Journal of Theoretical and Applied Information Technology
15th December 2011. Vol. 34 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTONOMIC SERVICE COMPOSITION THROUGH CONTEXT ORIENTATION APPROACH

JUNAID AHSENALI CHAUDHRY, SAQIB ALI · MD ASRI BINNGADI, ABDUL HANAN ABDULLAH

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
Tel.: +60-12-6479943
Fax: +6-07-5565044
E-mail: junaid@utm.my, asaqib2@live.utm.my, dr.asri@utm.my , hanan@utm.my

ABSTRACT

In this paper we present a novel service composition scheme for ubiquitous environments. As in a ubiquitous system there are many heterogeneous devices connected to each other in various topologies and one type of application that gives optimal performance on one node can perform differently from the other nodes. Our scheme addresses this issue through a dynamic service composition scheme that considers constrains on a local node and generates an application with near optimal performance with respect to its context. We simulate the scheme in our project environment consisting of hybrid mesh. The overall objective is to propose an efficient mechanism to meet the optimization demands at software level.

Keywords: *Autonomic Computing, Service Composition, Component based development, Autonomic Control Systems, Ubiquitous Systems.*

1. INTRODUCTION

Despite network challenges and limitations of infrastructure, the interest in omnipresent and the technology which is always available is growing [1], [2], [3]. It is expected to grow exponentially in times to come. Building on top of the ubiquitous infrastructure i.e. OSGi ® [4], the Service Oriented Platform (SOP) is well thought-out to be strong candidate for prospect ubiquitous networks [5].

Although, the research findings in Peer to Peer (P2P) technology are convincing [15], but we need to have some hybrid approach at infrastructure level to meet the functional requirements. Especially when the system is in direct interaction with the user meeting user requirements becomes a highly sophisticated issue. Services are considered building blocks of ubiquitous applications [16]. All types of such services that reside on a platform can combine to make applications that directly interact with the users [6]. In presence of a high scale of heterogeneity and application customization, the multi-faced application development dynamically is very difficult [7]. So it is vital to expand the control and build up context-oriented applications at various levels. Moreover since, two adjacent nodes can have totally different contexts, the probability that one application may not work at two nodes becomes quite high.

We present an autonomic service composition scheme in smart ubiquitous environments. Where, devices exchange context for completion of some common goal in a cluster. Each cluster shares almost the same context. It is highly likely that two clusters having different contexts or within a cluster two or more groups with drastically different contexts. In these environments, the importance of a context-oriented application becomes very high. We propose that instead of pipelining with an external application or even handoff procedure, each cluster develops its own application locally and manage them as well. Simulation results show a promising picture and considerable improvement. At the end this paper we present our proposed scheme with the help of an example.

The remainder of this paper is structured as follow. In section 2 we discuss the background and the system architecture in section 3. Our

Journal of Theoretical and Applied Information Technology
15th December 2011. Vol. 34 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

service composition scheme is discussed in section 4. Chapter 5 contains the analysis and in chapter 6 we present the simulation results. We end up with the conclusion.

2. OVERVIEW AND SYSTEM ARCHITECTURE

In Fig. 1 system architecture is shown. On top there is a Global Management Server (GMS) which contains various entities i.e. Context manager, Global policy manger, and management repository. Some clusters are connected to GMS. Every Cluster Head (CH) has Domain Policy Manager (DPM) and Mobile Code Execution Environment (MCEE).

At every level nodes have specific functions (shown in the boxes on right). These functions can also duplicate. In the case of duplications the parameters in those functions would be totally different from each other e.g. if a CH's bandwidth is optimized if its 20MBps then the bandwidth of a leaf node can't expected to be that high.

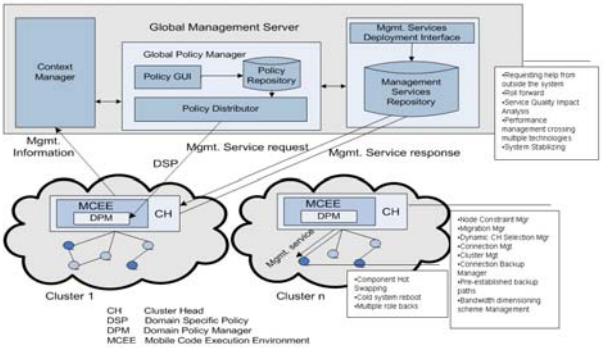

Fig. 1: System Architecture

The whole system is managed through management policies and some functional policies.

3. THE PROPOSED SCHEME

In this section, we present the architecture, component, their functionality and the algorithms proposed. Figure.2. shows the service architecture of our scheme.

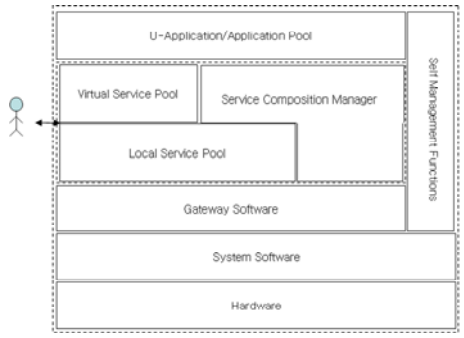

Fig. 2. The Proposed Scheme Layered Architecture

In figure 2, after system software the gateway software is the embedded software. In our case, we are using Prosyst's mPower Remote Manager®. There are two different types of service pools.1) Local service Pool and 2) virtual service pool. Those services which are not present locally reside in Virtual Service Pool.

Let S= service, which can have

$$S_{er} = \sum_{n=1}^{n=\infty} nF(1.\sum_{m=0}^{m=n-1} mF)$$

Where l = limit of the process memory or process global Area (PGA). And i is the instance of the service. Now in a service pool, there can be many services.

$$S_{pool} = \sum_{p=0}^{p=\infty} pS_{er} \qquad (1)$$

Now these services can be dependent,

$$S_{pool} = \sum_{q=0}^{q=\infty} q\left(\frac{2d}{q_d}\right) S_{er},$$

$$S_{pool} = \sum_{q=0}^{q=\infty} q\left(\frac{2d}{qd}\left(\sum_{i=1}^{i=\infty} iF(1.\sum_{m=0}^{m=i-1} mF\right)\right)$$

(2)

Or the services can be independent

$$S_{pool} = \sum_{p=0}^{p=\infty} pS_{er} \qquad (3)$$

Now combining 1 and 2,

$$S_{pool} = \sum_{p=0}^{p=\infty} pS_{er} = \sum_{q=0}^{q=\infty} q\left(\frac{2d}{qd}\left(\sum_{i=1}^{i=\infty} iF(1.\sum_{m=0}^{m=i-1} mF\right)\right)$$

.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

There are some self management functions running in parallel that makes the system fault tolerant. The Service Composition Manager (SCM) is the entity that generates Applications. The details of SCM are as follows.

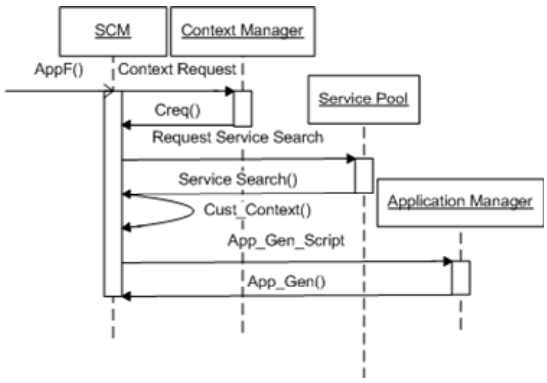

Fig. 3. Sequence Diagram of the process

Service Composition Manager (SCM): The Service composition manager consists of the following parts.

1- Service Searching
2- Instantiation and Binding
3- Context-orientation

Since

AF = (Service Searching)+(Service Instantiation and Binding)+ (Testing and Confidence Factor Calculation)

Let us discuss now Service Searching.

Service Searching: Dynamic selection of suitable services involve the matching of service requirements with service compatibility rather then simple search keywords. Universal Description, Discovery and Integration (UDDI) is an XML based registry for business services worldwide to list services in the internet. In [External Matching in UDDI] a new design for providing an external matching feature in matching services within UDDI is proposed. It allows multiple external matching services developed by independent service providers to be integrated with in a UDDI registry.

Normally the services have two types of similarity 1)Syntactic similarity; in this type of similarity only the service name and its description is checked. 2) Functional Similarity; in this similarity the actually functionality is checked. We use the following algorithm for searching in UDDI for our scenario.

$$MS(ST,CS)=\frac{w1*SyntacticSim(ST,CS)+w2*FunctionalitySim(ST,CS)}{w1+w2}\in[0..1]$$

Where w1, w2 are weights and ST is the Service Template, described in the Application File and CS is the Candidate Service. In the following text we will describe the SyntacticSim and FunctionalitySim functions.

SyntacticSim(): This function compares the names and description of the service templates and candidate service. The result of this function is 1 or 0. The details of the function are represented through the following equation.

$$SyntacticSim(ST,CS)=\begin{bmatrix} w3*NameMatch(ST,CS) \\ +\frac{w4*DescrMatch(ST,CS)}{w3+w4}\in[0..1]Descr[ST]\neq\phi or Descr[CS]\neq\phi \\ NameMatch(ST,CS)\,Descr[ST]=\phi, Descr[CS]=\phi \end{bmatrix}$$

Name similarity can be calculated through string matching algorithms and description similarity can be calculated through n-gram algorithm.

FunctionalSim(): It is not only the semantic similarity of the service but it should consider the input and output matching of CS and ST. The following formula describes the functioning of Functionalism().

$$FS=\frac{\sum_{i=0}^{n} fs_i}{n}\in[0..1]$$

Where

fs_i= best functional similarity of an operation of ST
n= number of operation of ST

$$getFM(OPst,OPcs)=best(getfm(OPst,OPics))$$

Where, OPi CS represents individual operation of CS and getfm()'s specifications are in the following formula.

$$fs=getfm(OPst,OPcs)=\frac{w5*SysSim(OPst,OPcs)+w6*ConceptSim(OPst,OPcs)+w7*IOSim(OPst,OPcs)}{w5+w6+w7}$$

Journal of Theoretical and Applied Information Technology
15th December 2011. Vol. 34 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Where SynSim() is the similarity of names and descriptions, ConceptSim() is the ontological similarity. The similarity of the input and output operations are not calculated in this work as we intend to do this task in future work. To connect external services in the Virtual Service Pool, the capabilities of UDDI can also be used.

Instantiation and Binding: The following is one of the binding tables provided by the service provider. It shows the current service in execution, service in pipeline, their time slots and parameters to pass them. There are also some instructions attached by service provider which is passed on to the executing services. The time slots synchronize with the master clock. In this process the sequence and starting time of instructions are important.

Table .1. Binding Table

	SID*	TSID`	Time Slot	Parameters	Instruction
1	C001455D	C001456D	001589GHJ	$t_1,s_1,w[w_1,w_{n-1}]$	DGHJ003
2	C001456D	C001457D	001589GHJ	$T_1,T_2,s_2,$	DGHJ013
3	C001457D	C001458D	001589GHJ	$T_2,T_3,S_3,$	DGHJ005
4	C001458D	C001459D	001589GHJ	$T_3,T_4,S_4,$	DGHJ001
5	C001459D	C001455D	001589GHJ	$T_4.T_5,S_5$	DGHJ007
6	C001455D	C001456D	001590GHJ	T_5,T_6,S_6	DGHJ004
7	C001456D	C001457D	001590GHJ	T_6,T_7,S_7	DGHJ014
8	C001457D	To Display	001590GHJ	T_7,R	DGHJ006

*SID= Service ID, `TSID= Target Service ID, w=weights

Context Orientation: In [CONTEXT AWARE SERVERS] the idea of context-aware servers is proposed. Taking that idea further, we propose the context orientation of services. In the application file published by the service provider, there are some context orientation functions that optimize the service according to the device context. As shown in Figure 3, the generated application is tuned according to the device context and then become ready for use. The application can also be tuned according to the user needs.

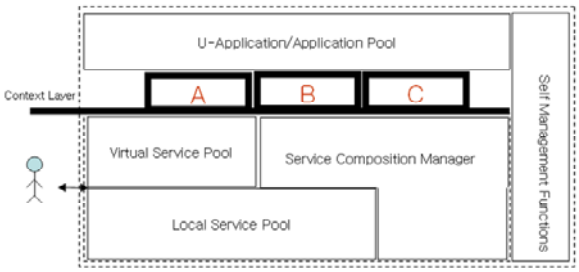

Fig.4. Context Orientation

There are three type of Context Tuning (CT) needed at this stage.

1- Recourses Context Tuning: The resources available to the application are specified.
2- Constraints Context Tuning: The known constrains i.e. average processing limit, bandwidth limit etc are specified to the application.
3- Guest Context Tuning: If some guest context is present within a cluster in the shape of some visiting node.

The application context is updated constantly so that the new constrains or recourse availability are specified to the application.

4. ANALYSIS

As mentioned above, a service S_{er} can be

$$Ser = \sum_{j=1}^{n} ib + ob(j)\{m\sum_{i=0}^{n} i\}$$

,

or

$$Ser = \sum_{j=1}^{n} ib + ob(j)\{m\sum_{i=0}^{n} i(k\text{conditi on}l)\}$$

,

Where, a service can be defined as an instruction/collection of instructions (i) with a specific number of iteration, in bounds (i_b), out bounds (o_b) with conditions (m) and links.

Now if we add those services in the service pool. From (2) we can say,

$$Spool = \sum_{m=0}^{m=\infty} m(\frac{2d}{md}(\sum_{i=1}^{n=\infty} nS(\lim_{\delta x \to 0}(i \to l))$$

.

Journal of Theoretical and Applied Information Technology
15th December 2011. Vol. 34 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Now after the generation of Application File (AF), the service selection/searching, binding and service orientation process starts. We can say,

$$MS(ST,CS)=\frac{w1*SyntacticSim(ST,CS)+w2*FunctionalitySim(ST,CS)}{w1+w2}\in[0..1]$$

,

$$SyntacticSim(ST,CS)=\begin{bmatrix} w3*NameMatch(ST,CS) \\ +\frac{w4*DescrMatch(ST,CS)}{w3+w4}\in[0..1]Descr[ST]\neq\phi or Descr[CS]\neq\phi \\ NameMatch(ST,CS)\ Descr[ST]=\phi, Descr[CS]=\phi \end{bmatrix}$$

,

For syntactic similarity and

$$FS=\frac{\sum_{i=0}^{n} fsi}{n}\in[0..1]$$

,

$$getFM(OPsr,OPcs)=best(getfm(OPsr,OPics))$$

,

and,

$$fs=getfm(OPsr,OPcs)=\frac{w5*SysSim(OPsr,OPcs)+w6*ConceptSim(OPsr,OPcs)+w7*IOSim(OPsr,OPcs)}{w5+w6+w7}$$

.

Now for instantiation and binding we can say,

$$Xpi=\sum_{m=1}^{\infty}\left(m\begin{pmatrix}p_{11} & p_{12}\\ p_{21} & p_{22}\end{pmatrix}*\begin{pmatrix}i_{11} & i_{12}\\ i_{21} & i_{22}\end{pmatrix}\right),$$

$$SBind=Xpi*\left[\sum_{k=0}^{k=\infty}\left(\left(\sum_{j=1}^{j=\infty}j.SerID\right)k.SerTSID\right)\begin{pmatrix}t_{11} & \cdots & t_{1n}\\ \vdots & \ddots & \vdots\\ t_{m1} & \cdots & t_{mn}\end{pmatrix}\right]$$

.

For the service orientation process we can formulate the following algorithm. In figure 5 the context of all the services present in the service pool are present in the context server. An application file is transferred to SCM and SCM requests the service searching and the selected services context and later it optimizes resources, satisfies constraints and other parametric changes. After context orientation, the application generation script is transferred to the application generator which generates the final application.

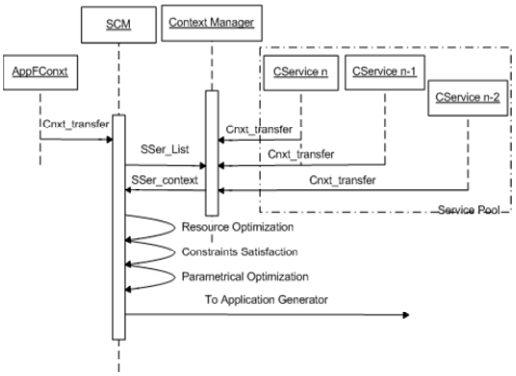

Fig. 5. Service Orientation Process

5. Simulation Results

The following figure shows the simulation results obtained after experimenting with three applications retrieving services from the local service pool. They show more or less linear trends on time scale with some jitter. We observe that for the first application, the jitter starts from somewhere at 390 sec point and the life time of the first application end at about 700sec. Since there is not much scheduling among the applications accessing services involved so not much variance is found.

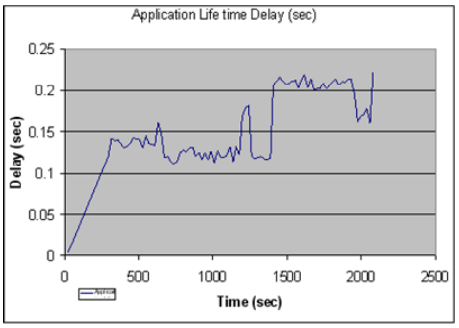

Fig. 6. Application generation from LSP.

In results shown in figure 7, the throughput behavior of different applications over time analysis is done. The applications under study have diverse throughout demands and whenever the application behavior exceeds the normal range, the proposed algorithm takes action and the throughout is stabilized. The applications are bind virtually through TCP sockets for the services that are residing with unique IP

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

addresses over distributed servers. When the service number 1 has finished binding with remote services, it gives way to application 2 (since application generation requests are placed in a queue). After remote binding and processing the application 3 begins processing. The SCM serves as a service gateway in the middle of the client application and the services present on the different servers. The results of this scenario are shown in figure 8.

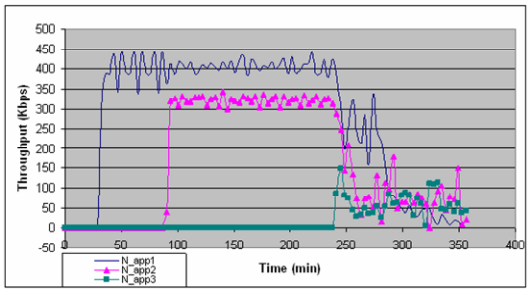

Fig. 7. Distributed Application Binding on time scale using VSP.

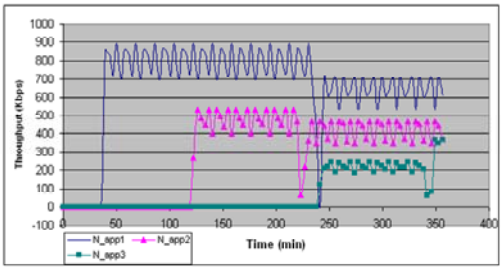

Fig. 8. Distributed Application Binding on time scale using VSP (Higher throughput).

Figure 9 shows the comparison between UDP and TCP. We observe that there is not much difference in the two approaches rather TCP performs better in regards to time.

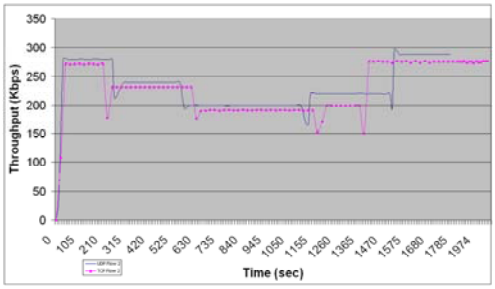

Fig. 9. Different Connection Schemes

6. CONCLUSION

In this paper we propose a new service composition scheme. An application file is generated by the service providers and that application file invokes the application generation process. The Service Composition Manager (SCM) works as a virtual bridge between the client services trying to access the remote services and it manages the application sequence.

The objective of this approach is to make applications more use oriented and in order to do that we have used service context. Later in our research we aim to enable the client generate its own application file that may assist SCM located on the gateway to provide the more customized application. It will not only reduce the service generation time but will bring accuracy too.

REFERENCES

[1] Junaid Ahsen Ali Chowdary, Won-Sik Yoon, Jai-Hoon Kim, and We-Duke Cho, "U-Kitchen: Application Scenario." *2nd IEEE Workshop on Software Technologies for Embedded and Ubiquitous computing System(WSTFEUS)*, pp. 169-171., May, 2004.

[2] D. W. McDonald, "Uiquitous recommendation systems," *Computer*. 36 (10): 111-112.

[3] S. R. Hedberg, "After desktop computing: a progress report on smart environments research," IEEE Intelligent Systems, pp. 7-9, 2000.

[4] Open service Gateway Initiative, http://www.osgi.org.

[5] M. Takemoto, H. Sunaga, K. Tanaka, H. Matsumura, E. Shinohara, "The ubiquitous service-oriented network (USON) – an approach for a ubiquitous world based on P2P technology," *Proc. P2P 2002*, pp. 17-21, 2002.

[6] M. Yu, A. Taleb Bendiab, D. Reilly, W. Omar, "Ubiquitous service interoperation through polyarchical middleware," *Proc. IEEE/WIC' 03*, pp. 662-665, 2003.

[7] A. Ricci, A. Omicini, "Supporting coordination in open computational systems with TuCSon," *Proc. WET ICE* 2003, pp. 365-370, 2003.

[8] Stan Moyer, Dave Marples, Simon Tsang, "A Protocol for Wide-Area Secure Network,"

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IEEE Communication Magazine for Comm. Appliances, pp. 52-59. 2001.

[9] P. Lucibello, "A learning algorithm for improved hybrid force control of robot arms," IEEE Transcation on Systems, Man and Cybernetics, pp. 241-244, 1998.

[10] D. Bansal, J. Q. Bao, W. C. Lee, "QoS-enabled residential gateway architecture," *IEEE Communications Magazine 2003,* pp. 83-89.

[11] A. Rantzer and M. Johansson, "Piecewise linear quadratic optimal control, *IEEE Transaction on Automatic Control 2000,* pp. 629–637.

[12] Orna Raz, Philip Koopman, and Mary Shaw; "Semantic Anomaly Detection in Online Data Sources," *24th International Conference on Software Engineering (ICSE'02),*2002.

[13] Abdelnaser Mohammad Adas and Amarnath Mukherjee, "Providing Heterogeneous Quality of Service Bounds for Correlated Video Traffic at a Multiplexor," *Perform. Eval.* pp. 45 Volume 1, 21-24 Sept. 2003-65, 1999.

[14] www.ibm.com/meshnetworks

[15] Yamada, M.; Ono, R.; Kikuma, K.; Sunaga, H., A study on P2P platforms for rapid application development, *Communications, 2003. APCC 2003. The 9th Asia-Pacific Conference on* pp. 368 - 372 , 2003.

[16] Junaid Ahsenali Chowdary, Seung-Kyu Park and Suk-Kyo Hong, On Recipe Based Service Composition in Ubiquitous Smart Spaces, *Journal of Computer Science* 2(1): 86-91, 2006.