

基于 Project Haystack 开源数据建模标准的建筑系统和设备数据建模指南规范

v2017-08-28 由北京水木天工科技有限公司翻译

1.0 目标：这个模型标准的目的是提供一套统一的、标准化的方法，用于命名和描述数据点，数据点用于表示包括设备自动化系统、设备系统、能源计量系统、移动设备在内的其他智能设备等相关信息，与设备相关的描述性信息，也可称为元数据。

1.1 背景：现代自动化系统和智能设备可以轻松收集大量数据，包括环境条件，设备运行状态，能源用量和效率等。然而事实上，这些数据通常没有统一标准，或以低级自定义的格式存在，如果不通过投入大量人力，很难对其进行趋势判断，性能分析，或生成有用的报告及可视化工作。结果是现在大量的非结构化数据充斥在我们周围，但我们却不能轻易地从中获得价值。

将智能设备的数据转化为可操作的智能数据的第一步是给出数据“上下文”，以便我们准确地理解每条数据的意思，以及其整个系统中的含义。例如，如果需要将空调机组的送风温度和回风温度进行比较，我们如何实现？目前，我们了解一个数据点的含义，通常是通过破译当初由系统集成商在系统调节阶段随意起的名称，例如“DA_TEMP”。

鉴于以前针对建筑系统和设施运维相关的数据点命名还没有共同接受的统一标准，所以行业面临的一个关键挑战是建立一套常用的词汇集，用于描述不同建筑系统收集的信息含义。从复杂系统的角度来看待这个挑战，显而易见，即使能够标准化命名，也不可能仅通过数据点名称来获取所需的全部信息。例如，建立一个完整的送风系统模型时，需要确定每台空调机组（AHU）对应了哪些变风量系统（VAV）末端；或者在配电系统中需要了解支路电表和设备负载之间的具体关系。建立了这样的模型，可以从系统层面，建筑层面，甚至从整个建筑投资组合的层面进行数据分析。这些信息如果都通过点的名称来描述，将使点的名称长得无法接受，因此需要对数据点结构化。

Project Haystack 的使命是定义一种方法和常用词汇表，使建筑系统和智能设备的模型可以实现在各种不同软件和基于 Web 的应用程序中实现自动解释。这将使业主，运营商，制造商和服务提供商更有效地从智能系统收集来的海量数据中获得价值。

Haystack 标准涵盖了建筑系统和相关智能设备的整个价值链。业主和咨询公司通过对建筑内楼宇自控系统和类似系统中包含的数据以 Haystack 标准约定，从而确保形成有统一标准的系统数据组织，

并易于与外部应用集成。Project-Haystack 组织还促进了 Haystack 语义标签与其他相关标准的“映射”。

2.0 技术概述

2.1 总体设计概念：对于建筑物和设备系统的 Haystack 数据建模标准，是基于时下广泛接受的“标签”概念定义的，如下所述。

标签：标签是以“键/值”对定义的，与 AHU（空调机组），电表等实体相关联。标签非常简单，并且可动态调整结构，可以灵活的为不同系统和设备建立标准化模型。标签是一种建模技术，可以轻松地在每个任务、每个项目或每个设备的基础上定制数据模型，同时保留供外部应用程序使用标准的定义的方法和词汇表来解释的能力。标签应支持以下基本数据元素的定义：

实体：实体是现实世界中物体的抽象。实体包括站点，设施，设备，传感器点，气象站等。在软件系统中，建模后的实体可以是数据库中的一条记录，也可以是楼宇自动化系统中的一个对象，或者可能只是 csv 文件中的一行代码或一页电子表格。

Id: id 标签用于使用 Ref 值类型对系统中实体的模型进行唯一标识。Ref 值类型由单独的应用决定。实体的范围可能未定义，但在给定的系统或项目中必须是唯一的。该标识符可在其他实体或者系统中关联和引用。

Dis: Dis 标签与实体一起使用来定义用于描述实体的显示文本。Dis 值应该设置的比较短（少于 30 或 40 个字符），并为用户完整描述实体。

标签种类：定义标签应遵循如下提供的标签值类型：

Marker: 这个标签类型是一个没有对应值的标记标签。Marker 标签用于表示一个“类型”或“是”关系。

Bool: 布尔值“true”或“false”。

Number: 用于测量计数的整数或浮点数，通常为所测量的值定义了度量单位。

Str: 表示一个 Unicode 字符。

Uri: 统一资源定位符

Ref: 引用另一个实体。这个标签类型用于相互引用实体，虽然 Haystack 项目目前尚未规定具体的说明或参考机制。引用标签的格式需要使用前导符“@”，同时要使用特定的 ASCII 字符子集：a-z，A-Z，0-9，下划线，冒号，破折号或点。

Bin: 二进制标签，具有格式为 Bin (MIME / plain) 的 MIME 类型的二进制数据类型。

Date: 日期标签，参照 ISO 8601 规定的日期格式：年，月，日，例如：2011-06-07。

Time: 时间标签，参照 ISO 8601 规定的时间格式：小时，分，秒，例如：09:51: :27.354。

DateTime: 数据时间标签，参照 ISO 8601 规定的时间，时间戳后跟时区名称，例如：

2011-06-07T09:51:27-04:00 New_York

2012-09-29T14:56:18.277Z UTC

使用示例: 以下为描述一个站点实体的示例：

```
id: @whitehouse
dis: "White House"
site
area: 55,000 ft2
geoAddr: "1600 Pennsylvania Avenue NW, Washington, DC"
tz: "New_York"
weatherRef: @weather.washington
```

该示例展现的实体由七个标签描述：id, site, dis, area, geoAddr, tz 和 weatherRef。通常，每个标签为一行或用逗号分隔。site 标签没有明确的值，它被认为是 Marker 标签。dis, geoAddr 和 tz 标签具有字符串值，通常由双引号指示，也可以进一步标准化。area 标签以数值结合 ft² 或 m² 的单位表示。weatherRef 标签是对另一个实体的引用，通常用“@”字符开头表示。

2.2 标准标签库和可扩展性标准库: Project Haystack 数据建模标准提供了一个标准标签的综合库，用于描述常用设备，系统和设备类型。标准开发社区提供了公开的论坛，用于行业专家和有关各方针对各种设备，系统和应用程序的标签和标准模式进行讨论，提交，微调并最终批准标签。讨论过程是公开的，透明的，以便持续开发和完善这种分类法，使其能够对行业内外的普遍的设施工程数据进行语义定义。

3.0 REST API

Project Haystack 数据建模标准提供了一套文档化的表示状态转移,应用程序编程接口(REST API),定义了一种简单的机制在 Web 服务上交换“标签化”数据。

REST 服务器被编程实现一系列“操作”。操作是接收请求并返回响应的 uri。标准操作定义为查询数据库,设置订阅或读/写时序数据的历史。操作是可插入的,供应商可以自定义增值功能的开放性 REST 接口,以实现其更多的业务应用。

请求和响应都被建模为网格 (grids)。网格使用标准的多用途 Internet 邮件扩展 (MIME) 类型进行网格序列化编码,可以通过 HTTP 内容协商和其他标准化 Web 服务协议进行插入。

Project Haystack REST API 的“操作”的设计更像是一个远程过程调用 (RPC) 模型,“REST”这个词是用来区分不同的设计,从传统的 Web 服务,使用可扩展标记语言 (XML)、简单对象访问协议 (SOAP),和其他的互联网标准,虽然目前的设计可以很容易地通过这些技术实现。

4.0 应用

Haystack 数据建模标准的目标是确保构建系统、设备和相关数据的一致性建模。下面的应用需求概述了建模标准在建筑物、能源和设施管理中的应用。

4.1 最小模型要求: Haystack 项目的实施应该利用定义的数据建模标签来创建一个扩展的、连贯的模型,在面向设施的应用程序中使用以下最小项、层次结构和关系:

Sites: 包括显示名称,描述,大小(面积)这三个作为最少条件。强烈推荐参考互联网可用的气象站,并创建标签反映站点的其他相关特征,如建造时间,设施使用类型,占用类别,运行时间表,建筑系统类型(例如,中央暖通空调(HVAC))

Equipment: 包括通过 ID 引用和显示名作为站点的标准化关联。还推荐设备和软件供应商、型号、安装年份和类似的描述性元数据。

Points: 包括通过 ID 引用,与站点和设备的标准化关联,计量单位作为最小条件。在可能的情况下,推荐增加可接受值的范围。

4.2 通过 REST API 访问 Project Haystack 模型：软件和 Web 服务应用程序包括控制系统设备将使用 Project Haystack 标准发布的 Project Haystack REST API 来公开上述模型的定义，在地址：<http://project-haystack.org/doc/Rest> 公开并保持更新。

4.3 软件参考实现：Project-haystack 标准提供了在 Java 中的一个参考实现，对应用程序的 Haystack REST 协议的实现提供的示例代码。

4.4 商业可用产品的开源代码模块。Project Haystack 社区已经开发并提供了一个全面实施的 Haystack 协议，基于 NiagaraAX 系统的软件模块。该模块称为 NHaystack，根据学术免费许可证(“AFL”) 3.0 版许可。可通过 project-haystack.org 站点公开访问 NHaystack 软件模块并进行维护。

当使用 NiagaraAX 的系统时，在基于 NiagaraAX 的设备和其他软件应用间读取数据或将命令写入 NiagaraAX 系统，NHaystack 模块应该是首选通信方法。

5.0 开源

Project Haystack 为智能设备和工程数据的标准数据建模方法，标准，支持文件和参考实现提供了开源许可并免费。

5.1 开源许可证使用开源计划学术免费许可证 3.0 模型。有关许可条款的详细信息，请访问：<http://project-haystack.org/doc/License> 和 <http://opensource.org/licenses/AFL-3.0>