

Visualisation Node-RED de données sémantiques du bâtiment

Ghislain Ateamezing¹, Yann Gripay², Sylvie Servigne², Frédérique Laforest²

¹ MONDECA, 18 rue de Londres, Paris, France

² Univ Lyon, INSA Lyon, CNRS, UCBL, LIRIS, UMR5205, F-69621 Villeurbanne, France

Résumé

Ce travail présente l'architecture d'une application de visualisation de données de capteurs dans le domaine du bâtiment intelligent. Cette visualisation utilise des modules de l'architecture du Servient de la plate-forme CoSWoT et le framework Node-RED pour faciliter la mise en place d'une visualisation de données de capteurs annotées avec des ontologies du domaine, notamment SOSA, SSN, BoT.

Mots-clés

Web of Things, Semantic Web, Servient, Dashboard, Smart building.

Le projet CoSWoT et les données du bâtiment : Le projet CoSWoT¹ a pour objectif d'intégrer les objets de l'IoT dans une architecture distribuée reposant sur les outils du Web Sémantique. Dans ce projet, une implémentation est en cours dans le domaine du bâtiment intelligent. Les développements présentés ici se concentrent sur un cas d'utilisation dérivé de données captées dans un bâtiment du LIRIS. Les capteurs déployés mesurent les conditions ambiantes telles que la température, l'humidité, la luminosité et le dioxyde de carbone de l'air.

L'architecture déployée tient compte des contraintes des objets, depuis la collecte jusqu'à la visualisation des informations. Un Servient a été développé et déployé pour gérer les différents capteurs, en utilisant le framework node-wot² pour interagir avec les capteurs. Node-wot est une mise en œuvre des normes du web des objets dans NodeJS. Il prend en charge plusieurs protocoles tels que MQTT, COAP, HTTP et différents types de médias tels que le texte, JSON. Il suit un modèle de conception qui garantit que les différents protocoles sont utilisés via la même API.

Le Servient inclut un module chargé de convertir les données brutes en RDF : il collecte les données des capteurs toutes les 15 minutes via une requête à l'API et les transforme en RDF en utilisant des ontologies bien connues du domaine (SOSA/SSN, BoT, SAREF, etc.), augmentées d'une ontologie spécifique. Il les stocke ensuite dans un triple store RDF dédié. Le format JSON est utilisé pour échanger les données entre les différents éléments de l'architecture.

Les propriétés des données JSON peuvent être utilisées pour filtrer les données, par exemple par intervalle de

temps, selon la localisation du capteur, selon le type de capteur...

Solution : L'objectif est de créer une interface utilisateur qui permet la consultation des données des différents capteurs du bâtiment intelligent en offrant des fonctionnalités de filtrage des données pour cibler des créneaux horaires spécifiques, des pièces spécifiques, des valeurs seuils, etc. La solution déployée permet ainsi aux occupants du bâtiment de visualiser les données des capteurs à l'aide d'un tableau de bord personnalisé présentant les faits selon leurs préférences (localisations, seuils) à différentes échelles de temps et d'espace.

Nous avons utilisé Node-RED³ pour créer le tableau de bord. Editeur basé sur un navigateur, Node-RED facilite le développement graphique d'applications orientées flux de données.. Cela permet de réduire le temps nécessaire à la création d'une application fonctionnelle. Le framework inclut des modules de tableau de bord, notamment node-red-ui-table, qui correspond pleinement à notre besoin.

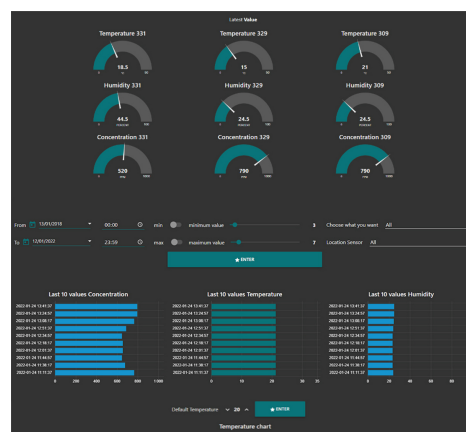


FIGURE 1 – Vue d'une partie du dashboard

Conclusion : Dans le cadre du projet CoSWoT, nous sommes intéressés à un use case autour du bâtiment intelligent offrant un tableau de bord personnalisé pour le suivi de données issues de capteurs du bâtiment. La solution développée est disponible en ligne⁴, avec des exemples sur des données brutes collectées. L'étape suivante pour notre projet est de faire une version distribuée de ce use case et d'étudier la collaboration inter-servients.

1. <https://coswot.gitlab.io/> CoSWoT est financé par l'agence nationale de la recherche sous la référence ANR-19-CE23-0012.

2. <https://github.com/eclipse/thingweb.node-wot>

3. <https://nodered.org/>

4. <https://smartbuilding-dasboard-liris.research.mondeca.com/>