

Project Haystack オープンソースデータモデル化標準に基づくビルシステムおよび機器のデータモデル化ガイド仕様¹

Guide Specification for Data Modeling of Building Systems and Equipment Based on Project Haystack Open Source Data Modeling Standard

v2021-03-16 OcubeC 西田竹志 (OcubeC: Tak NISHIDA) ダイキン工業 齋藤裕也 (Daikin: Yuya SAITO)

1.0 目的: データモデル化標準の目的は、ファシリティ自動化システム、機器システム、エネルギー計測システム、および可搬型設備を含むその他のスマートデバイスに関連するデータポイントの名前付けと記述、およびメタデータと呼ばれる該データに関連した情報の一貫性のある標準化された方法論を提供することです。

1.1 背景: 近年のビルシステムは、最新の自動化システムとスマートデバイスにより、環境状況、機器の動作状態、エネルギー使用量とパフォーマンスなど、大量のデータを簡単に収集できるようになりました。しかし、現状これらのデータは標準または一貫した統一性のない低レベルのアドホック形式で存在するため、多大な人手を介した作業なしで、その傾向の解釈、分析、有用なレポート作成や可視化することは困難です。その結果、私たちは大量の非構造化データが溢れる海の中で、そこから有益な価値を容易に導き出すことはできない状況にいます。

スマートデバイスのデータを、行動に移すことができるための知識に変換するための最初のステップは、データにそのデータに纏わる各種状況・文脈を表す「コンテキスト(context)(あるいはコンテキスト)」を付与することが重要です。これにより、各データが「意味していること」とシステム全体での位置付けを正確に把握することができるようになります。例えば、エアハンドリングユニット (AHU) の給気 (Discharge Air) 温度を還気 (Return Air) 温度と比較したい場合に、どのようにしてこの情報を見つけることができるでしょうか？今日、データポイントが何を意味するかを示す唯一の指標は、システムインテグレーターが付けたデータポイントの任意の名前(例えば給気温度であれば"DA_TEMP")から類推・解読することによって見つけるしかない手段がないのです。

ビルシステムとファシリティエンジニアリングに関連するデータポイントの命名規則について、これまで広く合意された標準がなかった状況を鑑みると、業界が直面する重要な課題の一つは、さまざまなビルシステムによって収集されたデータに意味を与える共通の語彙を確立することです。しかし複雑なシステムの観点から考えると、語彙が標準化されたとしても、データポイント名だけでは必要な記述情報を完全に取り込むことができないことが明らかです。例えば、どの AHU がどの VAV(Variable Air Volume)と繋がっているかを定義して空気配送システムを完全モデル化、または電気システムのサブメーターと機器の負荷とのすべての関係を知ることはできるでしょうか。これらの関係をモデル化することで、システムレベル、ビルレベル、またはビルのポートフォリオ全体に亘るビル運用の分析を行うことが可能となるのです。このような広範囲の情報を把握するには、ポイント名だけを結合して表現すると、とてつもなく長くなるためデータ間の構造化も必要となってくるのです。

組織としての Project Haystack のミッションは、メタデータ定義の方法論と共通語彙を提供することです。これにより人手を介することなく、ビルシステムとそれに関連するスマートデバイスのモデルを種々のソフトウェアやウェブベースのアプリケーションで自動的に解釈できるようになります。ビル所有者、オペレーター、メーカー、およびサービスプロバイダーは、スマートシステムが収集している膨大な量のデータからより効率的に価値を引き出すことができるようになります。

¹ 本書は Project Haystack のデータモデル化ガイド“Guide Specification for Data Modeling of Building Systems and Equipment Based on Project Haystack Open Source Data Modeling Standard. [原文](#)”を日本語日本語に翻訳。

Project Haystack は、ビルシステムとインテリジェントデバイスのバリューチェーン全体を総括し、ビルオーナーやコンサルタントがビル自動化システムや同様のシステムに含まれるデータに対して Project Haystack 命名則に準拠した名前付けを可能とします。これにより標準化に基づいたシステムデータの構築と外部アプリケーションとの統合を容易にします。また Project-Haystack では、他の関連するビルデータ標準と Haystack セマンティックタグ付けとの「マッピング」も行っています。

2.0 技術概要

2.1 全体的な設計コンセプト：ビルおよび設備システムの Project Haystack データモデル化標準は、広く受け入れられている「タグ」の概念に基づく単純なメタモデルを使用します。

タグ(Tags)： タグは名前と値の組合せであり、AHU、電力メーターなどのエンティティに関連付けられています。タグは単純かつ動的であり、構造化を可能とし、さまざまなシステムや機器の標準モデルを構築するために必要な柔軟性を具備します。タグを利用することによって、標準の定義された方法論と語彙を使用して外部アプリケーションによって解釈することができる機能を維持しながら、タスク毎、プロジェクト毎、または機器毎にデータモデルを簡単にカスタマイズできるモデル化手法です。タグは、下記の基本データ要素の定義をサポートします。

エンティティ(Entity)： エンティティは、実世界の物理オブジェクトを抽象化したものです。エンティティには、サイト、ファシリティ、機器、センサーポイント、気象観測所などが含まれます。ソフトウェアシステムでは、エンティティはデータベースのレコード、ビルオートメーションシステムのオブジェクト、または csv あるいはスプレッドシートファイルの行として記述されることがあります。

識別子(Id)： id タグは、Ref 値型を使用してシステム内のエンティティの一意の識別子をモデル化するために使用されます。Ref 値型は、個々のアプリケーションによって決定されます。エンティティのスコープは未定義の場合がありますが、特定のシステムまたはプロジェクト内で一意である必要があります。この識別子は、エンティティ、関連付け、およびシステムを相互参照するために他のエンティティによって使用される場合があります。

記述子(Dis)： dis タグは、エンティティを説明するための表示テキストを定義するためにエンティティとともに使用されます。Dis 値は短い（30 文字または 40 文字未満）ことを目的としています。人間のユーザーがそのエンティティを完全に理解できるように記述能力がある必要があります。

タグの種類 (Tag Kinds)： 本標準は下記の許可されたタグ値型を提供します。

- Marker：このタグ型は、意味のある値を持たない単なるマーカー注釈です。マーカータグは、“type”または“is-a”の関係を示すために使用されます。
- Bool：“true”または“false”のブール値。
- Number：測定単位に関連づけられた整数または浮動小数点数。理想的には、測定単位はさまざまなタスクで規定されます。
- Str：Unicode 文字の文字列。
- Uri：ユニバーサルリソース識別子(Universal Resource Identifier)。
- Ref：他エンティティへの参照。Project Haystack 仕様は現在、特定の ID または参照メカニズムを規定していませんが、エンティティを相互リンク付するために必要とされます。参照は先頭に「@」を付けてフォーマットされ、ASCII 文字の特定のサブセットが使用されます：a-z、A-Z、0-9、アンダーバー、コロン、ダッシュ、またはドット。

- Bin : Bin (text / plain) としてフォーマットされた MIME タイプのバイナリブロッブ(binary blob)。
- Date : ISO 8601 標準に基づく年、月、日の組合せで表現される日付。例えば 2011-06-07。
- Time : ISO 8601 標準に基づく時間、分、秒の組合せで表現される時刻。例えば 09 : 51 : 27.354。
- DateTime : ISO 8601 標準に基づくタイムゾーン名付きタイムスタンプ。
(例) 2011-06-07T09 : 51 : 27-04 : 00 New_York
2012-09-29T14 : 56 : 18.277Z UTC

使用例: 以下に、サイトを表現するエンティティの例を示します。

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

この例では、id、site、dis、area、geoAddr、tz、weatherRef の 7 つのタグを持つエンティティを示しています。慣例により、例を書くとき、各タグは個別の行にリストされるか、コンマで区切ります。サイトタグには明示的な値はなく、マーカータグと見なされます。Dis、geoAddr、および tz タグには文字列値があり、その多くは他機関によって標準化されている可能性があり、通常は二重引用符で示されます。エリアタグには、平方フィートまたは平方メートルの単位を持つスカラーで示される数値があります。weatherRef タグは別のエンティティへの参照であり、通常は "@" 文字を使用して示されます。

2.2 タグ標準ライブラリとライブラリ拡張性: Project Haystack データモデル化標準では、一般的な機器、ビルシステム、およびデバイス種類に対応するための標準タグの包括的なライブラリを提供しています。今後の新たな機器システム対応のライブラリ拡張、方法論の充実を目指して、オープンなディスカッションフォーラムである標準化開発者コミュニティを設置しています。本フォーラムでは、ビルシステム業界の専門家や標準化に興味を持つ人たちが、さまざまな種類の機器、システム、およびアプリケーションに対応するために、追加タグまたは標準スキーマについて話し合いを行っています。フォーラムでは新たなタグ、スキーマ提案、調整作業を通じて、包括ライブラリに含めるための合意形成ができるようにします。オープンフォーラムのプロセスは、業界内外の人たちのファシリティエンジニアリングデータの意味的な理解を可能とするデータの分類法(taxonomy)の継続的な開発を可能にするために、フォーラムでの議論は透過的であるようにしています。

3.0 REST API

Project Haystack データモデル化標準は、タグ付けされたデータを Web サービス間でやりとりするための簡易な機構として、REST API (Representational State Transfer, Application Programming Interface)を用います。

REST サーバーは、一連の操作(ops/ operations)を実装するようにプログラムされます。操作は、要求を受信、応答を返す uri です。標準的な操作は、データベースの検索、サブスクリプションの設定、または時系列データの読み取り/書き込みに対して定義されています。オペレーションはプラグイン可能であるため、ベンダーは独自のビジネス目的のためにカスタマイズされた付加価値機能を備えた REST インターフェースに拡張することができます。

要求と応答の両方がグリッド(grid)としてモデル化されます。グリッドは、シリアライズ用(serialize)の標準 MIME タイプを使用してエンコードされ、HTTP コンテンツネゴシエーションおよびその他の標準化された Web サービスプロトコルを使用してプラグインできます。

“ops”設計を利用する Project Haystack REST API は、RPC(Remote Procedure Call)モデルに似ていますが、REST という用語は、拡張マークアップ言語 (XML) , SOAP(Simple Object Access Protocol)、およびその他のインターネット標準を使用する従来の Web サービス設計と区別するために使用されます。ただし、現在の設計では、これらの従来テクノロジーを簡単にトンネリングすることができます。

4.0 アプリケーション

Project Haystack データモデル化標準の目標は、ビルシステム、デバイス、および関連データの矛盾のないモデル化を保証することです。下記のアプリケーション要件は、ビル、エネルギー、およびファシリティ管理に関連するアプリケーションでのモデル化標準の利用の概要を示しています。

4.1 最小モデル要件: ファシリティ 指向のアプリケーションで使用する場合に、 Haystack プロジェクトの実装では、下記の最小限の項目、階層、および関係を備えた、拡張性があり首尾一貫性のあるモデルを作成するために定義済みのデータモデル化タグを利用します。

サイト(Sites)： 表示名(display name)、説明(description)、サイズ (面積) (size (area))は最小限含めます。インターネットで気象データが取得可能な気象観測サイトへの参照を強く勧めます。また、ビル建設年、施設の使用タイプ、占有クラス、運用スケジュール、ビルシステムタイプ (例えばパッケージまたはセントラル HVAC) など Site の特性を表現するタグを生成することも必要です。

装置(Equipment)： id 参照を介したサイトとの標準化された関連付けと表示名を最低限含みます。機器およびソフトウェアのベンダー、機種番号、インストール年、および同様の説明的なメタデータも推奨されます。

ポイント(Points)： id 参照を介したサイトおよび機器との標準化された関連付け、測定単位を最小限含みます。可能な場合は、許容値の範囲の記述も推奨されます。

4.2 REST API を介したプロジェクト Haystack モデルの公開: 制御システムデバイスを含むソフトウェアと Web サービスアプリケーションは、Project Haystack 標準の一部として公開された Project Haystack REST API を使用して、上記のようなモデル定義を公開します。 <http://project-haystack.org/doc/Rest>

4.3 ソフトウェア参照実装: Project Haystack 標準は、Java 環境で動作する参照実装を提供しています。本参照実装では、ソフトウェアアプリケーションで Haystack REST プロトコルを実装するためのサンプルコードを公開しています。

4.4 市販製品のオープンソースモジュール

Project Haystack コミュニティは、Niagara AX ベースのシステムで使用するための Haystack プロトコルのソフトウェアモジュール開発を行い、オープンソースとして公開しています。[NHaystack](#) として知られるモジュールは、Academic Free License ("AFL") v.3.0 の下でライセンスされています。NHaystack ソフトウェアモジュールへのパブリックアクセスは、ソフトウェアの維持管理を行う project-haystack.org サイトを介してされるものとします。

Niagara AX ベースのシステムを使用する場合、NHaystack モジュールは、ソフトウェアアプリケーションが Niagara AX ベースのデバイスからデータを読み込んだり、制御コマンドを Niagara AX ベースのシステムに書き込んだりするための通信に推奨される方法です。

5.0 商用製品に利用可能なオープンソース: Project Haystack ファシリティエンジニアリングデータモデル化標準は、スマートデバイスデータモデル化のための方法論、標準化、サポートドキュメント、および参照実装は、オープンソースライセンスを前提として無料で誰でもが利用できるようになっています。

5.1 オープンソースライセンスは、[Open Source Initiative Academic Free License v3.0](https://opensource.org/licenses/AFL-3.0) モデルを使用するものとします。ライセンス条件の詳細については、次の URL をご覧ください。 <http://project-haystack.org/doc/License>、 <http://opensource.org/licenses/AFL-3.0>