A Reference Design for Logical Work process Administration Frameworks and the VIEW SOA Arrangement

[1]Deepa B [2]Naeem N
[1]GUEST FACULTY AMUV TORVI [2]GUEST FACULTY AMUV TORVI
[1]DEPARMENT OF COMPUTER SCIENCE
[1]AMUV TORVI ,INDIA

Abstract: Dynamic Logical work processes have as of late developed as another worldview for researchers to formalize and structure unpredictable and conveyed logical procedures to empower and quicken numerous logical revelations. As opposed to business work processes, which are commonly control stream arranged, logical work processes have a tendency to be dataflow situated, presenting another arrangement of prerequisites for framework advancement. These necessities request another compositional outline for logical work process administration frameworks (SWFMSs). Albeit a few SWFMSs have been created that give much understanding to future innovative work, an investigation from a compositional viewpoint is as yet absent. The principle commitments of this paper are:
1) In light of a far reaching study of the writing and recognizable proof of key prerequisites for SWFMSs, we propose the main reference engineering for SWFMSs;
2) As per the reference design, we additionally propose an administration situated design for VIEW (a VIsual Logical Work process administration framework);
3)We actualized VIEW to approve the attainability of the proposed models; and
4) we exhibit a VIEW-based logical work process application framework (SWFAS), called FiberFlow, to feature the utilization of our VIEW framework.

I. INTRODUCTION

Logical work processes have as of late developed as another worldview for researchers to coordinate, structure, and arrange an extensive variety of neighborhood and remote heterogeneous administrations and programming instruments into complex logical procedures to empower and quicken numerous logical disclosures [1]. A logical work process is the modernized assistance or robotization of a logical procedure, in entire or part, which more often than not streamlines an accumulation of logical undertakings with information stations and dataflow develops to mechanize information calculation and examination to empower and quicken logical revelation. A logical work process administration framework (SWFMS) is a framework that totally characterizes, alters, oversees, screens, and executes logical work processes through the execution of logical assignments whose execution arrange is driven by a modernized portrayal of the work process rationale. The outline of a reference design at a fitting level of reflection that tends to building business work process administration frameworks (BWFMSs) [4], [5], [6], [7], [8]. Notwithstanding, these reference models are not reasonable for SWFMSs as business work processes and logical work processes have diverse objectives. While the objective of business work processes is to diminish HR (and different expenses) and increment income, the objective of logical work processes is to lessen both human and calculation costs and quicken the speed of transforming a lot of bits and bytes into learning and disclosure. In addition, business work processes are normally control stream arranged, while logical work processes have a tendency to be dataflow situated, presenting another arrangement of prerequisites and difficulties for framework advancement, from the help of serious client collaboration and representation, adjustable and extensible GUI, reproducibility, top of the line registering, to heterogeneous information, programming device, and administration. While a few SWFMSs [9], [10], [11], [12], [13], [14] have been produced amid the previous couple of years, which give much understanding to future innovative work, an engineering reference that can give an abnormal state association of subsystems and their collaborations in a SWFMS is missing. The cutting edge is still specially appointed in logical work process outline, determination, advancement, execution, and provenance following, and so forth. To begin with, every framework utilizes an exclusive work process dialect, whose semantics has not yet been completely examined and formalized. Second, every framework has either no unequivocal architectural outline or the design is exclusive and confined enormously by the heritage framework that the SWFMS is based upon. For instance, Kepler is based on the Ptolemy II framework, and subsequently, each new necessity that is required by a SWFMS depends on expansions to the design of Ptolemy. Pegasus, then again, is based upon Condor and Dogman by including another work process mapper the prerequisites for SWFMSs is basic and testing. The Work process Administration Coalition (WfMC) proposed a reference engineering for business work processes [2] in 1995. From that point forward, the reference engineering and its variations [3] have been generally received in the advancement of various best of these two frameworks. Third, every one of these frameworks have distinctive provenance models, not just as far as what provenance data ought to be recorded, yet in addition as far as portrayal, stockpiling, and questioning models. We expect that the accessibility of such reference engineering can give a premise to correlation between various frameworks and a direction for the structural plan of a SWFMS in a particular logical space.

To address this issue,
1.we propose the main reference engineering for SWFMSs in view of a complete overview of the writing and recognizable proof of key necessities;
2.according to the proposed reference design, we additionally propose an administration arranged engineering for the VIEW framework. Utilizing SOA [15], VIEW comprises of six approximately coupled administration parts, every one of which relates to an utilitarian component that is distinguished in the reference engineering, whose usefulness is uncovered as an Internet benefit;
3.we actualized the VIEW framework to approve the plausibility of the proposed models; and
4.we present a VIEW-based logical work process application framework (SWFAS), called FiberFlow, to evil spirit strate the abilities of VIEW in help of client connection escalated, representation concentrated, and register serious logical work processes in a heterogenerous and circulated figuring condition.
Whatever is left of the paper is sorted out as takes after: Area 2 recognizes seven key compositional prerequisites for SWFMSs. Because of these prerequisites, Segment 3 proposes our reference engineering for SWFMSs. As indicated by the reference engineering, Segment 4 additionally proposes an administration situated design for VIEW (a Visual Logical Work process administration framework). Area 5 presents design administration for VIEW subsystems, and Segment 6 exhibits the FiberFlow framework utilizing Perspective as its basic SWFMS. From that point onward, we assess five agent SWFMSs utilizing the proposed reference design in Segment 7, trailed by related work in Area 8. At last, Area 9 closes the paper and remarks on future work.

II. ADVANTAGES

While the rise of SOA as an engineering worldview gives numerous advantages to conveyed processing [23], we distinguish the accompanying points of interest of utilizing SOA particularly for the advancement of a SWFMS:
1. Service free coupling: Administration free coupling smaller than expected mazes the conditions among subsystems of a SWFMS by the meanings of an arrangement of dialect and stage autonomous interfaces. In our proposed engineering, every subsystem's usefulness is ex-acted like an Internet benefit. Subsequently, a SWFMS can be made on request from different subsystems gave by various gatherings as Web administrations. One can likewise effectively change starting with one administration then onto the next for every subsystem. For instance, there might be a few provenance administration administrations accessible, and utilizing SOA, one can utilize and switch any provenance administration benefit on interest for a particular SWFMS.
2. Service deliberation and self-sufficiency: An Internet benefit gives a theoretical interface that is free from its usage. What's more, each Internet benefit is self-governing as in a specialist co-op has the control over the application rationale that the Internet benefit embodies. Therefore, a specialist co-op can progressively change the implementation and organization condition of an Internet benefit for a subsystem of a SWFMS with no downtime for the SWFMS as long thusly changes don't influence the characterized interface. Such self-sufficiency additionally significantly encourages the administration of the improvement and advancement of the entire framework.
3. Service reusability: As every subsystem of a SWFMS turns into a uniform figuring unit with standard interface depictions and all inclusive openness through standard correspondence conventions, it can be reused crosswise over different SWFMSs, even simultaneously utilized by both neighbourhood SWFMSs and different SWFMSs over the Web.
4. Service discoverability: As every subsystem of a SWFMS is executed as an Internet benefit that is enhanced with a semantic portrayal, one can enrol the administration in some open administration registries. Subsequently, a subsystem ends up discoverable and can be chosen and utilized by different SWFMSs on request.
5. Service interoperability: Administration interoperability is empowered by the open measures of messages and correspondence conventions for Web administrations, which are bolstered by an expansive assortment of IT industry and the Internet Administrations Interoperability Association (WS-I). Utilizing Web benefits, the interoperability crosswise over different SWFMSs (necessity R7: level 3) can be significantly progressed.
Likewise, a database in the VIEW framework could particularly serve one administration segment, or various reinforcement benefit segments, or even be shared by several benefit segments of the VIEW Bit. To help database failover, one serving database could likewise setup a few mirror databases on circulated machines, and every one of them is kept synchronized with the serving database, so once this database fizzles, its administration component(s) can change to some other mirror databases.

To empower an on-request VIEW Portion, the VIEW framework gives an administration part, called VIEW Arrangement Administration, to deal with the setups of all VIEW Piece benefit segments and their serving databases. In the first place, the Setup Administration GUI inserted in the VIEW Workbench enables researchers to enroll all conveyed benefit segments for the VIEW Piece and their serving database(s). The administration segments for a VIEW Part subsystem utilize the same WSDL to portray their normal interfaces. Second, when the VIEW framework is received by a particular SWFAS, a format of the VIEW framework can be created on request by arranging each VIEW Piece benefit segment and its database(s), which are as of now enlisted in the Setup Administration. Third, such format of the framework can conjure the picked administrations amid the runtime. Once an administration of the layout is inaccessible, design administration will conjure another elective administration. As the elective administration and inaccessible administration share the same database(s) and repositories in their subsystem stockpiling layer, work process run and errand run status are as yet legitimate, which makes it conceivable to continue the work process execution beginning from the administration downtime.

III. OVERALL DESIGN AND SUB-SYSTEM MODELS

The general design of VIEW in Fig. 2 comprises of six administration parts that relate to the primary practical subsystems proposed in the reference design. Other than Workbench, the interface for each administration part is characterized and depicted by WSDL: IWE; IWM ; IT M ; IP M , and IDPM for the interface of the Work process Motor, the Work process Screen, the Errand Administrator, the Provenance Supervisor, and the Information Item Chief, separately, which includes the VIEW Bit. In the accompanying, we concentrate our exchange on the engineering points of interest of the VIEW Part. Workbench. The Workbench subsystem executes the elements of work process plan, introduction, and representation distinguished at the Introduction Layer in the reference design. At present, it comprises of five components (see Fig. 3a): Work process Fashioner, Provenance Adventurer, and the GUIs for the VIEW Bit. Work process Planner gives a researcher amicable GUI to the outline and alteration of logical work processes. A researcher can move enrolled errands and information items into the outline board and connection them to each other utilizing different dataflow and control stream builds. Work process Creator is upheld by our proposed work process detail dialect, called SWL to characterize a logical work process, as indicated by the VIEW Work process Show, which underpins various levelled (settled) logical work processes. Work process definitions in the Work process Planner are spared in XML documents into a Neighbourhood Work process Archive. A work process definition in the VIEW Workbench comprises of three sections:

1)A work process determination to store the consistent structure and its constituent parts; 2) work process run parameters to store all parameters for each assignment run; and 3) a work process format to store the graphical format of the logical work process that is required to show the work process in the Work process Configuration Board. The initial two sections are required for the execution of a work process run, and the last part is to show and control a logical work process in the VIEW Configuration Board.

Provenance Pilgrim empowers a client to peruse and envision logical work process provenance metadata. More-finished, together with the GUI for the Information Item Supervisor, one can show and picture different information items from straightforward information esteems and plain messages to complex information writes. The VIEW Workbench bolsters Windows-based UIs for the VIEW Portion while reusing the same benefit segments. These researcher cordial GUIs communicate with subsystems by means of ITM , IWM , IPM , and IDP M , separately This prompts the design adaptability to enable researchers to alter their own particular GUIs for every specific SWFAS, along these lines fulfilling prerequisite R1. Work process Motor. The engineering of the Work process Motor subsystem is appeared in Fig. 3b. Based on Scheduler, the Work process Motor comprises of six useful modules: Scheduler, Interpreter, Control flow Administration, Dataflow Administration, Work process Status Administration, and Provenance Gatherer.

To begin with, Interpreter gives a mapping plan to translating a work process detail into a streamlined interior executable work process portrayal. Work process definitions conveyed from Work process Planner are spared into the Work process Definition Store by means of IWE. A work process definition in Workbench's Work process Storehouse ought to be predictable with the variant in Work process Definition Repository amid work process execution. Second, the detachment of control flow and dataflow administration from work process booking significantly enhances the extensibility of the VIEW Work process Show since the presentation of extra control stream or dataflow builds can be accomplished by overhauling their individual modules without changing different modules. Third, as Scheduler can bolster multithread preparing, it can instate and keep up various work process runs all the while, Work process Status Stockpiling gives an establishment to workflow run observing and disappointment taking care of (necessity R6). At long last, Provenance Authority is in charge of gathering all provenance data and putting away them into Provenance Chief by means of IP M . Since the VIEW Work process Motor backings an open and extensible SWL and is approximately combined with different subsystems, the work processes/sub workflows outlined by different SWFMSs can straightforwardly demand to and summoned by the VIEW Work process Motor by means of the Internet benefit correspondence and conjuring. Consequently, the sharing and mapping between the VIEW Work process Motor and different SWFMSs can be significantly encouraged 2).As opposed to BWFMSs that generally oversee control-stream arranged work processes, in which the request of assignment execution is unequivocally

determined by control stream builds, for example, consecutive, contingent, and circle, the VIEW Work process Motor is produced for dataflow-driven logical work processes. Therefore, the accessibility of information for an undertaking starts its execution, and the development of information by means of information channels decides the execution request of a work process.

Work process Screen. Our present execution of the Work process Screen utilizes a Distribute/Buy in demonstrate [19].

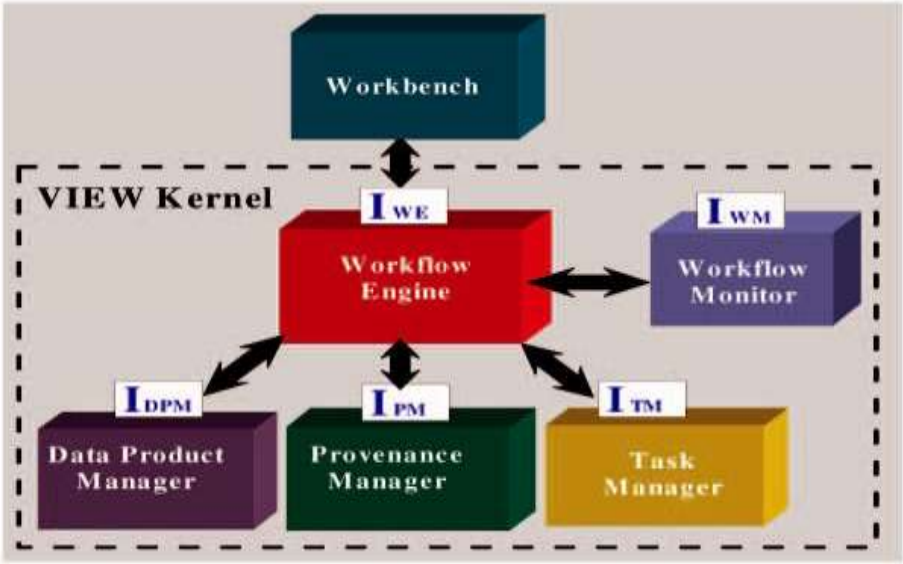

Fig. 2. Complete diagram of VIEW system

IV. RELATED WORK

Despite the fact that the expression "logical work processes" were first authored by Vouk and Singh in 1996 [31] for work process applications in logical PSEs, just as of late, there is an expanding energy for the innovative work of SWFMSs and their applications, because of the undeniably requesting necessities of numerous register serious and information intensive logical applications, empowered by the basic advances of figuring advances, eminently Administrations processing [13], Framework processing [32], and Distributed computing [17].Scientific work processes use existing methods produced for business work processes yet go astray from them because of an alternate arrangement of necessities raised from an extensive variety of science and designing issues [33]. While business work processes are control stream situated with the mission of doing business rationale to accomplish a business objective, logical work processes have a tendency to be dataflow arranged and gone for empowering, encouraging, and accelerating the inference of logical outcomes from crude informational indexes. In spite of the fact that the reference engineering proposed by the WfMC [2] has been all around embraced in the improvement of various BWFMSs, including the current advancement of the YAWL framework [4] that is gone for investigating different work process designs [34]. Existing designs for BWFMSs are not proper for SWFMSs since business work processes are normally control stream arranged, while logical work processes have a tendency to be dataflow situated, presenting another arrangement of prerequisites and difficulties for framework advancement, from the necessities of escalated client connection, redid UI, reproducibility, top of the line registering, between operability, to heterogeneous information item, administration, and application administration. Specifically, this reference design does not fulfil the key prerequisites from R1 to R5 for a SWFMS.

A few SWFMSs have been created in the course of recent years. Albeit some of them give structures, they are framework and area particular and neglect to fulfil a portion of the key compositional prerequisites for SWFMSs distinguished in Segment 2. The Kepler framework [28] is a Java-based open source SWFMS. Kepler is based upon the Ptolemy II framework and highlights a solid engineering with different extension modules for functionalities required for logical work processes [28]. The Taverna framework

[27] is another Java-based open source SWFMS for the most part focused forever science. Taverna highlights a three-layered engineering [27]: The Application Information Stream layer gives a client's perspective of a work process, concealing the multifaceted nature of interoperation of administrations; the Execution Stream layer is in charge of work process planning, benefit revelation, information, and metadata administration; and the Processor Conjuring layer is responsible for the summon of solid administrations. The Triana framework [14] has a refined graphical UI for work process structure and change, including bunch ing, altering, and zooming capacities. Originating from the gravitational wave field, the framework contains an extensive archive of devices for information investigation and preparing. The VisTrails framework [10] is created to oversee perceptions and is the main framework that backings provenance following of work process development notwithstanding following the information item inference history. The Pegasus framework [12] gives a system which maps complex logical work processes onto dispersed network assets. Manmade brainpower arranging methods are utilized as a part of Pegasus for work process organization. At last, the Quick framework [11] consolidates a novel scripting dialect called Swift Script with a capable runtime framework to help the determination and execution of expansive approximately coupled calculations over Lattice situations. In spite of the fact that these SWFMSs give much involvement in future innovative work, an examination from a structural viewpoint is as yet absent. The absence of reference engineering keeps the interoperability between various frameworks and farthest point the reusability, adaptability, and extensibility of frameworks, and also the arrangement of heterogeneous secluded subsystems. Therefore, it is difficult to execute one logical work process crosswise over various SWFMSs, and a subsystem in one SWFMS can't be reused by another SWFMS, despite the fact that they give a similar usefulness. Along these lines, there is a squeezing requirement for a compositional reference that can give an abnormal state association of subsystems and their communications in a SWFMS. The accessibility of such a reference engineering can't just give a direction to the building plan of a specific SWFMS, yet in addition give a premise to survey and analyze different SWFMSs and advance the interopercapacity between various frameworks.

Logical work process frameworks can be viewed as one sort of dataflow-situated PSEs [31], which give computational offices expected to take care of an objective class of issues. Logical work process frameworks are identified with and bear a few highlights of dataflow-based perception frameworks, for example, AVS, IBM's Information Pioneer, and SCIRun. From a product building point of view, logical work processes can be viewed as one procedure for segment based programming engineer-ing, where work process errands are the reflection of programming segments, and a logical work process application is created by the development of work process undertakings and their synthesis. As most logical work process frameworks give easy to understand interfaces to planning logical work processes, logical work processes are firmly identified with the field of visual programming dialects, where not just the segments and the structure of logical work processes should be portrayed yet in addition the design of work processes in show ought to be absolutely characterized. Our proposed reference engineering plans to be technology-autonomous for long haul maintainability. The VIEW SOA arrangement is a case of this reference engineering in the light of the outstanding S3 SOA reference design [35].As an outcome, our VIEW SOA arrangement is an improvement of the reference engineering with every one of the points of interest brought by the SOA innovation. This paper expands [36] with the accompanying extra commitments:

1. The design of the Information Item Administrator is given in subtle elements; 2. In Segment 4.2, we present another engineering part, called Assignment Agent, to help undertaking execution in heterogeneous and conveyed environments;

3. We include an imaginative Design administration to powerfully arrange, conjure, and oversee VIEW subsystems and VIEW Undertaking Executors, bringing about another segment, Segment 5;

4. A recently created SWFAS, called FiberFlow, is acquainted in Segment 6 with exhibit a use of the VIEW framework.

V. CONCLUSION

In this paper, we proposed the principal reference design for SWFMSs and exhibited a SOA answer for the VIEW framework. We actualized VIEW to approve the attainability of the proposed models and

presented a SWFAS, the FiberFlow framework, to grandstand the utilizations of our VIEW framework. Progressing work incorporates the expansion of our engineering to address security issues, a more comprehensive assessment of the VIEW framework, and the improvement of different SWFASs utilizing Perspective.

REFERENCES

[1] A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou, C. Pautasso, T. Heinis, R. Grønmo, H. Hoff, A. Berre, M. Glittum, and S. Topouzidou, "Developing Scientific Workflows from Heteroge-neous Services," SIGMOD Record, vol. 35, no. 2, pp. 22-28, 2006.

[2] D. Hollingsworth, Workflow Management Coalition Specification: The Workflow Reference Model, Document Number TC00-1003, v. 1.1, 1995.

[3] P. Grefen and R. de Vries, "A Reference Architecture for Workflow Management Systems," Data Knowledge Eng., vol. 27, no. 1, pp. 31-57, 1998.

[4] W. van der Aalst, L. Aldred, M. Dumas, and A. ter Hofstede, "Design and Implementation of the YAWL System," Proc. Center for Advancement of Informal Science Education Conf. (CAiSE '04), pp. 142-159, 2004.

[5] L. Liu, C. Pu, and D. Ruiz, "A Systematic Approach to Flexible Specification, Composition, and Restructuring of Workflow Activities," J. Database Management, vol. 15, no. 1, pp. 1-40, 2004.

[6] J. Miller, D. Palaniswami, A. Sheth, K. Kochut, and H. Singh, "Webwork: METEOR2's Web-Based Workflow Management System," J. Intelligent Information Systems, vol. 10, no. 2, pp. 185-215, 1998.

[7] G. Alonso, R. Gu¨ntho¨r, M. Kamath, D. Agrawal, A. Abbadi, and C. Mohan, "Exotica/FMDC: A Workflow Management System for Mobile and Disconnected Clients," Distributed and Parallel Data-bases, vol. 4, no. 3, pp. 229-247, 1996.

[8] F. Leymann and D. Roller, "Business Process Management with FlowMark," Proc. IEEE CS Int'l Conf. (COMPCON '94), pp. 230-234, 1994.

[9] B. Luda¨scher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Manage-ment and the Kepler System," Concurrency and Computation: Practice and Experience, vol. 18, no. 10, pp. 1039-1065, 2006.

[10] S. Callahan, J. Freire, E. Santos, C. Scheidegger, C. Silva, and H. Vo, "VisTrails: Visualization Meets Data Management," Proc. Special Interest Group on Management of Data Conf. (SIGMOD '06), 745-747, 2006.

[11] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Vonlaszewski, I. Raicu, T. Stef-Praun, and M. Wilde, "Swift: Fast, Reliable, Loosely Coupled Parallel Computation," Proc. IEEE Int'l Workshop Scien-tific Workflows (SWF '07), pp. 199-206, 2007.

[12] E. Deelman, G. Singh, M. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. Berriman, J. Good, A. Laity, J. Jacob, and D. Katz, "Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems," Scientific Pro-gramming J., vol. 13, no. 3, pp. 219-237, 2005.

[13] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, R. Greenwood, T. Carver, K. Glover, M. Pocock, A. Wipat, and P. Li, "Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows," Bioinformatics, vol. 20, no. 17, pp. 3045-3054, 2004.

[14] S. Majithia, M. Shields, I. Taylor, and I. Wang, "Triana: A Graphical Web Service Composition and Execution Toolkit," Proc. IEEE Int'l Conf. Web Services (ICWS '04), pp. 514-524, 2004.

[15] L. Zhang, J. Zhang, and H. Cai, Services Computing. Springer, 2007.

[16] C. Lin and S. Lu, "Architectures of Workflow Management Systems: A Survey," Technical Report TR-SWR-01-2008, 2008.

[17] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Proc. IEEE Grid Computing Environments Workshop, pp. 1-10, 2008.

[18] D. Georgakopoulos, M. Hornick, and A. Sheth, "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," Distributed and Parallel Databases, vol. 3, no. 2, pp. 119-153, 1995.

[19] K. Ostrowski, K. Birman, and D. Dolev, "Extensible Architecture for High-Performance, Scalable, Reliable Publish-Subscribe Event-ing and Notification," Int'l J. Web Service Research, vol. 4, no. 4, 18-58, 2007.

[20] A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi, "Storing and Querying Scientific Workflow Provenance Metadata Using an RDBMS," Proc. Second IEEE Int'l Workshop Scientific Workflows and Business Workflow Standards in E-Science, pp. 611-618, 2007.

[21] M. Atay, A. Chebotko, D. Liu, S. Lu, and F. Fotouhi, "Efficient Schema-Based XML-to-Relational Data Mapping," Information Systems, vol. 32, no. 3, pp. 458-476, 2007.

[22] A. Chebotko, M. Atay, S. Lu, and F. Fotouhi, "XML Subtree Reconstruction from Relational Storage of XML Documents," Data Knowledge Eng., vol. 62, no. 2, pp. 199-218, 2007.

[23] T. Erl, Service-Oriented Architecture Concepts, Technology and Design. Pearson Education, Inc., 2005.

[24] A. Chebotko, C. Lin, X. Fei, Z. Lai, S. Lu, J. Hua, and F. Fotouhi, "VIEW: A Visual Scientific Workflow Management System," Proc. IEEE Int'l Workshop Scientific Workflows (SWF '07), pp. 207-208, 2007.

[25] D. Pai, O. Muzik, and J. Hua, "Quantitative Analysis of Diffusion Tensor Images Across Subjects Using Probabilistic Tractography," Proc. Int'l Conf. Image Processing (ICIP '08), pp. 1448-1451, 2008.

[26] C. Lin, S. Lu, X. Liang, J. Hua, and O. Muzik, "Cocluster Analysis of Thalamo-Cortical Fiber Tracts Extracted from Diffusion Tensor MRI," Int'l J. Data Mining and Bioinformatics, vol. 2, no. 4, pp. 342-361, 2008.

[27] T. Oinn, M. Greenwood, M.J. Addis, M.N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D.J. Marvin, P. Li, P. Lord, M.R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, "Taverna: Lessons in Creating a Workflow Environment for the Life Sciences," J. Concurrency and Computation: Practice and Experience, vol. 18, no. 10, pp. 1067-1100, 2002.

[28] I. Altintas, O. Barney, and E. Jaeger-Frank, "Provenance Collection Support in the Kepler Scientific Workflow System," Proc. Int'l Provenance and Annotation Workshop (IPAW '06), pp. 118-132, 2006.

[29] E. Deelman and A. Chervenak, "Data Management Challenges of Data Intensive Scientific Workflows," Proc. IEEE Int'l Symp. Cluster Computing and the Grid (CCGRID '08), pp. 687-692, 2008.