



COSMIC 功能规模度量方法

4.0.1 版

数据仓库软件规模度量指南

1.1 版

2015 年 4 月

1.1 版的评审员（按字母顺序）		
Diana Baklizky TI Metricas Brazil	Arlan Lesterhuis Netherlands*	Luca Santillo Agile Metrics Italy
Charles Symons UK*	Monica K. Villavicencio Cabezas Espol Ecuador	Frank Vogelezang Ordina Netherlands
Chris Woodward Chris Woodward Associates Ltd. United Kingdom		

*本指南作者兼合作编辑

此 COSMIC 指南源自于荷兰 Sogeti 咨询公司 Harold van Heeringen [1]和 Luca Santillo 的敏捷度量论文。

中文版贡献者	
翻译组织者	麦哲思科技（北京）有限公司 www.measures.net.cn 电话：400-1780727
初版翻译	徐丹霞 麦哲思科技高级咨询顾问，CMMI 教员，大规模敏捷教练，COSMIC 讲师
校对	周雍正 麦哲思科技高级咨询顾问，COSMIC 讲师，多年银行系统开发经验

注：对 COSMIC 及本文档的任何疑问或指正之处，请加入 COSMIC 交流 QQ 群——309842452。

2015 年版权。版权所有。通用软件度量国际联盟（COSMIC）。非用于商业目的情况下，允许拷贝材料的部分或全部内容，但必须引用文档的标题、版本号和日期，并指明是获得 COSMIC 的授权许可。否则，拷贝需要特殊许可。

COSMIC 度量手册和技术报告的公开版本，包括其它语言的翻译版，可以从门户网站 www.cosmic-sizing.org 上找到。

版本控制

本数据仓库指南的变更履历如下表所示。

日期	评审员	修改/增加
2009 年 5 月	COSMIC 度量实践委员会	首次出版版本 1.0
2015 年 4 月	COSMIC 度量实践委员会	Please see Appendix B for a list of the main changes made for version 1.1 of this Guideline.本指南 1.1 版的主要变更请见附录 B。

TABLE OF CONTENTS 目录

1	简介	5
1.1	数据仓库系统	5
1.2	术语：功能性用户需求	5
2	什么是数据仓库？	6
2.1	概述	6
2.2	数据存储及各个数据仓库子系统	8
2.3	（多）维度数据存储	9
3	使用COSMIC方法度量一个数据仓库系统的规模	12
3.1	阶段1：度量策略	12
3.2	阶段2：映射阶段	13
3.2.1	ETL准备区域（ETL-SA）子系统	13
3.2.2	ETL数据仓库子系统（Inmon型）	14
3.2.3	ETL数据集市子系统（Inmon型）	15
3.2.4	ETL商业智能子系统	19
3.2.5	元数据管理子系统	20
3.2.6	数据集市子系统（Kimball型）	20
3.3	阶段3：度量阶段	21
4	使用COSMIC规模进行数据仓库子系统的工作量估算	22
5	参考文献	24
	附录A—术语表	25
	附录B—1.0版升级到1.1版的主要变更	26
	附录C——COSMIC变更请求和建议程序	27

简介

1.1 数据仓库系统

很多组织通过建立庞大的数据仓库获得持续的竞争优势，借助数据仓库他们可以随时评估各种业务操作的当前状态，并根据最新数据分析其趋势和关系。

数据仓库系统是一种特殊类型的业务应用软件系统，其被设计用于保存和/或展示详细数据及汇总信息，以便支持业务分析和决策。然而，经验表明想要估算建立这样一个系统的工作量是非常困难的。

估算开发一个数据仓库系统所需的工作量，首先要做的是估算其功能规模。作为第二代功能规模度量（FSM）方法，COSMIC 方法的设计完全基于软件工程的基本原理，并且适用于所有类型的软件，包括数据仓库软件。

关于数据仓库有两种流行的观点，一种是由 W.H. Inmon 定义的，另一种是由 R. Kimball 定义的。两种观点都对规模度量和工作量估算有影响。

COSMIC 指南旨在作为标准文档的补充，检验特定类型的软件如何使用 COSMIC 方法度量其规模。假定读者已掌握 COSMIC 方法“度量手册”[3]中所定义的基本概念，并且熟悉“应用软件度量指南”[4]。

本指南：

- 将介绍基于 Inmon 和 Kimball 观点的各种数据仓库系统模型，并定义其对应的各种子系统；
- 将展示如何使用 COSMIC 方法度量这些数据仓库系统的功能规模。

本指南的 1.1 版本与 1.0 版本的主要区别在于包含了 Kimball 模型，并更新了部分文本使其与 COSMIC 方法的 4.0 和 4.0.1 版一致。更正了 1.0 版的几处错误。基于 1.0 版的所有变更列表，请参见附录 B。

1.2 术语：功能性用户需求

COSMIC 方法的术语，参见度量手册[3]。本指南的特殊术语，参见附录 A。

COSMIC 方法，像其他功能规模度量方法一样，旨在度量软件的功能性用户需求（FUR）。关于 FUR 的定义，参见度量手册。

为了保持与 COSMIC 方法 4.0.1 版一致，我们限定术语“FUR”仅用于表示足够详细的功能性需求，这样的需求能够进行精确的 COSMIC 功能规模度量。

什么是数据仓库？

2.1 概述

几乎每个靠数据驱动的组织都离不开数据库的支持。多年来，数据库被不断优化以更好的支持这些组织的业务运营。然而，随着组织所拥有的数据库个数不断增加，从中抽取有用的数据也变得愈加困难。这是因为组织内各部分数据库所使用的数据没有使用统一的定义和命名。标准的数据库管理系统通常都没有配备数据合并功能，不能够生成多数据库合并的数据报告。于此同时，很多组织在探索“决策支持系统”和“大数据”技术，能够使其基于数据进行合理决策。建设一个优秀的决策支持系统，首先要明确组织的业务运营中最重要的方面，并完整、一致的收集这方面的数据。然后，才有可能基于这些数据生成报告。

数据仓库系统可以实现上述功能。Inmon，作为这一领域的开拓者，将数据仓库定义为：“一个面向对象的，集成的，时变的并且非易失的‘数据集合，用于支持管理决策流程’ [5]。另一个经常使用的定义是：“一系列决策支持技术的集合，使得具备知识的员工做出更快更好的决策” [6]。

数据仓库与传统的面向事务的业务应用系统有很大差别，后者包含大量用户和数据库间的交互（有时被称为 CRUD 功能，“CRUD”代表增删改查）。主要的差别如下表 1 所示。

	事务处理	数据仓库
目的	运行日常操作	数据检索和分析
结构	经过事务处理优化的RDBMS	经过查询优化的RDBMS
数据模型	标准化的	多维的
访问方式	SQL	SQL，加上先进的分析工具。
数据类型	业务操作生成的数据	为业务分析而生成的数据
数据性质	明细	明细和汇总
数据索引	较少	较多
数据连接	很多	部分
数据重复的风险	数据库一般是标准化的	数据库是非标准化的，如果其中包含聚合数据的话。
派生和聚合数据	极少	普遍

表 1：事务处理系统和数据仓库系统的差别

由于“第一代”功能规模度量方法主要是设计用于度量完整的事务处理系统，这些差别（尤其是数据仓库的多系统特点和数据库的多维性）使得这些方法很难用于度量数据仓库类型的软件。

数据仓库会从各个运维数据库中加载数据，清洗数据，将数据属性转换成统一定义，然后将其存储在数据仓库中。在这个过程中，为了提升数据仓库的性能，可能会形成一些聚合数据供用户使用。清洗和转换数据以及之后的聚合都要遵守既定的规则，这些都是“元数据”管理系统应用的一部分。用户可以利用诸如查询工具这类工具，从数据仓库中抽取数据。

图 2.1 显示的是一个完整架构的 Inmon 式数据仓库系统示例，图 2.2 则是一个 Kimball 式的示例。

¹ 所谓实时数据仓库已经被引入，只要业务操作数据发生变化，仓库就会实时更新。其目的是能够基于最新数据进行决策。实时进行数据更新实际上是一个“非功能性需求”。这一“NFR”不会影响功能规模，除非 NFR 影响到 FUR。

按照 Inmon 的观点，数据仓库的设计应该涵盖企业的所有数据，并且可以仅用一个项目开发完成。Kimball 则建议先进行数据仓库（一般被称为“企业级数据仓库”）的总体结构设计，然后分别用几个项目来开发预先设计好的数据集市。两者的区别是，Kimball 数据仓库没有考虑关系型数据，来自数据准备区的数据被直接加载到数据集市。总体来说，不是所有数据仓库系统都具备这些元素，也不一定按照本文介绍的这种方式交互。

"ETL"是指数据仓库系统的各个子系统从某个阶段将数据抽取（Extract）、转换（Transform）和加载（Load）到下一个阶段。ETL 子系统主要用于从数据源抽取数据，清洗数据，进行数据转换，并加载到目标数据仓库，然后再次加载数据集市。ETL 子系统也被用于生成和维护中央元数据库。

根据 COSMIC 方法的术语[3]，数据仓库系统的所有子系统都处于同一应用“层”，并且是“对等的”。各个 ETL 子系统不使用彼此的服务；它们甚至不会直接进行交互，而是经由共享数据库交互。在 COSMIC 模型中，持久存储介质对于任何需要它的软件都是可用的，不论其处于哪一层。

数据仓库的整体架构可能将 ETL 子系统定义为位于不同的子层内。这样的视角不影响下文中的 COSMIC 度量过程。

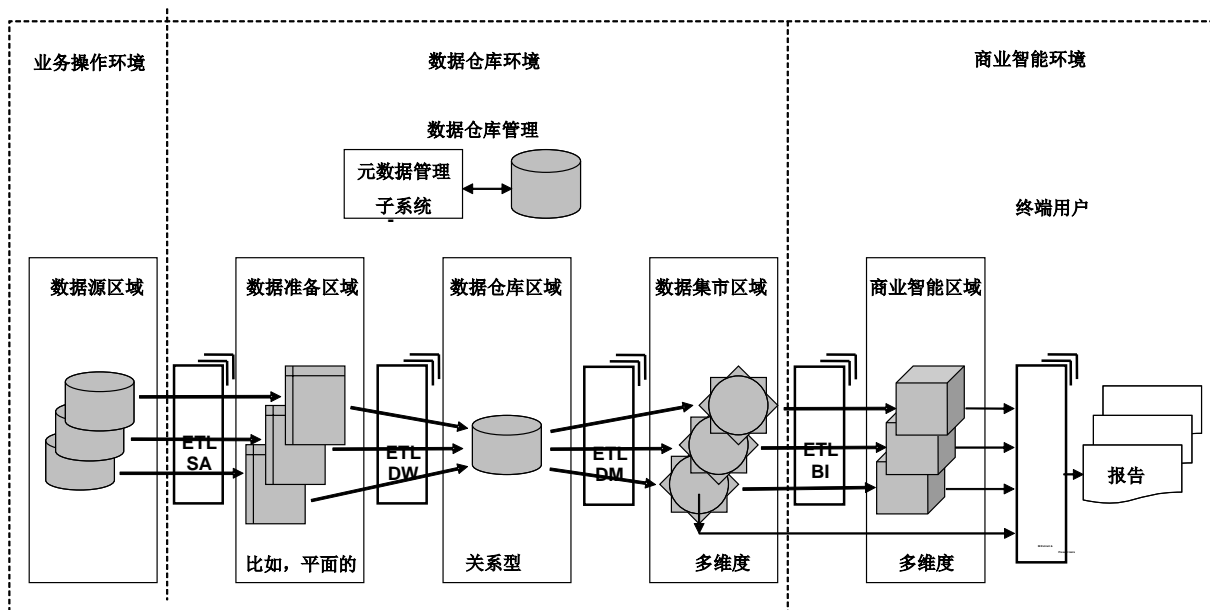


图 2.1 - 数据仓库示意图, Inmon 型

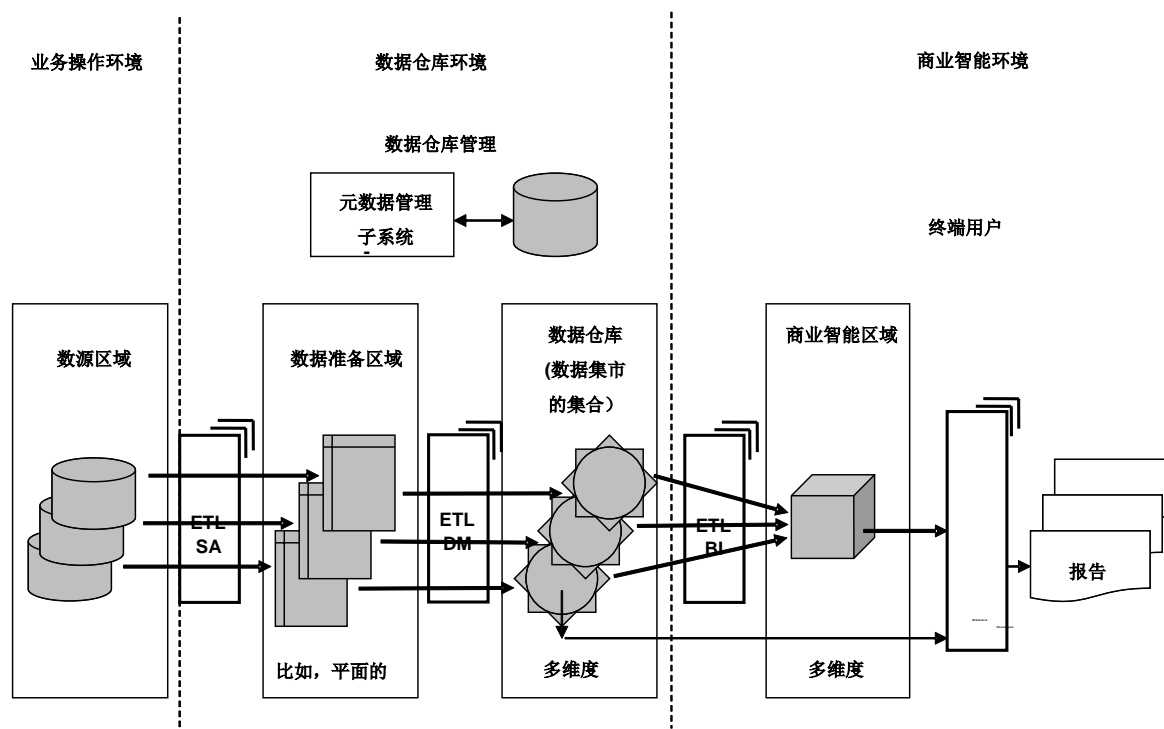


图 2.2 - 数据仓库示意图, Kimball 型

2.2 数据存储及各个数据仓库子系统

数据仓库主要的数据存储和子系统包括:

ETL 各子系统

数据仓库系统包括四个子系统, 相继完成数据存储。每个子系统的简要介绍在其存储数据的下方。各子系统分别是 ETL 准备区域子系统 (ETL SA), ETL 数据仓库子系统 (ETL DW), ETL 数据集市子系统 (ETL DM) 和 ETL 商业智能子系统 (ETL BI)。

业务操作数据源

操作数据源即业务操作系统的数据库, 系统中执行的所有处理都被捕捉并存储在该数据库中。源系统及其数据库并不是数据仓库的一部分, 因为数据仓库系统无法控制其数据内容和格式。这些系统中的数据可能是多种格式的, 从平面文件到具有层次的数据库和关系数据库都有。

数据准备区域

数据准备区域通常用来存储加载自“ETL SA”子系统的数。将来自多个操作源系统的数据进行抽取, 清洗、配对并加载为 (一般来说) 平面文件。(另一种我们在这里没有考虑的情况是业务操作环境中的应用系统直接向 ETL-SA 系统发送数据, 但这对于后面的分析造成的差别很小。)

数据准备区域禁止数据仓库的终端用户进入; 它不提供任何查询和展示服务。可能用到一个或多个数据清洗子系统来处理进入准备区域的数据, 例如解析名称和地址拼写错误等等。

数据仓库数据库

Inmon 型数据仓库模型中, 数据仓库数据库是一个关系型数据结构, 被优化以用于随后的发布。“ETL DW”子系统负责收集和整合来自各个业务操作系统的数, 这些数据在进入前经过了数据准备区域的

清洗，之后才存入数据仓库。后者成为唯一的共享数据源。在 Kimball 型数据仓库模型中，不存在关系型数据仓库；来自数据准备区域的数据直接进入数据集市。

数据集市

对数据集市（DM）最简单的设想是它是针对某一特定用户“部门”的数据仓库。在 Inmon 型数据仓库中，“ETL DM”子系统生成的数据集市源自中央数据仓库源。其原理是不论生成多少个数据集市，所有的数据都是取自同一个数据仓库数据库。从数据仓库到数据集市的数据分发就有可能形成新的汇总结果，此汇总结果是针对某一特定主题的、满足某一特定部门需求的信息。在 Kimball 型数据仓库中，“ETL DW”子系统生成的数据集市是源自数据准备区域存储的数据（平面文件）。如果大部分查询都能直接定位到数据集市里事先算好的、汇总的数据，这样的数据集市可以对终端用户的请求做出快速响应。

商业智能数据库和终端用户功能

维基百科的定义是，“商业智能”（或 BI）工具“通常使用数据仓库或数据集市收集到的数据，提供对商业运营的历史，当前和未来的预测视图”。涉及的术语包括决策支持系统（DSS），联机分析处理（OLAP）工具，经营信息系统，数据挖掘工具，以及使终端用户能够设置预定义标准或定制化查询的自定义工具。

就如图 2.1 和 2.2 显示的那样，BI 工具为了使得终端用户能够抽取和展现数据，可能从数据仓库或数据集市数据库直接读取数据，或者要求数据以一种特定的方式进行配置。如果是后者，可能需要重新配置 ETL-BI 工具并将数据存储于商业智能区域，如图 2.1 和 2.2 所示。

BI 数据库可以以立体形式展示，或更常见的作为维度存储（见下文）。维基百科对其的描述是，“运用 OLAP 技术配置的数据库，使用多维数据模型，可以进行快速的复杂分析和定制化查询”。

元数据管理

“元数据”的书面意思是“关于数据的数据”。这是数据仓库系统非常重要的一部分。元数据不是实际的数据；而是对数据属性特征的说明信息，比如数据的名称，定义，数据来自于哪里，怎样收集的，以及在存入数据仓库前应该怎样做转换（如果需要的话）。此外，有价值的元数据标识数据属性间的一些重要关系，这对于利用查询工具使用和解读数据的终端用户至关重要。元数据的维护和访问是经由元数据管理系统[7]实现的，如图 2.1 和 2.2 所示。不过，如果有特殊需要，ETL 子系统或者工具也可以对元数据进行整合和管理。

所有的数据仓库系统都必须有这种“技术上的”元数据，来规定数据清洗、转换和构建到数据仓库及数据集市中的规则。此外，有“业务元数据”和“流程元数据”之分，元数据的组织、维护和使用（被 ETL 子系统和终端用户工具使用）方式也多种多样。在本指南中，为了简化起见，我们只讨论对技术元数据进行访问的基本功能。度量人员要注意，使用和维护元数据的功能实际上会更加复杂。

2.3 （多）维度数据存储

图 2.1 和 2.2 显示了数据经过数据仓库系统的多个区域是如何转换的。当数据被加载到数据仓库中时，常见的方式是存储在关系表中，尽管有时候数据也可以以维度存储。在数据集市和商业智能区域就是这样。

多维数据存储的含义是把各种汇总和运算结果都存储起来，以便更快地检索数据。数据以维度存储的一个通常做法是使用“星型结构”。在一个星型结构中，事实表作为“核心（hub）”，各个维度表形成星型的各个“分叉”。图 2.3 给出了一个示例。事实表存储业务的一些主要信息。这种结构可以检索到预先选定的任何维度值组合的运算结果。

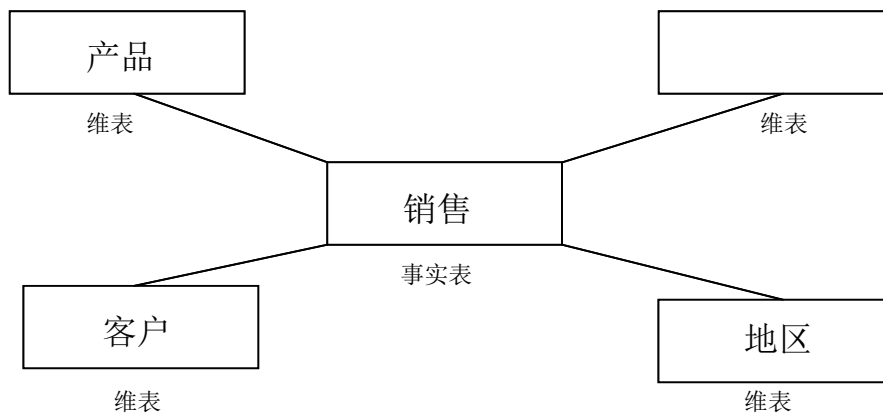


图 2.3 - 星型结构

维表一般由多层属性组成。因此，终端用户可以在查看他们的事实表数据时，任意收起或下钻一个维度。比如，在“时间”维度上，从每日收起到每周，到每月，再到每年。在“产品”维度，产品以收起到产品系列，到产品类型以及最终到“所有产品”（或“产品范围”）。

在图 2.3 的例子中，事实表的名称（或主题）是“销售”。假设最低等级的汇总是“每月销售情况”：

- 销售事实表可能有这些属性：“当年销售预算”，“实际每月销售”（月度），等等，
- 所有这些都是兴趣对象（OOI）“在指定区域和指定月份内销售给特定客户的特定产品”。

星型结构用于其他业务的例子如下：

- 对于某零售商，如图 2.3 一样，但是要增加一个“商店”维度；
- 对于某制造商，事实表可能叫做“采购”，其维度有“供应商”，“材料”，“工厂”，“时间”。

另外一个等价的数据存储形式是“雪花”结构，如图 2.4 所示。这种结构适用于每个维度都有其各自的分支且分支需要定义清楚的情况。“雪花”结构是通过关系数据库实现的，在技术上相较于星型结构有优点也有缺点，由于与本指南内容无关就不在此详述了。

由于两种结构从度量的角度都是相同的概念，下面我们只讨论星型结构。

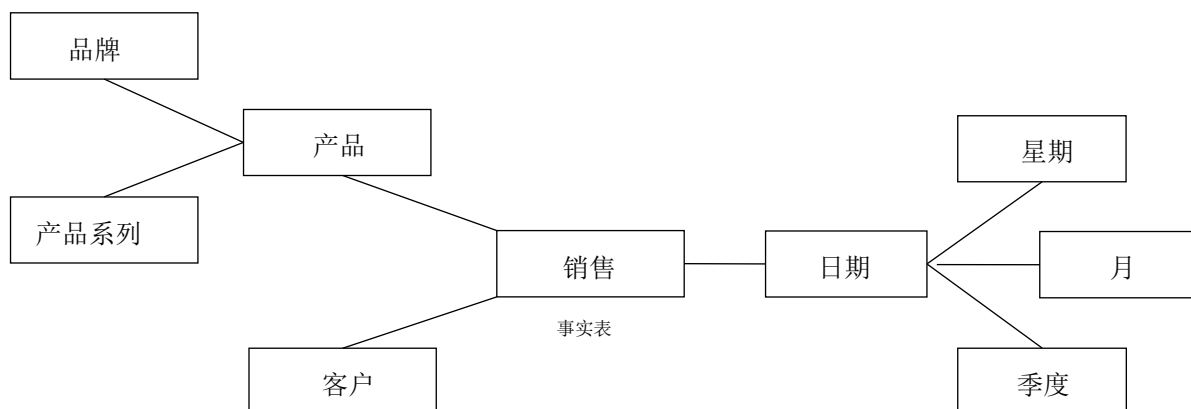


图 2.4 雪花结构

在商业智能区域，具有三个维度的数据可以被看成一个立方体。如图 2.5 所示。如果维度多于三个，我们仍然使用“立方体”这个术语，尽管实际上它是个“超立方体”。一个超立方体可以从其 N 个维度中任意取出三个维度进行图形化展示。

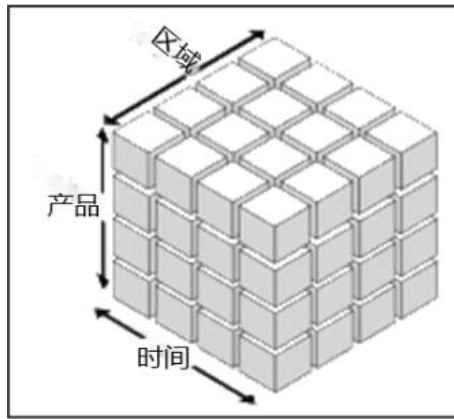


图 2.5 - 立方体结构

终端用户可以使用工具在这些阵列中查询自己想要的特定数据，不论是一个点、还是一行、二维、甚至三维或更多维，或者是阵列的任何一部分。这种查询方式在数据仓库术语中经常被称为“交叉分析”。

使用 COSMIC 方法度量一个数据仓库系统的规模

本指南假设读者熟悉 COSMIC 方法的原理和规则，其定义参见度量手册[3]。

用 COSMIC 方法进行规模度量，其流程包括三个阶段：

1. 度量策略阶段
2. 映射阶段
3. 度量阶段

3.1 阶段 1：度量策略

在度量策略阶段，需要定义目的、范围、功能用户、软件的分解层级和 FUR 的颗粒度级别。

本指南假设的模型是关于数据仓库的，其数据由周期性的批处理更新并用于在线查询，如第 2 章所述。接下来也会看到，所有的批处理都要使用持久存储数据，而数据可以被其他软件访问或请求。

目的：我们假设目的是度量数据仓库软件的功能规模，作为估算开发工作量的输入。

范围：假设度量的“总体范围”是整个数据仓库软件的 FUR，如图 2.1 和 2.2 举例的软件架构，不包括允许终端用户访问数据仓库、数据集市或商业智能数据库的“抽取和展示工具”。简单起见，我们假设这些是由外部辅助软件提供的，不需要度量，所以不在度量的总体范围内，但是，如果需要的话，进行这种工具定制的应用程序也可以用 COSMIC 方法度量。

数据仓库系统里的每个子系统或工具，即 ETL 子系统和元数据管理子系统，可能是独立运行的，也有可能单独开发的。那么鉴于度量目的，我们定义每个子系统都是单独的度量范围。包括：

- ETL 准备区域子系统；
- ETL 数据仓库子系统（只存在于 Inmon 型数据仓库）；
- ETL 数据集市子系统；
- ETL 商业智能子系统；
- 元数据管理子系统。

层：正如 2.1 节提到的，根据 COSMIC 术语，数据仓库系统的所有子系统都位于同一应用“层”，包括用于维护元数据的所有软件。

功能用户：因为 ETL 子系统不可能跟另一个 ETL 子系统直接交互，所以 ETL 子系统不可能是其他子系统的功能用户。（这些子系统都通过持久存储介质互相传递需要共享的数据。）然而如果像图 2.1 和 2.2 中的架构那样，所有 ETL 子系统都需要发送一个请求来访问元数据管理子系统，那么每个 ETL 子系统都是元数据管理子系统的功能用户，反之亦然。

ETL 子系统的其他功能用户，包括 DB 管理员或者 ETL 过程管理员，他们负责控制 ETL 运行，或者接收其输出（比如错误报告）。同样地，元数据管理子系统的其他功能用户就是元数据管理员。

商业智能工具的功能用户是发起查询的“终端用户”。

分解层级：因为每个子系统都是自治的，我们也将每个子系统都定义为一个范围、分别去度量，它们各自都可以被认为是一个“完整的应用系统”，即不需要再考虑分解。

颗粒度级别：我们假设每个子系统的 FUR 都详细到功能处理颗粒度级别，以便能够识别出单个的功能处理和数据移动。

3.2 阶段 2：映射阶段

映射阶段在度量过程中是最重要的一个阶段。在此阶段中，根据功能性用户需求（FUR）去识别功能处理、兴趣对象、数据组和数据移动。这一节主要针对 Inmon 型数据仓库（见图 2.1）。第 3.2.6 节探讨 Kimball 型数据仓库（图 2.2）的不同之处。

3.2.1 ETL 准备区域（ETL-SA）子系统

识别功能处理（FP），兴趣对象（OOI）和数据组

我们遇到的第一个功能处理是 ETL-SA 子系统中把数据从业务操作源移动到准备区的文件中，如图 3.1 所示。

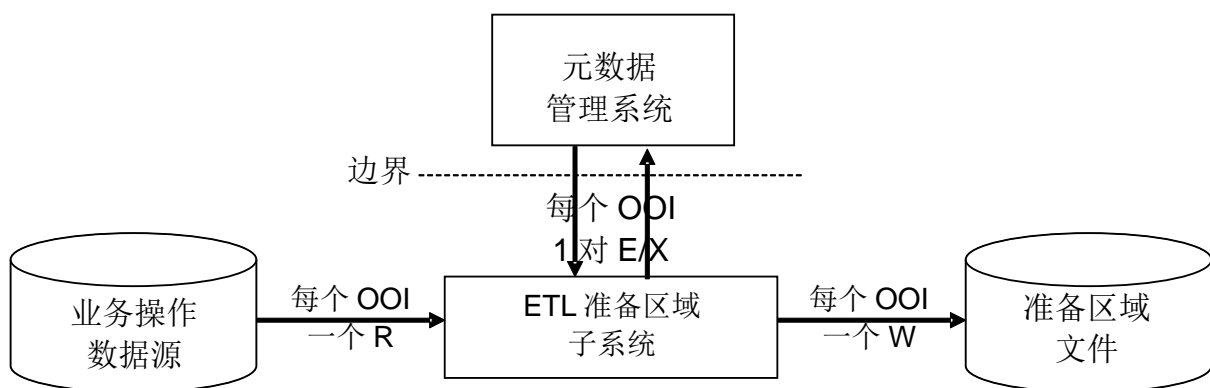


图 3.1 - ETL 准备区域子系统

识别功能处理时，首先我们应该在业务操作数据源中识别出需要移动到准备区的兴趣对象（OOI）。

单个业务操作数据源包含的数据可能是关于多个兴趣对象的。依据 FUR，必须为每一个 OOI 或者每一个“逻辑组合”识别一个功能处理：

- 从业务操作数据源提取数据组，
- 按照元数据管理系统定义的数据属性转换数据，
- 将数据组加载到准备区域。

一个“逻辑组合”的例子是在多项商品的订单中，需要在准备区维护表头/项之间的关系。一个功能处理可能需要抽取、转换和加载这样的订单，将表头和对应的多个商品项组合到一起，即描述了两个 OOI。

需要注意，“代码表”是否应该被看作是一个 OOI。对代码表的一般情况的处理见 COSMIC “业务应用软件指南” [3]第 4.2.3 节，特别是案例 6。在 ETL-SA 子系统中，如果有一个需求是让数据库管理员能够从业务操作数据库拷贝代码表到准备区，则可以识别为是一个功能处理，该功能处理将代码表作为一个 OOI。

在以下案例中我们假设 ETL 子系统可以使用存储在其边界内的数据，如果要访问元数据必须向元数据管理子系统发送请求。在后面的案例里，向元数据子系统发送请求和接收到请求的数据要分别识别为一个输出和一个输入。相比之下，元数据是存储在边界内的，获取这些数据只需要一个读。

案例 1：一个 ETL SA 子系统

ETL 准备区域（SA）子系统的简单的功能处理是要移动关于一个 OOI 类型的数据，识别的数据移动一般来说如下（E 代表输入，R 代表读，W 代表写，X 代表输出）：

E 启动该功能处理（比如一个时钟信号，如果是一个批处理过程的话）

- X 到元数据管理子系统来获取对该 OOI 的转换规则
- E 从元数据管理子系统得到所需数据
- R 业务操作数据源
- W 将转换后的数据写入准备区
- X 错误/确认消息

总规模：6 CFP。

下图 3.2 展示了例 1 中 ETL-SA 子系统的功能处理，以及它与元数据管理子系统进行交互的功能处理，以消息序列图的形式画出。

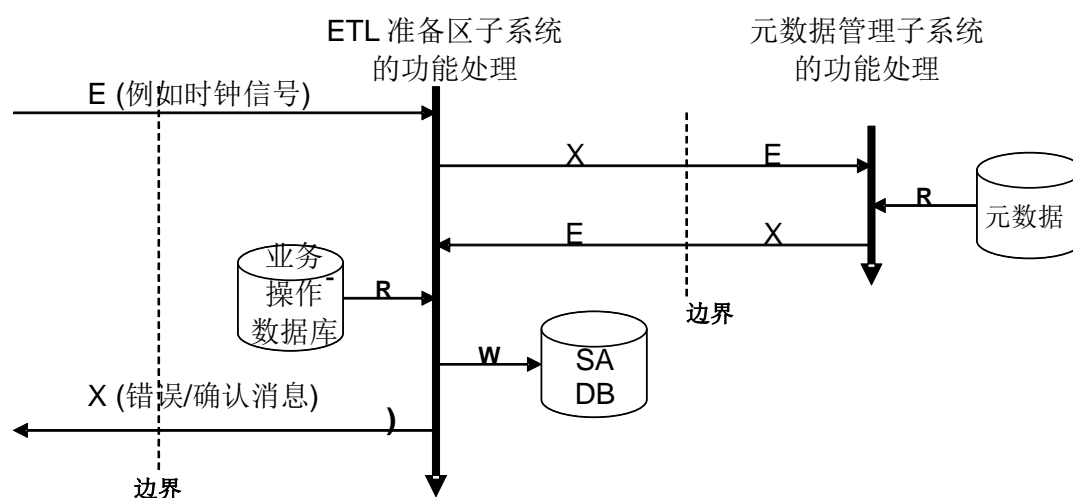


图 3.2 - 准备区域子系统与元数据管理子系统的交互

这是基本的 ETL-SA 子系统功能处理，实际的 ETL-SA 子系统可能会略有不同。

- a) 对于 ETL-SA 子系统的功能处理，如果其移动的比如说是，两个 OOI 的逻辑组合，则上面的多商品项订单案例应该额外增加一个读，如果数据都维护在跟准备区域同一个架构下，则为额外的 OOI 识别一个额外的写。（但是，跟元数据子系统查询第二个 OOI 的转换规则这一动作不需要额外识别一对输入/输出，假设第二个查询只是第一个查询的重复执行。）那么，这个功能处理的总规模就是 8CFP。
- b) 如果业务操作环境这边有一个应用程序给 ETL-SA 子系统提供数据（即 ETL-SA 子系统是“被动的”，等待业务操作数据的到来），那么案例 1 功能处理的触发输入就是一个时钟信号，可能从业务操作系统移动一个 OOI 的数据并加载到准备区域。因此，就没有必要识别对业务操作数据的一个读，这个功能处理的总规模就是 5CFP。
- c) 如果 ETL-SA 子系统需要的元数据是维护和存储在该子系统内部（而不是如图 2.1 所示，单独由元数据管理系统维护和存储的），则上面案例 1 中检索元数据所需的一对输出/输入应该用一个读来代替。这一点也适用于以下分析的其他 ETL 子系统。这个功能处理的总规模就是 5CFP。

注意，我们假设根据元数据处理规则进行数据组转换的数据运算包含在内了，根据 COSMIC 方法，这个运算在上例中是关联在写数据移动上。

3.2.2 ETL 数据仓库子系统 (Inmon 型)

识别功能处理 (FP)，兴趣对象 (OOI) 和数据组

在 ETL 数据仓库子系统中，我们发现给数据仓库提供数据的功能处理是从存储在准备区的文件中取得数据，如图 3.3 所示。

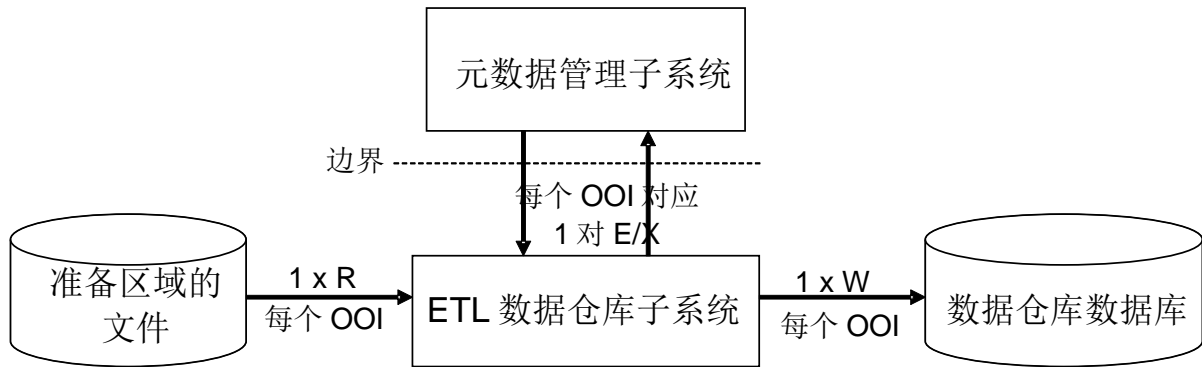


图 3.3 - ETL 数据仓库子系统 (Inmon 型)

同样地，对于每个 OOI 或 OOI 的每个逻辑组合，都应该识别一个单独的功能处理，该处理从准备区文件中抽取数据组，按照元数据管理系统提供的规则转换其数据属性，然后将转换后的数据加载到数据仓库数据库。

识别数据移动

ETL 数据仓库子系统的功能处理与数据准备区子系统的功能处理结构相同（假设数据仓库数据库中存储的数据不是多维的。多维事实表的维护案例，见 3.2.3 节）。

案例：关于 ETL 数据仓库子系统的简单的功能处理，该功能处理抽取、转换和加载的数据是关于单个 OOI 的，其结构如下：

- E 启动该功能处理（比如一个时钟信号，如果是一个批处理过程的话）
- X 到元数据管理子系统来获取对该 OOI 的转换规则
- E 从元数据管理子系统得到所需数据
- R 读准备区文件
- W 将转换后的数据写入数据仓库数据库
- X 错误/确认消息

总规模：6CFP

该功能处理的消息序列图与图 3.2 的结构是一样的。

3.2.3 ETL 数据集市子系统 (Inmon 型)

识别功能处理 (FP)，兴趣对象 (OOI)，数据组和数据移动

在 ETL 数据集市子系统中，我们发现给数据集市提供数据的功能处理是从数据仓库子系统中取得数据，如图 3.4 所示。

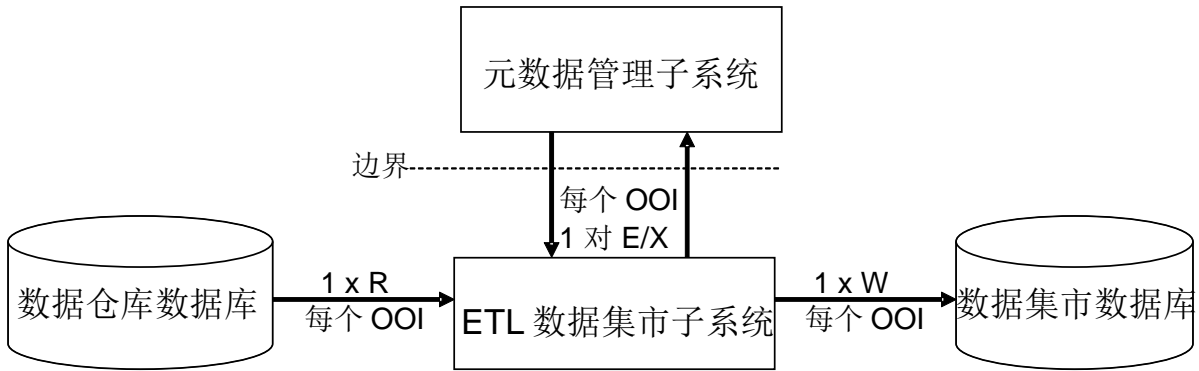


图 3.4 - ETL 数据集市子系统

在数据集市数据库中，数据以维度方式存储在如图 2.3 所示的星型结构中，既有“维表”又有“事实表”（有时是多个表）。很重要的一点是要分清事实表（即星型结构的“核心”）和维表（即星型结构的“分叉”）的创建方式。

一般从时间顺序上说（当然也取决于 FUR），维表会首先被加载，然后才是事实表。但是，为了方便起见我们先讨论加载事实表的过程，假设维表已经建立起来了。

为了计算 ETL 数据集市子系统的 OOI 类型个数、功能处理个数，最简单的办法是用一个星型结构案例展示这个过程，如图 3.5 所示。

案例：我们假设数据集市的功能性用户需求是存储销售数据，事实表中存储的是最低级别的数据（即没有经过汇总的），以及对四个维度各种可能的条件组合的汇总级数据²。这里既有“客户”（即按不同组织）也有“区域/国家”（即按地理区域）维度，因为我们假设某客户在一个国家的不同地区有多个办公地点。所以数据仓库负责人希望分别从组织和地理维度汇总销售数据。

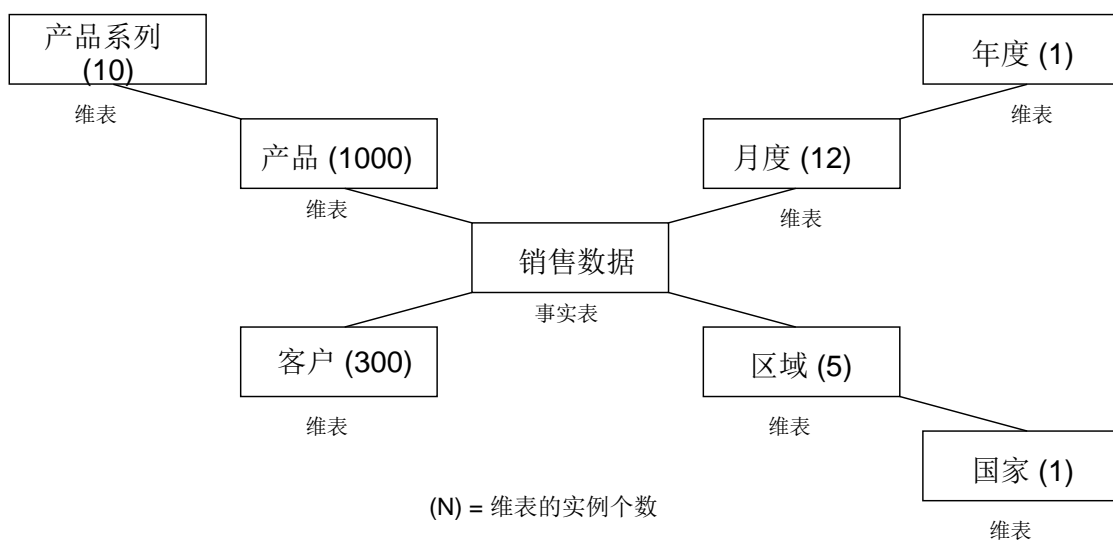


图 3.5 - “销售”数据集市的一个星型结构的例子

² 数据可能按照四个维度中的任意三个维度的组合被汇总至多个层级，四个维度中的三个是指产品，时间和客户或者区域/国家。但是功能性用户需求的要求是存储四个维度的所有组合的事实表。如果数据被汇总到更高级，就可以通过商业智能工具在数据集中查询到。此星型结构能汇总到的最高级别的数据是在这个国家的所有年份所有产品的销售数据。

我们假设最低层级的销售数据（即“在特定区域的特定客户在指定月份购买特定产品的销售情况”这一兴趣对象）是通过读取指定月份的所有已付款并发出的订单得到的。“销售情况”这一兴趣对象的属性可以通过读取订单的如下信息得到：“实际销售额”，“订单项”个数，“平均折扣”等等。

这些最低层级的销售汇总数据我们可以表示如下：

“销售情况（产品、客户、地区、月份）”：ETL 数据集市子系统中生成此销售数据事实表的功能处理可以包含如下数据移动：

- E 启动该功能处理（可能包括指定月份的起止日期）
- R 读取数据仓库里的订单头（比如获取客户名等等）
- R 读取数据仓库里的订单项（比如获取产品、数量、销售价格等等）
- R 读取数据仓库里的客户文件（比如获取客户下单的地点）
- W 销售数据（产品、客户、地区、月份）
- X 错误/确认消息

总数：6CFP

（这里我们假设，对于这个功能处理没有其他需要引用元数据管理子系统的地方。如果需要引用这个子系统的的功能，就需要在上述功能处理的基础上额外识别一对 X/E，跟之前的 ETL 子系统一样。）

但是，根据此案例的 FUR 描述，销售数据集市负责存储各维度所有可能组合的事实表。总共有八种可能的组合，可将这个数据集中存储的销售数据汇总到不同的级别。

此星型结构各维度的所有可能组合列举如下

- 指定客户在指定区域和月份购买指定产品的销售数据（最低层级）
- 全国范围内指定客户在指定月份购买指定产品的销售数据
- 全年内指定客户在指定区域购买指定产品的销售数据
- 等等

可以将其设想成卖出的一摞一摞产品；每一（类）摞产品都是现实世界中用户想单独保存的“东西”，即一个单独的 OOI。物理上，这八个事实表的每一个都是多维度的一种特定组合。

那么数据集市中的事实表包含八个 OOI，如下所示：

- 销售情况（产品，客户，地区，月份） **【最低层级】**
- 销售情况（产品，客户，国家，月份）
- 销售情况（产品，客户，地区，年份）
- 销售情况（产品，客户，国家，年份）
- 销售情况（产品系列，客户，地区，月份）
- 销售情况（产品系列，客户，国家，月份）
- 销售情况（产品系列，客户，地区，年份）
- 销售情况（产品系列，客户，国家，年份）

所以，在实践中，如果 ETL 数据集市子系统有一个功能处理是读取比如说数据仓库的订单，运算和加载到事实表的八个汇总维度，则这一功能处理需要有八个写数据移动，每个对应一个 OOI。这个功能处理的数据移动最小集合是：

- E 启动该功能处理（可能包括指定月份的日期）
- R 数据仓库数据库里保存的订单头（以便获取客户等等信息）
- R 数据仓库数据库里的订单项（以便获取产品、质量、销售价格等等）
- R 数据仓库数据库里的客户文件（以便获取客户下单位置所在地区）
- 8 个 W 所有八个维度的销售数据汇总
- X 错误/确认消息

总规模：13CFP。

注意以上案例为了便于表述做了极大的简化。首先，对于这个生成销售汇总数据的功能处理来说，我们假设所有 8 个销售数据事实表中所需的汇总数据都可以从输入数据中得到。这一假设意味着：

- 一个产品所属的产品系列可能需要从订单项中获取。（而实际上很可能某产品所属的产品系列以及某地区所属的国家可以从数据集市预先加载的数据中通过读操作得到。）
- 客户主文件可能不仅要包含客户办公地点所在的地区，还需要包含地区所属的国家；
- 每笔交易的日期都会被导入进去，以便从中提取交易月份和年份。

其次，如上分析的这个功能处理效率可能极其低下，因为每一层级汇总的销售数据需要从输入的订单数据中提取并计算。而高效的处理则可能要求，比如说，产品系列级别的销售数据是由（预先计算好的）产品级别的销售数据计算得到的，而不是订单项级别。由于可以通过多种手段达到高效的处理，我们就不对这一特定处理方式进行深入探讨了。只要知道在实践中，高效地执行几乎不可避免地要求将处理过程分解成多个步骤，就可能包含比上例中更多的读操作。（但是请记住，我们的目标仍然是度量 FUR，而不是度量处理效率如何实现的。）

第三，在这个案例中，为了简单起见，我们没有考虑各种业务问题，比如：

- 是否在实践中“年份”这一级别的销售数据汇总实际上是“年初至今的”，或者一个“滚动年的”；
- ETL 数据集市功能处理如何确定哪个订单项已开发票并在某月付款了（可能需要一个额外的读）；
- 事实上也有可能各级实际销售数据汇总的基础上存在其他数据，比如某一层级的预算或目标销售额等等；这些数据也需要功能处理进行加载；
- 该数据集市是否是每月从“空白”状态重新生成，还是在上个月数据集市版本上进行更新，加入新的销售数据；如果是后者，更新数据集市的功能处理就需要读上个月的数据。

重要的是：不要把数据集市事实表的数据移动类型和 OOI 类型跟其相应的实例数量相混淆。

案例：使用上图 3.5 给出的实例数据，该数据集市中每一层可能的销售记录汇总的实例个数计算如下：

销售数据（产品，客户，地区，月份）	$1000 \times 300 \times 5 \times 12 = 18,000,000$ 个记录
销售数据（产品，客户，国家，月份）	$1000 \times 300 \times 1 \times 12 = 3,600,000$
销售数据（产品，客户，地区，年份）	$1000 \times 300 \times 5 \times 1 = 1,500,000$
等等。	

除了加载和更新销售数据的功能处理之外，我们之前也提到需要有创建维度表和“加载主数据”的功能处理，这些都发生在销售数据加载之前。这些功能可以有多种形式。

案例：假设产品和产品系列主数据保存在数据仓库里。ETL 数据集市子系统中，负责建立相关维表的功能处理可能有如下数据移动：

- E 启动该功能处理
- R 读取数据仓库数据库里的产品数据
- W 把产品数据写入数据集市
- R 读取数据仓库数据库里的产品系列数据
- W 把产品系列数据写入数据集市
- X 错误/确认消息

总规模：6 CFP。

然后会有一个维护这些维表的功能处理（除了“新增”功能处理之外，可能只有“删除”功能处理），以及提供给终端用户查询维表的功能处理。例如：一个产品的删除功能处理可能非常复杂，因为需要检查是否有既存销售数据。

（注意，对于加载和维护维表的功能处理，每个维表的对象也是一个 OOI，即使它只有作为属性的“代码”和“描述”。因此，对于这个星型结构的例子，事实表就有八个 OOI，维表一共七个 OOI。）

最后，还有其他几个功能处理，提供给终端用户从不同汇总层级读取销售数据（见下一节），以及用于数据管理员删除或归档这些记录。

3.2.4 ETL 商业智能子系统

识别兴趣对象（OOI），数据组和功能处理（FP）

如上所述，商业智能（BI）区域会有“抽取和展现数据”的子系统使得终端用户能够直接从数据仓库或数据集市数据库、或者是经过特殊配置的 BI 数据库里查询到维度数据。后者通常需要有专业的终端用户工具支持，比如 OLAP 或数据挖掘工具。在这种情况下，就会需要一个 ETL-BI 子系统来提取数据仓库或数据集市数据库中的数据并转换和加载到所需的结构中。见图 3.6。

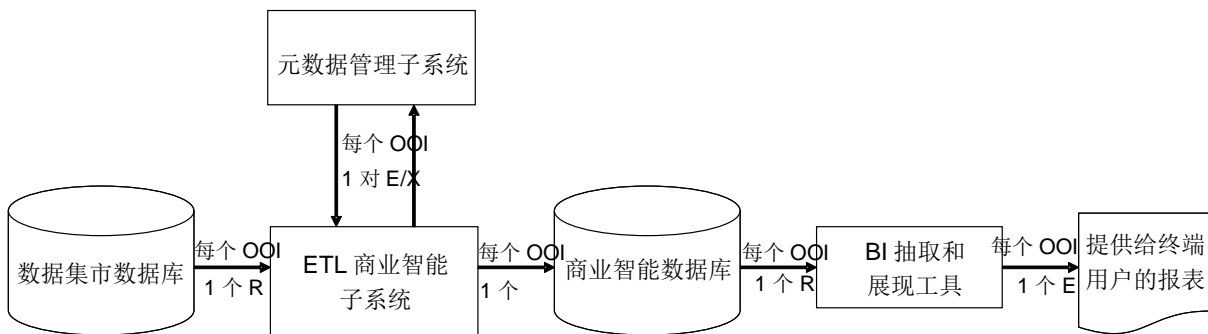


图 3.6 - ETL BI 子系统和抽取展现工具

对于 ETL-BI 子系统，识别其功能处理和数据移动要考虑的内容与上面的 ETL 数据集市子系统是完全一样的。

终端用户“抽取和展示”功能（数据仓库的术语叫“交叉分析”）可能差别很大。可以预定义好所有的查询，每个都要识别为一个功能处理。

案例：假设对于图 3.5 有一个需求是开发预定义的查询，来展示指定月份各产品系列的全国销售总额。这是对任意几个数据存储级别在某一个级别汇总。这个查询会有以下功能点（与上面的记法相同，且假设不需要错误/确认消息）：

- E 启动该查询，给出月份
- R 销售数据（产品系列，国家，月份）——假设数据都存在这一层级。
- X 销售数据（产品系列，国家，月份）——可能执行 10 次，每次汇总一个产品系列

总规模：3 CFP。

在本指南中，我们假设终端用户的“抽取和展示”工具本身是购入的软件，因此其功能规模不需要度量。但是，如果终端用户工具是本地开发的，就可以用 COSMIC 方法度量其功能（注意：不是度量该工具生成的那些查询功能的规模）。为此，需要对 OOI 进行建模、并用于识别工具本身的功能处理（即，不是终端用户使用该工具进行某数据库查询对应的那些 OOI 和功能处理）。这部分内容超出了本指南的范围。

如果是一个常规的查询，使得终端用户能够“下钻”（或“上钻”）来取得星型结构中任何维度组合的“事实”，这应该算作一个功能处理。各种选择准则对于所有查询来说类型都是一样的；每次查询选择的准则只不过是一个实例，即查询参数不一样。COSMIC 业务应用软件指南里有关于这类案例的分析 [3]。

3.2.5 元数据管理子系统

识别兴趣对象（OOI），数据组和功能处理（FP）

不管是一个单独的子系统（如图 2.1 和 2.2 所示），还是元数据被集成在每个 ETL 子系统中，元数据管理员都需要一些功能处理来完成他的操作。通过这些功能处理他可以创建新的元数据规则，维护或删除已有规则。兴趣对象的个数会因不同的数据仓库系统而有较大差别。处理元数据，比如更新频率或系统版本升级，用户档案，读取有特殊权限的文件，数据处理规则和对统计数据的使用，这些对于元数据管理员来说可能对应各种 OOI [7]。业务元数据，比如数据字典的内容，历史数据，某一数据所有者的数据，可能描述了另外一些兴趣对象。对于每个 OOI，可能都对应一系列功能处理，包括创建、更新、删除和报告这个 OOI。

对于所有这些 OOI，一个为响应数据仓库子系统的请求而抽取数据的功能处理，在图 3.1, 3.2, 3.3, 3.4 和 3.6 的一对儿 E, X 的基础上（或者有的地方是一个读数据移动），可能有多个额外的数据移动。

3.2.6 数据集市子系统（Kimball 型）

识别功能处理（FP），兴趣对象（OOI）和数据组

（本小节只针对 Kimball 型数据仓库）。

在 ETL 数据集市子系统中，我们发现有一个功能处理是把存储在分级区域平面文件中的关于一个 OOI 的数据输入给数据集市，如图 3.7 所示。

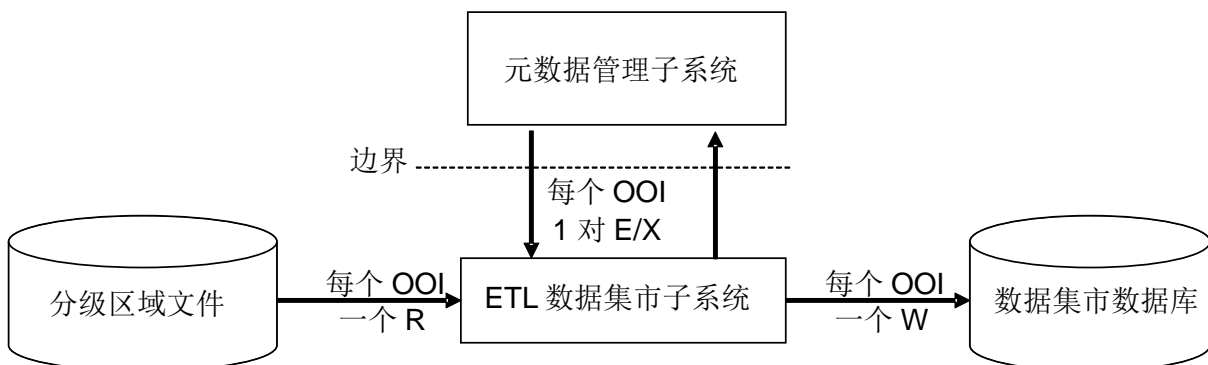
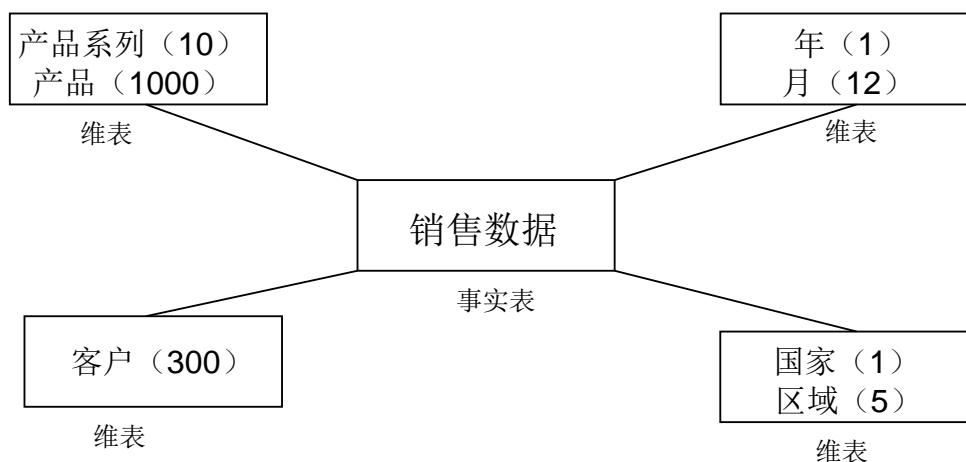


图 3.7 – ETL 数据集市子系统

对于 Kimball 型数据集市结构来说，图 3.8 的星型结构是与图 3.5 等价的 Kimball 型事实表和维度数据的物理存储模式。

在如 3.2.3 节同样的假设下，将转换后的数据加载到数据集市数据库的功能处理也与该节所述的功能处理相同。



(N) = 维表实例个数

图 3.8 – Kimball 型数据的星形结构

从 Kimball 型结构视角³来看，事实表包括如下内容：

- 多个维表的参数（外键），以及
- 一个业务过程涉及的“统计量”

统计量是该业务感兴趣的一些属性，比如订单总额，总成本，预订数量，毛利润等。

维表为事实表的这些统计量的查询提供选择参数。

实践中，维表可能包含多层属性。在 Kimball 型结构中，维表不是（而且为了便于查询则必须不是）标准化的。

注意，COSMIC 规模度量要求数据是标准化的，以便识别 OOI。因此，第 3.2.3 节以及相应图 3.5 的分析同样适用于本案例，不管我们用 Inmon 还是 Kimball 视角来看数据在物理上是如何存储的。

3.3 阶段 3：度量阶段

在度量阶段，把识别的所有功能处理数据移动个数进行汇总，得到 COSMIC 功能点（CFP）总数。每个数据移动计为 1CFP。

使用 COSMIC 规模进行数据仓库子系统的工作量估算

数据仓库软件的各个子系统可以分别用 COSMIC 方法度量。得到的规模也很现实地反映出对各种复杂数据结构的加载、更新或查询等功能处理（这些功能处理一般都比较复杂），以及各个子系统之间的交互。这意味着，在用于性能度量和估算时，用 COSMIC 度量的规模会比用第一代功能规模度量方法量出的规模更加准确。

但是，在用 COSMIC 度量的功能规模推导开发或维护数据仓库（DWH）子系统的工作量时，尤其是使用非 DWH 开发项目的基准数据做参照时，有些问题需要特别注意。

不同子系统的生产率不同

在估算开发整个系统的时间时，最好考虑一下不同子系统的项目生产率（规模/工作量）是不同的。例如，开发商业智能子系统（其主要是 GUI 功能）和开发其他子系统的数据库输入/输出所需的每 CFP 工作量可能有区别。因此，很重要的一点是：在定义、收集和记录工作量数据时，要加以区分，以便于后续对已完成项目的数据分析。

DWH 子系统的复杂度

度量人员需注意，实际的 DWH 子系统功能可能比本指南给出的案例更加复杂。跟这些案例相比，ETL 子系统可能：

- 与元数据子系统有更多交互，或者直接与元数据库交互，相比于上面给出的案例来说；
- 每个功能处理有更多的数据移动。

数据运算

数据仓库专家们可能会担心实际的数据转换过程中包含数据运算，这会花费很大一部分工作量，却没有在任何一种功能规模度量方法中加以考虑。如果度量目的是为了估算开发工作量，有以下两种方式处理这个问题：

- 假设项目交付效率（每功能点工时数）是可以调整的，以此补偿处理数据运算所需工作量。
- COSMIC 方法允许对方法规则进行本地化扩展。例如，有人建议未来研究一下如何度量 ETL 处理过程中实际的数据运算符处理规模，即把数据运算符处理单独计算功能点，而不是按照常规方式将其归并到与之相关的数据移动子处理中。

注意，如果使用“本地化扩展”方式，将只能“应用于本地”，即可能无法用于行业基准对比。

多层次数据汇总

用 COSMIC 度量多维度数据库时，由于事实表是以多维度汇总的形式存储，就造成其中一些功能处理的读或写可能包含大量的兴趣对象（OOI）。开发如此大数据量的 OOI 所需工作量可能与其数量不成比例。

为了解这个事情，我们先举一个简单的例子，对第 3.2.3 节中所例举事实表的查询会生成一个报表，事实表的星型结构如图 3.5，只是没有地区/国家维度。报表形式如下，其中展示了客户“John Does & Co.”购买的每个产品的月销售额，和汇总到十个产品系列的总额，以及这一年的每个产品销售额和各系列产品总额。

图中黄色阴影单元格是在报表中出现的四种不同类型的销售数据，对应四种 OOI 类型。（报表的主要内容都是这些销售数据的重复发生。）

2015年1月-3月销售报告（\$）								
客户：John Doe & Co								
产品	1月	2月	3月	4月	5月	年度汇总
A111	1951	2145	2137					6233
A112	95	130	240					465
A113	459	634	745					1838
...								
系列1汇总	4591	4791	5342					14724
B111	495	495	520					1510
B112	321	432	543					1296
B113								0
...								
系列2汇总	1037	1345	1599					3981
...								0
...								0
系列10汇总	49	44	43					136

在第 3.2.3 节的说明中，将数据生成报表的四种 OOI 如下（注意此报表不包含区域/国家维度，并且报表可能是为这一个客户生成的，或者反复为所有客户生成。）

销售情况（产品，客户，月份） 【最低层级】

销售情况（产品，客户，年份）

销售情况（产品系列，客户，月份）

销售情况（产品系列，客户，年份）

现在虽然报表展示的数据包含四个 OOI，实际上只有两个“处理”来生成汇总数据，假设最低层级的销售数据（产品，客户，月份）已经存在。这两个汇总“处理”是：

- 将每月销售数据从产品级汇总到产品系列级的处理
- 将每月销售数据汇总到年度销售数据，不管是针对月度某产品数据还是月度某产品系列数据都是同一个处理。

（一旦选定了汇总数据的路线，则只需要两个处理：从月度汇总到年度（同时适用于产品和产品系列），再从年度某产品汇总到年度某产品系列。）

从一个 n 维星形结构的最低层级汇总数据到所有层级所需的这种“处理”的个数就是所有维度汇总步骤的总数。在这个报表案例中，处理个数=2。在图 3.5 的星形结构案例中，其处理步骤是三个，尽管整个星形结构事实表的 OOI 个数是八个。看起来数“处理”个数应该与工作量更相关，这里的工作量包括分析、设计、编码和测试生成这些数据的软件，而不是数生成数据所涉及的 OOI 个数。

注意：我们必须明确地区分：

- 此报表的 COSMIC 功能规模，包括每个 OOI 数据的生成必须被识别为一个输出；
- 生成此报表所需的工作量可能跟上述“处理”个数更成比例（或者换句话说，单位 CFP 所需的工作量更低，或生产率比一般高）。

参考文献

下面列出的所有 COSMIC 文档，包括翻译成其他语言的版本，可以从www.cosmic-sizing.org网站的下载区域获得。

- [1] Van Heeringen, H., Measuring the functional size of a data warehouse application using the COSMIC method, Software Measurement European Forum Conference, Rome (Italy), May 2006.
- [2] Santillo, L., Size & Estimation of data warehouse systems, in FESMA DASMA 2001 conference proceedings, Heidelberg (Germany), May 2001.
- [3] The COSMIC functional size measurement method, version 4.0/ 4.0.1: Measurement Manual.
- [4] The COSMIC functional size measurement method, Guideline for sizing business application software, version 1.1.
- [5] Inmon, W.H., 'What is a Data Warehouse?', Prism, Volume 1, Number 1, 1995.
- [6] Chaudhuri, S. and Dayal, U., 'An Overview of Data Warehousing and OLAP Technology', ACM Sigmod record vol. 26 (1), 1997, pp. 65-74.
- [7] Sachdeva, S., 'Meta data architecture for data warehousing', DM Review Magazine, April 1998, www.dmreview.com/issues/19980401/664-1.html.
- [8] See for example, Wikipedia on the 'Common Warehouse Metamodel', 'Metadata Standards', etc.

附录 A—术语表

本指南使用的术语如下。COSMIC 功能规模度量方法（即“COSMIC”方法）通用术语，见度量手册 [3]。

为了方便参考，以下定义中下划线标出的术语在本术语表中另有定义。

商业智能：对业务操作数据的历史、当前和前瞻性预测分析。

数据集市：对数据仓库的一种特殊扩展应用。

数据准备区域：从业务操作数据源加载到数据仓库中的数据储备，存储前经过清洗并与已有数据匹配。

数据仓库数据库：为数据发布而优化的数据源。

数据仓库：一种特定类型的业务应用，用于保存和/或展示详细数据和汇总数据，以便支持商业分析和决策。

维表/维度表：用于展示事实表的各个属性（类型）的表格。（维基百科）

维度数据存储：对数据进行汇总和运算以便提升检索速度。

ETL 子系统：从一个存储区域将数据抽取、转换和加载到另一个区域的应用程序。

事实表：存储业务事件、处理或过程结果数据的表格。事实表的定义有如下三种：

- 业务处理事实表，记录某一特定事件（比如销售事件）的数据；
- 快照事实表，记录某一特定时间点的数据（比如月底的账户详情）；
- 累计快照表，记录某一时间点的累计数据（比如某一产品的当月累计销售情况）（维基百科）。

元数据：数据属性的特征值定义。

业务操作数据源（数据库）：业务操作系统的数据库，用于抓取业务处理数据。

雪花结构：多维度数据库中表的一种逻辑布局，是一种类似雪花形状的实体结构图。“雪花结构化”是一种将星型结构的维表进行标准化的方法。（维基百科）

星型结构：一种包含一个或多个事实表的结构，其中的事实表可引用任何数量的维表（维基百科）。

附录 B—1.0 版升级到 1.1 版的主要变更

本附录汇总了 COSMIC 数据仓库指南从 1.0 版到当前的 1.1 版的原则性变更。

“MUB”是指方法更新公告，发布于度量手册两次主要版本升级之间，以对这期间提交的变更进行公示和解释。

V4.0 目录号	变更
-	一些必要的文本更新，以保证与度量手册 V4.0 一致。
-	图表编号从顺序递增改为按照章节排序，使其与其他指南的编号方式保持一致（即图 5 改为图 3.2）。
-	术语的合理化调整。ETL “程序”，“工具”或“工具组件”现在统一称为 ETL “子系统”。“工具”一词只用在终端用户软件中，比如 DSS, OLAP, EIS, 数据挖掘等等这些用于查询数据仓库或数据集市数据库的软件。
1.1	引入两种数据仓库模型（Inmon 和 Kimball）。
第 2 章	增加副标题，重新整理第 2 章结构：2.1 “概述”，2.2 “数据存储及各个数据仓库子系统”，2.3 “维度数据存储”。最后一节不描述存储或软件各子系统，而是描述存储方式，所以单独成一节。
2.1	注 1 的最后两句话被挪到 3.1 度量策略章节。
2.1	根据方法 v4.0 版对层的定义的修订，对层相关的文本进行扩充。
2.1	增加 Kimball 数据仓库模型的示意图。
2.2	删除模糊逻辑相关内容，因为指南后面的内容无需用到此概念。元数据相关文本进行了扩充，强调除了本指南提到的方式之外，其可以以多种其他形式存在并被应用。
2.3	增加雪花结构的图示。
3.1	该节进行了重写使其结构更清晰。
3.2	增加颗粒度级别和分解层级的文本描述。
3.2	图 3.1, 3.3, 3.4 和 3.6 都进行了更新，以展示组件和数据库间的数据移动。
3.2.1	最后一句关于 PDR 的内容删除，因其与第 4 章第一条重复。
3.2.3	本节案例原先描述为“我们假设数据集市的功能性用户需求是存储销售数据，事实表中存储的是最低级别的数据（即没有经过汇总的），以及在所有维度各种可能的条件组合下的汇总级数据”。第二个“在”有所误导，应该改为“对”（各种汇总级别）。增加了脚注，举例说明数据在更高一级汇总数据是有可能的，即使数据按要求存储在八个级别中。
3.2.4	本节的案例有几个错误，并且与 3.2.3 节所说的事实表的存储是用于在每个级别汇总这一情况相矛盾。因此修改了部分文本。
3.2.6	新增章节，描述 Kimball 型数据仓库的 ETL 数据集市工具这一模块，并展示了 Kimball 型的星形（物理）结构。
4	这一章目前聚焦于当估算数据仓库软件的开发工作量时，在度量了 COSMIC 功能点之后，要考虑哪些因素。在计算多个级别汇总数据的过程中，增加了一些说明文字。
术语表	增加“雪花”结构的定义，并扩充“事实表”的定义。

附录 C——COSMIC 变更请求和建议程序

COSMIC 度量实践委员会（MPC）非常愿意接受反馈、建议，以及对此指南的变更请求（如果必要的话）。本附录展示了如何与 COSMIC MPC 联系。

所有与 COSMIC MPC 的联系都应该通过电子邮件的方式发送到下面的地址：

mpc-chair@cosmic-sizing.org

非正式的反馈和建议

关于此指南的非正式建议和/或反馈，比如理解或应用 COSMIC 方法的任何困难，一般性改进的建议等，都可以通过电子邮件发送到上面的地址。消息会被登记下来，并且通常在收到后的两周内给予答复。度量实践委员会不保证对一般性意见给予答复。

正式的变更请求

本指南的读者如发现某个文字缺陷，或不足之处需要澄清，或者一些文字需要加强，那么可以提交一个正式的变更请求（“CR”）。正式的 CR 会被登记下来，并在收到后的两周内给予答复。每个 CR 将分配一个序列号，在 COSMIC MPC 的成员中循环传递，COSMIC MPC 是一个世界范围内 COSMIC 方法的专家组。他们的正常评审周期最少需要一个月，如果变更请求很难解决，可能需要更长的时间。评审的结果可能是 CR 被接受，或者拒绝，或者“未决待进一步讨论”（后面这种情况，例如本 CR 要依赖另一个 CR），结果会尽快地反馈给提交者。

只有包含了下面的所有信息，一个正式的 CR 才会被接受。

- 提交 CR 的人员姓名、职位和单位
- 提交人的详细联系方式
- 提交日期
- 关于 CR 的目的的总体陈述（如“需要改进文本……”）
- 需要变更、替换或删除的实际文字（或澄清引用出处）
- 建议增加或替换的文字
- 关于变更的必要性的充分解释

提交 CR 的表格可以从门户网站 www.cosmic-sizing.org 获得。

COSMIC MPC 对 CR 的评审结果，以及在哪个版本应用这个 CR（如果被接受了的话）的决定是最终的决定。

COSMIC 方法应用的问题

COSMIC MPC 抱歉不能回答与 COSMIC 方法应用相关的问题。有商业机构能够提供本方法的培训和顾问工作，或者支持工具。详细情况请咨询 www.cosmic-sizing.org 网站。