

# 基于熵的信息系统业务模型复杂性度量\*)

王忠杰 徐晓飞 战德臣

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

**摘要** 业务模型的复杂度决定企业信息系统的复杂度,也对信息系统的重构性能具有很大程度的影响。目前研究多侧重于代码级软件的复杂度度量,而对业务模型的复杂度则关注较少。本文首先给出了企业业务模型的分层体系结构,依据模型实体之间的依赖关系与分解关系将业务模型分解为一组基本模型单元。然后重点提出一种基于熵的模型复杂性度量方法,使用信息熵来描述业务模型的复杂性,通过计算基本模型单元的复杂度递归得到各模型实体、依赖关系的复杂性,进而综合得到模型的复杂性。最后通过实际案例验证了此方法的可行性。该方法为信息系统的设计与构造过程提供了有效的参考与决策依据。

**关键词** 信息系统,业务模型,复杂性度量,熵

## Entropy-Based Complexity Metrics for Business Models of Enterprise Information System

WANG Zhong-Jie XU Xiao-Fei ZHAN De-Chen

(School of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001)

**Abstract** The complexity of business models determines the complexity of enterprise information system, thereby impacts reconfiguration performance of the system to a great extent. Current researches focus mostly on complexity metrics of code-level software systems, and pay little attention to business model's complexity metrics. To quantitatively measure model complexity, a multi-layer architecture of business models is firstly shown, in which a business model is divided into a set of Basic Model Units (BMUs) by the dependency and decomposition relationships between business entities. Then an entropy-based complexity metrics is emphatically presented, in which the complexity of model entities, dependency relationships, and business models can be recursively calculated by calculating the complexity of BMUs. Finally, a practical case is brought forth to validate the metrics. The method in this paper has great significance to the design and implementation of complex information systems.

**Keywords** Information system, Business model, Complexity metrics, Entropy

企业处于一个多变的环境中,频繁变化的需求要求信息系统具有良好的重构性能。复杂度是决定系统重构性能的一个关键指标,因此研究信息系统的复杂性度量方法具有重要的意义。

企业信息系统是一个复杂的系统,它由标准化的管理流程、知识化的数据体系以及集成化的软件系统构成,具有分层的体系结构,可划分为业务模型层、软件模型层、可运行软件层三个层次,分别对应系统的设计、构造与运行三个阶段的模型。其中业务模型层描述了企业的管理模式与管理流程,软件模型层是业务模型在软件系统中的映射,可运行软件层则为软件模型提供了具体的运行环境和平台。每个层次又可进一步细分为一系列的子层次。传统的系统复杂性度量方法通常集中于对软件模型层或可运行软件层的复杂性度量<sup>[1]</sup>,考虑构成软件系统的基本实体(例如变量、函数、对象等)及其之

间的依赖关系<sup>[2,3]</sup>。但这些方法只能够在信息系统最终实现之后才能进行相应的度量<sup>[4]</sup>,无法提前预知系统的复杂性。

由于业务模型与软件系统之间存在着严格的映射关系<sup>[5]</sup>,业务模型的复杂度决定着软件系统的复杂度。系统设计人员在设计阶段(即模型构造阶段),通过度量业务模型的复杂度,就可以得知基于业务模型构造的信息系统的复杂度。从多个设计方案中选择具有最优复杂性的一个,使信息系统既能满足业务需求,又易于以后的重构,是设计人员面临的一项重要任务。

### 1 业务模型的分层结构

业务模型可抽象为一组模型实体及实体之间的交互关系的集合。实体之间交互关系分为两种类型:依赖关系和分解关系。依赖关系描述了同一层次内部实体之间的交互,分解

\* )基金项目:国家自然科学基金资助项目(60573086)、“十五”国家“863”资助项目(2001AA414010,2003AA4Z3210,2003AA4Z3350)、高等学校博士学科点专项科研基金资助课题(20030213027)。王忠杰 博士研究生,主要研究方向:可重构 ERP 系统、软件、重用技术;徐晓飞 博士,教授,博士生导师,主要研究方向:企业智能计算、管理与决策信息系统、数据库等。战德臣 博士,教授,博士生导师,主要研究方向:软件重构与复用、企业资源计划系统与现代集成制造系统等。

- and Logical-Task Decomposition in Multimedia Surveillance Systems. Proceedings of the IEEE, 2001,89(10):1419~1440
- Lopes R J, Lindsay A T, David H. The utility of MPEG-7 systems in audio-visual applications with multiple streams. IEEE Transactions on Circuits and Systems for Video Technology, 2003,13(1):16~25
  - Lee D, Kim N, Kim S. The MPEG-4 streaming player using adaptive decoding time stamp synchronization. IEEE Proceedings of the Ninth International Conference on Parallel and Distributed Systems, 2002. 1521~9097
  - Internet2 - QoS Working Group. Network QoS Needs of Advanced Internet Applications- A Survey. <http://www.internet2.edu/qos/>,2003-07/2003-05
  - 范圣印,李波,张光卫. 基于 Web 的带宽自适应组件化多媒体监控系统. 北京航空航天大学学报, 2003,29(10):881~884
  - RFC 1890. RTP Profile for Audio and Video Conferences with Minimal Control
  - RFC 3016. RTP Payload Format for MPEG-4 Audio/Visual Streams
  - 陈耀武,伍鹏,汪乐宇. 基于流媒体技术的网络化嵌入式视频监控系统的. 工程设计学报,2004,11(2):58~61

关系则描述了相邻层次实体之间的关联。同样,相邻层次的依赖关系之间也存在着分解关系。表1描述了企业信息系统业务模型典型的分层结构及相关实体、依赖关系、分解关系

的类型。图1给出了业务模型分层结构的一个简单的示意图。

表1 业务模型的分层结构及其实体与关联关系

层次	实体类型	依赖关系类型	分解关系类型
业务系统层	子系统	数据交换、消息传递	
业务过程层	业务过程	同步交互、异步交互	业务过程组合成业务系统
业务活动层	业务活动、业务规则	执行次序、触发次序	业务活动、业务规则组合为业务过程
业务操作层	业务操作、业务规则		业务操作、业务规则组合为业务活动
业务对象层	业务对象	组合、继承、关联等	业务操作调用业务对象
对象属性/方法层	属性、方法	使用、调用	业务对象包含属性、方法

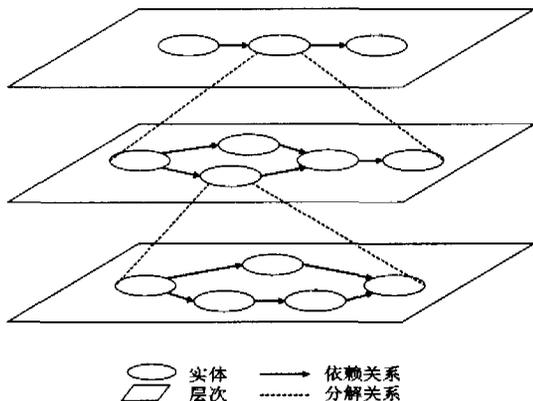


图1 企业信息系统分层结构示意图

设  $S$  表示分层的业务模型系统,  $i$  表示分层模型的第  $i$  层,  $L \leq i \leq H, i, L, H$  均为自然数。使用  $a_j^{(i)}$  表示第  $i$  层中的第  $j$  个实体,  $A^{(i)} = \{a_j^{(i)}\}$  为第  $i$  层中包含的所有实体的集合,  $A = \bigcup_{i=1}^H A^{(i)}$  为系统中所有实体的集合。使用  $d_{j,k}^{(i)} = a_j^{(i)} \xrightarrow{d} a_k^{(i)}$  表示实体  $a_j^{(i)}$  与  $a_k^{(i)}$  之间的依赖关系,  $D^{(i)} = \{d_{j,k}^{(i)}\}$  为第  $i$  层中包含的所有依赖关系的集合,  $D = \bigcup_{i=1}^H D^{(i)}$  为系统中所有依赖关系的集合。使用符号  $\omega(d_{j,k}^{(i)})$  表示  $d_{j,k}^{(i)}$  的依赖强度, 描述  $a_j^{(i)}$  与  $a_k^{(i)}$  依赖关系的紧密程度。

使用  $c_{j,k}^{(i)} = a_j^{(i)} \xrightarrow{c} a_k^{(i-1)}$  表示第  $i$  层中的实体  $a_j^{(i)}$  与第  $i-1$  层中的实体  $a_k^{(i-1)}$  之间存在的分解关系,  $C^{(i)} = \{c_{j,k}^{(i)}\}$  为第  $i$  层中的实体与第  $i-1$  层中的实体之间存在的分解关系的集

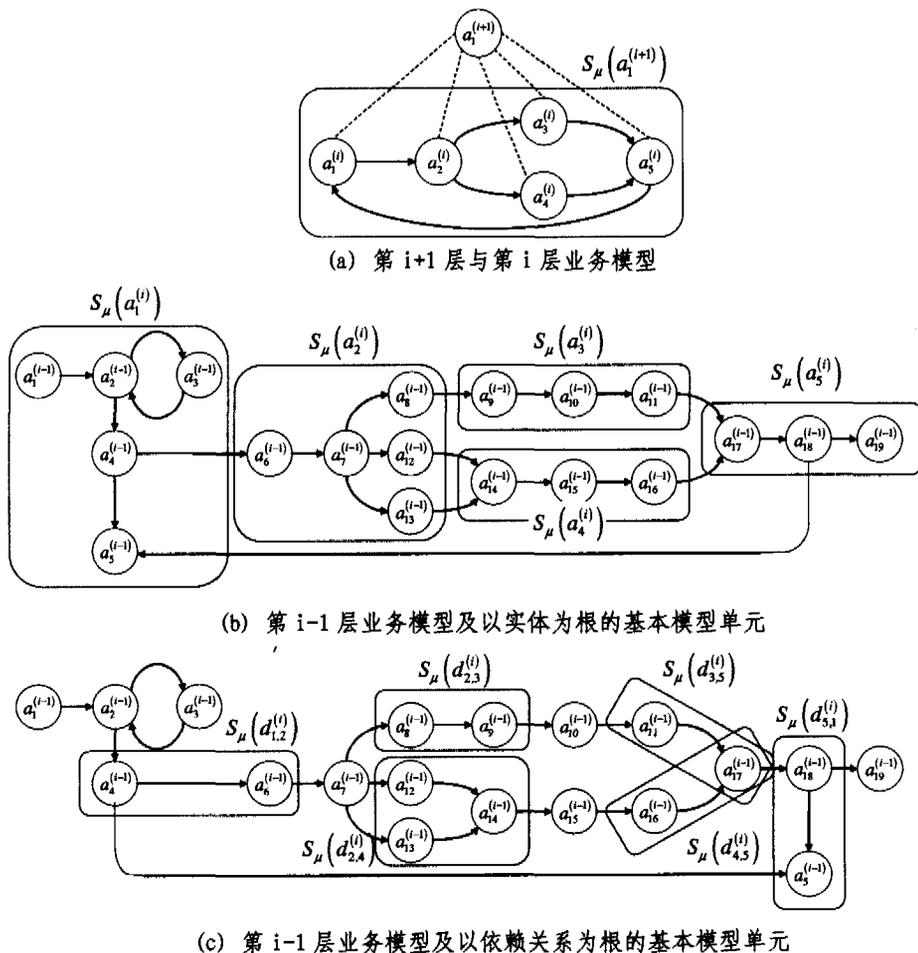


图2 多层业务模型的分解

合,  $C = \bigcup_{i=1}^H C^{(i)}$  为系统中分解关系的集合。如果存在  $a_j^{(i)} \xrightarrow{c} a_k^{(i-1)}$ , 则称  $a_k^{(i-1)}$  是构成  $a_j^{(i)}$  的子实体, 用  $SA(a_j^{(i)})$  表示实体  $a_j^{(i)}$  所有子实体构成的集合。对实体  $a(i-1)$ , 如果存在  $d_{p,q}^{(i-1)} \in D^{(i-1)}$ , 使得  $a_p^{(i-1)} \in SA(a_j^{(i)})$  且  $a_q^{(i-1)} \in SA(a_j^{(i)})$ , 即  $d_{p,q}^{(i-1)}$  为  $a_j^{(i)}$  的子实体之间的依赖关系, 则称  $d_{p,q}^{(i-1)}$  为  $a_j^{(i)}$  的内部依赖关系。  $a_j^{(i)}$  的所有内部依赖关系构成的集合表示为  $ID(a_j^{(i)})$ 。

对实体  $a_j^{(i)}$ 、 $a_k^{(i)}$  及其之间的依赖关系  $d_{j,k}^{(i)}$ , 如果  $\exists d_{p,q}^{(i-1)} \in D^{(i-1)}$ , 使得  $a_p^{(i-1)} \in SA(a_j^{(i)})$  且  $a_q^{(i-1)} \in SA(a_k^{(i)})$ , 则称  $d_{p,q}^{(i-1)}$  为  $d_{j,k}^{(i)}$  的分解依赖关系。  $d_{j,k}^{(i)}$  的所有分解依赖关系构成的集合表示为  $SD(d_{j,k}^{(i)})$ 。  $SA(d_{j,k}^{(i)})$  表示的  $d_{j,k}^{(i)}$  所有分解依赖关系的相关实体构成的集合。可知, 对  $\forall d_{j,k}^{(i)} \in D^{(i)}$ , 它要么为某一上层依赖关系的分解依赖关系, 要么为某一上层实体的内部依赖关系。

实体  $a_j^{(i)}$  称为原子实体, 当且仅当  $SA(a_j^{(i)}) = \emptyset$ , 即它不可分解。为规范化起见, 规定只有  $a_j^{(L)}$  为原子实体, 即模型最底层中包含的实体为原子实体。非原子实体称为复合实体。依赖关系  $d_{j,k}^{(i)}$  称为原子依赖关系, 当且仅当  $SD(d_{j,k}^{(i)}) = \emptyset$ , 即它不可分解。易知, 只有当  $a_j^{(i)}$  与  $a_k^{(i)}$  均为原子实体时,  $d_{j,k}^{(i)}$  才为原子依赖关系。

根据以上定义, 业务模型系统可表示为  $S = (A, D \cup C)$ 。

称  $S_\mu(a_j^{(i)}) = \langle a_j^{(i)}, SA(a_j^{(i)}), ID(a_j^{(i)}) \rangle$  为系统  $S$  中以实体  $a_j^{(i)}$  为根的基本模型单元。  $S_\mu(a_j^{(i)})$  由  $a_j^{(i)}$  所有子实体及内部依赖关系构成。特别地, 当  $i=L$  时,  $S_\mu(a_j^{(L)}) = \langle a_j^{(L)}, \emptyset, \emptyset \rangle$  为空基本模型单元。称  $S_\mu(d_{j,k}^{(i)}) = \langle d_{j,k}^{(i)}, SA(d_{j,k}^{(i)}), SD(d_{j,k}^{(i)}) \rangle$  为系统  $S$  中以依赖关系  $d_{j,k}^{(i)}$  为根的基本模型单元, 它包含了  $d_{j,k}^{(i)}$  的所有分解依赖关系及其相关实体。  $S_\mu(a_j^{(i)})$  与  $S_\mu(d_{j,k}^{(i)})$  统称为  $S_\mu$ ,  $S_\mu(A)$  和  $S_\mu(D)$  表示基本模型单元中的实体与依赖关系的集合。

基本模型单元描述了一个实体/依赖关系如何分解为下层的子实体/依赖关系, 基本模型单元的复杂度表征了其根实体/依赖关系的复杂度。业务模型系统可分解为一系列基本模型单元。在下面的例子中, 图 2(a) 描述了一个两层业务模型的第  $i+1$  层与第  $i$  层之间的分解, 图 2(b) 和图 2(c) 描述了第  $i$  层与第  $i-1$  层之间的分解。为清晰起见, 图 2(b) 和图 2(c) 中并没有给出层间的分解关系。图中的虚线框表示分解后得到的基本模型单元。

## 2 基于熵的复杂度度量

复杂度本身不可直接度量, 需要通过其他可直接度量的指标间接加以表示。很容易理解, 业务模型中包含的实体越多, 则其复杂度就越高; 同样, 实体之间的依赖关系越多, 则业务模型的复杂度也就越高。本文借鉴信息论中熵的概念, 使用信息熵来描述业务模型的复杂性<sup>[3,6]</sup>。信息熵最初由 Shannon 提出<sup>[7]</sup>, 利用统计热力学中熵的概念, 建立了对信息的度量方法。熵从量上反映具有确定概率的事件发生时所传递的信息, 与事件发生的概率有关: 事件发生的概率大, 其不确定程度就小, 则其包含的信息量就小。熵可定义为:

$$H_n(P) = - \sum_{k=1}^n (p_k \times \log_2 p_k) \quad (1)$$

其中  $P = \{p_1, p_2, \dots, p_n\}$  是事件的概率分布,  $p_k \geq 0$  且  $\sum_{k=1}^n p_k = 1$ 。

下面首先给出实体和依赖关系的复杂度表示, 然后给出

构成复杂度各部分的计算方法, 最后给出整个业务模型复杂性的度量方法。

### 2.1 原子实体/依赖关系的复杂度

原子实体是构成业务模型的基本单元, 不能或不需继续分解。因此, 可以不必考虑原子实体内部的结构, 认为其复杂度为 1, 即  $H(a_j^{(i)}) = 1$ 。同理, 原子依赖关系也不可分解,  $H(d_{j,k}^{(i)}) = 1$ 。

### 2.2 复合实体/依赖关系的复杂度

$S_\mu(a_j^{(i)})$  描述了实体  $a_j^{(i)}$  如何分解为一组下层实体及依赖关系, 因此  $a_j^{(i)}$  的复杂度可用  $S_\mu(a_j^{(i)})$  的复杂度计算得到, 即  $H(a_j^{(i)}) = H(S_\mu(a_j^{(i)}))$ 。  $S_\mu(a_j^{(i)})$  的复杂度由三部分构成: 本身的结构复杂度、包含实体的复杂度及内部依赖关系的复杂度, 可表示为:

$$H(S_\mu(a_j^{(i)})) = k_1 \times H_S(S_\mu(a_j^{(i)})) + k_2 \times H_A(S_\mu(a_j^{(i)})) + k_3 \times H_D(S_\mu(a_j^{(i)})) \quad (2)$$

其中,  $H_S(S_\mu(a_j^{(i)}))$ 、 $H_A(S_\mu(a_j^{(i)}))$ 、 $H_D(S_\mu(a_j^{(i)}))$  分别对应于上述三种复杂度,  $k_1, k_2, k_3$  则描述了三种复杂度的权值。传统的方法只考虑  $S_\mu(a_j^{(i)})$  本身的结构复杂度, 却忽略了构成  $S_\mu(a_j^{(i)})$  的实体与依赖关系都是可分解的, 它们本身的复杂度对  $S_\mu(a_j^{(i)})$  的复杂度也有影响, 因此这些方法只适合于不可分解的实体构成的系统, 即系统中只包含原子实体时的情况。

由于实体与依赖关系的可分解性, 其复杂度的度量方法可用下面两式递归地计算得到:

$$H_A(S_\mu(a_j^{(i)})) = \sum_{a_k^{(i-1)} \in S_{\mu(A)}} H(S_\mu(a_k^{(i-1)})) \quad (3)$$

$$H_D(S_\mu(a_j^{(i)})) = \sum_{d_{p,q}^{(i-1)} \in S_{\mu(D)}} H(S_\mu(d_{p,q}^{(i-1)})) \quad (4)$$

$H_S(S_\mu(a_j^{(i)}))$  的计算详见 2.3 节。

$S_\mu(d_{j,k}^{(i)})$  描述了依赖关系  $d_{j,k}^{(i)}$  如何分解为一组下层实体及依赖关系, 因此  $d_{j,k}^{(i)}$  的复杂度可用  $S_\mu(d_{j,k}^{(i)})$  的复杂度计算得到, 即  $H(d_{j,k}^{(i)}) = H(S_\mu(d_{j,k}^{(i)}))$ 。  $S_\mu(d_{j,k}^{(i)})$  的复杂度包含两部分: 本身的结构复杂度及包含的分解依赖关系的复杂度, 可表示为:

$$H(S_\mu(d_{j,k}^{(i)})) = k_4 \times H_S(S_\mu(d_{j,k}^{(i)})) + k_5 \times H_D(S_\mu(d_{j,k}^{(i)})) \quad (5)$$

$$H_D(S_\mu(d_{j,k}^{(i)})) = \sum_{d_{p,q}^{(i-1)} \in S_{\mu(D)}} H(S_\mu(d_{p,q}^{(i-1)})) \quad (6)$$

### 2.3 基本模型单元的结构复杂性

在度量基本模型单元本身的结构复杂性的时候, 不需要考虑构成基本模型单元的实体的复杂性, 可直接将其看作原子实体。借鉴文献中给出的方法<sup>[3]</sup>, 将基本模型单元  $S_\mu$  看作一条消息  $P$ , 构成基本模型单元的实体  $a_k^{(i-1)} \in S_\mu(A)$  构成  $P$  中包含的事件, 事件发生的概率通过式(7)计算得到, 即与该实体相关联的所有依赖关系的强度之和与基本模型单元所有依赖关系的强度之和的比值。

$$p(a_j^{(i)}) = \frac{\sum_{a_p^{(i-1)} \in S_{\mu(A)}} \omega(d_{j,k}^{(i)}) + \sum_{a_q^{(i-1)} \in S_{\mu(D)}} \omega(d_{j,k}^{(i)})}{2 \times \sum_{d_{p,q}^{(i-1)} \in S_{\mu(D)}} \omega(d_{j,k}^{(i)})} \quad (7)$$

文献中一般将所有依赖关系都看作是等同的, 并令  $\omega(d_{j,k}^{(i)}) = 1$ , 这无法体现出不同依赖关系之间的差别, 只能适用于原子依赖关系的情况。由于依赖关系的熵值刻画了其复杂度, 本文使用依赖关系本身的熵值作为其强度, 即

$$\omega(d_{j,k}^{(i)}) = H(d_{j,k}^{(i)}) \quad (8)$$

使用式(7)、(8)可计算得到每个实体的依赖强度, 作为该实体所代表的事件的分布概率, 并根据式(9)计算整个基本模

型单元的结构复杂度熵:

$$H_S(S_\mu) = - \sum_{a_k^{(i)} \in S_\mu^{(N)}} (p(a_k^{(i)}) \times \log_2 p(a_k^{(i)})) \quad (9)$$

### 2.4 业务模型的复杂性度量

在计算业务模型复杂度时,首先将模型分解为一组基本模型单元的集合,然后按照层次由低到高的次序,计算各基本模型单元的复杂度,并逐层向上递归计算,直到达到最高层次,最终的结果即为该业务模型的复杂度。也可抽取模型中的部分层次,对其中的实体或依赖关系进行度量,度量时只需将具有最小层号的实体看作原子实体即可。

## 3 实例

图2中给出的例子实际上是某企业ERP销售系统的局部业务模型,篇幅所限,该实例只给出了其中三个层次:子系

统层、业务过程层、业务活动层,并将业务活动层的实体视为原子实体。使用本文给出的复杂性度量方法,可得到各实体、依赖关系的复杂度熵,如表2所示。

在本例中,令  $k_1=5, k_2=1, k_3=1, k_4=5, k_5=1$ ,这是因为基本模型单元的结构复杂度在更大程度上决定了其总体的复杂度。

**结论** 业务模型的复杂性度量在信息系统设计、构造、实施与运行过程中扮演着重要的角色,是信息系统设计的重要依据,也对信息系统的重构性能具有很大程度的影响。本文给出了一种基于分层结构的企业信息系统业务模型复杂性度量方法,为信息系统的设计与构造过程提供了有效的参考与决策,使设计人员在业务模型设计完成之后,对模型进行有效的评估,为信息系统的重构打下良好的基础。

表2 销售出货子系统业务模型的复杂度

层次	实体		复杂度				依赖关系		复杂度		
	符号	含义	$H_s$	$H_A$	$H_D$	$H$	符号	含义	$H_s$	$H_D$	$H$
$i-1$	$a_1^{(i-1)}$	各项业务活动	1.00	-	-	1.00	$d_{11}^{(i-1)}$	活动间依赖关系	1.00	-	1.00
$i$	$a_1^{(i)}$	提货管理流程	2.12	5.00	5.00	20.60	$d_{12}^{(i)}$	提货-发货(下达提货单)	1.00	1.00	6.00
	$a_2^{(i)}$	发货管理流程	2.00	5.00	4.00	19.00	$d_{23}^{(i)}$	发货-计量(申请计量)	1.00	1.00	6.00
	$a_3^{(i)}$	计量业务流程	1.50	3.00	2.00	12.50	$d_{24}^{(i)}$	发货-质检(申请质检)	1.50	2.00	9.50
	$a_4^{(i)}$	质检业务流程	1.50	3.00	2.00	12.50	$d_{35}^{(i)}$	计量-出库(计量结果报告)	1.00	1.00	6.00
	$a_5^{(i)}$	出库管理流程	1.50	3.00	2.00	12.50	$d_{45}^{(i)}$	质检-出库(质检结果报告)	1.00	1.00	6.00
						$d_{51}^{(i)}$	出库-提货(回填出库数据)	1.00	1.00	6.00	
$i+1$	$a_1^{(i+1)}$	销售出货管理子系统	2.284	77.10	39.50	128.02					
$k_1=5, k_2=1, k_3=1$						$k_4=5, k_5=1$					

系统的复杂性由构成系统的实体以及实体之间的依赖关系的复杂性加以表征,与传统的方法相比,该度量方法的优点在于:

(1) 充分考虑了业务模型的分层结构,将实体与依赖关系都看作是分解的。上层实体与依赖关系的复杂度通过下层实体与依赖关系的复杂度加以度量;

(2) 可在系统的任何层次对任一实体或依赖关系进行复杂性度量,找出影响系统重构性能的瓶颈并加以改进;

(3) 可在系统的任何生命周期阶段进行,在进行下一阶段之前进行度量,修正缺陷,保证上一阶段的模型具有较优化的复杂性和重构性能。

### 参考文献

1 Snider G. Measuring the Entropy of Large Software Systems, [Technical Report HPL-2001-221]. HP Laboratories Palo Alto. 2001

2 Tran Cao-De, Lévesque G, Meunier J G. Software Functional Complexity Measurement with the Task Complexity Approach. In: Proc. of the 2nd Intl. Conf. of Vietnam & Francophone Informatics Research (RIVF'2004), 2004. 77~86

3 Kim K, Shin Y, Wu C. Complexity Measures for Object-Oriented Program Based on the Entropy. In: Proc. of the 1995 Asia Pacific Software Engineering Conf. 1995. 127~136

4 Bansiya J, Davis C, Eitzkorn L. An Entropy-Based Complexity Measure for Object-Oriented Designs. Theory and Practice of Object Systems, 1999, 5(2):111~118

5 Frankel D. Model Driven Architecture: Applying MDA to Enterprise Computing. Wiley Press, 2003

6 Abd-El-Hafiz S K. Entropies as Measures of Software Information. In: Proceedings of IEEE International Conference on Software Maintenance (ICSM'01), 2001. 110~117

7 Shannon C E. A Mathematical Theory of Communication. The Bell System Technical Journal, 1948, 27:379~423