

Ad hoc Things Collaboration based on Semantic of things

- Use case of Internet of Things

Marie Kim

IoT Convergence Department,
Electronics and Telecommunications Research Institute,
DaeJeon, Korea Rep. of
mariekim@etri.re.kr

Hyo-Chan Bang

IoT Convergence Department,
Electronics and Telecommunications Research Institute,
DaeJeon, Korea Rep. of
bangs@etri.re.kr

Abstract—Internet of Things is everywhere nowadays in industrial fields, academic fields, and also in standardization fields. Most researches and products are about identification and connectivity aspects. If everything is connected and identified uniquely, then what can happen? This paper focuses on the usage of Internet of Things from a view point of personalized service provision by ad hoc things collaboration deployed by multiple providers. In this case, things are deployed without organized plans by multiple providers and shared among any users with appropriate permissions. Therefore, discovery, grouping, management and scheduling on things for collaboration are the main issues to resolve.

Keywords—internet of things, discovery, things, collaboration, semantic of things

I. INTRODUCTION

IoT (Internet of Thing) has many different definitions [1][2] by different organization. But common characteristic of Internet of things defined by different organizations is things-connectedness. In any case, everything should be connected to the Internet to communicate with other things.

There are many researches and standards regarding how to connect things to the internet. IETF CoAP[3] (Constrained Application Protocol) is the application layer messaging protocol to connect constrained things to internet with light RESTful framework. IETF 6LowPan [4] is the light IPv6 protocol for the constrained things to connect to internet. A smart gateway [5] and WoT Broker [6] are the gateways which bridge between different air protocol things (Bluetooth, IEEE802.11.4, WiFi) and web protocol things. Nowadays many researches focus on Web of Things [4][6][7] which tries to connect things with web protocols. Meanwhile, WoO (Web of Objects) project [8] tries to build things manipulation environment based on SoA (Service Oriented Architecture) philosophy.

When it comes to identification, identification requirements for IoT [9] are being analyzed and summarized now in ITU-T. In a web area, URI (Uniform Resource Identifier) is the identification scheme for all connected things. Until now, it is an open issue whether one powerful identification scheme prevails over the whole IoT things, or

there just needs a smart interpreter or translator between different service/network area identification schemes.

II. RELATED WORKS

A. COMUS

COMUS (Common Open semantic USN Service platform) is an open USN semantic service platform developed by ETRI since 2010. USN is ubiquitous sensor network and it is focusing on sensors but deals with actuators also. One big characteristic of COMUS is that users send queries to COMUS to get needed sensor data with abstract level query such as “get temperature values from room number1”. Then COMUS looks up the USN repository to get sensors’ identifiers located within a room number1 and sends sensor data requests to those sensors. This is called as a dynamic logical sensor group management in ITU-T F.OpenUSN[10]. Therefore, users don’t need to know details of sensor networks.

B. Social Web of Things and Device Sociality

Convergence of sociality and things are the hottest IT trend nowadays and it is a very interesting and realistic model because there need ways to share connected things everywhere based on some permission mechanisms. Social web of things [5][7] and device sociality[13] approach a things-sharing issue with social networks. Paraimpu[7] provides things-distribution mechanism based on social network service and provides a social mashup editor to operate things which are allowed to use based on social relationship. So users can share his/her things with friends and build things-mashup easily. Meanwhile, device sociality aims at building and managing social relationships of personal devices by using human relationships obtained by social networking services. Multiple devices of one user can share data, application or more resources. And different devices of different users but with human-friendship relationship can be shared among those users.

C. Other works

ISO/IEC 20005 (Information technology -- Sensor networks -- Services and interfaces supporting collaborative

information processing in intelligent sensor networks) is a standard about node grouping for collaborative information processing. Actually it does not include any logic or context but it defines related services and interfaces to manage dynamic node group for collaborative information processing.

There were other approaches for collaborative application. [12] deals with collaborative application in VANETs(vehicular ad-hoc networks). While driving, if gasoline is running out of, then the vehicle initiates communication with other vehicles to get information about cheap and close gas stations. Then other vehicles can give requested information to a requesting vehicle. The requester broadcasts queries to the certain amount of distance, so size of ad hoc group of vehicle can vary to the circumstance and members of the ad hoc group of vehicle can vary where the broadcasting happens.

D. ThingsDoWeb

ThingsDoWeb is a dynamic things collaboration platform which is under being developed by ETRI since 2013. It aims at providing environments where things collaborate with each other for seamless service provision based on given things environments by using web technologies. Things do web to collect information and to control other things for proving user-friendly seamless service provision without humans' intervention. To do that, *ThingsDoWeb* platform provides connectivity for all things and provides dynamic collaborative group management based on things' context (semantic of things) and users' preference. Ad hoc things collaboration based on semantic of things is one of the main research issues of *ThingsDoWeb* project.

III. AD HOC THINGS COLLABORATION BASED ON SEMANTIC OF THINGS

The everything-connectedness is the motto of Internet of Things. Assuming that everything is connected, what can be done by using things? It necessarily brings up *services* which use multiple things to provide specific functions for human beings. When it comes to service provision, many cases require multiple things to provide target services what users want to receive. At this point, the important thing to be considered is which things should collaborate with others to serve the target services. This is the *thing discovery, selection and grouping issues* and it highly depends on the availability of things at the service time. Especially, when we consider user's mobility, then available things are changed according to the location of users. Even though a user does not move, a thing can be out of order and a new thing can join to the given physical space. Then group of things for a target service is to be changed dynamically. Therefore, available things list and their status information should be managed in real time and target service executors should manage and schedule things collaboration accordingly. This is the *management and scheduling on things issues*.

Figure 1 depicts the ad hoc things collaboration life cycle. When target service initiation is triggered, first step is *discovery* of available things to use for the target services. When triggering is initiated, target service information and

user information need to be detected by using appropriate mechanism. RFID technology can be used to detect and identify users and newly designed things collaboration protocol can be used to detect target service information or users' manual operation can be used. Then, based on detected information regarding services and things, *discovery* should be performed. As a result of *discovery* process, multiple available things list is to be built for the services. Then, next step is things *composition*. *Composition (ad hoc things collaboration space management)* includes selection and grouping of things and it should be performed based on the semantic of both services and things. Semantic of services is a specification of services which include service identifier, service location, service type, workflow, user's preference, etc. Semantic of things is a specification of things which include thing identifier, location, service type, device type, permission, etc. Next step is *scheduling* which plans execution flow of things. Things can be operated in parallel or serialized. It depends on the semantic of services. Then, next step is execution of the composed service. During execution, status of each thing should be monitored and if certain thing's status is changed then service composition should be adapted accordingly. It is *monitoring* step. Finally, when a user wants to stop services then thing ad hoc things collaboration is terminated. It is *termination* step.

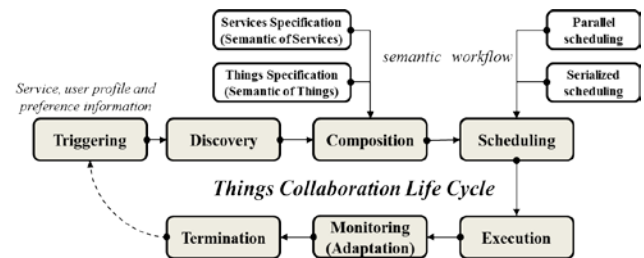


Figure 1 Ad hoc Things Collaboration Life Cycle

A. Triggering

Ad hoc things collaboration can be triggered by users' manual click or can be triggered by intelligent things which recognize needs of users.

B. Discovery

Generally, discovery can be performed in two ways. One is broadcast-based discovery, and the other is registration-based discovery. In case, where collaborating things are restricted in local domain, then both ways can be applied. But if things to collaborate with are in open domain and scale of things are varying and large, then broadcast-based approach is not reasonable from the scalability and network traffic points of view. Figure 2 shows broadcast-based discovery and collaboration.

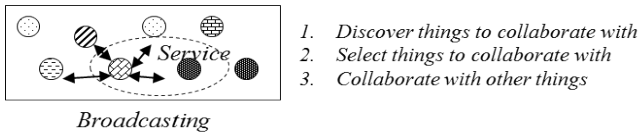


Figure 2 Broadcast-based discovery

Figure 3 shows registration-based discovery. For service provision within an open service domain by remotely located things, registration-based discovery may be a better choice but management of repository for vast number of things is also tricky problem. So, DNS-like hierarchical management policy may be applied to keep consistency and easy discovery.

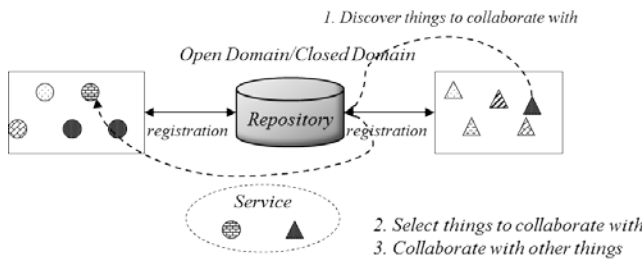


Figure 3 Registration-based discovery

C. Composition

When it comes to selection, there need criteria for selecting members. Criteria for selecting things for target services are what functionalities are needed to perform services and what kinds of things users can use. Optionally users’ preference on things can be an important factor also.

Semantic of services is service specification and it includes service identifier, service name, service type, required service type list, workflow of services, associated things list, user’s preferences, etc. Service specification gives the selection criteria such as a required service list which is provided by things and user’s preference. Meanwhile semantic of things is the thing specification and thing specification includes thing identifier, thing name, access address, resource list (or supporting service type list), device type, owner, location, etc. Thing specification gives selection criteria such as resource list, location, etc.

The definitions on things and services are very much arguable. In this paper, things are mapped into devices and devices provide multiple services. Figure 4 depicts the composition and relation between things and services.

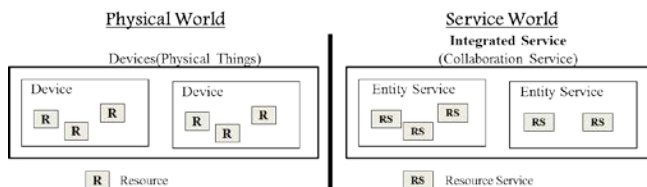


Figure 4 Things and Services

Table a shows semantic of things (thing specification) and semantic of services (service specification) based on concepts of Figure 4.

Table a Semantic of things and services

Thing Specification	Service Specification
• thing identifier	• service identifier
• thing name	• service name
• access address	• required service type list
• resource list(=supporting service type list)	• workflow
• deviceType	• associatedThings list
• owner	• user’s preference
• location	• location
• etc.	• service provider
	• etc.

D. Scheduling

Multiple things execution schedule is determined by a coordinator of the ad hoc collaboration. A coordinator can be preconfigured when things are deployed or a coordinator can be elected while operating based on a specific election algorithm. Both approaches have pros. and cons. But from a practical point of view, a preconfigured coordinator is more reasonable.

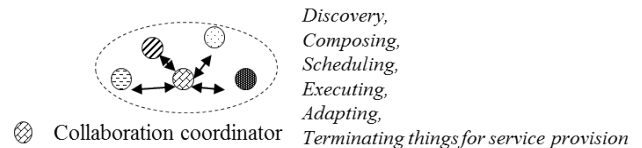
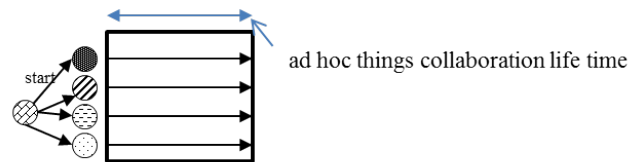


Figure 5 collaboration coordinator

E. Execution

Things collaboration can be executed in parallel or in serial. Figure 6 (a) shows a parallel execution case. Everything operates in parallel and this case can happen when a speaker, a player and a displayer operate together for movie playing.



(a) parallel execution

Figure 6 collaboration execution in parallel

In case of a serial execution, three modes are possible. First case is that a coordinator instructs each thing to start operation with an exact start time value and duration time value optionally. Then each thing operates accordingly. The second case is that a coordinator sends each task to each thing. And then a coordinator instructs to the first things for start operation. After finishing operation, the thing triggers next thing’s operation and so forth. The third case is that a coordinator transmits tasks to each thing. And then, a coordinator controls whole process. Everything should report

its service termination to the coordinator, then the coordinator determines next thing to operate, and so forth.

Figure 7 shows serial execution cases. Case (a) means a collaboration coordinator instructs to each thing to operate with start time and duration time. Then each thing operates accordingly. In case of case (a), time synchronization among things is very much crucial. Case (b) means a collaboration coordinator transmits each things task at a time. Then each thing operates based on the specified task and after doing its task, each thing gives a notification to next thing and so forth. In this case, a coordinator specifies workflow among things clearly and this information should be reflected into the things' tasks appropriately. Case (c) means each thing sends finish notification to a collaboration coordinator after finishing its task, then a collaboration coordinator instruct next things to do its task. This case is more flexible than previous case (a) and (b) from the point that when context or situation is changed during execution, a collaboration coordinator can react against the situation effectively.

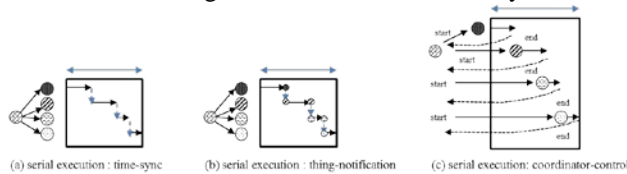


Figure 7 collaboration execution in serial

For all cases, monitoring process is very much important when physical things are concerned because of things' vulnerability.

F. Monitoring (Adaptation)

Ad hoc things collaboration space (members) is quite volatile because of things' unstable status and user's movement. Even though, services should be provided seamlessly in any situation. Therefore, monitoring on things is very important and when a thing (member of collaboration space) is out of order or away from a service area, then other candidate things should be ready for execution.

The collaboration coordinator should monitor the things' status and keep candidate things for any contingency.

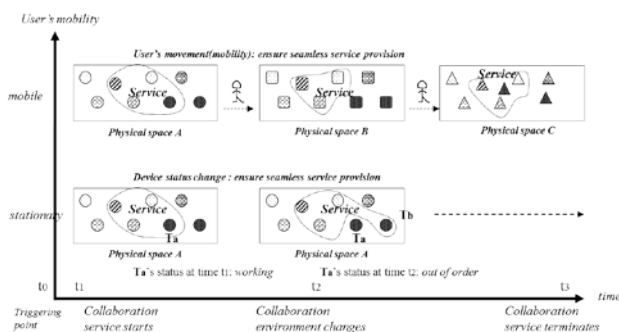


Figure 8 Ad hoc things collaboration space adaptation

G. Termination

When collaboration service ends all things become released. Then each things change each occupancy status and a collaboration coordinator stops working.

IV. CONCLUSION

This paper describes ad hoc things collaboration lifecycle management and draft version of service specification and thing specification. But many detailed parts need to be touched more because it is in an early stage of design and implementation.

More strict definition or distinction between things and services are needed and based on those things whole processes can be generalized such as SoA(Service Oriented Architecture) or RoA(Resource Oriented Architecture). Then semantic of things or services can be defined clearly enough.

Even though this is in early stage, it is obvious that context-based ad hoc things collaboration technology is a key technology to empower Internet of Things or Web of Things. Based on the connectivity given by IoT or WoT, things can be operated more human friendly and that is the goal of ICT.

ACKNOWLEDGMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korea government [13ZC1130, Development of USN/WoT Convergence Platform for Internet of Reality Service Provision].

REFERENCES

- [1] ITU-T Y.2060, "Overview of Internet of Things", 2012
- [2] The Internet of Things 2012 New Horizons, IERC, 2012
- [3] Constrained Application Protocol (CoAP) draft-ietf-core-coap-17, May 2013
- [4] <http://tools.ietf.org/wg/6lowpan>, IETF 6LoWPAN WG
- [5] D. Guinard, "A Web of Things Application Architecture", Thesis, 2011
- [6] ITU-T Y.2063, "Framework of the web of things", 2012
- [7] A. Pintus, D. Carboni and A. Piras, "Paraipmpu: a Platform for a Social Web of Things", Proc. Companion of WWW2012 - Demonstrations, pp. 401-404, April 2012
- [8] <http://www.web-of-objects.com>, Web of Objects, ITEA2 Project
- [9] ITU-T TD 149 (IoT-GSI), "Requirements and common characteristics of IoT identifier for IoT service", H.IoT-ID, Sept. 2012
- [10] ITU-T TD145(IoT-GSI), "Requirements and functional architecture for open USN service platform", F.OpenUSN, Sept. 2012
- [11] ISO/IEC 20005 (Information technology -- Sensor networks -- Services and interfaces supporting collaborative information processing in intelligent sensor networks), 2012.
- [12] W. Woerndl, R. Eigner, "Collaborative, Context-Aware Applications for Inter-Networked Cars", WETICE 2007
- [13] J. Moon, D. Kang and C. Bae, "Design of Device Sociality Database for Zero-Configured Device Interaction", MUE 2013.