

CIM 概述文件

2003 年 6 月



版权©2006 分布式管理任务组 (DMTF)。版权所有。

通用信息模型的价值 (为什么是 CIM?)

DMTF 技术委员会

管理企业的信息技术环境是一个很难的问题。由于每种技术 (有时候是每种产品) 都以不同的语义、术语、数据结构和协议来表述自己的管理数据, 导致这种情况更加恶化。IT 管理者大部分的时间和努力往往都花费在将这些管理数据的“竖井”融合起来, 以实现对所有管理数据实现统一的表述, 而不是解决问题或提高系统总体性能。本文件对 DMTF 在解决这些问题方面所做的努力进行了概述, 并说明 CIM 如何为企业的管理应用提供涵盖广泛的抽象层。

简介

收集管理企业计算机和网络环境所需的数据是管理难题的一部分。另一项重要的工作就是对这些数据进行标准化和组织。例如, 人们知道“OK”、“Functional”、“Operational”和“Working”都表达“状态良好”的意思, 但是, 计算机程序如何知道这些? 更糟的是, 计算机程序如何知道去哪里寻找这些信息?

不幸的是, 即使能够确定数据的位置及其语义也并没有完全解决问题。无论商业过程和服务本身是什么样的, 对于企业业务运营和管理 (包括已提供的和需要提供的) 的需求在不断增长。在这种环境下, 要求管理系统提供一些类似“风扇故障”这样的底层细节信息就显得并不重要了, 更重要的是要了解风扇故障导致了哪些服务丢失, 或哪些流程无法完成 (或必须迁移)。

在分布式环境中跨多个组件的端到端管理是一个现实问题也是一项需求。以孤立的方式管理个人电脑、服务器、子网、网络核心、储存和软件已经难以满足要求, 所有这些组件需要交互操作, 以提供连接性和服务。信息在这些组件之间传递, 管理也必须跨越这些边界。

这些是 CIM 架构解决的问题。这个架构的目的在于对组成企业系统的计算和网络元素以及这些元素之间的关系进行通用化的表述。最终解决 FCAPS 管理 (故障、配置、计费、性能和安全管理) 以及服务和企业运营的抽象和分解问题。

信息模型对设备和服务的通用和一致的语义进行定义和组织。模型的组织基于面相对象的范例 [1]——通过促进继承、关系、抽象和封装的使用, 提高管理数据的质量和一致性。

一致的信息模型是在组织内的业务和 IT 部门内以及与供应商之间成功的整合和使用管理数据所必需的基本构件。CIM 为企业和互联网管理领域提供了该信息模型。通过 CIM, 可以合作开发更先进的库存跟踪、根本原因分析和跨供应商管理客户解决方案。

CIM 模式结合了全球最好的标准, 参考了其他标准的概念 (例如互联网工程任务组 (IETF) 的标准 MIB[2]和国际电信联盟 (ITU) [3]), 并用 CIM 层次结构来组织信息。不能忽略或放弃使用其他标准, 而是将这些标准重用和映射到 CIM 中。

面向对象建模

CIM 的价值来源其面向对象[1]。对象设计支持其他“平面”数据格式所不支持的能力:

- 抽象和分类- 为了降低复杂程度, 明确定义了高级概念和基础概念 (管理领域的“对象”)。然后按管理数据的类型 (“类”), 通过确定数据的共同特征和特性 (属性)、关系 (关联) 和行为 (方法), 对这些对象进行分组。
- 对象继承- 在高级对象和基础对象下面, 可以通过子类化提供额外的详细信息。子类“继承”了高级对象定义的所有信息 (属性、方法和关联)。创建子类是为了将适当级别的细节和复杂性放在模型中适当的级别, 并提供管理信息的连续性。该对象继承可以被视为一个三角形——三角形顶部是“基础”对象, 在靠近底部时会有更多细节和类被定义出来。



- 依赖性、组件和关联- 各对象之间的关系都是非常重要的概念。在 CIM 之前，管理标准从多维数组或交叉参考的数据表中捕捉关系。对象范例提供了更丰富、更有表达力的方式来直接对关系和关联进行建模。此外，对这些关系进行命名和定义的方式描述了对象关联的语义（例如，考虑依赖性和组件关联）。可以在关联的属性中提供更多语义和信息（规定共同特征和特性）。
- 标准、可继承的方法- 对象的行为被称为方法，纳入方法后 CIM 使 IT 管理人员能够对管理数据采取措施。对标准对象行为（方法）进行定义的能力是另一种抽象方式。将标准方法与对象数据绑定起来就是封装。可以想象一下这种灵活性：应用这个标准就可以通过调用一个“Reset”方法来激活一个挂起的设备，而不用考虑硬件、操作系统或设备。

CIM 的目的和益处

简介中对建模目的进行了简要的说明（FCAPS 管理和服务抽象/分解）。但是，还需要对 CIM 的目的和用途进行更详细的讨论。

基本上所有目的和用途都来自为管理信息和服务语义定义一个统一模型的能力，以及确定与该模型相关的一切事项的能力。在不同层次的对象中进行抽象可以同时支持低级设备细节和高级服务组合。通过关系和关联，可以对（在提供功能和支持业务流程中）设备、软件和低级服务的作用进行描述。通过关联以及使用易于理解的对象层次，也可以将多个设置（配置）或统计（性能管理）与一个元素绑定起来。对象层次指示特定数据的“位置”，而关联描述的是实例的关系和适用性。

CIM 对促进数据的重复使用具有特别重要的意义，提供跨产品和发布版本的一致性信息。例如，用同类对象对机箱进行标识，不管是否是个人计算机的机箱还是高端路由器的机框。这两个例子中对机箱的基本抽象都是可行的，这与一个人无论看到什么品种的狗都会把它识别为一只“狗”一样。

在计算机和网络世界，有很多一般抽象可以跨提供商、产品和问题领域进行定义。例如，服务的概念（例如诊断和数据库）、设备（例如电源或监视器）或管理领域等。这些通用、一致的数据语义可以独

立于数据所属的知识领域或用于进行检索的传输协议。

从用户的角度来看，由于可以只编写一套管理工具和应用程序（按照明确定义的数据操作），从而控制成本。这些工具不需要随着各产品和发布版本而变更，因为这些工具基于单一的模型和一致的抽象。（当然，鼓励做出变更，以支持新的产品或管理特性，但是应尽量减少变更范围）。

例如，让我们对服务概念进行更详细的检查。所有功能都可以抽象成一个更高级的概念，该概念在 CIM 中称为“服务”。该类对开始和停止服务实例的方法以及指示服务状态以及是否正在运行的属性进行了定义。通过该抽象等级，可以定义“标准”的打印、诊断或储存管理子类。所有这些子类都可以用同一种高级方法启动和关闭。可以对所有状态进行检查。可以对易于理解的关系进行遍历以获取以获得更多信息，例如对依赖性进行分析。

CIM 的另一个目的和益处是灵活性和支持扩展。这意味着提供商或用户可以依赖于 CIM，覆盖特定的管理领域。可以对新的子类进行定义，和/或创建现有类的新实例，以描述计算机或网络环境。例如，用户可以定义他们的环境所特有的新的高级服务，然后对不同提供商提供的具体子服务间的依赖性进行规定。提供商的服务也可以用这个模型对进行定义，从而继承/使用相同的方法和属性。子服务的默认设置也可以使用和操控。

通过 CIM，工具可以专注于数据“竖井”的管理和整合。通过将模型与标准的、互操作的访问机制（例如在目录环境中，DMTF 在 HTTP 上的 CIM 操作 [4] 和 CIM XML DTD [5]、IETF 的 LDAP API [2]，或 OMG 的 Corba [6]）结合起来，可以创建一个完整的管理环境。标准的信息和标准的访问都是可得到的。DMTF 的目的在于使得多种访问和实现选项成为可能。我们的重点在于数据——可以通过各种方式实现（通过知识库和基础设施），以及各种协议和编码进行访问。

CIM 模式涉及到易于理解的抽象信息，可以映射到关系数据库、面对对象的存储、LDAP 目录和其他知识库。对于相关的知识库和访问协议，该模式被设计为“技术中立”。这是该模型的关键益处，因为没有一种实现可以适合所有环境。

结束语

企业的计算机和网络环境建模是一项非常艰巨的工作。这就是 CIM 模式的最终目标。随着技术的不断发展，这项工作将永远不会结束。因此，在编写本模型时假设了增量开发方式，采用自顶向下和自底向上的设计。展示的详细信息允许进行 FCAPS 分析（自底向上法）以及从 IETF 和 ITU 等各种其他标准体中映射最佳的详细信息。CIM 还提供了一般抽象以及合成/分解高级服务和功能的能力，允许更粗粒度的管理（自顶向下法）。不再有必须要转换、解释和规范化的数据“竖井”。通过 CIM，数据不再仅仅是在线路上传输的比特，数据本身具有了含义，并且这种含义在提供商和产

品中都可以保持一致。

参考资料

- [1] DMTF CIM Concepts White Paper and, DMTF CIM Specification
- [2] IETF RFCs
- [3] ITU
- [4] DMTF CIM Operations over HTTP
- [5] DMTF CIM XML DTD
- [6] OMG Corba

