Poster: ASSEMBLE: A Collaborative Business Process Development Tool

Ayesha Afzal, Ahmad Akhtar, Nabila Ishfaq, Basit Shafiq and Shafay Shamail
Lahore University of Management Sciences
Lahore, Pakistan
{13030032,16030059,basit,sshamail}@lums.edu.pk

Abeer Elahraf, Jaideep Vaidya and Nabil Adam
Rutgers University
Newark, NJ, USA
{aelahraf,jsvaidya,adam}@rutgers.edu

ABSTRACT

Business Process (BP) development is a challenging task for small and medium organizations who do not have sufficient resources for design, coding, and management of their BPs. Cloud infrastructure and service-oriented middleware can be leveraged for rapid development and deployment of BPs of such organizations. BP development in the cloud-based environment can be done collaboratively by exploiting the knowledge of existing BPs of related organizations. In this paper, we present ASSEMBLE, a tool for collaborative BP development in the cloud. ASSEMBLE implements our service mapping approach that utilizes the attribute, structural and semantics information of service operations of existing BPs in a given domain to help a user organization to compose its BP. Given a collection of related BPs and available service operations of a user organization, ASSEMBLE computes a mapping between service operations of the user organization and BP operations of other organizations. The tool also generates the executable BP code in standard BPEL language for deployment on a process execution engine on the user organization's site or on the cloud.

KEYWORDS

Business process development, Web services, service mapping.

1 INTRODUCTION

Cloud computing infrastructure and service-oriented middleware have enabled development of large-scale business Processes (BPs) that span multiple organizational domains. Such BPs require integration of user organization's internal Web services, services of partner organizations, and/or other third-party services hosted anywhere in the cloud. However, BP development is a challenging task for small and medium organizations. These organizations may not have sufficient resources to transform their BP specification and user requirements into an implementation level design that can be binded to the available internal, partner and/or third party Web services, for rapid BP development and deployment. BP development in cloud environment can be done collaboratively by exploiting the knowledge of existing BPs of other organizations. For example, the cloud service provider can compose BP of a user organization based on its requirements and internal/partner Web services by searching for existing BPs of related organizations and mapping the user organization's internal/partner services to the activities of existing BPs. However, the syntactical and semantic differences between the service APIs of existing BPs and internal/partner services of a user organization pose significant challenges.

For collaborative BP development we have developed a tool called ASSEMBLE. This tool implements our service mapping approach in [1]. Essentially, our approach exploits the attribute, structural and semantics information of service operations of existing BPs in a given domain to help a user organization to compose its BP. Given a collection of related BPs and available service operations of a user organization, ASSEMBLE computes a mapping between service operations of the user organization and BP operations of other organizations. ASSEMBLE also generates executable BP in standard Business Process Execution Langugage (BPEL), which can be deployed on any process execution engine anywhere on the cloud.

ASSEMBLE provides the following features for collaborative BP development: (1) user requirement specification, including available service operations and BP constraints; (2) mapping of internal / partner service operations of a user organization to existing BPs; (3) customization and refinement of the generated BP; and (4) generation of executable BP code.

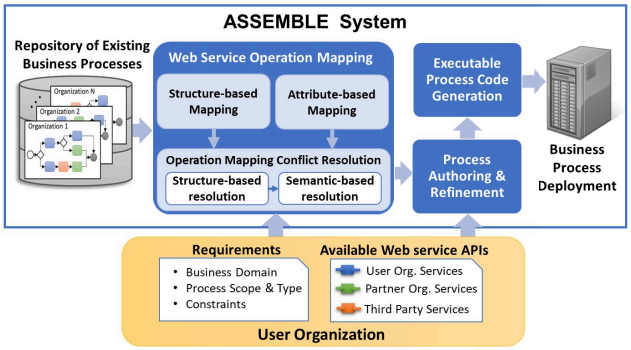

Figure 1: Architectural Overview of ASSEMBLE System

2 ARCHITECTURE AND IMPLEMENTATION

Figure 1 shows the component level architecture of the ASSEMBLE system. Below we discuss the steps for collaborative BP development performed by ASSEMBLE's components:

(1) Requirements specification: A user organization interested in developing a BP provides high-level requirements including business domain (e.g., sales, or manufacturing), process type (e.g., sales

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195017

order management), and subtype (e.g., create customer order). User can also specify any constraints on inclusion and/or exclusion of activities as well as any precedence or dependency constraints. ASSEMBLE then selects relevant processes of similar type/scope from its repository of existing BPs. The system also requires WSDL based application programming interfaces (APIs) of available services.

(2) Web service operation mapping: Given the collection of selected BPs, ASSEMBLE computes a mapping between the service operations in these BPs and internal/partner service operations of the user organization. This mapping is performed in two steps:

(i) *Attribute-based operation mapping*: An initial mapping between the user organization's service operations and operations of selected BPs is computed based on similarity of their attributes. The underlying assumption here is that functionally similar service operations have similar input and output attributes though their name and datatype/structure might be different. ASSEMBLE's attribute-based operation mapping approach builds on probabilistic schema matching approach proposed by Sarma et al. [2].

(ii) *Conflict Resolution*: Attribute-based mapping results may include inaccurate and conflicting mappings. Figure 2 shows such an example, where attribute-based mapping for *createAndSaveOrder* BP operation resulted in two conflicting mappings with user operations, *createOrder* and *updateOrderInventory*. ASSEMBLE implements two algorithms to resolve such operation mapping conflicts. First algorithm determines the mapping correctness based on the degree of overlap between the attribute-based matches of user operation and the structural matches of the corresponding BP operation in existing BPs. For finding the structural matches of service operations in existing BPs, the approach builds on the event matching algorithm proposed by Zhu et al. [3]. The second algorithm employs Colored Petri net (CPN) reachability analysis to resolve operation mapping conflicts. The idea is to explore those paths in the reachability tree that satisfy the precondition of a given user organization operation mapped to the BP operation. For any given mapped operation, if no such path exists from initial state corresponding to start of BP, then the mapping is considered incorrect.

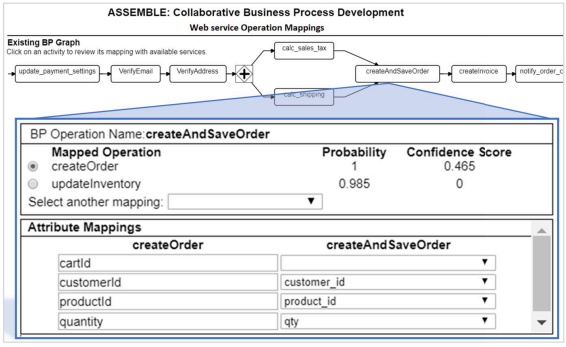

Figure 2: ASSEMBLE's Authoring and Refinement Interface

(3) Customization and refinement of generated BP: After service operation mapping, ASSEMBLE's authoring and refinement interface presents a ranked list of matching BPs to the user. The ranking is in order of number of BP operation mappings with available service operations. User can select one of the matching BPs for customization/ refinement. For any given operation in the selected BP, user can view and change its operation mappings as shown in Figure 2. Other refinements that the user may perform include modifying BP's controlflow by adding or removing Web services, and updating the dataflow by modifying the variable assignments and branching conditions. Once the user is satisfied with the process, executable code is generated in BPEL that can be deployed on a BP execution server at user organization's site or on the cloud.

3 EXPERIMENTAL EVALUATION

ASSEMBLE's BP repository contains over a hundred e-commerce BPs from several open source ERP systems and e-commerce platforms including Odoo, inoERP, Tryton, Shopify, BigCommerce, Volusion and Lexity, etc. For validation of mapping results, we considered the Web service APIs of three publicly available e-Commerce platforms, namely Magento, Konakart and WooCommerce in [1]. The number of relevant service operation considered were 53 for Magento, 269 for Konakart and 162 for WooCommerce.

Figure 3 shows the accuracy of our approach in terms of f-measure and the time for online computation of service operation mapping. These results show that our proposed approach effectively computes Web service operation mappings providing reasonably high accuracy and low computation time.

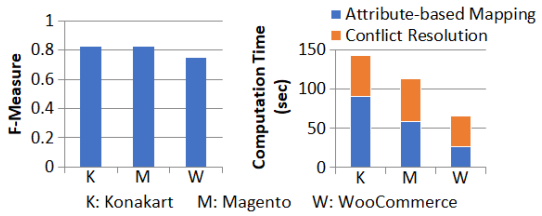

Figure 3: Experimental Evaluation Results for e-Commerce APIs

4 CONCLUSION AND FUTURE WORK

In this paper, we present an implementation of our approach for collaborative business process development that exploits the attribute, structural and semantics information of service operations of existing BPs. The intended users of ASSEMBLE are small and medium organizations that cannot bear with the high cost of personnel and software/hardware resources for BP design and development.

In future, we aim to conduct a user study by involving professional BP developers and domain experts. The key objective is to test ASSEMBLE's usability and evaluate its capability to develop real-world processes in different business domains.

ACKNOWLEDGMENTS

The work of Shafiq and Shamail is supported by Ignite grant and by HEC grant under the PAK-US Science and Technology Cooperation Program. The work of Adam is supported by the National Academy of Sciences, under the PAK-US Science and Technology Cooperation Program and by NSF IUCRC grant (CNS-1624503).

REFERENCES

[1] A. Afzal, B. Shafiq, S. Shamail, A. Elahraf, J. Vaidya, and N. R. Adam. Assemble: Attribute, structure and semantics based service mapping approach for collaborative business process development. *IEEE Transactions on Services Computing*, 2018.

[2] A. Das Sarma, X. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *Proceedings of 2008 ACM SIGMOD Intl. Conference on Management of Data*, pages 861–874, 2008.

[3] X. Zhu, S. Song, J. Wang, P. S. Yu, and J. Sun. Matching heterogeneous events with patterns. In *2014 IEEE 30th International Conference on Data Engineering (ICDE)*, pages 376–387. IEEE, 2014.