



Intégration de la gestion des caméras UPnP dans OpenHab

— Bruel Anna - Medewou Cenzo - —
Ndiaye Yacine

Plan

- I. Contexte
 - I.1. Présentation du projet
 - I.2. Matériels
 - I.3. Logiciels

- II. Déroulement du projet
 - II.1. Prise en main
 - II.2. Création du Binding
 - II.3. Création de l'interface de contrôle
 - II.4. Catalogue de référencement des caméras
 - II.5. Récupération du résultat de découvertes via l'API REST

- III. Conclusion
 - III.1. Problèmes rencontrés et solutions proposées
 - III.2. Perspective

- IV. Démonstration



I.Contexte

I.1.Présentation du projet

- Détection automatique des caméras UPnP
- Affichage des caméras découvertes sur l'interface OpenHab.
 - ◆ sous forme de liste
 - ◆ sous forme de mosaïque
- Visualisation du flux vidéo des caméras sur l'interface OpenHab.
- Contrôle de la camera tilt/pan via l'interface OpenHab

I.2. Matériel

Caméras supportant le protocole UPnP



DSC-932L



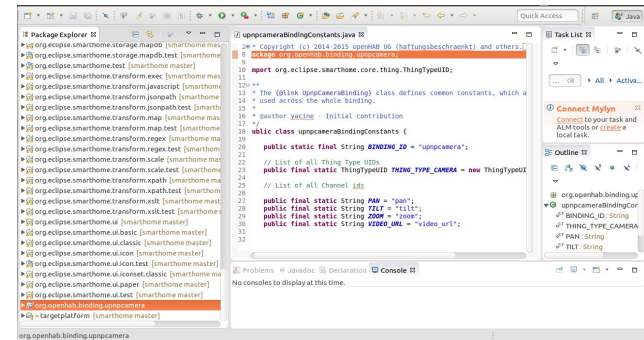
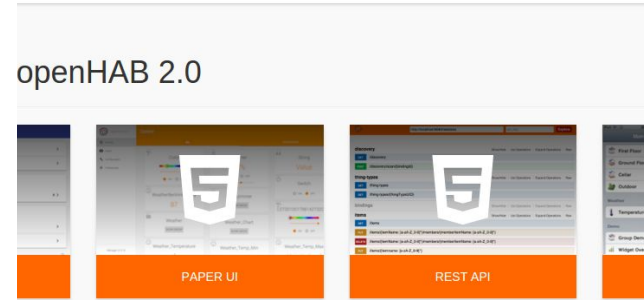
DSC-5222L
équipée d'un tilt/pan

I.2.Logiciels

→ OpenHab : logiciel d'automatisation pour les maisons connectées

→ Eclipse Smart Home : Framework et environnement de développement

Empowering the Smart Home



II. Déroulement

II.1.Prise en main

- Documentation et première approche (OSGI,openHab, Prototypage ...)
- Prise en main de l'interface D-Link(visualisation video, test pan/tilt sur les caméras...
- Apache Félix : Test de détection des appareils UPnP
- Réalisation d'un bundle permettant de détecter les appareils UpNp et récupération des infos sur ces appareils par des commandes.
- Mise en place IDE Eclipse SmartHome pour la réalisation du binding de découverte des caméras.
- Passage de OpenHab1 à OpenHab2



II.2.Création du binding

- Squelette de création de eclipse smarthome
- Définition des “things” et “channels”

II.3.Création de l'interface de contrôle

- mise en place d'une page HTML de contrôle des caméras
- Modification de la widget: Webview
- ajout de 2 packages :
 - ◆ org.eclipse.smarthome.ui.classic.rest.database
 - ◆ org.eclipse.smarthome.ui.classic.rest.render

II.4.Catalogue de référencement des cameras

- fichier camera.properties
 - ◆ sert à référencer les caméras que le binding peut découvrir
 - ◆ représente une base de données pour les caméras et contient donc les URLs de video, image, pan, tilt... de chaque caméra
- Une convention bien définie pour renseigner une nouvelle camera :
model|video_url|image_url|pan_url|tilt_url

II.5.REST API

- API fourni par Eclipse SmartHome
- Répond aux requêtes http
- Réponse en JSON

III. Conclusion

III.1.Problèmes rencontrés et solutions proposées

- Rétro-ingénierie difficile sur Eclipse Smart Home
 - ◆ Beaucoup de temps sur la compréhension du code de OpenHab.
- Configuration des caméras impossibles avec Wifi Campus
 - ◆ hotspot avec nos téléphones
- Zoom et tilt de la caméra DCS-5222L ne fonctionnent pas
 - ◆ Pas de solution même après upgrade, mais les fonctionnalités sont prises en compte par le projet
- Balise vidéo de OpenHab ne supporte pas le format cgi
 - ◆ Utilisation du widget Webview

III.2.Perspective

- Intégrer le projet sur myOpenHab
- Afficher le flux vidéo sur la smart TV de la salle Air
- Contribution au projet Eclipse SmartHome
 - ◆ Eclipse Smart Home : 123 000 lignes
 - ◆ Contribution : 1164 lignes

IV. Démonstration

Merci pour votre attention
