An Artifact-Centric Business Process Execution Platform

Kan Ngamakeur, Sira Yongchareon and Veronica Liesaputra
Department of Computing
Unitec Institute of Technology
New Zealand
sira@maxsira.com;
{kngamakeur, vliesaputra}@unitec.ac.nz

Chengfei Liu
Department of Computer Science and Software Engineering
Swinburne University of Technology
Australia
cliu@swin.edu.au

Jian Yu
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
New Zealand
jian.yu@aut.ac.nz

***Abstract*— Artifact-centric modeling has become an alternative, yet promising approach of business process (BP) modeling and management as it provides higher flexibility than that of traditional activity-centric approaches. However, existing BP execution engines require artifact-centric models be transformed to executable activity-centric BP languages (e.g., BPEL) in order to be executed and managed. We argue that the model conversion incurs losses of information and affects traceability and monitoring ability of BPs, especially where BPs span across inter-organizations. In this paper, we present the design and implementation of an execution platform for artifact-centric BPs. We evaluated our platform using a case study and that can demonstrate several key benefits over the use of existing systems to run artifact-centric BPs.**

Keywords—Business process execution engine; Artifact-centric workflows; Business Process Management systems; Service-Oriented Architecture

I. INTRODUCTION

An artifact-centric process modeling has emerged as an alternative approach of specifying a business process (BP). In the past several years, this new approach has become promising to BP management as research results have significantly demonstrated its higher modeling efficiency and flexibility than that of traditional activity-centric approaches [1]. The approach focuses on describing how business-relevant key data entities, known as "*artifacts*", evolve in a BP.

Although artifact-centric BP modeling has been well studied in the past many years, existing BP execution approaches require artifact-centric BP models be transformed to executable activity-centric BP languages (e.g., [4], [7], [9], [12]). Performing model transformation incurs losses of information and affects traceability and monitoring ability of BPs, especially in Service Oriented Architecture (SOA) where BPs span across multiple inter-business entities. In order to support direct and automatic execution of artifact-centric models while utilizing service-oriented architecture in the collaboration, we designed and developed an artifact-centric BP execution platform for collaborative artifact-centric BPs. Furthermore, we discussed key benefits over the use of existing systems to run artifact-centric BPs.

II. ARTIFACT-CENTRIC BP EXECUTION FRAMEWORK

In this section, we discuss our Artifact-centric BP Execution framework, as shown in Fig. 1.

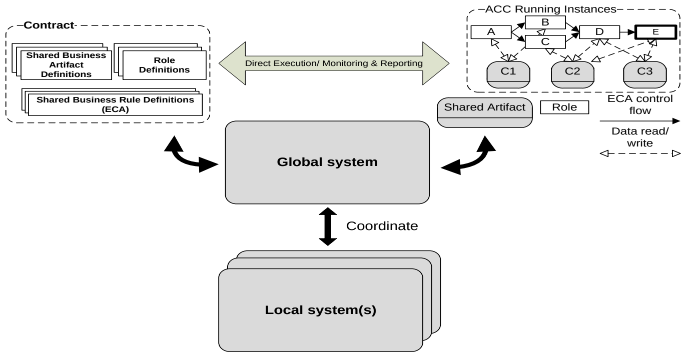


Fig. 1. Artifact-centric BP Execution framework for collaborative business processes

There are two parts in the framework: (1) artifact-centric BP model with a contract for collaborative BP and (2) BP execution can be performed based on the defined model and contract.

A. *Executable Model for Collaborative BPs*

We borrowed a concept of *Artifact-Centric Collaboration Model (ACC model)* [1] to design our executable BP model and the notions of *shared artifact* and *local artifact* are utilized in this work for executing artifact-centric BP in a collaborative environment. Based on that, we proposed an XML serialized version as an executable model of the ACC model.

We designed *shared artifacts* associated with a set of business rules and organization/actor roles and *shared business rules* to regulate state transitions of the shared artifacts. Each business rule contains a pre-condition on a state of the artifact. If the state of an artifact satisfies the condition of a business rule, a participating role is notified to take over a control of shared artifacts. We also use *roles* to define stakeholders that are responsible for controlling shared artifact for a period of time in a BP execution. Each role provides a set of service that makes changes on data and life cycles of artifacts.

B. *Platform Architecture and its Components*

The architecture of our platform is based on centralized system rather than peer-to-peer systems since the nature of business rules exhibits centralized-control behavior and they can be managed efficiently using a single repository [3]. We utilized the event-driven architecture [10] and the SOA to design and implement the centralized controller to support BP execution across organizations. Shared artifacts and business rules are used to serve as a contract and to govern interaction between organizations, respectively [1]. The platform comprises of an *Artifact-centric collaboration* (*ACC*) *system* and a *local Artifact-centric Process (ACP) system*, as illustrated in Fig. 2. The ACC system acts as a central controller that coordinates all local ACP systems, each of which runs and supports each organization involved in the collaboration.

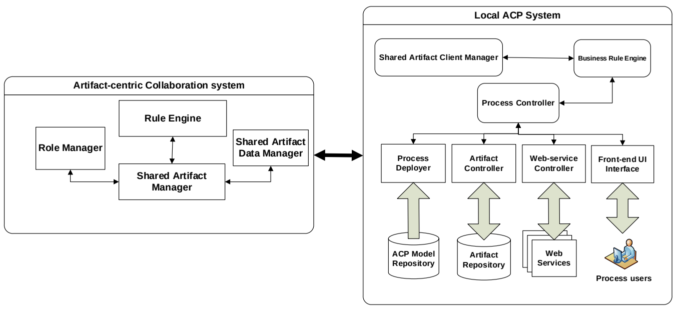


Fig. 2. Platform’s architecture consisting of ACC and ACP systems

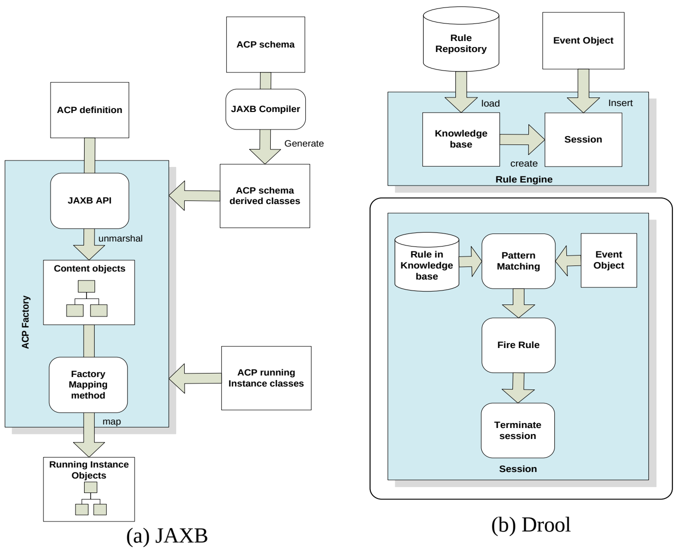


Fig. 3. ACP Factory using (a) JAXB and (b) Rule Engine using Drool

Let us briefly describe how the two systems work. In the ACC system, *Shared Artifact Manager* provides management functionality to ensure each contract running in the execution is created, managed and updated correctly. *Rule Engine* is used to deliver functionalities of business rule evaluation and service invocation. *Role Manager* handles a task allocated to each role involved in a particular BP. A task or a session allocated to each role/organization is determined by a rule engine. *Shared Artifact Data Manager* performs a task of reading or updating shared artifacts. Shared business data and artifacts’ states reflect the current stage of a collaborative BP. Although the system aims to manage operations on contracts in the centralized controller, a particular part of information of a shared artifact is taken from its corresponding local artifact that is managed by a certain role or organization. In a local ACP system, *Shared Artifact Client Manager* is designed to manage communications between the ACC system and its local system. The main functionalities include receiving and passing messages issued by the central controller to the local system, detecting a status of process execution of the local system and notifying the ACC system regarding a completion of a task or a session of the local system. Both types of systems work in a synchronized manner to provide and ensure consistency of process execution across all participating organizations.

III. Platform Features and Benefits

In this section, we illustrate and discuss the features and advantages of our platform.

A. *Coordination between ACC System and Local ACP Systems*

To effectively achieve the correct and consistent coordination among a global system and all local systems involved in the collaboration, we propose to use *coordination messages* and *notification messages* to facilitate the collaboration among ACC systems and local ACP systems, as illustrated in Fig. 4. For the coordination system, we propose an idea of using a *collaboration instance* to help coordinate the three types of artifact-centric process related instances: *SharedArtifact instance*, *SharedRule instance*, and *Role Instance*.

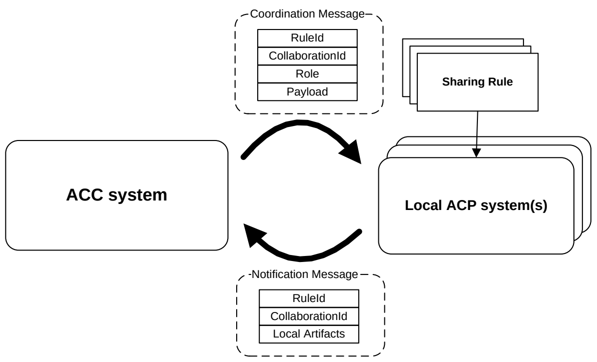


Fig. 4. The coordination between the ACC system and the ACP system(s)

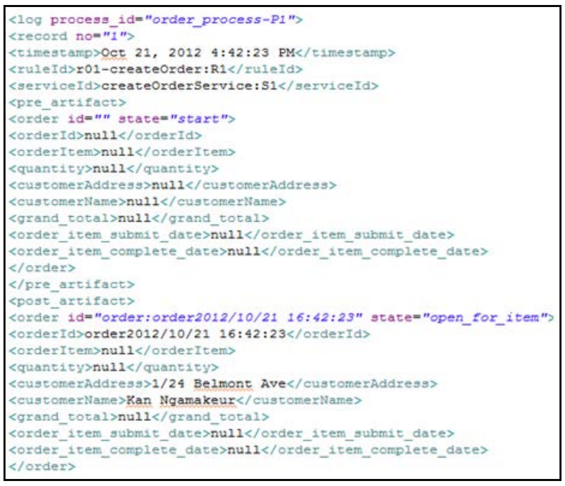
A coordination message containing details for supporting correlation between a local process and the global process is created. The content of a coordination message contains *collaboration id*, *rule id*, *role*, and *a payload message*. The coordination message is assigned with correlation data and a payload message and then is sent to a corresponding role.

Once a coordination message arrives at a designate local ACP system, the Shared Artifact Client Manager accesses the message and stores the correlation information in a list of correlation keys. Then, the payload message is passed to an ACP system. The detail of internal operations of the local system is omitted here as it is described in [8].

After an invocation of a service in a local system, data hold by artifact instances related to the process are checked whether it satisfies any sharing rule. If so, the client manager creates a notification message with related correlation information. Once the message arrives at The ACC system, the role manager determines a corresponding collaboration instance using correlation information stored in the message. After the collaboration instance is identified, the shared artifact data manager updates information on shared artifact(s).

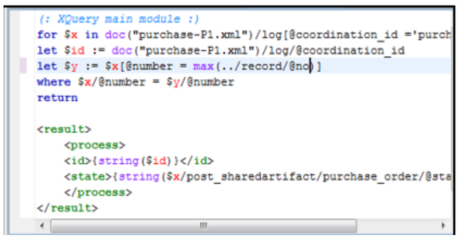
B. *Monitoring and tracking*

In an inter-organizational BP, each organization manages its own local artifacts and the progress of a local process can be tracked using local artifacts. However, the progress of a local process is only visible to an organization that the process belongs to. Thus, there is no way for the other participants in this collaboration to understand the current status or progress of a collaborative process. In this implementation, shared artifacts are used to address monitoring and tracking requirements across organizations. Shared artifacts store information that is shared from local artifacts that are required to support the global process. A shared artifact has its own life cycles and its state represents a current stage of some corresponding local artifact(s). Based on our case study, as the information and state of local artifacts are transferred to its associated shared artifact, we can monitor and track the global process. Fig. 5 shows an example of process log data stored in the ACC and Fig. 6 shows an XQuery script that is used to extract process information to track the progress of the process.



```
<log process_id="order_process-P1">
<record no="1">
<timestamp>Oct 21, 2012 4:42:23 PM</timestamp>
<ruleId>r01-createOrder:R1</ruleId>
<serviceId>createOrderService:S1</serviceId>
<pre_artifact>
<order id="" state="start">
<orderId>null</orderId>
<orderItem>null</orderItem>
<quantity>null</quantity>
<customerAddress>null</customerAddress>
<customerName>null</customerName>
<grand_total>null</grand_total>
<order_item_submit_date>null</order_item_submit_date>
<order_item_complete_date>null</order_item_complete_date>
</order>
</pre_artifact>
<post_artifact>
<order id="order:order2012/10/21 16:42:23" state="open_for_item">
<orderId>order2012/10/21 16:42:23</orderId>
<orderItem>null</orderItem>
<quantity>null</quantity>
<customerAddress>1/24 Belmont Ave</customerAddress>
<customerName>Kan Ngamakeur</customerName>
<grand_total>null</grand_total>
<order_item_submit_date>null</order_item_submit_date>
<order_item_complete_date>null</order_item_complete_date>
</order>
```

Fig. 5. Process log data



```
(: XQuery main module :)
for $x in doc("purchase-P1.xml")/log[@coordination_id ='purch
let $id := doc("purchase-P1.xml")/log/@coordination_id
let $y := $x[@number = max(../record/@no)]
where $x/@number = $y/@number
return

<result>
    <process>
    <id>{string($id)}</id>
    <state>{string($x/post_sharedartifact/purchase_order/@sta
    </process>
</result>
```

Fig. 6. XQuery script for quering Process log data

C. *Flexibility and changes management*

Flexibility is one of the main advantages of the artifact-centric approach as an artifact-centric model is defined in a declarative manner. With this approach, we are able to easily adapt BPs in response to changes in business environment. This advantage of the artifact model has been proved in many researches (e.g., in [2], [3], [5]). In our platform, an artifact-centric process model is realized using a direct approach, i.e., no model conversion needed. Therefore, the flexibility in a conceptual model is inherited to an executable model. Flexibility is well supported for both the design-time and the run-time. We also gain the benefit of loose coupling from using SOA in our execution platform. Process changes are allowed with the use of our proposed changes validation approach presented in [1]. Ad-hoc/on-demand changes of artifact's data and business rules are supported in which a local process is allowed to change as long as the changes do not affect the collaboration contract managed by the ACC system.

IV. Case Study

In this section, we introduce our case study based on a product ordering process as shown in Fig. 7. There are three roles or organization involved in this BP. These roles include *Buyer*, *Supplier*, and *Logistic*.

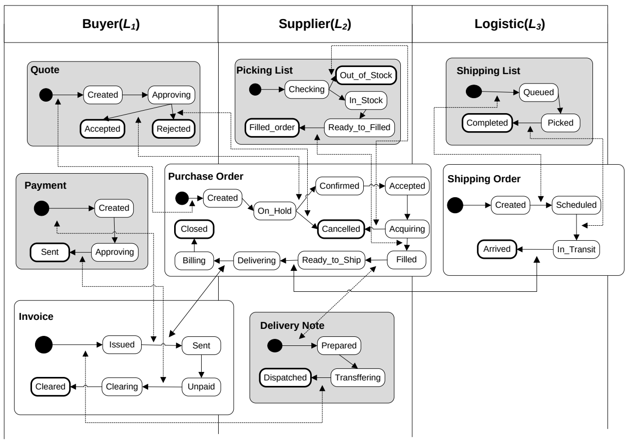
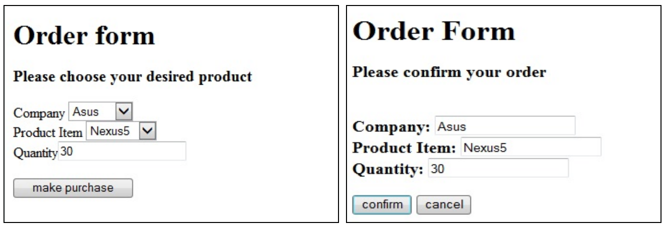


Fig. 7. An overall collaboration for the ordering process

The ACC model is directly mapped to the proposed process definition which will be deployed on the ACC system. A user can interact with the deployed process with the system via a Web service interface. This process starts when the purchase order request is lodged through the web user interface as shown in Fig. 8(a).


Order form

Please choose your desired product

Company: Asus
Product Item: Nexus5
Quantity: 30

make purchase

Order Form

Please confirm your order

Company: Asus
Product Item: Nexus5
Quantity: 30

confirm cancel

(a) Create the order (b) Confirm or cancel the order

Fig. 8. A simple web user interface for ordering process

The system receives and processes the request. The rule engine is invoked to validate the request against the business rule. If the conditions are satisfied, *Buyer* role will be called to create and update *Purchase Order* artifact including its attribute and state. After the rule is fired, *Buyer* role starts its local process and create its local artifacts. *Buyer* role shares *Quote* artifact's information with *Purchase Order* artifact managed in the collaboration process. At a certain stage of this process, a user needs to confirm or cancel the purchase order. The web user interface is provided for a user as shown in Fig. 8(b). Our ACC and ACP systems together with the case study are publicly available for download from [11].

V. Conclusion and Future Work

This paper studied how artifact-centric BPs can be executed in a collaborative environment. We then identified and analyzed requirements for the execution of artifact-centric collaboration model (ACC model) and proposed an execution platform that addresses the requirements. Our findings show several key benefits over the use of the existing systems to run artifact-centric BPs. However, there are still opening issues to be further investigated in order to improve the platform, such as run-time verification, exception handling, change management, and BP recovery.

References

[1] Yongchareon, S., Liu, C., Yu, J., Zhao, X.: A View Framework for Modeling and Change Validation of Artifact-Centric Inter-Organizational Business Processes, Information systems, 2015, vol. 47, pp. 51-81.

[2] Kumaran, S., Liu, R., Wu, F.Y.: On the Duality of Information-Centric and Activity-Centric Models of Business Processes. In: CAISE 2008, LNCS 5074, pp. 32-47.

[3] Bhattacharya, K., Hull, R., Su, J.: A data-centric design methodology for business processes, Handbook of Research on Business Process Modeling (2009)

[4] Liu, G., Liu, X, Qin, H, Su, J., Yan, Z., Zhang, L.: Automated Realization of Business Workflow Specification. In: ICSOC/ServiceWave 2009. LNCS 6275, pp. 96–108.

[5] Xu, W., Su, J., Yan, Z., Yang, J., & Zhang, L.: An artifact-centric approach to dynamic modification of workflow execution. In: Meersman, R., Dillon, T., Herrero, P. (Eds.): OTM 2011, Part I, LNCS 7044, pp. 256–273. Springer, Heidelberg (2011).

[6] Yongchareon, S., Liu, C., Zhao, X.: A Framework for Behavior-Consistent Specialization of Artifact-Centric Business Processes. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 285–301. Springer, Heidelberg (2012)

[7] Zhao, D., Liu, G., Wang, Y., Gao, F., Li, H., Zhang, D.: A-Stein: A prototype for artifact-centric business process management systems, BMEI 2011, vol. 1, pp. 247-250.

[8] Ngamakeur, K., Yongchareon, S., Liu, C.: A Framework for Realizing Artifact-Centric Business Processes in Service-Oriented Architecture. In S.-g. Lee et al. (Eds.): DASFAA 2012, Part I, LNCS 7238, pp. 63–78, Springer, Heidelberg (2012)

[9] Zhao, D., Liu, G., Jiang, Y., Gao, F., Wang, Y.: The execution and detection of artifact-centric business process. In: IEEE International Conference on Computer Science and Automation Engineering, vol.4, pp.491-495. IEEE Press, Shanghai (2011)

[10] Michelson, B. M.: Event-driven architecture overview. Patricia Seybold Group,(2006)

[11] Ngamakeur, K., Yongchareon, S.: ACC & ACP systems prototype, https://sites.google.com/site/maxsirayongchareon/artifact-s

[12] Fan, B., Li, F., Liu. S., Zhang, Y.: Run JTA in JTang: Modeling in Artifact-Centric Model and Running in Activity-Centric Environment, In: Asia Pacific Business Process Management 2015, Lecture Notes in Business Information Processing (LNBIP), vol. 219, pp 83-97.