Automata based Web Service Composition for Context-Aware Post-Surgery Hospital Ward

Ashok Immanuel V.
Associate Professor
Department of Computer Science
Christ University, Bangalore 560100

Pethuru Raj
Infrastructure Architect
IBM Global Cloud Center of Excellence
IBM India, Bangalore 560045

ABSTRACT

Context-awareness has become the key attribute for empowering our day-to-day systems and services to be pragmatic and distinctly powerful. Purpose-specific and generic systems, be it physical, mechanical, electrical, electronics, or computing systems are getting bestowed with enabling smartness in order to be cognitively adaptive in their actions and reactions. That is, based on the changing contexts, systems are internally as well as externally equipped to be smart in their operations, offerings and outputs. However precisely understanding the evolving contexts and feeding them into systems dynamically in time to empower them to exhibit desirable and delectable behavior are not an easy task. Thus mathematicians and computer scientists are in unison in exploring and expounding a variety of viable and venerable mechanisms and means in order to streamline and simplify the process of context-awareness.

In this paper, we have zeroed down on the unique capabilities of service automata in composing atomic and discrete actions into context information on which the designated system trusts and forges ahead in accomplishing what is initially intended. We have come out with a use case in a smarter hospital environment and proved that how service automata come handy in fulfilling the intrinsic needs of context-awareness.

Keywords

Context-aware computing, smart ward ubiquitous computing, service automata, web service composition.

1. INTRODUCTION

Next-generation systems need to have the intrinsic capability of context-awareness in order to be a part and parcel of the smarter planet envisioned. We have focused on the service automata and its mainstream feature in enabling context-awareness. We have planned this paper based on the widely used use-case approach. The approach, which eventually ends up as a reusable framework, can be tweaked in order to be highly extensible. The leverage of time-testing automata-based solution is turning out to be a strategically sound factor. We have given a brief of what is all about context-aware computing, how it is going to be a game-changing concept for the future era of knowledge systems. Every kind of personal and professional applications and digital assistants are to be made self, surroundings, and situation-aware to be greatly beneficial for humans. We explain the methods of capturing context information in the ensuing sections

2. CONTEXT AND CONTEXT AWARE COMPUTING

Ubiquitous computing or pervasive computing laid the foundation for context aware computing. It adds more meaning to the smart environments. Context awareness can be defined as interrelated and interdependent conditions of entities which exist or occur. Location awareness alone does not satisfy the highly smarter and mobile world. By applying the context awareness the user can be well served. Context aware applications can be placed into four broad categories: proximate selection, automatic contextual reconfiguration, contextual information and commands, and context-triggered actions [5]. Context aware computing can be and has been applied in various domains. Context-awareness is particularly an enabling technology in the healthcare domain and has been envisaged in “MobileWARD” and “Activity based Healthcare”. Context awareness has been used in monitoring and reporting health condition of patients suffering from brain tumor [3]. Context aware pill container and context aware hospital bed has been experimented earlier involving number of clinicians and patients based on which deployment of context-aware clinical computer applications have been suggested [9]. Design principles focused on framework, supporting deployment and development of such computer applications are largely based on non-service orient architecture involving dynamic composition of services.

3. SERVICE COMPOSITION

Services (business as well as IT) are modular software systems or applications which are loosely coupled and designed to participate in machine-to-machine interaction over a network. They are discovered, Orchestrated and coupled to form applications. Therefore urge for such web service composition has become more prominent and compelling [2][4]. Various methods have been proposed for service discovery and composition. Graph-Based approach proposes a modeling language called interface automata as the low-level construct [6]. Two categories of service composition include syntactic and semantic composition. The syntactic composition is a semi dynamic composition since it lacks the profile information, input and output state. Therefore it cannot adapt to a changing environment without human intervention. The semantic composition model registers functional information, the pre and the post conditions of services so that the domain can be logically inferred [7]. Workflow based approaches are well suited for syntactic composition while AI method is prescribed for semantic composition [8].

Several research initiatives has been attempted to integrate heterogeneous systems. Languages such as Universal Description, Discovery and Integration (UDDI) [10], Web Services Description Language (WSDL) [11], Simple Object Access Protocol (SOAP) [12] have been defined as standards for service discovery, description and composition. Business Process Execution Language for Web Service (BPEL4WS) [13] and DAML-S [14] Service Model, are focused on service compositions where bindings between services are known upfront.

4. SERVICE AUTOMATA

Service automata have been suggested to be used for service discovery and composition problem in context-aware systems. In this technique a context space comprising the context, situation, services and user preference are unified and dealt. The desired goal situation is predicted by the user preference and the situation faced by the user. A service composition is achieved by emphasizing the relationship between context and services and by formalizing semantic web services as automata. Further by using a composite service automaton, services are composed to achieve the goal situation.

Service Automata is defined a tuple consisting of context attributes (C) related to the service, a finite set of input, output and inner actions (∑), a finite set of states (Q), the transition function (δ), a set of initial states (I), a set of final states (F) and an assignment function (V) [1]. Service automata were also used in researches in enforcing security in distributed systems [15]. The thrust of service automata lies in the Context Space Model (CSP).

5. USE CASE

The motivation for this paper lies in the Intensive Therapy Unit (ITU) which is the recovery ward for most critically ill and post-surgery recovering patients. Typically heart surgery hospitals will have on an average of 30 to 35 bed ITUs. Each day there could be nearly 20 heart procedures. Each ITU is manned by 20 nurses.

Immediately after a heart surgery, the patient is very critical during the first 48hours. The patient has to be closely monitored and numerous vital readings of body condition have to be logged. During the surgery, mediastinal tube (drain tube) is placed in the pericardial space which serves the purpose of draining post-operative bleeding and pericardial effusion. Another drain tube known as the pleural tube is also place during the operation to drain blood and pleural fluid. Monitoring the pleural tube is very crucial. An excessive collection of blood indicates that a blood vessel would have not been properly cauterized or it might indicate a raptured graft. In both cases, if unchecked, the patient may go into coma and eventually die.

During the first hour after surgery a maximum of 150ml would have drained, if it is in excess of 150ml then the drain is watched for one more hour. If the drain continues to see an increase of more than 150ml then the surgeon is alerted on the situation. Meanwhile blood samples are drawn and fed into Arterial Blood Gas (ABG) analyzer machine. The ABG primarily measures the Potassium, Carbon Dioxide, Bicarbonate, arterial oxygen tension and oxyhemoglobin saturation levels. If the hemoglobin level is below 9g/dl and the hematocrit levels fall below 30% then blood transfusion has to be done for the patient. In order to do a blood transfusion, the blood sample is again drawn and sent to the blood bank for cross matching and checking for availability. Once the doctor arrives at the ward, the x-ray which was taken just after the surgery is studied for fluid collection. If the doctor requires an Echocardiogram is also taken for analysis. Depending on the condition the doctor decides either for a blood transfusion or re-operation for the patient.

6. Automata for the proposed webservice

In the above mentioned ITU environment, by monitoring the various parameters we can predict the context and enable service composition to achieve the desired situation. In the smart ward the situation should be predicted based on the current context and a plan has to be devised to achieve the goal situation using the related services. Depending on the context which is determined by the various levels which are measured by the devices we can arrange to achieve the goal situation by composing services. Two goal situations are defined for the above scenario: si_normal and si_raise_alert. The rule for this service is defined as:

IF increase_in_weight

THEN si_raise_alert OR si_normal

6.1 Automata for CHK_WEIGHT service

Fig 1 represents the automata for the CHK_WEIGHT web service. This service checks the weight of the drain by reading the sensor which is attached to the drain. If the weight is greater than 150ml then an alert with level=low is set. If the weight is greater than 300ml then an alert with level=high is set. The final state of the automat is shown with a concentric circle.

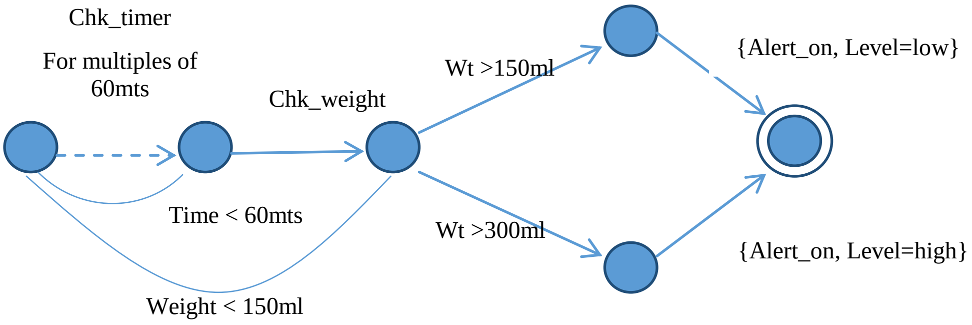

Fig 1: Automata for CHK_WEIGHT service

Table 1. State Transition Table for CHK_WEIGHT service

State	Input →State	Input→ State	Input→ State
1	Chk_timer → 2		
2	Time < 60mts → 1	Time > 60mts → 3	
3	Chk_weight < 150ml → 1	Chk_weight>150ml → 4	Chk_weight>300ml → 5
4	Weight>150ml→6, Level = low		
5	Weight>300ml→6, Level =high		

6.2 Automata for ALERT_ON service

Fig 2 represents the automata for the CHK_WEIGHT web service. The ALERT_ON service operates on two modes. The service expects an input parameter of level. The level attributes can take two values namely “Low” and “High”.

Whenever the alert_on service is invoked, it first checks if the flash light which is on the monitor by the patient bed side is on. For a level=low state, the flash light is on. If the state is ‘off’, then it is pushed to the ‘on’ state. Then the service checks if the level is set to high. If so then a sound alarm is also set on indicating an emergency situation.

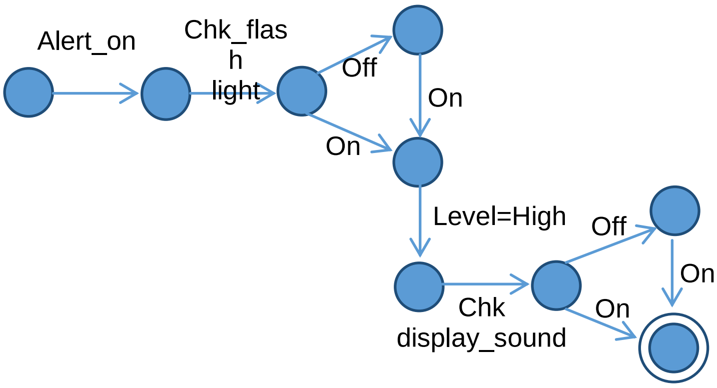

Fig 2: Automata for ALERT_ON service

Table 2. State Transition Table for ALERT_ON service

State	Input →State	Input→ State
1	Alert_on → 2	
2	Chk_flash_light → 3	
3	Off →4	On → 5
4	On → 5	
5	Level-High→ 6	
6	Chk_display sound → 7	
7	Off → 8	On→ 9
8	On → 9	

7. COMPOSITE SERVICE

Automatic composition of service highly rests on the definition of the pre and post conditions of the service. As we reach the final state of a service, the post condition of that service is matched with the matching pre condition of available services. Based on the defined rules, the algorithm has to determine the possible the most probable situation and decide on the service which would eventually enable to achieve the goal situation. In this case, the chk_weight service, the final state is alert_on. Based on the rule define under the use case, the probable service would be alert_on as its pre-condition matches the post condition of the earlier state. Fig 3 represents the automata for a composed web service. Services are interoperable, composable, discoverable, replaceable and sharable. All kinds of enterprise-scale applications are being built on verified and validated services. Services emerge as the powerful building-block for business-critical applications. Services, besides bringing in a number of unique innovations, enable modular and sustainable applications. Service-oriented architecture (SOA) is being projected as the most sought-after architectural pattern for the sustainability of enterprise-class IT.

Microservices architecture (MSA) is a direct offshoot of service orientation paradigm. With services are becoming pervasive and persuasive across the industry verticals such as healthcare, retail, energy, utility, finance, government and logistics, building inventive and people-centric applications out of services is being touted as the best way forward.

Service composition is an enabling factor in realizing high-end process-aware applications. Sophisticated applications can be built through the seamless and smart orchestration of a host of discrete, atomic and basic services dynamically as per the changing needs. There are several mechanisms being proposed for strengthening the aspect of service composition.

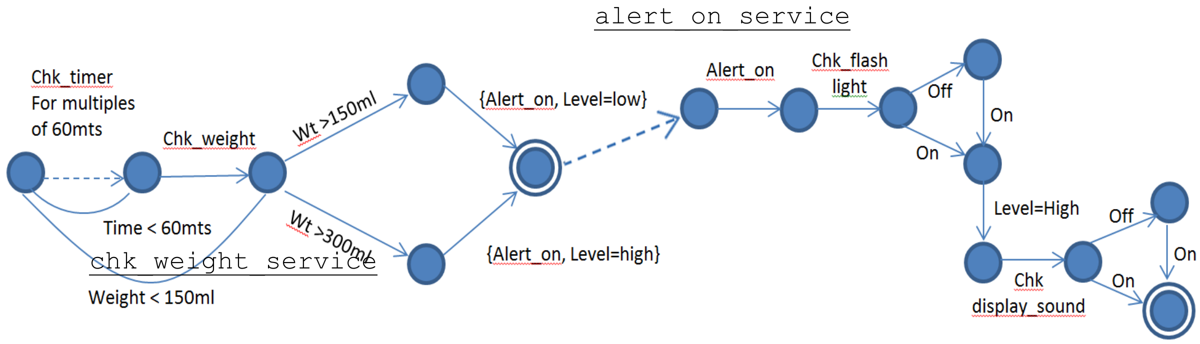

Fig 3: Automata for composed service

We have explored the utility and usability of service automata in order to simplify the process of service composition. With this kind of automaton, the composition factor can be speeded up and new services and applications can be quickly realized in order to sustain the software engineering journey. We are planning to come out with a comprehensive and compact framework by developing the service automata using any one of the popular programming languages. This framework implementation clearly lessens the workload of software engineers and developers. We are also exploring the possibility of leveraging timed automata in order to bring in more deeper and decisive automation into the service-oriented software engineering domain. The planned framework is to utilize the proven and potential state machine concepts fully towards the future of software development especially for the forthcoming connected and cognitive software applications.

The above composite automaton diagram combines two different and discrete services into one unified service depicting the bigger picture. Automaton, being the formal construct, realizes the composition within all the constraints. There are specific languages emerging in order to formally represent the data and control flows in automata in order to persist, interchange and model service automata.

8. CONCLUSION

Service automata are an excellent mechanism for composing multiple events, activities, and information to arrive at context information that can be used by systems to be adaptive in their executions. The people-centricity and service-propensity of future applications are being empowered by context-awareness, which is being positioned as the prime point. We have demonstrated how service automata come to our rescue in accurately realizing context information programmatically. The future steps include leveraging timed automata in order to come out with a competent framework for context-awareness for any kind of application domains.

9. ACKNOWLEDGMENTS

Our thanks to Namratha M, Assistant Professor, Department of Computer Science & Engineering, B.M.S. College of Engineering who has contributed towards developing the automata diagram and state transition tables.

10. REFERENCES

[1] Z. Zhang, W. Li, Z. Wu, and W. Tan, “Towards an Automata-Based Semantic Web Services Composition Method in Context-Aware Environment,” 2012 IEEE Ninth Int. Conf. Serv. Comput. (SCC 2012), pp. 320–327, 2012.

[2] Rostami, Narges Hesami; Kheirkhah, Esmaeil; Jalali, Mehrdad, "Web services composition methods and techniques: A review," International Journal of Computer Science, Engineering & Information; Dec2013, Vol. 3 Issue 6, p15

[3] Bhattacharyya, Shankari. "Context aware health care application." International Journal of Advancements in Technology 2.3 (2011): 461-470.

[4] R. Hull and J. Su, “Tools for composite web services: a short overview,” ACM SIGMOD Rec., 2005.

[5] Schilit, B.; Adams, N.; Want, R., "Context-Aware Computing Applications," Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on , vol., no., pp.85,90, 8-9 Dec. 1994

[6] S. V. Hashemian and F. Mavaddat, “A graph-based approach to Web services composition,” 2005 Symp. Appl. Internet, pp. 3–9, 2005.

[7] Xinlin Zhang; Huaikou Miao; Hongwei Zeng, "The Syntactic and Semantic Model of Web Services Composition Based Category," Advanced Computer Theory and Engineering, 2008. ICACTE '08. International Conference on , vol., no., pp.444,449, 20-22 Dec. 2008

[8] J. Rao and X. Su, “A Survey of Automated Web Service Composition Methods.” In Proceedings of the First international conference on Semantic Web Services and Web Process Composition (SWSWPC'04), Jorge Cardoso and Amit Sheth (Eds.). Springer-Verlag, Berlin, Heidelberg, 43-54.

[9] J. E. Bardram, “Applications of context-aware computing in hospital work: examples and design principles,” SAC ’04 Proc. 2004 ACM Symp. Appl. Comput., pp. 1574–1579, 2004.

[10] T. Bellwood et al. Universal Description, Discovery and Integration specification (UDDI) 3.0. Online: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.

[11] R. Chinnici et al. Web Services Description Language (WSDL) 1.2. Online: http://www.w3.org/TR/wsdl/.

[12] D. Box et al. Simple Object Access Protocol (SOAP) 1.1. Online: http://www.w3.org/TR/SOAP/, 2001.

[13] T. Andrews et al. Business Process Execution Language for Web Services (BPEL4WS) 1.1.Online: http://www-106.ibm.com/developerworks/webservices/library/ws-bpel, May 2003.

[14] D. Martin et al. DAML-S (and OWL-S) 0.9 draft release. Online: http://www.daml.org/services/daml-s/0.9/, May 2003.

[15] Richard Gay, Heiko Mantel, and Barbara Sprick. 2011. Service automata. In Proceedings of the 8th international conference on Formal Aspects of Security and Trust (FAST'11), Gilles Barthe, Anupam Datta, and Sandro Etalle (Eds.). Springer-Verlag, Berlin, Heidelberg, 148-163. DOI=10.1007/978-3-642-29420-4_10 http://dx.doi.org/10.1007/978-3-642-29420-4_10.