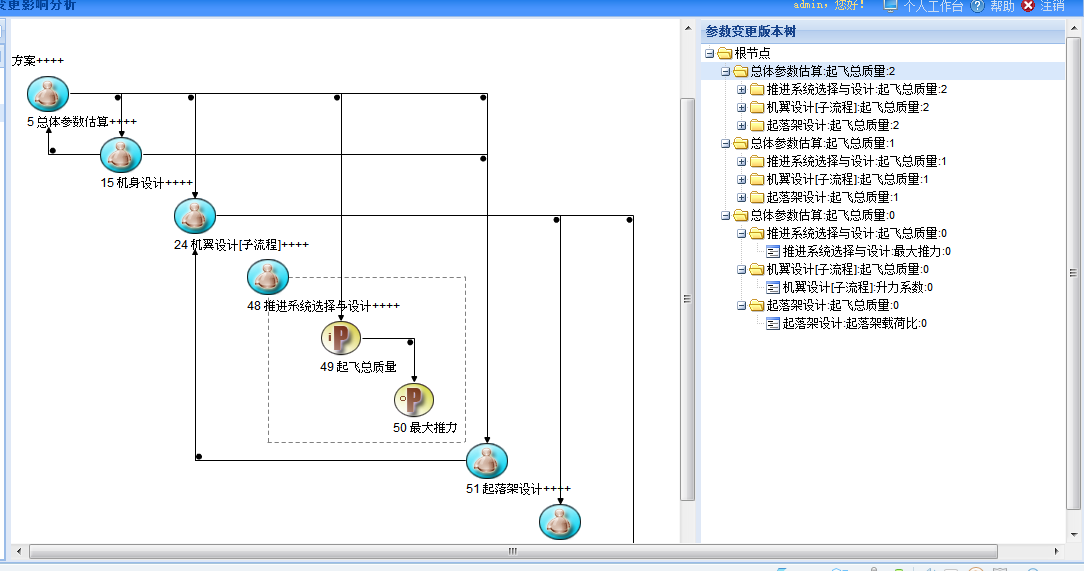


Figure 66 变更影响分析

#### 设计流程监控

**产品结构状态**

展示当前产品设计过程每个部件或者组件的参数详细信息。并且可以在设计的某个里程碑阶段进行基线配置——把某个版本的各个零部件的结构参数信息进行提取配置。

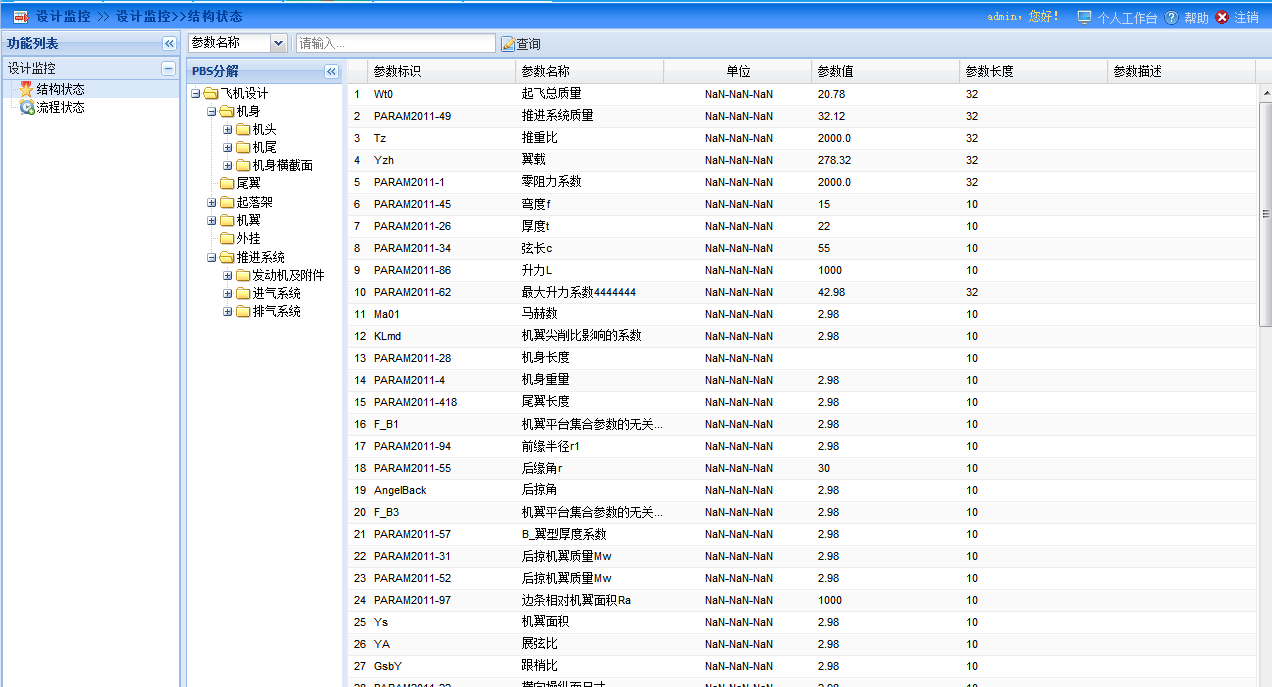


Figure 67 任务执行状态监控

**流程状态监控**

可以对流程的各个活动进行状态监控。

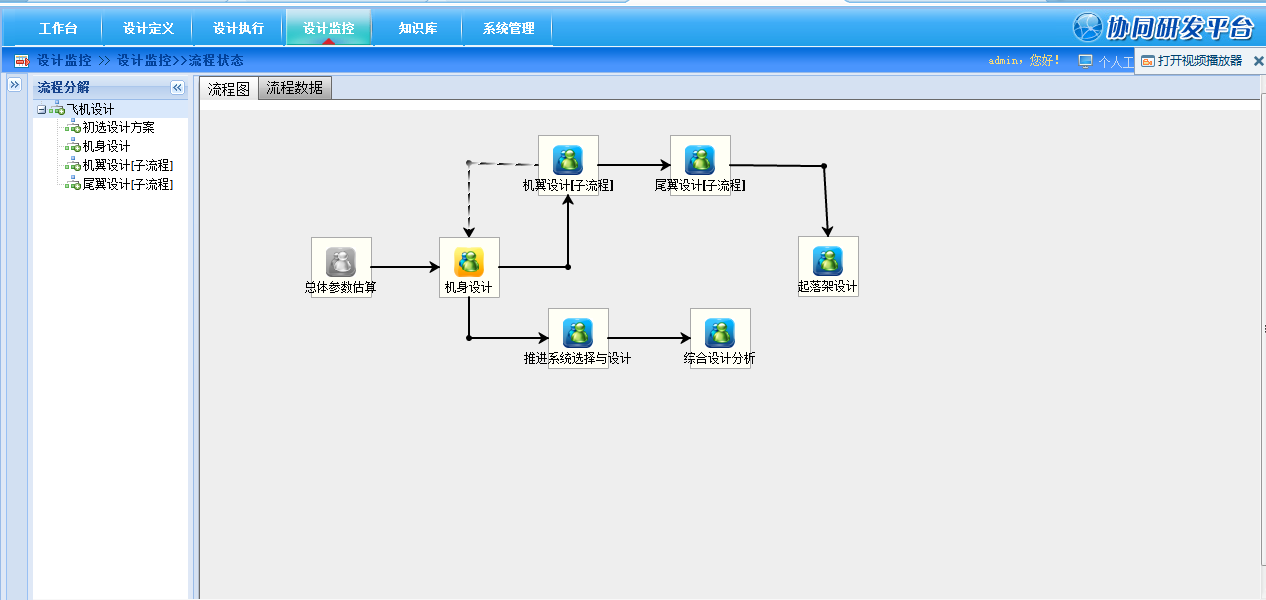


Figure 68 流程总体监控

**多方案比较**

由于设计是一个迭代的过程，允许自发变更，因此最后形成的设计方案也是很多的，此步骤就是从数据管理当中把符合要求的方案提取出来，并且可以给设计师进行比较筛选。

参数优化：

M :机重V:航速T:加速时间



Figure 69 多方案比较

使用函数工具：毯式图法确定基本参数（工具中未实现）。

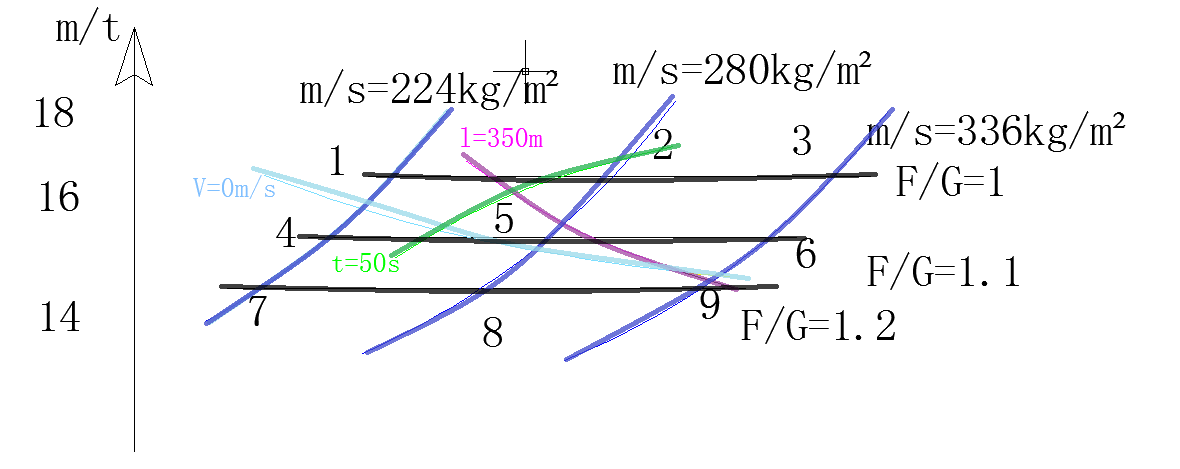


Figure 70 毯式图法

确定总体设计参数：



Figure 71 确定总体参数

主要流程在运行到这一步已算基本完成，在例子中我们把主要的WBS分解，任务定义，DSM定义，流程生成部署，任务执行、变更，监控，多方案比较的过程都运行了一遍，运行结果在前面的截图中都有相应的展示。

# 总结与展望

## 主要工作总结

本文首先针对当前复杂产品设计过程中存在的主要问题进行了分析总结：包括分布式团队设计、参数化地迭代设计过程，以及设计工具集成等。然后结合国内外的相关研究，在相关项目背景下，提出了协同设计的想法。

第二章论文对协同设计工作流所采用的关键技术进行了研究探讨，包括常见工作流技术包括传统业务工作流以及科学工作流，任务分解的工作分解结构技术WBS，设计结构矩阵技DSM，然后采用这些关键技术在第三章对协同设计工作流进行了建模，包括协同设计模型以及协同设计工作流模型。

第四章的设计与实现部分把工具主要分成建模工具与引擎两个模块进行设计，建模工具进行工作流可视化建模与展示，引擎负责流程的启动，运行以及监控等流程全生命周期的管理，使其可以满足协同的需求。

协同工作流工具当中，我们对通用DSM进行了扩展，从基于任务的DSM扩展成可以反映任务与任务，任务与参数，参数与参数依赖关系的综合DSM。反映任务与任务关系的DSM用来进行流程定义，反映参数与任务关系的DSM用来展现参数在任务间的流动，反映参数与参数关系的DSM用来展现参数间的约束关联关系。

最后我们采用了一个飞机总体设计的简单例子来验证了工具的功能与实用性。

## 下一步工作

在今后的研究中的我们的工作主要是：

加强协同设计数据方面的安全可靠性保障工作。包括输入输出数据的安全传输，以及过程数据的安全存储，尤其是针对高保密需求的军工国防的型号设计工程，对于信息安全的需求是第一位的。

建立并完善开放式的协同设计集成框架，协同设计工作流作为框架的核心组成部分，需要对各种CAD、CAE等设计工具以及其他设计辅助工具集成的支持更加灵活、广泛。在此基础上，实现协同设计平台的可定制可裁剪。设计师们可以根据用户的实际需求以及产品制造的领域特征进行协同设计过程与设计工具的自由定制，这样的协同设计平台才能够得到广泛推广。

总之，我们需要结合现在复杂产品设计的问题与特点，继续从原理、模型、方法等方面进行全面深入研究，与此同时利用现有资源多和国内外相关研究单位与复杂产品(船舶、航空航天等)研制单位进行交流学习，理论与实践相结合，从而建立更加完善的支持复杂产品的协同设计平台。

# 参考文献

1. 北大软件工程发展有限公司 ：《易方协同设计平台及应用情况汇报》。
2. 莫 蓉 《航空复杂产品协同设计的几个基本问题\*Several Basic Problems About Collaborative Design for Aviation Complicated Products》西北工业大学现代设计与集成制造技术教育部重点实验室
3. MarkR.Cutkoskyet.al，PACK: An Experiment in Integrating Concurrent Engineering System, IEEE Computer, 1993,26(1):28-37
4. 黄双喜.计算机支持的分布式产品设计方法与理论研究.南京理工大学博士学位论文，1999.
5. Dourish, P.; Bellotti, V. (1992). "Awareness and coordination in shared workspaces". Proceedings of the 1992 ACM conference on Computer-supported cooperative work. ACM Press New York, NY, USA. pp. 107–114
6. MarkR.Cutkoskyet.al，PACK: An Experiment in Integrating Concurrent Engineering System, IEEE Computer, 1993,26(1):28-37
7. W.I.Winner,The Role of Concurrent Engineering in Weapons System Acquisitio Report R-338.AD-A203/615，Dec，1988
8. 何浩、徐燕申，并行设计研究现状及其发展趋势[J],机械设计.1998，1,p1-4
9. Ludascher, B., et al.: Scientiﬁc workﬂow management and the kepler system. Concurrency and Computation: Practice and Experience 18(10) (2005) 1039–1065
10. Jae Yeol Lee and Kmangsoo Kim,A 2-D geometric Constraint SolverUsing DOF-based Graph Reduction,Computer-Aided Design,1998,30(11)
11. Xiao-Shan Gao and Shang-Ching Chou,Solvering GeometricConstraint Systems.A Global Propagation Approach,Computer-Aided Design,1998,30(1):47～54
12. Jae Yeol Lee and Kmangsoo Kim,A 2-D geometric Constraint SolverUsing DOF-based Graph Reduction,Computer-Aided Design,1998,30(11)
13. 并行工程 百度百科 <http://baike.baidu.com/view/50733.htm>
14. 安世亚太PERA,http://www.peraglobal.com/ch/index.html
15. 索为SYSWARE, http://www.sysware.com.cn/introduction.html
16. Workflow management coalition,Workflow process definition interface—XML process definition language[S],WfMC-TC-1025,Version1.0,2002
17. Workflow management coalition,Workflow process definition interface—XML process definition language[S],WfMC-TC-1025,Version1.0,2002
18. Workflow management coalition,Workflow process definition interface—XML process definition language[S],WfMC-TC-1025,Version1.0,2002
19. 工学博士学位论文 科学工作流管理及调度研究 刘灿灿 8-9
20. Guan Z, Hernandez F, Bangalore P, et al. Grid-Flow: A Grid-Enabled Scientific Workflow System with a Petri Net-Based Interface [J]. Concurrency and Computation: Practice and Experience. 2006,18(18):1115-114
21. Lin C, Lu S, Fei X, et al. A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution [J]. IEEE Transaction on Service Computing, 2009,2(1):79-9
22. 张卫民,刘灿灿,骆志刚,科学工作流技术研究综述,国防科技大学学报,2011,33(3):56-65.
23. Booz, Allen & Hamilton [Earned Value Management Tutorial Module 2: Work Breakdown Structure](http://science.energy.gov/~/media/opa/powerpoint/Final_Module_2.ppt), [Office of Science, Tools & Resources for Project Management](http://science.energy.gov/opa/project-management/tools-and-resources/), science.energy.gov. Accessed 27. Dec 2011.
24. Effective Work Breakdown Structures By Gregory T. Haugan, Published by Management Concepts, 2001, [ISBN 1567261353](http://en.wikipedia.org/wiki/Special:BookSources/1567261353), p.17
25. Program WBS Description MIL-HDBK-881A Figure 2-3
26. Steward D V.The design structure system: a method for managing the design of complex system. IEEE Transactions on erringeering Management ,1981, 28(3)：71-74
27. Yassine A A. An introduction to modeling and analyzing complex product development processes using the design structure matrix method. Italian Management Review,2004。
28. PARSEC CEE Programmer's Manual
29. 飞机总体设计:轩辕佩剑飞机设计方案总结报告 DT-1

# 作者就读期间科研成果简介

本人在攻读硕士学位期间，参与了如下相关项目的研发。

1. E2协同设计支持平台

平台基于协同设计理论，开展参数驱动的军工产品协同设计支撑环境研发，提供统一的协作门户平台，研究复杂的参数管理方法，实现一个参数依赖且可分级的军工产品设计过程管理机制，完成各种分析工具的统一集成。

平台的特点包括：基于版本控制和参数树的复杂设计参数管理技术，实现参数分解跟踪与任务之间动态参数关联和转换；基于浏览器的在线多级流程分解建模和协同工作的工作流程管理技术；支持流程的历史追溯与分析，包括流程中子流程、任务、参数等状态；采用多种分析工具封装方法，快速表达分析流程中工具；参数驱动的军工产品设计流程的固化与知识积累。

1. 航天五院协同设计平台

平台主要包含三方面功能：工作流程管理功能，包括工作流程定制、流程控制、节点任务处理、流程权限管理、单节点流程管理；文件管理功能，包括技术报告管理、数据文件管理、文件统一编号和分类管理功能等功能；过程数据管理功能，包括多种数据库类型、用户信息管理、项目信息管理、任务信息管理、应用程序信息管理、文件信息管理、数据库综合查询功能等。系统支持商用分析软件和自编分析软件的集成。除了完成必要的计算分析功能外，还能对各类分析任务进行规划、调度和流程定制，并借助过程数据库对动力学分析过程所产生的模型、文件按时间、型号、人员进行统一管理。

项目期间，并前往了航天五院，北斗二代半，武汉701所等多家军工单位调研协同设计平台需求以及学习相关的领域知识。论文将针对在项目过程中学习并发现的问题，做进一步深入和细致的研究。

北京大学学位论文原创性声明和使用授权说明

**原创性声明**

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名： 日期： 年 月 日

**学位论文使用授权说明**

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

按照学校要求提交学位论文的印刷本和电子版本；

学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务；

学校可以采用影印、缩印、数字化或其它复制手段保存论文；

在不以赢利为目的的前提下，学校可以公布论文的部分或全部内容。

（保密论文在解密后遵守此规定）

论文作者签名： 导师签名：

日期： 年 月 日

# 致谢

时光荏苒，转眼间就到毕业的时间了。回想当时初入燕园时自己的稚嫩，在北大里的这七年时间，成长了很多。现在的自己不仅仅收获了知识，学会了为人之道，更收获了很多的良师益友，在他们的无私的帮助下，自己渡过了美好的七年的北大时光，在这里衷心感谢他们。

感谢我的父母，是他们把我带到北京大学，辛劳工作，在经济上和精神上支持自己完成了本科四年以及研究生三年的学业，他们是我最要感谢的人。

感谢和自己一起走过三年的舍友和挚友，是大家的相互信任，相互支持，相互帮助，走过了这三年最美好的时光。

感谢我的指导老师张世琨，在我参与项目工作到论文写作的整个过程中，他不仅以身作则起到了优秀的领导和表率作用，更给予了我很多无私的指导与帮助。

感谢赵文老师，不但给了我很多机会能够参与到很多实际项目当中，指导我进行科研实践，而且在精神和物质上给予了我很大的支持与帮助。

感谢工程中心的刘学洋、胡文蕙、黄雨等老师。他们是益友，也是良师，在学术和生活上也都给了我很多建议和帮助。

感谢工程中心的所有同学，在我实习期间和论文写作过程中，得到过他们许多的帮助与鼓励。

最后再次得感谢所有关心和帮助过我的老师、同学、朋友和亲人。