

Analyzing Scientific Activity Dominancies on Scientific Workflow Models

Minjae Park*, Hyun Ahn**, Kwanghoon Pio Kim**

*BISTel, Inc., Seoul, Korea

**Dept. of Computer Science, KYONGGI UNIVERSITY, Suwon Kyonggido, Korea

mjpark@bistel-inc.com, {hahn, kwang}@kgu.ac.kr

Abstract—In this paper, we propose an algorithmic approach for analyzing (control-flow and data-flow driven) dominancies among scientific activities in a scientific workflow model supported by data intensive experiment procedures and large scale computing environments. Upon the **sciCN**-based scientific workflow model, we explicate the proposed approach from devising an activity dominancy analysis algorithm to exemplifying its application to a pseudo **sciCN**-based scientific workflow model, and finally define the activity dominancy net to represent the output of the algorithm formally and graphically. We expect that the analyzed activity dominancies ought to be helpful not only in the design of load balancing mechanisms for large scale distributed workflow systems, but also in the implementation of exception-handling and recovery mechanisms for concretizing data intensive and flexible scientific workflow systems.

Keywords-scientific workflow; dominancy; load-balancing; verification; exception handling and recovery

I. INTRODUCTION

In recent, what we see happening now in the workflow literature is that the applicability reaches up to the scientific knowledge discovery arena, which is so-called “Scientific Workflow[1][2],” that is able to not only support the whole stage of scientific experiments and simulations but also automate exploratory processes for discovering scientific knowledge involving cycles of observation, hypothesis formation, experiment design and execution. In this paper, we particularly focus on the quality of service issues in scientific workflow technologies, in terms of efficient load-balancing strategies in designing a large scale scientific workflow enactment architecture, and effective exception handling and recovery strategies in implementing a data intensive scientific workflow system. As we know, there exist tens of proposals[3][4][5][6] and rationales[7][8] about defining the concept of scientific workflow models and systems, so far. However, we, in this paper, choose a scientific workflow model that is extended from the conventional information control net methodology[9] to graphically and formally represent the scientific workflow model, which is so-called “Scientific Information Control Net” that is abbreviated to **sciCN**. Based upon the **sciCN**-based scientific workflow model defined in the paper, we conceive an algorithm that is able to analyze dominancies among the activities in a **sciCN**-based scientific workflow model. We would emphasize that the analyzed dominancies can be a reasonable solution not only for an efficient load-balancing strategy but

also for an effective exception-handling and recovery strategy in designing and implementing a large scale and data intensive scientific workflow system, respectively.

In organizing the paper, we start from introducing the formal representation of the **sciCN**-based scientific workflow model. The next section formalizes the activity dominancy analysis algorithm with an operational example. With describing the implications of the proposed activity dominancy for large scale and data intensive scientific workflows, we finalize the paper.

II. SCIENTIFIC INFORMATION CONTROL NET

In this paper, we would extensively adopt the original information control net (ICN) methodology[9] as the scientific workflow modeling methodology, which is so-called **sciCN**-based scientific workflow model. Based upon the scientific information control net, we are going to dig up a possible way of analyzing activity dominancies on a scientific workflow model in the next section. The scientific workflow model needs to be described by a formal representation so that it is able to provide a means to eventually specify the model either in textual language or in database format, or in both. The following definition is the formal representation of the scientific information control net:

Definition 1: sciCN: Scientific Information Control Net of the scientific workflow model. A basic **sciCN** is 10-tuple $\Gamma = (\delta, \gamma, \lambda, \theta, \varepsilon, \pi, \vartheta, \kappa, \mathbf{I}, \mathbf{O})$ over a set $\mathbf{A}$ of activities (including a set of compound/elementary/gateway computational steps), a set $\mathbf{T}$ of transition conditions, a set $\mathbf{R}$ of relevant data, a set $\mathbf{D}$ of invoked data-task applications, a set $\mathbf{Q}$ of invoked query-task applications, a set $\mathbf{P}$ of scientific roles, a set $\mathbf{L}$ of scientific data locations, and a set $\mathbf{C}$ of participants, where $\wp()$ is the Power-set function.

- $\mathbf{I}$ is a finite set of initial input data locations, assumed to be loaded with information by some external **sciCN**s before execution of the corresponding **sciCN**;
- $\mathbf{O}$ is a finite set of final output data locations, perhaps containing information used by some external **sciCN**s after execution of the corresponding **sciCN**;
- $\delta = \delta_i \cup \delta_o$
where, $\delta_o : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of a computational step to its set of (immediate) successors,
and $\delta_i : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a multi-valued mapping function

of a computational step to its set of (immediate) predecessors;

- $\gamma = \gamma_i \cup \gamma_o$
where $\gamma_o : \mathbf{L} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of a computational step to its set of output data locations,
and $\gamma_i : \mathbf{L} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of a computational step to its set of input data locations;
- $\lambda = \lambda_a \cup \lambda_p$
where $\lambda_p : \mathbf{D} \rightarrow \wp(\mathbf{A})$ is a single-valued mapping function of a computational step to its invoked data-task application with multiple threads,
and $\lambda_a : \mathbf{A} \rightarrow \wp(\mathbf{D})$ is a multi-valued mapping function of an invoked data-task application with multiple threads to its set of associated computational steps;
- $\theta = \theta_a \cup \theta_p$
where $\theta_p : \mathbf{Q} \rightarrow \wp(\mathbf{A})$ is a single-valued mapping function of a computational step to its invoked query-task application with multiple threads,
and $\theta_a : \mathbf{A} \rightarrow \wp(\mathbf{Q})$ is a multi-valued mapping function of an invoked query-task application with multiple threads to its set of associated computational steps;
- $\varepsilon = \varepsilon_a \cup \varepsilon_p$
where $\varepsilon_p : \mathbf{P} \rightarrow \wp(\mathbf{A})$ is a single-valued mapping function of a computational step to one of the scientific roles,
and $\varepsilon_a : \mathbf{A} \rightarrow \wp(\mathbf{P})$ is a multi-valued mapping function of a scientific role to its sets of associated computational steps;
- $\pi = \pi_p \cup \pi_c$
where, $\pi_c : \mathbf{C} \rightarrow \wp(\mathbf{P})$ is a multi-valued mapping function of a scientific role to its set of associated scientists,
and $\pi_p : \mathbf{P} \rightarrow \wp(\mathbf{C})$ is a multi-valued mapping function of a scientist to its sets of associated scientific roles;
- $\vartheta = \vartheta_i \cup \vartheta_o$
where $\vartheta_o : \mathbf{R} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of a computational step to its set of output relevant data,
and $\vartheta_i : \mathbf{R} \rightarrow \wp(\mathbf{A})$ is a multi-valued mapping function of a computational step to its set of input relevant data;
- $\kappa = \kappa_i \cup \kappa_o$
where $\kappa_i(\alpha) : \text{sets of control-transition conditions, } \mathbf{T}$,
on each arc, $(\delta_i(\alpha), \alpha)$, $\alpha \in \mathbf{A}$;
and $\kappa_o(\alpha) : \text{sets of control-transition conditions, } \mathbf{T}$,
on each arc, $(\alpha, \delta_o(\alpha))$, $\alpha \in \mathbf{A}$;
where the set $\mathbf{T} = \{\text{default, or(conditions), and(conditions)}$

In terms of the graphical representation, we principally adopt the extended notations of scientific information control net. There are possible primitive computation-flows and their related gateway G-type activities. That is, the conjunctive (or parallel) computation-flow type with a pair of conjunctive gateway G-type activities, split-AND and join-AND, is represented by solid dots($\bullet$), meanwhile the disjunctive (or decision) computation-flow type with a pair of disjunctive gateway G-type activities, split-OR and join-OR, is represented by hollow dots($\circ$). Also, the iterative (loop) computation-flow type with a pair of loop gateway G-type activities, split-LOOP and join-LOOP, is represented by double hollow dots.

Besides, in order to be syntactically safe, it is very important for these gateway G-type activities to keep the structured properties—proper nesting and matched pair properties. Therefore, in specifying a scientific information control net, not only each of the gateway G-type activities always keeps matched pair with split and join types, but also multiple sets of the gateway G-type activities keep in a properly nested pattern. Figure 1 depicts the graphical representation of a pseudo scientific information control net complying with the graphical notations and the rule of structured formations.

III. DOMINANCY ANALYSIS

In this section, we expatiate the conceptual meanings and theoretical rationales of dominancy, and formalize its analysis algorithm that is able to automatically generate a scientific activity dominancy net (abbreviated to **scADN**), which is a mathematical notation for ultimately representing the scientific activity dominancy model formally as well as graphically, from a **scICN**-based scientific workflow model defined in the previous section.

A. Dominancy

In this section, we elucidate the basic definition of scientific activity dominancy based upon very important functional relationships, “*strongly forward dominator*,” and “*immediate forward dominator*,” to decisively determine the dominancy between two computational steps stored in u and v , respectively. These functional relationships are based upon the following concepts and definitions, “*walk*,” and “*dominance*,” which is originally defined in [10], and we have revised them so as to fit into the basic principle of **scICN**-based scientific workflow model.

Definition 2: walk, W , in a **scICN**-based scientific workflow model. In a basic **scICN**, $\Gamma = (\delta, \gamma, \lambda, \theta, \varepsilon, \pi, \vartheta, \kappa, \mathbf{I}, \mathbf{O})$, a *walk*, W , is a sequence of computational steps, $\alpha_1, \alpha_2, \dots, \alpha_n$, such that $n \geq 1$ and $\alpha_{i+1} \in \delta_o(\alpha_i)$ for $i = 1, 2, \dots, n-1$. The length of a *walk*, $W = \alpha_1\alpha_2\dots\alpha_n$, denoted by $|W|$, is n , the number of computational step occurrences in W .

Note that a *walk* of length zero has no any computational step occurrences; such a *walk* is called empty. A nonempty *walk*, whose first computational step, u , and whose last computational step, v , is called a $u - v$ *walk*. If $Y = y_1y_2\dots y_m$ and $X = x_1x_2\dots x_n$ are *walks* such that Y and X are nonempty and y_m is adjacent to x_1 , then the concatenation of Y and X , denoted by YX , is the *walk*, $y_1y_2\dots y_mx_1x_2\dots x_n$.

Definition 3: Dominance in a **scICN**-based scientific workflow model. Suppose that a basic **scICN**, $\Gamma = (\delta, \gamma, \lambda, \theta, \varepsilon, \pi, \vartheta, \kappa, \mathbf{I}, \mathbf{O})$, satisfies each of the following conditions:

- 1) Γ contains two distinguished event steps: the initial event step, α_I , implying that $\delta_i(\alpha_I)$ is $\emptyset$, and the final event step, α_F , implying that $\delta_o(\alpha_F)$ is $\emptyset$;
- 2) Every computational step of Γ occurs on somewhere of $\alpha_I - \alpha_F$ *walk*.

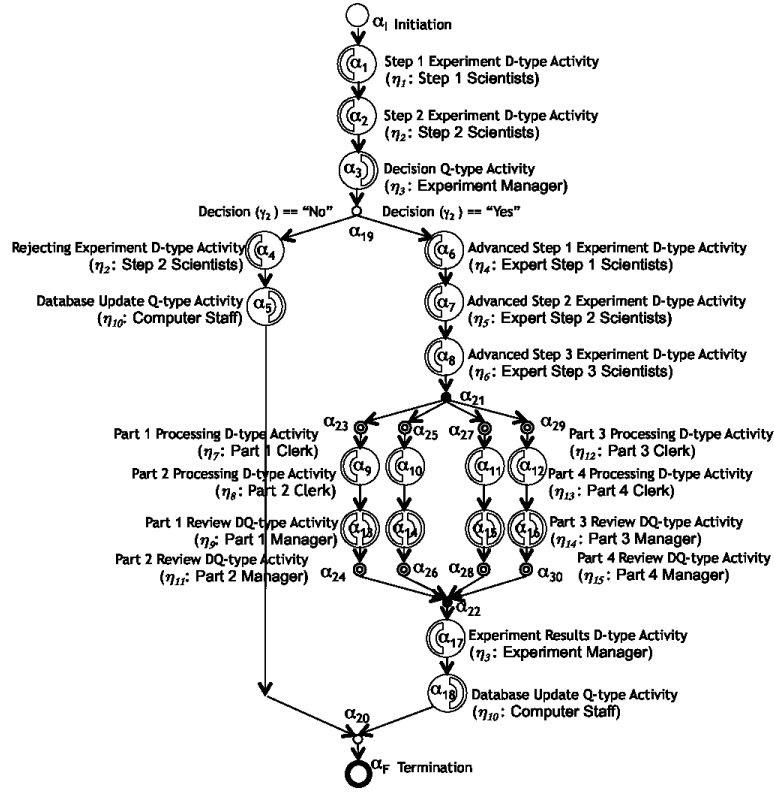


Fig. 1. A Pseudo Scientific Information Control Net

- Let Γ be a sICN-based scientific workflow model. A computational step $u \in A$ **forward dominates** a computational step $v \in A$ iff every $v - \alpha_F$ walk in Γ contains u ; u is **properly forward dominator** of v iff $u \neq v$ and u forward dominates v .
- Let Γ be a sICN-based scientific workflow model. A computational step $u \in A$ is **strongly forward dominator** of a computational step $v \in A$ iff u is properly forward dominator of v and there is an integer $k \geq 1$ such that every walk in Γ beginning with v and its length $\geq k$ contains u .
- Let Γ be a sICN-based scientific workflow model. The **immediate forward dominator** of a computational step $\alpha \in \{A\} - \alpha_F$, denoted $\text{ifd}(\alpha)$, is the computational step that is the first proper forward dominator of α occurring in every $\alpha - \alpha_F$ walk in Γ .
- Let Γ be a sICN-based scientific workflow model. A computational step $u \in A$ **backward dominates** a computational step $v \in A$ iff every $v - \alpha_I$ walk in Γ contains u ; u is **properly backward dominator** of v iff $u \neq v$ and u backward dominates v .
- Let Γ be a sICN-based scientific workflow model. The **immediate backward dominator** of $\alpha \in \{A\} - \alpha_I$, denoted $\text{ibd}(\alpha)$, is the computational step that is the first proper backward dominator of α occurring in every $\alpha - \alpha_I$ walk in Γ .

B. Scientific Activity Dominancy Net

In order to analyze the concept of dominancy on a sICN-based scientific workflow model, we would conceive a graph model, and we dub it “Scientific Activity Dominancy Net (scADN).” The net is built by analyzing the four types of control transitions depicted in Figure ???. That is, the formal representation of the scientific activity dominancy net uses the primitive operations (*walk* and *dominance*) firstly introduced by [10], which is operating on the four types of control

transitions, and a set of activity dominancy relationships can be extracted from a sICN-based scientific workflow model. And, the extracted activity dominancy relationships (knowledge) are used for formally describing the scientific activity dominancy net as followings:

Definition 4: Scientific Activity Dominancy Net of a sICN-based scientific workflow model. A scientific activity dominancy net is formally defined as $\Omega = (\varphi, \xi, I, O)$ over a set $\{\{A\} - \{G\text{-type}\}\}$ of D-type, Q-type, and DQ-type computational steps and a set T of transition conditions, where

- $\varphi = \varphi_{src} \cup \varphi_{sfd}$
where $\varphi_{sfd} : A \rightarrow \wp(A)$ is a multi-valued mapping of a computational step, $\alpha \in A$, to a set of other computational steps, each of which that is **strongly forward dominating** the computational step, α , which implies that the computational step is strongly effected by the mapped computational step(s),
and $\varphi_{src} : A \rightarrow A$ is a single-valued mapping of a computational step, α_{sfd} , to the computational step, α_{src} , that is **strongly forward dominated** by α_{sfd} , which implies that the results of the computational step, α_{src} , can be effected, computed, or updated by the results of the mapped computational step, α_{sfd} ;
- $\xi = \xi_{src} \cup \xi_{sfd}$
where ξ_{src} : a set of transition conditions, $\tau \in T$, on each of arcs, $(\alpha, \varphi_{src}(\alpha))$;
and ξ_{sfd} : a set of transition conditions, $\tau \in T$, on a single arc, $(\alpha, \varphi_{sfd}(\alpha))$, where $\alpha \in A$;
- I is a finite set of initial input repositories including data

The Scientific Activity Dominance Net Construction Algorithm:

PROCEDURE scACTIVITY-DOMINANCY()

```

1: Input A sclCN-based Scientific Workflow Model,  $\Gamma = (\delta, \gamma, \lambda, \theta, \varepsilon, \pi, \vartheta, \kappa, \mathbf{I}, \mathbf{O})$ ;
2: Output A Scientific Activity Dependency Net,  $\Omega = (\varphi, \xi, I, O)$ ;
3: Initialize
4:    $T \leftarrow \emptyset$ ; /*  $T$  holds the traversed computational steps;  $T$  is a global variable. */
5: PROCEDURE scACTIVITY-DOMINANCY( In  $s \leftarrow \delta_o(\alpha_I)$ ,  $f \leftarrow \alpha_F$ ; out  $new$ ; ) /* recursive procedure */
6: BEGIN
7:    $v \leftarrow s$ ;  $\varphi_{src}(v) \leftarrow \delta_i(s)$ ;  $T \leftarrow T \cup \{s\}$ ;
8:    $D \leftarrow \delta_o(v)$ ;
9:   FOR ( $\forall u \in D$ ) DO
10:    IF ( $u \notin T$  and  $u \neq f$ )
11:      Then do
12:        IF ( $u = \text{SFD}(v)$ ) /*  $u$  is a strongly forward-dominator of  $v$ ? */
13:          Then do
14:             $\varphi_{sfd}(v) \leftarrow u$ ;  $\varphi_{src}(u) \leftarrow v$ ; /* forming a dominance with a computational step! */
15:             $T \leftarrow T \cup \{u\}$ ;
16:             $D \leftarrow D - \{u\}$ ;
17:          End
18:        Else do
19:          Call PROCEDURE scACTIVITY-DOMINANCY( In  $s \leftarrow u$ ,  $f \leftarrow \text{'join'}$ ; out  $new$  );
20:           $D \leftarrow D - \{u\}$ ;
21:           $u \leftarrow new$ ;
22:           $\varphi_{sfd}(v) \leftarrow u$ ;  $\varphi_{src}(u) \leftarrow v$ ; /* forming a level of dominance! */
23:        End
24:      END IF;
25:    End
26:  END IF;
27:  IF ( $D = \emptyset$ ) /*  $D$  is empty? */
28:    Then do  $D \leftarrow \delta_o(u)$ ; End
29:  Else do
30:    IF ( $u = f$ )
31:      Then do
32:         $\varphi_{sfd}(v) \leftarrow u$ ;  $\varphi_{src}(u) \leftarrow v$ ; /* forming a level of dominance! */
33:         $T \leftarrow T \cup \{u\}$ ;
34:         $D \leftarrow D - \{u\}$ ;
35:      End
36:    END IF;
37:  End;
38: END IF;
39: END FOR;
40: return ( $new \leftarrow u$ ; )
41: END PROCEDURE

```

Fig. 2. The Scientific Activity Dominance Net Construction Algorithm

locations and relevant data;

- O is a finite set of final output repositories including data locations and relevant data;

For the formal definition, we need the concepts of *strongly forward dominating* and *strongly forward dominated*, which are based on the terminology, *walk* and *dominance*, defined in the previous subsection. Emphasize that the computational steps involved in the dominance relationships are only those classes of D-type, Q-type, and DQ-type computational steps.

The graphical representation of the scientific activity dominance net is a shape of the directed tree structure. In matching the scientific activity dominance net diagram to its formal definition, every node in the diagram is corresponding to each of the computational steps excluding G-type gateways and initiation/termination events in its sclCN-based scientific workflow model, and a solid arrow is to depict a strongly forward dominance relationship between two associated computational steps, which semantically implies that the destination node of the arrow is strongly forward dominating the source node of it. Additionally, each transition condition is positioned on

its associated solid arrowed edges. Of course, the scientific activity dominance net may be extensible to accommodate those compound computational steps, like subprocess.

Next, how can we build a scientific activity dominance net from a sclCN-based scientific workflow model? Definitely we need a dominance analysis algorithm for this. By using those special operations, such as *walk*, *dominance* with *strongly forward dominator* introduced in [10], we conceive the algorithm and expatiate it with its operational example in the next subsection.

C. Dominance Analysis Algorithm

This section expatiates a scientific activity dominance analysis algorithm that is able to construct a scientific activity dominance net from a sclCN-based scientific workflow model. The algorithm, as we see in Figure 2, is a recursive procedure named “scACTIVITY-DOMINANCY()”, and it uses a very important functional concept, “*strongly forward dominator*.” to decisively determine the dominance relationship between two computational steps stored in u and v , respectively. These

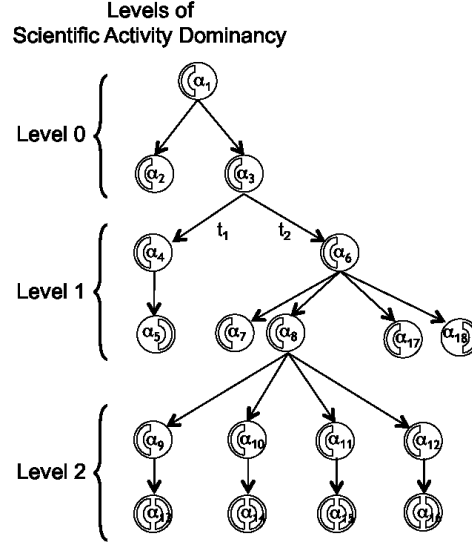


Fig. 3. Scientific Activity Dominancy Net from the Pseudo Scientific Workflow Model

functional concept are based upon the following concepts and definitions, which is originally defined in [10], and we have revised so as to fit into the basic principle of the *scICN*-based scientific workflow model.

Based upon the basic terms, *walk* and *dominance*, and their extended terms, *forward dominators* and *backward dominators*, we construct a series of dominator-determination functions, $fd()$, $bd()$, $ifd()$, $ibf()$ and $sfd()$ functions; we simply define them in the paper as the followings:

- The function, $fd(v \in A)$, returns a set of *forward dominators* of $v \in A$.
- The function, $bd(v \in A)$, returns a set of *backward dominators* of $v \in A$.
- The functions, $ifd(v \in A)$ and $ibf(v \in A)$, return an *immediate forward dominator* and an *immediate backward dominator* of its input parameter's computational step, $v \in A$, respectively.
- The function, $sfd(v \in A)$, returns a set of *strongly forward dominators* of $v \in A$.

Based upon these basic dominator-determination functions, the algorithm, presented in Figure 2, is able to construct a scientific activity dominancy net (*scADN*) from a *scICN*-based scientific workflow model. Additionally, the time complexity of the algorithm is $O(n^2)$, where n is the number of computational steps in a *scICN*-based scientific workflow model, because the function, $SFD()$, for deciding the strongly forward dominating relationship between two computational steps can be computed in $O(n)$, and the recursive procedure itself can be computed in $O(n)$. Therefore, the overall time complexity is $O(n^2)$. As an operational example, Figure 3 shows a scientific activity dominancy net in a graphical representation, which is constructed from the psuedi *scICN*-based scientific workflow model of Figure 1.

Conclusively, the paper conceives the *scADN* construction algorithm to automatically generate a scientific activity dominancy net of the corresponding *scICN*-based scientific workflow model, which can be used by any type of self-recovery mechanisms that are able to recover an error-involved scientific procedure from the exceptional and risky situations.

That is, the scientific activity dominancy knowledge analyzed by the algorithm eventually provides the proper point of roll-back operations to the self-recovery mechanism, which is pointing to the computational step starting a roll-back among the previous performed computational steps on the corresponding *scICN*-based scientific workflow model.

IV. CONCLUSIONS

So far, this paper has introduced the concept of scientific activity dominancies. Based upon these formal and graphical representations of the *scICN*-based scientific workflow model, we finally has conceived the algorithm that is able to construct a scientific activity dominancy net from a *scICN*-based scientific workflow model. As stated previously, in resolving the QoS issue, it is very important for the scientific workflow system to provide an autonomous error-detection functionality on its running exceptions and very safe self-recovery mechanisms, as well, for the exceptional error situations. Therefore, we are certainly sure that the scientific activity dominancy analysis algorithm proposed in the paper is able to discover the scientific activity dominancy knowledge that will be eventually used for realizing the self-recovery mechanisms. In recent, the literature needs various, advanced, and specialized autonomous error-detection and self-recoverable features and techniques. We so strongly believe that this work might be one of those impeccable attempts and pioneering contributions for improving and advancing the large scale and data intensive and scientific workflow management technology.

ACKNOWLEDGEMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (Grant No. 2012006971), Republic of Korea.

REFERENCES

- [1] A. Barker and J. van Hemert, "Scientific workflow: A survey and research directions," *Lecture Note in Computer Science*, vol. 4967, pp. 746–753, 2008.
- [2] J. Qin and T. Fahringer, *SCIENTIFIC WORKFLOWS: Programming, Optimization, and Synthesis with ASKALON and AWDL*. Springer, 2012.
- [3] N. H. T. M. Anne H.H. Ngu, Shawn Bowers and T. Critchlow, "Flexible scientific workflow modeling using frames, templates, and dynamic embedding," *Lecture Note in Computer Science*, vol. 5069, pp. 566–572, 2008.
- [4] e. a. Bertram Ludascher, "Scientific workflow management and the kepler system," *Journal of Concurrency and Computation: Practice & Experience*, vol. 18, no. 10, pp. 1039–1065, August 2006.
- [5] Z. Y. Ping Yang and S. Lu, "Formal modeling and analysis of scientific workflows using hierarchical state machines," in *Proceedings of IEEE International Conference on e-Science and Grid Computing*. IEEE, December 2007, pp. 619–626.
- [6] D. d. O. Eduardo Ogasawara and P. Valduriez, "An algebraic approach for data-centric scientific workflows," in *Proceedings of the 37th International Conference on Very Large Data Bases*. VLDB Endowment, September 2011, pp. 619–626.
- [7] A. G. Ustun Yildiz and A. H. Ngu, "Business versus scientific workflow: A comparative study," *University of California Davis, Department of Computer Science, Research Report, Project DAKS*, vol. n 2009-3, pp. 1–18, 2009.
- [8] e. a. Yolanda Gil, "Examining the challenges of scientific workflows," *IEEE Computer*, vol. 40, no. 12, pp. 24–32, December 2007.
- [9] C. A. Ellis, "Information control nets: A mathematical model of information flow," in *Proceedings of Conference on Simulation, Modeling, and Measurement of Computer Systems*. ACM, 1979, pp. 225–240.
- [10] A. Podgurski and L. A. Clarke, "A formal model of program dependencies and its implications for software testing, debugging, and maintenance," *IEEE Trans. on Software Engineering*, vol. 16, no. 9, 1990.



Kwanghoon Pio Kim Kwanghoon Pio Kim is a full professor of computer science department and the founder and supervisor of the collaboration technology research laboratory at Kyonggi University, South Korea. He received B.S. degree in computer science from Kyonggi University in 1984. And he received M.S. degree in computer science from Chungang University in 1986. He also received his M.S. and Ph.D. degree from the computer science department of University of Colorado at Boulder, in 1994 and 1998, respectively.



Minjae Park Minjae Park is a senior member of research staff at the solution R&D research center of BISTel, Inc., South Korea. He received B.S., M.S., and Ph.D. degrees in computer science from Kyonggi University in 2004, 2006, and 2009, respectively. His research interests include groupware, workflow systems, BPM, CSCW, collaboration theory, process warehousing and mining, workflow-supported social networks discovery and analysis, and process-aware factory automation systems.



Hyun Ahn Hyun Ahn is a full-time Ph.D. student of computer science department and a graduate member of the collaboration technology research laboratory at Kyonggi University, South Korea. He received B.S. and M.S. degrees in computer science from Kyonggi University in 2010 and 2012, respectively. His research interests include workflow systems, BPM, scientific workflow systems, workflow-supported social and affiliation networks discovery, analysis, and visualization.