

Survey Paper on Model Driven Application Development in CoT

^{#1}Rachana V. Patil, ^{#2}Ms. Dipalee Chaudhari

¹rac293@gmail.com
²dipalee.rane@gmail.com

^{#12}Department of Computer Engineering

D. Y. Patil College of Engineering , Akurdi, Pune



ABSTRACT

Cloud of Things is the integration of cloud computing and Internet of Things for the better development of the applications. To build the application using existing resources, some traditional technologies are used such as service composition and mashup. But it is difficult to develop the application using these technologies because change management is somewhat difficult and time consuming as it uses the IT component. New approaches like model driven development patterns are used to overcome challenges of traditional methods and also it provides the better change management so the redevelopment of the application is easy and for this purpose, different patterns such as role driven, data driven and process driven are used according to the requirement of the application.

Index Terms—Cloud of Things, Model Driven Patterns, IoT

ARTICLE INFO

Article History

Received: 7th January 2017

Received in revised form :

7th January 2017

Accepted: 10th January 2017

Published online :

10th January 2017

I. INTRODUCTION

Cloud computing and IoT both are emerging technologies, which are highly used in many companies for development of different application. IoT is based on smart and self-configuring nodes which are interconnected in a continuous changing and global network infrastructure but it has limited storage and processing capacity whereas in other hand cloud computing gives the platform for on demand access to various resources with greater capabilities in terms of storage and processing power. These two are complementary aspects of Internet which create CoT (Cloud of Things) where IoT can benefits from cloud's unlimited storage and processing capabilities and cloud can benefits from IoT by spreading its scope to manage real world service dynamically. Integration of these two requires smart gateway to perform complex tasks. In CoT, IoT objects are used as front end thing on the Internet and all distributed objects are used as a whole for various scenario.

Different application build using cloud platform are special IoT application in CoT. With the massive use of IoT, different devices allows to use services easily. With the large use of web based information technology, need arises to build application according to continuously changing requirement of the user. Therefore it is important to build application faster which also required to be flexible. For this

purpose developer can use already existing resources of cloud for faster development of the application.

Developing the application arises following issues like, communication between the devices is very complex, also contextual data is required for interaction between the devices to take the decisions. Due to the limited resources, services related to these application are limited. Many devices are used in IoT, therefore huge amount of data is generated from these devices. To handle such a complex heterogeneous data is huge task in developing any application because it is difficult to maintain relationship in heterogeneous data. So the main challenge in development of any application is the data integration and intelligent interaction.

To develop intelligent application, model driven service architecture is used in cloud computing. First a meta model is constructed which covers multiple business model. Based on metamodel representation, model transformation and integration of different heterogeneous data is done in three model driven development patterns. First by integrating CIM(Computational Independent model) and PIM(Platform Independent Model), a united meta model is build. This model connects the associated requirements with the components. Then three development patterns such as role

driven, data driven and process driven are used according to the need of the business application and changing mechanism is applied if any changes occurs while developing the application.

Some common approaches are used like lightweight mashup and service composition to develop the application using existing resources. Ontology based approach gives a IoT application by using service composition[2]. It integrate procedure and data to understand the management task. This approach helps in integrating several services to build application faster. On the other hand mashup[3] provides a lightweight composition for building a web based application. This method uses the existing API and small components and build a individual application which covers the heterogeneous requirement of the user. It is used for typical web-based application.

Mashup and service composition both uses the IT components rather than business logic for development purpose, therefore if any new requirement occurs then it require more time to build application using these methods. Also as components are used, change management is difficult in these methods, therefore adaptability is less. Semantic reasoning is used in these methods which is little complex, to use in any application[2][3]. Existing approaches provided a popular supporting architecture that focus on service modeling and application integration, but not a comprehensive solution to construct and execute an intelligent application. Application construction in cloud platforms requires not only data acquiring/transferring but also semantic integration of both web component resources and business elements. So both of these methods are not very useful in building application in flexible and faster manner.

II. RELATED WORK

There are many ideas and methods which are used in development of the various applications.

Simon Mayor proposed a generalized approach by using semantic integration for interface with various smart devices. In this interaction is happened not only with components but also with physical things like button and softwares[5]. To build and rapid prototyping of context aware services Tao Gu proposes the Service-Oriented Context-Aware Middleware (SOCAM) architecture[6]. It support to access and interpret different services easily. Soulimen Hasan proposes the approach for identifying semantic events for the Internet of things[7]. In this based on statistical model an abstract model is generated. All these approaches are used for generating model using contextual information. To avoid complexities of large web based applications Mohammed Mukhtar[8] proposes the use of model driven engineering to generate web application at abstract level. In this CIM belongs to business logic whereas PIM an PSM belong to components. Business model here is different from executable system components. Zhang, H. proposes the idea of using MDA so as developer focus on business logic more rather than the technical details[9]. Kang, W proposes the use of ontology in MDA for the software development for semantic disposing[10]. Martin

first gives the idea of use of EUD i.e the end user intuitive ways to modify the original application according to user requirements[11]. For this purpose UI centric approach is used and it focuses on adaptability and rapid changes. Liu, X. and Spahn, M also proposes the use of EUD, but most application failed to use of this due to situational integration and collaborations[12][13]. To develop application with incompatible services using cloud platform Martino proposes semantic services to identify automatic service discovery[14]. But as only existing services are used and not composed, it gives low flexibility. All of these approaches are used basically for system configuration in building the application.

In cloud platform, execution of services concern not only about performance[15] but also the security and adaptability[16][17]. K. Votis[18] proposed the service oriented architecture for distribution and administration of nodes. To meet continuous changing requirements C. Xie combined both structural and operational semantics for services[19]. Mapping concepts to distributed environments S. Hallstein-sena[20] proposed a model driven development framework with adaption mechanism which gives separation of business logic and contextual data. Kiev Gama[21] use SOC principles to integrate heterogeneous services to use the IoT services to realize dynamic services. By using event process-ing language(EPL)[22] context aware template methods was proposed to enhance business performance. These methods are used for service executions in cloud platforms.

All these approaches gives the supporting architecture that focus on service modeling and application integration but not gives a complete solution to build intelligent application.

III. METHODS

To build cloud enabled application using existing resources, common approaches used are lightweight mashup and service composition. But both these methods are not useful for continuous changing requirements.

To build application faster and in flexible manner while considering the different requirement such as device identification, interaction between the various devices and heterogeneous data collection as the data is gathered from various devices, framework using model driven development patterns is used. Considering these requirements, the framework is divided into three parts, information configuration for data gathering and system configuration for device identification and according to that configuration of the system and interaction configuration is used for communication between the devices which forms the runtime environment.

In information configuration stage, information of different IoT objects is represented in abstract manner. By using multi view model, a united model considering the specific requirement of the application is build. In system configuration stage, MVC(model-view-controller) is used as a reference for integration of service components. In this stage, related CIM are grouped as three types of components UI type, controller type and data type component. PIM in

web based system are represented as web pages, data connection and SOAP based services so as to connect to related components in CIM. In this stage three model driven pattern according to the requirement of the application are used. These patterns are role driven, data driven and process driven.

Role Driven : This pattern is used basically when the first requirement is role or user specific. Use case diagram is used in this pattern for development to clearly define the roles and function related to them. This pattern is mostly used in structural application.

Data Driven : This pattern is used basically when the first requirement is data specific. In the process of development, class diagrams or Entity Relationship diagrams are used for defining the entities and relationship between them. This pattern is mostly used in data-centric system.

Process Driven : This pattern is used basically when the first requirement is business process specific. In the process of development BPMN(Business Process Modelling Notation) diagrams and activity diagrams are used for defining the flow of tasks. This pattern is mostly used in work flow based system.

To build the flexible environment so as to meet the continuous changing requirements of the user, change management is used based on semantic integration of the components.

In role driven pattern, the process for using change management is, business logic is define based on functional model. Then for each role, webpage or UI is defined and using functional model SOAP services are encapsulated. Change management is applied easily in this pattern, as it is easy to reallocate the role authority which is a relationship between the service and the role. Using work flow engine, logic related to function is define and relationship between the SOAP services is define by O/R mapping if ERM is same.

In process driven pattern, the process as functional model(FM) is decomposed to define tasks and for this purpose BPMN diagram is used. Then based on relationship between the function and role, FM are used for allocation of task and web page is developed for each role. To find related data source, relationship between FM and ERM is used. SOAP services are generated based on FM's and according to data access requirements RESTful services and data connections are implements based on ERM. In this pattern change management is define as, relationship between function and SOAP services for reuse. ERM to RESTful services, transformation and data connection are easy to configure.

To build automatic model driven application, using data driven pattern, ERM is build using device abstraction and information requirements. For this purpose class diagram is used. Then based on each ERM, ERM's are combined to construct unique access module which generates RESTful services. Based on FM, generate UI and web pages, and

using authority, allocates the web pages to related users. The changing management in this pattern is define as, ERM changing is easy to dispose as services are loosely coupled with data and this ERM changing is used to update database and the user authority which is the basic key is easy to reallocate for maintaining the relationship between services and roles.

As changing mechanism is easy to apply in all these patterns, so use of this model driven patterns helps to build application in flexible manner.

IV. CONCLUSION

CoT gives the tremendous opportunities to develop the application which user can access whenever wants. As the demand of the end users changes continuously, using model driven patterns provides the flexible environment to develop application as applying changing mechanism by using relationship reasoning is easy in all role driven, data driven and process driven patterns. The patterns can be used according to specific user requirements as if the requirement is user specific then role pattern can be used, for workflow based system process driven pattern is used and for data centric application data driven pattern is used. So using these patterns, rapid development of application is achieved. This solution provides the semantic services at all different stages such as data abstraction, information configuration and intelligent interaction mechanism.

REFERENCES

- [1] ongming Cai, Yizhi Gu, Athanasios V. Vasilakos, Boyi Xu Jun Zhou,"Model-Driven Development Patterns for Mobile Services in Cloud of Things" IEEE Transactions on Cloud Computing.
- [2] N. Lasierra, et al., An autonomic ontology-based approach to manage information in home-based scenarios: From theory to practice, *Data Knowledge Engineering*, (2013), <http://dx.doi.org/10.1016/j.datak.2013.06.004>
- [3] Janner, T., Siebeck, R., Schroth, C., Hoyer, V.: Patterns for Enterprise Mashups in B2B Collaborations to Foster Lightweight Composition and End User Development. 2009 IEEE International Conference on Web Services (ICWS2009), pp. 976-983 (2009)
- [4] De S. , Barnaghi P., Bauer M., Meissner S., Service modelling for the Internet of Things., 2011 Federated Conference on Computer Science and Information Systems (FedCSIS 2011), pp. 949-955 [7]
- [5] Mayer, S., Tschofen, A., Dey, A. K., Mattern, F., User interfaces for smart things—A generative approach with semantic interaction descriptions, *ACM Transactions on Computer-Human Interaction*, 21(2), 12. (2014).
- [6] T. Gu, H.K. Pung, D.Q. Zhang, A service-oriented middleware for building context-aware services, *Journal of Network and Computer Applications*, 28 (1) (2005), pp.1-18.

- [7] Hasan, S., Curry, E., Approximate Semantic Matching of Events for the Internet of Things, *ACM Transactions on Internet Technology*, 14(1), 2. (2014)
- [8] Mukhtar, M.A.O., Hassan, M.F.B., Jaafar, J.B., Rahim, L.A.: Enhanced Approach for Developing Web Applications Using Model Driven Architecture. *International Conference on Research and Innovation in Information Systems (ICRIIS 13)*, pp. 145-150 (2013)
- [9] Zhang, H., Liu, J., Zheng, L., Wang, J.: Modeling of Web Service Development Process Based on MDA and Procedure Blueprint. *2012 IEEE/ACIS 11th International Conference on Computer and Information Science (ICIS)*, pp. 422-427 (2012)
- [10] Kang, W., Liang, Y.: A Security Ontology with MDA for Software Development. *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 68-74 (2013)
- [11] M Cinzia Cappiello, Maristella Matera, Matteo Picozzi; A UI-Centric Approach for the End-User Development of Multidevice Mashups; *ACM Transactions on the Web*, 9(3),(2015)
- [12] Liu, X. ; Ma, Y. ; Huang, G. ; Zhao, J. ; Mei, H. Data-Driven Composition for Service-Oriented Situational Application, *IEEE Transactions on Services Computing*, 2014, Doi: 10.1109/TSC.2014.2304729 (2014)
- [13] Spahn, M., Dorner, C., Wulf, V.: End User Development: Approaches towards a Flexible Software Design. *16th European Conference on Information Systems (ECIS 2008)* (2008)
- [14] Di Martino, Beniamino, Antonio Esposito, and Giuseppina Cretella. "Semantic Representation of Cloud Patterns and Services with Automated Reasoning to support Cloud Application Portability." *IEEE Transactions on Cloud Computing* DOI: 10.1109/TCC.2015.2433259 (2015).
- [15] L. Wang and S. U. Khan, "Review of Performance Metrics for Green Data Centers: A Taxonomy Study," *Journal of Supercomputing*, vol. 63, no. 3, pp. 639-656 (2013).
- [16] Z. Sheng, S. Yang, Y. Yu, Vasilakos A., McCann, J., Kin Leung. A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6), pp91-98. (2012)
- [17] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, Dechao Qiu. "Security of the Internet of Things: perspectives and challenges". *Wireless Networks*, 20(8), pp 2481-2501.(2014)
- [18] K. Votis, C. Alexakos, B. Vassiliadis and S. Likothanassis, An ontologically principled service-oriented architecture for managing distributed e-government nodes, *Journal of Network and Computer Applications*, 31 (2008), pp.131-148
- [19] Cheng Xie, Hongming Cai, Lihong Jiang: Ontology Combined Structural and Operational Semantics for Resource-Oriented Service Composition. *The Journal of Universal Computer Science* 19(13): 1963-1985 (2013)
- [20] S. Hallsteinsena, K. Geihs, N. Paspallis, F. Eliassen, G. Horn, J. Lorenzo, A. Mamelli, G.A. Papadopoulos, A development framework and methodology for self-adapting applications in ubiquitous computing environments, *The Journal of Systems and Software*, 85 (2012) , pp.2840-2859
- [21] Kiev Gama., Lionel Touseau, Didier Donsez, Combining heterogeneous service technologies for building an Internet of Things Middleware, *Computer Communications*, 35 (2012), pp.405-417
- [22] Pablo Rosales TEJADA, Jae-Yoon JUNG, Context-Aware Dynamic Event Processing Using Event Pattern Templates, *IEICE TRANS. on Information and Systems*, Vol.E96-D No.5, pp.1053-1062