



中华人民共和国国家标准

GB/T XXXXX—202X/ISO 5054-1:2023

企业数字化规范化模型 第1部分：架构

Specification for an enterprise canonical model—Part 1:Architecture

(ISO 5054-1:2023, IDT)

(征求意见稿)

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

XXXX - XX - XX 发布

XXXX - XX - XX 实施

国家市场监督管理总局
国家标准化管理委员会 发布

目 次

前言	II
引言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 网络安全	2
5 报文架构	3
5.1 概述	3
5.2 抽象报文架构	3
5.3 物理业务对象文档（BOD）报文架构	5
附录 A（资料性） 物理层 RESTful web 报文	11
A.1 概述	11
A.2 报文头	11
A.3 报文动作	11
A.4 报文体	11
参考文献	12

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是 GB/T XXXXX《企业数字化规范化模型》的第1部分。GB/T XXXXX 已经发布了以下部分：
——第1部分：架构。

本文件等同采用 ISO 5054-1:2023《企业规范化模型 第1部分：架构》。

本文件做了下列最小限度的编辑性改动：

——为符合我国数字化战略和标准化工作导向，便于企业快速理解本文件定位，将名称修改为《企业数字化规范化模型 第1部分：架构》；

——删除了 ISO 5054-1:2023 中提及 OAGi 自身时的表述；

——参考文献中文件已经采标的，改为对应国家标准。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国电子商务标准化技术委员会（SAC/TC 83）提出并归口。

本文件起草单位：厦门市标准化研究院、中国标准化研究院、厦门理工学院、厦门大学。

本文件主要起草人：张金梅、章建方、卢俊文、陈瑀、肖亮、黄建荣、蔡弘、李俊毅。

引 言

当前，企业在推进数字化转型过程中普遍面临两大难题：一是从不同供应商采购的业务应用程序无法有效协同；二是难以将所有系统（可能由不同供应商开发、运行在不同平台、使用不同语言）集成到一个连贯的信息技术基础设施中，以支撑业务运行。

针对上述问题，GB/T XXXXX 将集成标准定义为规范化模型，旨在为企业应用程序，包括企业资源计划（ERP）、采购、订单管理、财务、客户管理、质量数据等，提供单一事实来源（Single Source of Truth, SSOT）和统一语义语言，以语法中立的方式实现当前及未来企业所有应用程序的集成，解决各业务应用间昂贵的点对点集成困扰，使企业与软件供应商共同受益。

此外，GB/T XXXXX 不仅解决系统间的互联互通问题，更着眼于数据价值的深度释放。通过对业务数据的规范化处理与聚合，形成企业数字化转型的“通用语言”和“核心字典”，为商业智能、报表生成等数据分析应用提供标准化的数据底座，支撑企业实现从“数据”到“决策”的价值跃迁。

GB/T XXXXX 拟由六个部分构成。

- 第 1 部分：架构。目的在于定义规范化模型的三层架构体系，包括抽象模型、概念模型、物理模型，三者共同构成报文架构的完整规范框架，保证语法层面的互操作性。
- 第 2 部分：数据模型。目的在于采用 Score 工具建立与 GB/T 19256.5 兼容的数据模型。
- 第 3 部分：业务语义。目的在于定义业务语义（包括业务名词、报文、业务上下文），规定如何通过集成场景将其与业务流程关联，以及如何将这些语义定义映射到多种技术实现（如交付方法和数据格式）上，保证语义层面的互操作性。
- 第 4 部分：拓展与应用。目的在于定义规范化模型的可扩展性与可复用性，支持自定义、打包共享及基于核心构件的消息构建，通过 Extension 机制满足企业的特定需求，实现灵活性与标准化的统一。
- 第 5 部分：XML 序列化映射。目的在于规定业务语义配置文件向 XML 格式的序列化转换方法及映射规则。
- 第 6 部分：JSON 及 Open API 序列化映射。目的在于规定语义配置文件向 JSON Schema 及 Open API 规范的序列化方法，以适应现代 RESTful 架构的应用集成需求，逐步迁移到云原生 API 架构，保证技术层面的互操作性。

本文件是GB/T XXXXX 的第 1 部分，是企业数字化规范化模型构建的核心基础。本文件规定的报文架构，旨在为企业异构系统间建立统一的信息交换规范，从而为达成系统间的互操作性提供必要的语义基础、结构保障和技术映射手段。GB/T XXXXX 的其他部分是对本部分内容的延伸和细化，涵盖数据模型、业务语义、模型扩展性、XML及JSON/Open API序列化映射等。

企业数字化规范化模型 第1部分：架构

1 范围

本文件规定了企业规范化模型的架构，定义了抽象模型、概念模型和物理模型。

本文件适用于企业级系统的互操作性建设，也可适用于企业内部部门级或项目级系统的互操作性建设。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1

应用域 application area

向发送方、接收方及消息基础设施提供所需信息，以实现BOD实例（3.5）有效识别、标注、传输、路由及关联的结构。

示例：报文实例 ID、发送方 ID、关联 ID。

3.2

业务对象文档 business object document, BOD

符合BOD架构（3.3）的报文。

3.3

BOD 架构 BOD architecture

定义报文通用数据结构与行为的规则。

3.4

BOD 标识符 BODID

报文实例的通用唯一标识符（UUID）。

注：UUID的有关规则见参考文献[8]。

3.5

BOD 实例 BOD instance

BOD（3.2）的报文实例。

3.6

BOD 定义 BOD definition

以特定格式（JSON Schema或XML Schema）表示的BOD（3.2）。

注：JSON Schema见参考文件[9]。

3.7

构件 component

由子结构和字段（3.10）组成的数据结构。

注：见GB/T 19256.5—2022中3.23“核心构件”的定义。

3.8

数据区 data area

承载BOD（3.2）所传递业务语义有效载荷的构件（3.7）。

3.9

数据类型 data type

一些可区分的值的集合，这种区别由这些值的性质及对这些值的运算所表征。

注：数据类型可以是日期、时间、小数、数值等。

[来源：GB/T 18221—2000, 4.11, 有修改]

3.10

字段 field

数据元的最低层级。

注：这些元素包含具体数据值。

3.11

命名约定 naming convention

生成名称的规则集合。

注：GB/T 18391.5和GB/T 19256.5中规定了命名规则。

3.12

业务对象名称 noun

承载业务数据的构件（3.7）。

示例：采购订单、销售订单、报价单、物流路线、发货单。

注：业务对象名称包含在BOD（3.2）的数据区中。

3.13

接收方 receiver

BOD实例（3.5）的接收系统。

3.14

场景 scenario

业务流程的图形（UML序列图）和文本描述。

3.15

发送方 sender

创建BOD实例（3.5）的发送系统。

3.16

签名 signature

BOD实例（3.5）的数字签名。

注：见参考文献[13]。

3.17

动作 verb

规定业务对象名称（3.12）相关行为的构件（3.7）。

4 网络安全

网络安全对于信息系统的设计、实施与运行至关重要。应通过建立风险管理框架，将风险控制在可接受水平内。在业务流程开发过程中，宜融合信息技术（IT），实现风险评估的自动化和精准化。

5 报文架构

5.1 概述

为实现相互独立的系统、企业及价值链之间的互操作性，宜建立一个规范化报文架构，为可互操作的业务流程与沟通提供统一语义和方法。

基于该报文架构构建的报文，用于定义系统的交互逻辑，进而确立通用流程，实现系统间的互操作性。这些交互或场景提供了用于执行特定业务任务的分步指南。通过组合基本场景与附加消息步骤，还可以构建复杂场景来满足各类业务功能需求。

5.2 抽象报文架构

5.2.1 概述

抽象报文架构（或元模型）见图1。

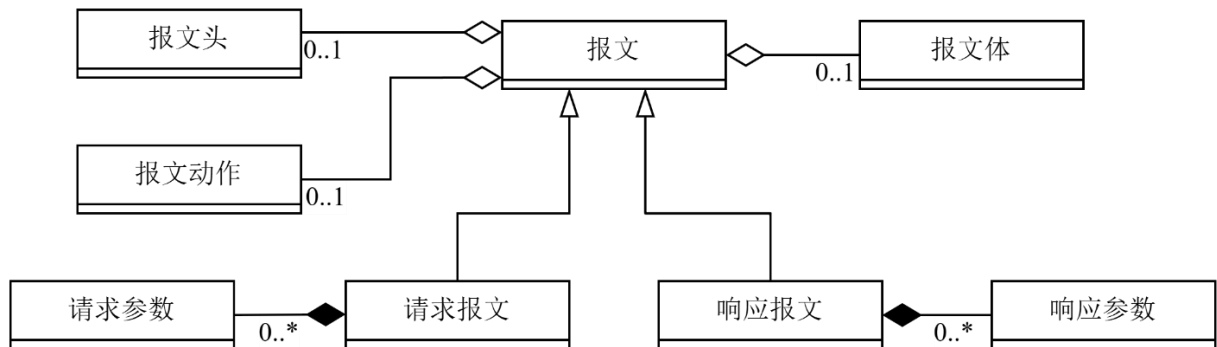


图1 抽象报文架构

报文（Message）指从发送方传递到接收方的信息，通常包含报文头（MessageHeader）、报文动作（MessageVerb）和报文体（MessageBody）三个部分：

- 报文头（MessageHeader）：包含报文实例的相关信息，如创建；时间、来源与去向、消息标识符及关联标识符；
- 报文动作（MessageVerb）：描述报文接收方需要执行的操作；
- 报文体（MessageBody）：包含具体的业务数据，也可包含与报文体实例相关的元数据，如记录数量。

报文类型包括请求报文（RequestMessage）和响应报文（ResponseMessage）。除上述报文属性外，请求报文（RequestMessage）中还可包含请求参数（RequestParameters），如选择和筛选条件。

5.2.2和5.2.3说明了如何实现这一通用报文架构。

5.2.2 概念性 BOD 报文架构

图2给出了抽象报文架构的一种实现形式，即BOD报文架构。

业务对象文档（BOD）实现了该报文。BOD应用域（ApplicationArea）和数据区（DataArea）分别实现报文头（MessageHeader）和报文体（MessageBody）。只有数据区（DataArea）中的业务对象名称或构件实现报文体（MessageBody）。数据区（DataArea）中的动作实现报文动作（MessageVerb）。BOD的请求动作实现报文请求（MessageRequest）的请求参数（MessageParameters）。BOD的响应动作实现报文响应（MessageResponse）的响应参数（ResponseParameters）。

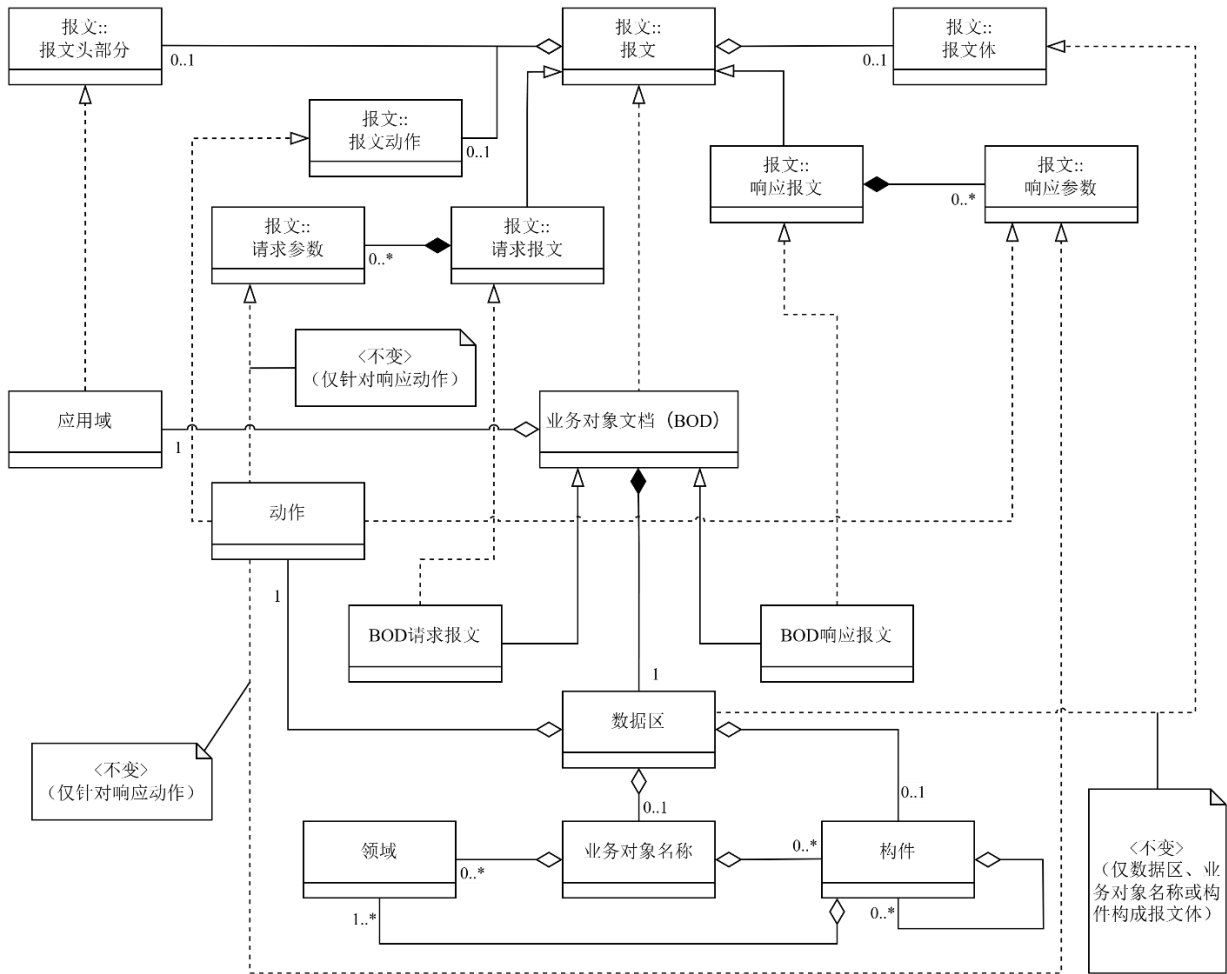


图2 BOD 报文架构

5.2.3 概念性 RESTful web 报文架构

图3给出了抽象报文架构的另一种实现形式，即RESTful web报文架构。RESTful web API是由互联网上公开服务提供的，符合REST（表示实体状态传输）架构风格的应用程序接口（API）。它采用网络的主要传输协议——超文本传输协议（HTTP）作为应用程序通信协议。通常，一个API包含一组命名操作，每个操作都包含一个请求报文及一个可选的响应报文。RESTful Web API的操作借助HTTP报文来定义其请求与响应报文（附录A）。这种HTTP报文架构是RESTful Web报文架构的基础。

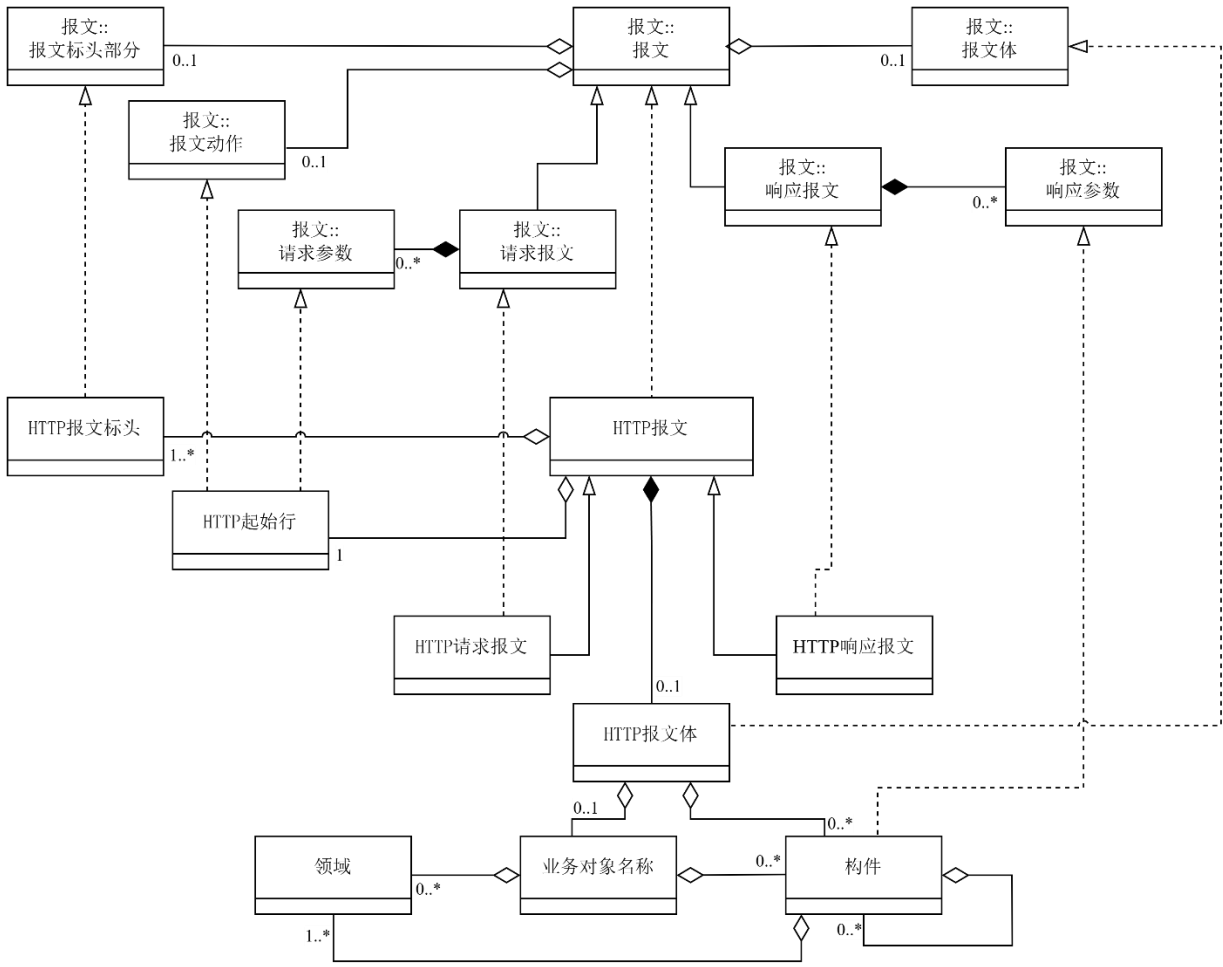


图3 RESTful Web 报文架构

HTTP报文 (HTTP-Message) 实现了该报文。HTTP报文头 (HTTP-MessageHeader) 和HTTP报文体 (HTTP-MessageBody) 分别实现报文头 (MessageHeader) 和报文体 (MessageBody)。HTTP报文体 (HTTP-MessageBody) 中包含了业务对象名称和/或构件。HTTP起始行 (HTTP-StartLine) 代表了HTTP请求报文 (HTTP-RequestMessage) 的请求行 (RequestLine) 和HTTP响应报文 (HTTP-ResponseMessage) 的状态行 (StatusLine)。请求行 (RequestLine) 包括HTTP版本号、HTTP方法和URL。状态行 (StatusLine) 包含HTTP版本号、响应状态码、响应状态描述。HTTP请求报文 (HTTP-RequestMessage) 的HTTP起始行 (HTTP-StartLine)，即请求行 (RequestLine)，实现报文请求 (MessageRequest) 的请求参数 (RequestParameters)。响应参数 (ResponseParameters) (如分页结果) 由构件实现。

5.3 物理业务对象文档 (BOD) 报文架构

5.3.1 概述

BOD架构为所有基于BOD的报文定义主模式。BOD本身是特定业务文档的主控者，定义了特定业务报文所需的可能数据范围；而场景中特定交互实例的实际创建，则根据具体的交互需求或合作方的要求来决定。

在某一场景或流程中承担单个步骤的BOD，应包含可交互的各类业务数据；从定义上看，BOD定义范围远大于实际交互的实例报文。

最终的实例交互发生在软件应用之间，这些应用能存在于部门或企业内、跨部门或企业范围内，也能覆盖供应链与价值链；实例可嵌入到多种传输协议中，以完成完整的交互步骤。

BOD还应能够向预期的接收系统传递应用与操作细节，如特殊条件、错误、应用需求或状态。

BOD可用W3C XML Schema (XSD) 或JSON Schema来表示，这两种格式都代表企业数字化规范化模型。本数据模型基于ISO 15000-5构建，ISO 15000-5将核心构件解决方案描述并定义为开发一组通用语义构建块的方法，这些构建块代表通用类型的业务数据，并支持新业务词汇的创建和现有业务词汇的重构。

BOD报文架构不依赖于任何特定的通信机制。它能与多种传输协议结合使用：既包括HTTP、FTP、SMTP等简单传输协议，也包括RESTful Web服务、基于SOAP的网络服务、AS1/AS2/AS3/AS4 ebMS、RNIF等更成熟的传输协议。

5.3.2 业务对象文档 (BOD)

5.3.2.1 概述

BOD报文架构可通过图4的形式呈现。

作为每个BOD的组成部分，BOD架构包括以下属性：

- 发布版本号 (ReleaseID)：标识 BOD 所属的版本。对于 10.0 及以上版本的 BOD，该属性的值为“10.0”。发布版本号是 BOD 的必要属性；
- 文档版本号 (VersionID)：标识 BOD 自身的版本。每个 BOD 都有独立的修订号，以具体明确其版本级别，而非仅关联发布版本号。具体的 BOD 文档版本号记录在每一份 BOD 文档中。最外层元素的名称不包含文档版本号，文档版本号以 BOD 属性的形式体现。文档版本号是可选属性；
- 系统环境代码 (systemEnvironmentCode)：标识该 BOD 源于“测试级”还是“生产级”集成系统。新系统上线时，应在生产环境中进行测试，以确保实现完整集成。通过该属性，集成方可将此类报文标记为测试报文。系统环境代码是可选属性；
- 语言代码 (languageCode)：指示 BOD 报文中数据的默认语言。对于可能需要携带多语言信息的字段（如备注、说明），有可能覆盖此 BOD 级别的默认设置。

BOD还承担两项功能：业务数据交换与应用交互；BOD由两个独立的子部分组成。应用级通信（如特殊条件、错误等）部分属于应用域。

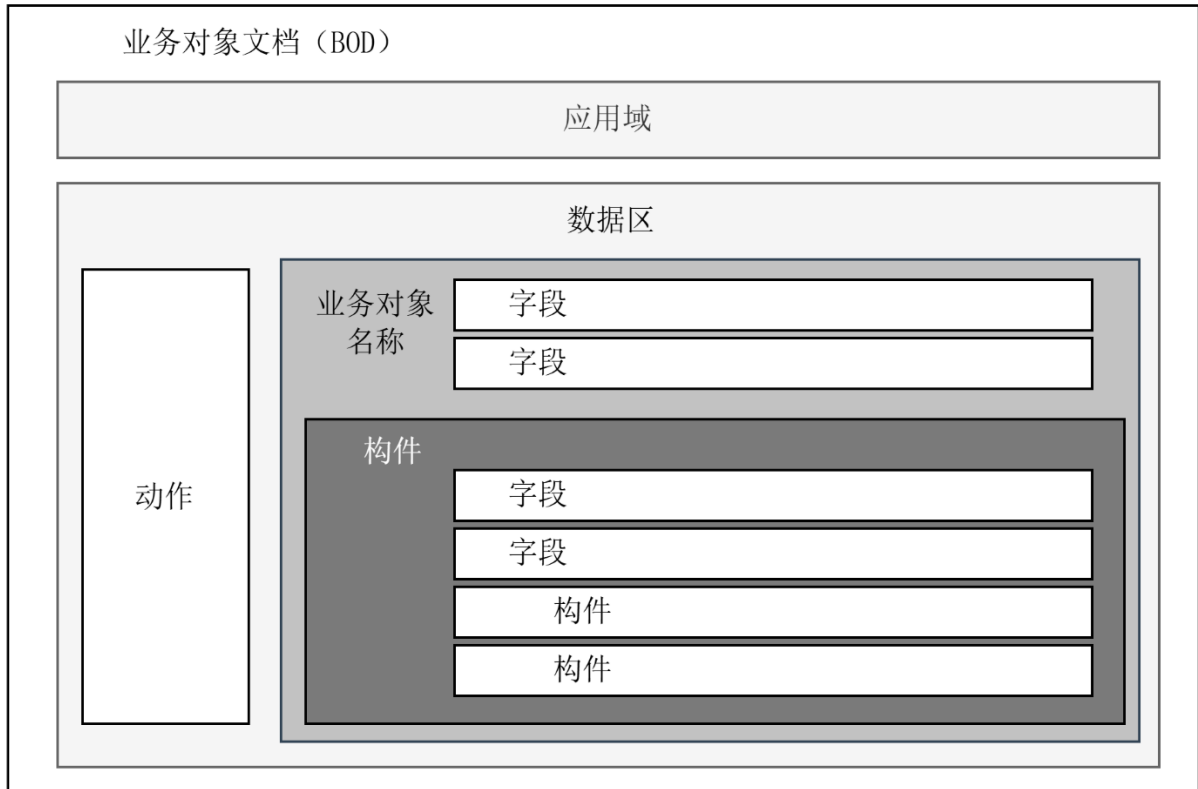


图4 BOD 架构

5.3.2.2 应用域

应用域 (ApplicationArea) 包括以下内容 (见图5)。

——发送方 (Sender)：标识创建此 BOD 实例的发送系统，指示了发送系统和/或数据库服务器的逻辑位置、发送系统本身及用于创建 BOD 的任务。它还允许发送系统请求接收系统提供确认回执。同时它支持接收系统建立审计追踪，结合发送方数据，更全面地理解该 BOD 实例中所传递的信息和流程。

- 逻辑标识符 (LogicalID)：标识发起该BOD的发送系统。它能用于建立由逻辑到物理的映射关系。其使用是可选的。建议各个系统或系统组合，都维护一份映射表，记录集成配置中应用系统的ID、逻辑名称或逻辑地址，以便将逻辑名称映射到网络所需资源的物理网络地址上。
- 构件ID (ComponentID)：提供比逻辑标识符更精细的控制级别，代表发布BOD的业务构件。其使用是可选的。宜采用发送系统的构件名作为命名约定。
- 任务ID (TaskID)：描述发起BOD创建需求的业务事件。其使用是可选的。尽管任务可因具体实施情况而异，但实现引用功能通常十分重要。任务示例包括回执 (RECEIPT)、调整 (ADJUSTMENT) 等。
- 引用ID (ReferenceID)：使发送系统能够识别触发该BOD创建的具体事件或任务的实例标识符。借助引用ID，可从BOD报文溯源到它的发送系统。在需要对所有交易保留审计轨迹的场景中，可要求使用。其使用是可选的。

- 确认代码 (ConfirmationCodes)：由发送方业务系统控制的一组选项，是对接收系统返回确认BOD (ConfirmBOD) 的请求。确认BOD (ConfirmBOD) 是一种用于确认BOD实例处理状态的BOD报文。它与业务流程无关，能够与任何其他定义的BOD一起使用。确认BOD (ConfirmBOD) 可标明原始BOD已处理成功，或在原始BOD处理不成功时，返回错误条件。

确认请求包括以下有效值：

- Never:不请求 (ConfirmBOD)；
- OnError:仅在发生错误时，返回 (ConfirmBOD)；
- Always:无论处理结果如何，始终返回 (ConfirmBOD)。

- 授权ID (AuthorizationID)：标识触发BOD实例报文发送的发送方 (用户、设备、发送系统或业务实体) 的授权信息。报文接收方会依据此授权信息，判断发送方的访问权限等级。在返回BOD报文时，接收方应用程序会将授权ID (AuthorizationID) 回传给控制器，以便将报文路由回手持终端。

——接收方 (Receiver)：在如今具有先进技术框架与云延伸技术的商业环境中，单个 BOD 可路由到多个目的地或接收方。在此环境下，发送系统无法确定 BOD 所有可能的目的地，但为了审计追踪和云追踪，应用域中包含接收方元素，以便需要在云端或“发布-订阅”基础设施中重复使用。接收方的每次出现都可以使用多种 ID 方式识别接收系统。

- 逻辑ID (LogicalID)：标识接收该BOD的接收系统，能用于建立由逻辑到物理的映射关系。其使用是可选的。每个系统或系统组合宜维护映射表，记录集成配置中应用系统的ID、逻辑名称或逻辑地址，以便将逻辑名称映射到物理网络地址。
- 构件ID (ComponentID)：提供比逻辑标识符更精细的控制级别，代表接收BOD的业务构件。其使用是可选的。
- ID：提供比逻辑和构件标识符更精细的控制级别，代表可用于细化接收BOD应用程序的重复ID。其使用是可选的。命名宜与发送系统的构件名一致。

——创建日期时间 (CreationDateTime)：创建特定 BOD 实例的日期时间戳。一旦赋值，该日期在 BOD 存续期间不应被修改。日期时间格式支持 GB/T 7408.1-2023 的日期和时间格式。

——签名 (Signature)：BOD 需要签署时使用。使用哪种数字签名，由用户及其集成需求决定。签名支持任意数字签名方式。资质机构 (qualifyingAgency) 用于标识提供签名格式的机构。

该元素允许通过外部 XML schema 命名空间声明来指定要使用的数字签名，而不是实际定义内容，从而实现了此元素的适应性。签名元素可以包含来自任意其他命名空间的任意内容。用户在 BOD 实例中可以携带数字签名。有关 W3C 的 XML 签名规范的更多信息，参见 [13]。

——BOD 标识符 (BODID)：承载通用唯一标识符 (UUID)，使每个 BOD 都能被唯一识别。尽管它是可选的，但需要 UUID 来实现以下服务或功能：

- 具有法律约束力的交易；
- 交易日志；
- 异常处理；
- 条件触发的重新发送；
- 报告；
- 确认；
- 安全；
- 云跟踪。

——扩展 (Extension)：通常仅当需要在一个构件中添加更多或不同的数据时使用，是扩展方法的核心特性。

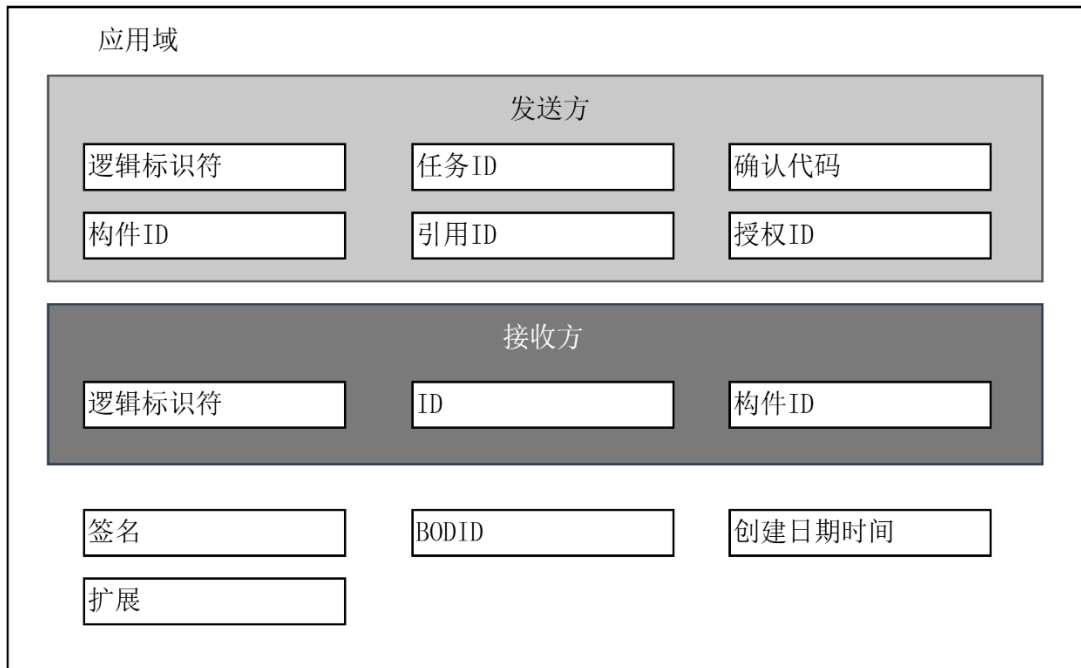


图5 应用域

5.3.2.3 数据区

实际业务信息承载于数据区中（见图4）。数据区通过自身的两个子部分，满足业务报文传递的另外两项独特需求。通常要求对正在交换的业务数据执行特定操作。

——预期操作通过动作来定义。它明确了要对所包含的业务信息执行的操作。表 1 给出了不同的动作，以及所使用的请求和响应动作的组合。

动作元素中还包含与动作指示的行为完全相关的附加信息。如，处理动作表示请求方要求接收方系统执行特定任务，且该任务支持确认与回执。

——业务信息本身包含在另一个子部分，即业务对象名称中。业务对象名称由可重复使用的业务数据和结构分组组成，这些数据和结构满足特定业务步骤或操作。业务对象名称本身由常见的、可重复的构件组成，这些构件可由更小的和字段组成。同样，业务对象名称本身也可由字段组成。

- 构件由其他通常更小的构件构成，而这些构件又能包含更小的构件与字段。每个构件都是通用的可重复业务元素，用于实现业务对象名称的部分业务功能（如采购订单头、行、地址、联系人等）。大多数构件是BOD共用的基础构建块。

构件的特殊含义来自其上下文的使用（部分由其所在的元素或构件决定）。为了促进构件重复使用，除需要新增独特构件外，不宜使用额外描述符。外层的“容器构件”决定了内层构件的值。因此，订单构件中的产品线构件以通用的构件数量来表示该产品线上被订购的“事物”的数量。

- 字段也是用于创建构件和业务对象名称的基础元素。

表1 动作组及示例

请求动作	动作响应	示例
Process（处理）	Acknowledge（确认）	请求：ProcessPurchaseOrder（处理采购订单） 响应：AcknowledgePurchaseOrder（确认采购订单）
Post（处理） （财务场景中 “Process”的同义词）	PostAcknowledge（过账确认）	请求：PostJournalEntry（过账日记账分录） 响应：PostAcknowledgeJournal（过账确认日记账）
Change（变更） （包含增/改/删/替换 选项）	ChangeAcknowledge （变更确认）	请求：ChangePurchaseOrder（变更采购订单） 响应：ChangeAcknowledgePurchaseOrder（变更确认采购订单）
Cancel（取消） （取消）	CancelAcknowledge （取消确认）	请求：CancelPurchaseOrder（取消采购订单） 响应：CancelAcknowledgePurchaseOrder（取消确认采购订单）
Sync（同步） （包含增/改/删/替换 选项）	SyncResponse（同步响应）	请求：SyncCustomerAddress（同步客户地址） 响应：SyncResponseCustomerAddress（同步响应客户地址）
Load（加载） （财务场景中 “Sync”的同义词）	LoadResponse（加载响应）	请求：LoadActualLedger（加载实际分类账） 响应：LoadResponseActualLedger（加载响应实际分类账）
Get（获取）	Show（展示）	请求：GetPurchaseOrder（获取采购订单） 响应：ShowPurchaseOrder（展示采购订单）
Notify（通知）	无需响应	NotifyRemittanceAdvice（通知汇款通知单）

5.3.2.4 扩展

架构的最后部分满足了业务级标准特有的需求，此类需求在通信、交通等低层级、非业务类标准中并不存在。该架构既适配差异化需求，又提供了具备互操作性的基础框架。各行业需求存在差异是共识；本文件既面向跨行业的通用需求，也覆盖具体业务线的需求。架构应支持以标准化方式实现报文定制，也应支持通过通用方法和构件实现扩展。

扩展能力允许产业及价值链在自身流程中，使用标准构件插入独特且可重复使用的非标准信息。

扩展能力通过一套基于可重复使用的构件的通用方法实现，构件能按需插入至相应位置，并支持进一步扩展。本文件在提供并维护基础标准的同时，提供了这种按需适配的能力。就此而言，本文件在业务类标准和在满足需求层面具有独特性。

附录 A

(资料性)

物理层 RESTful web 报文

A.1 概述

RESTful Web报文架构基于HTTP规范中描述的HTTP报文架构构建。然而，有几个与API设计相关的问题在规范中并未涵盖，而是将这些问题交由使用者自行处理。其中部分问题包括：API版本管理、URI设计与格式约定、资源管理方式（如，如何在读取请求报文中表达筛选与选择条件、如何表达分页条件与结果）、资源表示约定、附加请求头与响应代码的采用及使用。快速的网络搜索通常会得到大量的处理这些RESTful web API设计相关问题的方法，但并无明确的统一标准。RESTfulWeb报文架构通过补充细节来解决这些问题，这些补充细节既保持了与HTTP规范的一致性，又对其HTTP报文架构进行了扩展。5.2.3概述了RESTful Web报文架构的主要架构元素。完整规范的详细内容可在OAGi获取。

A.2 报文头

RESTful Web报文的报文头与BOD应用域(见5.3.2.2)的作用相似。本文件明确了一组可作为RESTful WEB报文中HTTP-报文头的标头。这些标头包括：HTTP规范中定义的标头子集、某些并非HTTP规范内容但已向互联网工程任务组（IETF）提交并提议的特定标头、一组自定义的标头。

A.3 报文动作

RESTful Web报文的报文动作与BOD数据区（如上所述）的动作作用相似。本文件明确了一组可作为HTTP-请求报文的HTTP-起始行（即请求行）中方法的动作。这些动作包括：HTTP规范中定义的方法、并非HTTP规范内容。此外，除了这些方法，本文件还支持控制器资源的概念。控制器资源是一个过程性概念，它并不对应任何标准或提议的HTTP方法。

A.4 报文体

RESTful Web报文的报文体与BOD数据区（见5.3.2.3）中的业务对象名称或构件作用相似。RESTful web报文的HTTP-报文体可包含业务对象名称和/或构件。需要注意的是，构件可表示业务数据（如地址）或响应元数据（如分页响应）；在后一种情况下，报文实例可同时包含一个业务对象名称实例和一个构件实例，或两个构件实例。

参 考 文 献

- [1] IETF RFC 8259, The JavaScript Object Notation (JSON) Data Interchange Format
- [2] W3C TR/2006/ REC-XML11:20060816, Extensible Markup Language (XML) 1.1 (Second Edition)
- [3] W3C TR/2012/REC-xmlschema11-1-20120405, W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures
- [4] ISO/IEC 20802-1:2016, Information technology — Open data protocol (OData) v4.0 — Part 1: Core
- [5] 开放应用程序接口规范 <https://swagger.io/specification/>
- [6] 超文本传输协议 I. E. T. F., HTTP/1.1, <https://datatracker.ietf.org/doc/html/rfc2616>
- [7] HTTP 语义 <https://datatracker.ietf.org/doc/html/rfc9110>
- [8] GB/T 17969.8 信息技术 对象标识符登记机构操作规程 第8部分：通用唯一标识符（UUIDs）的生成及其在对象标识符中的使用
- [9] SCHEMA J. S. O. N. <https://json-schema.org/specification.html>
- [10] GB/T 18221-2000 信息技术 程序设计语言、环境与系统软件接口 独立于语言的数据类型
- [11] GB/T 18391.5-2009 信息技术 元数据注册系统（MDR） 第5部分：命名和标识原则
- [12] GB/T 19256.5-2022 基于XML的电子商务 第5部分：核心构件规范
- [13] W3C TR/2013/REC-xmlsig-core1-20130411, XML Signature Syntax and Processing Version 1.1
- [14] NIST网络安全框架1.1版, <https://www.nist.gov/cyberframework/framework>
- [15] IETF RFC 7231, Hypertext Transfer Protocol — HTTP/1.1: Semantics and Content
- [16] FIELDING R. T., Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine, 2000.
- [17] W3C TR/2012/REC-xmlschema11-1-20120405, W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures
- [18] GB/T 7408.1-2023 日期和时间 信息交换表示法 第1部分：基本原则
- [19] Garden of Eden Pattern, <https://www.oracle.com/technical-resources/articles/java/design-patterns.html>
- [20] OData Version 4.0. Part 2: URL Conventions Plus Errata 03 <http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part2-url-conventions.html>
-