



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Okosotthon rendszerek konfigurációja

Jankó András I. évf, MSc mérnökinformatikus szakos hallgató

Konzulens: Dr. Ráth István, MIT

Kritikus rendszerek specializáció

Önálló laboratórium 2. összefoglaló

2018/19. I. félév

A félév során egy okosotthon rendszerek konfigurációját megkönnyítő tool-t kezdtem el fejleszteni, ami modell alapú szoftverfejlesztés és generatív technikák alkalmazásával egyszerűsítene a folyamatot.

Okosotthon alatt mindenki mást ért. Az én értelmezésemben egy okosotthon nem attól lesz igazán okos, hogy vezérelni tudjuk benne az eszközeinket (pl. lámpák, fűtés), hanem képes észlelni, hogy használni szeretnénk őket, vagy szükségünk lesz rájuk, és ennek megfelelően cselekedni. Azonban, hogy magától képes legyen reagálni a környezeti változtatásokra, szükség van egy okosotthon rendszerre és annak konfigurációjának elkészítésére. A probléma, hogy ennek a konfigurációnak az elkészítése igen nehéz folyamat. A félév során ebben a témakörben kutattam, hogy milyen megoldást lehetne találni erre a problémára.

A félév során több okosotthon rendszer működését kipróbáltam, hogy nagyobb rálátásom legyen arra, hogy mi okoz problémákat, melyek azok a részek amik ismétlődnek, és esetleg kiválthatók lennének valami mással.

Arra jutottam, hogy többek között az eszközök felvétele elég redundáns és nehézkes. Itt mindig meg kell adni az eszköz kijelzett nevét, egy egyedi azonosítót, amivel a rendszerben hivatkozhatunk rá, kommunikációs paramétereket, pl MQTT-n melyik csatornán érhető el, és az ott érkező adatokat hogyan értelmezze/konvertálja, és opcionálisan csoportokba rendezni. Ez két okból is szükséges, egyrészt a megjelenítő felületek akár tudnak egy automatikus megjelenítést készíteni, például ha szobákba rendezzük az eszközeinket.

Arra a következtetésre jutottam, hogy az eszközök konfigurációját meg lehetne fogalmazni eggyel magasabb szinten is, nem csak az okosotthon rendszerek specifikus nyelvén. Ebből a magasabb szinten megfogalmazott konfigurációból egyértelműen le lehetne generálni az adott okosotthon nyelvén is a rendszer konfigurációját. Ezzel kiküszöbölhetnénk a szintaxis hibákat, és még validációt is vezethetünk be a rendszerbe az általános szinten, például hogy az eszköz ID-je egyedi-e.

A félév során elvégzett munkám:

- Elkészítettem egy Xtext alapú domain specifikus nyelvet (DSL) az általános okosotthon konfigurációk leírására. Ezt az Xtext-es DSL-t Language Server Protocol (LSP) szerver részébe csomagoltam, hogy újrahasználható legyen a nyelv különböző kliensekkel is.
- Elkészítettem egy VIATRA *Model To Model* transzformációt, ami az általános nyelvből előállítja OpenHAB konfigurációt, és LSP-n keresztülküldött Command hatására, és a kliensnél meg is jelenik egy fájlként.
- Elkészítettem egy Visual Studio Code extension-t, ami az LSP kliens oldali részét implementálja. Ez használja az LSP szerver oldali funkcióit, hogy IDE funkcionalitásokat nyújtson, amivel jelentősen megkönnyíti a konfiguráció elkészítését. Továbbá tartalmaz Syntax Highlight-ot is, ami alaphoz nem része az LSP szerver oldalának, amit az Xtext generál.

A jövőbeli fejlesztésekként szeretném megvalósítani, hogy ne csak az eszközök konfigurációját lehessen így generálni, hanem például az automatizációs logikát is, vagy például a felhasználói felületek leírót, így egy teljese értékű okosotthon rendszert be lehetne már vele konfigurálni.