

# Lignes directrices pour la modélisation des données des équipements et systèmes en mécanique du bâtiment basée sur le standard de modélisation à code source ouvert Project Haystack

v2014-12-02, traduit de l'anglais par Christian Tremblay, ing.<sup>1</sup>

**1.0 Objectif:** L'objectif de la standardisation de la modélisation des données est de fournir une méthodologie uniforme et normalisée afin de décrire la signification des points et des données qui y sont rattachées, dans les domaines de l'automatisation des procédés, des équipements, des systèmes de mesurages énergétiques de même que des divers appareils "intelligents" et/ou mobiles. La norme de modélisation présente une méthode de représentation descriptive des informations relatives à ces équipements ou systèmes, plus généralement connue sous l'appellation "métadonnées" (metadata).

**1.1 Historique:** Les appareils intelligents et les systèmes modernes d'automatisation ou de régulation ont rendu accessible l'acquisition d'une vaste quantité de données dans les domaines aussi variés que les conditions météo, d'environnement, d'opération d'équipements, de consommation énergétique et de performance. La réalité, cependant, c'est que ces données existent à différents niveaux, sous différents formats, sans normes ou organisation spécifique, rendant ainsi leur utilisation extrêmement difficile. Un travail manuel de mappage des données très complexe précèdera toute visualisation ou analyse dans le but de générer un rapport utile. En bref, nous avons aujourd'hui accès à un volume de données immense mais ces données n'ont pas de structure et il est difficile d'en extraire de la valeur.

La première étape vers la valorisation des données est de **contextualiser** ces données afin de pouvoir connaître leur **signification**, ce qui nous amènera à mieux les **situer** dans un système global. Par exemple, pour comparer les températures d'alimentation et de retour d'une unité de ventilation, nous devons d'abord identifier le point du système de contrôle représentant chacune de ces informations. Dans les systèmes de contrôle actuels, l'indication la plus probante est sans doute le nom du point qui aura été donné lors de la configuration initiale du système, par exemple:

```
DA_Temp  
DA-T  
Disch_Temp  
nvoDischargeTemp
```

Seul un opérateur averti, familier avec la terminologie utilisée dans un projet en particulier sera à même de trouver rapidement l'information recherchée et/ou les liens existants entre chaque équipements ou systèmes. Dans plusieurs cas où des systèmes

---

<sup>1</sup> Certains termes sont gardés en anglais afin de faciliter la compréhension du standard. Bien que les termes puissent être traduits, l'implémentation du standard se fera en anglais dans les systèmes pour assurer la compatibilité.

ont été déployés sur plusieurs années, on remarquera différentes nomenclatures selon la technologie employée ou l'installateur.

Le défi principal de l'industrie est donc d'établir un vocabulaire commun pour décrire l'information acquise par les différents systèmes de contrôle. Un constat évident, dans une perspective de systèmes complexes, est le fait que même si nous normalisons les noms de points, nous ne pourrions y inclure toute l'information requise pour décrire complètement chacun de ces points. Comment modéliser un système complet de distribution d'air en y incluant les liens entre l'unité de ventilation et les boîtes à volume variables qui y sont rattachées? Comment décrire complètement les liens existants entre les appareils de mesures électriques globaux et les appareils mesurant chaque panneau ou chaque charge? Cette complexité nous amène au-delà de l'information disponible à l'échelle du nom des points de contrôle.

La modélisation des relations entre les équipements permet l'analyse de l'opération au niveau des systèmes, du bâtiment ou d'un ensemble de bâtiments. Les noms de points qui pourraient offrir autant d'informations seraient très longs et nécessiteraient une structure extrêmement complexe.

La mission de Project Haystack est de définir une méthodologie et un vocabulaire commun de manière à permettre aux informations d'être claires, descriptives et d'être analysées de façon automatique par des logiciels ou des applications web. Ceci permettra aux propriétaires, aux opérateurs, aux manufacturiers et aux autres fournisseurs de services de pouvoir extraire les informations ayant de la valeur parmi toutes les informations colligées.

Project Haystack englobe actuellement toutes les sphères des systèmes reliés aux bâtiments et des équipements dits "intelligents", permettant ainsi aux propriétaires et consultants de spécifier la norme de modélisation Haystack aux données contenues dans les systèmes de gestion du bâtiment (BAS) ainsi que des autres systèmes similaires afin d'assurer une organisation normalisée et cohérente des données de même qu'une intégration facile aux applications externes. Project Haystack facilite le mappage des données (sémantique Haystack) aux autres standards de l'industrie.

## 2.0 Aperçu technique de la modélisation standard des données

**2.1 Concepts de base:** Le standard de modélisation à code source ouvert Project Haystack est fondé sur le modèle-méta, lui-même basé sur le concept globalement accepté des “étiquettes” (tags) tel que décrit ci-bas:

**Tags** (étiquettes) : Une étiquette est une paire nom/valeur pouvant être associée avec des entités comme des unités de ventilation (AHU), compteurs électriques, sondes et autres. Les étiquettes sont simples, dynamiques, ajoutent de la structure et fournissent la flexibilité nécessaire à l'établissement d'un modèle auto-descriptif de divers systèmes et équipements.

Les étiquettes doivent supporter la définition des éléments de données essentiels suivants:

**Entity** (Entité): Une entité est l'abstraction d'un objet physique du monde réel. Les entités incluent les sites, usines, équipements, sondes, stations météo, etc. Dans un système informatique, une entité pourrait être modélisée par une entrée de base de données (enregistrement), un objet dans un système de régulation automatique (ex. consigne) ou une simple ligne dans un fichier CSV ou un tableur.

**Id:** L'étiquette **id** (identifiant) est utilisée pour modéliser l'identifiant unique d'une entité dans un système assurant un lien unique via une valeur de type **Ref** (référence). Les valeurs de type Ref sont déterminées selon l'application. La portée d'une entité peut être indéfinie, mais doit être unique à l'intérieur d'un système ou d'un projet. L'identifiant peut être utilisé par d'autres entités afin d'illustrer la relation entre entités ou systèmes.

**Dis:** L'étiquette **dis** (display / affichage) est utilisée avec une entité afin de définir le texte affiché pour décrire l'entité. Les valeurs **dis** doivent être courtes (30-40 caractères) mais décrire complètement l'entité pour le lecteur.

**Tag kinds** (types d'étiquettes): Le standard fournit les types d'étiquettes suivants:

- **Marker** (marqueur): cette étiquette est en fait l'équivalent d'une case à cocher sans signification particulière. Cette étiquette est utilisée afin d'illustrer un type ou une relation de type “est-un” (“is-a”).
- **Bool** (booléen): vrai ou faux
- **Number** (nombre): entier ou flottant présenté avec ou sans unité de mesure (préférentiellement avec)
- **Str** (string / chaîne de caractères): Une chaîne de caractères de type Unicode

- **Uri** (Universal resource identifier): Chaîne de caractères à syntaxe normalisée qui sert à désigner de façon unique et permanente une ressource par sa localisation ou par son nom.
- **Ref** (référence): Référence à une autre entité. Project Haystack ne définit pas actuellement de méthode de référencement ou de relation entre entités. Les étiquettes ref doivent être utilisées dans ce sens. Les étiquettes ref sont formatées:
  - Commencent par un “@”
  - suivi de caractères ASCII (a-z, A-Z, 0-9, \_, :, -, .)
- **Bin** (binaire): Un “blob<sup>2</sup>” binaire muni d’un type MIME Bin(text/plain)
- **Coord** (coordonnée): coordonnées géographiques latitude/longitude formatées tel C(lat/Ing)
- **Date**: une date en format ISO 8601 (année, mois, jour / 2015-02-05)
- **Time** (temps): une entrée de temps en format ISO 8601 (heure, minute, secondes / 09:51:27.354)
- **DateTime** (date-heure): une entrée en format ISO 8601 incluant date et heure suivies du fuseau horaire:
  - 2011-06-07T09:51:27-04:00 New\_York
  - 2012-09-29T14:56:18.277Z UTC

### Exemple d’utilisation

Cet exemple présente une entité décrivant un site:

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

L’entité est décrite par 7 étiquettes soit: *id,dis,site,area,geoAddr,tz,weatherRef*. Par convention, les exemples seront toujours présentés avec chaque étiquette sur une ligne distincte ou séparées par une virgule. Le tag “*site*” n’a pas de valeur en soit. Il s’agit d’un marqueur (marker). Les étiquettes *dis, geoAddr, tz* possèdent des valeurs de type Str (chaînes de caractères) pouvant être standardisées par d’autres, typiquement présentées encadrées de guillemets (“”). L’étiquette *area* (surface) a une valeur de type numérique et une unité de mesure (pieds carrés). L’étiquette *weatherRef* est une référence à une autre entité, facilement reconnaissable par le symbole “@”.

---

<sup>2</sup> **BLOB**, pour **(en)** *Binary Large Object*, mais qui s’écrit couramment en caractères minuscules, est un [type de donnée](http://fr.wikipedia.org/wiki/Binary_large_object) permettant le stockage de données [binaires](http://fr.wikipedia.org/wiki/Binary_large_object) (le plus souvent des fichiers de type image, son ou video) dans le champ d’une table d’une [base de données](http://fr.wikipedia.org/wiki/Binary_large_object). [réf: [http://fr.wikipedia.org/wiki/Binary\\_large\\_object](http://fr.wikipedia.org/wiki/Binary_large_object)]

**2.2 Librairie des étiquettes standards et évolutivité de la librairie:** Le standard de modélisation à code source ouvert Project Haystack fournit une librairie compréhensible d'étiquettes standards afin de décrire les éléments communs des équipements, systèmes et appareils. Une communauté d'utilisateurs sont engagés dans une discussion ouverte afin de permettre aux différents acteurs de l'industrie de commenter, proposer et éventuellement approuver des étiquettes additionnelles ou schémas pour décrire d'autres équipements, systèmes ou applications. Le processus de discussion se veut transparent et permet un développement continu de la nomenclature qui permettra une compréhension des données sémantiques dans l'industrie, par ses acteurs et les intervenants externes.

### **3.0 Interface de programmation REST (REST API)**

Le standard de modélisation à code source ouvert Project Haystack inclut une interface de programmation de type REST définissant les mécanismes d'échange des données munies d'étiquettes via les services web standards (comme HTTP).

Les serveurs REST sont programmés pour répondre à des ensembles d'opérations (ops). Ces opérations sont en fait des adresses URI qui reçoivent des requêtes et retournent des réponses. Les opérations standards sont définies pour interroger la base de données, souscrire en temps réel à des mises à jour d'informations ou encore, consulter ou créer des historiques ou des séries temporelles (time-series data). Les opérations sont conçues afin de permettre aux différents vendeurs de services d'étendre leurs fonctionnalités par l'utilisation d'applications tierces.

Les requêtes et les réponses sont construites à l'aide de grilles (grids). Les grilles sont encodées selon la norme MIME (Multipurpose Internet Mail Extensions) et peuvent être étendues via une négociation de contenu HTTP ou autres protocoles Web.

L'interface de programmation REST du Project Haystack utilise un modèle d'opérations (ops) très près d'un type RPC (Remote Procedure Call) mais le terme REST est tout de même privilégié afin de noter les distinctions avec les autres services Web traditionnels utilisant XML (Extensible Markup Language), SOAP (Simple Object Access Protocol ou les autres, bien que le design actuel permette une compatibilité avec ces technologies. Pour de plus amples détails sur l'interface de programmation REST du Project Haystack, consulter le site web à l'adresse suivante: <http://project-haystack.org/doc/Rest> (en anglais).

## 4.0 Applications

L'objectif du standard de modélisation à code source ouvert Project Haystack est de s'assurer d'une méthode de modélisation cohérente et normalisée des systèmes, appareils, équipements et leurs données associées dans le domaine du bâtiment. Le point suivant présente quelques exigences requises lors de l'implantation du modèle standard dans une application reliée au bâtiment, à sa consommation énergétique et à sa gestion.

**4.1 Exigences minimale du modèle:** L'implantation de Project Haystack utilisera les étiquettes normalisées définies afin de créer un modèle pouvant être élargi, qui sera cohérent et inclura les items de base suivants:

**Sites:** Un site est un emplacement (bâtisse, usine, pavillon, etc) qui sera traité comme une entité unique. L'étiquette "Sites" inclut au minimum le nom, une description et une surface. Il est fortement recommandé d'y lier les informations de météo correspondante de même que certaines informations telles: l'année de construction, type d'utilisation, type d'occupation, horaires, type de systèmes (unités de toit, géothermis,...), etc.

**Equip (Équipements):** L'étiquette "Equip" représente un assemblage mécanique ou un appareillage électrique ayant au minimum les informations suivantes: un nom et la relation à un site (via une étiquette siteRef). Les autres informations utiles sont souvent: le vendeur, le numéro de modèle, l'année d'installation et autres éléments pouvant décrire l'équipement ou sa fonction.

**Points:** Les points sont en fait sondes, consignes, commandes... Les points auront au minimum une relation à un équipement (equipRef) ou un site (siteRef) et une unité de mesure. Lorsque possible, une échelle des valeurs possibles est recommandée.

**4.2 Exposition du modèle Project Haystack à l'aide de l'interface de programmation REST:** Les logiciels, services Web et/ou les appareillages de contrôle seront en mesure d'exposer le modèle défini ci-haut en utilisant l'interface de programmation REST de Project Haystack. Cette interface de programmation REST est publiée comme partie intégrante du standard Project Haystack, ouvert, accessible et mis à jour ici: <http://project-haystack.org/doc/Rest>

**4.3 Référence d'implémentation logicielle:** Le standard Project Haystack présente une référence d'implémentation logicielle en Java fournissant des exemples de codes pour l'implantation du protocole REST Haystack dans d'autres applications ou logiciels.

**4.4 Modules à code source ouvert disponibles:** La communauté Project Haystack a développé et rendu disponible une implémentation du protocole Haystack sous la forme d'un module pouvant être utilisé avec les systèmes Niagara AX. Le module nHaystack est disponible sous licence AFL (Academic Free License v.3.0) L'accès et la maintenance est effectuée via le site <http://project-haystack.org>.

Dans le cas des systèmes utilisant Niagara AX, le module nHaystack devrait être la méthode retenue pour assurer la communication entre ces systèmes et les autres application Haystack interagissant avec eux.

Autres modules à venir

## **5.0 Code source ouvert**

Le standard de modélisation à code source ouvert Project Haystack, incluant les normes, la méthodologie, la documentation et les références d'implémentations devront être disponibles sans frais sous licence à code source ouvert.

**5.1:** La licence à code source ouvert sera du type Open Source Initiative Academic Free License 3.0. Les détails complets des termes de cette licence sont disponibles à l'adresse suivante: <http://project-haystack.org/doc/License> et <http://opensource.org/licenses/AFL-3.0>

Fin du document