DILT: A Hybrid Model for Dynamic Composition and Execution of Heterogeneous Web Services

Kanchana Rajaram[1], Chitra Babu[1], and Akshaya Ganesan[2]

[1] Department of Computer Science and Engineering, SSN College of Engineering,
Anna University, Chennai - 603110, Tamil Nadu, India
{rkanch,chitra}@ssn.edu.in

[2] St. Joseph's College of Engineering, Anna University, Chennai, Tamil Nadu, India
akshaya1988@gmail.com

Abstract. Business applications in domains such as e-Governance, require collaboration among both Governmental and non Governmental departments, which raises the need for composing SOAP-based as well as RESTful services. Existing works address this objective using static composition alone. However, it would be beneficial if users can specify the requirements during run-time, based on the outcome of the previous services executed. In general, business applications follow a predefined order and consequently the composition process can follow a template of business activities. Existing works on dynamic web service composition either separate the composition and execution phases distinctly or perform them in an interleaved fashion. The former approach cannot adapt to changes in run-time whereas the latter can select services based on the outcome of previous service executions. However, the interleaved approach does not support business activities that have a specific ordering among them. Hence, a novel hybrid model - Dynamic InterLeaved Template (DILT) - that enables interleaved composition and execution of web services based on predefined workflow templates has been proposed in this paper. This hybrid model lends itself naturally for composing both SOAP-based and RESTful services.

Keywords: Web Services, Dynamic composition, Service Oriented Architecture, SOAP services, RESTful Services.

1 Introduction

Applications in e-Governance are increasingly becoming widespread which require handling transactions across the various governmental as well as private departments. Whenever a single service is not sufficient for implementing a business process, multiple services need to be suitably composed. The web services are heterogeneous in nature and the two prominent types are SOAP-based and RESTful services. In general, SOAP-based web services are predominantly used in business applications where multiple competing providers exist for a single

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 239–244, 2015.

functionality. RESTful web services are preferred for Governmental department web services since, RESTful services are light-weight and less prone to security attacks.

Existing works on web service composition provide solutions for static composition of SOAP-based web services using Existing language standard for composition - Business Process Execution Language (BPEL), static composition of RESTful services [1], and semi automated solution for both types. In e-Governance business applications, the component services of a composition need input values from the end-user. Sometimes, the end-user might want to decide the service parameter values based on the outcome of the previous service execution. Several authors have surveyed and compared different approaches for web service composition [2]. Interleaved approach [3] supporting dynamic composition interleaves composition and execution of component services of a composite service. However, in enterprise applications, the business is conducted in a predetermined manner and hence the template based composition [3] is more suitable wherein the control flow among the business activities is captured. Existing works on composition of SOAP and RESTful services together [4,5], follow either static composition or severely restrict the interaction between heterogeneous services. The existing dynamic composition frameworks [6,7,8] do not allow the interleaving of composition and execution steps. Existing works on interleaved composition and execution [9,10] do not follow a specific ordering among business activities. Hence, *a novel* ***D****ynamic* ***I****nter****L****eaved* ***T****emplate model (DILT) is proposed in this paper, that dynamically interleaves composition and execution steps associated with component services of a composite business process, based on predefined control flow.* This hybrid model lends itself naturally for composing heterogeneous services.

2 Existing Work

Peng et al., proposed REST2SOAP [4] framework that integrates RESTful web services with SOAP-based services by converting them into SOAP services. In this work, the advantages of RESTful services such as lightweight nature and security are lost. Another hybrid orchestration approach of He et al. [5] is effective only when the interactions among different types of services are minimal. The MDCHeS [11] approach used BPEL engine to compose SOAP-based services, JOpera visual composition language to compose RESTful services, and BPEL for REST extension for composing heterogeneous services. All these works are semi-automatic approaches and do not address dynamic composition. An approach proposed by De Giorgio et al. [12], supported alternate service selection at runtime from a pre-discovered set of SOAP-based and RESTful services which is not truely dynamic.

A-DynamiCoS [13] framework supports user-centric interleaved service composition. However, it is not suitable for business applications where the activities follow a specific order. In Ardagna et al.'s [14] adaptive composition approach, web service selection as well as optimization and web service execution are interleaved. However, selection of each service is not interleaved with its execution, which makes specification of user requirements based on the outcome of the previous service execution, impossible. Ozorhan et al., [15] proposed an abductive event calculus framework that can be used to formalize the interleaved and template-based approach [3] separately. This framework does not address combining interleaved and template based approaches. All these three works do not address composition of heterogeneous web services.

3 Proposed Hybrid Model for Web Service Composition and Execution

The proposed hybrid model, DILT combines the features of interleaved and template based approaches for composing and executing web services. Figure 1 depicts the approach adapted by DILT model. This model involves three steps namely, dynamic selection, binding, and execution of services. The web services are selected from the respective service registry based on their type at run-time by matching the functional requirements specified by the user and the capabilities of the services advertised in the registry. Since, this paper does not focus on semantic discovery of services, it is assumed that the activity as well as the corresponding services are named similarly. The services from different service providers corresponding to a given activity are named as activity name followed by the name os the service provider. When several service providers deploy services for the same functionality, service selection involves comparing their non-functional capabilities which is outside the scope of this paper. The three steps involved in the composition are interleaved for each activity present in the template. Hence, the response of the previous service execution is useful for deciding the selection of services from the registry, for a subsequent activity in the workflow template. The process flow of the proposed DILT model is shown in Figure 2. The services are composed in sequence or in parallel based on the workflow patterns. The workflow patterns such as Sequence, AND-Split, AND-Join, XOR-Join, and XOR-Split have been considered in the schema. The approach adapted by the DILT model is depicted in Figure 3. A set of following middleware services have been designed and deployed.

- *WFParser* for parsing the workflow template
- *WSSelection* for selecting a SOAP-based web service from *WSRegistry* and *SOAPBinding* for binding them
- *URIRetrieval* for selecting RESTful web services from *RESTfulRepository* and *RESTBinding* for binding them

4 Case Study

The proposed DILT model is illustrated using an e-Governance application of Land Registration (LR) as depicted in Figure 4 . which involves searching, selecting and buying a residential plot that suits the user preferences and registering

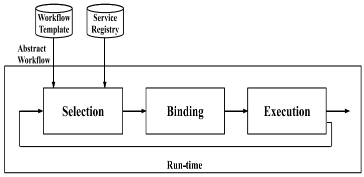


Fig. 1. Proposed DILT Model

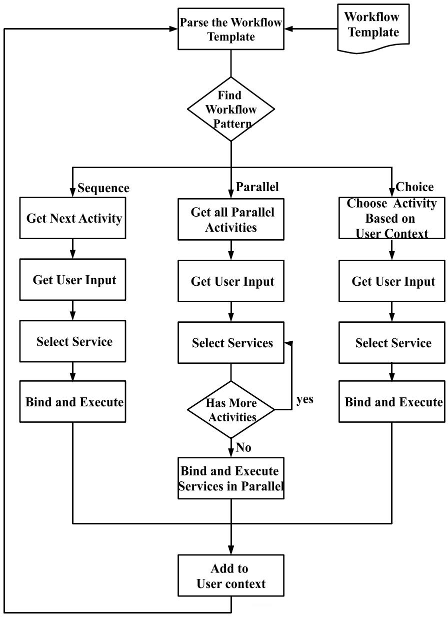


Fig. 2. Process flow of DILT model

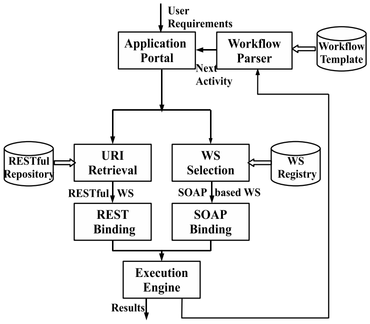


Fig. 3. Composition and Execution of Heterogeneous Web Services

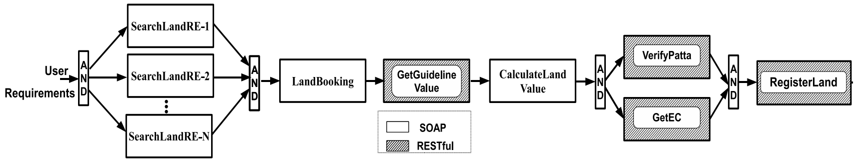


Fig. 4. Workflow for Land Registration System

the sale deed with the government department for registration.The LR application involves collaboration among the various real estate providers as well as Government departments such as the registration and the revenue departments. The business workflow for Land Registration depicted in Figure 4 involves sequential composition of searching for a land with specific requirements, selecting the land, obtaining the land value fixed by Government, calculating the total land cost, validating land details, and registering the land.

5 Experimentation

The LR application is implemented using the proposed DILT model. The application is implemented using the technologies, JAX-WS for SOAP-based web services and JAX-RS for RESTful services. The impact of DILT model on the

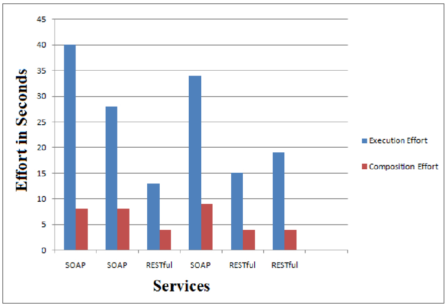
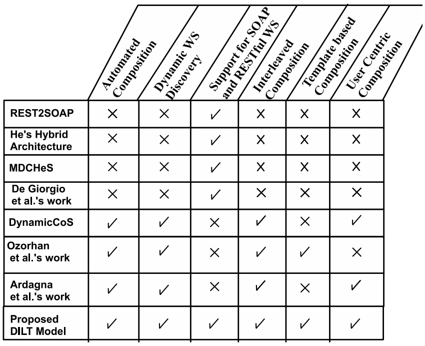


Fig. 5. Execution Effort versus Composition Effort



	Automated Composition	Dynamic WS Discovery	Support for SOAP and RESTful WS	Interleaved Composition	Template based Composition	User Centric Composition
REST2SOAP	×	×	✓	×	×	×
He's Hybrid Architecture	×	×	✓	×	×	×
MDCHeS	×	×	✓	×	×	×
De Giorgio et al.'s work	×	×	✓	×	×	×
DynamicCoS	✓	✓	×	✓	×	✓
Ozorhan et al.'s work	✓	✓	×	✓	✓	×
Ardagna et al.'s work	✓	✓	×	✓	×	✓
Proposed DILT Model	✓	✓	✓	✓	✓	✓

Fig. 6. Comparison of DILT Model with the Existing works

performance of LR application is studied by measuring the effort involved in executing the various activities. The execution effort for obtaining the results of an activity, ($\mathrm{EE_i}$) includes the efforts involved in composing a suitable service i, ($\mathrm{CE_i}$) and then executing it. $\mathrm{RT_i}$ represents the response time of a service. $EE_i = CE_i + RT_i$. The composition effort involved in dynamically composing a service i, ($\mathrm{CE_i}$) consists of the time incurred in selecting a suitable service ($\mathrm{ST_i}$) and binding the service based on its type ($\mathrm{BT_i}$). $CE_i = ST_i + BT_i$. The execution effort involved in the case of SOAP based services is high compared to RESTful services, since SOAP messages need to be prepared whereas in case of RESTful message, it involves only a HTTP request.

The percentage of additional effort spent in dynamic service selection by the proposed approach in DILT model, has been measured and shown in Figure 5. It is observed that on an average, 26% of execution effort is incurred in service selection during run-time. Though dynamic selection, adapted in the DILT model incurs an additional overhead, the advantages of the proposed approach such as adapting to new services/providers, service selection based on the outcome of previous service execution, and dynamic composition of heterogeneous services justify the overhead.

A comparison of the proposed DILT model with the existing works discussed in Section 2 is presented in Figure 6.

6 Conclusion

The major contribution of this research work is to compose heterogeneous web services *dynamically*. By interleaving the composition and execution steps, the outcome of the service saved as user context becomes useful in specifying the user preferences for selecting the subsequent services. Since e-Governance applications follow a predefined ordering among the activities of the business process, the template based approach adapted in the proposed DILT model suits such applications. The proposed model requires the user to specify the input parameter values for any service, only when the service is composed and executed. Thus, the user can preserve the privacy of data and can avoid security risks.

References

1. Pautasso, C.: Restful Web Service Composition with BPEL for REST. International Journal on Data and Knowledge Engineering 68, 851–866 (2009)
2. Schahram, D., Wolfgang, S.: A Survey on Web Services Composition. Int. J. Web and Grid Services 1(1), 1–30 (2005)
3. Agarwal, V., Chafle, G., Mittal, S., Srivastava, B.: Understanding approaches for Web Service Composition and Execution. In: ACM Conference COMPUTE 2008, pp. 1:1–1:8 (2008)
4. Peng, Y.-Y., Ma, S.-P., Lee, J.: REST2SOAP: A framework to integrate SOAP services and RESTful services. In: 2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–4 (2009)
5. He, K.: Integration and Orchestration of Heterogeneous Services. In: IEEE Joint Conferences on Pervasive Computing (JCPC), pp. 467–470 (2009)
6. Chan, P.P.W., Lyu, M.R.: Dynamic Web Service Composition: A New Approach in Building Reliable Web Service. In: 22nd International Conference on Advanced Information Networking and Applications, pp. 20–25 (2008)
7. Wang, X., Yue, K., Huang, J.Z., Zhou, A.: Service Selection in Dynamic Demand-Driven Web Services. In: IEEE Int. conference on Web Services (ICWS), pp. 376–383 (2004)
8. Zeng, L., Ngu, A.H., Benatallah, B., Podorozhny, R., Lei, H.: Dynamic Composition and Optimization of Web Services. International Journal on Distributed and Parallel Databases 24, 45–72 (2008)
9. Fan, X., Umapathy, K., Yen, J., Purao, S.: An Agent-based Approach for Interleaved Composition and Execution of Web Services. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 582–595. Springer, Heidelberg (2004)
10. Maamar, Z., Sheng, Q.Z., Benatallah, B.: Interleaving Web Services Composition and Execution using Software Agents and Delegation. In: Workshop on Web Services and Agent-Based Engineering (WSABE 2003), pp. 51–59 (2003)
11. Farokhi, S., Ghaffari, A., Haghighi, H., Shams, F.: MDCHeS: Model-Driven Dynamic Composition of Heterogeneous Service. International Journal of Communications, Network and System Sciences, 249–259 (1987)
12. De Giorgio, T., Ripa, G., Zuccalà, M.: An Approach to Enable Replacement of SOAP Services and REST Services in Lightweight Processes. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 338–346. Springer, Heidelberg (2010)
13. da Silva, E.G., Pires, L.F., van Sinderen, M.: A-DynamiCoS: A Flexible Framework for User-centric Service Composition. In: IEEE 16th International Conference on Enterprise Distributed Object Computing (EDOC), pp. 81–92 (2012)
14. Ardagna, D., Pernici, B.: Adaptive Service Composition in Flexible Processes. IEEE Transactions on Software Engineering 33(6), 369–384 (2007)
15. Ozorhan, E.K., Kuban, E.K., Cicekli, N.K.: Automated Composition of Web Services with the Abductive Event Calculus. International Journal on Information Sciences 180(19), 3589–3613 (2004)