Automatic System Composition Methodologies for the Internet of Things

Jamil , Muhammad Zubair
Faculty of Electrical Engineering and
Information Technology
Technical University of Munich
Arcisstr. 21 D-80333 Munich
Email: ga53zej@mytum.de

***Abstract*—The internet of Things (IoT) has become a very promising technology with potential to revolutionized the world of electronic devices by connecting the billions of devices to internet. IoT resulted in enormous amount of heterogeneous communication protocols, middleware and hardware. In order to gain maximum out of IoT, an automated system composition is required Due the heterogeneity automation of the composition of a system out of the IoTs is still a hard nut to crack. This article provides an insight into prominent methods of system composition and the components of mechanisms.**

I. Introduction

The idea of composition is to make a system combining services offered by the IoTs. Automatic Composition of such system would not only assist engineers but also make us able to deploy the system in dynamic enviroment without reconfiguration. It would reduce the configuration (and reconfiguration) cost of systems as well, for example, in industrial production facilities [5], healthcare, and home automation. Owing to fact that different IoTs are forged by a diverse range of backgrounds and knowledge and that no universal standard was in place during the evolution of IoTs, it is difficult to compose a system [2].

Several methods have been proposed in by researcher to achieve the goal using different techniques. The key to a practical automated system composer is the requirements described in [1]. Overall, Each method of Automatic Composition comprises of four steps (discovery, semantics modeling, selection/planning, and execution) "or their variation. In discovery, available services on each device are discovered and expressed the context of each service using semantics. This also helps to abstract away the hardware, the communication protocol, and interface details and to bridge the Heterogeneity [1]. Selection/planning is the core of the Automatic Composition which takes goals as input and uses a middleware (usually called reasoner) to compose the list of services to be used with appropriate configurations. At last stage the services proposed by reasoner are orchestrated to achieve the goals.

Scenario: In this article we will be considering a scenario where a user wants to adjust the physical environment variables like temperature, lighting and music according to his mood.

The term composer is used to collectively denote a whole automatic system composition method through out the article.

II. Prominent Methodlogies

In section II different prominent methodlogies are described along with the middleware they are using.

A. Using RESTdesc and Reasoner

This method[3] is based on goal-oriented approach which lets the user specify coarse goals in an intuitional way. The composer comprises of a discoverer which can be used with any system that allows clients to search for URIs of service endpoints. For example, search engines like Dyser. In order to express high-level semantics of a particular service, it uses RESTdesc with extended capabilities to make it suitable for reasoning(selection and planning stage). RESTdesc is a method for defining hypermedia API features, allowingmachines to autonomously discover what an API is doing. As RESTdesc is based on first order logic, therefore it is not able to express mutually exclusive states. In order to represent such states RESTdesc is added is extended.For example, in the given scenario it might be requried to have a service which can convert temperature units.It's RESTdesc descrption is mention in the Figure: 1.

```
{
  ?tempC a dbpedia:Temperature;
         ex:hasValue ?cVal;
         ex:hasUnit "Celsius".
}
=>
{
  _:tempF a dbpedia:Temperature;
          ex:hasValue ?fVal;
          ex:hasUnit "Fahrenheit";
          owl:sameAs ?tempC.

  _:request http:methodName "GET";
            http:requestURI ("http://converter.example.
     com/cel2degf?temp=" ?cVal);
            http:resp [ http:body ?fVal ].
}.
```

Figure 1: Code for Temperature conversion Service

Once having all the services described, a reasoner is used to select services according to the given goals by the user. The goals are provided using graphical interface is provided using ClickScript(a visual programming tool based on javascript) for users to get goals from user. The Figure: 2 depicts the overall composer structure.

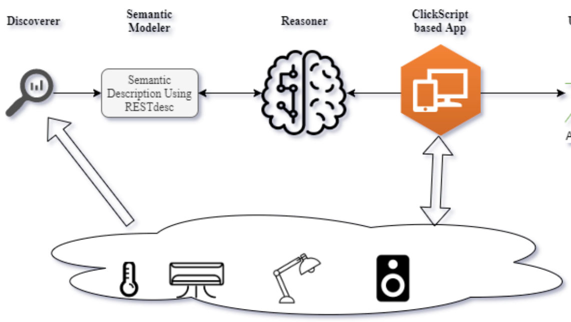


Figure 2: RESTdesc Composer Overview

B. *Using WoTDL and AI*

Embedding IoT with the Web is known as Web of Things. By this approach abilities of IoT devices are available through Web as web services which a big ease of access. WoT uses well established mechanics of Web technology and it helps to abstract away lots of hardware details of things connected to the internet and hence it is a good way to make interoperability of heterogeneous devices possible. Service description languages (e.g. Web Service Description Language) in the Web domain is good to describes services in the Web but they are not equipped enough to fully express IoT therefore Web of Things Description Language (WoTDL) has been created. Existing ontologies [7] neither support description of WoT and nor serve the requirements of WOT devices WoTDL provides a promising method to describe IoT services. Combining WoTDL with AI makes an automatic system composer.

1) **Core Concepts of WoTDL***:* Here are the core parts of concepts of WoTDL which are depicted in the figure:3 along with their relationships.
CompositeDevice : Expresses a physical hardware device which may be consisting of serval component.
State: Denotes the state of physical environment of user.
owl-s:Parameter: Specify the information available for the each state.
Measurement: It responds to a GET request and provide measured value of a parameter.
Sosa Actuation: It is used to trigger an action on a device for example turning a light "ON" or "OFF".
Transition : Describes the changes of states caused by the actuations. ***owl-s:Precondition:*** It obliges the system to check before a transition to be achieved. For example to turn "OFF" a light it's precondition should be stateON.

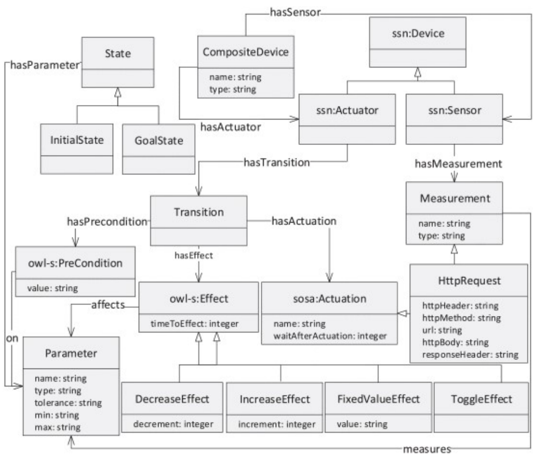


Figure 3

C. *EMAS*

Here IoT composer is using a multi-agent system (MAS) based approach to compose services.In this context multi agents refers to any of the entinties(targets, devices, servics, requests, and composers) involve in the system. MAS seems to be well known approach for service composition many solution have been proposed on the bases of this approach for example[9].

In this article, [8] has been discused which enhances MAS in such a way that it speeds up the composition process by categorizing services and it's named as Extended Multiagent System (EMAS). In this method service and request from user are treated into three subcategories(classes): trigger, processing and actuating. EMAS is achieveied by cordination of distributed agents as shown in Figure:6: A. Where as Figure::6:B and Figure:6:C shows request and service are treated into subcategories: trigger, processing, and actuating. However Figure::6:D depicts a subcategories of link agent. In order to realize the concept SysML (System Modeling Language) is used to design and Netlogo (a multi-agent programmable modeling environment) used to implement.

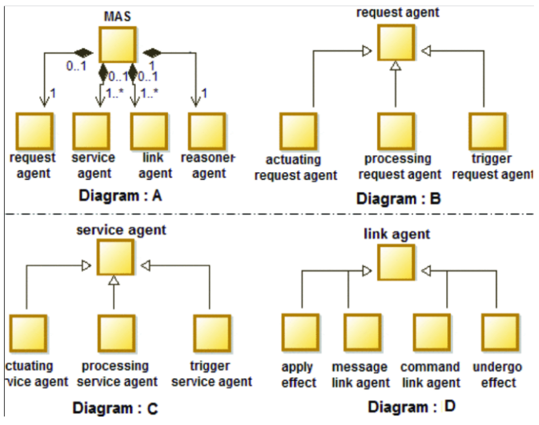


Figure 4

The figure **??** shows an example description of temperature service.

getTemp:(#,(Temp.,abstract,abstract),trigger,S-1,”Rafik”,02)
getAmb.Temp:(#,(Amb.Temp., double, ”C”), trigger,S-2,”Room”,0)
getB.Temp:(#,(B.Temp., double, ”F”),trigger,S-3,”Rafik”,0)

Figure 5

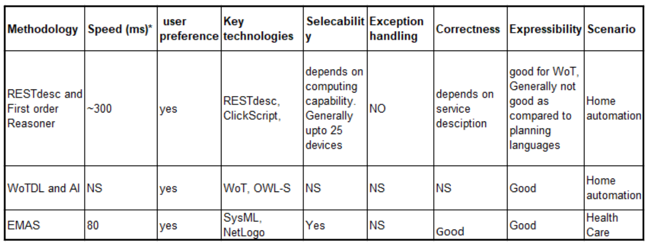
In order to select and plan services on the bases of user request a reasoner has been used. The pseudo-code of the reasoner has been shown in the figure: **??**. Basically what the reasoner is doing is taking user request and available service list as an input and find-out if there is any service that can be mapped according to the request.

```
Input: user's request, service lists
Output: agent composite (it represents a service composite)
    Initialisation :
 1: load the context
 2: check for mapping between (request agent, actuating
    agents)
 3: if (mapping is found) then
 4:     create link agents between request agent and the
        mapped agents.
 5:     check if composite is found.
 6:     if (composite is not found) then
 7:         check for mapping between actuating agents and
            processing agents.
 8:         check if composite is found.
 9:         if (composite is not found) then
10:             while composite is not found and set processing
                agents is not empty do
11:                 check for mapping between inter processing
                    agents.
12:                 check if composite is found.
13:             end while
14:         end if
15:     end if
16: end if
17: return agent composite
```

Figure 6

III. Discussion

First of all, clued above it is evident that goal-oriented -where user gives goals using a convenient environment which is usually a graphical interface- approaches for automatic system composition direct us towards the practical fully automated composition system. Where as process-oriented (user has to manually connect the services in order to achieve their goals) methods are not favorable for automatic composition.
In order to compare the methods described in section II. We are considering the following matrix: Speed of composition (in other words required time.), main technologies used, scalability, correctness, case scenario and expressiblity. Whereas scalability tells how suitable a particular method is to support up scaling (adding more devices and services into the system), correctness tells if the request was accomplished correctly(which could be messed up depending on how well the semantics have been described). Note that speed have been evaluated using different computation resources which may not provide accurate comparison between different methodologies. Figure: 7 1 illustrates the comparison.



Methodology	Speed (ms)*	user preference	Key technologies	Selecability	Exception handling	Correctness	Expressibility	Scenario
RESTdesc and First order Reasoner	~300	yes	RESTdesc, ClickScript,	depends on computing capability. Generally upto 25 devices	NO	depends on service desciption	good for WoT, Generally not good as compared to planning languages	Home automation
WoTDL and AI	NS	yes	WoT, OWL-S	NS	NS	NS	Good	Home automation
EMAS	80	yes	SysML, NetLogo	Yes	NS	Good	Good	Health Care

Figure 7: Comparison
speed for 1000 services
NS:Not Specified

IV. Conclusion

Fully Automatic IoT System Composition would be revolutionary. It would help in not only cutting down the reconfiguration of industrial processes but also it would bring new innovation for general consumers.
EMAS seems to be providing a good solution as it is modular and fast.Even though several solutions have been proposed for Automatic system composition, there are still several aspects of the Automation which are need to be address.

References

[1] Distributed Smart Space Orchestration,
Marc-Oliver Pahl,
http://mediatum.ub.tum.de/doc/1196145/78965.pdf

[2] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu,
“Web services composition: A decade’s overview,” Inform. Sci., vol. 280, pp. 218–238, 2014

[3] Smart Configuration of Smart Environments
Simon Mayer, Ruben Verborgh, Matthias Kovatsch, and Friedemann Mattern

[4] “Semantic web services in factory automation: Fundamental insights and research roadmap,”
J. L. M. Lastra and I. M. Delamer,
IEEE Trans. Ind. Informatics, vol. 2, no. 1, pp. 1–11, Feb. 2006.

[5] Semantics-Based Dynamic Service Composition
K. Fujii ; T. Suda
https://ieeexplore.ieee.org/abstract/document/1546104

[6] WoTDL: Web of Things Description Language for Automatic Composition
Mahda Noura,Martin Gaedke
https://dl.acm.org/citation.cfm?id=3352558dl=ACMcoll=DLpreflayout=flat

[7] A study of existing Ontologies in the IoT-domain
Garvita Bajaj,Rachit Agarwal,Rachit Agarwal,Pushpendra Singh,Valérie Issarny

[8] Extended multi-agent system based service composition in the Internet of things
Samir Berrani, Ali Yachir, Badis Djemaa,Mohamed Aissani
https://ieeexplore.ieee.org/document/8598503/authorsauthors

[9] ”A multi-agent system for dynamic service composition in ambient intelligence environments”,
M. Vallée, F. Ramparany, L. Vercouter,
Doctoral Colloquium - Pervasive 2005, 2005.