

全球最佳实践翻译丛书之一

OBASHI 方法论

——业务与 IT 关系管理

上海迈未企业管理咨询有限公司
2012-3-1



版权声明:

本书非出版物, 仅供内部学习参考之用, 请勿外传并严禁用于商业公开用途!

OBASHI® 是在英国和其他国家的注册商标, 版权归 OBASHI 公司所有。

ITIL® 是英国内阁办公室(CO)在英国和其他国家的注册商标。

PRINCE2® 是英国内阁办公室(CO)在英国和其他国家的注册商标。

MSP® 是英国内阁办公室(CO)在英国和其他国家的注册商标。

P3O® 是英国内阁办公室(CO)在英国和其他国家的注册商标。

中文翻译团队:

薛凌云 上海迈未企业管理咨询有限公司 总经理

陆渊明 中国惠普有限公司 高级经理

刘伟 武汉北大青鸟网软有限公司 副总经理

介绍

这一节提供了一个 OBASHI 方法论可执行的概述。它介绍了主要的概念，以及它们如何通过集成成为业务战略与变革。

下面是章节的一个列表以及内容的概述。

第一章- 介绍

OBASHI 是什么？-描述了什么、如何、为什么是 OBASHI 方法论。

一些 OBASHI 术语-突出在 OBASHI 方法论中用到的关键阶段和术语。

OBASHI 项目生命周期-OBASHI 项目生命周期由一系列项目阶段构成，这些阶段使用 OBASHI 方法论执行项目。

OBASHI 如何支持业务策略？-描述 OBASHI 如何支持组织的前进以及驱动业务发展。

OBASHI 如何帮助您？-描述 OBASHI 如何帮助组织中的不同角色。

OGC 最佳实践指导-OBASHI 方法论可以配合 OGC 的最佳实践，包括 P3O,MSP,PRINCE2,ITIL。

如何使用这本指南-描述本手册的章节。

第二章- OBASHI 与业务策略

你可以以清晰和有视野的方式开发与提高-描述了 OBASHI 设计的四个关键理念。

清晰（Clarity）-从此告别歧义；

视野（Vision）-展望的力量

开发（Develop）-提供能力

改善（Improve）-成为了一个更好的业务

第三章- OBASHI 与数字流

流的历史-描述如何理解流已经成为经济发展的一个重要部分。

通用语言-描述几百年来图表与图片如何以通用语言的形式为工程师、架构师服务。

流测量的经济学-描述理解流对于业务的价值。

IT 存在仅仅有一个原因-支持数据的流动-描述 IT 如何支持数据流的机制。

数字流-描述 OBASHI 与数据流之间的关系。

第四章- OBASHI 与业务变革

开发大图-描述 OBASHI 如何用来形成与映射组织策略与业务与 IT 图带来的帮助。

开发‘如果-如何’场景以及被认可的未来的业务体系结构-描述 OBASHI 图如何帮助询问与回答关于业务现在的与未来的状态的问题

开发蓝图-描述 OBASHI 如何用来创造一个新的策略图,该图显示了业务在一个地点、部门、系统或者服务级别将是什么样的样子。

开发详细图-描述 OBASHI 如何帮助实现业务解决方案相关的工程的策略。

运营与控制-描述了 OBASHI 如何提供被完全开发出来了与有成熟业务知识的人、流程与技术。

第一章-介绍

1.1 什么是 OBASHI?

OBASHI 是一个方法论: 一个思考的方法, 能够帮助你能够获得你业务如何运作的清晰视图。

1.2 OBASHI 如何运作?

OBASHI 方法论将是你能够制造出你的业务的视图。

该图将显示:

- 您的业务是如何运作的;
- 使业务运作的资产;
- 这些资产之间的相互依赖。

1.3 为什么 OBASHI 如此重要

OBASHI 是围绕着四个关键概念设计的, 可以描述为: **你可以以清晰和有视野的方式开发与提高。**

- **清晰 (CLARITY)** OBASHI 方法论生成了易读的图标, 这些图标可以帮助你进行业务决策时免于歧义性的困扰;
- **视野 (VISION)** OBASHI 方法论使你能够预期有可能组织业务达到目标的挑战与问题。
- **开发 (DEVELOP)** OBASHI 方法论能够通过成功的程序与项目的实施提升业务能力。
- **改善 (IMPROVE)** OBASHI 方法论能通过更好的运营管理改善业务性能。

OBASHI 和业务策略的细节在章节二中进行描述。

1.4 一些 OBASHI 术语

OBASHI 方法论生成业务与 IT 图 (B&ITs)。在图 1.1 中显示了一个例子, 业务与 IT 图提供

了一个框架，这个框架用于记录人、流程和技术是如何相互作用从而生成一个业务工作进行记录，并对这些信息进行建模。

一个业务与 IT 图有 6 个层组成，由水平线分开，从上到下依次为所有者（Owner）、业务流程(Business Process)、应用(Application)、系统(System)、硬件(Hardware)以及与基础架构层(Infrastructure)。他们第一个字母拼在一起形成：OBASHI，如图 1.2。

元素在这些层中代表业务中的人、流程与技术。

所有者与业务流程层拥有那些业务如何运作的相关信息。所有者层拥有那些业务中相关人的信息。包括雇员、利益相关者、部门以及地理信息的相关信息。业务流程层记录了业务流程即业务如何运作的信息。

通过将业务流程元素放置在所有者元素之下，我们可以很容易的看到哪些部门与人员使用哪些业务流程。这将有利于构建利益相关者图，该图可以帮助辨认出关键个人，这些人将受到流程的改变的影响或者受益于流程的改变。

下面四层记录了 IT 资产的信息。这些资产是数据流的驱动者，使得业务运作。应用层拥有应用元素，这些元素代表了业务流程所用到的软件。任何一个业务流程用到的软件在业务与 IT 图中都被放置到该业务流程下，这样可以很容易的看到哪些应用用来支撑该业务流程。

这种相关元素的位置关系在系统层、硬件层、基础架构层被重复的使用，其中系统层包括了操作系统，硬件层包括了计算机硬件，基础架构层包括了网络设备。

在今天的商业中，数据在不同的业务流程中流动，IT 是流的驱动者。一旦业务与 IT 图被创建，数据流可以在其上叠加。数据流是 21 世纪商业的心脏。支持、优化、流程化数据的流动对于业务性能最大化至关重要。

OBASHI 数据流分析视图（DAV）是所有业务与 IT 资源的图形化与统计化的表现形式，并对数据流赋予了财务方面的特性，从而支持单一的数据流。DAV 为那些业务专家揭示了 IT 为什么、如何与每天的业务流程交互。图 1.3 是一个 DAV 的例子。

OBASHI 使得**影响分析**能够得以执行，同时 OBASHI 拥有**影响规则**来决定分析如何进行。**OBASHI 规则**定义了如何在业务与 IT 图中放置元素可，以及在不同类型间的元素可以有哪些关系。

OBASHI 图表（业务与 IT 图和数据分析视图）去除了模糊性，方便了沟通，并提供了一个用于分析的平台。

1.5 OBASHI 项目生命周期

OBASHI 项目生命周期是一系列的项目阶段，在这些阶段中通过使用 OBASHI 方法论来

执行项目。

Section C 中详细描述了 OBASHI 项目生命周期的灵活性以及它是如何与现有项目管理方法论进行区分的。

项目生命周期阶段的简介如下：

1.5.1 界定范围

界定范围阶段定义了哪些是用 OBASHI 来分析的业务活动。

界定范围阶段要生成一系列项目生成的相关文档，其中定义了工作的方法以及关键产出结果。典型的相关文档：项目移交、项目交付物、利益相关者、交流策略、控制流程等。

1.5.2 获取阶段

获取阶段收集并分析用于创建 OBASHI 模型的信息。

该阶段包括识别数据资源的行动，对利益人相关进行访问、评估已经收集了的数据。

获取阶段有可能得出需要重新进入界定范围阶段，并对界定范围阶段的结论进行修改。

1.5.3 设计阶段

在设计阶段，将在获取阶段获得的数据进行建模，建模业务与 IT 图以及数据流成为本阶段的关键行动。

生成业务与 IT 图以及数据流分析图是用来详细描述在界定范围阶段识别出来的项目交付的特定领域。

设计阶段有可能得出需要重新进入界定范围阶段或者获取阶段，并对它们的结论进行修改。

1.5.4 细化阶段

细化阶段深入业务与 IT 图，增加所需要的更详细的信息。

该活动保证能够终结业务与 IT 图中由于需要进一步信息所暂时搁置的任何部分，同时为交接项目与注销项目做好了准备。

精细化阶段有可能得出需要重新进入界定范围阶段或者获取阶段或者设计阶段，并对它们的结论进行修改。

1.5.5 移交阶段

移交阶段确保了在项目中生成的移交物已经被纳入实际运作中的业务环境。

这将意味着要移交一个 OBASHI 模型或者一系列的 OBASHI 图表。

在这个阶段，一个完整而成熟的基于人员、流程和技术的业务知识将被获取，并被细

化到 OBASHI 模型可以被分类为一系列的主文档，这些主文档用来描述项目界定阶段标注出来的部分业务现在如何工作。这个模型可以由许多不同层面复杂的图表组成。主文档应该像其他重要的业务文件一样的控制与管理。

一旦主文档建好了，他们就组成了他们所代表的业务需要进行任何修改的基础。

变更的动力可以来自不同战略或者运营的建议-比如，新产品的介绍，一个针对财务压力的响应、一个技术问题等。

1.6 OBASHI 如何支持业务策略

现代业务持续的承受商业压力。新的业务运作方法和在一个全球以及更加竞争性的市场运营增加了分析一个业务如何运作，进而如何优化、设置其优先级的需求。

获取该需求的一个被认可的方法是将组织的目标以及这些目标的含义采用一种可以将他们对组织的意义可以被评估出来的方式匹配出来。这可以通过评估他们对现在和未来组织的重要性来评估。

表 1.5 显示了组织目标和业务倡议如何被分成四个关键领域来成功的管理组织变更。通过采取这种方法匹配目标，他们可以优先的成为业务发展的战略或者优先的作为提高效率、效用的战术变更。这个方法也可以将动力对业务的重要性以及危险的等级进行分类。

OBASHI 方法论可支持每一个关键领域的决策选择流程，通过提供业务如何运作的大图。OBASHI 的大图：

- 组织内各层次都可以进行交流；
- 将人员、流程、技术放在同一个业务上下文中；
- 显示在通常的情况下，业务如何获得它的目标（BAU）；
- 提供一个平台用来分析和模拟策略性地发展和风险性地机会；

表 1.1-角色与关键问题

你是谁？ 首席执行官 CEO	关键问题 我的业务是怎么工作的？ 我如何描绘我对我所拥有的业务的策略性抱负。 我的风险在哪里？
首席财务执行官 CFO	我如何评价 IT 对于我的业务的贡献。 我可以在哪里降低费用。

首席信息官 CIO	我怎么能看到对于第三方支持的依赖； 我的安全弱点在什么地方？
主管比如市场主管、销售主管、运营主管	我能在部门间提高沟通的效率吗？ 我的部门是什么样子的。 重构对提高效率有意义吗？ 我的意外事故计划能起效果吗？ 如果要改变应该是怎么改变？
战略的头	按照策略来看，组织战略的方向是什么？ 为了达到该战略，战略风险是什么？
项目组合管理专家	不同的项目群在什么地方交互？ OBASHI 可以给我在开发一个优先化和优化的项目组合上提供什么帮助？
业务分析师	谁是业务流的所有者； 我怎么描绘业务流程； 我怎么知道谁是利益相关者、什么时候我们开始考虑变化？ 我怎么知道在进行的变化中哪些其他流程、部门或者系统会受到影响？
基础架构管理者	我怎么知道我所拥有的基础架构哪里被暴露了，哪里有弱点？ 如果我认为应该去替换一个硬件或者软件，我该如何去发现什么会受到影响？
项目群管理者	哪个业务流并不直接受到我的项目群的影响，但是却被我的工作影响？ 我的项目群的范围是什么？ 我的项目群相对其他项目群之间的依赖关系有哪些，反之亦然； 我的项目群对交付公司的战略有什么贡献？

1.8 OGC 最佳实践指导

OBASHI 方法论可以与 OGC 已有的最佳管理实践：P3O,MSP,PRINCE2 以及 ITIL 密

切的结合。OBASHI 显示了在项目组合、项目群、项目管理以及 IT 服务管理之间的交互关系，而且可以在交付一个项目组合的变更的每个阶段都得到应用。

1.8.1 项目治理办公室（P3O）

一个 P3O 模型提供了相关决定，该决定启用或者交付一个组织内的所有变更的支持结构。OBASHI 可以用于所有 P3O 的流程，来帮助其获得这些战略性目标。附录 A 记录了更详细的信息。

1.8.2 成功管理项目群（MSP）

MSP 的目标是帮助组织去协调、执行以及监督一系列相关的项目的实施，并采取行动来交付与组织战略目标相关的结果和收益。

OBASHI 可以通过在关键点定义所期望的未来状态，同时保障与关键相关干系人有效果的沟通，从而帮助 MSP 有效实施。附录 B 记录了更详细的信息。

1.8.3 受控环境下的项目（PRINCE2）

PRINCE2 是一个结构化的方法，它可以帮助对项目实现有效的管理。OBASHI 方法论可以通过定义所期望的未来状态和保障有效果的沟通支持 PRINCE2 过程。附录 C 记录了更详细的信息。

1.8.4 ITIL

ITIL 提供了一个系统和专业的途径来对 IT 服务的提供进行管理。

OBASHI 为设计、管理、知识为基础的 IT 服务、基础架构提供了一个业务上下文，并在 ITIL 项目的很多方面提供了许多相关帮助。附录 D 记录了更详细的信息。

1.9 如何使用这本指南

这本指南是一本描述 OBASHI 方法论原理性指南读本。它的设计是用来向任何大小的组织的主管、经理、资深管理者以及专业人士介绍 OBASHI 业务、IT 概念以及 OBASHI 方法论。

Section A 是一个 OBASHI 方法论的介绍以及简介；

Section B 提供了理解 OBASHI 的更详细细节。它的目标读者是那些打算获得在工作中如何使用 OBASHI 方法论知识的人。

Section C 是为那些打算生成、分析和模拟一个 OBASHI 业务模型的专业人士准备的。它描述了开始一个 OBASHI 项目和业务与 IT 图展示的技术面貌所需要的实际步骤，如何建模，以及分析技术。

Section D 包括一些附录，涵盖了 OBASHI 如何和 OGC 最佳实践一起工作，OBASHI

的一些学习案例，一些可以帮助实践者建立 OBASHI 模型的模版文档，以及一些词汇表。

- 附录 A –P3O –OBASHI 如何适合 P3O;
- 附录 B-MSP-OBASHI 如何适合 MSP;
- 附录 C-PRINCE2-OBASHI 如何适合 PRINCE2
- 附录 D-ITIL-OBASHI 如何适合 ITIL;
- 附录 E-学习案例;
- 附录 F-企业体系结构;
- 附录 G-其他类型的数据流;
- 附录 H-模版文档;
- 附录 I-包括在本指南中所使用的术语的列表和解释。

第二章 OBASHI 和商业策略

2.1 你可以开发与改善随着清晰与视野

OBASHI 是围绕着四个关键概念设计的，可以概括成一句话**随着获得清晰与视野，你可以开发新的业务能力、改善产品与服务的表现。**

OBASHI 努力去交付：

- 清晰 通过一系列易读的文档帮助客户去除模糊性与二义性，同时容易被业务和 IT 理解；
- 视野 来展望那些可能阻止你获得所期望目标的问题和挑战；
- 一个方便的手段来开发一种业务能力，通过该能力能够实现对主动性变更、成功的项目群管理以及对项目实施进行合适的优化；
- 一种能够改善性能并通过更好的运营管理交付客户更高质量的产品以及服务的能力。

OBASHI 也可以与其他的方法论一起来使用，这些方法论已经开发用来给组织做指导，帮助组织实现一个非常复杂的任务，即将战略业务目标匹配到为了满足组织自己的战略而需要进行变更的路径图上。

尽管 OBASHI 并不打算替代任何这些其他的方法论，这些方法论的许多核心原则都可以被 OBASHI 交付。

2.2 清晰-摆脱模糊性与二义性

OBASHI 方法论使你能够生成你的业务如何工作的高层与详细的文档描述。

这种文档描述的中心点是一系列的图表，这些图表很容易理解，并且不需要读者是拥有特定领域知识的领域专家。翻译业务与 IT 图表并不要求有特别的、专业的知识。

业务与 IT 图表它们自己就可以扮演讨论文档的角色，不同方面的职业人士可以使用它们来获取或者描述他们做什么以及使用什么工具和信息他们可以用来显示他们的任务。

业务与 IT 图表提供了一个机制可以把这些信息全部抽取出来，然后放到一个单独的资源中，使得不同专业的职业人士能够有一个清晰视图，来显示他们领域的专业技能是如何用来支持业务的，并能和其他人就他们对于组织的贡献进行有意义的讨论。

由于这些图表提供了一个关于业务的所有层次的精确视图，因此关于业务流程如何运作，谁是使用这些流程的利益相关者，或者哪些资产是为了使这些流程开始运作的必需条件，这样的一些方面都不会再有模糊与二义性。

这些关于业务运营的清晰视图会经常被其他方法论作为通常的业务视图（BAU）引用，比如 P3O 和 MSP。

2.3 视野-预期的力量

在战略层，大部分组织拥有一系列的长期目标和抱负-一个关于组织将如何演变的视野。当战略业务决策被决定，比如进入新的市场，业务决策应该与组织所期望达到的战略目标方向一致。

通过拥有，并维护一个 OBASHI 业务与 IT 图表，可以进行分析来保证能够获取组织变更的路径图。

任何可能阻碍未来战略目标的当前运营环境都可以被识别，并被视为一组项目群，该项目群是组织为达到其所期望的而进行的变革活动。

一般情况下，有太多的问题被掩盖在项目群的活动，导致了预算超标，或者单独项目完成的延迟，甚至经常威胁到项目群中应该被实现的收益。

通过使用 OBASHI 方法论，能够预期这些不好测的事件，他们的影响能够被更好的管理。

2.4 开发-提高能力

通过项目的交付，业务能够获得提高了的能力，顺利的项目交付对于有效果的业务变革至关重要。

生成或者维护简洁的项目或者运营文档可以是一个极端沉重的任务，因为在业务与技术资源之间有着非常复杂的关系。

OBASHI 可以生成清晰、逻辑和易理解的文档，允许业务和 IT 专业人士沟通的更有效果，这使得：

- 通过更清晰的文档使得沟通的位置更加重要；
- 通过更好的系统分析能够更好的节约成本；
- 更容易的计划；
- 更容易识别冗余的设备；
- 对 IT 项目更详细的理解；
- 项目的沟通更加顺畅；
- 与第三方的沟通更加清晰。

2.5 改善-成为一个更好的业务

OBASHI 方法论设计是为了使得业务工作更有效率和更有效果，从本质上成为一个更好的业务。

有效果的沟通对此至关重要，同时也是构建一个组织中的联系的核心。一个业务需要好的沟通以便真正的从该领域获得收益。OBASHI 的关键正是这个思想，即业务与 IT 图表采用了一种简便、可达到的方式反映了业务和 IT 的信息，使得它成为更清晰沟通的理想媒介。

在任何组织内，审计都是一个提供公正的保证方法，同时审计也能增加组织运营价值。由于审计本身的本质，它是一个咨询活动，而审计一个 IT 系统是如何支持业务行动也是一个非常消耗时间的事情，尝试去定义系统的实际边界本身也是十分的复杂。

这是因为功能性 IT 系统如果被孤立的审视，他们就不能够被有效果的审计。除非得到一个从整个系统中贯穿的流程视图，否则就有可能忽视那些能够导致失败的领域以及关键的风险领域。

设想下面的一个例子来帮助我们理解一个进程如何从传统系统边界贯穿。想象一个技工被调度分派到一条产品线去修理摩托上的一个问题。在该技工离开其现在的车间，他需要被工作调度系统发布其从现在工作调出来的信息，接着他需要从 CAD 系统中调出该摩托的图纸，接着他需要从设备数据库中取出该摩托的规格说明，从图纸管理系统中取出安装图纸。同时，他需要从物资管理系统中得到该店的闲置的零件信息，或者调用请购单系统，最后要生成一个工作允许让别人知道他正在产品线的哪个领域工作。

这六个系统中的任何一个的一个问题都会对修理摩托问题的这个事情造成影响。只有当该业务所使用的 IT 系统被理解，以及理解它们如何成为一个整体才适用于该业务流程，这些系统失败对组织所造成的影响才能够被评估出来并被改善。如图 2.6

通过提供易于理解的信息，OBASHI 可以减少收集和彻底消化这些信息所需要的人工时间。通过使用业务与 IT 图表作为讨论文档，审计者和业务以及 IT 专业人士才能共享他们关于业务如何运作的知识，从而达到改善业务的目的。

第三章 ---OBASHI 和数据流

3.1 流的历史

自从工业革命开始，对于流的理解（第一个是从水流，其次是蒸汽，再接着是电、接着是油、原油和机器组件）已经成为业务、经济以及社会的关系生死存亡的因素。

1770s 机械化、工厂和运河-水

1830s 蒸汽机、煤、铁路-蒸汽

1870s 钢铁以及重工业，电话，制冷-电力

1910s 石油、大规模生产，和汽车-石油；机器组件-汽油。

随着时间的推移，体系结构与工程方面的专业人士以及很多科学家，共同合作去开发标准以及实践来测量管理、安全、优化以及价值量化，来精确的理解这些不同的流。

一个业务是如何工作已经清晰的被理解了。

今天，流已经无处不在：电力在家中流动，飞机在天空中流动，药物在病人身体内流动；旅客在交通系统中流动；食物在不同国家之间流动；飓风在整个欧洲流动，等等。

在商业中，钱的流动可以证明是最重要的流。

但是在 21 世纪，世界上超过 90% 的钱是以数据的形式存在的。每天，以兆计的美元电子化的在世界上的 IT 系统中流动。尽管这样带来很多好处，但是也造成了很多的风险。

3.2 通用语言

很多世纪以来，体系架构师和工程师使用图片与图表来简化复杂性并降低风险。

如果没有看见、理解项目，没有对项目进行成本计算，没有与项目相关的计划图表和图片，在商业上从来不会给一个摩天大厦或者桥梁付钱，。

这些由体系架构师、工程师所使用的可视化工具，其目的是作为一种通用语言来提供服务，并帮助实现风险最小化。

它们通过将关于所有事情是怎么融合在一起来使得业务工作的机制清晰化，使得在一个运行的项目中相关的体系架构师、工程师以及其他人员能够沟通、合作、协作。

不同群体并拥有不同能力的人员都可以对这些文档进行讨论，并搞清楚它们都是如何对公司战略或者单独项目做出贡献的。

在今天由电子化驱动的复杂商业世界里，很明显对于清晰的、精确的了解业务如何运作的是

一个很迫切、必须的需求。必须有一个简单的、清楚的、逻辑的方式去理解和沟通业务与 IT 的关系以及它所关联的风险。

对于 2008 到 2009 年的金融危机的一个解释是金融机构以及金融规则并没有将钱-或者说数据,是如何在单独的交易中流动的,以及代表钱的数据是如何贯穿整个财务系统的没有被精确的、清晰的描述出来。

随着微型计算机在 20 世纪 70 年代的面世,计算机系统的费用随之下降。在不同部门的经理使用他们的预算去采购能够代替部门日常职能的系统,从而使得他们的部门更有效率。随着个人电脑在 20 世纪 80 年代成为统治地位,他们成为了商业运营的不可分割的一部分。

随着商业的变革以及技术的发展,很多新的应用系统被安装,同时为已有系统开发了许多接口。这些新的系统经常并不完全实现原来老的系统的所有功能-从而造成了老系统并没有完全下线。产生的结果即是遗留系统,以及因此造成的复杂性的增加。维护老的和新的技术使得企业资源干涸,使得安全的获得经济上的收益受到阻碍。

对于许多组织而言,预算控制和 IT 的采购现在整合在一起,成为 IT 服务交付中的责任。

尽管这样阻止了功能性解决方案在部门级被采纳,新的 IT 项目却可能在缺乏业务认同的情况下被买进,并可能被视为‘寻找问题的解决方案’。如果希望业务部门能够参与到交付新的目标,那么在 IT 与业务之间清晰的沟通至关重要。

今天,像金融、商业在大部分的领域都依赖与人、流程和技术间复杂的交互。数据的流是现代商业生存的血脉。

但是不像早些年代的商业,现代商业还没有精确的理解它们是如何运作的。它们不能够看清单个业务资产是如何整合在一起从而使得数据流动。

从 20 世纪 80 年代开始快速的采纳看上去似乎容易承受的信息技术,导致了大量系统的产生,并增加了复杂性。随着时间的推移,这些系统成为遗留系统。在大部分领域,策略制定经常让路与部门战术级的解决方案,从而解决短阶段的问题或者从熟悉的领域得到财政上的收获。但是,这种做法经常会产生意外。高风险领域,例如石油-天然气以及核能,被法律要求按照尽可能的安全的方式来运作,并且理解和管理危险的流对于企业的生存和业务一样关键。

那么,业务该如何向前发展以保证能够达到战略目标呢?

无论什么样的政治体系,什么样的经济系统,采取什么样的管理制度,以上讨论的问题都客观存在。流的理解对于理解业务如何运作,业务如何向前发展以保证能够达到战略目标都至关重要。

3.3 流测量的经济学

流测量的经济价值在一个依赖计算机的环境中已经被石油与天然气产业的运营者所证实。在这些领域的公司意识到对于所有支持业务功能的组件之间如何交互有详细的理解是优化流程输出的唯一方法。

BP 或者 SHELL 所拥有的精炼厂使得原油或者天然气的流流程化，并将它们转变成为不同产品的流，比如乙醇或者石油。

每一个制造元素（比如，管道，容器，泵、油罐）都被测量，流的压力，密度，温度值被测量出来，并被用来优化产出。这些监视流程也保证了工厂的运营是安全的、有效率的。在每个测量的流中经济价值是通过监控花费并最大化收益体现的。

作为结果，组件并不增加价值或者导致费用降低，一个业务功能可以被去除，重新分配或者被替换。

另外，由于任何组件的问题所导致的业务风险都可以被牢固的控制，从而使得该风险降低。这意味着当机时间会被减少，收益将得到保证。

由于现在对贯穿所有业务领域的信息技术有着根本的依赖，应用于原油和天然气工业组件的技术也能够被其他领域应用。这不但会降低今天复杂的 IT 基础架构和它们与业务的交互的花费，并将实现对他们更好的管理。

今天，通过管理和优化数据流，将使商业效率能够得到相应的改善。比如，一个没有有效率的 IT 的金融服务公司，它拥有一个对于运营流程没有价值的遗留系统，这将使它在当今价格敏感的金融领域失去竞争力。理解数据流将使得那些过时的、没有效率的系统下线，并被等待时间短以及实时的系统所代替。

3.4 IT 存在仅仅因为一个理由：支持数据的流动

大部分事情都可以按照一个流程术语来描述：-有一个开始、中间阶段以及结束。

无论一个人是在用餐、开车或者处理邮件或者 EMAIL-每件事情都会开始、经过一个流程、然后结束。

IT 工业视图也采用这种方法去思考 IT 系统。

复杂的任务可以被分解为子任务，每个子任务都会接受输入，进行计算，然后返回结果。

这个概念并不新，但是 IT 的趋势是聚焦于 IT 系统以及它们如何工作，而不是为什么 IT 系统被使用。这种内部化的视角可以容易理解，但是错失了改变 IT 被其客户（-业务）所感知的方式的机会。

后退一步看一下，IT 系统从来不是开始或者结束。所有的 IT 系统都仅仅是一个大的流程的中间一个步骤，IT 系统仅仅是一个业务引擎。

IT 存在的理由是在于其可以使得一个流程简化并效率更高，或者去驱动一个流程启动，而若其不存在，则该流程的启动就不会出现。

但是这不应该是 IT 的本质，其本质应该是十分重要的。**其本质应该是数据在整个系统中的流动，产生了对一个对变更更有意义的效果。**

IT 存在的唯一理由是驱动数据的流动。

一个原油和天然气公司建立他自己的 IT 体系结构有 20 年了，他失去了他的 IT 系统的工作程序、连接性以及对于业务的价值。

经理人使用 OBASHI 方法论来描绘其部分业务。他们发现了一个软件应用系统运行在一个过时的硬件上，该系统也没有在被其业务所使用。

每年该公司仍然要花费昂贵的费用来签订硬件以及软件的合同来支持这个遗留系统。OBASHI 项目的一个产出结果就是建议取消这些合同，让该软件下线，而该硬件则用于作为另一个系统的备份机，该系统刚好也使用这种过时的硬件。

这帮助该公司节省了大量金钱，减少合同的数量，同时提高了灾难恢复计划的能力。

考虑下面一个制造业环境的案例，管理信息系统和控制系统成为为了生成更好的产品或者获得更好的利润的链条中的关键连接点-即成为有效果变革的管道。如果 IT 在这个变更产出的有顺序的流程中不存在，它的角色就是无意义的。是数据的流动将这些单个的项目按照顺序连接在一起。

通过理解流的业务上下文，而不是仅仅是 IT 的上下文，流存在的原因就很明显了。OBASHI 让你看到数据从哪里源起（开始），数据流所支持的业务流程，以及哪个业务流程使用那个数据（结束）。你也可以看到用来驱动数据去流动的 IT 系统（中间步骤）。

3.5 数字化的流

通过使用业务与 IT 图表，OBASHI 获取了人、流程与技术之间的关系与依赖性。

业务与 IT 图表显示了在业务中的人是如何对业务流程负责-例如，相关利益人。他们也可以显示哪些人与业务流程进行交互，该交互是业务运营中的一部分。人是业务的驱动力量，并且在业务流程中所提供的信息能够允许有知识的工人对业务做出相应决策。

在业务流程之间存在着交互。描绘出业务流程如何依据它们的关系连接在一起，依据它们的相互依赖性进行运作，从而使得交互能够被量化与可视化。这对于那些打算实现流程之间高度分离的组织来说至关重要。比如，一个组织有一个法定的需求，即部门或者流程必须分离的运作以将知识放置在不同的业务线上进行传播。这样的公司包括：健康、国防、审计、市场、诉讼等行业。

今天，业务完全的依赖于信息技术，人员和业务流程依赖单独的和关联的 IT 组件来执行他们的商业任务。

信息在 IT 组件以及业务流程之间以数据流的形式流动，来支持人们的活动。

术语-数字化流被用来描绘数据流与数据相关的人员以及支持数据的信息化技术之间的依赖关系。OBASHI 方法论通过使用数据流来描绘数字化流。

数字动态性是对数字化流的学习，在图 3.5 中显示：

OBASHI 数据流分析视图是对所有支撑某个单独的数据流的业务和 IT 资源的图形化和统计化的表现形式。DAV 使得单个的数据流能够被分析和学习。

经济价值可以与组成一个数据流的组件相关联。

没有理解一个业务如何如何依赖数据流的，业务风险将不能被完整的评估出来。除非资产能够被放在数据如何在其支持的业务中流动的上下文中考虑，否则一个资产出现问题可能对业务所造成的风险就无法被量化。在安全领域也是一样，如果不能了解为什么一个数据在一个组织中流动，什么样的收益是通过业务流程使用这些数据而产生的，这些数据在正常的业务操作中是如何被访问和更改的，那么一个安全模型就不能得到完整的评估。

在整个 OBASHI 方法论的开发中，数字动态性已经成为最基本的原则。OBASHI 与数字动态性的关系（图 3.5）是这样的：

- OBASHI 方法论遵循数字动态性的原则；
- 业务与 IT 图表描绘了在数字动态性中所突出的人员、流程和计划之间的关系和依赖性；
- OBASHI 使用数据流来表现数字动态性的数字流；
- DAV 允许学习数字流（即数字动态性）；

数字动态性的原则在 Section B,第五章中详细描述。

OBASHI 的原则在 Section B,第七章中详细描述。

第四章 OBASHI 与业务改造

4.1 开发大图

组织可以使用 OBASHI 来支持获得公司的目标，并可以在任何阶段支持业务改造。

项目组合、项目组、项目和 IT 服务生命周期是管理业务变更的方法。图 4.1 显示了这些生命周期的每个阶段，它们是如何相互适应，它们在整个生命周期中如何与其他方法进行交互，这些内容将被本章剩余部分引用。

OBASHI 可以被这些生命周期所使用，帮助它们提高清晰性、连接性并确保业务变更计划的交付成果与变更策略保持一致。OBASHI 可以帮助决定交付的途径和关键性能指标 (KPI)，保证交付按照变更策略进行。

对于每一个生命周期阶段的详细解释分别记录在附录 A、附录 B、附录 C、附录 D。

可以使用 OBASHI 来形成和描绘战略，OBASHI 能够提供以下帮助：

- 发现组织在哪里有一个关于业务是如何被 IT 所支持的程度较低的理解，这是一个关键的输入，将生成一个或多个理解程度更高的业务与 IT 图表来显示业务当前是如何总体运营的。这可以识别出需要（克服）扩展的需求、空闲的能力或者关键的依赖关系；
- OBASHI 突出了战术和战略意图之间的冲突，公司经常尝试去实施战略和风险性决断来使得业务向前发展。然而，这个目标因为资源的干涸经常很难达到，干涸的原因是为了使得工作更简单而降低管理水平的行为会造成的战术支持和关键性操作决断越来越多。

请看图表 4.2

OBASHI 支持业务活动，比如定义组织的目标，视野，价值，提供 IT 服务业务的例子如下所描述：

4.1.1 组合生命周期-定义组织目标、视野和价值

组织决定它的长期目标以及业务策略，同时考虑它的战略和操作优先级。典型的，该流程将涉及到下面的几个角色：

- 首席执行官 (CEO)
- 首席财务官 (CFO)
- 首席信息官 (CIO)
- 领导，例如市场，销售，操作者，IT 人员；

➤ 战略的头

图 4.2 显示了如何通过执行收益实现管理来成功的管理组织变更，可以通过将业务目标分配到四个关键领域来实现这一目标。一个业务需要识别与它的策略相关联的：

- 进入新的全球市场或者维护当前的市场地位；
- 开发新的产品和服务；
- 进入新的客户群体；
- 降低运营费用；
- 通过管理将外部咨询的费用降低
- 等等

组织成熟度在决定以上要求是如何获得上是一个关键的影响因素，因为：

- 一些组织已经对于业务和 IT 是如何交互的有好的理解，而其他的却没有；
- 一些组织在组合管理、项目群管理、项目管理以及风险管理上拥有较高的成熟度，而其他的却没有；

4.1.2 IT 服务生命周期-IT 服务业务案例

IT 服务管理支持组织业务流程的 IT 服务组合。

IT 服务业务流程紧密的与业务战略集成，它能够被业务战略所决定，反过来也可以帮助业务战略的形成。

作为结果，当 IT 能够精确的计算 IT 花费以及包括对于现存服务和基础架构的潜在影响与风险时，业务就能够更好的评估新的机会，。

4.1.2.1 场景例子

图 4.3 提供了一个名为 CARCOP 的汽车制造商现在的组织形式的 OBASHI 大图的例子，为了扩展其产品组合该公司已经收购了一个名叫 FABRITE 的公司来降低其运营费用，改变 CACORP 业务表现不佳的领域，从而提高公司的市场位置。

FABRITE 拥有一系列的设计和制造工厂，从而制造原型并开发汽车组件。

如果能够实现成功的结合，该收购将使得 CARCOP 能够提高市场位置、扩展产品的组合，并能降低运营费用。

4.2 开发‘如果-则’场景和被认可的未來业务体系结构

通过使用‘如果-则’场景来生成一个业务未来将怎么运营的图示，OBASHI 可以用来发现优化的方案来交付公司目标。这可以通过以下方法的帮助下获得：

- 主要通过强大的可视化建模，OBASHI 可以获取现有的体系结构、它们的配置以及它们

与它们所支持的业务之间的联系。另外，可以突出组件重用和费用管理的机会，OBASHI也使得对于任何对现有基础架构的预期变更有一个更精确的花费与风险的评估。

- **OBASHI 业务与 IT 图表**记录了数据流，业务所有者和业务流程，并对使得例如渗透入新的市场，降低运营费用，向新的市场领域交付，降低咨询费用这样的行动所需要得到的 IT 支持进行了记录。
- 业务和 IT 图表可以帮助交付策略的制定，寻找可行性与可供选择的方法(如果-则场景)。可以使用一个未来状态的图表（未来业务的体系结构）来显示业务尝试在未来采用什么样的方式来运营。
- 业务和 IT 图表识别出可接受与可管理的业务组件重用机会，并提供了清晰的、端到端的、支持识别真正的业务关注的 KPI 的数据流；
- 业务与 IT 图表能够提供合适的决定支持。被认可的业务与 IT 大图可以作为一种输入来定义交付途径和 KPIs；
- 业务与 IT 图表以大图作业的一部分生成，并且‘如果-则’场景可以用来识别哪些任务可以作为一个逻辑整体属于某个项目或者某个项目群。它们也可以用来识别这些项目之间的接口与依赖关系；
- 业务与 IT 图表的库可以用来保证当进行决策制定时，业务知识是可用的；
- 业务与 IT 图表可以用来评估
 - 基础想法以及它们与战略的符合度，通过理解什么被交付以及它们如何支持新的业务体系结构的严重性；
 - 与交付相关的花费；
 - 在项目群进行变革中内在的风险；
 - 组织所拥有的管理以及交付项目群的能力；
 - 那些对于要对组合优化的流程进行输入的关键干系人；

OBASHI 支持业务变革活动比如对项目 and 项目群的识别、定义、描绘以及优化。它也支持在比较早的阶段设计业务的服务。这样的一些活动描述如下：

4.2.1 组合的生命周期-定义交付途径与 KPIs

在定义交付途径与 KPIs 的流程中，一个组织将尝试去决定如何获得视野和战略对象，认同交付的途径，以及通过测量哪些关键指标 KPI 来判断是否成功。典型的，这个流程中包括如下角色：

- 首席财务官或者其他财务专家；

- 战略的领导；
- 首席信息官；
- 组合管理专家；
- IT 体系架构师；
- 服务设计经理

4.2.1.1. 例子

图 4.4, 4.5, 4.6 提供了一个生成未来业务体系结构的例子。大图业务与 IT 图表 4.4, 4.5 显示了 FABRITE 和 CARCORP 当前的流程如何运作，来设计和交付新的部件。未来业务体系结构被生成（图 4.6）可以通过内包部分设计和制造的过程来运作。FABRITE 和 CARCORP 各自运营一个设计流程，在未来状态图表中这个状态将作为增效流程显示，这表明了他们在战略上尝试怎么去运作。

4.2.2 组合的生命周期-识别并界定项目群和项目的范围

在识别并界定项目组和项目流程里，组织将尝试去识别为了获取公司的目标需要哪些项目组或者项目，并界定他们的范围来理解接口和依赖性。这个过程典型的将涉及到一下这些角色：

- 组合管理专家；
- 战略的领导；
- 项目组和项目管理专家

4.2.1.1 场景例子

这个场景提供了一个项目群，该项目群为了降低费用而计划进行结构重组。FABRITE 的设计部门包括计划、工程设计、安全标准化以及采购过程，在图 4.8 中可以看到更详细的描述。

CARCORP 设计部门，在图 4.9 中显示的，只有工程设计和安全标准化功能，然而 CARCORP 公司在运输部门有一个计划功能，使用这个功能可以在设计部门提供计划功能。同时，CARCORP 有一个专门的采购部门，可以替代 FARBRITE 的采购功能。

这突出了关于一个重组业务的项目群，它们具备集成 FABRITE 工程设计和安全标准化的功能到 CARCORP 公司的设计部门的可能性，因此允许 FABRITE 计划和购买功能被 CARCORP 现有流程所替代。这降低了维护相同功能的费用。

4.2.3 组合的生命周期-确定组合的优先级，优化组合并确认组合

许多组织提供了一个优化的框架来为资深的管理者（比如公司管理委员会以及组合管理

委员会) 优化项目群与项目提供决策的支持。这么做是为了保证组合能够最好的符合战略的方向, 并降低投资的风险。

这个优化的框架将审查所有潜在的项目群和项目, 包括在运营环境中正在进行的, 来识别可以优化的组合。该方案也考虑相关的限制, 比如资金和资源的限制, 这个过程典型的将涉及到以下这些角色:

- 组合管理专家;
- 战略的领导;
- 项目组合和项目管理的专家;
- CEO 的决策;
- 首席信息官;
- 首席财务官;

4.2.4 组合的生命周期-交付并跟踪组合交付的进程

随着项目群和项目开始交付能力并开始实现收益, 未来状态的模型部分开始形成。在项目群交付的最后阶段, 组合管理办公室将努力去识别所选择的组合的交付是否吻合战略。

在这个阶段, 所有用来最大化已经获得的收益或者对新的机会最大化的利用的决定已经被识别出来。这个过程典型包括一下角色:

- 组合管理专家;
- 战略的领导

4.2.5 项目群的生命周期-项目群授权

随着组合被排优先级、优化以及确定, 组合将开始准备给项目群授权。这赋予了项目群需要获得结果的背景, 并赋予项目群所有的限制条件以及项目群如何适合公司战略所需要的指示。

合适的业务与 IT 图表可以作为项目群授权的一个部分提供给项目群经理和资深的负责人。项目群授权的目的是识别适合相关目标的战略, 识别交付所需要的新的服务/能力并识别所有的关键限制。

业务与 IT 图表可以作为该流程的一个输入。

4.2.6 项目群的生命周期-识别项目概要

项目群的目标、治理、计划和业务案例被定义来决定项目群是否值得、是否切实可行。

为达到这个目的, 关键关系人必须识别出来。

4.2.6.1 场景例子

图 4.10 和图 4.11 提供了一个 FARBRITE 和 CARCORP 公司的业务利益相关者可以很容易在业务与 IT 图中识别出来的例子。通过下面的橙色突出显示相关利益人。

4.2.7 IT 服务生命周期-设计服务

服务流程的设计决定 IT 服务解决方案可以在可接受的花费与质量属性的前提下来支持业务需求。它重用那些现存的、合适的、没有对业务运营产生不可接受的限制或者风险的服务以及基础架构组件。

在这个设计阶段，IT 服务管理者将识别服务质量的关键 KPIs。最终的目标是要识别出以业务为中心的测量指标而不是 IT 的。

4.3 开发蓝图

OBASHI 可以生成一个新的战略图来描绘业务在一个地点，部门，系统或者服务级别是什么样子的。通过这个功能，以及在资金与致命资源使用前就对于未来状态的可分析与可视化，该流程可以通过下面方式提供帮助：

- OBASHI 业务流程层可以识别需要变更的关键流程；
- OBASHI 所有者层识别了将操作新业务流程的人员，也可以用来识别技能层次和人员数量；
- OBASHI 应用层、系统层、硬件层和基础架构层记录了为了运作该业务模型而使用的技术与工具；
- DAV 的目的是去理解和记录对于理解业务的管理信息所需要的数据流；

该流程支持了业务活动，认为一个项目群的活动是否值得以及是否可行，比如定义：在项目群生命周期中的定义项目群文档的流程。

4.3.1 项目群生命周期-定义项目群定义性文档

项目群的目标、治理、计划和业务案例被细化，结束项目群定义的干系人被识别。

当一个项目群被定义，一个业务蓝图被开发出来显示未来的操作。这包括了将当前业务的视图和‘被认可的未來业务体系结构’进入更详细的下一级别，来提高对于一个特定的部门应该如何运作的理解。

4.3.1.1 场景的例子

图 4.12 给出了一个业务与 IT 图表的例子，它提供了一个更高级别的细节。业务与 IT 图表给了一个对于运输部门如何运作计划流程的特别的理解。

业务与 IT 图表现在显示了支持计划流程的应用以及 IT 资源，包括它们在 CARCORP 网络体系结构中是如何相互连接的。

现在可以清晰的看到运输 2 部应用包括两个软件组件：日志（Jounral）与移动（Movements）。

4.4 开发详细的图表

OBASHI 在项目级帮助业务战略解决方案实现工程化，它通过以下方法来帮助这个目标的实现：

- 随着项目群的进展，项目着手开始为变更了的部门操作开发更加详细的进程。可以生成更详细的业务与 IT 图表来显示运营将如何工作。
- 大图-业务与 IT 图表通过变更来反映当前运营状态以及它们在现实中怎么运作。这可以很好的显示项目群与项目的新机会，甚至由于证据被收集显示组织是如何的收益于战略的部分实施，从而导致组织战略的变更；
- **OBASHI** 提供了关于当前的和计划实施的服务与 IT 基础架构的映射的详细知识；这对于计划、构建和彻底的测试至关重要，并能够实现在变更进入实际环境前就进行评估；这可以让客户相信服务将以被认可的级别得到支持并且不会对现有服务产生一个不可预测的效果；
- 业务与 IT 视图提供的信息还可以帮助选择一个合适的部署方案；
- 所有的业务与 IT 图表的拷贝将被作为成熟的业务知识基础存储起来，从而保证对于交付过程的审计线索能够被保存。这些更改了的图表将作为一个关键输入被组合管理使用，来跟踪 KPI 的变化以及新业务体系结构的交付。根据业务与 IT 图表可以为资深的管理者生成报告，从而显示交付策略是如何演变的。

OBASHI 支持这样的一些业务活动，这些活动指定了业务工作方法的变更，或者指定了支持业务工作的资产和资源的变更；这些活动的例子如下：

4.4.1 项目群的生命周期-管理阶段

项目群按照认可的治理被控制和监控。项目群和审核在每个阶段的结束时进行，从而对下一阶段实现承诺。

4.4.2 项目群的生命周期-交付能力

交付能力管理单独项目的启动、控制和结束，这些项目被视为是一个项目的组成部分。那些不能够作为项目一部分进行管理的风险和项目将在该流程进行管理，从而保证项目群的交付并不是妥协的产物。

4.4.3 项目群的生命周期-实现收益

该流程通过将项目输出嵌入到 BAU 中保证了项目输出能够得到收益。它测量了收益，

同时保证了在新的工作实践中运营能够继续工作，并解决已经产生的所有风险和问题。

4.4.4 项目的生命周期-单个项目的交付

更详细的业务与 IT 图表显示了在项目开始前的业务流程，在项目结束后的有计划的状态的图片，显示经过项目的改变后所产生的变更。它们也可以用来描绘被提议的变更有可能造成的影响并帮助对被提议的变更的决策支持。

项目的五个主要阶段一般被分为：

开始一个项目：建立项目管理团队并收集信息来指派项目委员会，并授权项目启动。

拥有当前状态（BAU）的业务与 IT 图表可以帮助该活动。项目途径也可以被建议采纳的方案的业务与 IT 图表所支持，它也会揭示所关联的风险。

初始化一个项目：在项目委员会授权项目前计划项目、完成一个完整的业务案例、并设置项目控制。业务与 IT 图可以给计划项目很大的帮助，帮助理解变更和所需要的新的增加物，并可以增加项目移交所需要的支持。同时这些图表可以作为那些需要被通知的和需要培训的干系人的辅助沟通的工具。

控制阶段：授权并控制所需要的工作包来完成该阶段的产品。监控并控制风险和问题来保证该阶段仍然处于项目委员会所能接受的范围内。现在和被建议的业务体系结构的业务与 IT 图表可以揭示先后次序以及所有的项目移交点。

管理阶段的边界：计划下一个阶段；根据已经完成的阶段与下一个阶段的详细信息来调整项目计划。检查业务案例是否仍然可行；审核质量、风险和问题的状态。业务和 IT 图表可以被调整来显示哪些已经完成以及所有需要交付的产品。应该仔细审核图表来发现风险的状态以及新的风险。

关闭一个项目：在项目计划中写下结束的记号。结束收益回顾计划；写下项目结束报告，教训分析报告，所有后续行动的改进建议。应该检查业务与 IT 图表来保证所有的产品都可用以及全体被影响的项目所有者都意识到即将到来的向未来状态的交割。

4.4.5 IT 服务生命周期-构建、测试和部署

在一个项目的生命周期中都有构建、测试和部署流程。它提供了所有需要的资源以及输入，来保证当 IT 服务被引入时，它在文档化、接受的和授权的层次都是可以支持与可测量的。建立、测试和部署阶段也可以确保 IT 服务不会对现有服务造成任何负面的和不可预测的影响。

4.4.5.1 例子

图 4.13 给了一个项目计划如何被业务与 IT 图表所支持，来理解所需要的变更与新的附

加物，增加转换所需要的知识。

更详细的作为项目实施的业务与 IT 图表被开发出来，并被界定范围。在图 4.13 当项目已经被进一步细化，网络基础架构被相应修改。

在项目设计阶段，可以标记出一个可能的机会，该机会可以利用 FABRITE 的网络体系结构来替代 CARCORP 的部分老的基础架构。这个新的技术方案被显示出来，表明它应该被作为项目实施的一部分进行部署。

相同的硬件、系统、应用系统仍然按照原始的设计使用，这符合公司关于使用 CARCORP 可以使用的 license 的战略。

4.5 运营和治理

OBASHI 图表可以阐述运营的稳定状态，如 BAU 视图。OBASHI 可以提供一个完整被开发的和成熟的关于人、流程、技术的业务知识基础， 这些可以通过下面这些方法得到帮助：

- 业务与 IT 图表被作为变革项目的一部分被生成，来反映正在运行的环境，并作为一个关键输入来影响目标，视野和组织的价值；
- 业务和 IT 图表使得运营与治理的收益能够获得，包括：快速的利益相关者识别、转换计划、配置管理、风险评估、在运营中识别威胁和机会、巩固机会、SLA 审核、有效果的变更管理、服务连续性、以及知识管理；
- 通过显示采用业务新工作方法而获得的收益和它们是如何支持组织交付公司目标的，业务与 IT 图表可以支持将项目和项目群移交到普通的运营中；
- OBASHI 业务与 IT 图表和 DAV 图表识别组件、服务和流程之间的依赖性。通过将偶然事件放入到会被影响的业务流程中，可以按照这些流程多么重要对偶然事件进行评估， 资源宕机对业务造成影响的价值；以及其对于其他 IT 资源和业务流程的‘连锁反映’。根据这个评估，可以对于偶然事件造成影响的支持响应进行优先级排序。OBASHI 给一个 IT 服务台事件提供了一个业务视图。

OBASHI 支持业务活动，实现、管理和管控业务，使得业务能够以一个不断前进的方式运营，如下面所描述：

4.5.1 组合的生命周期-BAU

一旦关键的项目群被交付，被描述在业务与 IT 图表（被认可的未來业务和体系结构的状态）所描绘的未来状态现在成为了 BAU，至少是部分。

新的运营状态形成了基线，并反馈作用于组织新的目标、视野、来年的价值和持

续发展的组合。

4.5.2 项目群生命周期-关闭项目群、交付项目群支持

项目群被关闭了，交付所期望收益的职责将移交到运营团队。这需要资深的业务经理持续的支持来确定收益不会被丢失。

一个业务变更团队将保证对于业务运作合适的支持是可以负担的，并保证新的变更已经真正的嵌入并且运营不会重新回到老的工作方式。

4.5.3 IT 服务生命周期-管理、测量、改善

在部署，管理、测量后，持续改善流程将根据业务影响度评估和响应服务事件，从而确保交付能够达到认可的服务目标，。

OBASHI 数据是配置知识的关键部分，允许 IT 支持团队能够有效果的识别、评估和响应服务故障。

这些详细的信息以及对于 IT 系统和它们所支持的业务流程之间关系的理解，使得管理、测量、改善流程能够去识别、评估持续服务改善的机会。

Section B OBASHI 方法论

这个章节提供了对于 OBASHI 方法论如何工作的理解。它描述了 OBASHI 组件以及它们如何组合在一起以及解释怎么去生成、解释和分析业务与 IT 图表。

第五章 -介绍 OBASHI 核心原则的原则

OBASHI 核心原则 (Principles) -描绘了 OBASHI 方法论的基础原则。

OBASHI 的法则 (Law) -控制 OBASHI 方法论和运作的法则。

数字动态性-研究数据在人、流程和技术之间的流动与传播。

业务和 IT 图表-图表化的表现形式,来表现在组织内的人员、IT 资产和资源以及它们所支持的业务流程和服务之间逻辑和物理的联系 (连接)。

数据流分析视图 (DAV) -一个图形化统计化的文档来阐述元素在一个或者多个业务与 IT 图表中的顺序。

第六章-元素和层

元素和层-通过将元素放置在六个水平的层中来描述组织。

第七章-OBASHI 关系规则 (Rules)

关系类型-描述了存在与元素之间的六种关系, 以及控制每一种关系的规则集。

规则-控制元素之间隐性和显性的关系。

第八章-技术和标准

对技术与标准建模-解释业务与 IT 图表和它们所描述的数据流, OBASHI 关系是如何被使用的;建模中的关键概念;可视化标准;OBASHI 如何被用来补充国际标准组织(ISO)和英国标准 (BSI)。

第五章 介绍原则

5.1 OBASHI 核心原则

OBASHI 核心原则勾勒了 OBASHI 方法论作为基础的原则。

OBASHI 核心原则是源自 1990 年代后期，从英国原油和天然气公司所进行的工作中研究出来。

该原则依据特定的工程经验中开发出来，这些实践包括开发、管理和分析人员、流程、对设备的操作、控制技术之间的相互依赖关系，以及它们与 IT 之间关系的关键依赖性。

英国原油和天然气公司很快就意识到计算能力将为业务带来优化的财务回报。但是这种在生产率上的提高也会带来风险的提高。相应的，高风险的系统的完整性，比如‘安全’，在对于系统运行所需要的 license 被授权前，变成必须的并需要健康和执行委员会的审批才能运作。

对于所有高风险系统的运作至关重要是数据的流动；

尽管为了在原油和天然气工业达到安全以及财政最优化而采取的严格的工程途径提供了形成 OBASHI 核心原则的模版，它们同时也适应于所有的工业领域。

- 1、对于数据的流动的理解是组织财务健康的基础；
- 2、业务资源和 IT 资产既是数据的提供者、消费者，也为数据流动提供了管道；
- 3、IT 的存在仅仅是为了一个理由，即使得数据在业务资产间流动；
- 4、除非一个数据流动中断的原因和所产生的影响，以及在该数据流中数据的变化能够在数据流动的上下文中被评估出来，否则不能够对业务风险实现完全的定性、定量；
- 5、除非一个数据流动中断的原因和所产生的影响，以及在该数据流中数据的变化能够在数据流动的上下文中被评估出来，一个数据安全模型不能够被完整的评估。

5.2 OBASHI 的法则

OBASHI 方法控制 OBASHI 方法论的建模和运作。这一点通过它们在 OBASHI 上下文中的内容中的作用得到阐明：

- 1、一个物质的或者非物质的元素可以代表任何业务资源或者资产；
- 2、一个元素只能放在它自己的 OBASHI 层，并且其大小不能超出它所在层的范围；

- 3、一个元素可以和其他任何元素有关系；
- 4、任何数据类型或者数据的分类都可以成为一个元素的属性；
- 5、元素间可以采用 6 个联系类型的一个或者多个进行联系；
- 6、六种关系是：连接、依赖、空间、集合、层和序列的；
- 7、关系类型符合 OBASHI 联系原则；
- 8、OBASHI 方法论符合数字动态性的法则；
- 9、任何数据类型、分类都可以成为一个数字流数据的属性。

5.3 数字动态性

数字动态性是对数据在人员、流程和技术之间的传播和流动进行研究。数字动力学发展的催化剂是由于对于设计、工程、运营、控制业务依赖于电子数据的流动这一事实的基础原则需要深入了解。

历史表明如果一个专业的工程能够采纳管理和优化流的方式（水、电、天然气和制造产品，汽车，列车，工厂等），那么就可以实现持续的改善来使得业务工作更加有效。

今天，通过管理和优化数据的流动，现代商业可以实现业务效率的相应改善。业务流程与业务服务的自动化使得一个组织依赖于数字流动的电子化。但是，对于数字流动的理解、测量和评估并没有被作为一个核心的业务功能得到承认。

数字动力学建立了现代业务的基本原理，为专业化工程提供了一个战略的业务解决方案。

术语‘数字化流’用于来刻画数据的流动、与该数据相干的人员、支撑该数据的信息技术之间的依赖关系。

5.4 数字动态性的法则

OBASHI 遵循数字动态性的法则。数字动态性有 5 个法则，如下所示：

- 1、如果数据要存在，那么必须有数据的流动产生；
- 2、数字流有两个或者多个节点；
- 3、一个数字流由一个或多个数字流组成；
- 4、数据流动或者传播的中断会产生一个效果；
- 5、一个数字流存在的可测量的价值必须由组成该数字流的每一个节点的价值累计。

5.5 业务和 IT 图表

OBASHI 方法论用来生成业务和 IT 图表。

业务和 IT 图表（图 5.5）是在一个组织中的人员、IT 资产和资源、业务流程以及它们所支持的服务之间的物理和逻辑关系的图形化表现方式。

业务与 IT 图表由放置在各层的元素组成。单个元素可以代表一个人员、一个地方、软件或者硬件设备的一个部分。在图 5.5 上，下面这些全部都是元素：产品经理、订单、商店、MS office、bull AIX、CALL O3、路由器-cisco 7206，VXR-A 公司。对于和元素有关系的数据的类型并没有任何的限制。所有的业务资产或者资源都可以在业务与 IT 图表中显示。

业务与 IT 图表可以精确的以一个容易理解的可视化格式刻画业务流程与支持它们的 IT 资产和资源以及数据的数字流之间的所有者关系、复杂的交互关系、依赖性。

一旦元素被加入到业务与 IT 图表，元素与元素间的关系库就被建立起来。尽管使用支持 OBASHI 的软件工具可以以电子化的方式管理该库，使用手工方式（图 5.6）或者蜡质模版来勾划 IT 与图表也是可以的。

然而，通过使用一个图形化的接口（如图 5.7）或者软件工具（图 6.1）来生成业务与 IT 图表，可以生成一个关于这些关系的交互性模型，并为分析提供更多的机会。

通过用手记录或者将它采用电子化的方式存入一个库内，任何类型的数据和信息都可以成为一个元素的属性。这些数据可以手工的或者自动的从数据所在的在线系统中获得。这些信息可以在其上下文环境中被浏览、操作以及分析。

当使用一个软件工具来生成图表，应该使用可视化标准来保证所有业务与 IT 图表的一致性。在 OBASHI 中有可视化标准来涵盖连接器与依赖关系是如何显示的，并有缺省的颜色来画元素。（参看第八章 OBASHI 涂色标准）。当采用手工方式来勾画业务与 IT 图表时，颜色标准有可能不能被应用，然而可以通过手工在业务与 IT 图上来增加其他的信息来增加清晰性。OBASHI 是一个由业务与 IT 图表组成的‘模型’。业务与 IT 图表也不应该被看成是一个‘仅仅是一个呆板的图片’。组成业务与 IT 图表的元素与连接可以被当作业务的不同面貌，反映出核心或者非核心的信息。

OBASHI 业务与 IT 图表勾画出了业务或者组织的 DNA。它们是重要的业务-决定性文档并且也应该被这么对待。应该有一个合适的授权流程来对生成、审核、编辑以及删除业务与 IT 图表进行管理。

OBASHI 在下面三个层面：

- **一个清晰的整体图**-OBASHI 业务与 IT 图表使得使用者可以来创建一个清晰的容易理解的业务模型；
- **OBASHI 业务知识**-一旦一个 OBASHI 模型被创建，所有的信息（花费、价值、支持、人力、所有需要和这些资源产生关联的对象）可以被关联到每一个 OBASHI 元素，并可以被报告。一个纸质的或者电子化的 OBASHI 库可以保留业务知识。

➤ **一个动态的模型**-每个单独的 OBASHI 业务与 IT 图表可以被认为是一个模型,并允许下面的活动被执行:

- 干系人分析;
- 组合管理;
- ‘如果-则’分析;
- 业务案例开发;
- 财务分析;
- 风险分析;
- 安全性分析

一旦更多的单个的业务与 IT 图表被创建出来,它们所获取的知识可以被认为是一个大的‘动态’的模型。随着动态模型在更多的业务与 IT 图表中拓展,为每一个元素能够提供更多的上下文信息,甚至提供更好的决策依据。

5.6 数据流分析视图

数据在业务资产中的流动是每一个现代组织的血液。是数据流将组成现代业务的人员、流程和技术连接起来。

OBASHI 的框架使得建模、分析、评估和报告所有的支撑业务的流程和服务的数据流成为可能。

整体来看,一个业务流程或者服务可以包括一个由不同类型的资产和资源的组合,包括设备、产品、人力、合同、政策、安全、健康和安全、软件和硬件的基础架构。

因为所有的这些项目都可以以元素或者与元素关联的方式表现出来,OBASHI 提供了一个机制来描绘数据如何在这些资产和资源之间流动。这种从一个元素到另外一个元素顺序的流动,允许了业务流程或者服务可以运作。

OBASHI 数据流分析视图是一个图形化和分析化的文档,阐述了元素的一个子集以预先定义的顺序,在一个或多个业务与 IT 图表将处于什么样的状态。

业务与 IT 图表提供了 DAVS 的信息源头。

一个 DAV 图表对支持一个数据的流动的两个或者多个节点之间的顺序关系进行建模。这个数据的流动的术语是‘数据流’。

一旦被建模,数据流为一个特定的数据流动所包含的所有元素提供了一个清晰的图片。

DAV 图表可以跨越多个业务与 IT 图表。DAVs 也可以用来讨论、审计预期目标,或者进行影响度评估和计划。

在任何业务与 IT 图表中的一个单独的元素都可以被分析，来审核穿过它的数据流。可以做出支持该元素的价值或者花费属性的报表。组成数据流的元素的数量以及步骤可以被计算出来。

所有数据流存在所需要的物理的和逻辑的资产以序列的方式被记录下来。

一个 DAV 将元素的价值和统计属性分隔开，这样可以使得一个特定的数据流对于业务的贡献可以被分析、沟通和理解。

关于被问起的复杂问题，例如花费和价值，贡献度，影响度，风险，可用性（1），脆弱性、未符合战略的领域以及可巩固的领域可以以一种更高的确定性被回答。

相应的，通过价值属性的不断变化可以从原始的 DAV 中生成更多的 DAVs，从而模拟和优化业务流程，业务服务，资源，资产和生产率成为可能。

- 1、可执行性（performability）与可用性（availability）不一样在于，可用性考虑的是系统可以运作（正在运作），而可执行性考虑的是当一个任务要求这么做时，一个系统是可以执行的(可以运作)。

举个例子，设想一个系统一个星期有一天不在线，那么它就有 80%的可用性，不可用性为 20%。而从可执行型的角度来看，如果一个系统仅仅只在周一工作而在周五当机，那么它的可执行是 100%，当机时间并不影响系统的可执行性。

可用性回答了系统什么时候工作的问题，而可执行性回答了如果需要执行时系统是否可执行。将系统按照可执行性的角度进行描述则必须了解到一个资产什么时候和如何被业务使用，并可参看这些信息来设计新系统的原则，并对一个项目的范围进行界定。

第六章 –元素和层

元素是业务与 IT 图表的关键组成模块。

每一个元素都代表了在组织中的一个资产或者资源-比如硬件的一个部分、一个人员或者一个应用等等。

一个元素以一个简单的矩形描述。它的长度可以根据它所支持以及支持它的元素的关系的长度来调整。因此，一个元素可能被描述成一个正方形。

一旦一个元素被创造并被放置到一个特定的层，那么它就被隐性的和该层联系起来。一个元素可以在自己所在的层里面移动。

有 6 个不同类型的元素，每一个对应于一个 OBASHI 层：

- 所有者元素
- 业务流程元素
- 应用元素
- 硬件元素
- 基础架构元素

元素只能放置在它们自己的 OBASHI 层。比如，一个所有者元素不能够放置在系统层。一个硬件元素不可以放置在业务流程层。一个元素的大小不能够超过它所在层的大小。

元素放置到一个业务与 IT 图表时要起一个名字。在开始一个 OBASHI 项目前，开发一个对于所有元素取名的约定是一个好的实践。

同一个元素可以在一个业务与 IT 图表中有多个实例。为什么这个建模技术能够应用在创建 IT 与业务图表有很多原因。

经常的，重复的将一个相同的元素放在一个业务与 IT 图表上可以增加清晰度，同时能够简化连接的路由路线。另一个原因是，它可以用来提高一个业务与 IT 图表的视觉效果，或者通过增加额外的关系来为图表增加更多的上下文信息。这些技术将在第八章中的‘关系持久性’和‘逻辑元素’章节中进行更详细的描述。

在同一个图表中使用同一个元素的多个版本如图 6.2 所示：

该业务与 IT 图表显示了 IT 资产的深层显示，在同一个业务与 IT 图表上对一个现存的 IT 体系架构和建议的 IT 体系架构进行建模。那些被保存和重用的现存元素在建议的体系结

构中被显示了两次。

任何时候一个元素在一个业务与 IT 图表中被使用了两次或者多次，就必须常见一个唯一的索引用来对该元素的每个实例进行引用。

通过突出单个元素或者元素的集合，元素的颜色能够帮助沟通变得更清晰。

每个元素类型都拥有自己的标准颜色（如第八章 中 OBASHI 颜色标准章节所示）。如果在一个元素上使用的不是标准颜色，该图表**必须**有对于使用该颜色的解释。

比如，图 6.3 以及图 6.4 上有一些上色了的元素，这些元素代表了由一个特定的第三方所提供或者支持的资源，从而为这种供应商关系提供了更清晰的视图。

元素应该在它们所在的层中清晰的排列，无论是垂直方向还是水平方向，以符合 OBASHI 的空间法则。

元素可以分层的排列或者按照它们的空间关系进行放置（请参见第八章中所描述的空间关系、定义层次关系元素、定义逻辑元素和逻辑或者物理建模）。

在其他的方法论中（如 ITIL、MSP 和 PRINCE2）元素被定义为配置项、工件或者实体。

6.2 OBASHI 层

OBASHI 方法论对一个组织按照六层进行建模。

这些层提供了一个框架，用来组织单个代表业务、IT 资产或者资源的元素。

这些层分别是：

- 所有者
- 业务流程
- 应用
- 系统
- 硬件
- 基础架构

将这些元素在框架中放在其他元素的上面或者下面显示了元素间的关系。

比如，将一个所有者元素放在一个业务流程元素的上面表明了该业务流程属于该所有者。

放置一个业务流程元素放在一个应用元素的上面表明了该业务流程使用该应用。

在图 6.6 中 ‘SYSTEM’ 元素的位置表明了它被所有者元素 ‘owner 1’ 所使用。

在一个业务与 IT 图表中元素可以连接起来表明一个物理的联系，比如在一个硬件元素和基础架构元素之间的连接。在第七章的连接关系的一节中对此进行了详细描述。

在图表中可以记录依赖关系来显示元素间明确的非显而易见的关系，比如业务流程对于

一个第三方资源的依赖性。在第七章的依赖关系一节中对此进行了详细描述。

每一个元素都可以被支持性的文档所引用，从而提供一个对于该元素的支持性上下文。

数据的流动（数据流）可以叠加在一个业务与 IT 图表上来描述支持一个业务服务所需要的元素的顺序。在第五章的数据流分析视图一节中对此进行了详细描述。

使用一个或者组合多个业务与 IT 图表可以组成一个分析人员、流程和技术是如何支持一个业务的上下文模型。在第八章的关系持久性一节中对此进行了详细描述。

6.2.1 所有者层

所有者层拥有代表人员、角色或者群体的元素，他们‘拥有’，使用或者对一个在业务流程层中的业务流程负责。

在 OBASHI 框架中这些元素放在它们拥有或者使用的业务流程上面。

对于所有者有三个分类：

- 1、**个人**：一个人员或者工作头衔；
- 2、**地点**：一个特定的地点，比如一个办公室/建筑，或者一个地点的名称；
- 3、**组织单元**：一个部门的名称或者预算实体。

设计在 OBASHI 中的所有者层是为了允许业务流程的利益相关者的信息被显示出来。

将一个地点的或者组织单元的元素放在一个个人的元素上面就生成了一个所有者的层次，从而就提供了关于所有者的附加的上下文信息。一个关于这种上下文信息的例子可以是一个组织结构图，它的描述显示了一个关于部门和部门地点的结构报告。

图 6.7 描述了一些下面关于所有者层元素层次化的例子：

个人：John Dow; Jane Smith, 计划经理、发票管理员；

地点：计算机房、数据中心 1、伦敦、悉尼主大道；

组织单元：会计、运输部门、CAPEX 451; OPEX 192.43; Budget WL345.3214

6.2.2 业务流程层

业务流程层拥有的元素代表着被所有者所使用的业务流程或者业务功能。在 OBASHI 框架中它们被放置在合适的所有者下面。这可以在图 6.8 中看到，业务流‘Sales Transactions’和‘Address Book’显示被所有者‘Logistics’所使用。

术语‘业务功能’或者‘业务流程’被业务分析所使用来描述一系列逻辑的、相关的活动，当被一起运行使它们将获得一个已定义的结果。

一个高层的业务流程经常可以被分解成相对小的流程。这种分解可以一直分解，直到大的业务流程被分解成为最小的任务，这些任务如果运行则可以生成那个大的业务流程

的可定义的结果。

当分解成为构成该任务的子任务，这些子任务组成了业务流程的工作流。

在 OBASHI 中的一个业务流程元素可以用来刻画任何大小的业务流程，从复杂的高层的流程到执行该工作流的单个的任务。

业务流程元素能够被放置在其他业务流程元素的上面或者下面来形成一个业务流程的层次化分解。这使得 OBASHI 可以对组成更大业务流程的单个业务活动进行建模。

OBASHI 可以通过子流程对业务流程进行清楚的定义，使得工作流定义非常清晰。

OBASHI 元素可以在多个图表中显示。这使得可以实现高层业务与 IT 图表的创建，这些图表可以显示业务流程是如何被这些业务所使用的。也正是这个能力使得创建关于一个业务从宏观尺度是如何运行的‘大图’成为可能。

更详细的对于单个流程的定义可以描绘在分离的业务与 IT 图表中，来显示一个高层的流程如何被分解成更小的流程。这个分解过程可以反复的进行，直到一个流程的操作以所期望的详细级别所显示。

就是这种在 OBASHI 项目生命周期中从流程中不断往下获取详细信息的方法，创建了详细的图表来支持项目的范围界定与交付。(见第四章 中 细化开发详细图表节)。

在 OBASHI 框架中，元素可以被分解，从而显示更详细的信息。通过将附加的元素放在其他元素之上或者之下。这些附加的元素提供了元素的更加粒状的层次细节，从而使得该元素成为一个更高层次的元素。

在图 6.9 中，会计流程被分解为结账和工资流程。

图 6.10 接着进行分解，将结账流程分解为收账和生成账单的业务流程，而工资流程被分解为银行业务和缴税业务。

将会计元素的大小变宽以使得它可以在其下面容纳更多的元素。变宽这样的行为成为一个该流程的定义有多宽的视觉信号。会计流程被定位成一个比定义在它下面的流程‘更高层’的流程。

总体而言，一个元素的层次越高，它在图表上显示其定义所拥有的和宽度就更宽。

反之，对于‘较低层’的流程亦然。它们可以被认为有一个流程定义了代表更窄的内容，并在业务与 IT 图表中有一个更窄的显示。

将一个业务流程元素的定义窄化提供了一个建模的额外粒度。OBASHI 建模提供的粒度越深，它所提供的上下文信息就越精确，也就是说随着业务和 IT 图表变得更详细那么从该模型中就可以获得更多的上下文信息。

业务流程元素的例子可以是：月度结算、销售交易、邮箱存储管理、产品数据、发票扫描并将扫描件电子邮件给发票管理员。

6.2.3 应用层

应用层拥有的元素代表了软件应用程序或者软件应用程序的被定义部分。在 OBASHI 框架中，应用程序元素被放置在使用它们的业务流程下面。可以参见图 6.11，应用 ‘ORACLE Financials’ 被放置在使用它的业务流程 ‘Monthly Balance’ 和 ‘Ledger’ 下面。

一个应用元素可以用来表示一个应用程序的集合或者一套应用程序。也可以用来表示一个较大的应用程序的部分，比如一个应用程序模块或者程序功能。

当在 OBASHI 中建模时，可以显示一个组成一个应用程序套件的单独的应用。进一步的，该单独的应用可以被分解来显示软件功能或者组成该应用的组件。

采用这种分解大的应用的方式可以使得相关利益者来识别应用的每个部分，在 OBASHI 模型中提供更多的业务上下文。

应用程序层创建了一个应用的组合，它可以用来实现对 license 的管理。应用的例子可以包括：Excel、Oracle、Sage、SAP General Ledger（总账）和 PeopleSoft（仁科）。

6.2.4 系统层

系统层包括的元素代表了操作系统。在 OBASHI 框架中，它们被放置在它们所安装的硬件之上而在它们上面安装的应用之下。

在操作系统上运行的应用系统被放置在合适的系统元素之上。这可以通过图 6.12 看到，应用系统 ‘SAP General Ledger’ 被显示在使用操作系统 ‘MVS’。

通过将操作系统放置在它们自己的层上面，在操作系统和应用之间确立了一个明确的区分。这使得对于操作系统的部署和授权分析成为可能。

系统层创建了一个计算机操作系统的登记机制，它可以用来实现对 license 的管理。

例子可以包括：Windows XP, Unix, Windows Server, Solaris, Linux 和 Vista。

6.2.5 硬件层

硬件层包括的元素代表了操作系统在上运行的计算机硬件环境。在 OBASHI 框架中，它们放置在那些操作系统下面。这可以在图 6.13 中看到，操作系统 ‘Unix’ 显示被硬件 ‘Server-OPEX’ 使用。

该层也可以包括那些非-网络设备，比如打印机。

硬件层创建了一个资产记录，里面的项目是硬件设备。

一个硬件元素可以用来代表一个单独的硬件，也可以代表一个硬件的集合，这个集合组合在一起成为一个新的硬件。一个例子是刀片服务器拥有多个刀片，或者处理器卡片。每个刀片都可以以一个硬件元素表现出来，但是 OBASHI 提供了一个机制来创建一个附加的硬件元素，刀片们可以排列在该元素上面。这在 OBASHI 建模上给出了一个提示：所有的刀片都放置在一个单独的基座上面，每个刀片都是冗余的。

硬件元素的例子可以是工作站、服务器、笔记本电脑、桌面个人电脑、集群、网格、虚拟机、打印机或者主机系统。

6.2.6 基础架构层

基础架构层包括的元素代表了网络基础架构。在 OBASHI 框架中基础架构元素被放置在使用它们的硬件元素的下面。基础架构元素也可以放置在其他基础架构元素的下面来创建一个支持业务的层次。这个可以在图 6.14 中看见，硬件元素 ‘Mainframe Cluster’ 显示使用基础架构 ‘Bridge4-Live2’。OBASHI 方法论的灵活性保证了可以满足任何网络设计（拓扑）的需要。弹性的网络体系结构可以通过使用逻辑元素来显示，（请参看第八章的定义逻辑元素节），同时不同类型的网络布线可以通过 OBASHI 连接器的连接来显示。（参看第七章连接关系节）

基础架构层创建了一个网络设备的资产登记。

将基础架构的层次以图形化展现提供了在系统里面的网络设备的数量的视觉提示。

当使用在数据流分析视图（DAVs）时，网络延迟的信息可以在基础架构层进行叠加，提供了一个分析网络时间的手段。（参看第八章 OBASHI 数据流分析视图获得更多的信息）。

在业务与 IT 图表中，图表的设计者通过对基础架构元素上色可以帮助 SLA 管理和业务风险管理（如图 6.3）。网络安全条目和对于潜在的单点失败也可以实现可视化，并被分析和沟通。

基础架构元素的例子可以是：交换机、路由器、复用器和 HUB。在 OBASHI 方法论中，无线网络设备和 modem 也被分类为基础架构，尽管它们直接的联系到或集成到象 PC 和服务器这样的硬件设备中。

第七章 –OBASHI 关系规则

7.1 OBASHI 关系规则

OBASHI 方法论包括以下规则来控制元素间显式的和隐式的关系。

- 1、一个元素放在另一个元素的上面或者下面都表明它与那个元素有一个隐式的关系；
- 2、所有在同一层的元素都与其他元素有一个隐式的关系；
- 3、相互连接的元素之间都有一个显式的关系，下面是控制连接的规则：
 - i. 一个连接是一个双向的关系；
 - ii. 一个基础架构元素可以连接到一个或者多个基础架构或者硬件元素；
 - iii. 一个硬件元素可以连接到一个或者多个硬件、基础架构或者系统元素；
 - iv. 一个系统元素可以连接到一个或者多个硬件、系统或者应用元素；
 - v. 一个应用元素可以连接到一个或者多个系统、应用或者业务流程元素；
 - vi. 一个业务流程元素可以连接到一个或者多个应用、业务流程或者所有者元素；
 - vii. 一个所有者元素可以连接到一个或者多个业务流程或者所有者元素。
- 4、依赖关系是一个单向关系，例如一个元素 X 可以依赖元素 Y，但是元素 Y 可以不依赖元素 X；
- 5、一个元素在一个层中可以有一个或者多个实例；
- 6、一个元素可以在 OBASHI 中的一个或者多个层中存在；
- 7、一个数据流包括了两个或者更多的连接的或有依赖关系的元素；
- 8、一个数据流可以包括一个或者多个数据流，可以形成数据流的层次；
- 9、数据流可以在多个业务与 IT 图表中流动；
- 10、元素关系在一个 OBASHI 模型中具备持久性。

7.2 OBASHI 关系类型

在元素间有六种连接关系，每种连接关系都以一种特定的方式工作，为 OBASHI 提供了元素间隐含或者清晰的意义。

连接关系是：

- 连接

- 依赖
- 层
- 集合
- 序列
- 空间

每种关系的详细解释如下：

7.2.1 连接关系

元素可以与其他一个或者多个元素连接。连接是一种双向的关系，即元素 A 如果连接到元素 B，那么元素 B 也连接到元素 A。连接关系是瀑布式的，即元素 A 如果连接到元素 B，而元素 B 连接到元素 C，那么元素 A 也连接到元素 C。

7.2.2 依赖关系

元素可以依赖其他一个或者多个元素。依赖关系是单向的，如果元素 A 依赖于元素 B 那么元素 B 就不能依赖于元素 A。一个元素可以依赖于多个元素。

7.2.3 层关系

层代表了业务与 IT 图上的一个区域。元素可以被放置在层上。当两个或者多个元素被放置在同一个层里面，那么他们有一个隐含的关系，即同层关系。

7.2.4 集合关系

一个集合是一组元素按照一个指定的关系成为集合。不向层，集合在一个业务与 IT 图表中没有地理上的限制。元素可以放置在集合中，并且一个元素可以属于一个或者多个集合。当两个或者多个元素被联系在同一个集合中，他们就有一个隐含的关系，即集合关系。集合提供了一种手段，对于一个逻辑群的元素不用考虑它们在一个或者多个业务与 IT 视图中放置的地方，却能创建这些相关元素的集合。

7.2.5 序列关系

最基础的序列关系是一个元素的列表，在该列表中的元素是按照顺序排列的。在该列表中相互毗邻的元素间必须有一个连接或者依赖关系。

除了元素，一个序列可以包括其它的序列，从而形成一个嵌套的结构。序列因此包括一个链，该链由有连接关系或者依赖关系的元素或者序列组成。不仅仅是毗邻的元素之间有顺序的关系，所有在该序列中的元素相互间都有一个隐含的顺序关系。

7.2.6 空间关系

空间关系揭示了在一个业务与 IT 图表中元素间，其放置位置在地理上的关系。这些控

制元素间所隐含的空间位置关系的法则在本章的后面详细讲解。遵循该法则的元素就可以说相互间有一个空间关系。

7.3 连接关系 连接的重要性

在 OBASHI 模型中连接是非常重要的组成部分，因为它们指出了数据流动的路径。在 OBASHI 中，数据流只能在有连接或者依赖关系的元素间创建。

7.3.1 什么是连接？

连接被用来显示元素间的逻辑和物理的连接。

连接是双向的，即如果一个元素被连接到元素 B，那么元素 B 也被连接到元素 A。

这种关于连接关系的双向性本质并没有说数据在元素间是双向交换的，但是可以清晰的反映出在元素间一定是有一个逻辑或者物理的连接存在。

在物理资源的情况下，连接代表了资产间的物理连接，比如网络电缆连接了一台电脑和交换机（图 7.1 所示），光纤电缆连接了两台路由器，或者背板连接了一个 CPU 刀片和它的宿主机。

将一个比如 Windows Vista 的操作系统的系统层元素连接到硬件层表明了 Vista 安装在那台计算机上，如图 7.2 所示。

在一个所有者、业务流程或者应用层，连接表明了一个逻辑的连接，如图 7.3，‘开发票进程’这个业务流程逻辑的和‘会计’部门和‘SAP’应用连接。

表 7.1 –OBASHI 关系法则 No.3-元素连接性

	所有者	业务流程	应用	系统	硬件	基础架构
所有者	X	X				
业务流程	X	X	X			
应用		X	X	X		
系统			X	X	X	
硬件				X	X	X
基础架构					X	X

7.3.2 元素连接性

OBASHI 关系法则 NO.3 控制哪些层可以与其他层连接。表 7.1 图形化表示了该法则，表明哪种类型的元素可以连接到其他层的元素。一个‘X’表示在两种元素间可以有连接关系。

一个元素类型由其所在的层决定。比如，一个元素在 OBASHI 的应用层则该元素就是应用元素。一个代表一个网络交换机的元素可以被引用为基础架构层元素，因为它放置在基础架构层。

没有一个元素可以与其他的所有层的元素连接。

另外，也没有一种元素可以与在它毗邻层之外的层的元素连接。相同类型的元素可以相互连接。这是因为通过同一层的元素的连接可以生成同一层元素的层次化结构。层次是如何使用的典型例子如表 7.2

所有者	组织结构图和地点可以通过在所有者这层使用拥有关系和拥有关系的连接来表现。
业务流程	业务层次使得业务流程和功能可以不断分解为子流程，根据需要可以直到工作流的级别。
应用	在一个应用层的软件可以被分解为模块功能，或者逻辑组件。
系统	在一个系统层，操作系统的层次可以被用来对虚拟化环境进行建模
硬件	当被在硬件层使用，在硬件项间的连接可以用来对冗余系统和硬件虚拟化进行建模，比如在一个刀片服务器中的刀片们
基础架构	在基础架构项间的连接可以实现对网络建模

7.3.3 在两个元素间的连接类型

表 7.3 显示了所有允许的在元素间的连接，并包括元素能够如何被使用的描述以及它在一个 OBASHI 模型中刻画的内容是什么。

表 7.3 –在 OBASHI 中允许的连接

元素 1	元素 2	描述
所有者	所有者	一个或者多个所有者元素可以连接到另外一个所有者元素来显示一个逻辑组，例如部门结构，或者物理连接，例如地点或者地方。连接也可以用来显示报告结构。
所有者	业务流程	这种连接定义了一个显示对于一个业务流程或者功能的关系
业务流程	所有者	
业务流程	业务流程	在业务流程层，一个或者多个业务元素可以连接来显示在它们之间的数据流动，比如一个以纸为基础的沟通，或者一个业务流程是如何包括其它子流程的。

业务流程	应用	该连接定义了一个业务流程功能或者流程为了使它自己运行如何使用一个应用。
应用	业务流程	
应用	应用	相互连接的应用表明了或者数据在该应用间穿过，在这种情况下它们必须拥有一个共享的基础元素，或者该应用可以被分解为子应用（模块、程序、功能）
应用	系统	表明该应用元素运行在系统层元素所代表的操作系统上
系统	应用	
系统	系统	一个系统元素和另外一个系统连接用来显示虚拟操作系统是如何相互交互的。
系统	硬件	将一个系统元素和一个硬件元素连接起来定义了操作系统运行在的硬件
硬件	系统	
硬件	硬件	在硬件元素间的连接用来刻画物理的非-基础架构项是如何连接的-例如，一个打印机连接到 PC。连接两个都是计算机的元素只能在没有网络基础架构被使用并且这两个计算机直接的相互连接的情况下使用。
硬件	基础架构	该连接用来显示在硬件和网络基础架构间的物理连接。它通常也用来显示计算机如何被连接到一个网络。
基础架构	硬件	
基础架构	基础架构	连接起来的基础架构元素用来显示网络如何被设计和/或者配置

7.3.4 多重连接

OBASHI 并不限制在两个元素间可以存在的连接数量。多重连接可以在两个元素间存在，比如，对有多个网络卡，并连接到同一个交换机的计算机的建模。

一个元素可以和任何数量的其他元素连接，OBASHI 关系规则控制这些连接。

7.3.5 刻画连接

连接被刻画成元素间的直线。它们可以是垂直的或者水平的，或者是一个通过使用垂直和水平的线组成的有右-角度的连接。为了最大化的清晰与方便理解 OBASHI 模型，连接不应该采用弯曲的或者有角度的线。连线的标准颜色是黑色。任何不是黑色的连线都在图中使用都必须做出对该颜色的意义的解释。

7.3.6 瀑布式连接

在元素间瀑布式连接表示，如果 A 被连接到 B，而 B 被连接到 C，那么 A 可以说通过

B 连接到 C。

7.4 依赖关系

依赖关系的存在允许建模者去显示如果一个元素要工作正常则需要依赖于另外一个元素。它显示了如果有错误出现，那么将会对元素的运作产生影响。

这可以是一个物理层面的-比如，一个打印机依赖于一台工作的电脑。或者也可以是一个操作层面的，比如一个业务流程依赖于另外一个业务流程提供信息。它甚至可以是一种捆绑式的，比如一个开发票的流程依赖于一个工作打印机。

这个隐式的关系表明，如果一个元素停止工作正常，那么所有依赖它的元素都会受到该问题的影响。

不象连接关系，它在 OBASHI 关系规则中控制哪些类型的元素可以连接（比如系统、基础架构等），依赖关系可以存在于任何元素之间，无论它们在哪个层。

一个依赖关系是一个单向关系。这是说，如果一个元素 A 依赖于元素 B，那么不是说元素 B 也依赖于元素 A。

如果两个元素间是同时关联，即元素 A 依赖元素 B，而元素 B 依赖元素 A，那么 OBASHI 需要创建两个依赖关系。

数据流仅仅只能在拥有连接关系或者依赖关系的元素之间创建。

7.4.1 可视化指导原则

一个依赖关系的显示方式是：一条红色的从原始元素向依赖元素的带箭头的连线。它可以从任何路径，并和一些元素重叠，但是线必须清晰并且容易被识别。描述依赖性的文字可以在该线上面，比如‘元素 A 依赖元素 B’。

一个依赖关系必须在线上有一个特定的结束标记。在依赖元素处，线的终结点必须有一个箭头指向依赖元素。线的起始处，被依赖元素的地方必须有一个圆圈作为标识，如图 7.5 所示。

7.4.2 依赖关系的使用

为什么在 OBASHI 模型中使用依赖关系有两个初始原因。

1、对完整性建模

OBASHI 方法的灵活性部分是因为它可以获取信息并快速的将其进行建模。通过使用一个依赖关系，建模者可以很容易的显示一个业务或者 IT 资产是如何消费其它业务或者资产所提供的服务，同时没有这些服务它们将无法正常的运行。

设想在财务部门有一个关于通过更有效的运作来降低费用的战略需求。如果业务与 IT

图表不存在，那么可以创建该图表来记录财务部门当前是如何运作的。这就是正常业务状态。

通过对话和咨询可以了解到，很明显为了正常运营，财政部门依赖电子邮件系统。在 OBASHI 图表中有两种方式对这种依赖关系进行建模。电子邮件系统可以使用 OBASHI 来记录，在业务功能-财政和电子邮件系统中建立一个数据流，但是这么做会占用时间并且会被认为超出了当前财政项目的范围。可以选择的另外一种方法是，一个电子邮件元素可以被创建在财政业务与 IT 图表上，或者当它本来就在 OBASHI 模型中进行重用，一个依赖关系来显示这种依赖。

在项目的稍后期，电子邮件系统通过使用 OBASHI 来记录，依赖性可以用一个数据流来替代从而获得得到一些附件的收益。这种 OBASHI 模型的进化形成了 OBASHI 项目生命周期的部分，高阶的图表被细化来提供额外的清晰度来显示一个业务如何工作，以业务的正常业务模型工作。第一章可以获得更多的 OBASHI 项目生命周期的信息。

2、一个视觉上的理解

理解系统间的相互依赖性通常是设计、支持和管理 IT 系统中的一个关键方面。刻画并理解相互依赖性也是能够提出改变业务或者 IT 运作的建议的一个关键。

可以创建使用依赖关系的高层业务与 IT 图表来提供一个相互依赖性的视觉浏览图，这可以帮助沟通和理解。如果需要的话，可以通过不断的细化业务与 IT 视图和数据流分析视图创建更多的细节。

依赖性使得一个业务视图可以非常快速的获得附加的完整性，知识和视觉效果。

7.5 层的关系

当两个或者多个元素被放置在同一层，它们就拥有了一个隐式的关系-同层关系。

所有在同一层的元素间都有一个隐含的相互关系。概念上，这十分重要，以为它使得每一层都成为一个相关资产和资源的登记处。

通过创建业务与 IT 图表的层次关系，这个应用于整个 OBASHI 模型中的资产登记可以被分解成一个逻辑结构，如图 7.6

一个单独的业务与 IT 图表包括了那些使用层关系进行关联的元素。一个或者多个业务与 IT 图表可以组成群，这些群可以被认为成是业务与 IT 图表的逻辑目录。OBASHI 由每一个目录中的每一个业务与 IT 图表组成。图 7.6 显示了这个层次是如何组成的。

通过使用层关系，一个业务和 IT 资源的登记表可以在一个模型层次、文件夹层次或者一个业务与 IT 层次形成。这种可以在多个业务与 IT 图表中使用一个元素，并且可以将

一个业务与 IT 图表放置在多个文件夹中的能力，使得业务和 IT 资源可以在更详细的上下文信息中被分类和分析。

一个例子可以是：识别在一个特定的文件夹里所有的所有者可以用来创建潜在的利益相关者，或者通过分析在两个文件夹中的共用的应用，从而使得项目群管理活动的的能力得到提升。

7.6 集合关系

集合提供了将元素进行逻辑分组的能力，无论元素在业务与 IT 图表中的位置。在一个集合中的元素可以不在同一个业务与 IT 图表中，也就是说一个集合可以涉及到多个业务与 IT 图表。

集合关系并不受其他 OBASHI 关系的限制，比如，层关系受到 OBASHI 法则 2 的限制，即哪些类型的元素可以放在哪里是受到限制的。因为每一层都在图表中有一个定义好了的区域，而这些元素必须放置在该区域内。

集合关系并没有这个限制。任何元素都可以分配到一个集合中，一个元素可以分配到多少个集合中并没有限制。当两个元素被放置在同一个集合中，那么它们就有一个隐式的关系，同时它们和同一集合中的其他元素也都有一个隐式的关系。这个关系就是集合关系。

OBASHI 中使用集合来作为一种将有某种形式联系的元素组成一个集体的手段。没有正式的规则来控制集合，从而形成了一种一群元素集合在一起的松耦合。这种灵活性使得对于 OBASHI 元素更丰富的上下文信息。

例如：OBASHI 建模者可以将所有与电子邮件系统有联系的元素放置在一个‘电子邮件集合’，或者所有的 CISCO 的音乐流网络设备放置在一个‘CISCO Disco 集合’。

7.7 序列关系

对于一个序列的最基础的形态是一个元素的列表，该列表中元素的顺序形成了该序列。在该列表中的元素可以说相互之间有一个序列的关系。

组成一个列表的先决条件是在列表中毗邻的元素必须有一个连接或者是依赖的关系。

除了元素，序列也可以包括序列，生成一个嵌套的结构。序列因此包括一个互相连接的或者依赖的元素和序列的链条。

在 OBASHI 中序列关系被用来对数据的流动建模。

连接或者依赖的元素间的序列形成了数据流动的路径。数据流从序列中的第一个元素源起，它被称为提供者，按顺序通过列表中的每一个元素，最后在称为消费者的最后一个

元素处终结。

在 OBASHI 中使用数据流的一个好处是更好的理解、记录和分析数据是如何在一个组织中流动的。

如果一个资产出现故障而它又是数据流的一部分,那么很明显该数据将不能正常的流动到它的目的地。该链条的一个连接被中断了。

将元素视为数据流链条中的一个连接可以对怎么样使用于资产和资源来给业务交付收益,以及它们是如何对收益做出贡献的提供更多的上下文信息和清晰度。

7.8 空间关系

在 OBASHI 中一个核心的概念是如果一个元素被直接的放置在另外一个元素之上或者之下,那么在两个元素间就有了一个隐式的关系,该关系可以作为元素间的上下文信息。这是业务与 IT 图表建模的一个基础。

空间关系所代表的关系只是存在于元素在业务与 IT 图表上相对于其他元素的地理位置关系。空间规则控制的方法可以提供这些隐式的空间关系。符合该规则的元素间可以说相互间有空间关系。

7.9 关系总结

元素可以通过这六种关系的组合相互关联:

连接: 元素可以显式的连接来显示一个双向的耦合;

依赖: 元素可以显式的以单向的方式与另外一个元素连接来显示一个元素依赖于另外一个元素;

层: 存在于所有的在一个特定的层上的元素间的一个隐式的关系(比如所有者、系统、业务流程层);

集合: 元素的显式的逻辑组,该关系不用考虑元素间在一个业务与 IT 图表中的位置关系。

序列: 一个显式的关系表明在一个列表中的毗邻元素都有一个连接关系或者依赖关系-比如一串互相连接的元素。

空间: 一个隐式的关系,由那些放在相互之间上面或者下面的元素组成的关系,比如业务流程放在一个所有者下面。

7.10 空间关系

在 OBASHI 中空间关系用来对于一个元素如何和另外一个元素在业务与 IT 图表中关联进行分类。

根据一个元素在业务与 IT 图表中与另外一个元素的空间关系不同，可以有很多种空间关系的分类。这种关系有两个部分，第一个是显示在 X-轴空间的位置关系，详细的如表 7.4，另外一个显示 Y-轴空间的位置关系，详细的如表 7.5

表 7.4 –X 轴空间分类

X-轴空间分类	描述
在之内	元素 A 完全的处于元素 B 的 X 轴空间范围内
左边	元素 A 的 X 轴起点在元素 B 在 X 轴起点的左边
右边	元素 A 的 X 轴起点在元素 B 在 X 轴起点的右边
超出	元素 B 完全的处于元素 A 的 X 轴空间范围内

表 7.5 –Y 轴空间分类

Y-轴空间分类	描述
在之上	元素 A 在元素 B 的上面
在之下	元素 A 在元素 B 的下面
顶端	元素 A 的 Y 轴起点在元素 B 在 Y 轴起点的上边
底端	元素 A 的 Y 轴起点在元素 B 在 Y 轴起点的下边
在之内	元素 A 完全在元素 B 的 Y 轴坐标内
超出	元素 A 完全在元素 B 的 Y 轴坐标外

在描述一个空间关系时，通常是用 X-轴与 Y-轴关系联系在一起描述，其中 X-轴信息在前面，Y-轴信息在后面。比如，元素 A 空间上是（在之内，在之上）元素 B，或者元素 A 空间上是（在之内，在之内）元素 B。

对于空间关系的理解使得 OBASHI 的建模者可以设计一个更有意义和精确的业务与 IT 图表。

正确的将元素在 OBASHI 图上定位对于 OBASHI 建模能够正确的被解释至关重要。

空间规则对于使用 OBASHI 来进行影响度分析十分关键，它们也是 OBASHI 影响度规则（后面进行详述）的核心组成部分。当出现了一个资产或者资源问题时，可以用影响度分析来发现其所造成的影响。

表 7.6 给出了用于空间分类的代数上的表达式。如果 OBASHI 模型中的信息将存储在一个数据库或者一个电子表格里，那么可以使用这种表达式来保证报表在空间分类上能够以精确和连续的形式生成。

在 OBASHI 中，X 轴与 Y 轴的起点是在业务与 IT 图表的左上。表 7.6 通过两个矩形元素的引用来描述 X 轴与 Y 轴空间分类，A 和 B，在一个业务与 IT 图表上，每一个元素都有一个它的左下角的坐标 (X1,Y1) 和一个右上角的坐标 (X2,Y2)，如图 7.8

表 7.6-使用代数表达式来对空间分类进行定义

轴	空间分类	代数表达式
X 轴	在之内	$AX1 \geq BX1 \text{ AND } AX2 \leq BX2$
X 轴	左边	$AX1 \geq BX1 \text{ AND } AX2 \leq BX2 \text{ AND } AX2 \neq BX1$
X 轴	右边	$AX1 \neq BX1 \text{ AND } AX2 > BX2 \text{ AND } AX1 \leq BX2$
X 轴	超出	$AX1 < BX1 \text{ AND } AX2 > BX2$
Y 轴	在之上	$AY1 \neq BY2$
Y 轴	在之下	$AY2 \leq BY1$
Y 轴	顶部	$AY1 > BY1 \text{ AND } AY2 > BY2 \text{ AND } AY1 < BY2$
Y 轴	底部	$AY1 < BY1 \text{ AND } AY2 < BY2 \text{ AND } AY2 > BY1$
Y 轴	在之内	$AY1 \geq BY1 \text{ AND } AY2 \leq BY2$
Y 轴	超出	$AY1 \leq BY1 \text{ AND } AY2 \geq BY2$

7.11 OBASHI 影响度分析

一旦一个 OBASHI 业务与 IT 图表被创建，它们可以用来对一个 IT 或者业务资产的问题所造成的可能的业务影响进行分析。

例如，考虑假设一个电脑出现问题所造成的影响。如果计算机出现问题，那么运行在其上的操作系统也会停止工作，而没有操作系统，应用系统也会停止工作。如果应用停止工作，那么使用该应用的业务流程也会收到影响。如果该业务流程被影响，那么与该业务流程的利益相关者-所有者就会受到影响。

OBASHI 影响度法则决定影响是如何在 OBASHI 模型中是一级级的传播的，如表 7.7。这定义了每一个元素是如何将影响传递到其他元素。它也显示了 OBASHI 关系是如何在一个影响度分析中使用。

表 7.7 –元素类型是如何对影响进行传播

被影响元素 A	被传播的影响 B	为了传播影响所必须存在的关系
基础架构	基础脚骨	在 AB 间必须有直接连接，同时 B 空间上在 A 上面
	硬件	在 AB 间必须有直接连接
硬件	硬件	在 AB 间必须有直接连接
应用	应用	在 AB 间必须有直接连接

	业务	在 AB 间必须有直接连接
业务	业务	B 在空间上在 A 的上面或者下面
任意	任意	B 依赖 A
任意	任意	B 在空间上在 A 的上面
任意*	任意*	在序列中是下一个的关系*
任意*	任意**	空间上在 B 之内的, 且在 B 之内的所有元素都被影响了**

*是表明只有数据流被包括进影响度分析;

** 这个法则表明了任意包含其他元素的元素并不会受到在它之内元素的影响, 除非在它之内的所有元素都被影响了;

这些法则应该被反复的使用直到没有更多被影响元素被发现。

第八章 –技术和标准

8.1 介绍

本章包括以下主题：

- 翻译一个业务与 IT 图表；
- 关系持久性；
- 数据流分析视图；
- 逻辑和物理建模；
- 定义逻辑元素；
- 冗余性建模；
- 虚拟化环境建模；
- 定义层次化的元素；
- 使用多连接；
- 连接关系对比依赖关系；
- OBASHI 颜色标准；
- ISO 和英国标准

8.2 解释一个业务与 IT 图表

图 8.1 是一个非常简单的业务与 IT 图表。请学习和解释该图表。

Peter、Craig 和 Jane 在会计部门工作。Craig 和 Jane 处理支票支付工作，使用在他们的 Dell PC（运行 windosxp）的 excel 进行工作。而 Peter 则负责公司被称为 PayMeNow 的薪酬应用，该应用运行在他的 HP PC 上，使用 Vista 操作系统。

所有的在会计部门的电脑都连接到一个局域网网络交换机，而这个交换机则通过一个交换机连接到在计算机房的公司的主路由器上。

一个业务与 IT 图表非常容易解释和理解。

8.3 关系持久性

在一个业务与 IT 图表中使用的元素可以在其他的业务与 IT 图表中得到重用。OBASHI 模型包括了所有的这些业务与 IT 图表。OBASHI 的一个关键概念是在一个业务与 IT 图表中形成的关系可以在所有其他的业务与 IT 图表中得到保持。

如果在一个业务与 IT 图表中，一台计算机显示是在一个特定的操作系统上运行，那么无论该计算机在其他的业务与 IT 图表中如何显示，无论该操作系统是显示还是没有显示，该计算机仍然是运行在该操作系统上。

这个概念被称为关系持久性。在一个业务与 IT 图表中的关系在其他的所有图表中都会保持不变。

关系持久性和能够将元素在多个图表中重用的能力使得复杂的物理和逻辑关系能够通过使用 OBASHI 建模。

图 8.2 显示了 Jane Smith、Peter Jones 和 Frdrick Hass 工作的地点。

图 8.3 显示了 Jane Smith、Peter Jones 和 Frdrick Hass 工作的部门。

Jane Smith 和 Frdrick Hass 是技术支持部门的一部分，Peter Jones 工作在会计部门。

通过将所有的图表结合起来进行分析，可以理解技术支持是与 New York 和 Berlin 相关联的，因为他们有共享的元素 Jane Smith 和 Frdrick Hass。

也可以理解到会计部门和 New York 通过 Peter Jones 联系起来。

关系持久性也可以用来为一个特定的读者量身定制制作图表。

请考虑一下下面的例子图表

图 8.4 是为了一个目标读者所制作的，该读者只关心业务是如何使用 SAP 的。它显示会计部门和仓库都拥有使用 SAP 应用的业务流程。

因为他们被放置在 SAP 应用之上，所以他们是 SAP 应用的利益相关者。这就是说，他们都使用该应用作为他们业务流程的一部分。这可以从模型中看到，甚至于尽管业务流程元素没有连接，它们仍然有一个空间关系表明了它们使用 SAP 应用。（更多的关于使用连接的信息可以在本章的‘逻辑和物理建模’中看到）

图 8.5 是为了更技术化的读者制作的，它可以被一个打算通过整固 licenses 或者将硬件下线的方式来降低运营费用的项目所使用。比如，它显示了 SAP 运行在一个 Unix 操作系统的 SAP-01 的服务器上，SAP-01 使用交换机 SP1，该交换机和路由器 RT9 连接。

通过关系持久性可以理解会计部门和仓库依赖 SAP-01 服务器，因为业务与 IT 图表 1 和业务与 IT 图表 2 共享了 SAP 元素。

在业务与 IT 图表 1 中联系 SAP 与业务流程和所有者的关系在业务与 IT 图表 2 中仍然保持，尽管所有者和业务流程元素并没有被显示。

8.4 数据流分析视图

数据流在 OBASHI 模型中通过一个、两个或者多个连接、依赖的元素的序列形成。元素间

存在数据流必须是相互间有连接的或者依赖的关系。连接、依赖和数据流由 OBASHI 关系规则控制。（见第七章）

一个元素可以在多个业务与 IT 图表中存在，因为它们也可以在多个图表上和其他元素有依赖和连接关系。所造成的结果之一是可以创建在多个图表中流动的数据流的能力。为了描述不是所有的元素都在同一个业务与 IT 图表的数据流，OBASHI 有一个视图被称为数据流分析视图。

数据流分析视图描述了一个使用 OBASHI 层作为放置互相连接的元素的框架的数据流。除了这些元素外，DAVs 应该有描述数据流是什么的文字性描述，为什么它们重要，以及对所有的在 DAVs 上显示的统计数据的总结。（见图 8.9）

在 OBASHI 框架中放置的元素遵循它们在数据流上显示的顺序。元素以带箭头的连线连接起来，代表了流的方向。

元素按照顺序放置在在数据流中在它们前面的元素的上面或者下面。如图 8.6

当不能采用这种做法时，比如当它们的前序的前序是同一个元素类型，那么该前序应该被扩展从而使得连接能够返回前序的前序的那个层。一个例子如图 8.7。

可选的，附加的文字信息可以覆盖在每一个元素和每一个连接上来帮助数据流的分析。这些文字信息可以是财务的、分析的或者引用的数据。比如，在每一个元素和连接上叠加延迟的时间可以帮助计算部分或者完整数据流的延迟时间。叠加人力或者开销/价值的信息可以使得数据流的财务优化成为可能。（视图 8.9）

DAV 除了可以对整个数据流进行整体的描述，还可以通过将元素从 DAV 中移出来，对部分的数据流进行描述。该过程可以在元素级进行，或者在 OBASHI 层进行，所有在该层的元素都不被显示出来。

任何从该数据流移出来的元素**必须**被一个红色的带箭头的虚线所标记，箭头指向被省略的元素那一头。一个数据流的开始和结束的元素**必须**显示，并且不能够从一个 DAV 中移走。采用这种方式将层移走使得 DAV 可以为目标读者裁剪。例子如图 8.8，将硬件与基础架构层移走提供了图表的清晰度，使得它适合于为非 IT 专业人士显示。

如果所有的信息都被叠加到 DAV 上，那么那些被移除掉的元素所累积的信息应该在那些替代那些元素的红色虚线箭头处显示出来。（图 8.9）这么做是为了 DAV 中用于分析的信息不会被遗漏掉，从而保持根据 DAV 上的信息而进行的决策制定流程得以保持完整性。

DAVs 可以在影响度分析中起到很重要的作用。DAVs 显示了那些为了使数据流运行必须显示和运行的元素的顺序。如果在 DAVs 中一个代表了资产的元素和连接收到了破坏，而

且数据不能在这些设备或者资产中流动,那么由此受到影响的元素将不能够接收到它所期望的数据并且数据流将停止。

能够看到这个结果对于故障根源分析的诊断有帮助,并且能够帮助变更控制流程,因为它提供了帮助,能够帮助理解关于资产是如何被一个组织使用的以及资产在组织中的责任是什么。

DAVs 提供了对于元素是如何被使用的更好的理解,并且通过将这些元素放置在一个数据流的上下文中,从而可以执行新式的风险管理。

当要评估对于数据的风险,精确理解数据是如何被组织所消费的十分重要。比如,本地的一个委员会可以有一个部门照看一个公园的草地的维护。他们使用从天气预报中得到的数据来调度草地维护活动的交付。当被公园部门评估时,天气数据可以被分类为非重要,因为它只能被用来帮助做非关键的决策。然而,考虑相同的天气数据被公路管理委员会。如果不能收到该数据将阻碍的委员的能力,去满足他对公路维修的调度职责,因为当温度下降到一个特定的阈值时,公路维修工作有可能会受到很大的影响。因此,这个数据可以被分类为对于公路管理部门的关键数据。通过分析天气信息完整的数据流,而不是孤立的看待天气信息数据,就可以做出合适的风险评估。

在 OBASHI 中一个数据流的第一个元素被成为**提供者**,而最后一个元素被成为**消费者**。

数据流由一个提供者初始化,而在一个消费者处终止。所有在提供者和消费者之间的元素使得数据流可以运行。数据流分析视图提供了一个重要的机制来对一个数据流中的元素的顺序进行沟通和分析。

8.5 逻辑和物理建模

OBASHI 的一个基础概念是在多个业务与 IT 图表中实现元素的重用。这也就是说在某个业务与 IT 图中使用的元素也可以在另外的业务与 IT 图表中显示。这样的元素,尽管在不同图表中都得到描述,它们仍然是被作为相同的逻辑和物理实体所引用。

这种方法的一个好处在于提供了一种能力,能够精确的描述元素在物理上怎么被连接在一起或者物理上在图上处于什么位置,同时又可以在另外的图表上显示它们是如何被逻辑使用的。

单个的业务资产和单个的 IT 资产可以在一个组织的多个部分中被使用是很常见的,但是它们很少被组织中超过一个实体管理,控制或支持。

在业务流程层,人员运行一个业务功能一般情况会给一个与他运行的业务相关的经理人进行汇报。该业务流程因此最终被一个经理或者部门的领导所有,然而该流程可能是被整

体或者部分的被组织中的不同部门所使用。这就是说，该流程可以作为一个服务提供给组织的其他部分。

使用 OBASHI，一个物理模型可以被创建，来显示真正的所有者关系以及报告业务功能的结构。连接这些元素显示了这个报表的结构和元素如何不仅仅是被一个特定的所有者结构使用，但是同时也属于该所有者结构。

通过将元素在其他的图表中重用，OBASHI 显示了怎么使用元素来支持其他的业务功能或者组织的相关部分。通过不连接这些元素，它显示了业务中的那些依赖部分的所有者是业务功能所提供的服务的利益相关者。

比如，图 8.10 显示了一个所有者元素，‘Sales’ 是 ‘Take Order’ 和 ‘Contact Customer’ 业务功能的最初所有者。可以从模型中看出来介入了这些功能的职员 为 ‘Sales’ 进行了工作，但是 ‘Sales’ 使用了其他部门的 ‘Process Order’ 和 ‘Check Warehouse’ 的服务。‘Sales’ 是这些元素所提供的服务的利益相关者。

通过观察第二个业务与 IT 视图（如 8.11），可以发现 ‘Process Order’ 是一个由 ‘Accounts’ 交付和支持的，而 ‘Check Warehouse’ 则是 ‘Logistics’ 的一部分。

通过用这种方法建模，利益相关者的身份变得十分的清晰。无论由 ‘Accounts’ 所提供的 ‘Process Order’ 服务产生什么样的变化，对于受这个改变影响的业务部分都会清晰的被看到。那些由于服务提供所产生改变而应该被通知或者介入的人员都很容易被识别出来。

使用连接来显示基础架构层和物理层的物理连接清晰的显示了 IT 资产是如何的物理部署的。将硬件层和应用层的元素连接起来描绘了哪些应用和操作系统部署在同一台硬件元素上。

在图 8.12 中的例子说明了一旦一个元素的连接在物理级被显示，它不需要在一个与其相关的元素已经被显示的逻辑模型上重新画出来。

在这里显示的 IT 部门负责 IT 基础架构和应用的支持和维护的监管。然而，应用 ‘SAP’ 与 ‘ORACLE’ 由会计和运输部门使用来支持它们的业务流程。

在大部分的业务中，IT 资产通过都是由 IT 职能来支持。然后，由 IT 资产所提供的服务的用户遍布一个组织中的各个部门。逻辑和物理建模提供了一个方式来清晰的描述物理连接的资产（物理模型），同时也显示它们似乎如何逻辑的被组织所使用（逻辑模型）。

8.6 定义逻辑元素

8.6.1 什么是逻辑元素？

逻辑元素是在 OBASHI 模型中那些代表非物理实体的元素。它们为一些目的服务。

逻辑元素在建模流程中被创建。它们可以以一组逻辑元素一起的方式来帮助建模变得更清晰。

它们也可以用来建立一个元素的层次,这个层次可以在模型中的任何地方被引用而不用进行复制。

8.6.2 生成逻辑元素

生成一个逻辑元素和生成一个非逻辑元素类似。和非逻辑元素一样,逻辑元素的生成必须依据 OBASHI 与它生成后所在层的规则控制。

在 OBASHI 模型中增加一个元素,要为其起一个代表一个非物质实体或者属性的概念的名字,这个概念可以定义其它元素的特征。

通过连接业务流程元素, workflows 可以被建模。在图 8.13 中显示了一个例子,该例子代表了当一个公司接收到一个发票时发票处理流程是怎么样。OBASHI 模型显示了在业务运行的每一个阶段会涉及到哪些利益相关者。Jane 检查这个收到的支票,看其是否符合相关规定与标准。Steve 与相关部门进行确认,确保在提交到支持流程之前支票是没有问题的。Joe 执行每周的 BACS,将支付递交到支付流程中。尽管这已经很精确和清晰了,但是进一步提供两个说明性元素可以进一步的为 OBASHI 提供更多的上下文信息。在图 8.14 中的 ‘Accounts Dept’ 和 ‘Invoice Processing’ 元素阐述了逻辑元素的概念。

被成为 ‘Accounts Dept’ 的逻辑元素代表了一个其他元素的逻辑组。通过将 Jane、Steve 和 Joe 放在该逻辑元素的下面,可以从该模型推断出他们工作在会计部门。

逻辑元素 ‘执行发票’ 提供了一个对于单个的业务流程 ‘Review for Compliance’、‘Check with Department’、‘Submit to Payment’ 与 ‘Weekly BACS Transfer’ 逻辑的结构。使得它们成为 ‘Invoice Processing’ 的一个子流程。

可以使用逻辑元素来对地点进行建模,比如国家、城市和办公室等。在图 8.15 中给出了例子,为会计部门提供了更详细的上下文。

增加逻辑元素 ‘New York’ 和 ‘London’ 在图 8.15 中允许建模者看到该模型中的更多的关系。可以看到,会计部门工作在 New York 和 London。同时也能看到关于 Jane、Steve 和 Joe 的额外的关于工作地点的上下文信息, Jane 和 Steve 都工作在 New York,而 Joe 则工作在 London。

除了能够提供关于业务如何运行的更多的上下文信息,逻辑元素也可以被用来帮助建模。由于 OBASHI 中有关于元素间的地理关系,因此可以定义一个逻辑元素来代表其他的多个元素。

不管一个元素在 OBASHI 模型中的哪里被使用,一个元素的所有关系都一直存在。这被成为关系持久性,在本章的前面详细说到过。

在图 8.15 中, ‘Invoice Processing’ 的元素有一个与四个子流程的关系。无论 ‘Invoice Processing’ 的元素在哪个图表中被重用, 这个关系都一直存在, 并且适用于该元素的任何实例。

子流程也是 ‘Supply Chain Management’ 的一部分。

因此, 流程 ‘Review for Compliance’、‘Check with Department’、‘Submit to Payment’ 与 ‘Weekly BACS Transfer’ 都是组成 ‘Supply Chain Management’ 的一个部分。

这帮助了建模, 即, 在一个业务与 IT 图表中重用或者引用所有这四个元素可以采用直接简单的引用 ‘Invoice Processing’ 的方式。

另外一个关于在 OBASHI 中如果重用元素, 则应该对元素进行分解的例子已经在该模型中使用了。

考虑如 8.17 的业务与 IT 图表。

它显示了 PC1、PC3 和 PC4 运行 ‘Office’ 应用, 而 PC2 和 PC5 则运行 ‘Office Pro’ 应用。

考虑将以下的元素引用到该业务与 IT 图表中 (图 8.18)。

对 Office 定义分解到它的子组件允许我们获得 ‘Office’ 到底是什么的清晰理解。

在 OBASHI 中, 无所谓哪个业务与 IT 图表首先被创建。拥有通过扩展一个元素的定义到子元素从而将模型细化的能力可以使得模型更加清晰, 并且可以帮助快速的建立模型。还是关注通过定义 Office Pro 来使用 Office 的例子, 它显示了 Office Pro 是 Office 的一个超集, 包括了额外的 ‘Groove’ 和 ‘Access’ 应用。

8.7 冗余性建模

当一个或者多个元素在另外一个元素的内部建模, 一个空间就被建立起来。该空间关系表明外面的元素包括那些在它里面的元素。这对于影响度分析有着特殊的意义。

OBASHI 影响度规则 (参见第七章) 定义了任何将其他元素封闭在其中的元素, 例如 空间 (在之内, 在之内) 其他元素, 不会受到在它里面元素的影响, 除非所有它里面的元素都受到影响。

这使得 OBAHSI 可以来对冗余性建模。考虑如图 8.19 的例子。

在硬件层, 硬件元素 Cluster A 包括 Vax1、Vax2 和 Vax3, 而 Cluster B 包括 Vax4、Vax5 和 Vax6。

OBASHI 项目生命周期

简介

本章节主要描述 OBASHI 项目阶段的管理。其中包括在每一个阶段必须包

含的过程以如何得到和梳理所需要的信息用于 OBASHI 的建模。

本章节可以让读者领悟到 OBASHI 项目生命周期具有高度的可伸缩性和灵活性,用以推动 OBASHI 项目的交付。

以下是章节名称和章节简述

第 9 章-OBASHI 项目生命周期
OBASHI 项目生命周期-回顾

第 10 章-范围

“范围”阶段主要定义业务的需要。
OBASHI 项目的目的是为业务需求服务。

第 11 章-获取

“获取”阶段主要收集和分析用于 OBASHI 模型的信息。

第 12 章-设计

在设计阶段, B&IT 和数据流模型成为核心活动, 来为获取阶段得到的信息建模。

第 13 章-精炼

精炼阶段, OBASHI 被用于开发详细的 B&IT 关系图。

第 14 章-交付

交付阶段主要是确保在 OBASHI 项目中生成的交付物最终能传递到业务运行环境。

第 15 章-其它

B&IT 编号
OBASHI 项目文档

第 9 章 - OBASHI 项目生命 周期 - 回顾

9.1 回顾

OBASHI 项目生命周期如图 9.1 所示是一系列项目阶段用于指导基于 OBASHI 方法论的项目实施。

OBASHI 项目生命周期的每一个阶段都将有详细的描述，将一个项目从初期的构想过渡到成功的交付。

范围

在 OBASHI 项目生命周期中，范围阶段将创建一系列的文档来支撑项目的立项。这些文档可以用于获得项目执行的授权。

获取

获取阶段收集和分析用于建立 OBASHI 模型的信息。过程明确数据来源，与干系人进行访谈以及对已获取数据的评估。

设计

在设计阶段，“获取”阶段获得的信息被与 B&IT 和 DAV 一起进行建模，得到 B&IT 的模型图。其中包括了在“范围”阶段所确定的项目实施领域。

精炼

精炼阶段对 B&IT 模型图进行钻入分析，增加更多的细节。这一活动的目的是确保 B&IT 的最终效果能交付用户并得到有效的项目成果确认。

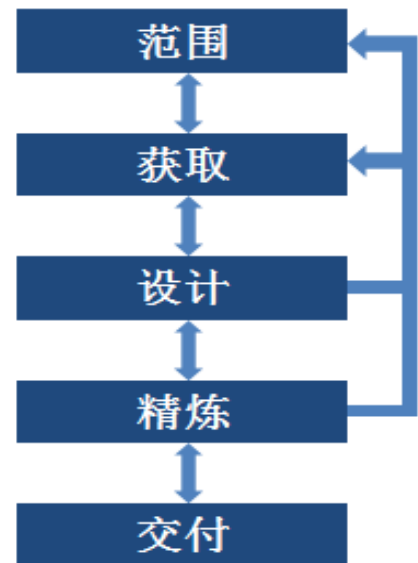
交付

交付阶段是为了确保项目过程的交付物被传递至业务运行环境。

9.2 项目行政管理

图 9.2 显示了如何利用 OBASHI 项目生命周期来完成 OBASHI 项目。

每一个阶段都被分解为核心任务、文档以及流转过程。项目行政管理也参与其中，项目管理过程和生命周期阶段能和好的融合。项目行政任务中的“指导项目”和“控制阶段”可以适当的调整

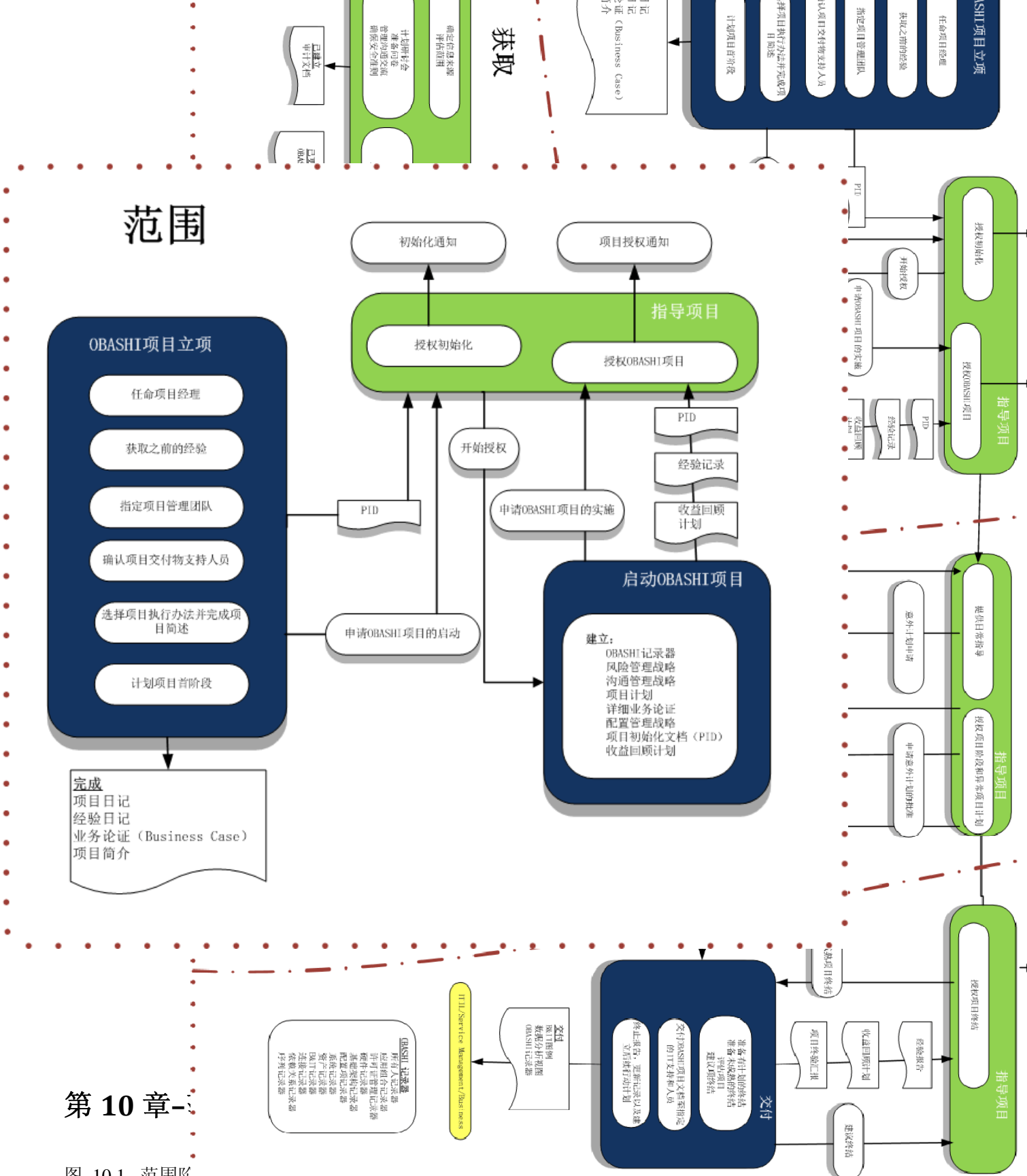


已符合被选的项目管理方法论。这些方法论包括 PRINCE2，PMBOK 或企业/机构内部的项目管理办法。

PRINCE2 通常被认可为最基本的项目管理架构。如果了解该项目管理方法论的话会对图 9.2 中的某些内容比较熟悉。

本文目的不在阐述项目管理方法论，只是试图将项目管理方法论作为一项参考，为 OBASHI 项目生命周期的实施提供管理依据。因此，OBASHI 项目生命周期不会针对某一特定的项目管理方法论。事实上 OBASHI 项目生命周期可以与任何一个成熟的项目管理方法论配合。

任何其他项目管理方法论的用户可以考虑将图 9.2 中蓝色字体部分替换为自身的专用项目管理术语。



第 10 章-

图 10.1 –范围阶

10.1 定义范围

本章对范围这个项目阶段的实施进行了描述。
如图 10.1 所示，范围阶段定义了业务需求。
OBASHI 项目目标是满足业务需求。

范围阶段的目标建立项目立项文档。项目简述定

核心活动覆盖了业务需求的定义，建立 OBASHI 项目概述和项目基本结构的设定。参看本章节‘核心任务’部分。

交付物或是‘整体效果图’或是一系列文档来帮助项目立项过程的授权。

10.2 建立一个 OBASHI 项目

在启动 OBASHI 项目时，需要执行 6 项核心任务。

对项目管理方法论比较了解的读者会很快识别出 OBASHI 项目范围定义阶段的任务结构。项目的规模和复杂度将决定该阶段需要耗费的时间。

10.3 核心任务

1. 委任行政及项目经理

从项目干系人中确定核心决策人及行政人员。行政人员必须是管理团队的成员。该成员项目的核心干系人，他/她的地位可以影响他人来力挺这个项目。

确定最合适的项目经理。

2. 获取以前的经验

在项目初期，对以前执行过的 OBASHI 相关项目的经验进行回顾和总结。这些经验可以是无价的信息来源，用于确保新项目能在预算范围内按时交付。

3. 委任项目管理团队

项目管理团队需被指定和委任。项目管理团队可以包含项目委员会（Project Board），项目保障（Project Assurance）以及项目支持团队。这一步的目标是使每一个成员理解在项目管理和支持过程中，他们的角色和职责以及项目的汇报机制。项目管理团队的每一个人必须理解和认可谁向谁汇报，汇报什么；谁的职责是什么；汇报的线索及接口人。

4. 指定移交后的支持团队并使他们介入项目

确保项目至运行支持的高效移交将大大增加项目成功几率和价值展现。较早的使运行支持人员介入项目的执行过程可以降低项目推广的阻力。需在运行团队中指定核心人员作为项目的支持人，或者将这些人员纳入项目沟通管理战略中。

5. 选择项目推进办法(Project Approach)和形成项目简述

在计划项目前，需要确定项目的推进办法。这些决策中的一部分可以分解到 OBASHI 专有的方法论。

OBASHI 专有的决策包括项目是自顶向下或自底向上，项目的范围以及数据的颗粒度。这些都对项目的成功起到了关键作用。

将影响项目推进过程的非 OBASHI 相关决策包括企业或机构的战略和动机、业务策略、行业法规和管控对时间、金钱、质量以及资源利用率和资源可获取性。

项目简述（Project Brief）包含了使项目能到达预期的目标所必须完成的项目工作内容。这是一项重要文档，它可以帮助确定项目的规模。

6. 计划和启动阶段

该阶段详细定义了为获取项目授权所需要的文档系列的创作计划。这个阶段包括对业务论证（Business Case）、项目计划、风险管理策略以及

沟通管理策略的详细计划。
除此之外还应包含下述文档。

10.4 文档

以下是范围阶段应该产生的文档列表。部分文档模板包含在附录 H 中。

■ 日记录

可以用于记录未在其他 OBASHI 项目文档中记录的一些行为和重大事件。可以把日记录理解为项目经理的项目日记。内容可以包含事件和行动。例如日期、职责以及事件和行动的结果。参见附录 H。

■ 经验记录

项目执行过程中遇到的问题都有可能会对其他 OBASHI 带来参考价值。这些问题应该记录至经验记录。记录应该包含场景的描述, 对项目的影响, 采取的行动和效果。参见附录 H。

■ 项目简述

项目简述提供了项目的方向和范围。它形成了项目管理团队和机构管理团队的相互理解。项目简述包含的内容形成了项目初始化文档 (PID) 的基础。参见附录 H。

■ 概要业务论证 (Business Case)

这是初始化文档, 目的是建立项目的成本价值分析, 对项目的成本以及财务和非财务的投资回报分析。该论证证明了为什么要启动项目。

■ 风险管理策略

该文档详细描述了在 OBASHI 项目中如何管理项目的风险。包含的内容包括风险如何归类和评估, 考虑事件发生的概率, 以及事件可能带来的影响。作为风险管理的一部分, 项目管理团队应询问机构对项目风险的接受程度。对项目风险的接受度进行勾画和明确。例如, 低影响度、高可能性的风险可能被接受, 因为这类风险对项目整体的影响可以相对忽略而高影响度, 高可能性的风险则不能接受。

■ 沟通管理策略

该文档勾画了项目的沟通该如何进行, 包括项目内部沟通的线索以及信息该如何从项目发布至机构更复杂的环境中。OBASHI 项目可能会需要收集涉及商业机密的信息。项目团队应该认识到这点

并在沟通的内容和传播方式上体现这方面的考量。

■ 项目计划

项目的实施计划, 包含核心里程碑和交付时间。

■ 详细业务论证

这是概要业务论证的详细版本。该文档提供了项目授权所需的信息。

■ 资产记录器

资产模型覆盖范围内对资产信息的记录器。记录了资产的名称、类型、简要描述以及唯一识别号。参见附录 H

■ 拥有人记录器

模型中拥有人成员的记录器。参见附录 H。

■ 业务流程记录器

模型中所有业务流程成员的记录器。参见附录 H。

■ 应用组合记录器

模型中所有应用成员的记录器。参见附录 H。

■ 许可证管理记录器

模型中每一项应用和系统的许可证记录器。参见附录 H。

■ 硬件记录器

模型中所有硬件成员的记录器。参见附录 H。

■ 基础架构记录器

模型中所有基础架构成员的记录器。参见附录 H。

■ 配置项记录器

对每一个配置项及相关关系的记录器。参见附录 H。

■ 系统记录器

模型中所有系统成员的记录器。参见附录 H。

■ B&IT 记录器

对每一个 B&IT 图表信息的记录器。参见附录 H。

■ 立项文档

该文档可以作为一个检查列表, 来确保项目立项

所需要的 OBASHI 项目的核心方面都得到了讨论和考虑。立项文档可以作为整个项目实施过程的参考文档。参见附录 H。

需要注意的是，通过实施数据库应用或者其他软件产品来记录和关联这类信息是非常有价值的。

第 11 章-获取

■ 已有文档

■ 个人知识

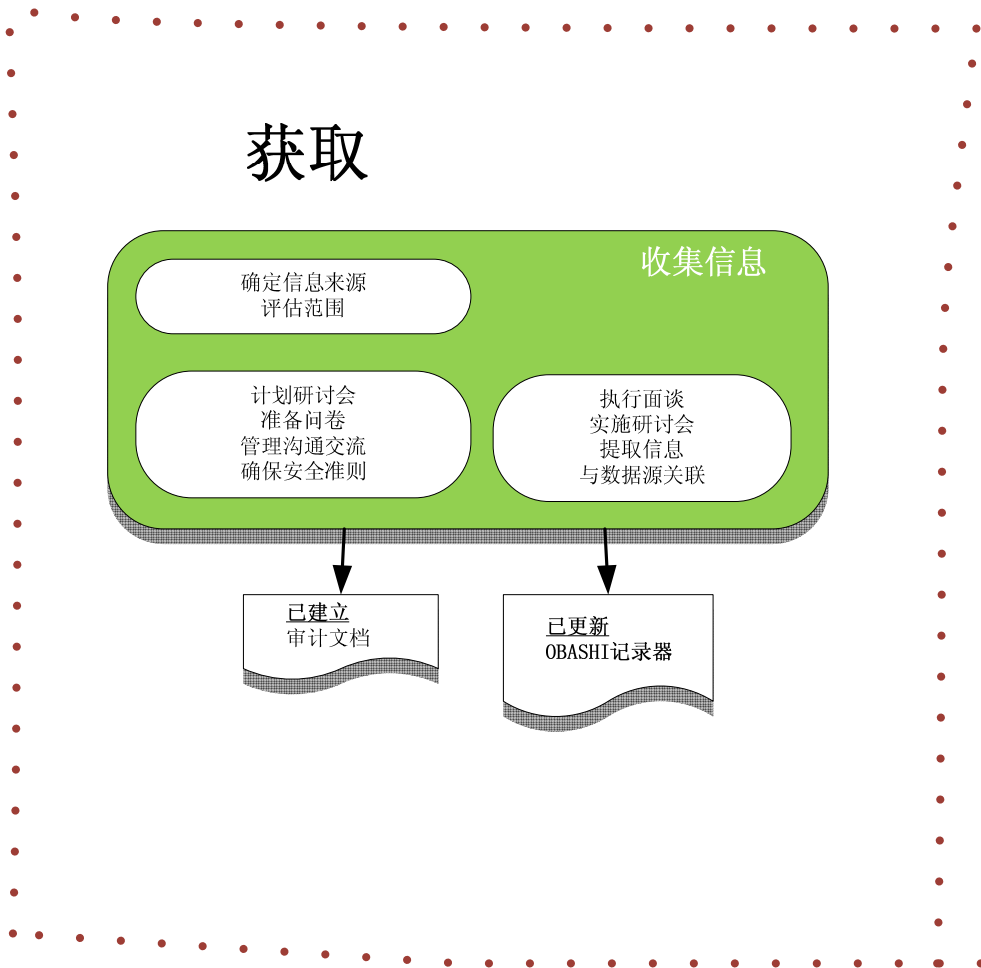


图 11.1 –获取阶段

11.1 收集信息

获取阶段目标

是为了获得和校验信息。这些信息用于代表 OBASHI 不同层面的 people, process 和 technology。

核心活动

包括与 OBASHI 个层面相关的人员的访谈。明确资产和资源，获得授权以分析数据库、信息库以及计算机网络。

可交付物

该阶段的可交付物为每一层的主要参照记录器。这些记录器包含了所有核心的资源和资产信息，这些信息被用作建立 B&IT 图的参考信息。

在发现数据源时需要考虑和分析以下 4 项基本领域：

- 系统和应用
- 网络和硬件

11.2 核心任务

11.2.1 第 1 步-确认

信息来源

在项目范围之内，从确定可能的信息来源开始。方法是参考项目启动文档(PID)，与项目行政人员沟通。

确定信息来源以使：

- a) 确定文档的所有人和地点。
- b) 确定需要介入或了解项目的干系人，部门领导以

- 及核心人员。机构的电话簿是一个比较好的入口。
- c) 确定系统管理员和应用管理员。
 - d) 利用资产管理/基础架构管理/服务台工具等数据库的输出和报表。

表 11.1 列举了可以采集的信息分类。

11.2.2 第 2 步—基于信息源评估范围

这一步的主要目的是对第一步获取的数据源进行量化和评估,并且找出与需要导入到 OBASHI 记录器的数据要求差距。这些信息需要通过举行研讨会和咨询活动的方式来获得,最终促成设计阶段的启动。

- a) 回顾在项目启动文档(PID)中定义的项目范围
- b) 评估项目范围与已获取信息源的差异,舍弃无关及不可靠信息。

至此,可靠的信息基线便已确立。

- c) 评估 OBASHI 的每一层与基线信息的差异,记录缺失的信息。

11.2.3 第 3 步—计划研讨会(Workshop)

计划的目的是:

- a) 从第 2 步得到的输出中明确需要添加至记录器中的信息和数据类。例如,PC 信息、系统拥有人、基础环境资产。
- b) 明确掌握这些信息的,或者了解如何得到这些信息的核心员工。
- c) 明确第三方和服务供应商的介入情况。
- d) 确定研讨会的时间计划。

综合考虑项目的规模和复杂度,举行多 OBASHI 层面的研讨会来截取或发现信息源的采集流程。

表 11.1—信息源

信息源	格式	信息采集描述	备注
电话目录	纸质/电子	个人、部门和联系方式	电子
网络拓扑	纸质	A3 格式图形	可能已失效-需 IT 提供建议
组织结构图	纸质	部门领导	需要更新
个人	Word	人力资源角色和职责	联系 HR 部门的某某
供应商列表	Word	第三方支持	联系采购部门
桌面资产管理	DBF	Novell 桌面资产管理应用	需要管理员权限 已获取资产索引
具体系统	服务台	服务台	服务经理需要提供数据库抽取的建议

11.2.4 第 4 步—准备问卷

通过与项目成员的对话可以确定文档和信息层次和质量。

通常文档和信息的质量和数量无法尽善尽美,而且可能信息已经过时。这步过程的目的是准备问卷用于和核心人员进行面谈以获取最新的信息。

- a) 与核心人员沟通
- b) 获取任何已有的本地文档
- c) 分析并决定本地文档的有效性、质量和准确性。
- d) 将有用的信息源载入目录
- e) 通过修改业务流程问卷模板来创建业务流程问卷。

添加业务流程的描述和在应用列中添加选择项

- f) 对每一类 OBASHI 项目覆盖的业务流程执行“e)”过程
- g) 与核心人员回顾问卷中的缺失内容和错误
- h) 更新问卷

至此,业务流程问卷文档准备完毕可以进入第 7 步。

问卷的样本可以参见附录 H。

11.2.5 第 5 步—管理沟通

各方的合作是 OBASHI 项目的核心要素。与干系

人以及高层管理团队的沟通渠道必须在进入第 7 步—执行面谈前进行明确。

该步骤的目标是建立良好的沟通关系，最大程度降低对现有业务的影响。

- a) 明确将被 OBASHI 项目影响的部门领导和高级管理人员
- b) 请求项目行政人员通过邮件或以个人的方式将您介绍至所有相关部门的领导和高级管理层
- c) 通过电话或个人的方式与每一位作简短的介绍
- d) 给每一位发送邮件，内容包括项目的范围简述，大致的项目周期，介绍信息质量的重要性
- e) 解释第 8 步—举行研讨会对于需要各核心人员和部门协同的意义
- f) 勾画项目的里程碑和预计的时间跨度

11.2.6 第 6 步—确保安全许可

从项目行政管理获得网络访问的授权。

- a) 与项目行政管理一起明确项目中需要得到安全许可的领域。这些可能包括但不限于，楼宇、网络、数据和文档。
- b) 列出可能受影响的项目人员
- c) 讨论并决定项目人员的能力和和安全许可级别
- d) 以字面的形式，请求项目行政管理为 OBASHI 项目人员获得安全许可。

11.2.7 第 7 步—执行面谈

这步的目的是通过问卷来从业务人员的大脑中获取业务流程的知识。

- a) 安排与面谈对象的会议
- b) 描述项目的范围和排除内容
- c) 基于事先定义的问卷来执行面谈

11.2.8 第 8 步—举行研讨会

这步的目的是通过举行一系列的研讨会来得到 OBASHI 层级中需要的知识。

其中可能需要与第三方协调来得到和整理数据。例如，数据中心的部分可能外包给第三方来维护部分信息。

表 11.2—数据集索引

- a) 给出 OBASHI 项目的概要介绍
- b) 描述需要收集的信息和数据集
- c) 讨论并记录获取信息和数据的流程
- d) 指导成员来计划和准备第 9 步—提取信息和第 10 步—建立与数据源的链接
- e) 获取并处理研讨会上发现的信息。

11.2.9 第 9 步—提取信息

这步的目标是以可管理的方式和系统化的方法从电子系统中获取数据。每一项数据集需要被命名并归档至目录以备查询，如表 11.2 所示。

- a) 与网络管理员/系统管理员举行会议，建立访问网络环境的安全规程
- b) 着重描述在进行网络分析过程中设备在线的重要性。在沟通管理的策略框架下讨论如何与员工沟通。
- c) 建立会议和工作启动的时间计划
- d) 计划哪些网段需要纳入分析过程以及哪些数据类型需要提取。如果有条件，可以通过网络分析和发现工具来采集和提取数据至数据集。数据集应以检索的方式进行归类，如表 11.2 所示。
- e) 与网络管理员/系统管理员举行技术相关会议，了解独立运行的 PCs/LANs/外围设备的细节和地点信息。
- f) 计划哪些 PCs/LANs/外围设备需要被分析和调查，哪些数据类型需要被提取。利用网络工具来收集未加入核心网络的 LAN。独立运行的 PC 和外围设备需要手工采集。数据集应以检索的方式进行归类，如表 11.2 所示。
- g) 核心网络数据集此时应被回顾，清理和验证，以符合映射至 OBASHI 记录器或数据库的要求。
- h) 单独运行设备此时应被回顾，清理和验证，以符合映射至 OBASHI 记录器或数据库的要求。

第 10 步—链接至数据源

- a) 与数据库管理员/系统管理员举行会议，建立访问数据库环境的安全规程
- b) 着重描述在进行数据源链接过程中数据库在线的重要性
- c) 建立会议和工作启动的时间计划
- d) 链接和查询数据库用于产生数据集。数据集应满足表 11.2 对索引的要求。

	数据集描述	数据库名
数据集 a		
数据集 b		
数据集 c		
数据集 d		
数据集 e		
数据集 f		
数据集 g		
数据集 h		
数据集 i		
数据集 j		
数据集 k		
数据集 l		
数据集 m		
数据集 n		
数据集 o		
数据集 p		
数据集 q		
数据集 r		
数据集 s		
数据集 t		

11.3 输出

获取阶段的输出包括：

11.3.1 审核文档

获取信息的过程形成了审核的线索，该线索包含信息的获取源，获取办法和检索办法。这些内容构成了未来即将创建的业务知识库中信息的审核文档集。

11.3.2 OBASHI 记录器

获取阶段得到的信息将被更新至 OBASHI 记录器。如果记录器被手工维护和更新，记录器以目录和文档形式存在。如果记录器以数据库的方式存在，数据集将被更新至数据库。

OBASHI 记录器的列表：

- 资产记录器
- B&IT 记录器
- 连接记录器
- 依赖关系记录器
- 集合记录器
- 拥有人记录器
- 业务流程记录器
- 应用组合记录器
- 系统记录器
- 硬件记录器
- 基础架构记录器
- 许可证管理记录器
- 配置项记录器
- 序列记录器

第 12 章-设计

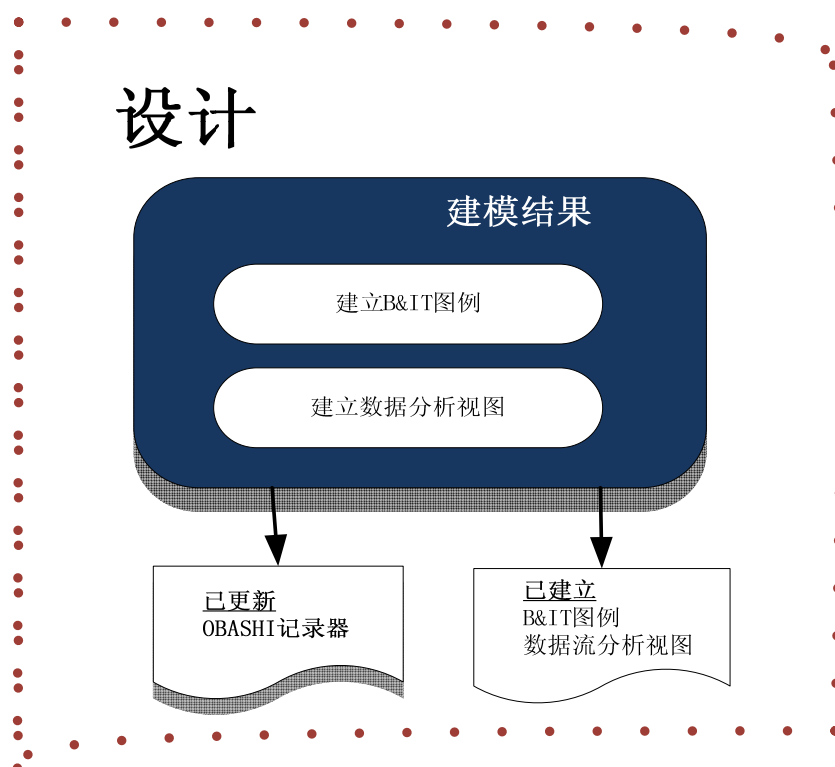


图 12.1 -设计阶段

12.1 模型结果

如图 12.1 所示，设计阶段的目标是对获取阶段得到的信息进行分析和建模。

该阶段的核心活动

是设计和开发业务及 IT 图例（B&IT）以及数据流分析视图（DAV）用以分析和评估出下一过程的动作。

交付物

在该阶段通过对输出物的分析，B&IT 以及 DAV 便成为未来业务架构优化工作的基础。

分析过程可能会指回项目的范围阶段以校正项目目标，或者会指回到获取阶段以期在进入优化阶段之前对数据模型的关联度进一步深化。

12.2 建立一个 B&IT 图例

B&IT 图例可以手工绘制或通过图形软件绘制，或者通过 OBASHI 认证的软件应用来建模。

OBASHI 认证的软件，可以利用拖拉（drag and drop）的方式直接将资产列表中的资产和资源生成为图例中的元素。

在此阶段需要对元素的摆放位置进行思考，以保障摆放的位置符合层次化的逻辑结构模型。

B&IT 图例可以从任意层级开始往上或往下建立。可以选择自顶向下（拥有人至基础架构）或自底向上（基础架构至拥有人）。任何一个层级可以作为一个起始点，并保持与范围阶段中定义的项目推进方法一致。参见附录 B 中有关 B&IT 的创建信息。

12.2.1 第 1 步-添加元素（通过手工绘制或软件工具）至：

12.2.1.1 拥有人层级

以拥有人记录器为源，将拥有人元素加至 B&IT 图例。输入记录器中存放的拥有人名。对 B&IT 图例需要的所有拥有人重复以上操作。

12.2.1.2 业务流程层级

以业务流程记录器为源，将业务流程元素加至 B&IT 图例。输入记录器中存放的业务流程名并放至 B&IT 图例。如果业务流程的拥有人需要在 B&IT 图例中展示，则将业务流程元素置于拥有人元素

下。对 B&IT 图例需要的所有业务流程重复以上操作。

12.2.1.3 应用层级

以应用记录器为源，将应用元素加至 B&IT 图例。输入记录器中存放的应用名并放至 B&IT 图例。如果业务流程元素或系统元素需要在 B&IT 图例中展示，则将应用元素置于业务流程元素之下，系统元素之上。对 B&IT 图例需要的所有应用重复以上操作。

12.2.1.4 系统层级

以系统记录器为源，将系统元素加至 B&IT 图例。输入记录器中存放的系统名并放至 B&IT 图例。如果应用元素或硬件元素需要在 B&IT 图例中展示，则将系统元素置于应用流程元素之下，硬件元素之上。对 B&IT 图例需要的所有系统重复以上操作。

12.2.1.5 硬件层级

以硬件记录器为源，将硬件元素加至 B&IT 图例。输入记录器中存放的硬件名并放至 B&IT 图例。如果系统元素或基础架构元素需要在 B&IT 图例中展示，则将硬件元素置于系统流程元素之下，基础架构元素之上。对 B&IT 图例需要的所有硬件重复以上操作。

12.2.1.5 基础架构层级

以基础架构记录器为源，将基础架构元素加至 B&IT 图例。输入记录器中存放的基础架构名并放至 B&IT 图例。如果硬件元素需要在 B&IT 图例中展示，则将基础架构元素置于硬件流程元素之下。对 B&IT 图例需要的所有基础架构重复以上操作。

12.2.2 第 2 步—回顾元素的摆放位置

一旦所有的元素都已放至 B&IT 图例，需回顾它们的位置的数量。检查连接的路径最符合视觉冲击力和清晰度。

在这个阶段或许有必要举行一系列的会议，将 B&IT 图例展示给拥有者和干系人以获得验证，以进入连接流程（第 3 步）。

12.2.3 第 3 步—连接元素

- 手工将关联的元素画上黑色的连接线，并与第八章中描述的 OBASHI 视觉标准保持一致
- 利用认证的软件工具完成连接

用于完成关系映射和依赖关系的知识源于获取阶段的面谈和对后台支撑资产及资源的了解。

12.2.3 第 4 步—关联相互依赖元素

- 手工画一个起始点为圆圈的红线（针对其他元素依赖于此的元素）另一端为箭头指向依赖的元素，并与第八章中描述的 OBASHI 视觉标准保持一致。红线上可以通过文字来表述依赖关系
- 利用认证的软件工具中的依赖函数来完成这项工作

典型情况下，一个项目会有多个 B&IT 图例。重复以上步骤直到所有元素、连接和依赖关系都被加至 B&IT 图例。

12.2.5 第 5 步—为特别信息添加标示（Hold）和注释（Note）

标示（Hold）是 B&IT 或 DAV 图例上的提示，用以告诉查阅人图例中的部分区域被确定为包含特别重要的信息需求或者需要提请注意。

标示（Hold）含有 3 项特点，如图 12.2 所示：

- 云状的区域覆盖了标示的范围
 - 文本框详细描述了特别重要的信息需求或者需要提请注意的内容
 - 若干红色的三角，提示了标示（Hold）云与文本框的关联
- a) 在需要关注的元素或部件周围画一个标示（Hold）云
 - b) 在标示（Hold）云的里面或周围放一个小红三角。三角中间的数字对应右边的文本注释框的一条注释
 - c) 在注释列中加入编号的的文本，描述（Hold）云中的问题。

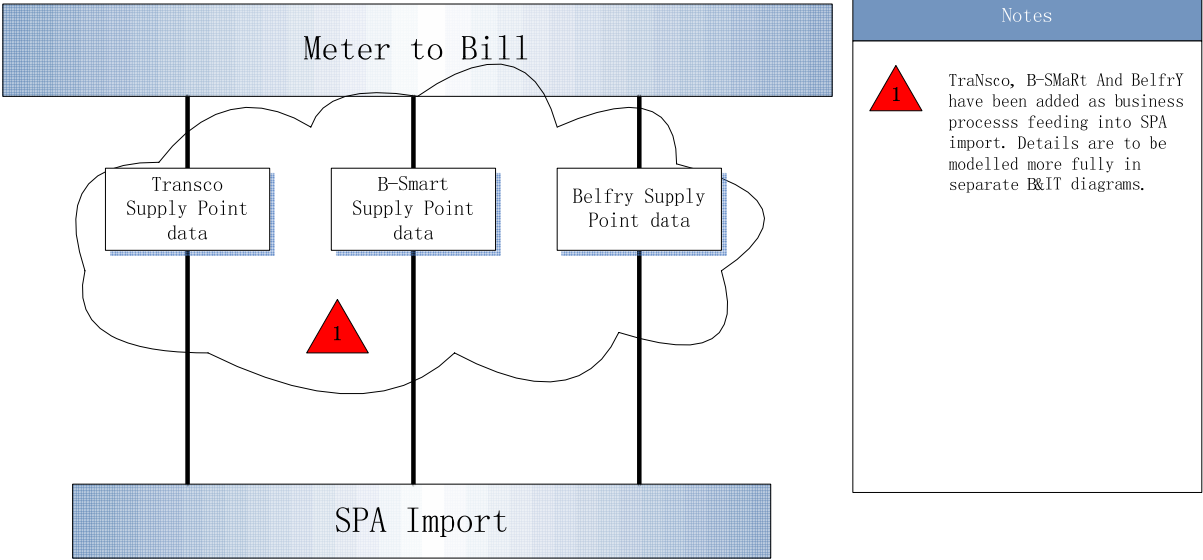


图 12.2 -标示 (Hold) 举例

第 13 章-精炼



图 13.1 –精炼阶段

13.1 精炼 B&IT 图例

精炼阶段的目标

是准备 OBASHI 模型以及与项目有关的所有业务和 IT 图例、数据流分析视图。目的是获得审批通过。

核心活动

在此阶段的核心活动包括通过回顾、批注以及更新 B&IT 及 DAV 图例来达到分析的目的，以及评估出下一轮的动作。分析的结果或者会指回到项目的范围阶段以期对项目的目标进行调整或者指向获取阶段以期对数据模型的关联度进一步深化。核心活动还包括对 OBASHI 记录器的回顾。在回顾阶段得到标示的风险需得到有效升级以获得校正。因为这些风险可能对项目的财务或范围带来影响。

任何被发现的新问题或新机会也需要被升级，以供讨论和检查。

交付物

交付物为被审批通过并签收的一系列 B&IT、DAV 图例以及 OBASHI 记录器。这些交付物作为‘主拷贝’的运行文档，可以被交付至业务部门使用。

13.1.1 第 1 步-回顾 B&IT

B&IT 图例至此应该已经成为结构化状态。项目团队成员应该可以清晰的从图例中看出拥有人与业务流程的对应关系，以及这些业务流程用到了哪些技术资产和资源。这一 B&IT 的回顾过程可以被用作对每个 B&IT 的精炼和优化以期它们能非常接近于项目范围中定义的业务期望和要求。

- 组织回顾团队-理想情况下，这个团队应包含 OBASHI 各层级的业务人员以及 OBASHI 项目的参与者
- 讨论对 B&IT 图例的大的结构调整。例如，将一个图例拆分为两个图例以表述客户端和服务器端两种情况，或者将核心业务信息放在一个图例中，

而把供应商信息放在另外一个图例中。

新。

- c) 用红笔标注回顾团队的想法和讨论结果中的追加内容
- d) 对每一个 B&IT 图例中的标示 (hold) 云以及对应的注释进行讨论。决定这些标示是应该被抹去或者保留至下一个版本。

B&IT 图例上的标示 (hold) 或标示 (hold) 云代表了缺失的信息或者代表该部分内容需要调查和检查。

将回顾团队的的意见进行标注。

13.1.2 第 2 步-回顾 DAV

DAV 是 B&IT 图例的输出。

如果是手工完成 DAV, DAV 元素在层级中的序列和摆位的准确性、关联的一致性需要与 B&IT 进行对比, 确保双方的一致。

DAV 的任何错误应该追溯至 B&IT 图例并在 B&IT 图例中标示。

13.1.3 第 3 步-更新 B&IT

- a) B&IT 图例中如果包含标示, 则应该回到设计阶段进行进一步的开发
- b) 一旦 B&IT 标示更新完成, DAV 图例应随即更

13.1.4 第 4 步-回顾记录器 (仅供被授权的人士)
OBASHI 记录器可能包含保密的业务和个人信息。需要确保只有被授权的人士可以有权查看或访问记录器中的信息。

- a) 对照 OBASHI 记录器和 B&IT, 检查是否存在不一致和异常。

13.1.5 第 5 步-第一版的审批签署以及收集建议

a) B&IT 审批-在 B&IT 的标题框填入修订的详细内容。

修订号: 1
描述: 设计 (添加合适的修订描述)
日期: 0/00/00 (添加当前日期)
签名:

- b) DAV 审批 - 在 DAV 的标题框填入修订的详细内容。

修订号: 1
描述: 设计 (添加合适的修订描述)
日期: 0/00/00 (添加当前日期)
签名:

- c) 记录器审批-与 B&IT 图例审批流程一起, 所有 OBASHI 记录器必须得到签署。

第 14 章-移交

14.1 移交

移交阶段的目标

是完成 OBASHI 项目并将 OBASHI 项目的知识和文档转移至业务日常流程的负责人。理想状态下，文档主体（即，B&IT 图例、DAV 以及 OBASHI 记录器）应进入关键业务文档的流程化管理，得到有效的控制和管理。

该阶段的核心活动

包括对项目结束的准备。含对计划完结项目，未正常完结项目以及建议完结项目的关闭。项目的评估以及项文档交付至指定的 IT 支持团队也在范围之内。

交付物

为知识的转移以及全套原版的 OBASHI 项目文档。这些应该包括当前业务现状以及项目生命周期中形成的说有图例修订和标示。这些修订和标示形成业务转变的路径。

14.2 结束 OBASHI 项目

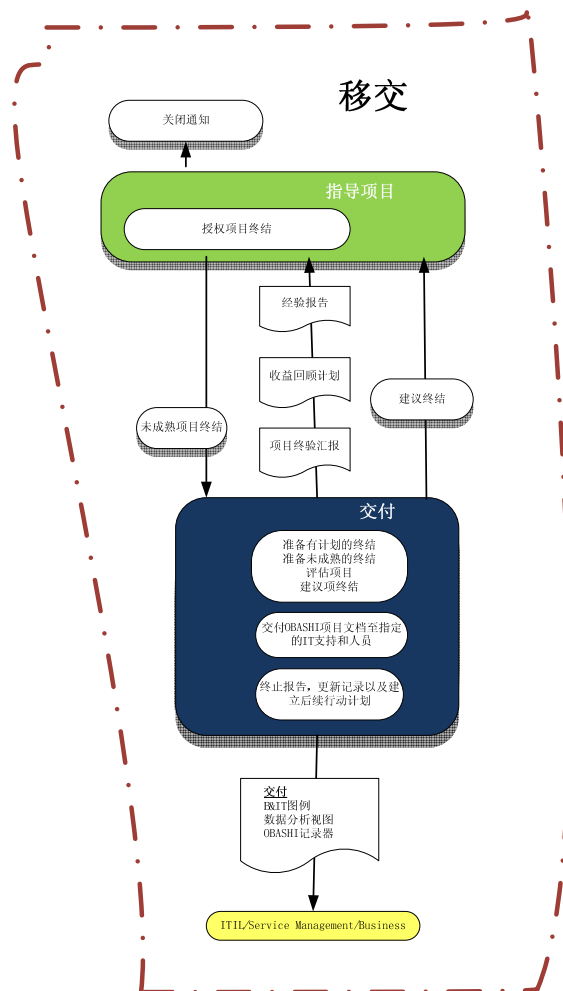
项目流程的结束必须与已选择的项目方法论保持一致。

14.2.1 第一步-准备有计划的关闭

- 确定核心运行人员负责 OBASHI 项目文档的维护人
- 与负责移交的支持团队举行会议讨论文档的类型、数据库的格式积极电子化数据的格式
- 设计移交的过程和机制。应考虑变更控制流程
- 建立移交的日程计划和过程

14.3 ITIL/服务管理/业务

B&IT 图例以及数据流转分析视图（DAV）包含了具有业务价值的资产和配置数据。事实上，这些图例是配置模型。这些图例模型会被用于在运行环节来管理和维护配置管理系统。因此，这些图例应该包含在变更控制之中。



在 ITIL 的环境中，应该是负责服务资产和配置管理流程的功能实体负责接受 OBASHI 模型。

第 15 章-其它

15.1 B&IT 编号系统

B&IT 是关键的运行和战略文档，用于展示资产、资源以及数据流之间的核心关联关系，给机构的业务服务带来价值。
大量关键业务决策将基于这些图例所展示的事实。

每一项 B&IT 图例应被视作文档的主版本，其控制过程应遵循关键业务文档的控制办法。

所有 B&IT 图例应该有一个唯一的识别编号，可以顺序递进。

一下是 B&IT 图编号系统中 5 个部分的举例。其中包含 5 个组：

- (1) 两位数数字，
 - (2) 两位数数字，
 - (3) 3 个字母的识别符，
 - (4) 5 位数数字，
 - (5) 字母 R,C 或 E，后面是一位或两位数字。
- 这 5 部分应该通过短横线连起来。

				2	R
				5	2
				1	2
				0	A
				0	
				-	
			(((
			3	4	5
)))

- 1. 两位数数字：可以代表用户编号或者项目编号
- 2. 两位数数字：可以代表物理位置
- 3. 3 个字母：可以代表数据源所在部门，小组或个人
- 4. 五位数数字：可以代表唯一的 B&IT 图例号
- 5. 4 位数字和字母组成的编号：可以代表对 B&IT 图例和文档负责维护的部门、小组或个人

往往 B&IT 文档包含多个 B&IT 图例，因此每一个图例应该有相应的编号 – 例如，36-10-FIN-2500-R22，页面 1/3,2/3,3/3。

15.2 OBASHI 项目文档

除了 B&IT 及 DAV，OBASHI 项目的文档可以分解为两个不同的区域：

15.2.1 项目管理

图 15.1 所示项目管理文档可以为基于 OBASHI 方法论的项目提供严谨的工程化的管理办法。在本部分（C 部分）我们采用了 PRINCE2 的项目管理方法论。但任何其他项目管理方法论同样适用。

15.2.2 表单和计数器

表单和计数器文档负责管理 OBASHI 模型的数据关联一致性。OBASHI 模型的关键组件的构造确保了任意两个元素间的关联关系的建立，以及之下的数据可以很容易的被获取。

- 当一项元素被建立时，下述的动作也被执行：
- 1. 一个条目被输入至资产计数器。该条目有一个唯一的识别好（资产 ID），条目的类别，名称（资产名）以及描述（资产描述）
 - 2. 一个条目被输入至 OBASHI 层级计数器，例如拥有人计数器。该条目包含资产 ID、资产名和资产描述
 - 3. 与元素类型相匹配的的表单被建立–例如，匹配拥有人元素的拥有人表单。元素相关信息应被填至表单。这样的表单可以是问卷形式的，在 OBASHI 项目生命周期的获取阶段得到填充。
 - 4. 一个针对该元素的资产 B&IT 表单被创建。该表单记录了资产名和资产号。该资产 B&IT 表单与元素表单共同存在，用于记录保存该元素的 B&IT 图例。
 - 5. 必要的话，可以在配置项计数器中添加一项，在配置项表单中填写相关信息，并与元素表单一起存放
 - 6. 必要的话，可以在许可证管理计数器中添加一项，来记录该元素的许可证信息。通常这被用于系统和应用类的元素。

在向 B&IT 图例中添加元素时，也会在该元素的资产 B&IT 表单中增加一项记录。

在建立两个元素间的连接关系时，也会在连接计数器中建立一条记录。

在建立两个元素间的依赖关系时，也会在依赖关系计数器中建立一条记录。

在建立一个集合（Set）时，也会建立一个集合计数器。集合的名称、描述以及建立集合的原因需

要加入到计数器。集合中的每一个元素被记录至集合计数器。集合包含元素的原因也须被记录。

当一个元素被加入集合中或被从集合中删除时，集合计数器也会被更新，用来反映集合的更改。

在建立序列关系时，也会建立一个相应的序列计数器。序列名、描述以及建立序列关系的原因需

要加入到计数器。序列中的每一元素被记录至序列计数器。

在建立 B&IT 时，也会在 B&IT 计数器中建立相应的记录。会记录 B&IT 图例的唯一识别符、名称和描述。B&IT 的唯一识别符需遵循第 15 章中描述的编号系统。

附录 A – OBASHI 如何服务于 P3O

A. 1 组合、项目群和项目管理（PORTFOLIO, PROGRAMME AND PROJECT OFFICES – P3O）

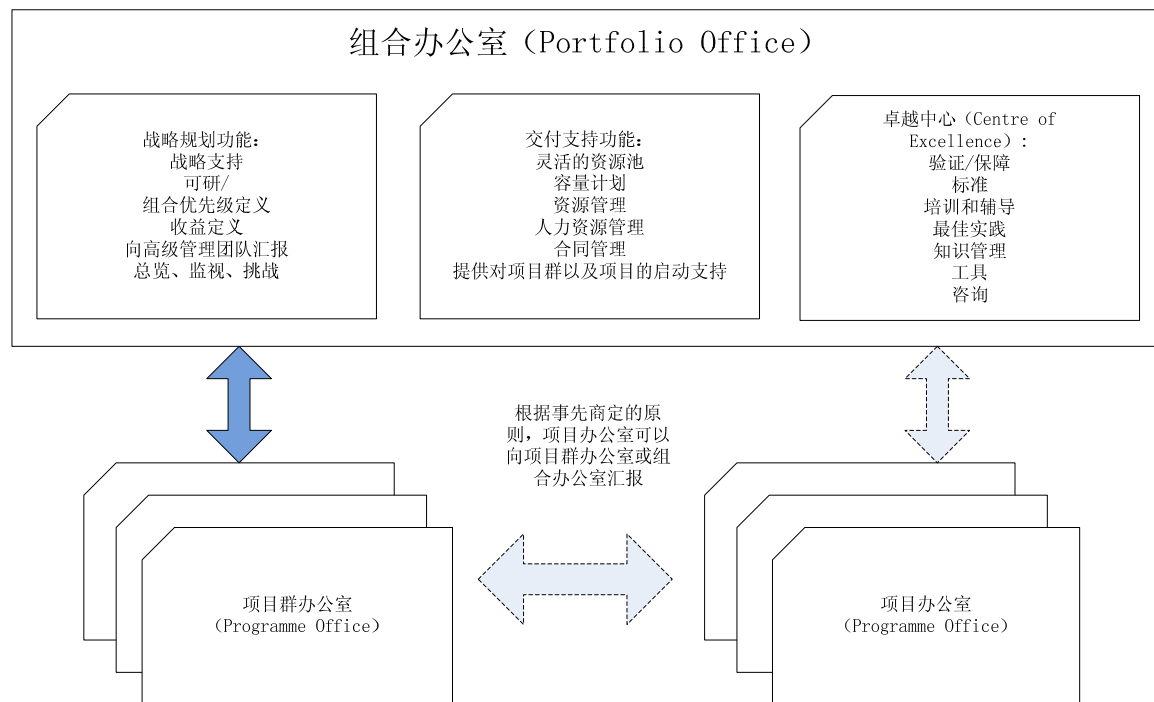


图 A.1 –组合, 项目群和项目办公室 (P3O) 模型

组合, 项目群和项目办公室 (P3O) 模型的建立应该基于各自不同机构的个性化要求。因此, 模型本身、模型的职能体以及模型的角色都会因此有所不同。图 A.1 举例说明了一种 P3O 的设置方式。

虽然 P3O 模型可以有多种不同的设置方式, 但在决定模型时一个核心的判断条件是机构的组合、项目群、项目以及风险管理 (P3RM) 的成熟度。

机构的 P3RM 成熟度可以有如下的影响:

- P3O 办公室的选择
- 模型所提供的功能和服务的范围
- 模型包含的角色
- 模型中包含的人员数量

P3O 模型的目的是帮助一个机构更好的从项目群以及项目的投资中攫取价值。P3O 模型通过以下原则来交付这些价值:

- 有效性的治理
- 对责任人的确定
- 对投资的优先级以及与业务战略的融合与调整
- 价值的保护
- 对人员和流程的投资

- 通过高亮和异常报告来跟踪进展

A.1.1 OBASHI 能帮到什么

- 提供了衡量进展的基准
- 提供了高层管理团队的理解方式和决策能力
- 使得工作目标能写入对工作岗位的描述, 对工作的交付职责有更明确的定义
- 通过对单点故障的认知来确定风险点
- 监视进展以定义可能存在的机会和与战略的差异
- 通过协助完成对交付过程业务和 IT 开支的确定, 为业务论证 (Business Case) 流程提供支持
- 提供了变更控制的基础

A.2 组合办公室

在 P3O 的语汇中, ‘做正确的项目群和项目’ 是与组合办公室向关联的 – 亦可称为 ‘验证 (validation)’:

- 业务变更战略 – 我们是否在做正确的事?
- 业务变更价值 – 我们是否得到了正确的回报?

其它比较典型的由组合办公室提供的战略规划功能包括:

- 战略支持
- 项目群和项目的优先级设定

- 向高层管理团队汇报（管理仪表盘）
- 治理支持
- 总览，监视和挑战

- 工具
- 咨询

A.2.1 能帮到什么

- OBASHI 通过提供对当前以及预计中未来的业务架构和 IT 运维的宏观的、一致性的视图来（big picture）支持上述这些功能。

通过提供如下内容来支持服务的交付：

- 灵活的资源池
- 容量计划
- 资源管理
- 人力资源管理
- 合同管理
- 提供对项目群以及项目的启动支持

A3 卓越中心(CENTER OF EXCELLENCE - COE)

COE 的核心功能之一是检验机构是否‘正确的执行项目群和项目’ - 这也可以被称为校验（verification）：

- 业务变更设计-我们是否以正确的方法在工作？
- 业务变更交付-我们的工作是否得到了有效执行？

其它 COE 功能通常包括：

- 标准
- 培训和辅导
- 最佳实践
- 知识管理

A.3.1 OBASHI 能帮到什么

如果能将 OBASHI 的角色加至 COE, OBASHI 的适用性就明显了。对一些特定的 COE 角色，包括风险管理角色或工具使用专家，OBASHI 可以展现其辅助的增效价值。OBASHI 角色可以成为相关 B&IT 图例（知识管理）的‘监护人’，确保标准的合规性（B&IT 图例中定义的需求的交付），提供 OBASHI 软件的使用以及培训，以及为组合内项目群和项目提供咨询形式的服务来为战略目标的交付提供建议和指导。

OGC 系列产品通常覆盖了管理组合、项目群或项目需要做哪些。但通常很少说明‘How to’。OBASHI 通过提供当前业务的运作情况、通过提供各种‘假如’场景以及通过提供蓝图级别的图例来弥补了这一空缺。

A.4 项目群办公室（临时的或永久的）

可以成立永久的项目群办公室来支持机构内部所有的项目群或者成立临时的项目群办公室来支持某一个项目群的项目交付。

项目群办公室为项目群经理以及配置管理提供建议和支持。

A.5 项目办公室（临时的）

项目办公室的建立是为某一个项目提供支持，为项目经理提供建议和支持。

附录 B – OBASHI 如何服务于 MSP

B.1 管理成功项目群（MSP）

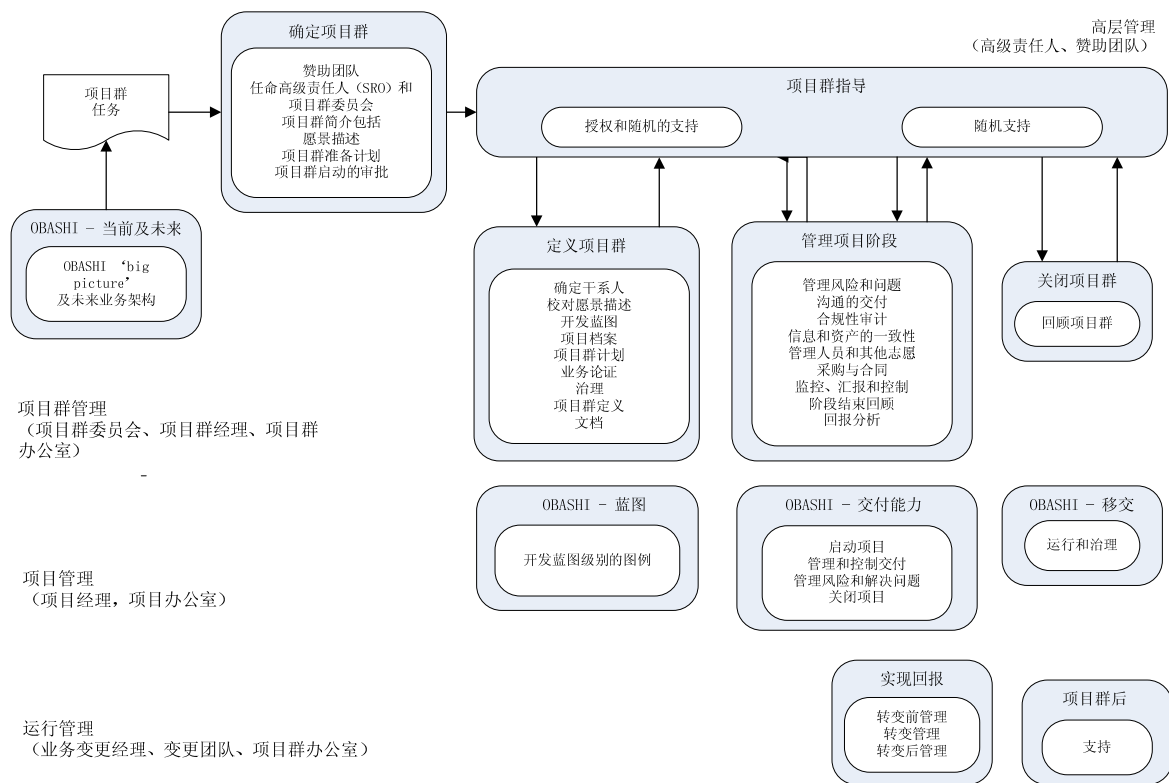


图 B.1 – 项目群生命周期：管理成功项目

基于管理成功项目群（MSP）的项目群生命周期如图 B.1 所示。该图展现了一个项目群如何从初期的想法到业务收益的交付，以及成功交付和实施一个项目群所涉及到的不同的管理层级。它也同时展现了 OBASHI 方法论是如何帮助项目群团队依据机构的目标、愿景和价值观来交付项目群。

B.2 项目群任务（PROGRAMME MANDATE）

项目群经理或高级责任人（Senior Responsible Owner - SRO）将获得项目群任务用以启动进一步对项目群的可行性调研。

项目群任务描述了：

- 项目群的战略目标
- 交付的最佳战略
- 可以预期的提升和回报
- 项目群如何与其他动机目标吻合

B.2.1 OBASHI 能帮到什么

- B&IT 图例描述了得到共识的未来的业务架构。这些信息提供了核心的信息输入用以确定上述要素。

B.3 确定项目群

在对项目群的任何一个重大投资前，应对项目群是否可行和必要进行初步的分析和回顾。需要约定核心人物来参与决策。这些核心人物包括赞助组、高级责任人和项目群委员会。还需要准备一份项目群简介来对项目群的交付内容以及可行性进行描述。

B.3.1 赞助组

赞助组由高级经理组成。这些高级经理是业务的核心责任人，他们需要对项目群提供一定级别的业务承诺。

B.3.2 OBASHI 能帮到什么

- OBASHI 的拥有人层级可以帮助确定哪些代表不同业务的高级经理将组成赞助组。

B.3.3 任命 SRO 和项目群委员会

需任命高级责任人（SRO）来前瞻性的引导项目群。项目群委员会由业务经理组成，为 SRO 提供支持。

B.3.4 OBASHI 能帮到什么

- OBASHI 拥有人层级在确定从哪些干系人中选择 SRO 和项目群委员会成员起了关键作用
- OBASHI 业务流程层级可以帮助确定变更对最核心业务的影响，从而帮助确定在项目群委员会上需要展示哪些核心业务领域。

B.3.5 包含了愿景描述的项目群简介
项目群简介的核心内容包括：

- 愿景概要描述
- 对机构对能力诉求的概要描述
- 对核心干系人的早期查看
- 业务论证的概要
- 对回报的概要描述，包括如何衡量
- 风险
- 已知的限制、假设、问题和冲突
- 大致的时间跨度和成本
- 交付的可选项
- 对当前状态的评估
- 初始的候选项目清单

B.3.6 OBASHI 能帮到什么

- 被认同的未来业务架构可以为愿景概要描述提供重要的输入，用于展现可能被启用的新流程和新的 IT 架构以及哪些人可能会被变更所影响
- B&IT 图例可以帮助确定开支以及新的业务架构，例如，通过简化流程，如何降低开支和提升服务水平，以及如何通过提升容量和减少咨询成本来增强新市场的渗透能力
- 核心风险，例如单点故障和对人工系统的依赖，可以被识别。OBASHI 使得对这类风险的管理有更多的办法。
- 假如（what-if）场景可以帮助确定交付的选择
- 大图（big picture）展示了当前的运行场景，也可以通过它来理解当前工作模式的限制
- B&IT 图例可以在更高级来帮助定位所需要的项目

B.3.7 项目群准备计划

项目群计划在确定项目群的过程从产生。项目群计划定义了作为定义项目群流程中一部分的‘项目群定义文档’的先决条件。它包括：

- 定义项目群的资源条件
- 作为项目群定义文档一部分的交付物，包括蓝图该如何、何时进行开发
- 治理和管控
- 项目群管理团队，包括项目群委员会、业务

变更团队和项目群经理

B.3.8 OBASHI 能帮到什么

B&IT 图例中的 OBASHI 获取阶段展示了被认可的的未来业务架构。该架构提供了需要交付哪些 IT 和变更哪些流程以最终交付项目群的高层级视图。通过理解项目群的规模、复杂度和风险以及完成定义项目群定义的要素，可以帮助项目群的准备计划的完成

B.3.9 进展的审批

此项内容包括：

- 独立的检查，可以由卓越中心（COE）或第三方的保障团队负责对以下内容进行检查：
- 项目群简介
- 项目群准备计划
- 赞助组对进入定义项目群前的审批

B.3.10 OBASHI 能帮到什么

- 与组合管理（组合办公室）有主要的界面。组合管理可以输出某一项目群的计划战略交付物并与已定义的项目群任务和被认可的业务架构未来状态进行比对。

B.4 定义项目群

项目群的详细计划和定义开始启动。用于勾勒机构未来状态的蓝图和项目群计划得以产生。这些内容可以告诉大家项目群是如何逐渐被交付的，以及项目群的业务论证是如何考虑业务的需要的。

B.4.1 确定干系人

为了使项目群能够成功，与核心干系人的有效沟通，得到他们的认可和投入是很重要的。需要建立项目群沟通计划以确保在正确的时间将正确的信息告诉给正确的人。

B.4.2 OBASHI 能帮到什么

- OBASHI 拥有人层级可以帮助确定哪些人与系统和流程发生关系，以及哪些人会被项目群影响。

B.4.3 优化愿景描述

在明确项目群初期形成的愿景概要描述在此得到回顾和优化以确保在项目群的核心沟通平台上所有的干系人需求都得到了考虑。

B.4.4 OBASHI 能帮到什么

- 被认可的未来业务架构图为明确干系人有哪些、核心的信息是哪些提供了重要的输入。这些信息甚至也可以与愿景描述同时存在。

B.4.5 开发蓝图

对于特定的业务领域或部门，了解业务的目标运作模型是非常重要的。

项目群寻求建立以下模型：

- 流程和业务功能
- 运作新的业务所需要的机构、技能以及人员
- 工具、技术和设备
- 支持新业务运作的信息和数据

B.4.6 OBASHI 能帮到什么

- OGC 项目群管理指南，MSP，并不提供如何建立项目群的建议。OBASHI 能提供相关的工具来完成此项工作。可以通过创建更详细的蓝图级的 B&IT 图例来支持蓝图的开发。
- OBASHI 业务流程层级可以帮助找到需要变更的核心流程。
- OBASHI 拥有人层可以帮助找到负责运作新业务流程的人员，并用于确定人员需要的技能和人员数量
- OBASHI 应用、系统、硬件和基础架构层级记录了运作业务模型所需要的技术和工具
- B&IT 图例的整体目标是理解和记录数据流。这些数据流对了解业务的管理信息至关重要

B.4.7 回报档案（Benefit Profile）

了解为什么要变更是非常重要的，例如，可以期望哪些提升。MSP 记录了预计的回报（可衡量的提升）并归档于回报档案。回报档案应该包括：

- 有哪些衡量途径来确定所需要的改善确实已经实现了
- 提升的预期时间和衡量时间
- 为了实现提升效果，应如何改变流程和系统
- 哪些业务会从提升中获益

B.4.8 OBASHI 能帮到什么

用于支持蓝图的 OBASHI B&IT 图例可以帮助对上述的各因素进行确定和记录

B.4.9 项目档案（Project Dossier）

项目档案可以帮忙确定需要哪些项目以及每一个项目的交付物，项目与项目间的依赖关系，每一个项目的贡献和回报以及项目正常交付需要的时间

和资源。

B.4.10 OBASHI 能帮到什么

用于支持蓝图的 OBASHI B&IT 图例可以帮助项目群经理确定和记录这些因素。

B.4.11 项目群计划（包括资源需要）

项目群计划将依据项目档案的记录将项目按照交付的顺序和逻辑顺序排列。排列的逻辑顺序依赖于：

- 相互依赖关系
- 风险
- 共享的资源
- 支撑其它项目的项目（例如，基础建设项目）
- 干系人的需要
- 为了减少对正常业务的影响而需要交付耦合的能力
- 对机构的回报，包括为了交付快速的效果（Quick wins）

B.4.12 能帮到什么

- 蓝图级别的 B&IT 图例可以帮助确定依赖关系和风险点，尤其在应用、系统、硬件和基础架构层面。另外，在这些层级上的流程间的相互依赖关系也会从业务流程层级中显现出来。
- 这些图例可以帮助确定长期的和短期的交付物，从而帮助确定快速效果。

B.4.13 业务论证（Business Case）

业务论证收集以下信息并进行平衡：

项目群计划中提供的时间、开支和资源信息以及回报档案中列举的回报。

平衡的内容有：

- 一边是回报
- 一边是开支、时间、风险、资源、负面影响和问题

B.4.14 能帮到什么

- B&IT 图例可以被用于获取开支信息和协助业务论证的产生
- 通过对 B&IT 图例的分析来找到单点故障（分别从应用、系统、硬件和基础架构）、对核心人员的依赖（拥有人层级），以及对不适用的系统和应用的依赖，从而完成对风险的确定
- B&IT 图例还可以用于对业务回报的确定。例如简化的流程、降低的成本和提升的容量可以带来更

高的销售额或更低的运行风险。

性。确保资产得到安全保护，其保密性得到维护。

B.4.14 治理

每一个项目群需要建立一套自身的管控框架。如果组合管理已经具备的话，大量关于治理，包括报表的需求和模板，应该已经被定义以满足机构的需要。一项持续的需求是监督未来业务架构（蓝图）的交付进展，监督项目群与机构目标、愿景和价值观的融合度。

B.4.15 OBASHI 能帮到什么

- 蓝图级别的 B&IT 图例可以被用作监督进展的基线

B.4.16 项目群定义文档

这是定义项目群过程中产生的信息的汇总，被用于将项目群展现于赞助组，帮助赞助组授权项目群的推进。

B.4.17 能帮到什么

蓝图级别的 B&IT 图例可以提供项目群所交付的能力和 new 服务的总结。这些总结无需 IT 背景，无需大量介入项目群即可被核心干系人理解。

B.5 管理项目阶段

项目群需按照认同的治理框架进行控制和监督。与机构战略目标的融合度需要得到监督。SRO 及项目群管理委员会需要回顾项目群，赞助组需要在阶段末确认其对项目群持续不断的承诺。

B.5.1 管理风险和问题

任何威胁已建议蓝图交付或业务论证可行性的风险和问题需要在项目群这个层次得到解决。解决人可以是 SRO 甚或是赞助组。

B.5.2 沟通的交付

干系人的介入是项目群成功的关键。没有核心干系人的认同和承诺，项目群将失败。

B.5.3 合规性审计

审计被用于决定项目群是否满足预定义的治理要求。保障性的回顾，例如关卡回顾（gateway review）将建议项目群是否可行和值得。

B.5.4 信息和资产一致性

项目群应保障项目群信息和资产配置管理的有效

B.5.5 管理人员和其他资源

项目群需要确保人员得到有效管理，人员评价得到有效执行。还必须确保其他资源的有效利用。

B.5.6 采购及合同

项目群需要确保项目群的采购行为得到有效执行，同时合同得到有效管理。

B.5.7 监控，报告和控制

项目群从以下来源收集数据：

- 项目（进展以及风险/问题）
- 运行（业务正常）
- 业务变更团队（按预期的回报交付）
- 干系人（沟通的反馈和风险及问题的更新）

项目群以此将项目是否按照期望在执行等情况汇报至核心干系人。

B.5.8 阶段结束的回顾

项目群被分解为若干阶段（tranches）。这些阶段代表了项目交付物中阶段性能力和回报的变更。在这些项目群的关键点上，需要有回顾来决定项目群是继续执行、重新调整或应该停止。

B.5.9 收益回顾

收益回顾对截至到当前，预期的收益是否达成，或者预期的收益是否还能获得。

B.5.10 能帮到什么

- 作为项目群定义文件的一个部分，蓝图级的 B&IT 图例得到签署确认。这些图例用于被检查要求的能力或服务的提升是否正在或已经得到了交付。
- 这些图例可以作为保持与干系人沟通的一种途径，通过它来告诉干系人项目群需要交付哪些内容

B.6 能力的交付

项目群与其项目的交互界面存在于此。项目群经理将项目的大致情况告诉项目经理，内容包括需要交付哪些内容，什么时候需要交付，以及可能会影响项目的风险和问题。

B.6.1 启动项目（项目简述）

项目群将项目简述提供给项目经理，内容包括对核心交付物的概要描述，风险，问题，报表需求

以及容忍度（允许偏离的最大程度，无需上报至项目群）。

B.6.2 管理和控制交付

项目经理被委派负责项目的交付。但项目经理需要向项目群定期汇报。汇报内容包括项目是否能按计划，在成本范围内完成，尤其重要的是项目期望得到的回报是否能最终获得。

B.6.3 管理风险和解决问题

因为风险和问题在项目的层面被确定，这些将在项目这个层面被管理和解决，除非这些情况在认可的容忍度范围以外。这类问题和风险应该升级至项目群经理处寻求处理。

B.6.4 关闭项目

项目的输出被交付后，项目即可被关闭。变更落地的职责被转移至运行团队。

B.6.5 能帮到什么

- 蓝图级的 **B&IT** 图例将成为这一流程的核心输入，并作为项目简述的一部分告知项目经理
- 这一流程的核心输出将形成详细的图例（**OBASHI** 优化阶段）

B.7 实现回报

业务变更团队将确保项目的输出符合要求，并与项目经理密切配合来确保项目回报是可获取的。

B.7.1 转变前管理

业务变更团队通过沟通、培训甚至对工作职责描述的变化以及角色职责的变化来为这种新的工作模式和系统的变革作准备。

B.7.2 转变管理

项目的输出被嵌入运行环境，因此成为正常业务的一部分。在这样一个困难的转变阶段，运行团队人员需要得到项目群和业务变更团队的支持

B.7.3 转变后管理

回报得到衡量，业务变更团队确保运行在新的工

作方式下持续运作，同时处理和解决出现的问题和风险。

B.7.4 能帮到什么

B&IT 的详细图例可以被用于向运行的干系人解释新的流程是如何运作的，尤其是这种新的工作方式和新的系统能带来哪些提升。这是为干系人准备好变更的一项核心过程。

B.8 关闭项目群

项目群可以被关闭，因为：

- 蓝图被交付
- 回报如期被实现并被期望在项目群关闭以后继续提供回报

如果在项目群以外需要提供进一步的支持，需要在此进行安排。

B.8.1 回顾项目群

对项目群进行回顾，判断项目群是否得到了有效管理，是否在时间和成本范围内交付了期望的成果。

B.8.2 能帮到什么

- **B&IT** 图例可以被用于检验项目群是否交付了要求的未来业务架构
- 一旦项目群被交付，被项目群交付的未来业务架构即成为事实和当前运行状况。这一状况被用于决定未来目标、愿景和机构价值的基础

B.9 项目群后的支持

项目群得到关闭并将交付回报的职责转移至运行团队。这需要高层业务经理或财务领域提供持续的支持以确保在项目群的管理约束消失后，回报还能继续。

B.9.1 能帮到什么

- **OBASHI** 稳定状态
- OBASHI** 图例用于描述当前的运行状态，现已成为正常业务运作的一部分，可以帮助确保新流程遵循既定规则以适当的支持得到保障。

附录 C – OBASHI 如何适用于 PRINCE2

C.1 项目生命周期（PRINCE2）

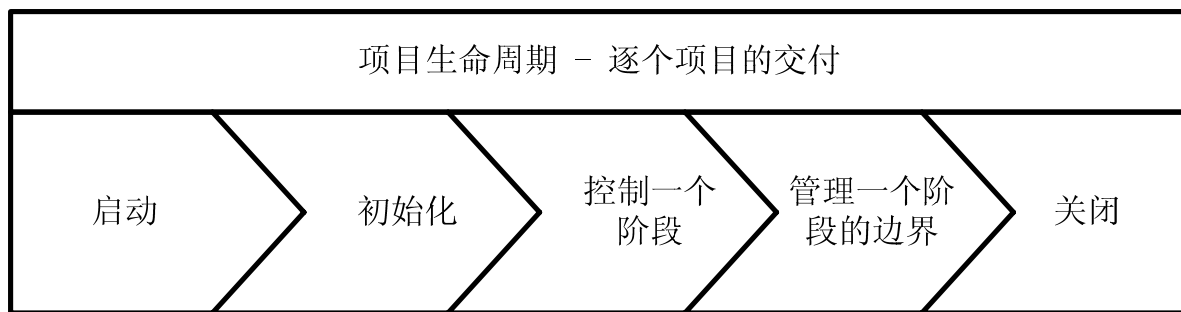


图 C.1 –项目生命周期（PRINCE2）

PRINCE2 生命周期如图 C.1。图中展现了一个项目如何被一个项目任务触发、被发展、被用于认可项目的可行性的信息所充实，经历一系列的阶段来交付要求的产品并最终得到可控的关闭。OBASHI 方法论可以在其生命周期的很多阶段提供支持。

C.2 启动一个项目

C.2.1 项目任务

项目任务由公司或项目群提供。前提条件是他们被授权。‘项目任务’被定义为任何代表用于触发项目的信息。它可以是可行性调研的结果，项目建议邀请书，或者其他一些非正式的方式。项目任务需要提供项目的参考信息，并且至少需要定义谁对项目的授权负责，以及项目管理委员会的行政人员。

C.2.2 OBASHI 能帮到什么

- 高层的 B&IT 图例可以展示数据流的拥有人。从中项目的高层管理权威可以定义最佳项目行政管理人员。

C.2.3 设计和委派项目管理委员会

行政管理和项目经理设计项目管理委员会来代表最终产品的使用人以及为项目提供专项产品的团队。这些将是核心的资源提供者，辅助行政人员进行决策判断。

C.2.4 能帮到什么

- B&IT 图例展现的拥有人通常符合高级用户角色。高级供应商角色可能需要等待项目的方案确定以后才能被明确。

C.2.5 获取以前的经验

原理上，这是指查看以前项目来了解新项目可以从中的到有价值的经验。事实上，公司可能刚开始采纳 PRINCE2，没有经验报告用于回顾。在此情况下，项目经理可能需要与从前项目的项目经理和项目的用户进行沟通和讨论。

C.2.6 能帮到什么

OBASHI B&IT 以及 DAV 图例可以帮助明确将被取代的产品所关联的应用、系统、硬件以及用户，和公司的核心数据流。

C.2.7 准备业务论证的概要

虽然项目管理委员会不会在当前阶段期望一个完整的业务论证，但还是会期望能得到启动项目所需要的是否值得相关投入和开支的支持证据。业务论证的概要应该至少包括项目执行的原因。或许从以前的工作中能得到更多信息，例如该项目是一个项目群的一部分，而该项目群已经得到了全面的业务论证，或者从更早的科研中已经有了部分评估。

C.2.8 OBASHI 能帮到什么

- 一旦项目方案得到确认，OBASHI 可以协助分析新的解决方案与现有的解决解决方案相比，能否降低开支。

C.2.9 选择项目的推进方案

启动项目的任务之一是设计项目的推进方案；用技术的语言描述就是如何将选定的业务目标付诸实现。

C.2.10 OBASHI 能帮到什么

- OBASHI 可以通过图例来描述一个解决方案，也可以以此来显示数据流中存在的问题。它可以显示容量的问题、时间的问题、运行和维护资源的问题、

缺乏产品的问题、其它相关的数据流以及它们的拥有人、需要得到建议和认可的新用户界面。作为选择流程的一个部分，OBASHI 可以帮助对多种项目方案的开支和成本进行比较。

C.3 初始化

C.3.1 优化业务论证

本流程的主要目标是建立对执行项目原因、期望的回报以及相关风险的共识。项目原因应该在项目启动时即以明确。在这个阶段，项目管理委员会期望能看到预计的回报和金钱为衡量单位的开支节省。

部分工作是建立回报评估计划用以展现回报的发生时间点以及如何对回报进行衡量。大多数的回报发生时间在最终产品交付使用了一段时间以后。用以衡量回报的内容可以包括：‘我们是否将订单处理的时间从 3 天缩短为 1 天？’；‘我们是否有效的将维护人员的数量缩减了 4 名？’；或‘我们是否将每个订单的收益增加了 25%？’

C.3.2 能帮到什么

- OBASHI 可以协助分析新的解决方案与现有的解决解决方案相比，能否降低开支。
- OBASHI 可以为新数据流的成本开始计算提供帮助。

C.3.3 准备风险管理的战略

项目需要有如何应对风险的战略。作为活动的一个部分，我们需要创建风险计数器。

C.3.4 OBASHI 能帮到什么

- 项目中数据流的 B&IT 以及 DAV 图例可能显示风险，例如，与其它拥有人之间存在不同的兴趣点，缺少人员和设备的冗余或者来自于其它数据流的变更可能会影响本项目所用到的应用、硬件、基础架构或资源
- OBASHI 还能给风险影响分析提供帮助，查看各类建议和设想与已计划的数据流，或其它数据流是否相互影响。因为项目的业务论证中可能包含的对资源的精简会给另外一个数据流带来风险。

C.3.5 准备质量管理战略

项目初始化阶段需要定义质量标准以确保用户对项目的质量期望以及如何对质量进行评估。

C.3.6 能帮到什么

- 对 OBASHI 图例的回顾可以使项目经理理解与项目相关的应用、系统、硬件和基础架构的标准，以及一些程序标准和所需要的培训。

C.3.7 准备沟通管理战略

该文档包含了项目管理团队如何与机构范围内与参与项目及项目相关人员的信息互通的细节。其中包括项目干系人。

C.3.8 能帮到什么

- OBASHI B&IT 及 DAV 图例可以显示项目相关干系人。

C.3.9 建立项目计划

在对项目的重大开支做承诺之前，需要确定项目的时间框架和资源需求。为了建立项目计划，项目经理必须了解项目要交付什么，包括任何先决条件、外部的依赖、限制和假设。项目的执行方法也必须清楚。

C.3.10 能帮到什么

- 建议方案中的 B&IT 图例可以提供先决条件、外部的依赖的信息，例如，属于其它数据流的或拥有人的应用、以及限制和假设的信息，例如，时间、可用性、计划中的数据流的更改，以及应用、系统、硬件和基础架构的信息。

C.4 阶段的控制

该部分覆盖了项目经理在管理项目阶段的活动：

- 分派工作
- 监控性能表现
- 检查新的风险或问题
- 留意质量的结果

管理产品交付的流程也可以被视作控制阶段活动的一部分。管理产品交付覆盖了项目专属产品的开发。该项工作通常由团队经理负责。

C.4.1 OBASHI 能帮到什么

- 在整个流程中，OBASHI 可以帮助完成风险影响分析、评估方案建议、问题影响分析和产品开发
- 在创建工作包时，OBASHI 可以帮助确认运行和维护的节点

C.4.2 授权工作包（Work Package）

(以及认可工作包)

让项目的实施人员主观选择活动的开始会形成混乱局面。因此,通过项目经理来授权工作的开始和继续非常重要。为了达到这个效果,需要创建工作包。

C.4.3 OBASHI 能帮到什么

- OBASHI 图例可以帮助理解各项工作在整个业务场景所处的位置,并在开发产品时提供界面信息以及工作包交付产品的运行和维护界面信息。
- OBASHI 可以提供关于风险和问题的信息。团队计划和解决的解决方案可以与 OBASHI 的信息进行对比和判断合理性,尤其是有关哪些应用、硬件和基础架构会受到影响。

C.4.4 截取和检查问题及风险

在控制一个阶段或执行一项工作包时,不同的问题和风险会出现。需要对这些问题和风险有一致的、可靠的办法。在对行动过程做决策前,需要对每一个问题和风险进行记录并进行影响度评估—不光只是问题或风险的影响度,也包括出现各种反映的影响度。

C.4.5 OBASHI 能帮到什么

- 可以与建议的数据流进行对比来判断问题和风险的影响度
- 所有建议方案需要与建议的数据流以及其它相关数据流进行比对以判断影响度

C.5 管理阶段的边界

管理阶段边界流程的目标是使项目经理有足够的信息提供给项目委员会来回顾:

- 当前阶段成功与否
- 更新的项目计划
- 更新的业务论证
- 风险情况
- 授权下一阶段计划

项目并非永远会进入计划。一个意外情况可能意味着该阶段超出了一项或多项的容忍范围。该情况汇报至项目管理委员会后,项目管理委员会可能会要求对该阶段的剩余内容重新计划并将计划按标准阶段计划的要求提交至项目管理委员会审批。

C.5.1 OBASHI 能帮到什么

- OBASHI 可以提供部分相关的干系人的信息
- OBASHI 可以提供关于设施或资源及测试要求何时可以到位的时间信息

C.5.2 计划下一阶段

下一管理阶段的阶段计划在当前阶段的末尾建立。这不是一项孤立的过程。项目经理需要与项目管理委员会、项目质量保障、团队经理及其它干系人进行协商以获得可行的计划。

C.5.3 OBASHI 能帮到什么

- 因为这是比较具体的计划,OBASHI 图例可以显示部分产品因为过小,不适合放在等待调整的项目计划中
- 在开发新产品时,OBASHI 可以帮助确定产品间的依赖关系并用于撰写产品的描述信息
- 对新产品,OBASHI 可以提供对该新产品接口的信息用于开发以及运行和维护
- OBASHI 图例可以帮助确定下一个阶段产品的干系人。这些干系人对数据流的使用会与这些产品发生关联或影响

C.5.4 更新项目计划

项目管理团队利用项目计划以管理和衡量项目的进展。

项目计划根据实际的阶段结束数据和日期以及预告的项目周期和下阶段项目计划的开支进行更新。

C.5.5 OBASHI 能帮到什么

- 可能需要咨询干系人。大量这些咨询的信息可以从检查 OBASHI 图例中获取
- 整体画面中新增加的数据流可能会影响项目的推进方案或引进新的干系人
- 对数据流的变更可能导致项目新风险或改变已有风险
- 对数据流的变更可能导致对一项或多项风险容忍度的修改

C.5.6 更新业务论证

根据项目必须符合业务论证需求这一原则,项目委员会必须对业务论证进行回顾和更新来确保项目的长期有效性。因此,需根据更新的项目计划中的开支信息来更新业务论证。

同时,需对预告的项目收益进行回顾,检查是否

有相关的改变。

收益回顾计划应被检查是否发生了变化。

C.5.7 OBASHI 能帮到什么

- OBASHI 提供了数据流及其资源发生变化的信息，这些变化可能导致项目最终输出物的收益产生
- 如果新增加了数据流，这些数据流可能影响成本或资源
- OBASHI 可以帮助确认预估的最终项目输出物运行成本没有发生变化

C.6 关闭项目

- 关闭项目流程的目标是：
- 验证用户对项目产品的认可
- 确保当项目解散时，用户方有能力支持项目的输出物
- 以基线为基准，对项目性能进行检查（项目启动文档）
- 评估已经取得的任何收益
- 更新其它预告的收益并检查收益回顾计划中的细节是否得到了更新
- 确保对未解决问题和风险有相应的处理方式，通常作为后续行动建议进行记录

C.6.1 OBASHI 能帮到什么

- OBASHI 图例可以展示在哪里、与谁进行回顾以确保项的输出物可以被运行和维护人员接手。

C.6.2 准备未成熟关闭

在某种场合下，项目委员会可能会要求项目经理对项目进行未成熟关闭。在此情况下项目经理必须确保正在进行的工作不是简单被抛弃，而是确保项目的价值是否尽可能得到保留并将与最终项目目标的差距汇报至机构或项目群管理团队。

C.6.3 OBASHI 能帮到什么

未结束的项成果可以与 OBASHI 图例进行对比来展现成果当前的位置，并提供对损失影响的评估。

C.6.4 移交成果

在项目关闭之前，需要将项目成果交付至运行和维护环境。可能需要对收益回顾计划进行更新以包含对成果在运行环境使用情况的回顾。回顾会展示是否有衍生作用（正面或负面）的产生，这些作用或许可以为其它项目提供有用的参考。

准备后续的行动建议以应对未完成的工作、问题或风险。

C.6.5 OBASHI 能帮到什么

- OBASHI 图例可以展示运行环境
- OBASHI 展示的运行环境可以帮助确定是否需要对项目成果的运行和使用进行回顾，以确定是否存在无法预计的衍生效果
- OBASHI 图例可以展示运行和维护的目标资源，以及将收到后续行动建议的数据流拥有人信息。

附录 D – OBASHI 如何适用于 ITIL

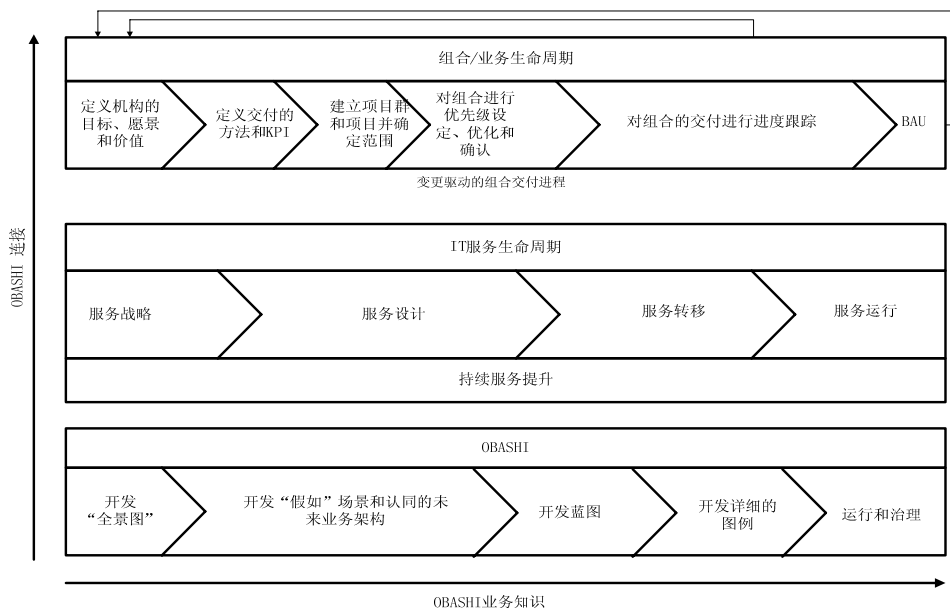


图 D.1 – OBASHI 和 ITIL

ITIL 服务生命周期如图 D.1. OBASHI 方法论可以支持 ITIL 流程整个生命周期的实施。

D.1 服务战略

服务战略是关于将 IT 服务管理转变为一种战略资产，将服务组合中的服务价值最大化、将风险最小化。服务战略回答以下问题：

- 我们应该提供哪些服务给谁
- 我们如何与我们的竞争对手区分
- 我们如何为我们的客户提供价值
- 我们如何为战略的投资进行论证
- 我们如何从不同的服务质量提升路径中去选择
- 我们如何在组合的说有服务中有效的分配资源
- 我们如何解决对共享资源需求的冲突

ITIL 服务战略 1.3

D.1.1 OBASHI 能帮到什么

对成功完成服务战略的最基本要求是被 IT 所管理的，支持业务流程、业务战略和业务目标的不同的服务、系统和进程的详细了解。OBASHI 提供了对映射 IT 基础架构和业务流程及组织机构之间关联关系和依赖关系的强大的方法论。

D.1.2 流程：服务组合管理

服务组合管理被描述为：

“...治理整个企业对服务管理的投资并对其价值的提供进行管理”

ITIL 服务战略 5.3

目标是维护一组服务，最大化 IT 的投资回报，同时最小化风险。

D.1.3 OBASHI 能帮到什么

- OBASHI 提供了记录服务组合的方法论。该方法论提供了对 IT 基础架构与业务流程和组织架构之间建立关联关系和依赖关系的映射的方法。
- B&IT 及 DAV 视图提供了 IT 基础架构、系统和应用与业务流程之间的详细关联关系信息
- 所有的组合数据（例如，财务、基础架构、服务链、业务论证数据）在 OBASHI 中建立模型
- OBASHI 图例记录了服务模型
- OBASHI 是动态的方法论– B&IT 图例可以很方便的被更新用于反映变更或用于测试和验证建议的未来状态。

D.1.4 流程：财务管理

财务管理：

“...为业务和 IT 提供了 IT 服务的财务价值论证、提供的服务之下的资产价值，以及运行预告的合理性论证”

ITIL 服务战略 5.1

换一种说法，为了给是否将某项服务保留于服务组合中或添加一项新的服务至服务组合中提供可靠的战略决策依据，我们需要知道这项服务能产生的价值和产生过程中的开支。在任何规模与复杂度企业中的，有大量共享资源的 IT 基础架构中，

这项工作是非常有挑战的。

D.1.5 OBASHI 能帮到什么

- B&IT 图例提供了对基础建构是如何支持服务的理解。直接将硬件、软件和网络部件与使用它们的服务进行连接，提供在服务及构件层级上对开支/价值的评估
- 所有的财务数据（例如，提供的价值、服务的预期价值）可以通过 OBASHI 建模，模型可以直至基础环境的构件级，并与业务流程关联。

D.1.6 流程：需求管理

需求管理是关于对实际和预期的业务对服务的需求的理解，以预估服务的性能和开支情况，从而得出合适的决策。

D.1.7 OBASHI 能帮到什么

OBASHI DAV 图例在帮助确定关键路径中的部件，并确保这些部件不被忽略起了重要作用。这可以使得服务需求引发的变更影响可以被准确的进行建模和评估。

D.2 服务设计

服务设计的主要目标是设计新建或变更的服务。其目的可以被归纳为在最高效的形式下，通过对现有构件和服务的重复利用，执行一项服务设计，以满足业务的要求，规避对已有服务的负面影响。

D.2.1 OBASHI 能帮到什么

与服务战略一样，为了获得成功的服务设计，对已有服务、系统和其支持的业务流程、业务战略和业务目标的关系的理解非常重要。

D.2.2 流程：服务级别管理

服务级别管理的目标：

“...确保双方共同认可的 IT 服务级别被提供至所有当前的 IT 服务，并且未来的服务也将按双方认可的可达到的目标要求提供”

服务设计 4.2.1

D.2.3 OBASHI 能帮到什么

- 对 B&IT 图例中包含的已有服务和提供者了解可以帮助确定在设计新服务时的复用的机会，同时将风险标示出来（例如，对某第三方服务提供者的过度依赖，对目标的设定与现有服务冲突等）
- B&IT 图例将是服务设计包的核心部件，提供了下

游阶段（服务设计和服务运行）需要的完整的基础架构模型。

D.2.4 流程：可用性管理和能力管理

这两项流程确保在有效的成本约束下基础架构能提供需要的服务可用性和性能。

D.2.5 OBASHI 能帮到什么

- OBASHI B&IT 图例提供了清晰的对业务服务和 IT 基础构建的映射关系。可以帮忙对以下内容进行定义和准确的评估：
 - 部件故障对业务运行和用户的影响
 - 可能影响可用性的单点故障
 - 构件和人员的依赖关系
 - 构件和服务的恢复定时
 - 对可用资源的不合理分派
 - 低利用和过度利用的构件
 - 基础架构中对数据流路径优化的可能
- 技术规格（例如，性能阈值、预估评价故障周期）可以在构件层面进行定义以支持对预估的构件和服务性能的建模。
- 定义过度使用的某项资源的风险和可能的业务影响

D.2.6 流程：IT 服务连续性管理

IT 服务连续性管理的目标：

“...通过确保 IT 技术和服务设施可以在要求的，认可的时间区间被恢复以支持整体业务连续性管理流程”

服务设计 4.5.1

是关于业务认为已足够重大到灾难级的事件，IT 服务连续性管理是在业务连续性管理的整体框架中执行的。一项非常重要的要点是 IT 服务连续性管理经理必须清楚每一项服务的失去对业务带来的影响以及在今后一段时间内的影响；对运行连续性可能存在的风险。

需定义和实施适合的应对方案用于降低风险，包括建立灾难恢复计划，在可接受的时间范围内恢复核心服务的能力。

OBASHI 中建立的 B&IT 及 DAV 图例可以帮助为 IT 服务连续性经理的努力提供关键的信息。

D.2.7 OBASHI 能帮到什么

- B&IT 及 DAV 图例清晰的对已有基础架构是如何支持业务流程进行了映射，定义了：
- 关键部件
- 漏洞
- 支持准确的风险评估，以对风险进行量化并对实施应对措施进行论证。并且提供给非技术人员以图形化的现实效果。
- OBASHI 建模技术也可用于对在出现灾难以后，对支持关键数据流的最基础灾难恢复环境进行建模。

D.2.8 流程：信息安全管理

与连续性管理一致，信息安全管理在更广泛的业务安全框架中运作。

信息安全管理的目标是建立和强化安全策略，该安全策略需与业务安全保持一致。目的是保护机构的数据。至于连续性管理，这包含了对业务需求的理解；基于每一项服务去理解一项安全故障和违规可能带来的业务影响；在细节上理解基础架构在安全威胁上的漏洞。制定出合适的应对方案来降低风险。OBASHI 中的 B&IT 及 DAV 清晰的描绘了数据是如何在机构和基础架构的流转过程，这将提供 IT 安全管理努力的基础知识和信息。

D.2.9 OBASHI 能帮到什么

- 为了保护数据，我们需要知道数据是如何建立和存放的，数据是如何在基础架构中流转的，以及这些基础构件的安全程度。OBASHI 数据流图例提供了这些内容
- 技术规格可以在构件层来帮助定义可以复用的构件
- B&IT 及 DAV 图例可以被用于确保安全的需求被覆盖到所有的支持构件中
- B&IT 及 DAV 图例还可以显示管理和发送业务数据的第三方角色，标示出额外的安全顾虑。

D.2.10 流程：服务目录管理

服务目录管理的目标是建立和维护服务目录。两项目录的视角得到维护 - 业务视角和技术视角。

D.2.11 OBASHI 能帮到什么

- OBASHI 记录基础架构于 B&IT 图例的方法可以产生大量详细的关于服务和基础架构的知识。这些知识被用于建立服务目录
- B&IT 图例的子集可以包含于服务目录中用于展示

服务/基础架构端到端的映射关系。例如，业务部门的主要兴趣点在头层，而服务的供应者应该关心所有的层面。

D.2.12 流程：供应商管理

供应商管理的目标是确保当启用第三方供应商时，他们的价值和风险在严密的管理中。

D.2.13 OBASHI 能帮到什么

- OBASHI 图例清晰的定义了第三方供应商提供和支持的构件和服务，以及服务或构件在失去时的业务影响
- 可以帮助确定无法接受的风险（对某一供应商过度的依赖）或可能带来开支节省的整合机会。

D.3 服务转移

服务转移是关于变更的计划和管理，确保

“服务设计中包含的服务战略需求被有效的在服务运行中实现，同时管控失败和中断”

服务转移 1.2.3.3

服务转移是关于将基础架构从一个已知的状态转移至另外一个状态，在投入生产环境前，确保变更被记录、测试和验证。OBASHI 中的‘现状’和‘未来’可以在此生命周期的阶段其重要作用。

D.3.1 流程：知识管理和配置管理

这两个流程提供了 ITIL 的核心知识系统-服务知识管理系统及其支柱，配置管理系统。这些系统的质量和准确性将是决定生命周期中所有 ITIL 有效性的要素。

D.3.2 OBASHI 能帮到什么

OBASHI 是一个方法论。以 ITIL 的语言描述，是建立配置模型的方法论。ITIL 在更高的层面描述了配置管理系统（CMS）应该包含哪些内容，该如何定义结构。OBASHI 定义了详细的过程描述来指导服务、基础架构以及关联关系数据的收集、记录和组织。

D.3.3 流程：变更管理，发布和部署管理，校验和测试/评估

这些是转移管理套件中的组成，用来确保上线前 IT 服务及基础环境的任何变更被记录、授权、计划、测试和评估。目的是确保最大化正面收益和最小化负面影响。

D3.4 OBASHI 能帮到什么

上述核心流程依赖于 CMS 系统的信息质量。

OBASHI 方法论可以确保：

- B&IT 图例通过清晰定义构件和业务流的依赖关系来保障准确的分析和成本评估

B&IT 图例可以帮助定义一项变更可能影响到的干系人

DAV 图例可以帮助确定变更对下游的影响，因为数据流可以通过大量流程和应用进行跟踪

D.4 服务运行

服务运行的目标是按设计要求，满足服务级别的要求提供服务。服务运行是关于管理日常事件，突发时间和问题以及服务请求，OBASHI 可以为这些内容提供帮助。

D.4.1 流程：事件、突发事件和问题管理

这些流程对应系统的监控、对服务中断的响应以及在适当的时机分析问题根源并找到解决方案以防止故障的再次出现。

D.4.2 OBASHI 能帮到什么

- B&IT 及 DAV 图例能在构件出现故障时提供帮助，借助构件/服务/数据流的相互依赖关系进行准确的影响分析；例如，一旦一个构件出现问题，哪个数据流会终止，哪个服务会受到影响，影响的程度有多大。
- DAV 图例可以帮助定义出现故障后可能存在的‘下游’影响，否则这种影响通常无法即刻被察觉
- B&IT 的拥有人层级可以帮助在出现影响服务的事件或问题时找到干系人。

技术规格/事件阈值可以存放于构件层，用于帮助对事件进行分类并定义适当的响应办法（确定是属于警告或是严重告警）。

D.4.3 访问管理

访问管理负责授权或剥夺系统和服务的访问权力

和许可。它负责将安全策略的规则得到执行。

D.4.4 OBASHI 能帮到什么

如上所述，B&IT 以及 DAV 图例可以被用于确保在服务层级上安全的要求被传播至所有的下级支撑构件。访问管理可以在校验服务或系统的访问请求时利用这项知识。

D.4.5 流程：请求履行

该流程负责为用户提供快速有效的标准服务，减少官僚的情形和成本。与其他许多 ITIL 流程一样，这个流程依赖于配置管理中信息的质量，以及定义哪些服务可以成为标准服务并且建立请求履行的模型。

D.4.6 OBASHI 能帮到什么

通过 OBASHI 对服务基础构件的详细了解可以帮助确定原本分散的服务构件。

D.5 持续服务提升

“CSI 的核心目的是持续将 IT 服务与不断变化的业务需求进行关联和保持一致，方法是通过确定和实施支持业务的 IT 服务的改善方案”

持续服务提升 2.4.1

持续服务提升与其他生命周期的阶段不同。他需要在整个生命周期中持续被选用，通过不断的提升服务战略、服务设计、服务转移和服务运行的能力以确保 IT 持续为它的用户产生和维护价值。

CSI 方法围绕建立一个基线，将提升付诸实现以及衡量效果并与基线进行比对。最终，所有的提升被与业务影响进行衡量-以是否跟好的支持了业务目标或降低了成本为条件。

OBASHI 图例提供的清晰的业务流程与 IT 基础架构的映射关系可以帮助所有生命周期阶段的经理、服务的拥有人以及流程拥有人确定强化服务质量或降低成本或风险的机会。

附录 E—案例学习

前言

案例学习 1

合并或收购的管理

一家汽车制造公司名叫 CARCORP。这家公司收购了另外一家公司叫 FABRITE，目标是扩大产品组合，减少运行成本、将表现不佳的 CARCORP 业务进行转变，提升其市场定位。

案例学习 1 被大量选摘于第 4 章节，用于显示 OBASHI 生命周期是如何适用于并购和收购案例的。

案例学习 2

影响分析/风险分析—业务经理的视角

一家大量依赖 IT 以交付产品和服务的机构正在考虑扩展本地和全国的运营。对扩展运营的影响和风险进行评估变得非常重要。

案例学习 2 显示了 OBASHI 是如何帮到这项流程的。

案例学习 3

基于业务的考虑，重新定位 IT 成本开支

一项业务正通过将每个部门转变为业务部门来进行改组。改组后负责 IT 服务的业务部门经理需要衡量和管理 IT 资源的质量，并且建立提套收费机制来对被使用的 IT 服务进行收费。如果没有对 IT 资源使用情况进行清晰的展现，其它业务部门就可以趁机利用这种模糊性。

案例学习 3 探讨了如何通过 OBASHI 来消除这种模糊性。

案例学习 4

管理并降低外部咨询成本

存在降低逐步扩大的外部业务和 IT 咨询成本的需要。如果不清楚当前的活动，而尝试去对涉及外部咨询师介入之项目基于成本/价值去定义项目的重要级别是非常困难的。对项目与核心业务、项目与项目之间的影响度和依赖关系进行量化会变得非常耗时并且困难。

案例学习 4 讨论了 OBASHI 是如何建立一个业务知识库，为项目群和项目建立基线提供基础。

案例学习 1：合并或收购的管理

E.CS1.1 场景

一家名为 **CACORP** 的汽车制造企业收购了另外一家名为 **FABRITE** 的汽车厂。目的是扩充产品线、降低运行成本，并以此对表现不佳的业务进行转变以期提升公司的市场定位。

FABRITE 拥有一系列的设计和制造工厂，可以提供车辆的模型制造和部件制造。

为了达到降低运行成本的目的，**FABRITE** 必须能很好的融合到 **CARCORP** 中。重叠的人员、流程和技术需要得到整合。

您的任务是评估这两个业务及其运行的情况，提出整合的战略方针。

E.CS1.2 问题

两个公司都有自己的业务流程和 IT 系统，并且这些流程和系统已经在运行中进化了很多年。

FABRITE 的重心在为大量企业用户提供制造的专业服务，而 **CARCORP** 的工作焦点是为普罗大众生产产品。

因此，两家公司的战略、文化、愿景和目标都有很大差异。但是，它们之间也有部分业务是重叠的，其中包括 **HR**、采购、物流、设计和生产。

它们各自都有业务活动、产品、系统、资产、客户和项目的管理组合。这些管理组合需要得到有效的协同。明确以上内容在哪家公司的更强是优化和强化 **CARCORP** 的核心要素。

没有这种协同的机制，**CARCORP** 无法负担同时运作两家公司。

E.CS1.2.1 OBASHI 如何帮助解决这些问题

为了使 **FABRITE** 能被融合至 **CARCORP**，**CARCORP** 需要定义其长期目标及业务战略，同时考虑战略和运行的优先级。其中一项核心的输入是一个或多个高层次的 **B&IT** 图例，展示出 **CARCORP** 和 **FABRITE** 当前的运作情况（‘big picture’）。

B&IT 的 big picture 可以帮助确定 **CARCORP** 的目标、愿景和价值，帮助了解两家公司的业务运作

情况。

CARCORP 还需要明确如何通过实施新的业务模式来达到既定的愿景和战略，来通过 **KPI** 去衡量成功的程度。是否能对成果进行衡量是判断两个公司整合效果的核心要素。

人员、流程和技术之间的关系形成了业务战略的核心。**OBASHI B&IT** 图例记录了数据流、业务所有人和业务流程以及用于整合 **FABRITE** 及 **CARCORP** 的支撑 **IT**。这为决策提供了业务方面的知识。

B&IT 图例的使用还能通过对可行性及可选项（‘what if’）的查看为交付战略的定义提供支持。未来状态图例可以用于展示 **CARCORP** 和 **FABRITE** 在未来如何运作。变更管理可以被整合到这个流程中用于明确对人力支援的影响，用于管理部门员工迁移的战略。这也包含了对未来业务状态的文化及行为模式的分析。

在完成对新公司和运行模式的定义后，**CARCORP** 需要确定通过那些项目群和项目来达到公司的目标。另外，还需要对项目群和项目的范围进行定义，目标是理解系统级和业务流程级的借口和依赖关系。

作为‘big picture’及‘what if’的一个组成部分，**OBASHI B&IT** 图例可以被用于确定那些工作可以在逻辑上被归入一个项目或项目群，以及定义其中的接口和相互依赖关系。变更管理分析可以确保培训及企业文化不会变成新运行模式的障碍。

因为采用了这种方式，**CARCORP** 现在已经有了 **B&IT** 图例的图库来确保这些核心业务知识能被用于在任何阶段为决策的执行提供支持。

CARCORP 需要对项目群和项目进行优先级设定以确保这些项目与战略驱动力的最佳吻合同时须投入最少的风险。通过使用 **OBASHI** 方法论可以帮助确保提交的最终的业务解决方案与已部署的 **IT** 资产有较好的融合度，使得在项目在推广时不会有返工。

B&IT 图例可以帮助 **CARCORP** 来评估：

- 核心思想以及这些思想是如何与战略进行吻合的
(通过了解交付物的重要意义以及对新业务架构的支持)
- 交付的成本开支
- 组合管理中的内在风险
- 机构管理和交付项目的能力
- 可以对项目组合设定有限级的干系人

CARCORP 现在已进入将项目群的详细定义环节, 已经可以定义未来运行的蓝图。这里包含对当前业务状况的 ‘big picture’ 以及约定的未来业务架

构进行详细的定义, 并了解某一特定的部门是如何进入协同状态的。

可以启动项目以为新的运行模式定义出更加详细的流程。底层的 B&IT 图例的输出可以帮助展示整合后的 CARCORP 和 FABRITE 是如何工作的。另外, ‘big picture’ B&IT 图例需要得到更新以反映当前的运行状态和现实。这些可以帮助找到新项目的机会甚至是机构的战略调整, 因为我们已经证明了部分战略调整给 CARCORP 带来的收益。

案例学习 2: 影响/风险分析-业务经理

您是一家机构的业务经理,大量依赖 IT 来交付产品和服务。机构的规模在增加,董事会在考虑扩展本地和全国的运行规模。您被要求提供一份报告,报告的内容包括在当前业务运行状况下进行扩展的影响和风险。

E.CS2.1 问题

为了完成分析并提交报告,您需要与经理们在全国不同的地域进行沟通。考虑可能需要若干个月的时间。但是,你也意识到为了准确的分析公司扩展的影响和风险您需要充分理解个地区所有的核心业务流程以及相关的工作流。

E.CS2.2 当前的做法是怎么样的?

您需要快速的理解建议中扩展方案的物理内容。例如,不动产、物流、人力资源和仓储。与您沟通的每一个业务经理会给你非常准确的日常工作流程的理解。但是,在期望清晰和准确的得到对工作流是如何通过数据在 IT 系统流转而得到支撑时问题就出现了。

在理解数据流程对机构的重要性以后,了解支持这些数据流的所有部件就变得非常关键了。取决于相关的业务工作流和流程,任何数据流的中断将给业务运行带来不同程度的影响和风险。您非常清楚每一个 IT 构件,例如,路由器、服务器、交换机和应用,可以为很多业务流程服务。这些业务流程同时为企业不同的工作流服务。然而,准确的分析和汇报支持这些工作流的 IT 资源或资源组的实效和停机时间非常困难。

您所在机构或许足够成熟以致有资产管理工具或网管工具或甚至有业务流程管理应用来帮助提示整个企业 IT 资源的分布情况。但是,通常您需要依赖 IT 部门的高层人士来帮助理解‘什么完成什么’。即使如此,他们通常只能告诉你哪个部门哪些资源,但并不清楚工作流的准确细节。

您还是处在无法对业务流程的影响和风险进行可视化和建模状态-没有对业务流程与 IT 资源相关关系的清晰视图。

通常,IT 系统被部署和服务于一个特定功能需求。没有对当前部署与已有业务流程关系的清晰视图。一旦业务职能发生扩充或改变,常见的 IT 讨论结果是安装更多的系统以符合新的业务结构。这将带来更多的支持、维护和安装成本。这通常有悖于业务战略。

E.CS2.1 OBASHI 如何帮助解决这些问题

您与 IT 经理举行会议。他带来了一沓 OBASHI B&IT 图例。这些图例简单、清晰的展示了机构的业务资源是如何被部署和支撑业务活动的。

头一次,您可以将 IT 资产如何支持业务流程进行可视化:人力资源、硬件、应用、系统和网络基础架构。

IT 经理介绍说 OBASHI 使得这些资产以及数据流可以与业务流程以及他们所有人进行捆绑以获得 IT 开支的 What, where, why。

可以从原始模型中建立扩展方案的混合模型。可以对 IT 资源进行模拟,用于准确的评估企业中每一项业务流程的影响和风险。

OBASHI B&IT 可以帮助形成强大的成本、影响和风险的模拟模型。这将大大提升您对每一个地区与业务扩展相关的核心业务流程及相关工作流的理解。因为 OBASHI 模型的可伸缩性,您可以针对跨地域的单个、群体资产进行风险评估汇报。

通过 OBASHI 的使用,差旅和调查的费用大大得到缩减,并且可以在更短的时间交付高准确度的汇报材料。

案例学习 3: 在业务范围内调整 IT 成本开支

您是 IT 经理。董事会决定调整业务结构，把每个部门转变为业务单位。每个业务单位的经理的目标和职责变为以季度为单位确保获得利润或降低损失，并将这些报告汇报至董事会。

E.CS3.1 问题

作为 IT 服务的业务单位经理，你需要衡量和管理将多少 IT 资源投入至每个部门中的每项业务流程。并且，建立一项收费的机制，以季度为单位对每一个业务部门基于其对 IT 的消费进行收费。

E.CS3.2 当前的实践方式？

从传统来看，IT 是被当做成本中心的。

IT 经理每年获得预算并将预算分摊至已有的 IT 服务。一旦成本的分摊模型已经确定，这项管理工作是相对直接明了的。因为如果预算金额和结余都正确的话，预算的分摊和成本核算的控制不是那么繁重。

然而，一旦转变为基于盈余和损失系统的业务部门，我们将面临更加复杂的挑战。首先，准确的计算出 IT 的成本，全部或者分摊至不同的业务，是一项非常困难的工作。其次，从不同业务部门的经理处得到双方认同的收费模式是非常耗费时间的。每一个人都将为各自所在的角色而坚持己

见。想要获得一致的意见很少能比较顺利-如果没有非常明晰的 IT 资源使用统计，聪明的经理们会利用其不明确性来保护自己。

IT 经理的挑战是如何准确的可视化，解释及沟通 IT 的资源使用，使得其它业务单位经理能理解为什么他们需要为使用的 IT 资源付费。

E.CS3.2.1 OBASHI 是如何解决问题的

管理 IT 业务单位的核心成功要素是充分了解 IT 的人员和技术资源是如何支持并关联到业务流程的。

通过部署 OBASHI B&IT，你得到了清晰，简单的机构逻辑图。这可以帮助你可视化及沟通复杂的业务和 IT 的关系。

拥有这个视图和对结构的理解是至关重要的。因为只有这样，你才可能准确的将成本或成本的百分比归口至具体的资源实例中。然后，OBASHI 可以帮助完成建立基于这些成本的完整的财务模型，确保真实的成本分摊建模和分析的发生。

通过将这些视图提供给业务单位的同事，你可以消除 IT 资源利用情况的的不确定性。另外，他们还能看到提高效率节省成本开支的机会。

案例学习 4：降低外部咨询成本在业务范围内调整 IT 成本开支

你是项目群经理。董事会点明了降低日益增长的机构外部业务及 IT 咨询方面的成本开支。行政管理层对当前业务正在介入的项目及项目群组合没有清晰的视图。你被要求确定所有的涉及外部咨询资源的项目及项目群，并根据投入产出比明确每个项目的必要性。董事会还想知道项目及项目群对核心业务活动及它们之间的相互影响和依赖关系。

E.CS4.1 问题

通过与行政管理层，业务单位经理以及高级部门经理的面谈，一些情况已经变得非常清晰。业务知识方面存在很大的差距。

初始的发现已经确定没有一个人或一组人或文档或系统能提供过去或当前的，跨不同业务单位的项目或项目群的清晰视图。更关键的是对外部咨询师的绩效或项目交付物没有衡量，监督和分析来与业务战略进行比对。因此，投入产出比，项目范围的重叠以及双重支付的定位等无从谈起。

经过更进一步的调查，发现这些状况的发生和持续的根本原因是多年来各个业务孤岛内部的技术构成在不断增加，但对这些技术的从整个业务来看的战略重要性，如果存在的话，没有充分的认识。

这些技术的拥有者和治理管控从来没有在战略上得到管理，因此，一项准确的成本运行模型是欠缺的。

外部咨询师始终在部门内部按照部门的意图完成功能性的项目，因此是与战略目标隔离的工作。他们没有清晰的整体业务，而且从财务角度来看，他们也没有建立这种战略视角的兴趣。

E.CS4.1.1 OBASHI 如何帮助问题的解决

从哪里开始是最大的挑战。这里不存在‘big picture’或者‘small picture’。

在利用 OBASHI 方法论时，最开始的步骤是建立业务的图形，这将给你提供基本的框架。在此之上你可以规划进行中的咨询活动。从此，你可以开始获得内部部门及干系人是如何利用第三方咨询师的清晰视图。

在明确了项目群的活动以后，每一个项目的启动原因即可收集获得，同时还能得到项目的范围及交付物的描述。各项目中人员、流程、技术资源及资产的成本得以核算并与项目的预算进行比对。通过对各项目驱动力的考虑，其中包括核心运行、战略、支持和预见的收益，来完成对项目当前以及未来价值的评估。这可以使得每个项目对业务的预期贡献可以通过财务、战术和战略的语汇进行计算。

项目因此可以被定义为标杆并进行业务关键性的比对。通过利用 OBASHI 方法论，项目与项目的范围是否重叠，项目的依赖性如何等也得以检验，给第三方咨询服务的利用率增加透明度。

附录 F – OBASHI&企业架构

到关联的业务对象以获取相关数据。

F.1 概览

企业架构(EA)可以被分割为三部分：

1. 静态模型：Zachman 是一种结构化的框架，可以静态的对人工产品(artifacts)进行分类。
2. 动态模型：TOGAF 是一项流程模型，它是动态的，需要通过配置来适用于每项实施。
3. 混合模型：DoDAF 及推导出来的框架用于覆盖某纵向市场特殊兴趣点的人工产品。

选择上述任何一项框架需要得到业务的认同以期成功。

虽然企业架构框架为理解现代业务的运作做出了一定贡献，但是复杂性仍然存在。

企业架构是一项持续的流程，不是一锤子工程。因此，需要得到业务的有效承诺。为了成功，初始的投入是需要的。不光是资金，也包括机构的愿望。通常，其他机构目标会得到更高的优先级，因为企业架构有时为被视作额外开销。

什么能成为一个轻量级的框架，这种框架更易于实现，更能基于需要将焦点集中于某一特定业务范围，能更快的为机构得到收益-也即被称为‘低垂水果(low hanging fruits)’，已成熟可以随时被摘取。

这既是 OBASHI 能带来的。

F.1.1 OBASHI 能为企业架构带来什么？

通过 OBASHI，用户可以快速确定某项特定业务目标，例如‘发票打印’的后台支撑系统。同时，通过对数据流的了解，‘发票打印’的上游及下游中任何应用、系统、硬件或基础架构都可以得到分析。

因为 OBASHI 用户在建立模型时可以任意选择颗粒度，同时模型可以包含物理、逻辑、或同时包含两种情况。用户可以将内容项与地理位置进行关联，以显示它们的分布情况（地理位置、文化、业务单位、盈利中心、业务计划、用户分类等）

用户可以看到哪些逻辑活动将他们的业务流程与他们的应用联系起来。他们还能在这些活动中看

这意味着 OBASHI 的用户可以从组合中获取与一组逻辑活动关联的应用列表，并基于功能特点识别出重叠的功能内容。

OBASHI 图例可以用于获取资产是如何支撑业务的证据。一旦图例建立以后，数据流可以叠加至 OBASHI 图例来展示资产是如何交互的。

这些交互通过汇聚来展示 IT 交付给业务的服务视图。

在这些概括的支撑文档的任意点上，可以将指南、SLA 等文件作为附件。这可以在某一项资产级、数据流或业务服务级别发生。

从而，用户可以添加更多基于上下文信息至图例，并获取服务交付的管控原则。

F.2 ZACHMAN FRAMEWORK

F.2.1 什么是 Zachman？

Zachman 框架通过 6x6 的二维矩阵来将企业的基本构成进行分类，从而提供了对企业高结构化的定义：

水平的：框架里水平的单元格基于常规的问题：What，关联到数据；How，关联到功能；Where，是网络或地域；Who，明确至人或实体；When，关系到时间；以及 Why，覆盖了原动力。

垂直的：垂直的单元格关联到干系人组，计划，详细描述了范围（基于上下文的）；所有人，定义了业务模型（概念的）；设计人，文本定义了系统模型（逻辑的）；建设者，技术模型（物理的）；实施人/签约人，展示了详细的陈述（脱离上下文的）；以及工人，详细描述功能性的企业实例。

在充实框架时，每一个单元格必须与其上和其下的对应起来，并且所有的水平单元格也必须被整合。这就直接显示并且评估了当前正处于被分析阶段的 IT 组合与业务的融合状况。

多个软件解决方案可以帮助完成存放和处理咨询师收集到的信息。解决方案的输出通常形成在 Zachman 模型的周围。

F.2.2 常见问题

不幸的是 Zachman 高度依赖于主观的数据，这些数据是小颗粒的。可能被认为难以获得平衡。试图从多个人工制品得到一致的模型会是一个问题，通常被总结为：一项架构的‘How?’是另一项架构的‘Why?’

F.2.3 合并 OBASHI

Zachman 可以与 OBASHI 合并以获得广泛涉及的环境来促使更好的决策。

为 Zachman 收集的数据可以很容易的通过 OBASHI 方法论来建模。Zachman 的水平横排包含了关于业务流程的信息，这些信息可以直接映射至 OBASHI 的业务流程层。这些信息还包含了关于网络基础架构、物理位置以及干系人的物理信息。这些信息映射至 OBASHI 的基础架构及所有人层。

Zachman 还提供了收集关于战略和时间跨度的支持信息的手段。OBASHI 可以帮助分析、管理和记录业务战略，同时业务规划的信息可以与构想中的项目交付所影响到的所有元素发生关联。

通过 Zachman 方法论获取的支撑文档可以通过与之关联的元素集成至 OBASHI 模型，无关它们在 OBASHI 的哪一层。这又增加了 OBASHI 内容的丰富程度。

F.3 TOGAF

F.3.1 什么是 TOGAF

TOGAF 是‘详细的方法及一套支撑工具用于开发一套企业架构’。是由 TOGAF(The Open Group Architecture Group)开发的。

TOGAF 的核心是四类架构，这四类架构被视作整体企业架构的子集：

业务架构：定义了业务战略、治理、组织架构及核心业务流程。

数据架构：描述了机构的逻辑和物理数据管理资源 and 数据资产的结构。

应用架构：描述了应用系统部署的蓝图，它们的

交互及它们与在业务架构中定义的核心业务流程的关系。

技术架构：描述了用于支撑业务的软件和硬件的逻辑上的能力，数据架构及应用架构。例如，网络、IT 基础架构、中间件、标准等。

F.3.2 常见问题

TOGAF 需要通过定制来符合特定的企业，因此需要对该方法论有非常强的知识及对业务模型的了解。找到这样素质的团队是困难的。将这样的团队带入项目需要得到批准，这需要得到机构内高层领导中具有核心影响力的主观愿望和支持。

TOGAF 是动态的流程模型，需要在实施中进行配置。

F.3.3 合并 OBASHI

OBASHI 可以与 TOGAF 一起使用来提供合并的四项 TOGAF 架构视图。这四个架构与 OBASHI 框架有直接的映射关系。

TOGAF 的业务架构包含了与机构中所有的核心业务流程的关联。这些流程可以在 OBASHI 的业务流程层完成建模。

组织架构的信息可以直接映射至 OBASHI 的拥有人层。相关的支撑业务战略和治理文档可以作为关联信息与对应的每一项元素进行关联，为 OBASHI 模型提供更多基于上下文的背景资源。

TOGAF 的数据架构详细描述了机构的逻辑及物理的数据资源和数据资产。这些可以通过物理和逻辑层的元素，在 OBASHI 的应用层进行建模。来自 TOGAF 的有关数据实体和结构的信息可以作为支撑文档与 OBASHI 相关的元素进行勾连。介于应用和数据存放空间之间的界面可以通过数据流来进行映射。

TOGAF 的应用架构记录了一个机构已部署的应用组合。这可以通过 OBASHI 的应用元素来建模。它们的交互和它们与核心业务的关系可以通过数据流和 B&IT 图例来建模。

TOGAF 的技术架构可以通过硬件、系统和基础架构元素来代表。它们与业务、数据和应用架构可以通过 B&IT 图例来建模。支撑文档可以与

OBASHI 的每一个元素交叉关联。

F.4 国防部架构框框(DoDAF)

F.4.1 什么是 DoDAF

DoDAF 是一种管理框架。是美国国防部作为将企业架构或系统架构组织为一致的相互补充的视图的一种标准方法。美国国防部坚持认为所有的大型武器和技术系统的采购必须用到这个框架。

在 DoDAF 中，对人工制品的分类核心是互通性这个概念。这又被分为几个级别。

针对这些人工制品，有四项基本的视图：

全视图(AV)：提供了对整个架构的描述，并且定义了范围、目标、假定用户、环境等。

运行视图(OV)：对 DoDAF 来讲，这个视图是独一无二的，描述了所有需要用于达到国防部使命的内容。其中包括任务、活动、I/O、规则、命令-控制-协调的关系。

系统视图(SV)：提供了系统构件、网络、逻辑数据模型、顺序的活动、时间跨度等。

技术标准视图(TV)：允许抽取当前的标准和未来的标准。

DoDAF 框架的开发目标是提供使人诧异的丰富的环境文档。为了在采购流程中评估投标方案，

任何关于人工制品的内容项必须进入目录并且在速度、表现、界面要求和某项任务的适用性方面进行量化。因为美国国防部定义系统的性质，DoDAF 是极为复杂和严格一种框架。

有多种工具可以帮助用户收集和汇报丰富的数据以满足 DoDAF 的规格要求。

F.4.2 常见问题

为了符合 DoDAF 的要求，需要投入非常多的工时，因此成本约束可能会制约 DoDAF 在业务中的采纳。

当 DoDAF 成为业务开发必须的内容时，如在国家防御行业，成本要求必须纳入相关的 DoDAF 项目中。

F.4.3 合并 OBASHI

当需要映射系统视图内信息时，OBASHI 能很好的完成与 DoDAF 的合并。这项视图中的信息可以与应用、系统、硬件和基础架构进行建模。

然而，DoDAF 开发的初衷是提供详细的文档来确保军标的硬件和软件规格，虽然这些文档能可以与 OBASHI 的元素进行关联，OBASHI 还是应该被当做图示化和沟通整体设计目标的辅助手段和 DoDAF 引导的解决方案的系统集成的辅助。

OBASHI B&IT 图例可以用于这项目标。

附录 G –其它数据流类型

G.1 简述

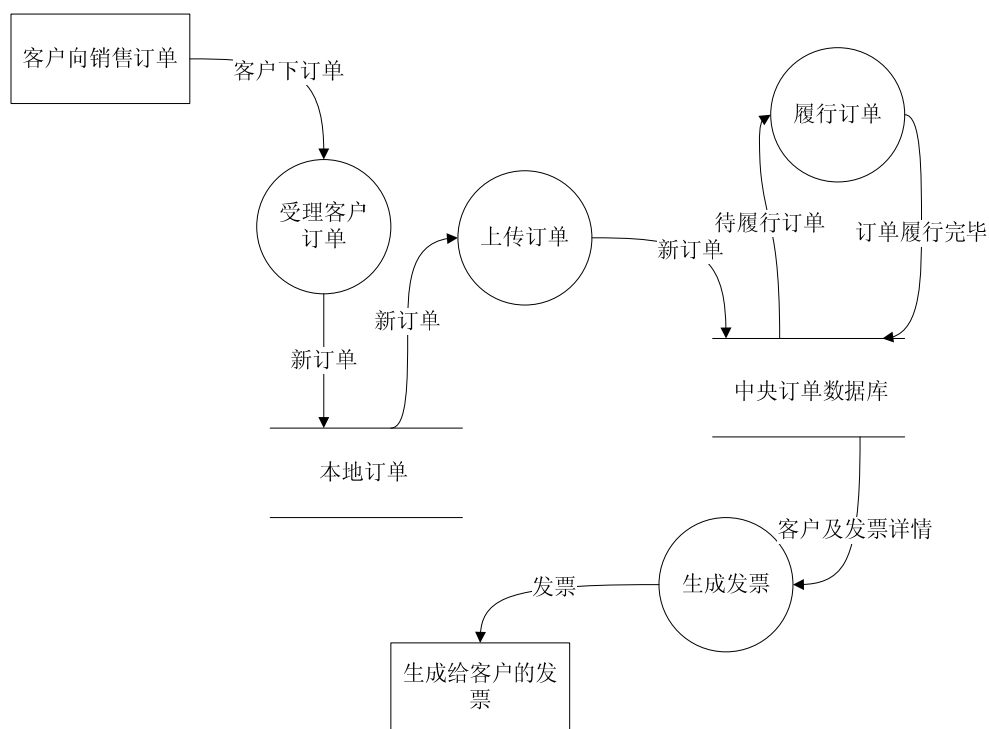
多年来数据流在系统分析和设计社区中都被认为是重要的。多个不同的方法论被推荐用于执行数据流分析。

通常，这些方法论源于系统分析和软件工程的理论。因此，它们聚焦于记录数据是如何在应用中和应用间流转的。

OBASHI 扩展了这种数据流视图，将之放在所有 OBASHI 层的上下文中：

拥有人、业务流程、应用、系统、硬件和基础架构。

以下就 OBASHI 如何被用于展现数据流的模型进行讨论。



G.2 YOURDON/DEMARCO

Edward Yourdon 和 Tom DeMarco 研发了一套结构化的分析方法论，其中包括了数据流图例。这套方法论成为 1980's 年间被广为选择的方法论，并且被系统分析员所广为接受。

建立这套图例的流程包括建立上下文的图例，详细描述被分析系统之外的外部影响因素（实体）。与其名称对应，基于上下文的图例显示了分析之外的环境和主要界面。

流程被分解为更小的单元构件，数据流被放置于构件之间。针对每一项构件，创建更多的图例。这一技法被重复直至被建模的构件有足够的详细度来支持规格文档。

Yourdon/DeMarco 之所以今天没有像 1980's 和 1990's 年代那样使用广泛的原因之一是这种分析或设计系统会导致系统围绕着隔离的数据文件以及孤岛数据。分析员开始趋向以数据库为中心的设计方法，分析数据是如何获取和复用的，而不是功能性流程中数据是如何存放的。

G.2.1 与 OBASHI 的主要差距

Yourdon/DeMarco 鼓励那些聚焦于数据是如何存放的系统而不是数据是如何适用的系统。这适合在应用级工作，但不考虑数据是如何在机构内部移动的。Yourdon/DeMarco 在进行数据映射时完全不考虑数据流在使用中的上下文关系。

Yourdon/DeMarco 的转入式特点支持数码动态的第三定律，其中描述了‘一项数码流可以包含一项或多项数码流’但是，与 OBASHI 不同，它不支持其它四项定律。

G.2.2 如何将 Yourdon/DeMarco 映射至 OBASHI
将 Yourdon/DeMarco 映射至 OBASHI 还是可能的。每一个流程在 Yourdon/DeMarco 中被描绘成圆圈，可以被映射至一项 OBASHI 应用元素或 OBASHI 业务流程元素。类似的，每项数据存储空间可以被映射至业务流程元素（在手工交互的情形下），或代表关系型数据库系统的应用元素。

Yourdon/DeMarco 的外部影响作为 OBASHI 的数据流被记录。

在 OBASHI 中，逻辑元素可以被用于直接与 Yourdon/DeMarco 的转入文档中包含的流程层级，同时子数据流可以被用于映射流的转入。

G.3 SSADM

结构化的系统分析和设计方法论(SSADM)在 1980s 被开发为一套标准并在英国的政府项目中赢得一席之地。确实，SSADM 目前是英国政府的注册商标。

不只是聚焦于数据流，SSADM 包含三个主要的技术，覆盖了逻辑数据设计、物理设计和逻辑流程设计。

逻辑数据建模和设计焦点在分析系统需要哪些数据，并建立这些数据的逻辑定义（个体）。这些个体间的关系被建立模型用于记录不同类型

数据间的关联关系。

逻辑流程设计的数据流建模与 Yourdon/DeMarco 的方法非常接近。系统被分解为构件，逻辑数据模型的个体与数据流一起被记录用于展示系统构件之间那些数据被传递。

物理设计聚焦于逻辑和数据流模型是如何映射至项目群设计的。它会考虑系统如何实施系统以评估映射逻辑结构至物理介质的需求。其中的例子是物理的数据库系统将用到存储中的流程来更新和插入数据，以及项目群中的各种函数是如何相互形成界面的。

SSADM 选择了严密的方式来设计和分析—政府性质的动机可想而知。但是，这一方法可能费力费时，因此通常不会全盘照搬。

在项目初期以这种方式来设计系统会导致非常高的额外工作。许多 IT 专业人士相信在项目初期投入更多可以避免后期代码反复修改(更糟糕是设计的反复)。这类返工经常导致开发超出预算，项目超时。

还需要注意的是，转变为灵活开发是与 SSADM 对立的，但会包含初始阶段的原型法和同级回顾以尽量后期的返工。

G.3.1 与 OBASHI 的主要差异

SSADM 的设计初衷是提供结构化的方法来设计一套计算机系统，用于满足业务对设计的要求。SSADM 通过逻辑数据建模、数据流建模以及实体行为模型建模来达到这个效果。OBASHI 可以展现资产和资源是如何被用于支持业务或机构，数据流如何被映射至 B&IT 图例。

SSADM 是关于记录关于数据的‘What’及‘How’的问题。OBASHI 的焦点更多在‘Why’数据流存在并且提供了关联 SSADM 文档到所用资产的机制。

G.3.2 如何将 SSADM 映射至 OBASHI
SSADM DFD(数据流图例)可以直接映射至 OBASHI

在 DFD 的每一个节点上，都会被创立一个应用的元素。然后建模者可以利用依赖关系来展示他们是如何关联的，或者可以利用 B&IT 图例来建立

应用是如何部署的并连接至基础架构部件。

一旦应用被通过连接或依赖关系建立连接，数据流的映射即可建立于应用元素之间来记录数据的流转。

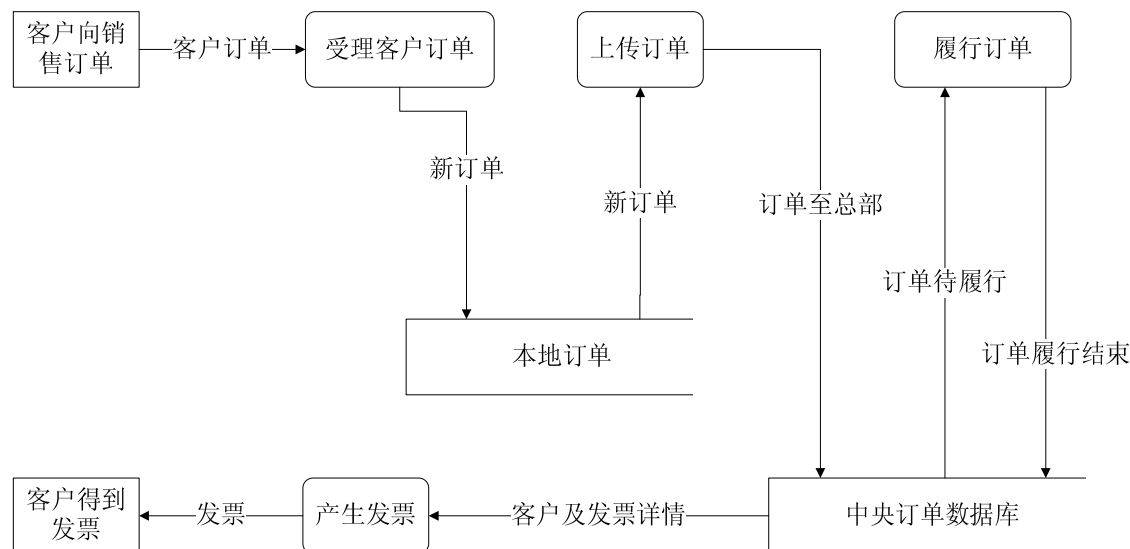
所有 SSADM 与 OBASHI 元素关联的文档可以基于该元素被交叉引用。

出现于 1970s 的中晚期。将数据流转模型从物理的系统设计抽象出来。通过去除物理方面的数据流，这些数据流即可被以逻辑的方式进行分析与设计。然而，一旦建模完成并且被修改，物理设计必须被重新考虑以支持实施的开展。

Gane-Sarson 为系统分析员提供了建立高层流程的方法。这些高层流程然后被分解为子流程。这些子流程被进一步分解直至底层图例被创建好。

G.4 GANE-SARSON

非常类似于 Yourdon/DeMarco，Gane-Sarson 方法



G.4.1 与 OBASHI 的主要差别

Gane-Sarson 鼓励那些聚焦于数据是如何存放的系统而不是如何使用的系统。它只工作在应用层，而不考虑数据在机构中的流动。在数据映射中 Gane-Sarson 完全不考虑数据流使用的上下文关系。

Gane-Sarson 的转入式特点支持数码动态的第三定律，其中描述了‘一项数码流可以包含一项或多项数码流’但是，与 OBASHI 不同，它不支持其它四项定律。

G.4.2 如何将 Gane-Sarson 映射至 OBASHI

将 Gane-Sarson 映射至 OBASHI 还是可能的。每一个流程在 Gane-Sarson 中被描绘成圆角的长方形，可以被映射至一项 OBASHI 应用元素或 OBASHI 业务流程元素。类似的，每项以开口长方形为标示的数据存储空间可以被映射至业务流程元素（在手工交互的情形下），或代表关系型数据库系统的应用元素。Gane-Sarson 的外部影响作为 OBASHI 的数据流被记录。

在 OBASHI 中，逻辑元素可以被用于直接与 Gane-Sarson 的转入文档中包含的流程层级，同时子数据流可以被用于映射流的转入。

G.5 总结

不同于上述的方法论，OBASHI 方法论被设计用于基于上下文的框架来展示流程间及数据存储空间之间，以及其他业务和机构之间的关系。通过对数据流在同一框架中流转的映射，这些具有上下文的信息提供了更清晰的，数据是如何在机构中被使用的信息。

OBASHI 不光覆盖了应用层或业务流程层，而且也考虑了业务或计算机系统工作所需的所有内容，并提供了数据流。

大量现有的方法论可以直接建立与 OBASHI 的映射关系。

词汇表

缩写表:

缩写	解释
B&IT	业务与 IT 图
BAU	通常情况下的业务状态
BSI	英国标准组织
CAD	计算机辅助设计
DAV	数据流分析系统
ISO	国际标准化组织
KPI	关键性能指标
OBASHI	所有者、业务、流程、应用、系统、硬件与基础架构
OGC	政府商务办公室
SLA	服务级别协议

名词定义表:

词语	解释
在之上 (Above)	在 Y 轴上的定义
应用层 (Application Layer)	应用层上的元素代表着软件应用。这些应用在使用他们的业务流程的下面，而在合适地系统曾元素的上面。应用元素可以在其他应用元素的下面，从而形成一种应用的层次结构。
应用组合 (Application portfolio)	一个数据库或者结构化的文档用来在应用的整个生命周期中实现对应用的管理。应用组合记录了所有应用的关键属性。应用组合经常被作为服务组合或者配置管理系统的一部分来实施，
资产 (Asset)	任何资源或者能力。一个服务提供商的资产包括任何能够对传递服务有贡献的对象。
资产注册表 (Asset Register)	该注册表包括在该模型中所有用到的元素，包括详细的名字、类型、一个简洁的描述以及独一无二的标识符。
审计 (Audit)	正式的检验与验证来检查是否一个标准或者一系列指导被遵循，包括记录是否准确，效率以及效益目标被满足。一个审计可以被内部或者

	外部团队执行。
在之下(Below)	一个 Y 轴空间的定义
最佳实践(Best Practice)	被多个组织成功使用，从而被证明的行动或者流程。ITIL 就是一个最佳实践。
双向地 (Bi-Directional)	给数据提供的能力，使得它能够双向流动。
大图 (Big Picture)	一个或多个业务与 IT 图
蓝图 (Blueprint)	业务或者组织的模型，包括他们的工作实践和流程、以及为了交付远景描述中所描述的能力所需要的信息与技术。
底部 (Bottom)	一个 Y 轴空间的定义
底部至上部 (Bottom—up)	一种方式，元素可以按照层次的方式在他们自己的层内进行描述，比如在应用层，整个应用可以在底部显示，而他们相关的模块可以放在他们的上面。
业务与 IT 图 (Business & IT Diagram)	一个框架用来记录与建模，来阐述人、流程与技术之间如何通过交互来实现一个业务工作。一个业务与 IT 图包括 6 个层次，由水平线来分开。这些层由图的上面到下面，名称分别是：所有者、业务流程、应用、系统、硬件以及基础架构层。
一般情况下的业务状态 (Business as usual)	一般情况下，业务取得其目标的方式。
业务关键文档 (Business-Critical Documents)	拥有对于持续良好运行的业务至关重要信息的文档。这些信息必须准确、安全，同时以合适的方式被控制与管理。
业务运营 (Business Operation)	业务流程每天的执行、监控以及管理。
业务流程 (Business Process)	业务拥有并执行的流程。一个业务流程能对交付产品或者为业务的客户提供服务做出贡献。
业务流程图 (Business Process Diagram)	业务流程层包括的元素代表了那些被‘所有者’使用地商业流程或者功能。这些元素被放置在合适的‘所有者’下面，业务流程元素可以在其它业务流程元素的下面，从而形成一个业务流程的层次。
互依赖 (Co-Dependent)	一个元素可以是互依赖的，即 A 元素依赖 B 元素，而且 B 元素依赖 A

	元素。
组件（Component）	一个通用的术语，指一个更加复杂的对象的一部分。比如，一个计算机系统可以是一个 IT 服务的组件，而一个应用程序可以是一个人力资源部门的组件。
连接（Connection）	一个连接显示了一个元素间的逻辑或者物理连接。
连接关系（Connection Relationships）	通过一个连接形成的双向关联。
连接性（Connectivity）	在一个组织中的人、IT 资产、资源与业务流程，以及它们所支持的服务，它们之间的逻辑或者物理关联被称为连接性。。
消费者（Consumer）	在 OBASHI 中，一个数据流的最后元素被称为消费者。
数据流（Dataflow）	OBASHI 框架中元素之间数据的流。
数据流分析视图（Dataflow Analyses View）	一个图形化与统计化的表达方式，也就是一个图，包括了所有支持一个单独的数据流的业务与 IT 资源。数据流分析视图是的单个数据流能够被分析与学习。
交付物（Deliverable）	一个项目必须生成的作为其需求的一部分内容。它可以是最终结果的一部分，或者是一个或者多个随后的交付物所依赖的中间结果。
依赖关系（Dependency relationships）	一种关系，指一个元素为了使自己功能正常必须依赖另一个元素，而且一旦另一个元素出现问题，将对其操作造成重大的影响。一个元素可以依赖一个或多个元素，依赖关系是单向的，而且可以有多个依赖元素。
设计（Design）	一个行动或者流程，其能识别需求，然后根据需求能定义一个解决方案来满足需求。
设计阶段（Design phase）	OBASHI 项目生命周期的一部分，在该阶段信息以业务与 IT 图以及数据流分析视图模型化的方式来收集。
数字化动态性（Digital Dynamics）	研究数据在人、流程和技术之间转化与流动的原则。
数字流（Digital Flow）	数据流、与数据相关的人员以及支持数据的信息技术之间的依赖关系。
元素（Element）	在 OBASHI 中，一个元素代表了一个组织中的一个资产或者资源。一个元素在业务与 IT 图与数据流分析视图中以一个简单的矩形形式表

	现。
交接阶段 (Handover phase)	OBASHI 生命周期的一个部分。在该阶段保证了在项目中生成的交付物已经交付到可操作的业务环境中。
硬件层 (Hardware layer)	硬件层包括的元素代表着操作系统运行所依赖的计算机硬件。这些元素放在合适的操作系统之下，而在合适的网络基础架构之上，硬件元素可以放在其他硬件元素之下，从而形成硬件的层次。
元素的层次 (Hierarchy of elements)	元素在同一层中放在其他元素的上面或者下面。
保持云状 (Hold Cloud)	在一个业务与 IT 图中，保持云状可以代表缺失信息或者需要对一个图的一部分进行进一步的调查或者检查
混合图 (Hybrid Diagram)	一个建模技术，通过该技术可以把逻辑与物理的元素在同一个图中显示。
影响度分析 (Impact Analysis)	由影响规则应用到一个或一组元素后产生的一系列对元素的结果。
影响规则 (Impact rules)	即影响如何从 OBASHI 模型像瀑布一样作用，以及每种元素如何级联地影响其他元素。
基础架构层 (Infrastructure layer)	基础架构层的元素代表着哪些用于硬件连接的网络基础架构，基础架构层元素可以放置在其他基础架构元素层下，从而形成一个支持业务的层次。基础架构元素可以放置在其它基础架构元素下从而形成基础架构的层次。
IT 服务管理 (IT Service Management)	IT 质量的实施与管理，来满足业务的需要。IT 服务管理由 IT 服务提供者通过合适地方式融合人、流程与信息技术来实现。
OBASHI 规则 (OBASHI rules)	控制 OBASHI 方法论的建模与操作的规则。
层 (Layer)	业务与 IT 图六个领域中的一个，这些领域一起提供组织元素的框架。
层关系 (Layer Relationships)	放置在同一 OBASHI 层的两个或多个元素之间清晰的关系。
左边 (Left)	一个在 X 轴空间的定义
逻辑元素 (Logic)	在 OBASHI 模型中，代表非物理实体的元素

Element)	
逻辑模型 (Logical Modeling)	显示元素如何被组织使用、而不用管它们的物理位置或者连接性的能力
主控文档 (Master Document)	受认可与被控制地一系列业务与 IT 图
建模 (Modeling)	用于创建 OBASHI 业务与 IT 图、数据流分析视图的技术。另外，它是 OBASHI 中用来预测未来资产、人、流程和技术地行为的技术。
节点 (Node)	提供或者消费数据的人、流程或技术的表现方式
OBASHI 色彩标准 (OBASHI Color Standards)	用于在每一层标记元素颜色地标准色彩使用方法
OBASHI 核心原则 (OBASHI Core Principles)	OBASHI 方法论的基础原则
OBASHI 图	一个业务与 IT 图或数据流分析视图
OBASHI 框架 (OBASHI Framework)	用于组织代表业务、IT 资产、IT 资源的单个元素并将其录入 OBASHI 六个层的框架。
OBASHI 方法论 (OBASHI Methodology)	一个思考方式，它能帮助你获得一个清晰的关于你的业务如何工作的图像。
OBASHI 模型 (OBASHI Model)	OBASHI 模型包括了与项目相关的所有图、记录与表。
OBASHI 项目生命周期 (OBASHI lifecycle)	使用 OBASHI 的方法论来执行一个项目所使用地一系列项目阶段。
OBASHI 规则 (OBASHI Rules)	定义元素如何能在一个业务与 IT 图中放在一起，以及在不同类型元素间允许哪些关系。
外面 (Outwith)	一个 X 轴空间定义，也是 Y 轴定义
所有者层 (Owner layer)	拥有着层的关系代表着个人或群体，他们拥有或者对业务流程层所展示的业务流程负责。

PMBOK	一个项目管理标准，它被项目机构所维护与印刷。PMBOK 用于支持项目管理知识的主体。
前序（predecessor）	一个元素按照顺序的放在另一个元素前
PRINCE2	英国政府用于项目的标准方法论，它是 Project IN a Controlled Environment 的缩写。
项目群（programme）	一个临时的，灵活的组织结构，用于协调、指导以及监视一系列相关项目活动地实施，以交付组织战略目标相关的结果。
项目（Project）	一个临时的组织，包括人以及其他资源，用于获得一个目标或者其他结果。每一个项目都有一个生命周期，典型的包括初始化、计划、执行、结束等。项目经常被一个标准的方法论管理，比如 PRINCE2
项目执行者（Project Executive）	管理者的一个部分，他们是关键的利益相关者或者决策制定者
提供者（Provider）	在 OBASHI 模型中，数据流的第一个元素被称为提供者
红色三角形（Red Triangle）	意指在一个保持云状的图形和在业务与 IT 图上用于描述的记录列上的文本之间的引用
细化阶段(Refine Phase)	OBASHI 项目生命周期的一部分，该阶段对在设计阶段所生成的业务与 IT 图进行进一步的深入挖掘，从而增加那些所需要的进一步信息
关系持久性（Relationship Persistence）	描述了一个事实，即一旦一个关系被定义给一个元素，那么无论这个元素怎么被重用，该关系都会持续。
关系规则（Relationship rules）	控制那些元素间隐式的或者显示的关系的规则
关系类型（Relationship Types）	在元素间有 6 种关系，连接（connection），依赖（dependency），层（layer），系列（set），序列（sequential），空间（spatial）。每种类型的关系都在一个特定的方式下运作，从而赋予模型隐式或者显示的意义。
库（Repository）	信息的一个存储集合
右边（Right）	一个 X 轴的空间定义
范围（Scope）	一个项目能作用的边界或者幅度

定义范围阶段（Scope phase）	OBASHI 项目生命周期的一个部分，该阶段定义了 OBASHI 模型所定义了的业务行动所应该获得的结果
资深职责拥有者（Senior Responsible Owner）（SRO）	单个的个体，他负责保证一个项目或者程序能够满足它的目标，同时将项目收益交付给相关利益者
序列（Sequence）	一系列元素，同时这些元素的顺序构成了序列，序列中还可以拥有其他的序列
序列关系（Sequential relationship）	OBASHI 中使用序列关系来建模数据流。连接的或者依赖的元素序列形成了数据流动的路径，数据流从序列中的第一个元素起源（第一个元素也称为提供者），从列表中的元素按顺序依次流过，最终在序列中的最后一个元素终结（它也被称为消费者）
系列关系（Set relationship）	一组元素，它们是按照一个指定的关系联系在一起。在一个业务与 IT 图中，一个系列内的元素没有地理的限制，也不受层的约束
空间类别（Spatial classification）	区分两个不同部分的分类，元素相对于另外一个元素如何放置在一个业务与 IT 图上。
空间关系（Spatial relationship）	一个隐含的关系，通过在一个或者多个业务与 IT 图中元素与另外一个元素的地理位置关系显示出来
空间规则（Spatial rules）	控制空间关系如何被应用的方法。遵循这些规则的元素可以说是和其他元素之间有空间关系。
利益相关者（Stakeholder）	在一个组织、项目、IT 服务等中有利益的人。利益相关者有可能会与活动、目标、资源、或者交付物相关。利益相关者可以包括客户、合作伙伴、雇员、股东、或者所有者等等。
利益相关者图（Stakeholder Map）	一个矩阵，显示了在项目群中利益相关者以及他们特定的利益。
标准（Standard）	一个强制性的需求
系统层（System layer）	系统层包含那些包括代表应用运行在其上的操作系统。这些元素放置在合适的应用之下，而在合适的硬件之上。系统元素可以放置在其他系统元素之下来形成一个系统的层次
标题块（Title block）	它维持并显示与一个业务与 IT 图特定相关的各种不同信息细节。标题

	块放在一个业务与 IT 图的右下角，它至少应该包括该图的名称、标识号（文档号）以及文档更新的版本
顶部（Top）	一个 Y 轴的空间分类
从上至下（Top-Down）	它指元素可以在它们自己的层内进行层次化描绘的方法，比如，在应用层，整个应用在顶部，而相关的模块放在下面
改革（Transformation）	一种独特的变更，组织可以依此对其整个或部分的业务进行改变
单向（Uni-Directional）	提供了一种能力，使得数据只能单向流动
视觉标准（Visual Standards）	正确描述 OBASHI 图中组件如何显示的方法
在之内（Within）	一个 X 轴的空间分类，同时也是一个 Y 轴的空间分类