A Demo: Semantic-Based Re-Engineering of Automation Systems

Aparna Saisree Thuluva[1,2], Kirill Dorofeev[3], Monika Wenger[3], Darko Anicic[1], Sebastian Rudolph[2]

[1]Siemens AG - Corporate Technology, [2]TU Dresden, Germany,
[3]Fortiss An-Institut Technische Universität München, Germany,
aparna.thuluva@siemens.com, dorofeev@fortiss.org, wenger@fortiss.org,
darko.anicic@siemens.com, sebastian.rudolph@tu-dresden.de

Abstract. Manufacturing processes are being digitalized and automated using Automation Systems. But Automation Systems are complex and they contain lots of under-used equipment. The cost and time of manufacturing can be reduced if the under-used equipment on Automation Systems is used efficiently. Discovering the existing functionalities on an AS and installing new functionalities with low effort plays an important role to efficiently use an Automation System equipment. In this demonstration, we will present an approach employing Web of Things and Semantic Web technologies, to make an Automation System equipment transparent, and present how the effort of installing new functionalities on the field devices can be lowered if the Automation System equipment is transparent.

1 Introduction

Industry 4.0, which is also referred to as the fourth Industrial revolution aims at lowering the time and cost of manufacturing individualized products [1].[1] But an Automation System (AS) used in a manufacturing process posses under-used equipment, which when used efficiently can reduce the cost and time of manufacturing products. This poses certain challenges as Automation Systems (ASs) are complex, they contain lots of capabilities, configurations and functionalities running on them, which should be taken into consideration while installing new functionalities on them. Therefore, it is a challenge to: (1) discover and access functionalities on an AS, (2) create new functionalities for an AS, (3) check if the new functionality is compatible with the target AS. In this demonstration, we present how these challenges can be addressed by employing Semantic Web (SW) technologies and Web of Things (WoT) technologies. WoT and SW technologies are good candidates to model the information about an AS equipment and make it discoverable and accessible. W3C WoT working group[2] is standardizing the Thing Description (TD), which provides an abstract description of a

[1] http://ukmanufacturing2015.eng.cam.ac.uk/proceedings/Industry4.0AN10715.pdf
[2] https://www.w3.org/WoT/WG/

physical thing in terms of its interactions. TD enables interoperability between things belonging to different domains. TD of a device is stored on the device itself to access physical devices on the Web in a similar fashion as accessing Web pages using existing Web standards[3] [3–5]. On the other hand, employing SW technologies such as semantic enrichment of a TD and semantic based discovery enhances transparency and interoperability of an AS equipment and provides efficient discovery of an AS equipment. Employing these two techniques in this demo, we show the following: (1) how an AS equipment can be made transparent, (2) semantic modeling of a functionality, (3) how the effort of installing a new functionality on field devices of an AS can be lowered using semantic querying and reasoning techniques [2]. The demo will feature an engineering tool, which enable users without background in SW technologies to do engineering, configuration and re-engineering of ASs.

2 Architecture

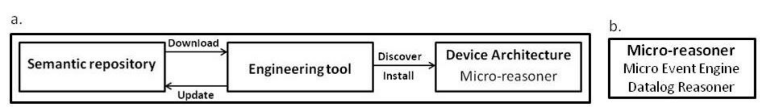

Fig. 1: The system architecture for semantic-based re-engineering of AS

We developed the architecture as shown in Figure 1a for semantic-based engineering of ASs. The first component is a semantic repository with RESTful interface, which is used to store semantic descriptions of new functionalities. The second component is semantic-based engineering tool that, provides a graphical user interface to do engineering with our semantic-based approach. The tool also provides an easy access to the repository and an AS to discover its equipment and functionalities or to deploy new functionalities on an AS.

The third component is the device architecture. The device here refers to an edge device that is embedded into an AS, which controls the operation of field devices (sensors and actuators) of an AS. Each edge device is endowed with a Micro-reasoner that empowers it to interpret TDs of the field devices and semantic models coming from the engineering tool, process the data from the field devices or to control them locally on the AS itself. Micro-reasoner as shown in Figure 1b is offered as a RESTful Web service. It consists of two components: **Micro Event Engine** implemented in C and a **the Datalog reasoner** to do reasoning in datalog [7], which is an open source C and LUA[4] implementation.[5] Micro Event Engine, which is based on the work from Anicic et al. [6] uses event rules to do Complex Event Processing (CEP) of field devices.

[3] http://mqtt.org/documentation
[4] https://www.lua.org/
[5] http://www.ccs.neu.edu/home/ramsdell/tools/datalog/datalog.html

3 Use Case

In this section we describe a use case for re-engineering an AS on the FESTO[6] Process Automation workstation. The workstation consists of two tanks, there are various sensors and actuators attached to the tanks to monitor their state. Among others, the workstation consists of an ultrasound sensor that measures level of liquid in a tank, float sensors, a pump and a pneumatic valve. We attached a micro-controller to each sensor and actuator of the workstation where the TD of the field device is stored and accessed. In this demo, FESTO workstation is equipped with three SIMATIC IOT2040[7], which are used as edge devices.

In our use case, initially field devices on the workstation are engineered to ensure overflow protection on the upper tank, using the float sensor on the upper tank and the pneumatic valve, such that when the float sensor detects overflow of liquid in the tank then the pneumatic valve lets liquid from upper tank to the lower tank. In this settings, if the float sensor is malfunctioning then, overflow protection cannot be ensured on the upper tank until the malfunctioning float sensor is replaced with a new sensor. In this demonstration, we will present how the ultrasound sensor deployed on top of the upper tank can be re-engineered with less effort, to be used in place of the float sensor to ensure overflow protection on the upper tank.

4 Demonstration Overview

A TD is created for each edge device, which describes the properties, events and actions of field devices controlled by the edge device. Figure 2a shows a snippet of a semantically enriched TD, which controls the ultrasound sensor. The TD is then converted to datalog facts and stored in the Datalog reasoner on the IOT2040 as shown in Figure 2b. In a similar fashion, new functionalities and their requirements are described semantically and stored in the repository. An example of a functionality used in this use case is: a field device should detect overflow status of a tank, if the level of liquid in a tank is over certain threshold. The requirement to install such a functionality on a device is that the device should be capable to measure level of liquid in a tank.

Having semantic descriptions of devices and functionalities, then re-engineering is done using engineering tool in the following steps: (1) the required functionality is discovered by an engineer from the repository, (2) semantic-based discovery and automated compatibility check is done to discover a suitable device on the FESTO workstation on which the new functionality can be installed, (3) the functionality is deployed on the workstation from the engineering tool. The Micro-reasoner on the edge device interprets the semantic description of the

[6] http://www.festo-didactic.com/int-en/learning-systems/process-automation/compact-workstation/mps-pa-compact-workstation-with-level,flow-rate,pressure-and-temperature-controlled-systems.htm

[7] http://docs-europe.electrocomponents.com/webdocs/1536/0900766b815365c3.pdf

a. Ultrasonic Sensor Thing Description


```
{ "@context" :
  ["https://w3c.github.io/wot/w3c-td-
      context.jsonld",
"http://SWAS/interactions/interactions-
      context.jsonld"],
"qu":
    "http://purl.oclc.org/NET/ssnx/qu/qu#",
"ssn":http://purl.oclc.org/NET/ssnx/ssn#,
"schema": "http://schema.org/",
"eclass": http://www.ebusiness-
      unibw.org/ontologies/eclass/5.1.4/#,
"name": "MyUltrasonicSensor",
"@type": ["ssn:Sensor",
    "eclass:C_AKE655002-tax"],
"uris" :
    ["coap://192.168.2.82:5683/ultrasonicSe
    nsor",
    "http://192.168.2.82:8080/ultrasonicSen
    sor"],
"encodings":["JSON"],
"ssn:onPlatform":"Tank1",
"ureasoner" : "true",
"properties":[ {
    "@id" : "level",
    "@type":
http://SWAS/interactions/liquidLevelProperty,
    "name" : "liquidLevel",
    "valueType" : {"type" : "float"},
    "writable" : "false",
    "qu:unit" : {"@type"  : "qu:millimetre"},
    "schema:minValue"  : "0.2",
    "schema:maxValue" : "800",
    "hrefs" : ["liquidLevel"] } ]}
```


b. Ultrasonic Sensor Datalog facts

```
name("td","MyUltrasonicSensor").
ureasoner("td","true").
uris("td","coap://192.168.2.82:5683/ultrasonicSensor").
onPlatform("td","Tank1").
properties("td","propetery1")
name("propetry1","liquidLevel").
hrefs("propetry1","liquidLevel").
@type("property1",http://SWAS/interactions/liqui
      dLevelPropetry").
```

Fig. 2: Thing Description of an edge device that controls ultrasound sensor

functionality and converts it into an event rule executable by Micro Event Engine. Therefore the functionality of malfunctioning float sensor is replaced with ultrasound sensor using semantic-based approach.

References

[1] Siemens: Modeling new perspectives: digitalization - the key to increased productivity, efficiency and flexibility (white paper). In: DER SPIEGEL (2015). (https://www.siemens.com/digitalization/public/pdf/FoM-modeling-new-perspectives.pdf)

[2] Thuluva, A.S., Dorofeev, K., Wenger, M., Anicic, D., Rudolph, S.: Semantic-Based Approach for Low-Effort Engineering of Automation Systems. In: Proceedings of ODBASE 2017 - The 16th International Conference on Ontologies, DataBases, and Applications of Semantics. Rhodes, Greece. Forthcoming 2017

[3] Fette, I., Melnikov, A.: The websocket protocol, RFC 6455. (2011)

[4] Belshe, M., Peon, R., Thomson, M., Melnikov, A.: Hypertext transfer protocol version 2.0. internet draft (2013). (https://tools.ietf.org/html/draft-ietf-httpbis-http2-04)

[5] Kovatsch, M., Duquennoy, S., Dunkels, A.: A low-power CoAP for contiki. Mobile Adhoc and Sensor Systems (MASS), 2011. In: Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (2011)

[6] Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Stream reasoning and complex event processing in etalis. Semantic Web 3(4) 397-407 (2012)

[7] Ceri, S., Gottlob, G., Tanca, L.: Logic programming and databases. Springer Science & Business Media (2012)