Computers, Materials & Continua
DOI:10.32604/cmc.2021.015280
Article

Tech Science Press

Formal Approach to Workflow Application Fragmentations Over Cloud Deployment Models

Hyun Ahn and Kwanghoon Pio Kim[*]

Division of Computer Science and Engineering, Gyeonggi, 16227, Korea
[*]Corresponding Author: Kwanghoon Pio Kim. Email: kwang@kyonggi.ac.kr
Received: 13 November 2020; Accepted: 31 December 2020

Abstract: Workflow management technologies have been dramatically improving their deployment architectures and systems along with the evolution and proliferation of cloud distributed computing environments. Especially, such cloud computing environments ought to be providing a suitable distributed computing paradigm to deploy very large-scale workflow processes and applications with scalable on-demand services. In this paper, we focus on the distribution paradigm and its deployment formalism for such very large-scale workflow applications being deployed and enacted across the multiple and heterogeneous cloud computing environments. We propose a formal approach to vertically as well as horizontally fragment very large-scale workflow processes and their applications and to deploy the workflow process and application fragments over three types of cloud deployment models and architectures. To concretize the formal approach, we firstly devise a series of operational situations fragmenting into cloud workflow process and application components and deploying onto three different types of cloud deployment models and architectures. These concrete approaches are called the deployment-driven fragmentation mechanism to be applied to such very large-scale workflow process and applications as an implementing component for cloud workflow management systems. Finally, we strongly believe that our approach with the fragmentation formalisms becomes a theoretical basis of designing and implementing very large-scale and maximally distributed workflow processes and applications to be deployed on cloud deployment models and architectural computing environments as well.

Keywords: Cloud workflows; cloud deployment model; workflow application fragmentations; information control net

1 Introduction

Workflow management technology is used to model, automate, monitor, and optimize repetitive tasks within a variety of workflow processes, including business processes and scientific workflow processes. As the proliferation of distributed computing systems and the service-oriented architecture (SOA) has progressed rapidly, the field of workflow management has encountered a new challenge: The transition from conventional workflow management technology into this new state of the art technology. In particular, the need for scalability and on-demand services for large-scale workflow processes has been emphasized in traditional workflow management environments. In this regard, cloud computing deserves to be considered as the most well-suited computing environment for managing large-scale workflow processes, particularly for scientific workflows. Accordingly, many researchers and practitioners have begun to recognize the potential of the cloud workflow management system (CWMS), which runs on a cloud computing environment, and the need to further develop its functionalities needed.

A CWMS is capable of enacting complex workflow processes requiring massive computing resources distributed across multiple clouds. Therefore, technical efforts for scalability are not required because computing infrastructure services are provided by third-party providers (e.g., Amazon, Microsoft, and IBM), and one can easily access their services at any time. The availability of on-demand-service is another advantage of the CWMS. End users with different goals (e.g., process designers and process performers) can access the CWMS and immediately request and receive the services they require.

In this paper, we focus on distributing workflow applications across multiple cloud environments. Workflow applications, which are implementations of tasks (or activities) comprising workflow processes, are characterized by their heterogeneity and demand for substantial computing resources. For instance, a scientific workflow process, which is a type of large-scale workflow process, generally consists of a set of data-intensive tasks implemented as diverse and massive applications for scientific experiments. Therefore, the effective distribution of workflow applications across different cloud environments would lead to greater management and maintenance capabilities for large-scale workflow processes. To address this issue, we propose an approach to workflow application fragmentations based on cloud deployment models to satisfy high-level requirements related to the execution of cloud workflow processes (e.g., the security issue). We also present formalisms for describing cloud workflow models and workflow application fragmentations based on deployment models. Concretely, the formalisms we propose are based on the information control net theory [1], which was developed for modeling workflow processes. We add a new modeling feature for specifying cloud deployment information to this theory to fit our approach.

The remainder of this paper is organized as follows. In Section 2, we detail the need for workflow application fragmentations and describe how workflow applications can be fragmented based on cloud deployment models. In Section 3, we provide some formal definitions and describe our proposed algorithm for workflow application fragmentations. In Section 4, we introduce a preliminary architecture for a cloud workflow engine supporting our proposed approach. In Section 5, a summary of related works will be presented. Finally, we conclude this paper and discuss future work in Section 6.

2 Cloud Workflow Environments with Cloud Deployment Models

2.1 Cloud Deployment Model

Over the past few years, we have witnessed an era of remarkable growth in the field of cloud computing and its applications. Although there is no doubt that cloud computing is the most attractive option for building enterprise information systems, there are still issues that must be addressed carefully. One of the key issues in cloud computing is deployment models. It is important to choose a method for deploying cloud services based on the preferences of an organization. IT policy, business characteristics, the required level of data governance, elasticity, and security should be taken into account when selecting a deployment model. Regarding cloud deployment, there are four major types of deployment models: private, community, public, and hybrid. Details of each deployment model are summarized in Tab. 1.

Table 1: Summary of the cloud deployment models

Deployment type	Description	Managed by	Advantages	Challenges
Private cloud	The cloud infrastructure and services are exclusively provisioned for a single organization and managed by the organization.	Organization	High utilization of existing in-house resources Full control over security/policy concerns	Required in-house skill and staffs High risk of ownership
Community cloud	The cloud infrastructure and services are exclusively provisioned for a specific community of organizations that have common objectives.	Organization (community member) or third-party provider	Support for large inter-organizational projects	Interoperability and compliance issues
Public cloud	The cloud infrastructure and services are provisioned for open use by the general public.	Third-party provider	Faster deployments	Elasticity and flexibility
Hybrid cloud	The cloud infrastructure and services are configured as a combination of two or more deployment types (private, community, or public).	Organization or third-party provider	High elasticity Customizability	Most complicated configuration Standardization, interoperability, and compliance issues

The private cloud model is characterized by being built and provided for only a single organization. That organization has ownership of the management and operations of the cloud infrastructure and services. However, this means that most of the responsibility for the availability of services and data privacy also belongs to that organization. In terms of computing resource utility, a private cloud may be well-suited for organizations that already own data centers and developed IT infrastructure and want to reutilize existing in-house computing resources. On the other extreme, a public cloud can be accessed by any customers with different needs and concerns. Organizations using a public cloud may benefit from the advantages of the offerings provided by cloud providers, such as scalability and reliability of services, as well as fast deployment. Because of these benefits, the public cloud model is the dominant deployment model in cloud computing. However, there is the potential for failure of service-level agreements (SLAs) because the organization does not claim full control of the cloud infrastructure. A community cloud infrastructure is constructed to support a specific inter-organizational project. The community consists of member organizations that have shared concerns or objectives, and their cloud is managed by the community (single or multiple members) or a third-party organization. The core function of a community cloud is to accelerate cooperation between member organizations and the integration of resources for the project. Although there are excellent examples of community clouds, such as the European organization for nuclear research, known as CERN, the challenges of interoperability and compliance still exist. Finally, a hybrid cloud is a combination of two or more deployment models (private, community, and public). Organizations apply this model to leverage the advantages of multiple models by the outsourcing low-priority activities (or high-computation tasks) to a public cloud and controlling core activities through a private cloud [2].

2.2 The Rationale of Workflow Application Fragmentations

Historically, workflow management technology and systems are known to have originated from efforts in the field of office automation in the mid-80s. Technological maturation was achieved in the early 90s [3], and these systems now have two main branches: business process management (BPM), and scientific workflows.

- Workflow applications in business domains: BPM technology, as a successor to workflow technology, has been actively employed in many industries, including the financial, e-government, and manufacturing industries. In particular, as the SOA concept has become a key role in the development of enterprise information systems in recent years, most industries are leveraging these technologies for orchestrating and enacting web services (e.g., WS-BPEL). In this context, the migration issue for traditional workflow applications has arisen (i.e., the transformation from legacy systems to web services). Consequently, many enterprises have begun focusing on exploiting the potential of cloud computing for migrating workflow applications and accelerating their business process enactments. For example, traditional workflow applications in the banking industry are typically composed of numerous program functions and implemented on legacy systems. In this situation, Reference [4] suggested that the employment of cloud computing would facilitate the separation of tasks for the business processes of a banking firm and their related workflow applications into a sensitive task group and non-sensitive/computation-intensive task group. The tasks in these groups could then be performed flexibly, either on-premise or in the cloud.
- Workflow applications in scientific domains: Cloud infrastructures are more crucial in scientific workflows compared to the BPM domain. In fact, several scientific experiments require substantial data-intensive tasks and encompass a wide variety of workflow applications that perform scientific tasks [5,6]. In order to handle such vast computations, distributed computing paradigms (e.g., grid and cluster computing) have naturally been introduced to enable scientists to analyze their data in a parallel and distributed manner. In particular, cloud computing is the most attractive option for scientists because it masks the complexities of developing and managing a computing infrastructure for scientific experiments. According to [7], many fields of science, such as astronomy, physics, biology, and geology, have been supported by scientific workflow technology through cloud computing.

In summary, the cloud workflow has proven useful in automating and managing large-scale workflows for business and scientific experiments to achieve competitive advantages by utilizing cloud computing (e.g., scalability and alleviation of concerns regarding maintenance). In this context, we argue that a proper method to fragment and deploy workflow applications in different cloud environments should be considered during the stages of designing and implementing a CWMS. The term “workflow application” refers to an invoked application, which includes program logic for the execution of one or more tasks within a workflow process. Workflow applications have historically been implemented in the form of legacy systems or monolithic programs that are not strongly integrated with a workflow management system. For the numerous workflow applications operating in heterogeneous environments, standardization and maintenance issues became increasingly difficult and led to the requirement for expensive modernization efforts.

As web service-based technology has emerged, coarse-grained workflow applications have been decomposed and reconstructed for the granularity of service to fit modern business processes or other business needs. In this context, we introduce a new approach to fragment and distribute a group of workflow applications based on cloud deployment models. This approach facilitates the intentional configuration of workflow application distributions to satisfy high-level requirements for operating a CWMS (e.g., privacy concerns for enacting workflow processes).

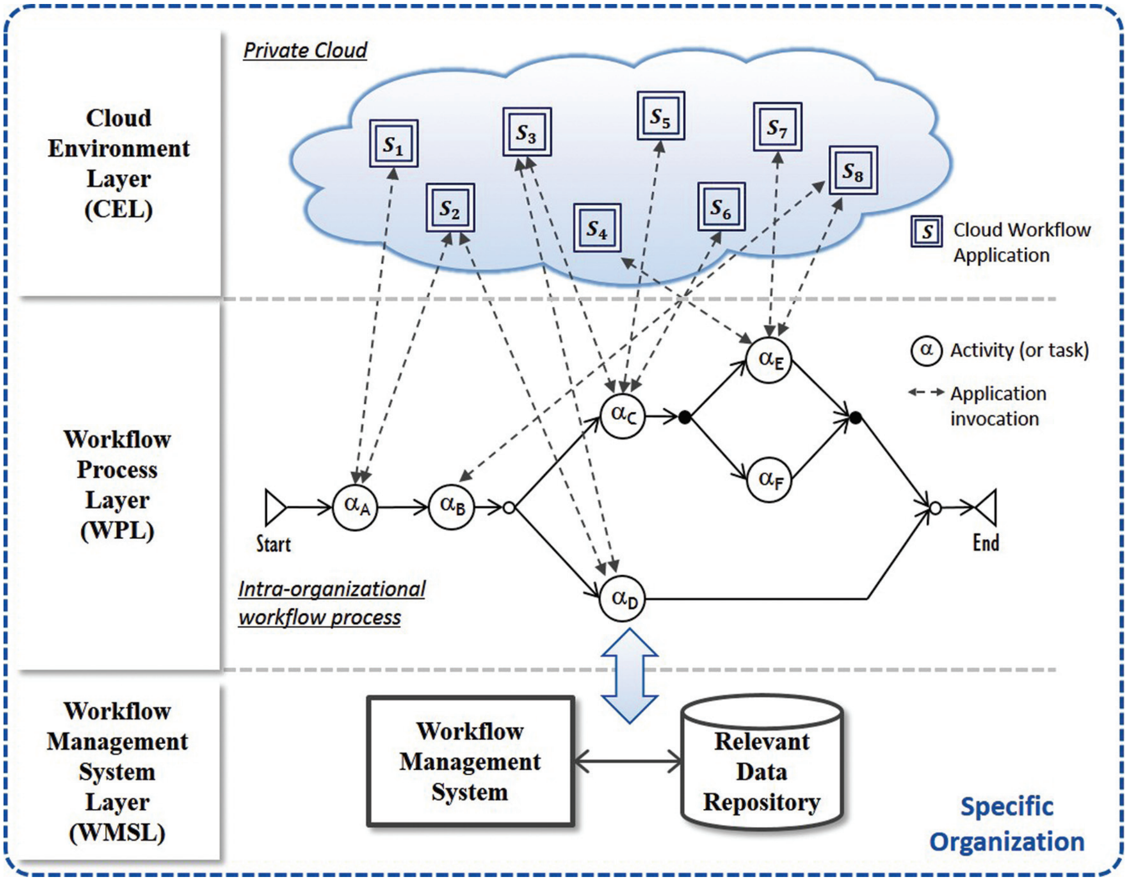

Figure 1: A private-model-based operational environment

2.3 *Cloud-Deployment-Specific Workflow Environments*

Assuming that an organization exploits a cloud deployment model when designing and implementing a CWMS, the operational environment will vary to a large extent upon the selected deployment model.

- Private-model-based environment: The environment for a cloud workflow management system consists of three sublayers: the cloud environment layer (CEL), workflow process layer (WPL), and workflow management system layer (WMSL), as shown in Fig. 1. According to the private model, the set of workflow applications pertaining to security-critical tasks or workflow processes is deployed on a private cloud. The CEL manages migrated workflow applications and is accessed only by a specific organization. The WPL and WMSL are also built and operated within the coverage of the organization. If an organization requires a high degree of security for the execution of workflow processes and the core processes fall within an intra-organizational scope, the private deployment model should be selected as the best-fitting model.

- Community-model-based environment: The community cloud deployment model assumes that many organizations participate in a community with a common goal and have expectations for co-ownership of the cloud. Therefore, in this model, each organization independently manages its own WMSL to provide the system functionalities for performing shared workflow processes (i.e., inter-organizational processes). As you can see in Fig. 2, the layers that a single organization can control are limited to WMSLs while WPLs and CELs are controlled and managed by community members. A supply chain management (SCM) process in which multiple business organizations collaborate during the entire process from manufacturing to distribution is a good example of this model. In the field of scientific workflows, many international research projects adopt the community deployment model and operate inter-organizational workflow processes for large-scale scientific experiments.

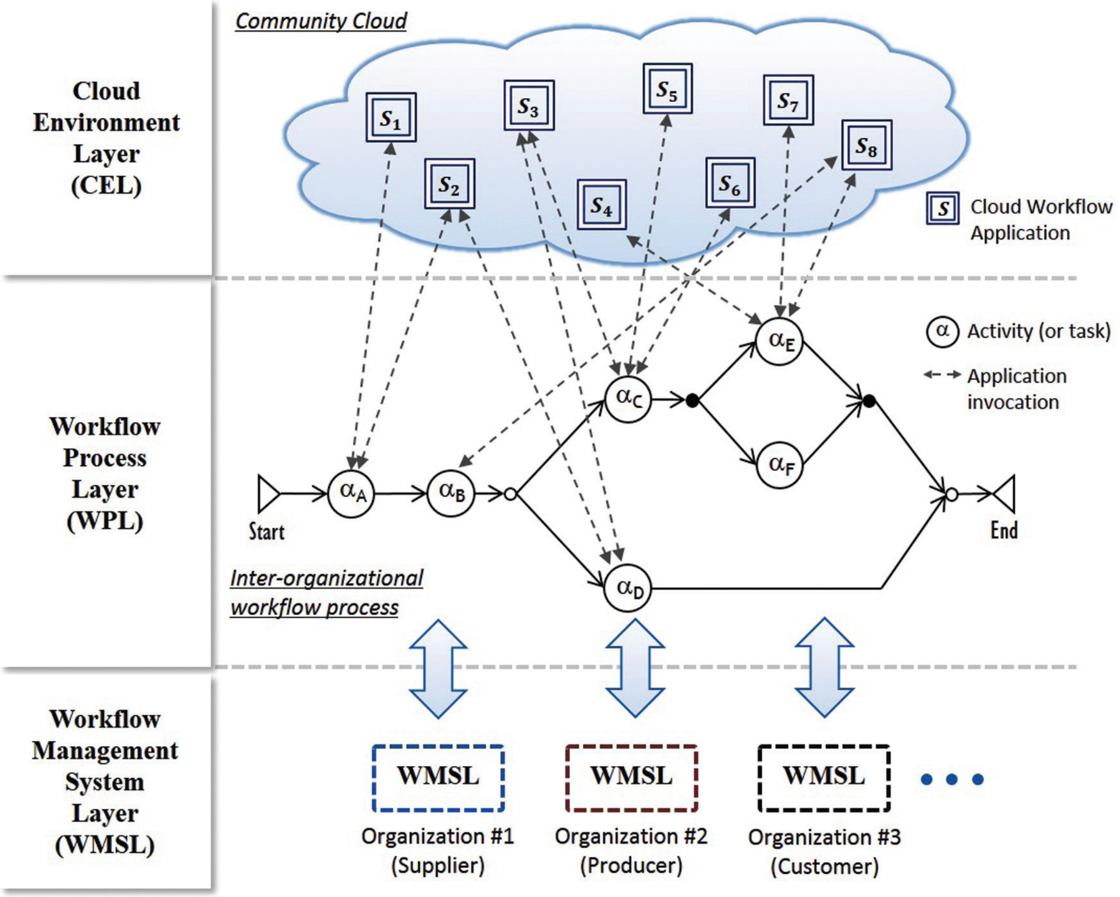
Cloud Environment Layer (CEL)
Community Cloud
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
S Cloud Workflow Application
Workflow Process Layer (WPL)
α Activity (or task)
Application invocation
Start
α_A
α_B
α_C
α_D
α_E
α_F
End
Inter-organizational workflow process
Workflow Management System Layer (WMSL)
WMSL
Organization #1 (Supplier)
WMSL
Organization #2 (Producer)
WMSL
Organization #3 (Customer)

Figure 2: A community-model-based operational environment

- Public-model-based environment: According to the characteristics of the public model, all workflow applications are openly exposed and shared with all organizations accessing a public cloud. Organizations, such as research institutes, enterprises, and government agencies, that independently manage WMSLs and WPLs, are permitted to freely utilize workflow applications provided by a public cloud managed by a third-party provider, as shown in Fig. 3. For example, APROMORE [8] is a public-model-based cloud platform that shares business process models and resources and supports business process analytics. Many related researchers have utilized and contributed to this public platform. Therefore, the public model is attractive because it encourages public interest through workflow application sharing and communication activation.
- Hybrid-model-based environment: The hybrid model, which includes multiple deployment models, is an appropriate model for organizations that simultaneously operate inter- and intra-organizational workflow processes. Additionally, this model is well-suited to environments in which cloud bursting [9] is crucial when enacting large-scale workflow processes.

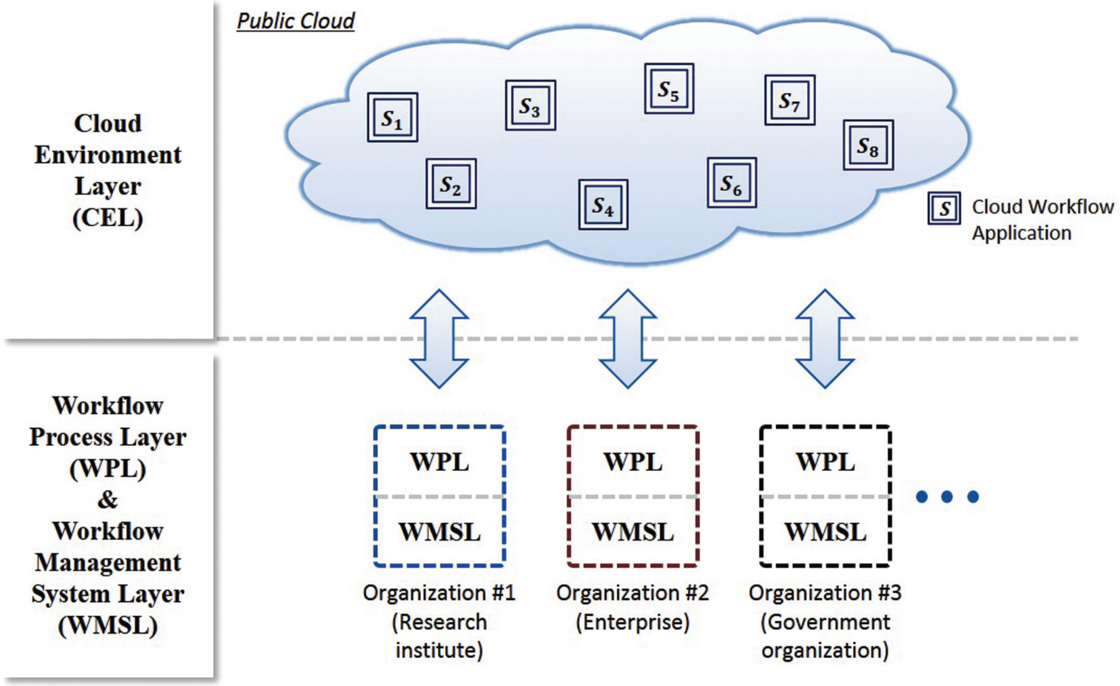
Public Cloud
Cloud Environment Layer (CEL)
S1
S2
S3
S4
S5
S6
S7
S8
S Cloud Workflow Application
Workflow Process Layer (WPL) & Workflow Management System Layer (WMSL)
WPL
WMSL
Organization #1 (Research institute)
WPL
WMSL
Organization #2 (Enterprise)
WPL
WMSL
Organization #3 (Government organization)
• • •

Figure 3: A public-model-based operational environment

As above, we describe different cloud workflow environments based on their type of cloud deployment model. For the sake of clarity, we presented simple examples, in which a single deployment model was applied to, and assumed a process-wise deployment strategy. However, the ultimate operational environment for cloud workflows that we pursue in this study is geared toward multi-cloud environments with activity-wise deployments. In other words, it is more natural and practical to apply a deployment model to each activity so that workflow applications invoked from each activity will be deployed to different cloud environments, rather than deploying the entire workflow applications to a specific cloud. This feature enables more sophisticated configurations for the enactments of cloud workflows and related workflow application distributions. Fig. 4 illustrates a composite environment for cloud workflows that encompasses multiple deployment models, processes, and organizations, and facilitates activity-wise deployments of workflow applications.

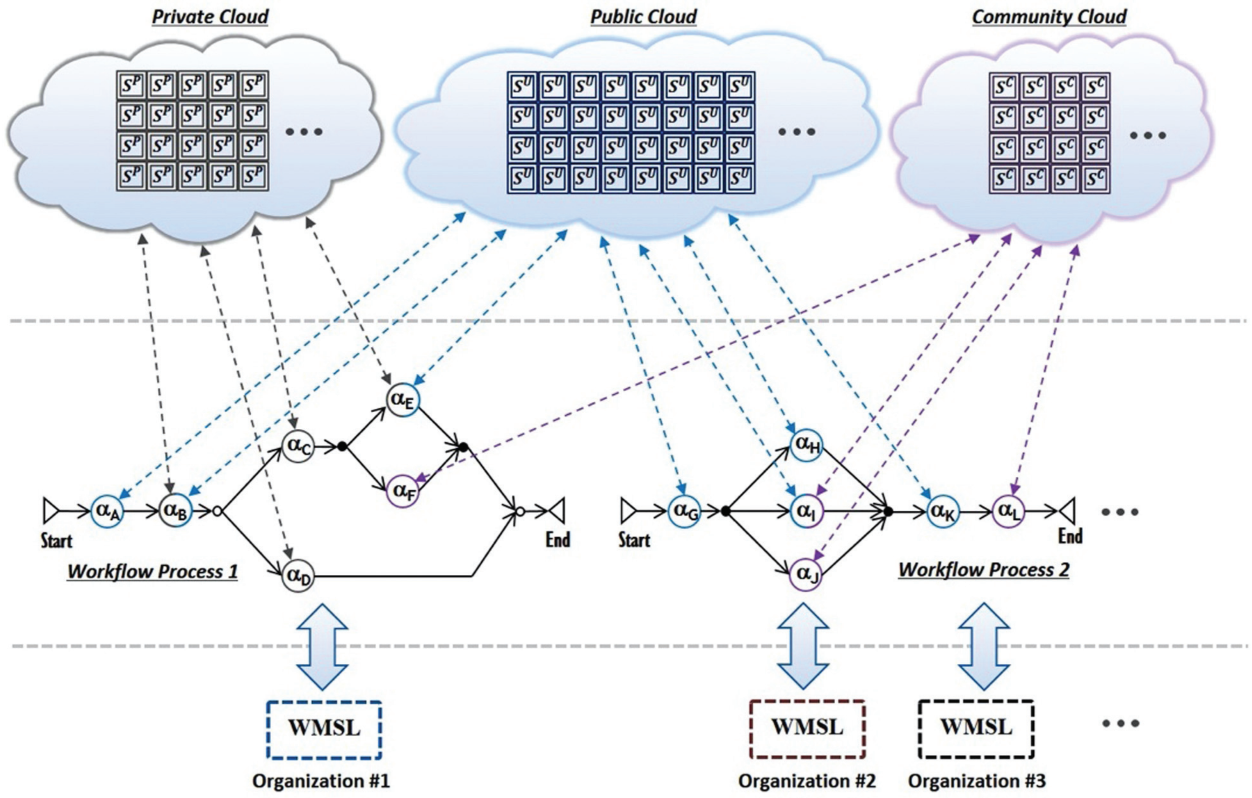

Figure 4: Composite environment with multiple clouds and deployment models

3 Formal Description of Our Approach

In this section, we formally describe the cloud workflow model that represents the theoretical basis for this work. We then describe the steps for workflow application fragmentations based on cloud deployment models. Additionally, the information control net (ICN) theory, presented in [1], is used as a process modeling theory in this paper.

3.1 Cloud Workflow Model

A workflow process is described by incorporating information from a few primary aspects: control-flow, data, organization, and resources. Based on this convention, we add the cloud aspect to the ICN methodology to describe cloud workflow processes. Fig. 5 depicts the meta-model of our ICN-based cloud workflow. This meta-model consists of essential entity types and their relationships. The following are the basic definitions of the primary entity types in the meta-model.

- Cloud workflow process: Defined by a predefined set of activities (or tasks) and their precedence/succession relationships. A CWMS has functionalities for modeling, enacting, and controlling defined workflow processes in cloud environments.
- Activity: An entity type that represents the basic unit of work comprising a cloud workflow process. Each activity has not only precedence relationships with other activities but also association relationships with the entity types of several aspects of the cloud workflow, e.g., transition condition, relevant data, role, workflow application, and deployment type.
- Role: A logical unit of the organizational structure. It is concerned with duties, skills, and authorities required for the execution of a particular activity. In addition, each role can take responsibility for multiple activities.
- Actor: A person capable of performing an activity through the associated role. Each actor can participate in multiple roles.
- Relevant data: Input/output data objects used to perform an activity. They can also be embedded in transition conditions corresponding to disjunctive patterns (i.e., OR-split and XOR-split) for making decisions in a workflow process enactment.
- Workflow application: A software program that is invoked during the phase of execution of an activity and automatically processes related computational tasks.
- Deployment type: Refers to the cloud environment responsible for executing a particular activity. Workflow applications associated with the execution of the activity will be deployed on different cloud environments based on this specified information.

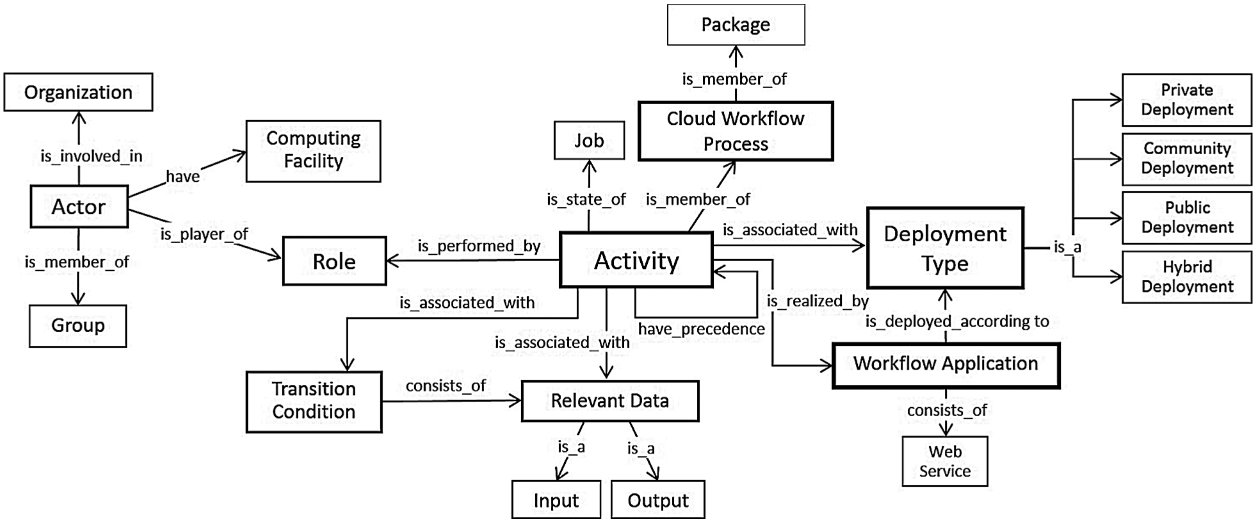

Figure 5: Meta-model of the ICN-based cloud workflow

As described above, a completely defined ICN-based cloud workflow model contains workflow entities for various aspects and the relationships between those entities. In this paper, we focus on the cloud aspect, particularly focusing on fragmenting workflow applications based on the deployment types assigned to activities. The following is a definition of a partial workflow model (PWM), which serves as the cloud-deployment-oriented portion of the cloud workflow model:

Definition 1 (Partial workflow model). A *partial workflow model* (PWM) is formally defined as a six-tuple $M = (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}, \delta, \lambda, \tau)$, where $\boldsymbol{A}$ is a finite set of activities, $\boldsymbol{S}$ is a finite set of workflow applications, $\boldsymbol{D}$ is a finite set of cloud deployment types, and three mapping functions (δ, λ, τ) are defined as follows:

- $\delta = \delta_i \cup \delta_o$, where, δ_i: $\boldsymbol{A} \rightarrow \mathcal{P}(\boldsymbol{A})$ is a multi-valued mapping function from an activity $\alpha \in \boldsymbol{A}$ to a set of activities which directly precede to α in the execution orderings; δ_o: $\boldsymbol{A} \rightarrow \mathcal{P}(\boldsymbol{A})$ is a multi-valued mapping function from an activity $\alpha \in \boldsymbol{A}$ to a set of activities which directly success to α in the execution orderings.
- $\lambda = \lambda_s \cup \lambda_a$, where, λ_s: $\boldsymbol{A} \rightarrow \mathcal{P}(\boldsymbol{S})$ is a multi-valued mapping function from an activity $\alpha \in \boldsymbol{A}$ to a set of applications invoked for the execution of α; λ_a: $\boldsymbol{S} \rightarrow \mathcal{P}(\boldsymbol{A})$ is a multi-valued mapping function from an application $s \in \boldsymbol{S}$ to a set of activities which invoke s during their execution.
- $\tau = \tau_d \cup \tau_a$, where, τ_d: $\boldsymbol{A} \rightarrow \mathcal{P}(\boldsymbol{D})$ is a multi-valued mapping function from an activity $\alpha \in \boldsymbol{A}$ to a set of deployment types associated with α; τ_a: $\boldsymbol{D} \rightarrow \mathcal{P}(\boldsymbol{A})$ is a multi-valued mapping function from a deployment type $\mu \in \boldsymbol{D}$ to a set of activities associated with μ.

Fig. 6 presents an example PWM. In this figure, activities, workflow applications, and deployment types are visualized as circles, double-lined squares, and pentagons, respectively. The visualization of a pentagon for the hybrid deployment type is skipped because this type can be specified as a combination of other deployment types (e.g., private and public). For the control-flow aspect, the model depicts the precedence relationships between activities (unidirectional arcs) and gateway activities such as OR-split/join (small empty circles) and AND-split/join (small-filled circles). The invocation relationships between activities and workflow applications are depicted as bidirectional dotted arcs, and the association relationships between activities and deployment types are depicted as dotted edges. Finally, the two triangles represent the start and end events of the PWM, respectively.

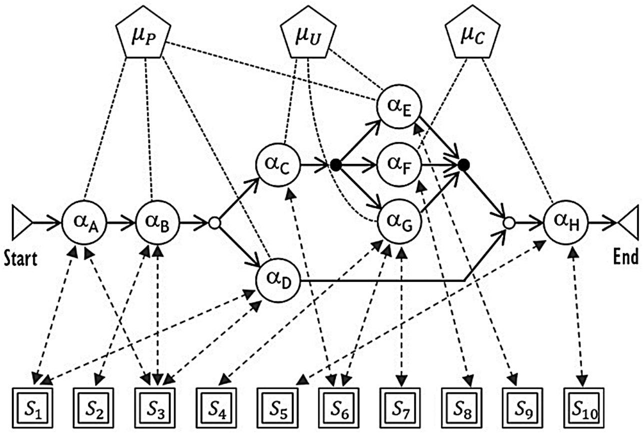

Figure 6: An example of PWM

The model contains eight activities $(\alpha_A, \ldots, \alpha_H)$, ten workflow applications $(s_1, \ldots, s_{10})$, and three deployment types (μ_P, μ_U, μ_C). There are many-to-many relationships established between activities and workflow applications, e.g., the activity α_A invokes two workflow applications (s_1, s_3) in its execution and the workflow application s_3 is invoked from three activities $(\alpha_A, \alpha_B, \alpha_D)$ in their executions. Except for the activity α_E, each activity has an association with a single deployment type. The activity α_E is connected to two deployment types (μ_P, μ_U) because of the assignment of the hybrid deployment type. Consequently, the workflow application s_9 invoked from the activity α_E will be deployed on both private and public cloud environments. The detailed formal representation of the model is listed in Tab. 2.

Table 2: Formal representation of the example model

$\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}, \delta, \lambda, \tau)$		/* The Partial Workflow Model */		
$\boldsymbol{A} = \{\alpha_A, \alpha_B, \alpha_C, \alpha_D, \alpha_E, \alpha_F, \alpha_G, \alpha_H\}$		/* Activities */		
$\boldsymbol{S} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$		/* Workflow Applications */		
$\boldsymbol{D} = \{\mu_P, \mu_U, \mu_C\}$		/* Cloud Deployment Models */		
$\delta = \delta_i \cup \delta_o$	$\delta_i(\alpha_A) = \varnothing;$	$\delta_i(\alpha_B) = \{\{\alpha_A\}\};$	$\delta_o(\alpha_A) = \{\{\alpha_B\}\};$	
	$\delta_i(\alpha_C) = \{\{\alpha_B\}\};$	$\delta_i(\alpha_D) = \{\{\alpha_B\}\};$	$\delta_o(\alpha_B) = \{\{\alpha_C\}, \{\alpha_D\}\};$	
	$\delta_i(\alpha_E) = \{\{\alpha_C\}\};$	$\delta_i(\alpha_F) = \{\{\alpha_C\}\};$	$\delta_o(\alpha_C) = \{\{\alpha_D, \alpha_E, \alpha_F\}\};$	
	$\delta_i(\alpha_G) = \{\{\alpha_C\}\};$		$\delta_o(\alpha_D) = \{\{\alpha_H\}\};$	$\delta_o(\alpha_E) = \{\{\alpha_H\}\};$
	$\delta_i(\alpha_H) = \{\{\alpha_D\}, \{\alpha_E, \alpha_F, \alpha_G\}\};$		$\delta_o(\alpha_F) = \{\{\alpha_H\}\};$	$\delta_o(\alpha_G) = \{\{\alpha_H\}\};$
			$\delta_o(\alpha_H) = \varnothing;$	
$\lambda = \lambda_s \cup \lambda_a$	$\lambda_s(\alpha_A) = \{\{s_1, s_3\}\};$	$\lambda_s(\alpha_B) = \{\{s_2, s_3\}\};$	$\lambda_a(s_1) = \{\{\alpha_A, \alpha_D\}\};$	$\lambda_a(s_2) = \{\{\alpha_B\}\};$
	$\lambda_s(\alpha_C) = \{\{s_6\}\};$	$\lambda_s(\alpha_D) = \{\{s_1, s_3\}\};$	$\lambda_a(s_3) = \{\{\alpha_A, \alpha_B, \alpha_D\}\};$	
	$\lambda_s(\alpha_E) = \{\{s_9\}\};$	$\lambda_s(\alpha_F) = \{\{s_8\}\};$	$\lambda_a(s_4) = \{\{\alpha_G\}\};$	$\lambda_a(s_5) = \{\{\alpha_H\}\};$
	$\lambda_s(\alpha_G) = \{\{s_4, s_6, s_7\}\};$		$\lambda_a(s_6) = \{\{\alpha_C, \alpha_G\}\};$	$\lambda_a(s_7) = \{\{\alpha_G\}\};$
	$\lambda_s(\alpha_H) = \{\{s_5, s_{10}\}\};$		$\lambda_a(s_8) = \{\{\alpha_F\}\};$	$\lambda_a(s_9) = \{\{\alpha_E\}\};$
			$\lambda_a(s_{10}) = \{\{\alpha_H\}\};$	
$\tau = \tau_d \cup \tau_a$	$\tau_d(\alpha_A) = \{\{\mu_P\}\};$	$\tau_d(\alpha_B) = \{\{\mu_P\}\};$	$\tau_a(\mu_P) = \{\{\alpha_A, \alpha_B, \alpha_D, \alpha_E\}\};$	
	$\tau_d(\alpha_C) = \{\{\mu_U\}\};$	$\tau_d(\alpha_D) = \{\{\mu_P\}\};$	$\tau_a(\mu_U) = \{\{\alpha_C, \alpha_E, \alpha_G\}\};$	
	$\tau_d(\alpha_E) = \{\{\mu_P, \mu_U\}\};$	$\tau_d(\alpha_F) = \{\{\mu_C\}\};$	$\tau_a(\mu_C) = \{\{\alpha_F, \alpha_H\}\};$	
	$\tau_d(\alpha_G) = \{\{\mu_U\}\};$	$\tau_d(\alpha_H) = \{\{\mu_C\}\};$		

3.2 *Workflow Application Fragmentations*

In order to formally describe the step of workflow application fragmentations, we define the workflow application fragment model that is a transformation result from a partial workflow model by a fragmentation algorithm.

Definition 2 (Workflow Application Fragment Model). A *workflow application fragment model* is formally defined as a six-tuple $\boldsymbol{M}_{\mathbf{F}} = (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}, \boldsymbol{E}, \gamma, \chi)$, where $\boldsymbol{A}$ is a finite set of activities, $\boldsymbol{S}$ is a finite set of workflow applications, $\boldsymbol{D}$ is a finite set of cloud deployment types, and $\boldsymbol{E} \subseteq (\boldsymbol{D} \times \boldsymbol{D})$ is a set of arcs, and two mapping functions are defined as follows:

- $\gamma = \gamma_s \cup \gamma_d$, where, $\gamma_s\colon \boldsymbol{D} \rightarrow \wp(\boldsymbol{S})$ is a multi-valued mapping function from a deployment type $\mu \in \boldsymbol{D}$ to a set of workflow applications distributed in a cloud environment of μ; $\gamma_d\colon \boldsymbol{S} \rightarrow \wp(\boldsymbol{D})$ is a multi-valued mapping function from a workflow application $s \in \boldsymbol{S}$ to a set of deployment types $\boldsymbol{D_s} \subseteq \boldsymbol{D}$ iff s is deployed in cloud environments of $\boldsymbol{D_s}$.
- $\chi = \chi_a \cup \chi_e$, where, $\chi_a\colon \boldsymbol{E} \rightarrow \wp(\boldsymbol{A})$ is a multi-valued mapping function from an arc $(\mu_x, \mu_y) \in \boldsymbol{E}$ to a set of activities is in between μ_x and μ_y; $\chi_e\colon \boldsymbol{A} \rightarrow \wp(\boldsymbol{E})$ is a multi-valued mapping function from an activity $\alpha \in \boldsymbol{A}$ to a set of arcs in which α is placed on.

A workflow application fragment model is a network model in which the cloud deployment type that acquires each workflow application fragment is a nodal type. The flow relationships between these deployment types are formed through the execution of the cloud workflow process in the example model. The following represents details of the proposed algorithm for the theoretical fragmentation of workflow applications in a cloud workflow model. Specifically, the algorithm transforms a PWM (Definition 1) to a workflow application fragment model (Definition 2) by linking deployment types and workflow applications based on the relationships with the corresponding activities.

The fragment model generated by the fragmentation algorithm is graphically represented in Fig. 7. The application fragment assigned to each cloud deployment type by the algorithm is expected to be physically deployed on the corresponding cloud environment using appropriate distribution mechanisms. The arcs between the nodes of cloud deployment types represent flow relationships (e.g., the relationship from μ_P to μ_U, and its in-between activity α_C). In summary, the fragment model formally and graphically represents both the information related to workflow application distribution and the execution flow of a deployment-type-centered cloud workflow process.

Algorithm 1: Workflow application fragmentations

Input: A partial workflow model, $\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}, \delta, \lambda, \tau)$

Output: A workflow application fragment model, $\boldsymbol{M_F} = (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}, \boldsymbol{E}, \gamma, \chi)$

```
procedure
  $\boldsymbol{E} \leftarrow \varnothing$;
  for $\forall \alpha \in \boldsymbol{A}$ do                                              ▷ for each activity $\alpha$
    $\gamma_s(\tau_d(\alpha)) \leftarrow \gamma_s(\tau_d(\alpha)) \cup \lambda_s(\alpha)$; ▷ add applications to the corresponding deployment type
    for $\forall s \in \lambda_s(\alpha)$ do                         ▷ for each application invoked by activity $\alpha$
      $\gamma_d(s) \leftarrow \gamma_d(s) \cup \tau_d(\alpha)$;          ▷ add deployment types to the corresponding application $s$
    end for
    $\boldsymbol{E} \leftarrow \boldsymbol{E} \cup (\tau_d(\alpha), \tau_d(\delta_o(\alpha)))$;                  ▷ flow relationships between deployment types
    $\chi_a(\tau_d(\alpha), \tau_d(\delta_o(\alpha))) \leftarrow \chi_a(\tau_d(\alpha), \tau_d(\delta_o(\alpha))) \cup \alpha$;          ▷ in-between activities on arcs
    $\chi_e(\alpha) \leftarrow \chi_e(\alpha) \cup (\tau_d(\alpha), \tau_d(\delta_o(\alpha)))$;                              between deployment types
  end for
end procedure
```

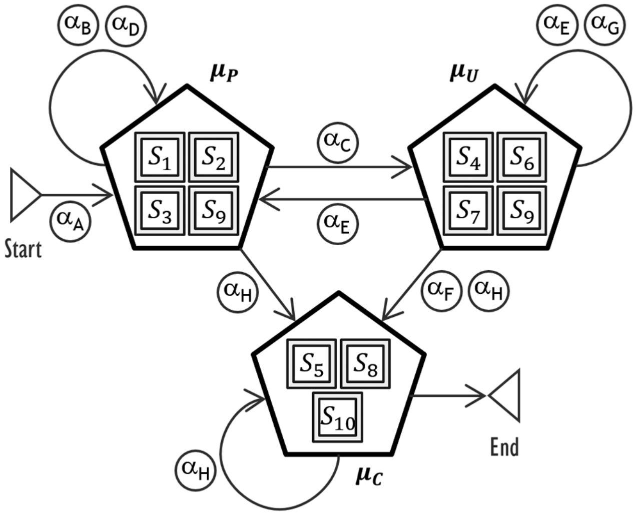

Figure 7: The workflow application fragment model generated from the example model

A formal representation of the generated fragment model is provided in Tab. 3. For convenience of explanation, two atypical deployment types (μ_{start}, μ_{end}) are introduced. We can interpret this information as the presence of undetermined cloud environments where the start and end events of the example process occur.

Table 3: Formal representation of the fragment model

$\boldsymbol{M_F} = (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}, \boldsymbol{E}, \gamma, \chi)$	/* The Workflow Application Fragment Model*/		
$\boldsymbol{A} = \{\alpha_A, \alpha_B, \alpha_C, \alpha_D, \alpha_E, \alpha_F, \alpha_G, \alpha_H\}$	/* Activities */		
$\boldsymbol{S} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$	/* Workflow Applications */		
$\boldsymbol{D} = \{\mu_{start}, \mu_P, \mu_U, \mu_C, \mu_{end}\}$	/* Deployment Types */		
$\boldsymbol{E} =$	/* Arcs among Deployment Types */		
$\{(\mu_{start}, \mu_P), (\mu_P, \mu_P), (\mu_P, \mu_U), (\mu_P, \mu_C), (\mu_U, \mu_P), (\mu_U, \mu_U), (\mu_U, \mu_C), (\mu_C, \mu_C), (\mu_C, \mu_{end})\}$			
$\gamma = \gamma_s \cup \gamma_d$	$\gamma_s(\mu_P) = \{\{s_1, s_2, s_3, s_9\}\};$	$\gamma_d(s_1) = \{\{\mu_P\}\};$	$\gamma_d(s_2) = \{\{\mu_P\}\};$
	$\gamma_s(\mu_U) = \{\{s_4, s_6, s_7, s_9\}\};$	$\gamma_d(s_3) = \{\{\mu_P\}\};$	$\gamma_d(s_4) = \{\{\mu_U\}\};$
	$\gamma_s(\mu_C) = \{\{s_5, s_8, s_{10}\}\};$	$\gamma_d(s_5) = \{\{\mu_C\}\};$	$\gamma_d(s_6) = \{\{\mu_U\}\};$
		$\gamma_d(s_7) = \{\{\mu_U\}\};$	$\gamma_d(s_8) = \{\{\mu_C\}\};$
		$\gamma_d(s_9) = \{\{\mu_P, \mu_U\}\};$	$\gamma_d(s_{10}) = \{\{\mu_C\}\};$
$\chi = \chi_a \cup \chi_e$	$\chi_a(\mu_{start}, \mu_P) = \{\{\alpha_A\}\};$	$\chi_e(\alpha_A) = \{\{(\mu_{start}, \mu_P)\}\};$	
	$\chi_a(\mu_P, \mu_P) = \{\{\alpha_B, \alpha_D\}\};$	$\chi_e(\alpha_B) = \{\{(\mu_P, \mu_P)\}\};$	
	$\chi_a(\mu_P, \mu_U) = \{\{\alpha_C\}\};$	$\chi_e(\alpha_C) = \{\{(\mu_P, \mu_U)\}\};$	
	$\chi_a(\mu_P, \mu_C) = \{\{\alpha_H\}\};$	$\chi_e(\alpha_D) = \{\{(\mu_P, \mu_P)\}\};$	
	$\chi_a(\mu_U, \mu_P) = \{\{\alpha_E\}\};$	$\chi_e(\alpha_E) = \{\{(\mu_U, \mu_P), (\mu_U, \mu_U)\}\};$	
	$\chi_a(\mu_U, \mu_U) = \{\{\alpha_E\}\};$	$\chi_e(\alpha_F) = \{\{(\mu_U, \mu_C)\}\};$	
	$\chi_a(\mu_U, \mu_C) = \{\{\alpha_F, \alpha_H\}\};$	$\chi_e(\alpha_G) = \{\{(\mu_U, \mu_U)\}\};$	
	$\chi_a(\mu_C, \mu_C) = \{\{\alpha_H\}\};$	$\chi_e(\alpha_H) = \{\{(\mu_P, \mu_C), (\mu_U, \mu_C), (\mu_C, \mu_C)\}\};$	
	$\chi_a(\mu_C, \mu_{end}) = \varnothing;$		

4 Conceptual Architecture of a Cloud Workflow Engine

Thus far, we have discussed the concept of cloud deployment models and formalisms of workflow application fragmentations through these models. To put our approach into practice, we now describe the conceptual architecture of a cloud workflow engine while accounting for the execution of cloud workflow processes using application fragmentation based on deployment models. The conceptual architecture includes client components, the workflow engine, and cloud environments. An overview of this architecture is illustrated in Fig. 8.

The architecture includes client components, such as the workflow process modeler, runtime clients, and monitoring and analysis clients, and they have graphical user interfaces for end-users. They communicate with the workflow engine that provides the functionality supporting cloud workflow management actions. The workflow engine is the core component of the CWMSs and it consists of two major parts: The modeling and deployment part, and the enactment part.

- Modeling and deployment part: The components in this part are responsible for assisting in the various phases of modeling and deployment of cloud workflow processes. First, a workflow process designer performs process modeling using various information, such as model definitions, organizational information, and relevant data provided by the modeling data agent. In this step, the information about the deployment types corresponding to each activity should be specified. After the completion of the process modeling step, the designed model is submitted to the workflow engine for deployment. Using the process model parser, the model is transformed into an executable model (e.g., WS-BPEL) and the process model verifier determines if syntax errors exist within the transformed model. Finally, based on the specified deployment type information, the workflow application manager automatically deploys workflow applications on different cloud environments through communications with application service brokers. The overall procedure for process model deployment is depicted in Fig. 9.

- Enactment part: A deployed process model can be instantiated by a human worker who has the authority to create a process instance or by triggering an event with a specific condition for starting the process. The statuses of all activated process instances are managed by the process instance manager. The enactment scheduler builds concrete plans that specify execution orders for active instances with consideration for the current statuses of cloud environments. The work item handler requests the application service broker to execute a task within the active process instance and the broker determines which cloud should perform the execution based on deployed workflow applications. As an operational example, Fig. 10 presents a sequence diagram representing a scenario in which a process instance of the deployed model is initiated by a runtime client. A certain automatic task then is executed through the invocation of web services corresponding to the required workflow applications deployed on cloud environments.

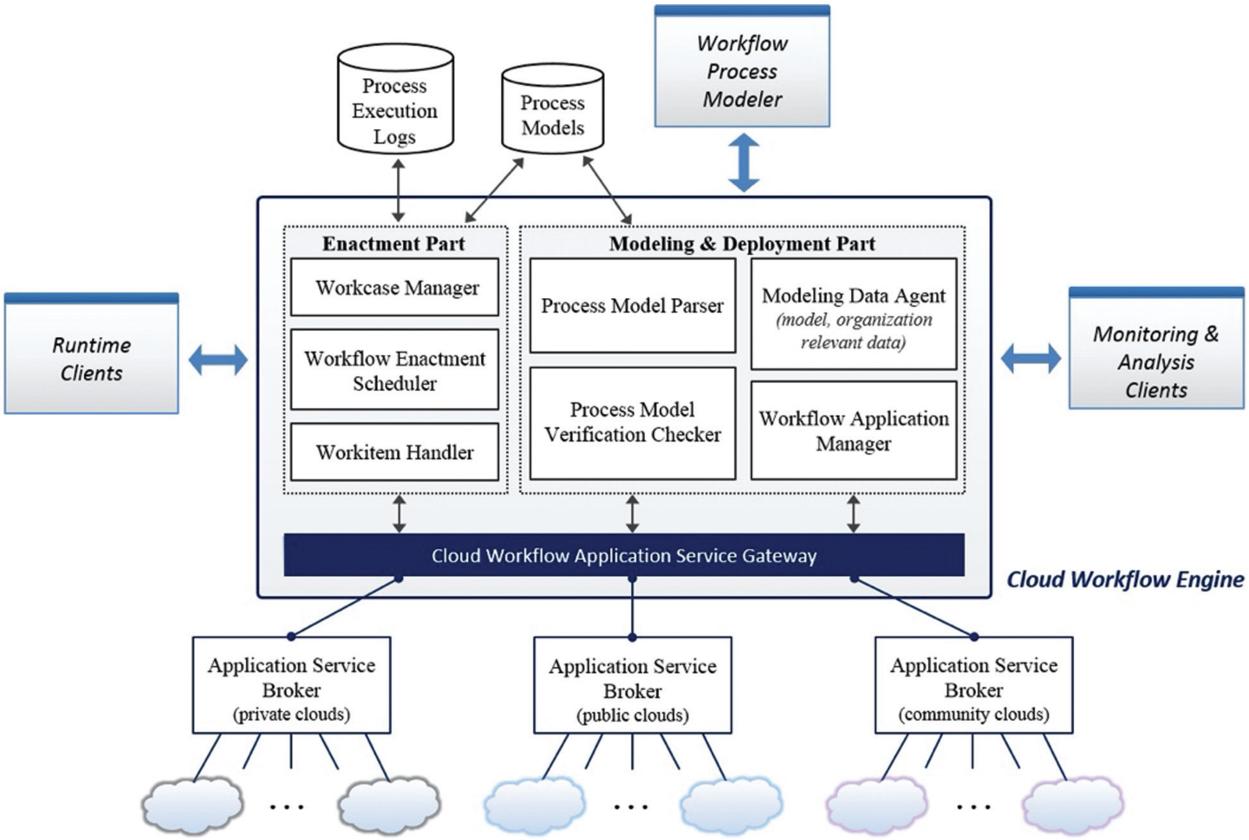
Process Execution Logs
Process Models
Workflow Process Modeler
Enactment Part
Workcase Manager
Workflow Enactment Scheduler
Workitem Handler
Modeling & Deployment Part
Process Model Parser
Modeling Data Agent
(model, organization relevant data)
Process Model Verification Checker
Workflow Application Manager
Runtime Clients
Monitoring & Analysis Clients
Cloud Workflow Application Service Gateway
Cloud Workflow Engine
Application Service Broker (private clouds)
Application Service Broker (public clouds)
Application Service Broker (community clouds)
. . .
. . .
. . .

Figure 8: Architecture of the cloud workflow engine

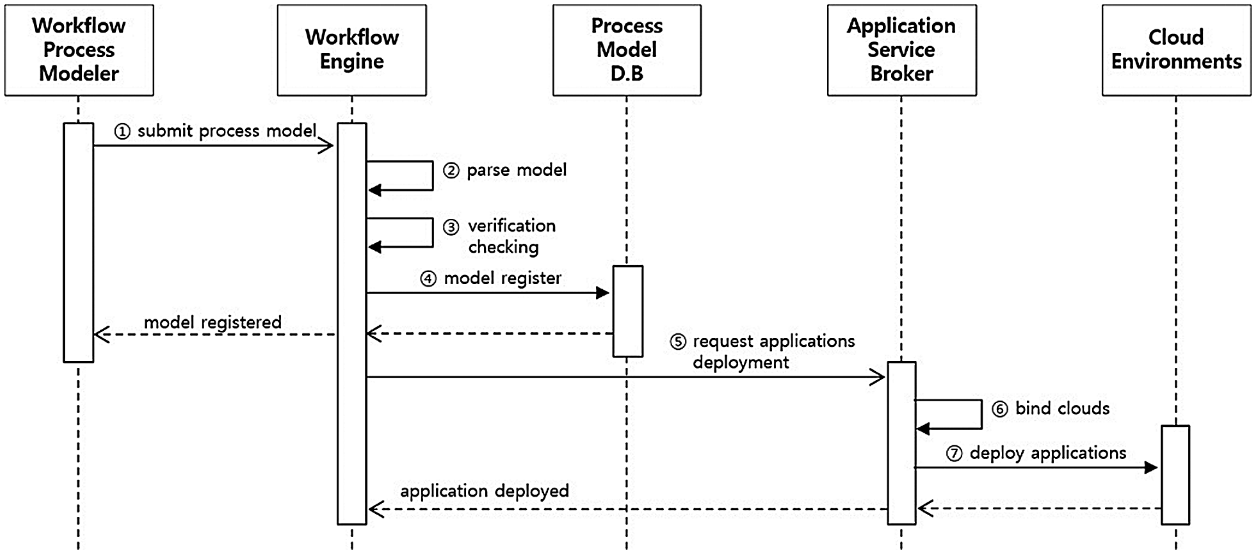

Figure 9: Sequence diagram of process model deployment

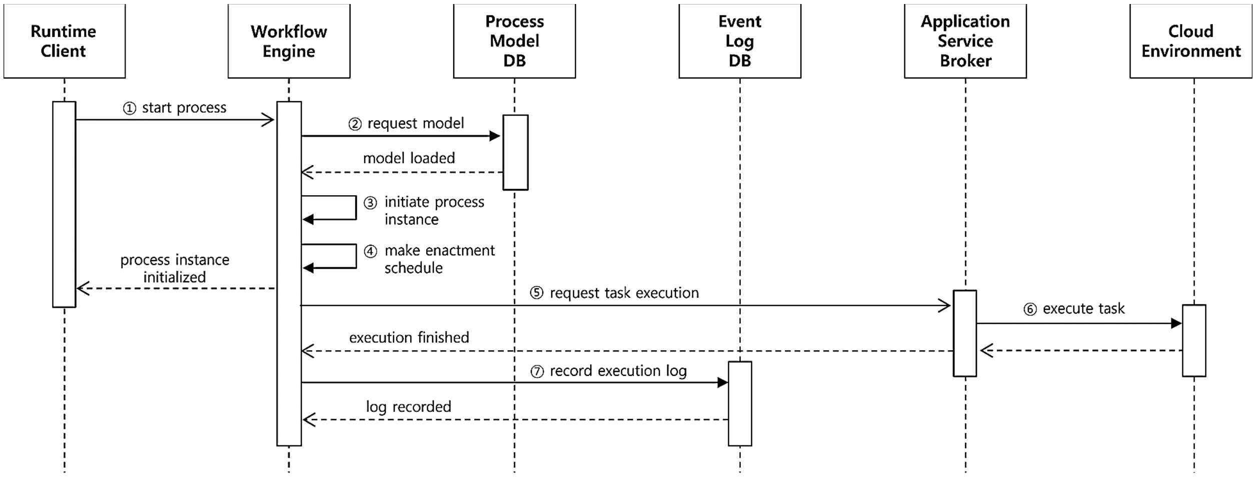

Figure 10: Sequence diagram of task execution

To apply our approach to CWMSs, we presented a preliminary system architecture, largely focused on the workflow engine. According to this architecture, the step of workflow application fragmentation is performed when deploying a process model via communication between the workflow application manager and application service brokers that are responsible for a group of homogeneous cloud environments. Although the presented architecture requires additional elaboration to facilitate the implementation of a CWMS, it suggests that our deployment-based fragmentation approach is viable and can potentially aid in the flexible distribution of large-scale applications for cloud workflow processes.

5 Related Work

The convergence of workflow management systems with distributed computing paradigms and their potential synergies have long been discussed as a major research topic. Beyond past paradigms, such as cluster computing and grid computing [10,11], many researchers and practitioners have been contemplating cloud computing as a promising solution for managing large-scale workflow processes.

Regarding cloud workflow management, its fundamental characteristics were investigated in [4,12–14]. From the viewpoint of BPM, Reference [4] discussed the state of the art of cloud workflow management and its open challenges, such as scheduling [15–17], resource allocation [18], and decentralized coordination. In other hand, [12,13] examined the feasibility, architectural issues, and performance aspects of executing data-intensive scientific workflows in cloud environments. In particular, both studies concentrated on the integration of existing scientific workflow management systems and cloud environments by leveraging open cloud platforms, such as Aneka and OpenNebular.

A few general issues associated with cloud computing have also been examined in conjunction with cloud workflow management. To achieve cost-effective computation, Reference [19] presented a data storage strategy to efficiently store the intermediate data sets of scientific workflows. Similarly, Reference [20] formulated a task-resource mapping method to achieve cost-optimized executions of cloud workflow applications. On the other hand, the issue of the guarantee of security for cloud workflow management has been discussed in a few studies. For instance, to enhance security for large-scale workflow management systems, Reference [21] proposed the application programming interface (API) for specifying the access control based on the wall security model.

The main topic that we concentrated on in this paper was the fragmentation and distribution of workflow applications for cloud workflow processes. References [22,23] investigated the properties of workflow applications that support the execution of human tasks of workflow processes and presented a classification framework for workflow applications based on levels of granularity.

Regarding the fragmentation issue, Reference [24] proposed a process-mining-based fragmentation method. Based on the actual performance of process executions discovered from event logs, they attempted to optimize model fragmentation to reduce the lead time for process execution and, to achieve efficient resource utilization. Similarly, Reference [25] proposed a dynamic fragmentation method to enhance the scalability and elasticity of distributed executions of workflow processes. Both studies attempted to improve overall performance and flexibility, but detailed fragmentation criteria were not considered. Reference [26] presented a fragmentation framework including various fragmentation criteria that facilitate the distribution of workflow models and their applications by considering syntactic aspects (e.g., control-flow and data-flow) and semantic factors (e.g., instance-based, activity-based, and role-based).

In contrast to other studies on the fragmentation issue, our work sets a scope for fragmentation at the activity level so that a group of workflow applications will be partitioned and distributed based on the deployment types attached to each activity. Therefore, the deployment-type-based approach we are pursuing will enable us to configure the sophisticated deployment of workflow applications to satisfy high-level requirements at the stages of designing and implementing a CWMS.

6 Conclusion

In CWMSs, it is crucial to find an appropriate method of distributing the executions of large-scale workflow processes to accomplish the objectives (e.g., scalability and performance of process executions) that are expected when applying cloud computing to the workflow management.

We have presented an approach to workflow application fragmentations based on cloud deployment models. To conceptualize our approach, a series of deployment-model-based operational environments for cloud workflow management was described. For the sake of clarity, we also presented formal descriptions for the workflow application fragmentation of an ICN-based cloud workflow model by specifying activity-wise deployment type information. Through our fragmentation algorithm and the resulting fragment model, we formally verified our approach.

In summary, the main advantages of our approach are as follows: First, the capability to satisfy high-level requirements is the major merit of our approach. Activity-wise deployment allows us to distribute workflow applications with an adequate reflection of desired high-level requirements (e.g., security issues and aligning business policy with IT policy). Second, we can achieve higher degrees of scalability and flexibility in the execution of cloud workflow processes by applying different deployment models. For example, cloud bursting, in which a CWMS outsources computation-intensive tasks within a private cloud to a public cloud, can be enabled by applying the hybrid deployment model. We believe that our approach contributes to the foundation of the design and implementation of CWMSs for large-scale workflow processes executed on multiple clouds. In the future, we will concretize our fragmentation concept to embody a CWMS based on deployment-model-based fragmentation.

Acknowledgement: The authors would like to thank the support of Contents Convergence Software Research Institute and the support of National Research Foundation of Korea.

Funding Statement: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant Number 2020R1A6A1A03040583).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Kim and C. A. Ellis, "ICN-based workflow model and its advances," in *Handbook of Research on Business Process Modeling*, Hershey, PA, USA: Information Science Publishing, pp. 34–54, 2009.

[2] T. Dillon, C. Wu and E. Chang, "Cloud computing: Issues and challenges," in *Proc. AINA*, Perth, Australia, pp. 27–33, 2010.

[3] S. Jajodia, "Interview: Amit Sheth on workflow technology," *IEEE Concurrency*, vol. 6, no. 2, pp. 21–23, 1998.

[4] S. Schulte, C. Janiesch, S. Venugopal, I. Weber and P. Hoenisch, "Elastic business process management: State of the art and open challenges for BPM in the cloud," *Future Generation Computer Systems*, vol. 46, pp. 36–50, 2015.

[5] B. Balis, "Hypermedia workflow: A new approach to data-driven scientific workflows," in *Proc. SCC*, Washington, USA, pp. 100–107, 2012.

[6] E. Deelman, "Grids and clouds: Making workflow applications work in heterogeneous distributed environments," *International Journal of High Performance Computing Applications*, vol. 24, no. 3, pp. 284–298, 2010.

[7] L. Liu, M. Zhang, Y. Lin and L. Qin, "A survey on workflow management and scheduling in cloud computing," in *Proc. CCGrid*, Chicago, USA, pp. 837–846, 2014.

[8] M. La Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling *et al.,* "APROMORE: An advanced process model repository," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7029–7040, 2011.

[9] R. Moreno-Vozmediano, R. S. Montero and I. M. Llorente, "IaaS cloud architecture: From virtualized datacenters to federated cloud infrastructures," *Computer*, vol. 45, no. 12, pp. 65–72, 2012.

[10] J. Yu and R. Buyya, "A taxonomy of workflow management systems for grid computing," *Journal of Grid Computing*, vol. 3, no. 3–4, pp. 171–200, 2005.

[11] K. -H. Kim, "A layered workflow knowledge grid/p2p architecture and its models for future generation workflow systems," *Future Generation Computer Systems*, vol. 23, no. 3, pp. 304–316, 2007.

[12] Y. Zhao, Y. Li, I. Raicu, S. Lu, W. Tian *et al.,* "Enabling scalable scientific workflow management in the cloud," *Future Generation Computer Systems*, vol. 46, pp. 3–16, 2015.

[13] S. Pandey, D. Karunamoorthy and R. Buyya, "Workflow engine for clouds," in *Cloud Computing: Principles and Paradigms*, Hoboken, NJ, USA: John Wiley & Sons, pp. 321–344, 2011.

[14] K. Sun, H. Ahn and K. P. Kim, "A cloud workflow model based on the information control net," *Journal of Internet Computing and Services*, vol. 19, no. 3, pp. 25–33, 2018.

[15] Z. G. Chen, Z. H. Zhan, Y. Lin, Y. J. Gong, T. L. Gu *et al.,* "Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2912–2926, 2018.

[16] Z. J. Wang, Z. H. Zhan, W. J. Yu, Y. Lin, J. Zhang *et al.,* "Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2715–2729, 2019.

[17] K. Devi, D. Paulraj and B. Muthusenthil, "Deep learning based security model for cloud based task scheduling," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 9, pp. 3663–3679, 2020.

[18] A. Abid, M. F. Manzoor, M. S. Farooq, U. Farooq and M. Hussain, "Challenges and issues of resource allocation techniques in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 7, pp. 2815–2839, 2020.

[19] D. Yuan, Y. Yang, X. Liu, G. Zhang and J. Chen, "A data dependency based strategy for intermediate data storage in scientific cloud workflow systems," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 9, pp. 956–976, 2012.

[20] M. Mollajafari and H. S. Shahhoseini, "Cost-optimized GA-based heuristic for scheduling time-constrained workflow applications in infrastructure clouds using an innovative feasibility-assured decoding mechanism," *Journal of Information Science and Engineering*, vol. 32, no. 6, pp. 1541–1560, 2016.

[21] Y. C. Hsiao and G. H. Hwang, "Chinese wall security model for workflow management systems with dynamic security policy," *Journal of Information Science and Engineering*, vol. 29, no. 3, pp. 417–440, 2013.

[22] J. Becker and M. zur Mühlen, "Towards a classification framework for application granularity in workflow management systems," in *Proc. CAiSE*, Heidelberg, Germany, pp. 411–416, 1999.

[23] J. Becker, M. zur Muehlen and M. Gille, "Workflow application architectures: Classification and characteristics of workflow-based information systems," in *Workflow Handbook*, Florida, USA: Lighthouse Point, pp. 39–50, 2002.

[24] S. X. Sun, Q. Zeng and H. Wang, "Process-mining-based workflow model fragmentation for distributed execution," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 41, no. 2, pp. 294–310, 2011.

[25] W. Tan and Y. Fan, "Dynamic workflow model fragmentation for distributed execution," *Computers in Industry*, vol. 58, no. 5, pp. 381–391, 2007.

[26] K. Kim, "A model-driven workflow fragmentation framework for collaborative workflow architectures and systems," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 97–110, 2012.