白皮书 | 2016 年 6 月

超越屏蔽和子集构建：实现测试数据管理的价值

Huw Price
CA Technologies

ca technologies

目录

第 1 部分

简介

测试数据管理 (TDM) 可能已不是什么新鲜概念，但其往往得不到重视、被低估而且被误解，其可为企业带来的全部价值也尚未实现。许多组织和供应商仅将 TDM 视为环境问题，相当于对生产数据进行复制和屏蔽，还可能包括子集构建。然后，这些迁移的数据被视为“黄金副本”，用于整个 QA 和开发环境，而更好的 TDM 的目标变成更快地管理这种副本并满足对新副本的需求。

合成数据生成（如果真的使用）会基于逐个项目单独得到使用。这通常基于以下假设：数据生成对于单个团队或项目是良好的实践，但在整个企业范围内不可行。现代企业通常拥有若干复杂的大型数据库，其中使用不相干的工具集并以各种格式存储着数千万条记录。那么，争论在于，与尝试对这些复杂的生产数据进行配置和建模相比，复制现有数据必定更为轻松，而要生成真实的合成数据又需要进行配置和建模。

然而，此假设值得怀疑，希望实现更好的 TDM 的益处的组织应重新评估整个企业的合成数据生成以及他们存储、管理和配置数据的方式。通过正确的技术、流程和结构化团队变更，合成数据生成不仅可在时间、质量和资金方面提高效率，而且经证明比完全屏蔽生产数据更轻松、更安全。

本文从环境视角将 TDM 的常见问题与潜在益处进行对比，同时考虑组织中通常存在的各种流程、技术和团队结构[1]。

第 2 部分

纯逻辑 TDM 方法的弊端

对依赖于将生产数据用于测试和开发环境的组织而言，屏蔽和子集构建是必不可少的做法。必需进行屏蔽以满足当前数据保护立法要求，而建立生产数据子集可帮助降低高得离谱的基础架构成本。

但是，如果 TDM 从头到尾都采用屏蔽和子集构建的方法，其为企业带来的真正价值将无法得到实现。通过两个假设可以体现这一点：第一，任何现代组织收集的数据都包含个人可识别信息，而根据法律，这些数据不能以可识别的形式离开生产环境；第二，屏蔽数据用于测试和开发目的，因此必需提供足够的屏蔽数据以用于必需执行的所有测试案例和实现任何新的功能。

根据这两个假设，首先说明为何屏蔽事实上可能不是用于配置适合测试和开发环境的安全数据的最安全或最有效的方法。然后将考虑典型组织实施的团队结构、技术和流程，论证除生产数据通常的管理方式之外，数据极低的覆盖率也意味着，如果仅依赖于屏蔽和子集构建，项目延迟、预算超支和生产缺陷的常见痛点将永久存在。

屏蔽并非简单的解决方案

首先，安全地复制和屏蔽用于测试和开发环境的生产数据是整个复杂 IT 资产中更为简单的方案 — 这一观点可能受到质疑。事实上，要在屏蔽生产数据的同时维护引用完整性通常比数据建模更困难，而有效地屏蔽则需要先对数据进行配置。而且，作为一种妥协，数据的某些复杂方面通常并未被屏蔽，因此在很多情况下，屏蔽可能不是最安全的方案。

当屏蔽用于非生产环境的数据时，即使屏蔽了敏感内容，也必须保持原始数据的列间关系。简言之，这表示需保持引用完整性。然而，说复杂点，这意味着需保持复杂的列关系。例如，如果有一个计算两列或更多其他列总和的“总和”字段，这些列必须在屏蔽数据中保持一致。最复杂的是时态数据关系和因果关系，例如基于时间的总和，这两种关系要实现一致的屏蔽极其困难。

此外，这些关系不仅需要在数据库（系统内）中实现一致的屏蔽，而且对具有多种数据库类型和复杂的组合应用程序的典型大型组织而言，还需要跨数据库（系统间）保持这些关系。数据越复杂，越难保持关系，而且，数据更容易被破解，因为可关联的信息更多了。

通常，数据屏蔽主要集中于内容、系统间和系统内关系，以便保持引用完整性。保持内容、列关系和时态关系的复杂性通常同时意味着某个方面被遗忘，例如，两个列之间的时态关系通常被忽视。尽管从法律或技术角度看这不是敏感内容，但保留的这些信息可通过与数据外部信息相关联来识别敏感信息。

以一组屏蔽的事务日志为例，攻击者知道某个人在某时进行了一定量的事务。尽管在已屏蔽的敏感信息中找不到这些信息，但时态信息仍可能出现在屏蔽数据库中，因为要一致地屏蔽内容、数字总和及事务时间极其困难。在第一个实例中识别敏感信息后，便可跨数据库和系统地识别这些信息，因为已保留了系统间和系统内完整性。实际上，这将对数据去匿名化。

即使已对复杂的数据进行配置和屏蔽，仍然不能将其视为安全的数据，因为在数据更复杂的方面仍保留着信息。例如，攻击者可通过将系统内、系统间和因果信息与快照对比等时态法相结合，从而识别事务日志的因果瀑布效应。然后可推断在操作过程中数据经历了怎样的更改，还可推断可能出现的任何因果瀑布效应（例如任何触发数据库的触发因素）。再次强调，屏蔽数据的强度取决于其最薄弱的环节：识别此信息后，攻击者便可跨列间关系发起攻击，并解密商业敏感要求，或者，更严重的是，解密个人可识别信息。

思考这样的假设：任何组织的数据实际上都包含某些个人可识别信息，屏蔽不是将数据配置到非生产环境的安全或有效的方式。在各组织对彻底保护其数据的需求不断增长的情势下，此情况尤为突出：因数据泄露导致的平均罚款在 2013 年增加了 13%，高达 350 万美元 [5，Ponemon，2014 年]，而即将出台的《欧盟一般数据保护条例》(EU General Data Protection Regulation) 旨在使现行立法更具可执行性，现行立法禁止出于任何原因使用个人信息，出于收集数据的原因而使用的情况除外。

进一步思考 Symantec“State of Privacy Report 2015”如何发现数据安全顾虑可确定 88% 的欧洲消费者的购物场所和方式，而使用非生产环境中的数据的风险不会单独存在。建立将生产数据迁移到测试和开发环境的法律风险后，我们可以来考虑做这项工作的效力了。

人员

数据依赖

组织安排一个中央团队来负责 TDM 的情况并不多见，通常都是不同的单个团队负责管理、查找和生成自己的数据。因此，各团队在自己的测试或开发环境中相互独立地开展工作，但他们往往依赖于相同的数据源。这种缺少集中化和协作的情况导致数据依赖约束，如果一个团队更改数据库，每个团队都会受到影响。随着测试无任何明显原因地失败，接踵而至的是沮丧、延迟和返工，各团队无法确定这种情况是否由代码缺陷导致，或者是否为数据错误。各团队通常还缺乏对测试数据进行版本控制或参数化的能力，也无法在其他团队调用数据库后对数据库进行回滚操作。

继而，SDLC 被进一步视为一系列线性阶段，一个团队完成某些工作，然后传递给下一个团队。因此，由于“上游”延迟，开发和测试团队通常还会遇到项目停滞不前的问题。他们需等待数据变得可用，或者等待其他团队完成规范和数据馈送，因此大量 SDLC 时间可能花在了等待数据上。

缺乏并行化与目前推动许多组织实现持续交付的需求（其中，团队在每个 SDLC 阶段都需要数据）形成了强烈的反差。例如，这样无法在高效地对旧版系统执行生命终期开发的同时针对新版本使用现有数据。各团队只能进入新的开发环境，但该环境中没有所需的数据。

技术

系统依赖

除数据依赖之外，硬件和系统约束也是现代测试和开发团队的一贯痛点。过去二十年来，应用程序日趋复杂，无论在组织内部还是外部都与其他系统建立了越来越多的依赖关系。这通常会在希望测试某个系统时造成障碍，因为服务可能不稳定或不可用。测试和开发团队表达的另一个顾虑是缺乏完整的环境。各团队可能会发现其他团队对他们希望使用的环境具有优先权，或者可能无法获得正确的数据，因为其他团队正在使用。尽管这些约束起初可能不会表现出与数据的直接关系，但不久后便会说明为何更好的 TDM 是克服这些约束的必要步骤。

数据存储

现代组织存储着大量生产数据。维护这些数据库的副本并在开发设备上运行它们既昂贵又缓慢。存储这些数据会造成巨大的基础架构成本，包括硬件、许可和支持成本。而且，服务器需满足越来越多的要求，它们需要同时向若干作业提供海量数据，而且还需要运行打开的连接，支持打开的文件并处理数据注入[2]。

除非重新评估数据的管理方式，否则不可能降低这些成本。普通企业收集和存储的数据量每年翻一番[3]，“大数据”的出现意味着各组织现在都是以 PB 而非 TB 为单位来谈论数据。所以，数据存储已成为 IT 预算增长最快的部分之一，并且有迹象表明数据存储行业目前可能已获得高达 20% 的年度增长[4]。因此，各组织应考虑生产数据的每个副本是否都有存在的必要 — 假设某些组织趋向维护单个数据库的多个副本，答案可能是否定的。

数据挖掘和配置

若没有半自动化或全自动化技术，数据发现将是希望按时交付完全经测试的软件的团队所面临的最大挑战之一，尤其是在敏感开发环境或持续交付框架中。测试人员可能将一半的时间用于查找数据，从而迫使他们在运行每个测试以防止耗费成本的缺陷进入生产环境与按时交付软件之间做出不情愿的妥协。

在不受控的电子表格中不一致地存储数据（例如几乎没有交叉引用或索引），而非在集中化的数据仓库或存储库中进行存储，会使这种情况变得更糟。而且，数据库通常未经正确地记录，各组织往往缺乏包含测试数据属性和相关 SQL 查询的中心词典。因而数据发掘和分配受到影响，团队无法基于标准化模板或形式或者根据他们所需的特定条件来请求数据。结果，他们常常必须手动查找适合的小型数据集，以便满足各个测试案例和要求 — 这是一个费时且易于出错的过程。

缺少自动化数据配置和挖掘工具还会增加违规风险。将数据存储在不受控的电子表格中时，如果数据没有恰当的列字段，或者如果所有相关字段已满，则敏感信息可能出现在任何地方，例如在备注列中。若无法搜索特定字段，则不太可能找到这些信息并因此可能使这些信息进入非生产环境。这将违法数据保护条例，该条例规定只能出于收集数据的原因而使用数据。违规将造成平均达 350 万美元[5]的高额罚款，并可能严重损害公司的盈亏。

流程

生产数据并非“黄金副本”

由于无法查找适合测试案例和要求的数据，我们在非生产环境中使用生产数据时面临的最大问题是无法处理组织可能收到的大部分数据请求，尤其是那些来自开发环境的请求。如上所述，测试和开发团队在每个开发阶段都需要数据。经过充分考虑，“黄金副本”数据库必须包含可用于重复进行

测试的标准数据集，并且必须包含满足每个可能的测试所需的数据。而且，数据还应保持最新状态，类似于生产数据，包括“不良数据”以及所有之前的数据。生产数据仅满足其中两个条件：类似生产数据，可能包含所有之前的数据。那么，它不能称为“黄金副本”。

很多生产数据都非常相似，并涵盖“一如往常的”事务，按性质经过审查以排除将导致系统崩溃的数据。因此，生产数据不包含不良数据，而且在测试过程中不会处理新情景和消极路径。然而，正是这些意外结果、离群值和边界条件通常导致系统崩溃，开发和测试的目标应该是测试边界情况、消极路径和意外情景。如果仅依赖于抽样方法，缺陷将进入生产环境，他们将多花 1000 倍的成本进行修复[6]，并多耗费 50 倍的时间来解决问题[7]。

而且，鉴于 IT 团队面临的环境变化和不断改变的业务需求，从生产环境抽样的数据几乎从来都不是最新数据。由于并非独立于环境来存储数据，当环境改变时，必需刷新系统或升级版本。这可能调用已根据多个生产数据源构建而成的数据和情景，而且可能丢失有用的数据集。然后又必需经历繁琐的过程手动重新创建这些数据。此类系统刷新往往也十分缓慢。例如，我们曾见证与我们合作的组织花了长达 9 个月的时间才完成此过程。

手动创建数据可能实现短期修复，并允许执行手头的即时测试案例。然而，由于在创建这些数据时考虑了特定要求或测试案例，数据几乎马上就会变得过时。货币汇率或交易模式就是一个不错的例子。在这里，数据每天都会变得过时，即表示手动创建的数据无法重新使用，因此通常“被销毁”。然后我们必须费力地为每个测试创建新数据，从而造成逐渐增加的项目延迟，同时测试滚动到下一个冲刺或进入开发生命周期的下一个阶段。

如果跨多个数据库的有效屏蔽需要先配置数据，并且仍然不能保证安全性或提供优质数据，那么会引发下面的问题：成功配置生产数据后，为何组织不只采用合成方式来生成所需数据？

第 3 部分

理想的替代方案 — 人员、流程和技术

更好的 TDM 策略涉及采用结构化、至关重要的集中化方法来管理企业范围的数据。凭借正确的技术，这不仅能够解决上述许多痛点，而且事实上还是配置测试和开发环境数据的更经济、更高效且更轻松的方法。

将有用数据存储为环境外部的资产并按需将数据推入环境，可消除 TDM 的环境问题，然后需要考虑的是如何将数据推出来。反过来，通过集中化的数据存储方法可高效地执行此操作，其中，数据建

模为可重用资产并可作为精确的子集按需提取。不使屏蔽和子集构建成为必要的操作，合成数据生成因此体现出一个战略目标，该目标随后与更广泛的 TDM 策略相结合，从而可以在预算内按时交付经完整测试的软件。

技术

自动化数据配置

已经广受争议的是，要有效地屏蔽生产数据，就必须先配置数据，但是屏蔽的数据也并非完全安全，并且无法提供测试和开发环境所需的数据。然而，自动化技术的存在确实可减少跨复杂 IT 格局配置数据所需进行的工作。要配置数据，必需先“注册”数据，从而收集尽可能多的元数据。这些元数据包括表格名称、列名称和列类型等，存在于非关系型数据库系统中，形式包含大型机系统的抄写本和固定宽度或带分隔符的平面文件的映射文档。

执行注册后，即可应用基于数学的数据发现算法。例如，通过 CA Test Data Manager（以前称为 Grid-Tools’ Data Maker），将首先针对单个方案执行此算法，并识别个人可识别信息 (PII) 和逆向工程数据库关系（如有必要）。执行此算法后，各系统便可集中在一起。在这样做的过程中，CA Test Data Manager 使用立方体视图对数据中最复杂的关系进行配置，并创建多维数据集，其中“立方体”的每个维度代表数据的属性。组织通过此配置过程可准确了解存在哪些数据，数据存储在哪里，以及识别功能覆盖中的任何缺口。

合成数据生成

准确了解存在哪些数据以及确定测试环境需要的其他数据之后，便可自动生成任何缺失的数据。由于每个真实世界都可视为另一个数据点，数据可经过建模和创建，覆盖 100% 的功能变体。此类数据包括前所未有的未来情景，以及“不良数据”、离群值和意外结果。这样实现有效的负面测试和新的系统或子系统的开发。它提供系统化的测试方法，因此，测试人员脑海中可能不会出现的意外结果和情景将不会造成系统崩溃，同时在缺陷进入生产环境之前便被检测出来。

如果没有足够的数据用于重复测试，也可采用自动化技术创建海量数据。CA Test Data Manager 提供可直接与 RDBMs 或 ERP API 层搭配使用的自动化工具，以处理功能允许的速度尽可能快地生成数据。不断膨胀的脚本可使组织拥有的数据量加倍，与基础架构能够处理这些数据的速度相同。总之，这种自动化技术允许快速创建类生产数据，以及运行测试（包括负面测试）所需的所有数据，以及运行重复测试所需的足够数据。

集中化数据管理

通过前述的技术改进，组织已经走完了本文中定义的“黄金副本”（请参阅“生产数据并非黄金副本”部分）建立的大部分道路。通过进一步将建模为可重用对象的数据存储在中央测试数据仓库中，他们还可培养快速按需识别用于测试和开发环境的特定数据子集的能力。

数据克隆

在“测试集市”或测试数据仓库中将数据建模为对象，并且建立包含数据资产和相关查询的词典后，便可对特定数据子集进行识别、克隆，并推出至测试和开发环境。

例如，CA Test Data Manager 的数据克隆模块从多个互相关联的生产和开发系统中提取出小型相干的测试数据集，替代了缓慢而成本高昂的大型复杂数据库复制和迁移的过程。提取、复制和仅提供所需数据的能力意味着组织不再需要维护生产数据库的各种完整副本。

拥有集中化数据仓库并能够克隆数据，可进一步消除团队之间对数据的依赖，并将数据的配置和使用分离。这意味着数据可经过克隆，然后并行交付给多个团队，这样消除了花在等待“上游”数据变得可用的延迟时间，并防止各团队更改数据时相互造成负面影响。

通过将数据建模并集中存储为可延展并可重用的对象，还可轻松再现错误和有趣的情景。在错误报告中显式提取数据后，采用灵活、快速的克隆技术可重复复杂的罕见测试，并且不会耗尽数据。这在执行数据刷新时尤其有用，因为这意味着无需合并数据，也不会丢失有趣的数据集。

硬件约束

结合虚拟化工具包时，合成数据生成还有助于克服硬件和系统约束。这可模拟消息层，并使用系统的元数据准确配置和生成真实的服务消息（包括 Soap、REST 和 MQ 文件以及平面文件）。这里的自动化数据生成引擎基于虚拟机运行，以填充真实的消息响应，例如请求/响应对。

整个虚拟机的虚拟化意味着可创建多个开发环境。这表示即使当相互依赖的组件不可用时各团队也可在环境中工作，并防止上游延迟，同时还可虚拟化成本高昂的旧系统和硬件以便进行测试。

流程

并行开发和可重用性

除环境问题外，根据特定数据属性发现和克隆数据的能力也可解决 TDM 的另一个主要顾虑，即如何高效地将数据配置到每个测试人员或测试团队。该功能允许按需请求、共享和重用数据。

测试人员和开发人员可通过对集中化、基于 Web 的“按需测试数据”门户（例如，CA Test Data Manager 随附的门户）的访问权限，精确请求他们完成手头上的任务所需的数据。提交特定条件（如测试数据属性）后，该门户会向批量作业引擎发送作业，然后该引擎将从后端系统中查找恰当的数据，或者将克隆数据并将其推送出去。这样便无需手动搜索数据，也无需手动创建数据，从而大幅减少执行数据请求所花的时间。

问题的形式越标准化，就越能让各团队更好地重用对方的工作。例如，如果您拥有之前创建的合成数据，您可以参数化输入并通过门户中的下拉列表显示此输入，从而允许每个人都能请求数据，即使测试案例有所不同。除测试数据之外，还可存储数据创建框架、单元测试、虚拟资产以及自动化脚本，并在未来的工作中将它们用作构建块。

版本控制

强大的版本控制可实现在预算内按需持续开发新系统所需的并行度。例如，CA Test Data Manager 的测试数据仓库允许团队从存储库复制数据（实际上是“继承”数据），并使指针向后指向之前的版本。这样可实现跨多个版本的数据演变，因为它为特定团队锁定数据，同时允许数据轻松回滚或前滚，以及与不同版本协调。当一个地方发生改变时，将上下影响各版本，而原始版本保持不变。假如某个团队需要向整个数据库添加一个新列：通过 CA Test Data Manager，如果他们具有硬编码父项，则可查找所有链接的子项并设置默认值，或者可使用序列或标准功能生成数据。

人员

最后，团队的结构变更通常可补充这些技术和程序上的改进，并可进一步帮助解决所讨论的环境痛点。专门团队的 TDM 集中化与是否具有中央数据存储、管理和配置资源有关，这些条件可以更高效地满足企业需求。此团队可能拥有数据配置和数据管理资源，并可进一步负责构建新数据和配置数据（如有必要）。

这样不仅可帮助避免团队之间的数据依赖约束，而且还意味着可通过一次传送来报告错误。因此可实现严格的质量监管，同时可将数据所有权集中化到 IT 安全团队下。CA Test Data Manager 的“按需测试数据”门户提供的动态形式的构建支持此功能，因为它已超越基于角色的访问权限，并且仅将敏感数据配置到请求数据的授权人员。

第 4 部分

参考

1 实际上，在非生产环境中使用生产数据的弊端已经在其他地方探讨过。例如，请参阅 Huw Price 的“Reduce Time to Market with Test Data Management”以及“How better Test Data Management is the only way to drive Continuous Delivery”

2 Jacek Becla 和 Daniel L.Wang，Lessons Learned from managing a Petabyte，P.4. 于 2015 年 2 月 19 日检索自 http://www.slac.stanford.edu/BFROOT/www/Public/Computing/Databases/proceedings/

3 Lessons Learned from managing a Petabyte

4 http://www.computerweekly.com/feature/Meeting-the-demand-for-data-storage

5 http://www.ponemon.org/blog/ponemon-institute-releases-2014-cost-of-data-breach-global-analysis

6 http://benderrbt.com/Bender-Requirements%20Based%20Testing%20Process%20Overview.pdf

7 http://www.softwaretestingclass.com/why-testing-should-start-early-in-software-development-lifecycle/

第 5 部分

CA Technologies 的优势

CA Technologies (NASDAQ: CA) 致力于提供有助于客户管理和保护复杂 IT 环境以支持敏捷业务服务的 IT 管理解决方案。企业组织利用 CA Technologies 软件和 SaaS 解决方案加快创新，转变基础架构并保护从数据中心到云的数据和身份信息。CA Technologies 致力于确保我们的客户通过使用我们的技术实现所需结果和预期业务价值。有关我们客户成功案例的更多信息，请访问 **ca.com/customer-success**。有关 CA Technologies 的更多信息，请转到 **ca.com/cn**。

第 6 部分

关于作者

Huw Price 拥有近 30 年的丰富经验，曾担任多家美国和欧洲软件公司的首席技术架构师，并曾为许多跨国银行、主要公用设施供应商和保健提供商提供高水准的架构设计支持。Huw 被 QA Guild 推选为“2010 年度 IT 总监”，他曾投入数年时间专门研究测试自动化工具，并且已推出多款创新产品，重新精心打造了适合软件行业使用的测试模型。目前他经常在一些知名的国际活动上担任演讲嘉宾，他的文章也常见于多本杂志，例如 Professional Tester、CIO Magazine 以及其他技术出版物。

Huw 最新创立的 Grid-Tools 公司于 2015 年 6 月被 CA Technologies 收购。近十年来，大型组织处理测试战略的方法已发生巨变。通过 Huw 充满远见的方法和领导力，企业在测试中引入了以数据为中心的强大方法，并实施了一些由 Huw 自己构思的新概念，例如“数据对象”、“数据继承”和“中心测试数据仓库”。