第3章

工业大数据治理

3.1 工业大数据治理产生的背景

毫无疑问，在大数据时代，工业大数据已经成为企业的重要资产和核心生产要素。众多企业将大数据思维贯穿到企业生产经营活动的各个层面和各个领域，企业决策行为向数据驱动的创新模式转变，数据采集、数据处理、数据存储、数据传播、数据分析成为企业的常规决策流程和日常工作内容。数据资产对企业的重要性不言而喻，大部分企业都将大数据作为企业发展战略的重要组成部分并进行相关的技术研发和创新应用。然而在实施过程中，大数据管理存在诸多弊端，严重影响了大数据的应用成效。

1. 工业大数据质量参差不齐

工业系统的精密可控对工业大数据质量提出了完整性、一致性、可靠性的要求。而实际情况是，由于采集技术和管理流程的限制，工业大数据的质量普遍低下，数据缺失、数据异常、数据不一致的情况频繁出现。同时，随着大数据资源由企业内部的物联网数据向外部的互联网和社交媒体数据的扩展，数据类型的多样化和数据关联的复杂化进一步加大了数据质量的差异。

2. 工业大数据孤岛普遍存在

目前，一方面，由于大部分企业缺乏整体规划，信息系统的封闭式开发模式造成数据资

源散落在不同的系统架构中，缺乏有效的集成，尤其在横向的不同信息系统之间以及纵向的信息系统与工业系统之间普遍存在明显的数据壁垒，形成了众多的数据孤岛；另一方面，随着企业边界向外部环境的拓展与延伸，工业大数据的来源更加复杂多样，大数据的多模态、高通量和强关联等特性进一步增强，工业大数据的多源异构性使企业所收集到的数据较为独立和分散，严重影响了大数据资源在整个产业链的流通性，进而限制了工业大数据应用的深度和广度。

3. 工业大数据缺乏标准化

我国企业工业大数据的标准化工作仍处在起步阶段。由于缺乏数据标准，“信息孤岛”的数据标准不统一，缺少全局的数字字典编制，不同生产环节的数据资源难以对接和共享，数据资源在不同信息系统中重复定义和存储，数据出现大量冗余和歧义，影响了数据分析结果的可靠性。

由此可见，一个真正的数据驱动型企业应该非常清晰地掌控何时何地运用何种数据进行何种分析从而做出科学合理的决策。数据质量是保证企业管好用好大数据的重要前提，而数据质量的保障需要大数据治理的有力支撑。

3.2 工业大数据治理的概念

大数据治理是为了确保大数据质量而提出的一种数据管理概念，包含用于管理企业大数据资产的技术、流程和策略。简单地说，大数据治理的目的是“把散落在产品全生命周期不同环节、结构异质、语义混乱、体量庞大、溯源不清的看似杂乱无章的多个数据孤岛通过梳理、转换、整合等手段，归整为有序组织的、语义明确的、血缘清晰的、流动可追溯的一体化数据资产”。经过治理的大数据资源，一方面，具有明显的数据完整性、一致性、可靠性和时效性，可以直接进行数据分析和数据决策；另一方面，数据治理具有可扩展性，不会随着数据量的不断增加和新数据的持续加入，增加数据管理的工作量。

工业大数据治理确保了工业大数据在产品全生命周期的优化、共享和安全使用。有效的工业大数据治理计划可通过改进决策、缩减成本、降低风险和提高安全合规等方式，将价值回馈于业务，并最终体现为增加收入和利润。有效的工业大数据治理能够促进工业大数据服务于创新和价值制造，有助于提升组织的工业大数据管理和决策水平，并能够产生高质量的数据，增强数据可信度，降低成本，提高合规监管和安全控制，并降低风险。

3.2.1 大数据治理的概念

大数据治理包含很多相关概念，概念之间的关系比较复杂。本节重点介绍“大数据”“数据治理”“大数据治理”的概念含义和概念间的关联关系。图 3.1 展示了大数据治理相关

概念的逻辑关系及演化路径，将大数据治理的相关概念分为三类，分别是基本概念、衍生概念和概念组。本节要介绍的“大数据”属于基本概念；“数据治理”和“大数据治理”属于衍生概念，由基本概念层中的“数据”“大数据”与“治理”衍生而来；“大数据治理、数据治理”属于概念组，由衍生概念中的“数据治理”与“大数据治理”组合而成，用来体现大数据治理与数据治理间的关联关系。

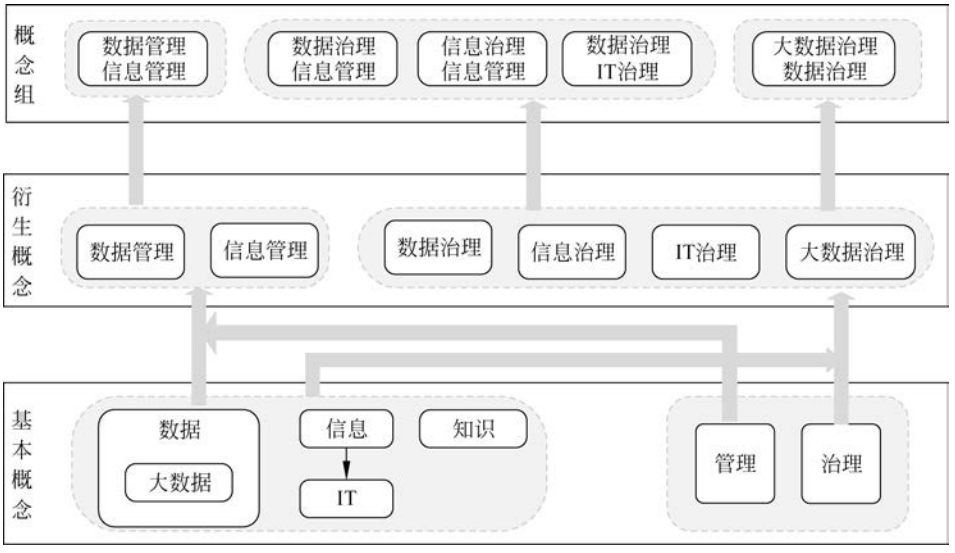

图 3.1 大数据治理相关概念的关系

1. 大数据

“大数据(Big Data)”这一概念最早由 20 世纪 80 年代著名未来学家阿尔文·托夫勒提出，他将“大数据”称为“第三次浪潮的华彩乐章”，但由于受到当时信息技术的限制，这种局面直到 2009 年才逐渐出现。

目前，业内统一认识到大数据具有以下特征：数据量大、生成和处理速度快、多样性、价值大但密度低。根据大数据特征，一般可将大数据分为以下三种类型：

1）企业数据

企业数据主要来源于企业的应用系统，比如客户关系管理系统中关于客户、产品、财务、售后服务等信息，供应商关系管理系统中关于供应商的相关信息，产品生命周期管理系统中与产品设计、产品工艺、产品生产等相关信息，ERP 系统的事务性信息及企业其他相关应用系统(比如集团协同办公平台)数据等。这些数据一般都是以结构化数据的形式存储在关系型数据库中，当然也有部分数据是以文档、图片、视频等非结构化数据以文件形式存储于文件系统中。

2）机器对机器的数据

机器对机器技术，简称 M2M，指设备通过无线或者有线的方式与其他设备进行通信。机器对机器的数据主要是由智能仪表、制造传感器从生产设备、生产车间、商业建筑中所采

集的数据或者是设备自身产生的工作日志等，属于过程性数据，这些数据一般都是以半结构化的形式存储的。需要说明的是，虽然机器对机器的数据量很大，但一般能用于决策的数据估计不到1%。

3）Web和社交媒体数据

随着信息技术的广泛应用，人们普遍使用Web和社交媒体（比如博客、微博、论坛等）来进行意见、见解、观点和经验的传递与分享。Web和社交媒体数据每天都在通过社交媒体源源不断地产生，这些数据一般以半结构化或者非结构化形式存在。

2. 数据治理

数据治理的实践早在20世纪90年代就开始了，例如IBM在1993年就开始了数据治理的探索，通过不断加以完善，目前数据治理在实践方面已经卓有成效。在理论研究上，由于数据治理是一个新兴的研究领域，所以目前的研究成果还不是很多。

到目前为止，学术界对于数据治理还没有达成共识和形成一个确切一致的定义。IBM数据治理委员会给出的定义为：数据治理是针对数据管理的质量控制范围，它将严密性和纪律性植入企业的数据管理、利用、优化和保护过程中。DGI（Data Goverance Institute）给出的定义为：数据治理是指针对信息相关过程的决策权和职责体系，这些过程遵循“在什么时间和情况下、用什么方式、由谁、对哪些数据、采取哪些行动”的方法来执行。DMBOK给出的定义为：数据治理是指对数据资产管理行使权力和控制的活动集合（如计划、监督和执行）。上述定义非常简洁和抽象，为了方便理解，下文将从数据治理的核心、职能、目标以及遵循的过程和规范四方面来解释。

1）数据治理的核心

虽然数据治理的定义很多，但数据治理的核心是数据资产管理的决策权分配和职责分工，这一点在学术界已基本达成共识。数据治理不涉及具体的管理活动，而是专注于通过什么机制才能确保做出正确的决策。做出正确决策的有效核心机制正是决策权分配和职责分工，因此数据治理的核心就是上述内容。

2）数据治理的职能

从决策的角度，数据治理的职能是“决定如何做决定”，即数据治理必须回答在数据相关事务的决策过程中所遇到的问题，即“在什么时间和情况下、在哪些领域、由谁、对哪些数据、做出哪些决策”；从具体活动的角度，数据治理的职能是“评估、指导和监督”，即评估数据利益者的需求，以达成一致的数据资源获取和管理的目标，通过优先排序和决策机制来指导数据管理的发展方向，然后根据目标和方向来监督数据资源的绩效。

3）数据治理的目标

通过数据治理建立一套完善的数据资产管控体系，确保统一的数据来源，确保数据始终处于规范化、标准化的状态，降低数据集成、管理、维护的成本，从而达到数据治理的目标，提升信息化能力、提升业务运营效率，实现数据资产价值的最大化。

4）数据治理遵循的过程和规范

“过程和规范”在上述定义中出现多次，过程主要用于描述数据治理的方法和步骤，是为了加强对数据的流程化管控，分别有数据业务上的控制、数据技术上的控制及数据逻辑上的控制。规范主要用于约束数据治理的过程，确保数据治理具有较强的严密性和纪律性，使企业的数据满足行业标准，符合国际、国家的法规等。

综上所述，数据治理的本质是对企业数据进行管控和利用，促进数据和服务紧密地结合，实现数据的内在价值，从而为企业创造经济价值。

3. 大数据治理

大数据治理是在大数据兴起以后才逐渐发展起来的，目前该领域的研究成果很少。大数据治理这一概念是根据大数据的特性，在数据治理的基础上进行扩展定义的。

目前，业界比较权威的“大数据治理”定义是由桑尼尔·索雷斯在《大数据治理》一书中给出的。大数据治理是广义信息治理计划的一部分，即制定与大数据有关的数据优化、隐私保护与数据变现的政策。可将大数据治理定义分解为以下六方面进行解释：

（1）信息治理机构必须将大数据治理整合到信息治理框架中，实施全方位信息管理。

（2）大数据治理需要识别使用大数据的核心业务流程和关键政策。

（3）大数据治理必须对元数据、主数据、数据质量、数据生命周期进行优化。对于元数据，需将因大数据新增的元数据与其所在组织的元数据库进行整合；对于主数据，将有关大数据整合到主数据管理环境中；关于数据质量，包括数据概要分析、数据审核、数据修正及数据整合；关于数据生命周期，需根据业务需求和规则，来决定对数据的删除、存档操作。

（4）大数据的隐私保护非常重要，大数据治理需识别敏感数据，并制定有关使用政策。

（5）大数据必须变现，使公司具备将大数据转换为现金的能力。

（6）大数据展现了跨功能的自然冲突，因此大数据治理必须能够协调多种跨功能的冲突性目标。

综上所述，该定义明确了大数据治理应该重点关注的领域，如大数据的优化，大数据的隐私保护以及大数据的变现；明确了大数据治理需要协调各个职能部门来制定策略；明确了大数据治理必须整合到信息治理框架中。

4. 大数据治理与数据治理关联关系

大数据治理与数据治理，从字面意义上可以发现，两者的本质都是治理，只是治理对象不同，因此治理对象之间的关系就决定了大数据治理与数据治理的关系。

大数据的本质是数据，是传统数据的一个新阶段。类似地，数据治理是大数据治理的基础，大数据治理是数据治理的新阶段。即数据治理的方法论，比如数据治理的原则、范围、框架和成熟度模型，只要适用于大数据特性，都能应用到大数据治理中。然而由于两者的侧重点不同，数据治理提供对数据的管理、应用框架、策略和方法，以确保数据的准确性、

一致性和访问性，而大数据治理则是为了发挥数据的应用价值，通过优化和提升数据的架构、质量和安全，推动数据的服务创新和价值创新，因此需要对大数据治理进行适当调整和扩充，包括：

（1）对大数据治理组织架构的改进与升级；

（2）大数据中新增元数据与原有元数据库的集成；

（3）大数据治理的隐私保护，即大数据的加密与屏蔽；

（4）大数据的质量管理；

（5）大数据生命周期的管理；

（6）大数据分析。

3.2.2 大数据治理框架

图 3.2 描述了《大数据治理与服务》中提出的大数据治理框架，从大数据治理的原则、范围、实施与评估三个维度展示了大数据治理的主要内容。

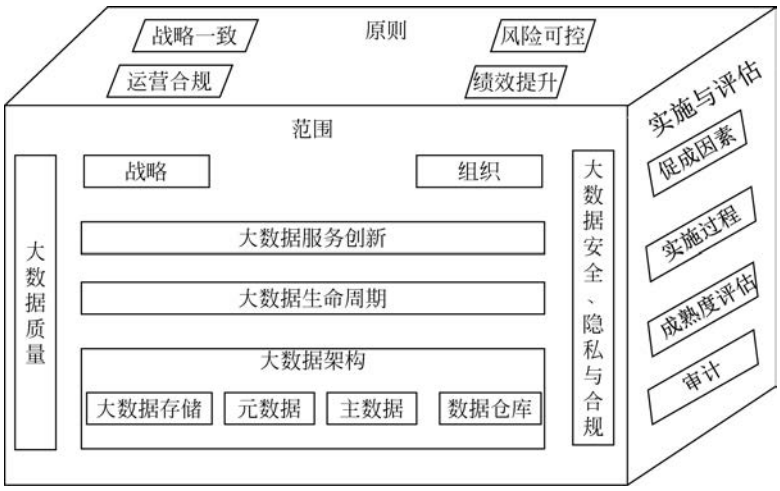

图 3.2 大数据治理框架

原则维度给出了大数据治理中必须遵循的指导性法则，即战略一致、风险可控、运营合规和绩效提升。这四项原则对大数据治理实践有指导的作用，只有将它们融入实践中，才能更好地执行大数据治理的战略和实现大数据治理的目标。

范围维度描述了大数据治理的关键域，即明确了大数据治理决策层应该在哪些关键领域内做出决策。范围维度包含了 7 个关键域：战略，组织，大数据质量，大数据安全、隐私与合规，大数据服务创新，大数据生命周期和大数据架构。

实施与评估维度主要包括促成因素、实施过程、成熟度评估与审计四方面，涉及大数据治理所需的实施环境、实施步骤和实施效果评价。

可根据原则维度中的四个指导原则，按照实施与评估维度中的方法论，对范围维度中

的 7 个关键域进行科学的决策，持续稳步地推进大数据治理工作。

下面将详细地介绍大数据治理框架中的大数据架构，大数据架构主要包括 5 部分：大数据采集层、大数据存储层、大数据管理层、大数据分析层和大数据应用层，如图 3.3 所示。

图 3.3 大数据架构

1. 大数据采集层

大数据采集层的功能是采集多源异构数据，数据主要包括关系型数据库中的结构化数据，XML、HTML 文件等半结构化数据和文本、视频、图像等非结构化数据。其中，对关系型数据库数据的采集可使用 Sqoop 工具，它的主要功能是在 Hadoop 和关系型数据库之间传递数据，将关系型数据库（如 MySQL、Oracle、Postgres 等）中的数据导入到 Hadoop 的 HDFS 中，也能将 HDFS 的数据导出到关系型数据库中。对 XML、HTML 等半结构化数据可采用日志采集框架 Flume 工具，它支持在日志系统中定制各类数据发送方，也具有对数据进行简单处理，将各种数据写到数据接收方的功能。对文本、视频、图像等非结构数据可采用文件数据处理工具 Kettle，它是一个 ETL 工具集，允许用户管理来自不同数据库的数据。

2. 大数据存储层

大数据存储层的功能是对采集层所采集的数据进行存储，可采用数据仓库、非关系型

数据库、分布式文件系统进行数据存储。

数据仓库系统设计师 Inmon 将数据仓库定义为“数据仓库是支持管理决策过程的、面向主题的、集成的、稳定的、不同时间的数据集合”。根据定义可得出数据仓库的四个特性：面向主题的、集成的、不可更新的、包含历史数据的。数据仓库是面向主题组织数据的，“主题”对应于客观分析领域的对象，用于明确集成哪些部门或系统的相关数据。“集成”体现在数据仓库中的数据是从分散的数据源中抽取出来的，由于每一个主题对应的原始数据可能存在重复、冲突和不一致的地方，因此数据进入数据仓库需要进行集成处理。“不可更新”是因为数据在导入数据仓库后，企业对数据进行时间趋势、区域状况的分析决策只需进行查询操作，而不需要进行增加、修改和删除操作。“包含历史数据”是指数据仓库记录的是企业的历史数据，通过分析历史数据来预测企业未来的发展趋势。传统的数据仓库采用 MySQL、Oracle 等关系型数据库，新型的数据仓库可采用基于 Hadoop 的 Hive 数据库等。

NoSQL 数据库摒弃了关系模型的约束并弱化了一致性的要求，以解决大规模数据集合中多种数据种类带来的挑战，尤其是大数据的应用难题。目前，NoSQL 数据库主要有四大类：键值存储数据库、列存储数据库、文档型数据库和图形数据库。键值存储数据库主要使用哈希表，表中有一个特定的键和一个指向特定数据的指针。键值存储数据库在 IT 系统中容易部署，使用简单。然而对部分值进行查询或更新时，键值模型效率比较低。文档型数据库的数据保存载体是 XML 或 JSON 文件，以支持灵活丰富的数据模型。一般文档型数据库可以通过键值或内容进行查询。图形数据库使用灵活的图形模型，将数据保存在图中的节点或者节点间的关系上，同时能够扩展到多个服务器上。

根据上述四种类型的 NoSQL 数据库的描述，得出 NoSQL 数据库所具有的几个特性：数据模型比较简单、数据库性能比较好、存储在数据库中的数据不是高度一致的和对于给定键比较容易映射复杂值等。目前比较常用的 NoSQL 数据库有 Redis(键值存储数据库)、HBase(列存储数据库)、MongoDB(文档型数据库)、Neo4j(图形数据库)等。

分布式文件系统指管理网络中跨多台计算机存储的文件系统，即文件系统管理的物理资源是分布式部署的若干台独立的计算机，计算机之间通过网络进行互联。分布式文件系统的设计是基于客户机/服务器模式，一个分布式文件系统可能包括多台供多用户访问的服务器，或因对等特性，系统允许某些计算机扮演客户机和服务器的双重角色。目前，常见的分布式文件系统有 GFS(Google 公司为了满足本公司需求而开发的基于 Linux 的专有分布式文件系统)、HDFS(HDFS 是 Apache Hadoop Core 项目的一部分，是一个高度容错性系统，适合部署在廉价的机器上)、Lustre(由 Sun 公司开发和维护的一个大规模的、安全可靠的，具备高可用性的集群文件系统)等，它们都是应用级的分布式文件存储服务，可根据具体的应用领域进行选择。

3. 大数据管理层

大数据管理层主要包含元数据管理、主数据管理等，本节主要介绍元数据管理和主数

据管理。

1）元数据管理

元数据(Metadata)通常被用来表达实体数据的描述信息，即可称为“数据的数据”，是抽象出用来表述数据特征的数据，比如数据的存储位置、数据的语义描述、数据的结构描述，其核心是对数据的统一管理，实现数据资源的科学整合，便于数据的长期保存。在大数据时代，元数据还包括对各种新型数据类型的描述，比如用户的点击次数、文件标签、传感器位置、传感器感应方向等。

元数据通常按照功能分为三种类型：业务元数据、技术元数据和操作元数据等。业务元数据描述了信息系统中业务领域术语、业务规则、运算法则及业务语言等，一般业务用户比较感兴趣；技术元数据描述了信息系统正常运行所需的信息，比如系统数据表结构信息、数据处理流程信息以及对存储过程的描述信息等。操作元数据是指描述信息系统的运行日志记录，比如用户访问量、记录数以及各个组件的分析和其他统计信息等。

在大数据时代，将元数据管理与大数据结合，可以对大数据中的敏感信息进行分类、标记，对大数据在信息供应链中的流动进行监测，及时了解流程中某处的工作是否出现故障或某些数据的丢失情况；也可以支持数据血统和影响的分析，回答诸如“数据来自何处”“数据要到哪里去”“数据流动中发生了什么事件”和“一个数据产品如何影响另一个数据产品”等基本问题；也可以创建针对非结构化数据的结构化索引，以支持非结构化数据的检索。

2）主数据管理

企业主数据是企业运营中担当关键角色的核心业务实体，一般指客户、供应商、产品、物料以及组织架构等数据，这些数据分散地存在于企业的各个业务系统中。只有确保企业主数据的完整和准确，才能保证企业业务流程的正确执行和应用系统产生正确的交易数据。

主数据管理是一组约束、方法和技术解决方案，主要功能是保证整个信息供应链中企业主数据的完整性、一致性和准确性，为报表提供一张主数据整合视图，或者为交易提供一个主数据的中央数据源，避免主数据的歧义，降低外部应用系统访问主数据的复杂性。

在大数据时代，将主数据管理与大数据进行整合，可以达到提升数据质量或达到大数据治理等目的，同时也为大数据分析提供了一个可靠的支撑载体。

4. 大数据分析层

要挖掘大数据的大价值必然要对大数据进行内容上的分析与计算。目前，越来越多的应用涉及大数据，而大数据的特性包括数量、速度、多样性等都呈现出不断增长的复杂性，因此分析方法十分重要，分析方法决定了能否从大数据中分析出需要的数据价值。大数据分析的理论核心是数据挖掘，基于不同的数据类型和格式的各种数据挖掘算法，可以呈现出数据本身的特征，使得大数据内部的价值得以被发现。大数据分析的应用核心是大数据预测。大数据预测完全依赖大数据来源，具有“全样非抽样，效率非精确，相关非因果”的特

征。大数据分析的结果主要应用到智能决策领域。

5. 大数据应用层

在大数据应用的过程中，无论是数据的使用者还是数据的开发者，都是通过数据访问接口来获取数据。数据访问接口为大数据的应用提供了通用机制，从而实现了平台、语言和通信协议无关的数据交换服务。在平台可视化和应用接口的支撑下，大数据应用层主要有三种典型的应用模式：大数据共享和交易模式、开放平台接口模式和大数据应用工具模式。通过数据资源共享、数据接口以及服务接口的聚集，可以实现数据交易及数据定制服务等共享服务、接口服务以及开发支撑服务。

3.2.3 工业大数据治理的概念

工业大数据是指在工业行业中，基于典型智能制造模式，在产品全生命周期各个环节所产生的各类数据和相关技术及应用的总称，以产品各类数据为重点，同时延伸了传统工业数据的范围，并涵盖了工业大数据相关技术和应用。工业大数据不仅具备广义大数据的4大特征，还具有关联性强、模态多样化和传输通量高等特点。

工业大数据分析工作应本着需求牵引、技术驱动的原则开展，如图3.4所示。在实际操作过程中，要以明确用户需求为前提、以数据现状为基础、以业务价值为标尺、以分析技术为手段，针对特定的业务问题，制定个性化的数据分析解决方案。工业大数据分析的直接目的是获得业务活动所需各种的知识，贯通大数据技术与大数据应用之间的桥梁，支撑企业生产、经营、研发、服务等各项活动的精细化，促进企业转型升级。工业大数据的分析要求用数理逻辑去严格地定义业务问题。由于工业生产过程中本身受到各种机理约束条件的限制，利用历史过程数据定义问题边界往往达不到工业的生产要求，需要采用数据驱动＋模型驱动的双轮驱动方式，实现数据和机理的深度融合从而较大程度地解决实际的工业问题。

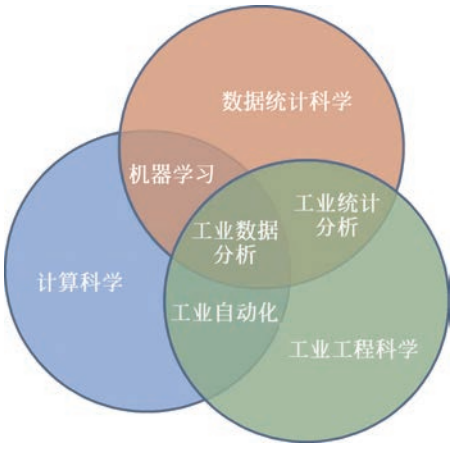

图3.4 工业大数据分析多领域交叉示意图

对工业大数据进行高效治理并基于工业大数据相关技术进行潜在信息挖掘，是实现工业大数据应用价值的有效方式。目前，工业领域对多源异构数据的处理与利用仍缺乏通用的标准，因此如何实现有效的工业大数据治理，包括提高数据的准确性和完整性，实现数据资源的有效共享，推进数据资源的整合与对接等处理流程，从而为后续的数据价值挖掘奠定基础是工业大数据研究领域的重要内容。

3.3 基于语义网的工业大数据治理

1. 数据语义是解决治理问题的总钥匙

工业大数据应用的本质目标就是从高维、复杂、关联的海量数据集中挖掘有价值的新信息，发现新模式与新知识。这些海量数据的关系密切、关联性强、语义稳定度高的特点使数据语义成为解决大数据治理问题的总钥匙。因此，从建立数据语义模型入手，是开展工业大数据治理的关键。建立良好的语义模型，犹如为工业大数据注入了优质基因，具备语义的工业大数据是“聪明”的大数据，具有打通多领域数据的本领，可以为面向复杂“大机理”，发现具备强泛化能力的新知识、获得多学科协同融合的“无缝智慧”提供关键支撑与坚实保障。

2. 基于语义的工业大数据治理方法

基于语义的工业大数据治理，除了包括制定战略战术、建立组织架构、明确职责分工等传统数据治理内容外，还要强调语义在工业大数据治理中的核心作用，其目标是实现工业大数据的语义化，实现数据的互联互通互融、风险可控、安全合规、绩效提升和价值创造，并为不断创新的大数据服务提供源源不断的充沛泵力。

基于语义的工业大数据治理思路如图 3.5 所示。通过对工业大数据进行业务术语规范和语义标注，利用本体技术及语义映射构建语义网模型。语义网(semantic web)是能够根据语义进行自动判断的智能网络，通过资源描述框架(resource description framework，RDF)和 Web 本体语言(Web ontology language)等对语义层面的本体关系(同义、反义、关联、隶属、属性约束等)进行定性与定量描述。计算机根据语义网模型，可以理解词语及其概念，并能够理解不同数据之间的语义关系，从而实现更深层次的语义关联、语义查询与逻辑推理。

语义网犹如搭建在工业大数据与分析之间的一座“逻辑桥梁”，在本体和原始数据之间建立映射关系，通过逻辑层面的本体关联，控制物理层面的数据关联。语义网向下实现数据之间的语义互联与互融，执掌工业大数据治理的技术落地；向上支撑工业大数据深度分析与知识发现，肩负工业大数据的创新应用实现。

3. 基于语义的工业大数据治理内容

基于语义的工业大数据治理工作主要包括：制定元数据管理策略、确定元数据集成体系结构、定义业务问题、建立组织职能、本体建模与构建语义网、数据监管及度量与评价六部分，如图 3.6 所示。

本体建模与构建语义网是核心内容。该步骤可以采用自上向下的方法，即首先确定业

图 3.5 基于语义的工业大数据治理思路

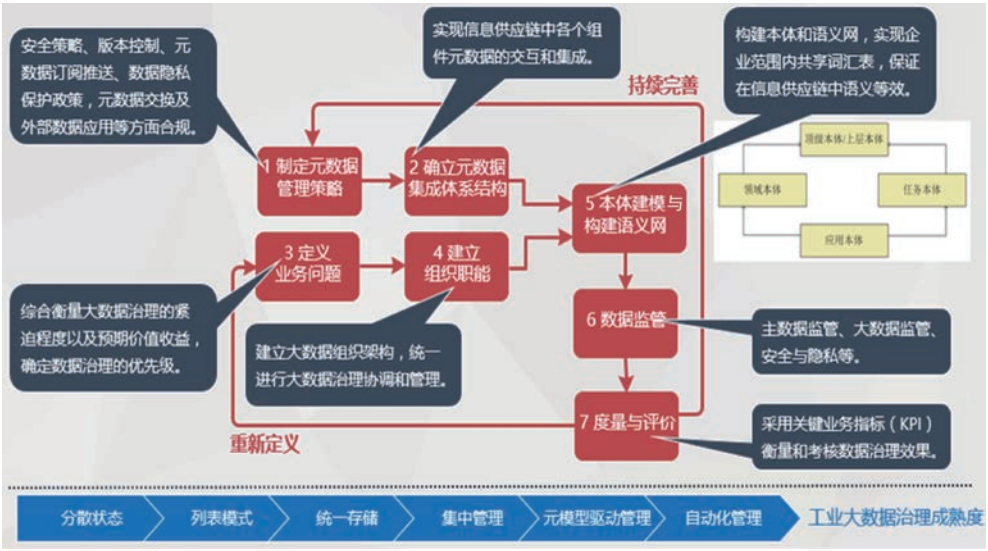

图 3.6 基于语义的工业大数据治理流程

务目标，从用例分析入手，明确业务范围；其次，根据业务目标，确定实现该目标分析的主数据；然后，根据主数据，逐步探寻业务范围内的关联信息，包括外部半结构化与非结构化数据；最后，建立本体模型，同时考虑依据大数据隐私策略，对相应的敏感数据进行标记和分类，并将数据与本体进行关联与映射，形成类似数据神经网络的语义模型，通过语义查询与推理，对本体模型和语义模型进行验证。

透过语义网络，可以监测大数据在整个数据链中的流动；可以通过数据血缘分析实现数据的正向追溯和逆向追溯，了解数据所经历的变化；还可以通过某个具体字段的变更来分析了解对数据链中其他字段造成的影响。

上述工作也可以与自下而上（数据字典提取、标签自动标注等）的方法相结合，更高效

地完成概念分类及整合、规范化业务术语、建立本体及确定本体之间的定性与定量关系，保证每个数据元素在信息供应链中语义等效。

3.3.1 本体论

1. 本体定义与分类

本体起源于哲学领域，是形而上学的重要分支之一，研究自然以及事物的组织，试图回答什么刻画了存在，以及存在是什么，即本体是对客观存在的系统或领域的解释或说明，关心的是客观现实的抽象本质。

20 世纪 80 年代初，信息科学开始对自然世界认知的形式化表示进行了重点研究。在 1990 年初，计算机领域使用了“本体”一词，含义为可被计算机表示、解释和利用的知识的形式化研究。

随着对本体的深入研究，研究人员给出了许多不同的本体定义。1991 年 Neches 等给出的本体定义是通过抽取相关领域的词汇和关系，定义基本术语和关系，利用术语和关系的演绎规则进行规范性的定义。Gruber 在 1993 年定义本体是概念化的规范说明。后来 Studer 等(1998)又对上述定义进行了补充，指出本体是领域知识规范的抽象和描述，是表达、共享、重用知识的方法，认为本体是共享概念模型的明确的形式化规范说明，它包含 4 层含义：概念化(Conceptualization)、明确化(Explicit)、形式化(Formal)和共享(Share)。

虽然不同的研究者对本体给出了不同的定义，但是关于本体的本质都是相近的。本体的本质概括起来就是在相关领域内，通过构建共享词汇库，明确领域中的概念及概念间的关联关系，为不同对象(如人、机器、软件系统)之间的交流提供语义基础。

虽然各类本体的本质是相近的，但是研究者根据本体的实际应用，提出了很多本体的分类方法。本节根据 Guarino 提出的两种维度，即本体的详细程度和领域依赖程度来对本体进行分类。按照详细程度，可将本体分为参考本体和共享本体，前者比后者的详细程度高。按照领域依赖程度可划分为四类本体，即顶级本体、领域本体、任务本体和应用本体，如表 3.1 所示。

表 3.1 本体按照领域依赖程度分类

本体名称	本 体 描 述
顶级本体	描述的是最通用的概念及概念间的关系，如空间、时间、事件、行为等，完全独立于特定的问题和领域，其他本体都是该类本体的特例
领域本体	描述的是特定领域(如医学、地理、企业运营等)中的概念及概念之间的关系
任务本体	描述的是特定任务或行为中的概念及概念之间的关系
应用本体	描述的是依赖于特定领域和任务的概念及概念之间的关系

另外，其他研究者提出了其他分类方法，比如按照本体的形式化程度和是否具有推理功能进行分类，如表 3.2 所示。

表 3.2 本体按照形式化程度和推理功能分类

本体分类方法	本体名称	本体特征
形式化程度	高度非形式化本体	使用自然语言松散表示
	结构非形式化本体	使用限制的结构化自然语言表示
	半形式化本体	使用半形式化(人工定义的)语言表示
	严格形式化本体	所有术语都具有形式化的语义,能在某种程度上证明完全性和合理性
本体是否具有推理功能	轻量级本体	不具备逻辑推理功能,例如叙词表和 WordNet
	中级本体	具有简单的逻辑推理功能,本体中一阶谓词逻辑的表达式可以被系统识别
	重量级本体	具有复杂的逻辑推理功能,本体中复杂的二阶谓词逻辑的表达式可以被系统识别

2. 本体建模原则与方法

Gruber 在 1995 年提出了最有影响的 5 个本体构建原则,如下所示:

1) **清晰性(Clarity)**:本体必须能够有效表达所定义术语的内在含义。术语定义必须是客观的,不能被局限于具体的场景或需求;是形式化的,必要时可采用逻辑公理来描述;可以使用自然语言加以说明;

2) **一致性(Coherence)**:本体必须能够支持与其定义相一致的推理,推理结果不能与定义相矛盾;

3) **可扩展性(Extendibility)**:本体必须能够支持在已有概念体系上扩展新的概念体系,以满足可预见的任务;

4) **编码偏好程度最小性(Minimal encoding bias)**:编码是为了满足描述或者执行的便利性,在本体的设计中编码偏好程度应该达到最小,从而在不同编码的描述系统中实现知识的共享;

5) **本体承诺最小(Minimal ontological commitment)**:本体对模型化的领域提供尽可能少的要求,即仅仅定义满足知识交流所必需的概念即可。

虽然 Gruber 提出了 5 个本体构建原则,但是由于没有成熟的理论给本体建模方法作指导,导致本体建模方法都是针对具体领域和项目提出的,因此目前存在很多本体建模方法。具有代表性的本体构建方法主要包括骨架法、TOVE 法、IDEF5 法、斯坦福七步法、五步循环法、METHONTOLOGY 法、KACTUS 法、SENSUS 法和循环获取法等。

骨架法、TOVE 法和 IDEF5 法是用于描述和获取企业本体的方法。骨架法是基于流程导向的构建方法,它只提供开发本体的指导方针,其构建流程如图 3.7 所示。

TOVE 法专用于构建多伦多虚拟企业本体,TOVE 本体包括了企业设计本体、工程本体、计划本体和服务本体,其构建流程如图 3.8 所示。

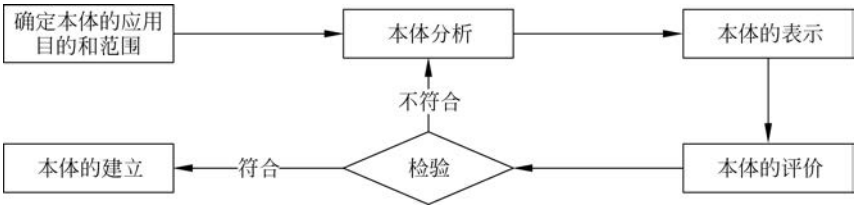

图 3.7 骨架法流程图

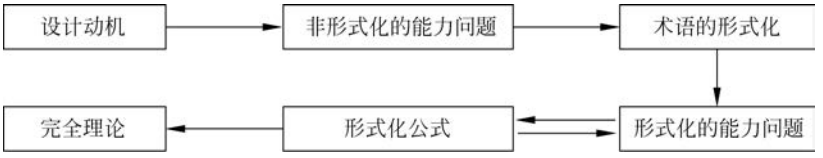

图 3.8 TOVE 法流程图

IDEF5 法采用图表语言和细化说明来构建企业领域本体，其本体开发步骤如下：

1）定义课题、组建队伍；

2）收集数据；

3）分析数据；

4）建立初始化的本体；

5）本体的精炼与确认。

KACTUS 法、METHONTOLOGY 法、SENSUS 法和斯坦福七步法主要用于构建领域知识本体，它们的不同之处是：KACTUS 法主要是对已有本体的提炼、扩展，主要用于解决知识复用的问题；METHONTOLOGY 法专用于构建化学知识本体；SENSUS 法遵循自上而下的层级结构，可操作性较强；斯坦福七步法是基于本体构建工具 Protégé 的本体建模方法，目前应用广泛，其构建流程如图 3.9 所示。

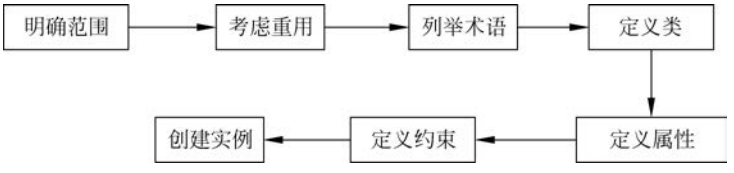

图 3.9 斯坦福七步法流程图

3. 本体建模元语

Perez 等按分类法来组织本体，归纳出本体包含的 5 个基本的建模元语（Modeling Primitive），如下所示：

1）**类（Classes）**：也叫作概念（Concepts），表示对现实世界中个体的抽象，表示对象的集合。

2）**关系（Relations）**：表示领域中概念之间的交互关系，一个关系包含定义域和值域两部分，即关系在被限定了所适用范围的同时，也将概念进行了关联。本体包含了 5 种基本的

关系，如表 3.3 所示。

表 3.3 本体中的基本关系

关 系 符 号	关 系 描 述
Is-a	表达概念之间的继承关系
Part-of	表达概念之间整体与部分的关系
Attribute-of	表达某个概念是另一个概念的属性
Association	表达两个概念是相关的关系
Instance-of	表达概念与实例之间的关系

3）**函数（Functions）**：表示一类特殊的关系，表示前 $n-1$ 个元素可以唯一决定第 n 个元素。

4）**公理（Axioms）**：用于描述概念或关系之间等价、包含、对称关系等的永真断言。

5）**实例（Instances）**：代表类所对应的对象，即概念的具体化。

在实际构建本体的过程中，不必严格地按照上述 5 个本体建模元语进行本体的构建，而是可以根据具体的应用需求以及领域的特征，对建模元语进行扩展，以满足实际应用需求。

4. 本体描述语言

1）资源描述框架（RDF）

RDF 是由 W3C 提出的一种用于描述 Web 资源的信息框架，其主要目的是以最低限度的约束，灵活地描述信息，从而得能够存储、理解和处理关于数据本身的元数据信息。RDF 中提及的资源不同于人们日常所理解的资源，它已经泛化成关于任何能识别事物的信息。

W3C 为 RDF 定义了一个抽象的语法，该语法描述了一个简单的基于图的数据模型。该数据模型包含节点和连接弧，节点之间通过带有箭头和标记的连接弧进行连接。RDF 的基础构件是由主体、谓词（属性）、客体（取值）组成的三元组，通过对主体的属性进行赋值，来描述资源的元数据。RDF 三元组是断言，它说明三元组中主体和客体所表示的事物之间存在谓词表达的二元关系。一个三元组可以用来表示主体的一个特性，即一系列三元组可描述主体的多个特性。

由上文可知，RDF 采用一种建模的方式来描述数据语义（严格来说，是描述语义关系），使得 RDF 可以不受具体语法表示的限制，即同一个模型既可以采用 Turtle 语法来表示，也可以采用 XML 语法来表示。另外，为了实现 RDF 在 Web 上的应用，一般都将 RDF 序列转化为 XML 进行表示，因为 XML 是被广泛支持的 Web 数据表示标准，这样做有利于使 RDF 获得更好的 Web 应用支持。总之，RDF 采用 XML 表示，可以很好地实现对数据的语义描述、建模和系统之间数据的真正交换。

但是，由于 RDF 为了保持充分的灵活性，遵循着“越少规则越好”的最小权利原则，其仅仅提供了一个描述关系的通用模型，而对资源的属性没有施加任何限制，导致在具体应用中容易出现一词多义和一义多词的现象。由于这种语义上的含糊，会导致机器对语义的

识别错误，而这些问题可通过本体描述语言来解决。

2）RDFS

RDF 允许用户使用自己的词汇来描述资源及资源间的联系。RDFS（RDF Schema，资源描述框架模式）是在 RDF 的基础上，以“http://www.w3.org/2000/01/rdf-schema#”作为命名空间的词汇表，用户必须按照这个词汇表的标准来构建特定领域的本体模型。图 3.10 展示了 RDFS 词汇之间的关系。

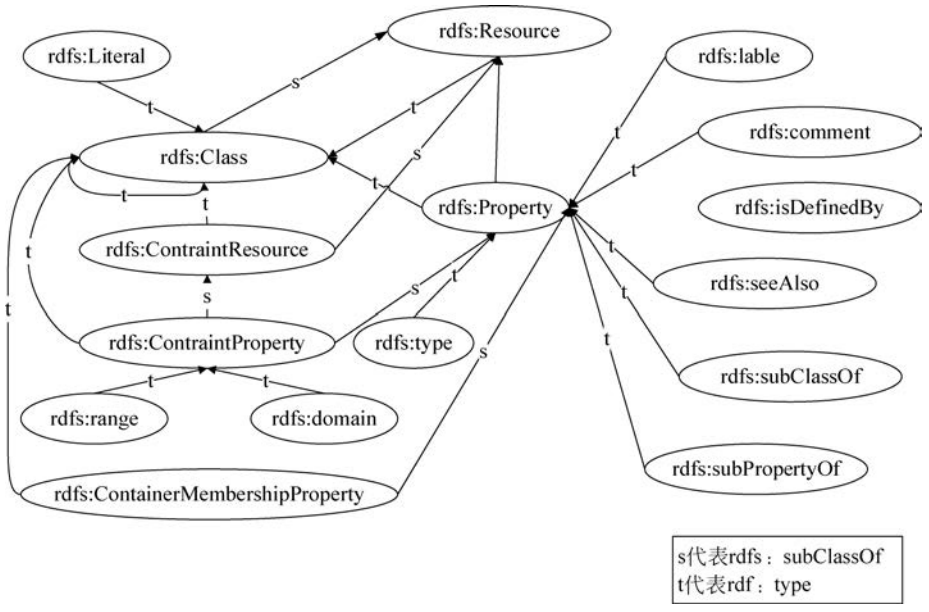

图 3.10　RDFS 词汇及关系

（1）核心类

rdfs:Resource，所有资源的类。

rdfs:Class，所有类的类。

rdfs:Literal，所有文字的类。

rdf:Property，所有属性的类。

rdf:Statement，所有具体化声明的类。

（2）定义联系的核心属性

rdf:type，将一个资源关联到它的类，该资源被声明为该类的一个实例。

rdfs:subClassof，将一个类关联到它的超类。需要注意的是，一个类可能有多个父类。

rdfs:subPropertyOf，将一个属性关联到它的超属性。

（3）限制属性的核心属性

rdfs:domain，指明一个属性的定义域，声明任何拥有某个给定属性的资源是定义域类的一个实例。

rdfs:range，指明一个属性的值域，声明一个属性的取值是值域类的一个实例。

(4) 对具体化有用的属性

rdf:subject,将一个具体化声明关联到它的主语。

rdf:predicate,将一个具体化声明关联到它的谓语。

rdf:object,将一个具体化声明关联到它的宾语。

(5) 容器类

rdf:Bag,包的类。

rdf:Seq,序列的类。

rdf:Alt,选择的类。

rdf:Container,所有容器类的超类。

(6) 功能属性

rdfs:seeAlso,将一个资源关联到另一个解释它的资源。

rdfs:isDefinedBy,是 rdfs:seeAlso 的一个子属性,将一个资源关联到它的定义之处。

rdfs:comment,注释,一般是长的文本,可以与一个资源关联。

rdfs:lable,将一个与人类友好的标签与一个资源关联,其中一个目的是在将 RDF 文档进行图形化表示时作为节点的名称。

从这些 RDFS 元语可以发现,RDFS 已经初具定义模式知识的能力,因此它被认为是一种简单的本体描述语言。

3) OWL

由于 RDFS 的表达能力较弱,W3C 又发布了 Web 本体语言(OWL),进一步为应用领域提供更加丰富的知识表示和推理能力。OWL 以描述逻辑为理论基础,将概念用结构化的形式表示,通过 RDF 中的 URI 将本体分布在不同的系统中。OWL 的设计核心是要在语言表达能力和提供高效智能服务的推理能力之间找到一个合适的平衡。

OWL 提供了 3 种子语言,分别是 OWL Full、OWL DL(description logic,描述逻辑)和 OWL Lite。OWL Full 的表达能力很强,但是它表示的本体不能进行自动推理,所以 OWL Full 一般适用于可判定性不强或不用计算完全性的场合。OWL DL 与 OWL Full 相比表达能力弱一些,它的基础是描述逻辑。由于描述逻辑是一阶逻辑的一个可判定的变种,因此采用 OWL DL 描述的本体可以进行自动推理,使计算机能区分出本体中概念的分类层次,以及判断本体中概念是否一致。OWL Lite 是 OWL 子语言中最简单的一种,适用于构建层次结构简单、复杂程度低、容易操作和只包含简单约束的本体,比如在将叙词表及分类系统转化为计算机可读的形式方面,OWL Lite 可以很好地发挥它的优势。

3.3.2 语义网

1. 语义网概述

目前互联网的发展可以分为两个阶段:第一阶段,给用户提供一个界面友好、交流方

便、信息共享的平台；第二阶段，计算机具有识别和处理互联网上信息的能力，实现计算机或者用户之间信息的交互。从互联网近几年的发展来看，互联网第一阶段目标已经实现，用户可以在网页上进行无障碍交流和合作，但是如何使计算机能够处理网页上的信息是目前研究的热点和难点。为了解决上述问题，W3C 在 HTML 和 XML 的基础上提出了语义网(semantic web)的概念，相较于互联网，语义网最大的优势在于“计算机可理解”。

语义网是互联网的延伸，语义网中的信息能够被计算机理解，使得计算机可以重用这些信息并且可以对信息进行自动处理，从而实现计算机与计算机之间、计算机与用户之间无障碍的交流和合作。互联网是面向文档的，网页主要使用 HTML 标记语言，着重于网页的表现形式，如大小、颜色、布局等，却忽略了网页中信息的内容和含义，而语义网是面向数据的，着重于网页信息的语义内容，核心是计算机对网页信息的理解与处理，并且语义网还具有一定的推理能力。

2. 语义网体系结构

语义网体系结构如图 3.11 所示，自下而上分别是编码定位层、XML 层、资源描述层、本体层、逻辑层、证明层、信任层等，下面进行简要介绍。

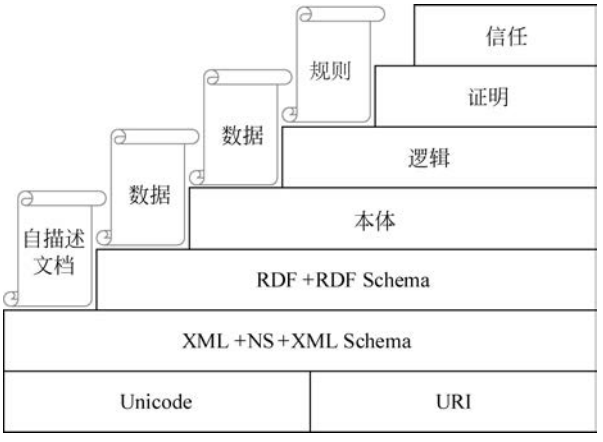

图 3.11 语义网体系结构

1）编码定位层

编码定位层是语义网的基础，由统一字符编码 Unicode 和统一资源定位符 URI 组成。Unicode 字符集通过两个字节可以表示高达 65536 个字符，Unicode 是一种能够涵盖所有语言的字符，因此采用 Unicode 可以将网络上的任何字符进行统一编码，有利于资源的共享和传递。URI 可以唯一地标记网络上的资源，因此 URI 是资源的标识。Unicode 和 URI 有效地解决了网络上资源的定位、各个地区之间字符编码的问题。

2）XML 结构层

XML 结构层又称语法层，包括 XML、Name Space、XML Schema 等相关技术，其中 XML 是一套用户可以自定义的标签，同时拥有 SGML 的强大功能和 HTML 语法简洁的优

点。在进行程序描述时，为了避免不同的程序使用相同的标签来表述不同的事物，W3C 提出了 Name Space 命名空间机制。XML 的标签结构由 DTD 或 XML Schema 进行规范，同时 XML Schema 还具有数据校验的功能。在语义网体系结构中，XML 结构可以根据用户的需求灵活改变、Name Space 可以保证数据的准确唯一性、XML Schema 对 XML 标签结构具有约束作用，这三者的良好结合实现了语义网中数据的交换、传递和共享。但是 XML 结构层仅定义了数据的语法，欠缺机器可以学习的形式化语言，因此语义网又增加了资源描述层。

3）资源描述层

资源描述层又称元数据层，包括 RDF、RDFS。RDF 是一个开放的元数据框架，语法上符合 XML 规范，用于描述 Web 上的信息资源。RDF 可以将 XML 的描述语言转为计算机可以识别的语言，且可将网络信息资源描述为“资源(resource)-属性(property)-属性值(value)”形式的三元组数据模型。RDFS 采用计算机可以识别的框架来定义资源以及资源之间关系的词汇(属性和类)。RDF 和 RDFS 一起称为 RDF(S)，它们共同实现对 Web 资源的描述。但是 RDF(S)不能解决一义多词、一词多义等语义模糊的问题，并且 RDF(S)无法平衡表述能力和推理能力。

4）本体层

本体层在 RDF(S)的基础上描述领域知识。本体用概念描述领域知识，其知识表达能力强于 RDF(S)，更有利于揭示复杂丰富的知识关系。同时，本体的一致性原则保证了知识的准确唯一性，解决了 RDF(S)存在的词义模糊不清的问题，并且本体具有数据结构严谨、语义表述明确的优点，因此本体更加广泛地用于知识的表达和推理研究。综上所述，本体层解决了资源描述层无法平衡表达能力和推理能力的问题。

5）规则层

语义网中的规则层包括逻辑层(logic)、证明层(proof)和信任层(trust)。语义网的推理依赖于数据和规则，本体层中对领域知识的规范化描述提供了推理所需数据，而逻辑层的主要任务就是提供与语义网结构相适应的规则。逻辑层定义的规则和公理由计算机自动读取执行。证明层给出计算机推理结果的解释。信任层用于数据和推理结果的评价验证，通过验证交换和数字签名建立信任关系，从而证明语义网的输出结果有效可靠。

3.3.3 关键技术

1. 本体建模工具

本体的构建离不开工具的支持。随着本体在人工智能、语义网、信息抽取、信息检索和数据整合等领域的广泛应用，有上百个团队开发出了许多不同的本体建模工具。不同的本体建模工具功能差异很大，比如软件操作的便利性、界面的友好性、功能的易用性、扩展性等方面不同，还有对本体语言的支持能力、表达能力、逻辑推理能力等都不同。目前使用广

泛的建模工具有 Protégé、ontoEdit、WebOnto、Ontolingua 等。

由于本书案例部分使用了 Protégé 软件来进行本体模型的构建，下面将主要介绍 Protégé 软件。Protégé 软件是由斯坦福大学医学院采用 Java 语言开发的。这款软件是构建本体的核心开发工具。Protégé 提供了构建本体的概念、关系、属性和实例的功能，采用图像化界面来进行本体的构建，即用户构建本体只需停留在概念层次，无须关心具体的本体描述语言。

下面具体介绍 Protégé 软件的界面。启动 Protégé 软件，单击“Create new OWL ontology”，就进入本体编辑窗口，如图 3.12 所示。

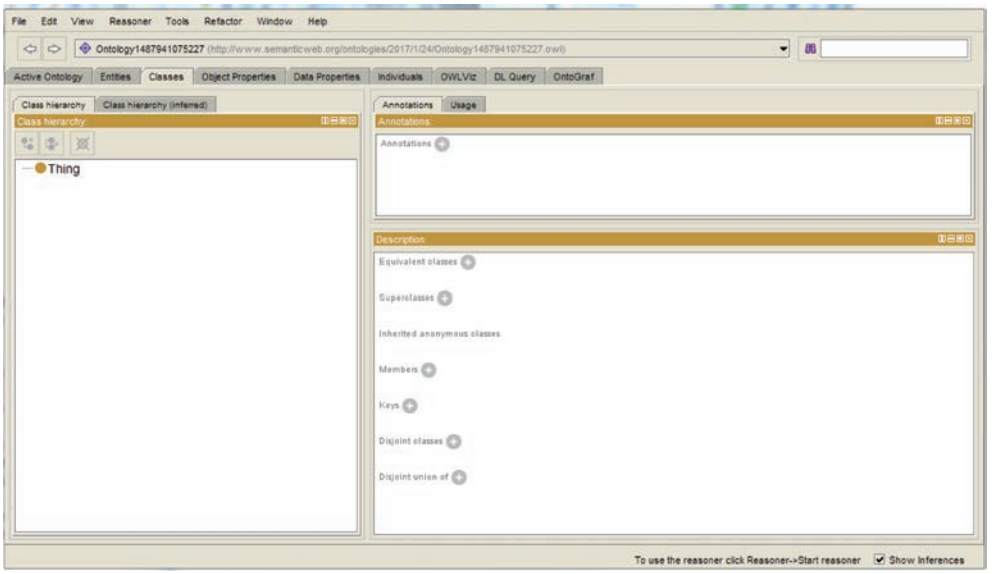

图 3.12 本体编辑窗口

Protégé 软件的界面包含以下几种开发画板：类(OWL classes)画板，可在类画板中添加和编辑类、子类，并以树形结构来展示类的层次结构；对象属性(object properties)画板，可在对象属性画板中添加和编辑对象属性，即类之间的关联关系；数据属性(data properties)画板，可在数据属性画板中添加数据属性，即类的属性；实例(individual)画板，用于添加某个类的实例。通过使用 Protégé 软件包含的功能，可将本体中类与类、类与对象属性、类与数据属性进行关联以及类与实例进行绑定。

通过对 Protégé 软件的功能描述，可以发现，使用 Protégé 软件构建本体无须掌握具体的本体描述语言。另外，相比于其他建模工具，Protégé 软件还具有功能扩展性比较强、支持众多的插件等特点，比如使用 OWLViz 可进行本体的可视化、使用 Pellet 可进行本体的推理以使用 Graphviz 可实现中文关系的显示等，这也是它成为国内外众多本体研究机构首选工具的原因之一。

2. 本体存储技术

目前，本体储存支持文件存储和数据库存储两种方式，在具体应用中可根据实际的需

求，选择合适的存储方式。文件存储方式是将本体以 RDF 或者 OWL 的文件形式存储在文件系统中。应用程序需要获取本体信息时，会将整个文件读入到计算机内存中，然后在内存中处理本体，处理完毕后，需要对本体中的所有内容进行保存。文件存储方式比较适合于小型本体，对小型本体的编辑、更新、备份比较方便，而不太适合较大的本体，因为较大的本体会占用计算机较多的内存，影响计算机处理本体的效率。特别地，若计算机所操作的本体包含很多推理规则，那么处理本体时会占用计算机更多的内存。

对于数据库存储方式，可细分为 2 种，一种是采用关系型数据库进行存储，另一种是采用 NoSQL 数据库的多索引表进行本体的存储。若采用关系型数据库存储本体，则使用关系型数据库的表结构来存储本体中的三元组数据，目前有 MySQL、SQL Server、Oracle 等关系型数据库支持存储本体。关系型数据库存储本体需要配置一个数据库连接，通过这个连接将本体包含的数据写入数据库中。其中，数据库连接包括的参数有数据库 URI、用户名、密码和数据库类型。将本体存储到数据库中后，用户就可以对本体进行查询、编辑、推理等操作。该方式由于研究时间较长，因此相关技术比较成熟，存储系统比较稳定。但是由于关系型数据库模型与 RDF 数据模型不太一致，因此在对本体进行存储、查询、推理操作时转换的开销比较大。

总之，基于数据库的本体存储方式相比于文件存储方式，能处理更大、更复杂的本体，而且不需要显式地保存数据模型，效率更高，但是需要对数据库参数进行设置。

3. Jena

Jena 是惠普公司开发的用于创建语义网应用系统的 Java 框架，它为本体的开发提供了开发环境，同时也为本体的检索和推理分别提供了查询引擎和推理引擎。用户可通过使用 Jena 开发包对 RDF、OWL 等文件进行解析和处理，从文件中读取本体信息并存储于特定的模型中，从而能够方便地对本体进行编辑和解析，并可以根据一定的推理规则，通过推理引擎对本体进行推理，从而实现语义推理和语义检索。

Jena 将 RDF 图作为其核心的接口，主要由以下 5 部分构成：

1）将 RDF 模型视为一组 RDF 三元组集合的 RDF API；

2）用于对 RDF 数据进行查询，可伴随关系型数据库存储一起使用以实现查询优化的查询语言 RDQL；

3）基于 RDF、OWL 等规则集的推理，也可自己建立规则的推理机子系统；

4）对 RDF 进行内存暂时存储和在 Oracle、MySQL、PostgreSQL 中进行数据持久化存储；

5）提供不同接口支持的本体系统来解析和处理 OWL、DAML＋OIL 和 RDFS。

Jena 主要由 API 和 SPI 组成，SPI 为 Jena 提供核心数据结构，用户若需要使用 Jena 的功能，只需要调用 Jena API 即可。Jena 以 Jar 包进行管理，在开发中经常需要用到的包有：

1）com.hp.hpl.jena.rdf.model 包。主要包含对 RDF 图的创建、编辑等功能，是 Jena API 的基础，包结构如图 3.13 所示。

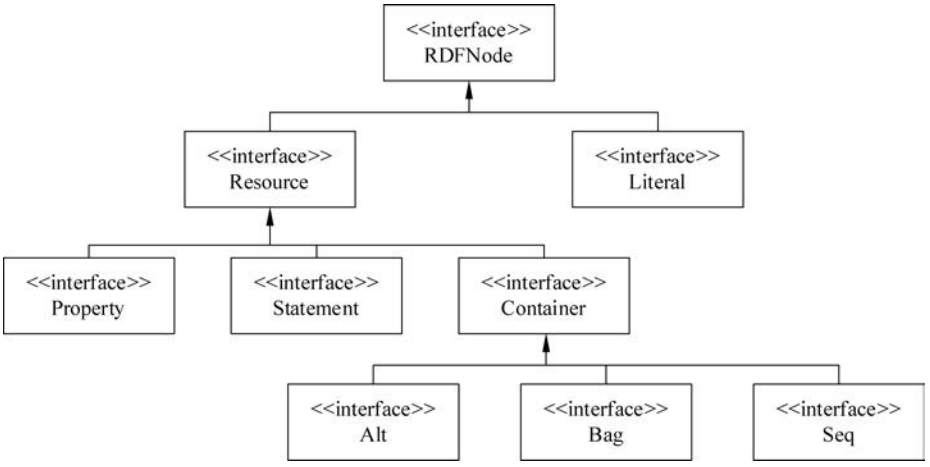

图 3.13　rdf.model 包主要接口函数

2）com.hp.hpl.jena.ontology 包。为操纵基于 RDF 的本体提供了抽象接口和实现，结构如图 3.14 所示。

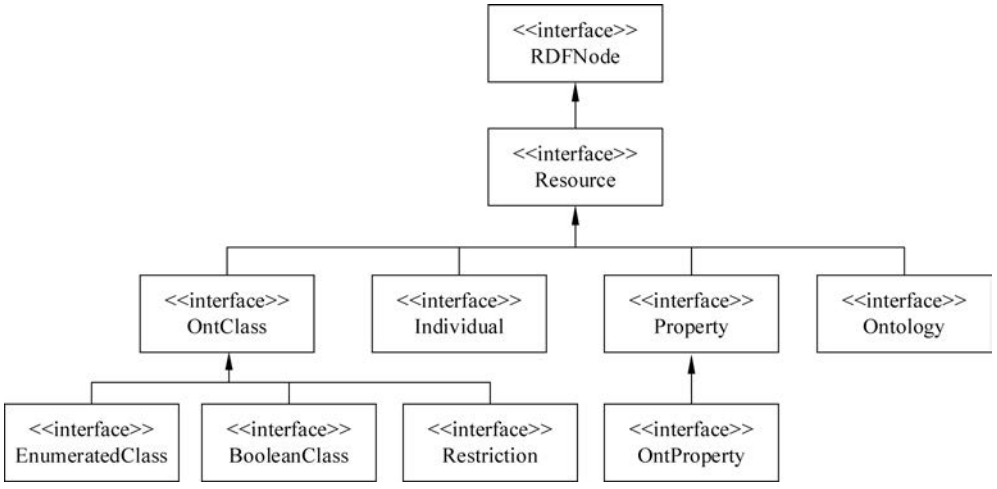

图 3.14　ontology 包的主要接口函数

4. D2R

D2R 目前是一款比较流行的用于发布 RDF 数据的工具，其主要功能是将关系型数据库的表数据发布为 RDF 数据，目前 Oracle、MySQL、Microsoft SQL Server、Microsoft Access 等主流的关系型数据库都支持其功能。图 3.15 描述了 D2R 的结构体系。

从图 3.15 可知，D2R 主要由 D2R 服务器、D2RQ 引擎以及 D2RQ 映射语言组成。

1）D2R 服务器

D2R 服务器是一个 HTTP 服务器，通过 D2R 服务器访问数据可以将查询结果以图形

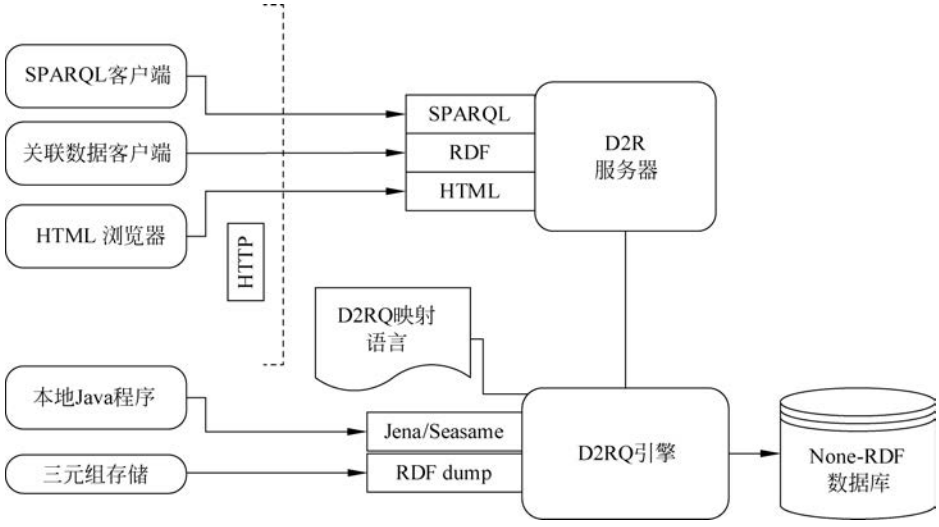

图 3.15 D2R 结构体系

化的形式展示出来，数据间的关系非常清晰明了。从 D2R 体系结构中可以发现，用户可以通过 SPARQL 客户端、关联数据客户端或者 HTML 浏览器来分别调用 D2R 服务器提供的 RDF 数据查询接口来访问数据。

2）D2RQ 引擎

D2RQ 引擎通过采用 D2RQ 映射语言编制的 D2RQ 映射文件，将关系型数据库的表数据转换为 RDF 格式数据。其中，D2RQ 并没有将表数据发布为真实的 RDF 数据，而是通过映射文件将表数据映射为虚拟的 RDF 数据。若需要将表数据转换成真实的 RDF 数据，用户可以通过本地 Java 程序或者三元组存储分别调用 D2RQ 引擎提供的 Jena/Seasame 或者 RDF Dump 来获取真实的 RDF 数据。

3）D2RQ 映射语言

D2RQ 映射语言用于构建将关系型数据库的表数据转换成 RDF 数据的映射文件。D2RQ 映射语言包含的两个重要的概念，分别是 d2rq:ClassMap 和 d2rq:PropertyBridge。

（1）d2rq:ClassMap 代表本体模型中的类，一般与关系型数据库中数据表的表名进行映射，它包含 2 个重要属性：d2rq:Class 表示该 ClassMap 所对应的类，其取值可以来自本体模型，也可以根据数据表中的数据特征定义新的类；d2rq:UriPattern 描述了一个 URI 模板，指导数据表转换为实例资源的真实 URI。一般用“makt/@@makt.物料号|urlify@@/@@makt.语言代码|urlify@@”来表示，“/”前面的部分为表名，后面“@@”之间的部分为表的列，若“@@”中包含多个列，则用“|”隔开。

（2）d2rq:PropertyBridge 代表本体模型中类的数据属性，一般与关系型数据库的数据表中的列进行映射。其中包括几个重要属性：d2rq:belongsToClassMap 表示该 propertyBridge 所属的 ClassMap，即用于说明该属性属于哪个类；d2rq:property 表示该 propertyBridge 所

对应 property，其取值可以取自现有的本体模型，也可以根据数据表中的数据特征定义新的数据属性；d2rq:column 表示该 propertyBridge 关联的关系型数据库中表的某列；d2rq:refersToClassMap 表示该 propertyBridge 引用了其他的 ClassMap，从而使 propertyBridge 的取值来自于它所引用的 ClassMap，而不是它所属于的 ClassMap 的值。当 propertyBridge 使用多个 d2rq:refersToClassMap 时，需使用 d2rq:join 来指明各个 ClassMap 之间的关联条件，d2rq:join 类似 SQL 语句中的 where 条件。

3.4 基于知识图谱的工业大数据治理

3.4.1 工业大数据与知识图谱

当前，新一轮科技革命席卷全球，大数据、云计算、移动互联网等技术成为构筑开放合作的制造业新体系的基石，扩展了制造业创新与发展的空间。制造业正迈向转型升级的新阶段——由数据驱动转向知识驱动，制造业隐性知识面临显性化需求。

传统的大数据具有数据容量大，数据生成速度快，数据格式类型多，数据置信度较低以及数据价值密度低的特点。与传统的大数据不同，工业大数据具有装备类型多、工况种类复杂、生产要素多等特征，并且需要较高的领域知识支撑。

知识图谱是一种研究数据之间关联关系的新兴技术，能有效地展现错综复杂的数据之间的各种关联关系，清晰地表达数据的知识结构，让数据的分析结果具有一定的可解释性。工业大数据知识图谱具有数据构成复杂、知识体系特殊等特征。区别于传统真实世界知识图谱的自然文本输入，工业大数据知识图谱的输入数据一半来源于已经结构化了的传感器数值数据，另一部分是半结构化数据和来自于一些具有高度规则的文档、图片、音视频的非结构化数据。同时工业大数据知识图谱包括一些高度领域化的实体及实体关系。因此，需要研究构建适应于特定场景的工业知识图谱的方法。

通过构建基于工业大数据的知识图谱实现工业大数据综合治理，可以提高数据管理的统筹能力。基于工业大数据的知识图谱可以将多源异构数据层层分解并关联起来，将离散的、分段的数据实现知识层面的集成，反映工业生产场景的整体面貌。

3.4.2 工业大数据环境下的知识图谱构建

工业大数据环境下的知识图谱通常包括一些通用的物理机理领域知识和一些面向特定工业生产过程以及特定生产装备的专业领域知识。通过构建面向特定工业领域的知识图谱，可最终实现面向特定工业问题的辅助决策分析。知识图谱具有很多构建理论，一般从知识图谱的基本结构来看，构建需要以下 5 个步骤：

（1）**明确数据来源**。领域不同，构建知识图谱的数据也相应不同，且数据来源往往存在于不同的架构体系中，表现形式难以统一。

（2）**知识提取**。结构化数据往往已经过清洗加工，可直接进行知识提取。而对于非结构化数据往往要通过实体抽取技术将实体及其关系抽取出来，从而得到具有实体关系表征的元数据。

（3）**数据清洗及合并**。知识提取后得到的数据依然缺乏层次，并可能存在错误重复的问题，没有进行有效组织，需对其进行清洗、合并等进一步处理，以得到高质量数据。

（4）**数据处理及知识模型构建**。用人类逻辑对这些海量高质量数据进行抽象和组织，并构建知识模型，使数据组织符合人类认知。此阶段需要人工高度参与。

（5）**构建知识图谱**。根据上一步的结果构建知识图谱，并且随着知识的更新，不断对其进行升级迭代。

构建工业大数据环境下的知识图谱的核心思想在于构建一个知识体系，以对海量多源异构数据的管理实现有力的支撑。如图 3.16 所示为工业大数据环境下一般知识图谱的构建流程。其中虚线框内部分为知识图谱的构建过程，同时也是知识建立和更新的主要流程。首先是原始数据处理，数据源可能是结构化的、非结构化的以及半结构化的，然后通过一系列自动化或半自动化的技术手段，从原始数据中提取出知识要素，即若干实体关系，并将其存入知识图谱的模式层和数据层。构建知识图谱是一个迭代更新的过程，根据知识获取的逻辑，每一轮迭代包含知识存储、信息抽取、知识融合、知识计算四个阶段。

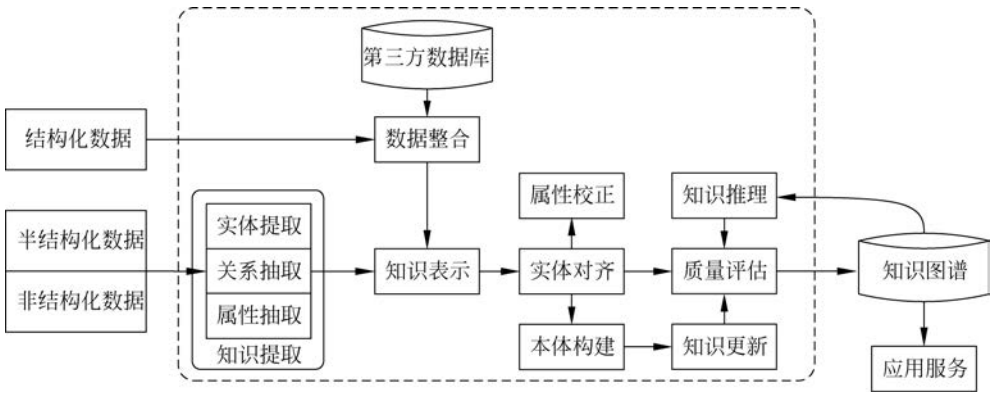

图 3.16 工业大数据环境下一般知识图谱的构建流程

（1）**知识存储**：针对构建知识图谱设计底层的存储方式，完成各类知识的存储，包括基本属性知识、关联知识、事件知识、时序知识、资源类知识等。存储方式的优劣将直接影响查询效率和应用效果；

（2）**信息抽取**：从各种类型的数据源中提取出实体、属性以及实体间的相互关系，在此基础上形成本体化的知识表达；

（3）**知识融合**：在获得新知识之后，需要对其进行整合，以消除矛盾和歧义，比如某些实体可能有多种表达，某个特定称谓也许对应多个不同的实体等；

（4）**知识计算**：对于经过融合的新知识，需要经过质量评估之后（部分需要人工参与甄别），才能将合格的部分加入知识库中，以确保知识库的质量。

在知识图谱构建完成后，将其与工业领域数据和业务场景相结合，将助力企业在该领域取得实际的商业价值。