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.

Index terms

Internet of things, semantic technologies, service composition

**Section 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.

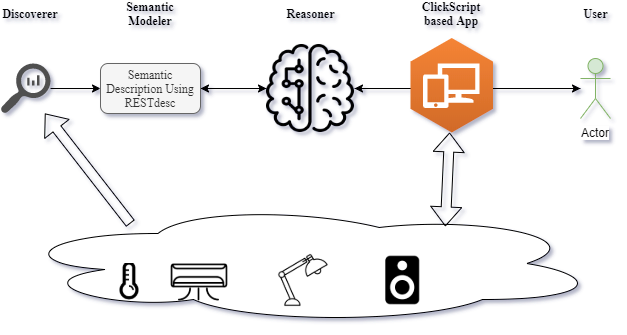
**Section II (Prominent Methodlogies)**

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

**A:Composition 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. Then 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 depicts the overall composer structure.



**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.

## **Core Concepts of WOTDL**

Here are the core parts of concepts of WoTDL which are depicted in the figure along wither their relationships.

**CompositeDevice** : Expresses a physical hardware device which may be consisting of serval component.

**State**: Denotes the state of physical environment of user.

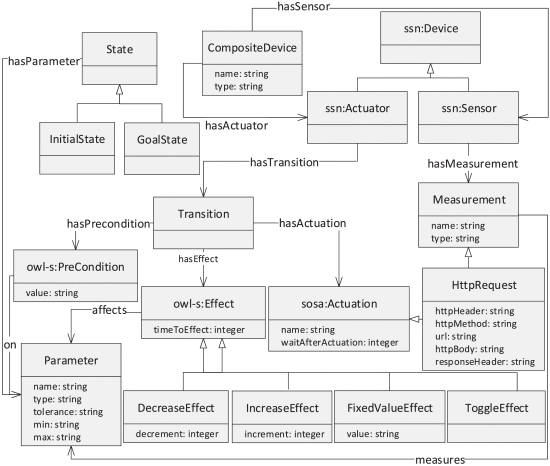
**owl-s:Parameter:**  Specify the information available for the each state.

**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.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/8904043/8909402/8909439/noura1-p413-noura-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/8904043/8909402/8909439/noura1-p413-noura-large.gif)

Using user defined goal (state) AI planning (which is functionally similar to the reasoner in the last section) yields a sequence of actions if the preconditions are in accord with the state transition. For example the user can define a state by mentioning the temperature and lighting settings.

C:

Discussion :

As 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Methodology | speed | User preference | limitation | Key technologies | scalability | correctness | expressibility |
| RESTdesc | Very fast (100s of ms: for upto 1000 services) | yes | Gets very slow with increase in number of devices | RESTdesc, | upto medium sized (250 devices) | depends on service annotation | good for WoT, Generally not good as compared to planning languages |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |