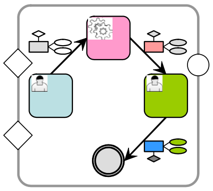

Workshop on Data-Centric Workflows

May 11-12, 2009

Arlington, VA

Sponsored by the National Science Foundation

`http://dcw2009.cs.ucsb.edu/`

Richard Hull[1] and Jianwen Su[2], co-chairs

[1] IBM T.J. Watson Research Center, Hawthorne, NY

[2]Department of Computer Science
University of California at Santa Barbara

Workshop Organizers/Co-Chairs

Richard Hull, IBM TJ Watson Research Center, Hawthorne, NY
Jianwen Su, Department of Computer Science, University of California, Santa Barbara, CA

Managing Program Directors

Frank Olken, National Science Foundation, Washington, DC
Maria Zemankova, National Science Foundation, Washington, DC

Workshop Participants

Serge Abiteboul, INRIA, Paris, France
Landen Bain, Clinical Data Interchange Standards Consortium (CDISC), Round Rock, TX
Roger Barga, Microsoft Research, Seattle, WA
Lawrence Brandt, National Science Foundation, Washington, DC
Peter Buneman, School of Informatics, University of Edinburgh, Edinburgh, UK
Michael Carey, Department of Computer Science, UC Irvine, Irvine, CA
David Cohn, IBM TJ Watson Research Center, Hawthorne, NY
Steve Datena, Lifecom, Portland, OR
Susan Davidson, Department of Computer & Information Science, University of Pennsylvania, Philadelphia, PA
Umeshwar Dayal, HP Labs, Palo Alto, CA
Alin Deutsch, Department of Computer Science & Engineering, UC San Diego, La Jolla, CA
Juliana Friere, School of Computing, University of Utah, Salt Lake City, UT
Dennis Gannon, Microsoft Research, Seattle, WA
Yolanda Gil, USC Information Sciences Institute, Marina Del Rey, CA
Eduard Hovy, USC Information Sciences Institute, Marina Del Rey, CA
Sheila McIlraith, Department of Computer Science, University of Toronto, Toronto, Canada
Tova Milo, School of Computer Science, Tel Aviv University, Tel-Aviv, Israel
Theresa Pardo, Center for Technology in Government, SUNY at Albany, Albany, NY
Joan Peckham, National Science Foundation, Washington, DC
Calton Pu, College of Computing, Georgia Tech, Atlanta, GA
Roberto Rocha, Clinical Informatics Research and Development (CIRD), Partners HealthCare, Wellesley, MA
Jochen Scholl, Information School, University of Washington, Seattle, WA
Wil van der Aalst, Department of Math & Computer Science, Eindhoven University of Technology, Eindhoven, Netherlands
Victor Vianu, Department of Computer Science & Engineering, UC San Diego, La Jolla, CA
William B. Zielinski, Office of Retirement and Survivors Insurance Systems, Social Security Admin., Washington, DC
Michael zur Muehlen, School of Technology Management, Stevens Institute of Technology, Hoboken, NJ

Table of Contents

EXECUTIVE SUMMARY

On May 11-12, 2009, a group of about 30 researchers in the areas of workflow and business process, healthcare informatics, digital government, and scientific workflow gathered in Arlington, Virginia, to identify and discuss the central research challenges in workflow and business process management. The following lists the key findings and recommendations of this group; they remain relevant two years later at the time of preparing this document.

Findings

A. ***The need for workflow management, which is the management of tasks and services to achieve specified goals, is ubiquitous.***

Speaking broadly, there are two kinds of workflow: "transactional", as arises in managing the processes of, e.g., businesses, digital government, and healthcare delivery; and "transformational", as arises in the analysis of scientific data, social networking data, and of data produced by execution of transactional workflows. With increased reliance on the world-wide web, out-sourcing, and globalization, there is increasing need for automation of both kinds of workflow.

B. ***In the context of increasing automation, increasing scale, and increasing out-sourcing and geographic distribution of activities, the widely used workflow technologies do not provide adequate support for a variety of essential functionalities.***

Design and deployment of large-scale workflows is risky, with a high abandonment rate. Most integration of workflows, and interoperation between them, is largely manual, and thus costly, error-prone, and limited in scope. Large-scale workflows are hard to understand, which makes it very costly to maintain them over time, and makes workflow re-use a near impossibility. The tools to understand the provenance of data produced by workflows, and the history of workflow designs, are nascent, and the tools for analyzing past performance of workflow and driving process improvements are largely ad hoc.

C. ***For transactional workflow, a key inhibitor in overcoming these challenges is the lack of intuitively clear ways to combining the various aspects of workflow.***

Today's conceptual models for representing the different aspects of transactional workflows (including processes, data, people and automated agents, rules, and incorporation of legacy applications and external services) are largely disparate, and do not provide unified frameworks that are holistic and simple to understand.

D. ***There is a "long tail" phenomenon in connection with the application contexts that need and/or use workflow management technologies.***

While large-scale deployments of workflow automation tools can be found in most application areas, there are many contexts across all application areas in which workflow is being performed, but the automation tools are too expensive and/or burdensome. As a result, a variety of ad hoc techniques are used for managing both the data (often maintained in spreadsheets) and the processes (often completely manual). Aggregation of that data and processes is very costly. There has been little research to date into workflow paradigms and technologies that address challenges of the long tail.

E. ***The application areas of business, digital government, healthcare delivery, and scientific workflow face many common and overlapping problems, but are developing paradigms, techniques and tools largely in isolation.***

For example, the business process management community needs tools for managing transformational workflows that analyze business process performance, but seems largely unaware of recent advances in scientific workflow. As another example, the notions of "business artifact" and "business object" in the business community, the notion of "record-centric workflow" in the digital government community, and the notion of "cases" in healthcare delivery, social services, and elsewhere, are all highly related and yet the communities are developing frameworks and technologies largely in isolation.

Recommendations

1. ***The emerging area of "data-aware" conceptual models for transactional workflow models holds the promise of enabling substantial progress towards resolving many of today's challenges in transactional workflow.***

Data-aware workflow appears to provide a new perspective on how data and process can be combined, and can spawn fundamentally new Computer Science paradigms and techniques. Research is needed to explore this perspective at a fundamental level, and also in application to challenges such as developing new, unifying conceptual models for workflow; workflow design, evolution, and re-use; workflow interoperation; anaylitics for business process execution histories; and automated reasoning about workflow.

2. ***Research into the emerging area of "workflow as data", which originated in the context of provenance of scientific (transformational) workflow, should be expanded in scope along two key dimensions.***

The dimensions are: (1) Adaptation of the techniques to the transformational workflows used in the analysis of transactional workflow performance, which holds the promise of improving our ability to evolve, re-use, and integrate those workflows; and (2) Generalization of the techniques to work directly with transactional workflows, which holds the promise of improving our ability to analyze and optimize transactional workflow performance, provide decision support based on large volumes of similar medical cases, and develop new ways to ensure and enforce compliance of workflows with enterprise policies and government regulations.

3. ***The research community should include in its focus the expansion of current and new approaches to address the needs of "long-tail" workflow applications, their interoperation, and their aggregation.***

Attention should be given to developing data-aware workflow frameworks, design paradigms, languages, and implementation techniques that can seamlessly support workflow management deployments for differently scaled problems, from very large scale to very small scale, and enable easy interoperation between them.

4. ***It is essential that the core issues be studied in the context of deep collaborations between experts from Computer Science and Management Information Science on the one hand, and domain experts (from business, digital government, healthcare, and science) on the other hand.***

It is through this multi-disciplinary approach that the field will most efficiently develop the fundamental advances in Computer Science in ways that they can be applied to address the root causes of today's overarching workflow challenges.

5. ***Cross-fertilization should be fostered and/or strengthened along two dimensions: (a) incorporating techniques from disciplines ranging from Computer Science, Operations Research, Management Information Science, and Technology and Business Management into workflow and BPM; and (b) between the different research communities that are studying workflow in different application areas.***

Cross-fertilization will lead to a faster and deeper understanding of the fundamental issues shared by the application areas, and enable quicker dissemination and adoption of techniques from one domain to another. Some specific Computer Science technologies include cloud computing, software as a service, software engineering, Human-Computer Interaction, data management, and distributed systems. Both formal and systems aspects of these technologies are of fundamental importance.

1 Introduction

A *business process* is an assembly of tasks performed by human participants or by computing and other devices to accomplish a business objective.[1] They naturally occur in all sectors of modern societies, including governments, education, business, services, healthcare organization, and more. Examples include approving requests for permits, licenses, or budgets, establishing laws, processing loan or other applications, treating patients, etc. In many applications, business process *models* or *schemas* are designed and used to guide execution of *individual instances* or *enactments*. Not only the number of enactments can be large, the number of models in a single application context can range from hundreds (e.g., in a real estate management office of a large city) to tens of thousands (e.g., in a super high speed train manufacturer in China). Based on the nature of execution results, workflow can be divided into two kinds: *transactional* workflow that can update shared databases or the environment (e.g., a loan application), and *transformational* workflow that will compute outputs from all inputs acquired (e.g., scientific workflow). Note that repeated applications of a transformational workflow always produce the same results, but this property generally fails for transactional workflows. This report focuses on transactional workflow; an earlier report on scientific workflow can be found in [24].

Prior to the prevalence of computing and IT technology, managing business processes relies basically on physical paper for documents, bookkeeping, records, letters, etc. and on better logistic arrangement of offices and resources to remain efficient and effective. A major help came more than 30 years ago when data management (software) systems became available. These systems automate storage and management of some of more structured data in digital form. They not only are far more efficient than human, but also provide more reliable and systematic support for data modeling and management. Data management software provides a fundamental basis for managing business processes.

In response to the demand from applications for more automation, the word “workflow” was coined in the late 80’s to refer to representation of business processes that can be understood by software systems. A new class of software systems, called “workflow systems”, was subsequently born to assist in many aspects management of business processes including but not limited to: managing business data and documents, monitoring execution of activities, delivering data to the needed activities, logging business process executions for many purposes (such as auditing and process improvement), and even providing new business services (e.g., online banking/shopping).

Early research on workflow (management) systems narrowly focused on chaining together activities concerning data accesses, extending the notion of *database transaction*. These studies were mostly reported in the database research community and quickly revealed the difficulty of extending the ACID properties to *workflow transactions*. The research activities soon quieted down. The workflow concept, however, was picked up by researchers studying Petri nets. Models of business workflows focusing on *tasks/activities* and their *control flow* were investigated, concerning verification of workflow, properties of workflow schemas, etc. The exclusion of data in these models limits applicability of these work. This report thus aims at workflows that include data as “first-class citizens”, or *data-centric* workflows.

In the last three decades application demand on workflow management has been growing rapidly and created a huge opportunity for the software industry. Today, workflow management systems are available by vendors (SAP, Oracle, IBM, etc.) and in the public domain (e.g., JBPM). However, beyond the ability to create and execute workflow, existing systems barely provide sufficient functionality to support development, analysis, evolution, and interoperation of workflow. This deficiency causes many difficulties in application development. Development of workflow and its interaction with application data is not guided by any principles. Decisions of what to be included are largely arbitrary. It is quite often that the number of factors involved in workflow executions is large and there are also unpredictable number of uncertain factors. It

[1] In this report, “business” or “business process” refer generally to these activities rather than limiting to only *profitable* activities.

makes the workflow hard to understand and hard to manage. Also, much of development and management activities of workflow is done by human and this often results in errors in the workflow (design inconsistencies, omissions, etc.). For examples, a technical support system might not be aware of trouble tickets that were not looked at for a long time, a city office real estate management office had to rely on telephone recordings to reflect state changes of workflow enactments. Furthermore, there is generally no effective way to figure out extra-functional properties (performance, availability, etc), e.g., for validation of workflow in the context of compliance (e.g., legal requirements and changes) and constraints (e.g., run-time resource scheduling). Workflow development relies mostly on experiences, and it occurs often that two similar applications end up with very different designs. This is a genuine source for the challenge of interoperation and integration of workflows. Last but not least, making changes to workflow remains a highly difficult problem. In particular, this inability to change is aiding the "long tail" phenomena [4] and hurting workflow applications. The long tail phenomena refer to the situations that while normal execution scenarios of a workflow are a majority and usually specified in the workflow schema, there is a large number of different types of infrequent abnormal scenarios. Most of these abnormal scenarios need ad hoc handling by human and are more costly. On the other hand, including a large number of abnormal scenarios in the workflow schema would significantly increase the development cost. Perhaps the only solution lies in automating the discovery the abnormal scenarios and inclusion into the current workflow.

Clearly, effective support for performing and managing business processes efficiently can significantly reduce cost, increase productivity, and have very positive societal and economical impact. Over the last decade, research activities on workflow systems have been significantly increased. However, most of these research activities happened within individual application areas, namely business, digital government, and healthcare delivery in isolation and without much coordination and collaboration cross application areas.

In order to address overall challenges of workflow development fundamental to all application areas, and to formulate key research challenges, this workshop was held. It assembled practitioners from government agencies, healthcare organizations, health data exchange consortium, and researchers in data management and software systems, in business process management, in digital government, and in healthcare delivery for a 2-day meeting with alternating presentations and discussions. (The complete list of participants is in page 2 of this report.) The workshop identified six research challenge themes for data-centric workflow and made five findings and five recommendations. The findings and recommendations are listed in pages 4–5 of this report.

Six research challenge themes are the following. (1) Holistic conceptual models that include evolving data, performers, resource, and process flows can lay the foundation for making workflow understandable, reasonable, and flexible. (2) Not only workflow consumes and generates new data, workflow can also be viewed as data. This includes both the workflow schema and provenance of workflow execution. Effective tools for managing and reasoning the workflow data provide a significant help in dealing with some of the current difficulties. (3) Reasoning techniques should be developed and practical design methodologies can help simplifying management problems. (4) Better understanding of system issues for workflow design and execution may be a key to obtaining independence properties similar to data independence in database systems. (5) A workflow changes often during its lifecycle. Workflow analytics should provide tools and techniques to support making decisions concerning changes including impact of changes and making suggestions of changes. Process mining provides methods to obtain workflow schemas from execution logs that are an integral part of analytics. Finally, (6) research on workflow interoperation and integration must be paid a significant effort.

Among the recommendations, three of them concern in-depth study on data and workflow. In addition to pointing out the promise of data-centric approach to workflow modeling, one recommendation encourages further investigation into workflow as data, and the other urges development of techniques for dealing with the long tail problem based on workflow models with data. The workshop also emphasizes the necessity of inter-discipline collaboration for workflow research among scientists in computer science and management information science and experts from application domains. The final recommendation elaborates two dimensions, namely technologies and application domains, along which cross-fertilization can be most effective.

Incorporating data into workflow modeling is a relatively young idea, this workshop focused on the modeling approach and conceptual issues that are of immediate concerns, rather than trying to include all workflow issues. Notably missing from this report are the issues of privacy and security—extremely important for most workflow applications, resource modeling and management, and organization models. Once a basic understanding of the data-centricity for workflow is obtained, it will then be a better time to consider these important issues.

This report is organized as follows. Section 2 elaborates in detail the difficulties in workflow application development. Section 3 addresses the need for elevating data and tools concerning data in workflow research and development. Section 4 discusses each of the six research challenges. Section 5 raises the concern on isolated workflow research activities in application domains and recommends collaboration among researchers in computer science and management information systems and experts in application domains.

2 Demands in Increasing Automation in Business Process Management

Business processes occur in almost all organizations, such as schools, government agencies, business, hospitals, etc. and often in the form of routine tasks in the daily life such as shopping, banking, shipping packages, checking in/out books from libraries. A trend rapidly impacting business processes is that an increasing number of business processes are being helped and managed by special purpose software systems called workflow systems. For example, the worldwide market revenue of software products related to business process management was $250 millions in 2001, grew to $1.8 billion in 2008, and is expected to reach $7 billion by 2018 (according to a 2012 WinterGreen Research study). If middleware is also included, the market already reached $16.1 billion in 2010 by a 2011 IDC study (International Data Corporation). It is estimated that the growth rate of BPM Suites (workflow engines and associated products) has been around 50% in Americas and Europe and close to 40% in Asia/Pacific. The primary driver for adopting BPM (workflow software) is "automation or accelerating highly manual process", with "understanding process" being the most important obstacle to overcome.

From the technology perspective, a fundamental deficiency today is the lack of effective tools to support development of business processes (both software and manual) and workflow to capture, understand, analyze application data (analytics and decision support), and how/why a workflow operates.

Business process and workflow design and management, in general, need to distinguish between and address two types of complexity: (a) *detail complexity*, which relates to a high number and relationships of variables involved in a workflow including enactments, and (b) *dynamic complexity*, which relates to an uncertain number of variables and uncertain relationships involved in a workflow including enactments of workflows coupled with unpredictable interactions between variables. While workflows coping with detail complexity appear as highly structurable or pre-programmable, at least in principle, workflows addressing dynamic complexity such as situational unpredictability (for example, regarding time and space), unforeseeable resource constraints, and constrained information access including information distortion might need different approaches, which might include algorithmic and non-algorithmic approaches.

Consider the chronic disease management of type II Diabetes Mellitus as an example. Optimal workflow management of diabetic patients requires melding of patient controlled workflows (diet, exercise, self-medication, plasma glucose assessment, etc.) necessary for the individual compliance with treatment, with clinical workflows such as medication dosing, surveillance, and prevention of complications which may be performed by a physician or their designated surrogates. Overlaying these clinical processes are administrative workflows and data requirements in support of billing and coding which will become even more critical and complex as the industry adopts pay for performance. Adding the possibility of additional patient chronic disease layers (hypertension, congestive heart failure, etc.) each with its own clinical and administrative workflow needs, the ever-changing nature of therapy and the requirement to provide general care to the patient within the variable context created by the chronic disease(s) makes the magnitude of the problem apparent.

Government agencies have to manage cases of varying complexity and resource intensity (for example, from construction permit applications to terrorist attack prevention or recovery). While workflows follow a general scheme from triggering event, official case initiation, case investigation, evaluation, response preparation, negotiation, response execution (including legal filing if appropriate), and recording/archiving, they may vary widely in their exact sequences and enactments across cases, agencies, and jurisdiction. Like in health care, government data sets are large, noisy (of variable certainty, reliability, and contextual relevance) and constantly changing over time, both in content and format (data schema). Despite explicit and detailed rules, regulations, statutes, the same workflows can produce case outcomes of great variability. Cross-case workflow enactment analysis might greatly help increase consistency of decision making across cases and establish an effective case controlling mechanism. They might become instrumental for overhauling and streamlining workflows including the necessary revision of rules, regulations, and statutes.

Increase automaton for complex business processes needs tools to support the automation of workflow, from workflow schema design, evolution, to integration, including the run-time support, dynamic adaptation, and process improvement.

When increasing the degree of automation of health care workflows described in scenarios mentioned above, workflow schema design and workflow execution become a moving target for developers necessitating great flexibility and room for both growth and evolution. For example, from the patient point of view, appropriate caring often requires integration of data sets from various sources such as hospitals, labs, specialized physicians. The care is carried out by layered and judicious prioritization of multiple distinct and evolving workflow or protocol elements.

In government, over the past thirty years quite a few efforts have been undertaken to streamline processes. The Paperwork Reduction Act of 1980, the Clinger-Cohen Act of 1996, and the Electronic Government Act of 2002 exemplify legislative initiatives aimed at improving the effectiveness of government operations by cutting down on unnecessary complexity and using ICT for supporting business processes. In 1999, the Clinger-Cohen Act instigated the development and implementation of a Federal Enterprise Architecture Framework (FEAF). The FEAF layers comprise various interrelated reference models (performance, business, service components, data, and technical reference models) and have helped guide process re-engineering and ICT infrastructure redesign, at least, with regard to shared federal assets and resources. However, while FEAF has undoubtedly helped develop a more coordinated and strategically aligned approach to government information sharing and ICT enablement among federal agencies, federal agencies have still a long way to go before a truly integrated process and workflow landscape emerges. Furthermore, the enterprise architecture used in the Federal Government has no equivalent in many of the fifty states or the around 3,200 counties, let alone the 60,000 smaller local governments and other local jurisdictions. Since the various levels of government cannot be forced to comply with federal standards, the evolution of process standards will most probably be slow and partial.

However, standardized architectures are the prerequisite for smooth integration of processes and interoperation of systems. Remarkably, independent jurisdictions have demonstrated a willingness and capacity to effectively collaborate in terms of process/information integration and system interoperation even over extended periods of time, if a shared interest is strong and persistent enough. Automated tools, which let government agencies thoroughly but quickly compare processes, workflows, and methods in use, analyze gaps, help them design and prototype integrated workflows and integrated enactments would be highly supportive of cross-jurisdictional initiatives for process/workflow integration and system interoperation. Despite all detail complexity, governments themselves and in collaboration with other governments and businesses have a need and demonstrate a willingness to simplify and streamline processes and workflows. Tools for schema design, integration, and evolution of integrated schemas would have to be reflective of the varying levels of formal training and expertise of government ICT personnel. Runtime execution, monitoring, and support of integrated workflows enabled by inter-operating systems might also require sophisticated tools, which has to be operated by a workforce with diverse training backgrounds and expertise.

Furthermore, many complex business processes require automation with guarantees*, specifically, tools to support the specification of "extra-functional" properties (performance, availability, etc., essential for service level agreements or SLAs) and their validation in business process routines, procedures, and protocols in the context of compliance (e.g., legal requirements and changes) and constraints (e.g., run-time resource scheduling).*

Automated health care workflows require execution guarantees far beyond simple "best of effort" in several dimensions such as real-time, availability and business continuity (e.g., in a hospital), and accountability due to legal and ethical concerns. Generally, modeling and implementing reasoned workflows tailored to contextual synthesis of data will provide multi-factorial decision support for relevant pattern recognition, in initial evaluation and diagnosis, chronic disease management, jurisdictional reporting, clinical and administrative performance analytics, administrative accounting, conflict resolution amongst competing priorities, intervention, relevant surveillance, and education.

Government rules, regulations, and statutes (in the following just "rules") do not only present a challenge for businesses and other non-governmental organizations but also for government agencies themselves. Since those rules keep changing over time, government-internal workflows also need to be modified, added, or replaced. Those changes in rules occur with unpredictable frequency and to extents unforeseeable. On the one hand, changes over time will make process mining across previous workflow enactments the more challenging the more changes have occurred. On the other hand, the very nature of frequent changes in rules makes a strong case for data-aware tools and methods, which may help automate the monitoring of compliance with rules inside and outside government.

Law enforcement has increasingly relied on computer-aided analysis and data mining methods. With regard to the upcoming badly needed changes of rules in the financial sector, for example, the foremost problem regulators face is not the definition and scope of the new rules but rather their enforceability. This is where data-aware workflow management might help add a dimension of enforceability not available before. Real-time auditing and transaction tracking with embedded rule-compliance checking would enable regulators to make effective rules, which are harder to circumvent. With regard to compliance surveillance, it is not only necessary to monitor in detail, in which ways data have been processed but also which exact data were provided by whom and when before any processing took place. The traceability of information provenance is a cornerstone of rule enforceability. Ideally, with workflows designed and used in a data-aware fashion, electronic traces would be searchable and findable not only in real time but also after the fact via targeted pattern searches and other algorithmic approaches. Built-in data-aware traceability might even become a central element of new rules in the financial sector.

A key difficulty many businesses facing currently is the lack of techniques and tools for supporting interoperation and integration of automated complex business processes with guarantees, including evolution and validation aspects.

Consider as an example the expense reimbursement process at a university for participants traveling to a workshop that is funded by an NSF grant. This scenario involves at least the administrative and accounting units and is further complicated by the facts that the workshop is held at a different city with participants coming from all over the world who may not be employees of the university. Also the funding agency may determine that US carriers be used for international flights, and some participants may have a combined trip for multiple purposes. The logical entities involved in the reimbursement workflow include the Reimbursement Request (RR) form, receipts, the trip, the traveler, the grant, the PI, and the department chair. The processing of an RR starts with the submission of a completed RR-form with all necessary receipts. Upon receiving the RR form and receipts, an administrative assistant fills out a standard form used by the university accounting and does an initial examination of receipts and other constraints. The PI and the department chair approve the form (in that order). The workflow then resumes in the accounting department where several steps are taken. First, validity of expenses is checked, this includes expense types and amounts, daily limits, etc., provision of receipts is verified for all expenses where required, and necessary signatures are in place. Accounting office will then sum up the expenses and compute overhead (note that the overhead rate for the PI and other participants are different). To complete the workflow, the corresponding research grant is debited, and a check is produced and sent to the traveler. During the processing, the traveler may be requested for more information (evidence, justification, etc.) if needed.

There are many issues arising from this workflow alone. One interesting and challenging issue concerns workflow interoperation. Consider the case where a traveler from a foreign institution combines her workshop trip with her trip to visit a company. In that case she would like to reimburse part of her trip expenses from the university and the remaining from her institution. Suppose she starts both reimbursement workflows roughly around the same time, clearly there will be interactions between the two workflows. The traveler will unfortunately serve as the "hub" between the workflows. For example, if she has only one set of receipts and both the university and her institution need (to examine) the original receipts, she will be rather involved in resolving this issue. Also, suppose she paid her expenses with her corporate credit card, she might have to receive check from the university and then pay back to her institution. It would be much desirable to empower the traveler to facilitate interoperation between two workflows, rather than overburden the traveler unnecessarily.

The need for integration and interoperation arises virtually in all application areas. From a patient point of view, contextually relevant health care data comes from multiple jurisdictions (personal health records, EHR's, ED records, multiple clinician offices, public health agencies, medical literature etc.), databases (records) and time periods. From a provider point of view, interactions among the general and specialist physicians, hospitals, laboratories, insurance companies, governments, and the patients, interoperation and integration of each one's workflows is needed for the elimination of paper files.

In government, process and information integration enabled via system interoperation can be viewed as the premier challenge of government in the information age. The greater challenges in this particular area might not even lie in the area of technology, which provides ever more powerful tools and methods for system interoperation and even dynamic inter-operability. Unlike business with its more or less monolithic and hierarchical governance structures, government operates in an environment with deliberate division of powers. The founding fathers were unanimous regarding no other principle more than the prevention of a central and overpowering type of government in this country. Extended process and information integration in government, however, might have the capacity to ever so slightly move the constitutional system away from its engrained principles. The question, hence, is how much process and information integration enabled via system interoperation can be afforded or tolerated before core constitutional principles are compromised. Where is the demarcation line, beyond which the system of checks and balances and the factual division of powers would have ceased to exist? In other words, resistance to process and information integration in government might become stiff even from within government (for example, between the Feds and the States, the State and the counties, etc.), which could make projects aimed at cross-governmental workflow integration and system interoperation an arduous undertaking.

While the space, in which governments can integrate their processes and information sources on the basis of system interoperation without meeting increasing resistance, still needs to be charted out and better understood, integration and interoperation have successfully been practiced in government at all levels and in all branches. Emergency and disaster response management (EDRM) and recovery provides an example, which demonstrates how better process and information integration based on effective system interoperation can mitigate the adverse effects of emergencies and disasters. However, EDRM also demonstrates how difficult an integration/interoperation effort can become in the practical case of a disaster despite integration and system interoperation. Responding to a disaster is highly complex in terms of the simultaneity of tasks and workflows, with at times unpredictable interplays of tasks and workflows, and particularly in workflow interruptions via superimposition of higher-priority tasks. In disaster response management the latter problem frequently occurs even in a cascading fashion. While cascading interruptions of workflows are most prominent in EDRM, they are also a typical phenomenon in Daily Routine Operations (DRO).

Hence, from a perspective of current knowledge it appears unlikely that algorithmic tools and methods can be designed, which comprehensively address and solve the problems of dynamic complexity, for example, as outlined above. However, even partial solutions in terms of process and information integration and system interoperation would have the capacity to significantly improve EDRM and DRO in terms of more informed decisions. For example, data-aware workflows in EDRM could be designed in a way that the workflow enactment at the point of interruption would be fully captured and stored. While upon resuming the interrupted task major variables might have changed their values and even new variables might have appeared, EDRM personnel would have far better information at hand before and upon returning to the interrupted task. Even a sensor- or human-based monitoring left with the affected site at interruption could be envisioned, which updates the interrupted workflow enactment and continuously provides the disaster responders with new clues.

Business process and workflow automation must effectively deal with variability, in particular, "long tail" phenomena, and need tools o support evolution.

Unlike traditional software systems, a business process is often expected to have exceptional situations that happen with a small probability and deal with them effectively. The long tail phenomenon refers to the fact that there are many such types of exceptional cases [4].

Virulent infectious diseases provide a complex example of the challenges created by the workflow needs of multiple agencies (clinicians, health care systems, public health agents, and governmental organizations) which are often in conflict or may come into conflict as the outbreak evolves and the overall response must adapt. Existing limitations of our current system begin with clinician recognition of disease and the potential threat of spread. If the disease is unusual, the clinicians may not be up to date on the initial diagnosis, treatments, and policies and procedures regarding related reporting requirements. The confirmatory tests may not be readily available. The systems in place to initiate tests, capture test results, share results with necessary partners are inconsistent, non-inter-operable, and often manual. Clarity of reporting responsibilities—i.e. who do I report to—are unclear and involve multiple actors in including the public health arena (local, state, federal and others). Patient confidentiality restrictions are complicating factors. The facility for retrospective identification of additional patients who satisfy the criteria of the now recognized outbreak is limited by the inability to mine populations of data (How many people had the following 7 symptoms within x miles of the following locations—how many people are in this person's social network, travel, etc.) Adaptation of the usual clinician workflow in light of the threat requires atypical knowledge and processes. Governmental challenges include decisions such as when to employ public resources to respond in nonclinical ways (in the West Nile example, spraying still water; swine flu, closing schools, restricting travel). At some point in the outbreak health care should move from a per patient delivery model into a public health threat. Metrics and reasoning processes to identify such tipping points are often absent or ambiguous. Clinician data critical to engage the business rules regarding situationally appropriate responses to the level of threat are not always readily available to decision-makers at all levels. Cross disciplinary problems abound such as how to prevent the "worried well" from overtaxing clinical resources.

When increasing the degree of automation of health care workflows described in the above scenarios (e.g., in workflow protocol and process optimization), workflow schema design and workflow execution become a moving target for developers necessitating great flexibility and room for both growth and evolution. For example, from the patient point of view, appropriate caring often requires integration of data sets from various sources such as hospitals, labs, specialized physicians. The care is carried out by layered and judicious prioritization of multiple distinct and evolving workflow or protocol elements.

3 Trends Towards Data-Centricity in Workflow Management

Data plays a critical role in virtually all workflow applications. The first evidence is that data and its associated semantics is a fundamental piece in formulating workflow semantics, i.e., what actually a workflow performs and how its actions are related to the environment/context. Consequently, the data faithfully records the progress of individual workflow executions (instances), including execution status, resource usage and status, and correlations with other workflow instances. The third aspect is that executing a workflow would generate additional data for a variety of reasons such as monitoring for performance or business concerns, auditing, compliance checking, etc. Finally, even workflow schemas and enactments can be viewed as data so that they can be managed, queried, mined for processes, and analyzed.

This section provides an overview of emerging trends and approaches to incorporating a philosophy of data management into the realm of workflow management. Two broad themes have arisen, which are briefly described in the first two subsections. The third subsection describes some of the evidence that suggests that data-centric perspectives on workflow can help to resolve the application challenges described in Section 2.

3.1 Data-aware workflow

Early research and tools for workflow and business process management understood the intimate relationship of flows of tasks and the data used by the tasks to record their impact. For example, the STEP system specified workflow processing in part using Event-Condition-Action (ECA) rules that acted on underlying databases [36] In the 90's, however, attention began to focus largely on the activity flows, with the data being manipulated becoming a second-class citizen and in some cases largely lost from view. This trend is also reflected in the closely related area of composition for SOA and web services, which has emphasized standards such as BPEL, that are founded on process calculi and emphasize message passing but ignore how persistent data might be stored and manipulated. Over the years, disparate conceptual models have arisen to capture some aspects of the data, including the emergence of workflow analytics and extract-transform-load (ETL) systems, which generally use the relational database model to support historical views and queries over workflow enactments, and vocabularies and rule languages for specifying the rules and requirements to be satisfied by the workflow enactments. Yet other models have been used in connection with complex event processing (CEP) and the management of the actors involved with performing workflow actions [3].

Beginning in the late 1990's, there has been a small but growing trend to bring the data back to a level of prominence, and more recently, to experiment with approaches that attempt to tightly combine the data and process into the core building blocks of workflow specifications. These approaches hold the promise of providing the foundation for a unified, holistic conceptual model into which the other aspects of workflow can be incorporated. This subsection briefly highlights some of the key developments that are shaping this trend of "data-aware" workflow models.

An important early development in data-aware workflow was the emergence of document engineering [35], which focuses on the "documents" or data values that are passed between (sub-)organizations in a workflow. This approach is very natural in the context of the Web's Representational State Transfer (REST) [29], which enables a loosely coupled style of service composition with its focus on specifying how resources (documents) are transferred between services, the internal operation of the services, and the ways they might be sequenced, are not part of the specification. Several products have been developed around this perspective, including IBM's FileNet.

The area of *data services* in the context of Service-Oriented Architecture (SOA) is quite relevant to data-aware workflow at the infrastructure support level. As noted above, SOA has focused largely on the process flow of services, with little attention to the persistent data that the services manipulate. Data services enable architects to reveal key data sets and the supported create, read, update, delete ("CRUD") capabilities, all within the SOA framework. Data services often focus on families of key business entities, such as customer or product. BEA Systems (now Oracle), Composite Software, IBM, Microsoft, RedHat/MetaMatrix, and DataDirect/XCalia are among the growing list of enterprise infrastructure software vendors that recognize the important role that data services can play in SOA and business processes. All are either offering or developing products that simplify the problem of service-enabling data.

More recently, workflow models have emerged that support a much tighter coupling of the data being manipulated and the sequencing of activities that perform those manipulations. A central notion in this work, called here *dynamic artifact* (originally introduced as "business artifact" [53]), is used to capture the essential properties of key conceptual objects which evolve as they move through a workflow. There are two essential components to the specification of a class of dynamic artifacts: (i) a data schema (or information model) for holding information about the artifact as it moves from creation, through the workflow, and in some cases, to archival storage, and (ii) the lifecycle schema which describes how and when tasks (or services) might be invoked on the artifacts as they move through the workflow. A prototypical example of an dynamic artifact is the notion of "air courier package delivery", whose data schema can hold information about a package including sender, receiver, the steps occurring in transport, and the billing activity, and whose lifecycle would specify the possible ways that the delivery service might be carried out. Indeed, the typical package tracking information provided by commercial delivery services can be understood as providing a subset or "view" of the data value associated with the delivery artifact as it progresses through the courier's workflow, along with an abstracted view of the lifecycle, shown as the likely steps that will lead to completion of the enactment. The basic notion of dynamic artifact have been called variously "business artifacts" [53, 8, 40, 7], "business entities" [67], "business objects" [51], "adaptive documents" [46], and "adaptive business objects" [52] in various research and industrial endeavors. In application, systems based on dynamic artifacts typically involve several distinct artifact types, and communication and synchronization between related artifact instances must be supported. A pre-cursor of dynamic artifacts was the notion of *proclet* [69], which provides constructs for specifying workflows as weakly-connected interacting lightweight workflows; this allows a shift away from having to create a single, monolithic overarching workflow for an organization. That work focuses mainly on specification of process flows within a proclet (using Petri nets), and communication between proclet instances based on a language/action perspective (e.g., [30]).

Dynamic artifacts are closely related to the notion of "case" in the context of case management systems. Both involve the notion of a conceptual entity that progresses through time, according to some set of guidelines or lifecycle schema, and both taking advantage of a growing set of data that tracks the lifecycle. Dynamic artifacts are intended for use in a wide variety of application domains, some of which are not typically viewed as cases. Furthermore, with dynamic artifacts it is typical to have two or more dynamic artifact types that have some interactions, whereas a given category of cases often stands alone, with little or no interaction with other categories of cases.

There are parallels between the artifact approach to business operations modeling and the Entity Relationship (ER) approach [18] to modeling the data managed in a business. Both are systematic approaches that use a small set of natural and intuitive constructs. Also, dynamic artifact specifications are *actionable*, in the same way that ER diagrams are actionable, i.e. the specification can be used to automatically generate an executable system. There is a contrast between how information is typically clustered in artifacts vs. in database schema design and document management systems. With database schemas, there is a tendency to break data into fairly small "chunks": ER-based techniques use separate entity types and their relationships; normal forms for relations break data apart to avoid update anomalies. This is valuable when data is used by a variety of applications. Similarly, document management systems often focus on the company's literal document types rather than on the single conceptual entity that multiple document types together represent. In contrast, an artifact information model clusters the various kinds of data which correspond to the stages in the dynamic artifact's lifecycle.

3.2 Workflow as data

The second broad trend in data-centric workflow is that of using data management perspectives and techniques to understand how workflows have performed and are performing. It is fruitful in different application areas to consider *both* workflow schemas and workflow enactments *as data*. This approach is deeply important, regardless of whether the underlying workflow is transactional or transformational, and whether it is process-centric or data-aware. This subsection presents three challenge areas where the perspective of workflow as data is critical, and describes techniques that are being developed to address them.

The first challenge area concerns *scientific workflow*. As some branches of science started to use extended computational processes to explore large-scale data sets, it became imperative to be able to accurately record, and faithfully replicate, the computations. As reported in the NSF Workshop on Challenges of Scientific Workflows in May, 2006 [24] (see also [21]), it became apparent that workflow was an appropriate technology for both managing the computational processes, and recording their *provenance* or history. Indeed, for the context of transformational workflows, the field of scientific workflow has contributed greatly to our understanding of how to view workflow schemas as data, including the development of techniques for storing and querying libraries of workflows (e.g., [31, 9, 17]). Another useful direction has been in the area of modifying and splicing of these workflows, in order to customize the "product" (or output) of a scientific workflow, or to create a family of products through slight variations in the workflow used.

The second challenge area is the need to be able to *effectively and efficiently query large numbers of health care cases*, which are enactments of health care protocols. In this context, a protocol is essentially a workflow schema, possibly with conditionals, that describes the steps to be taken during diagnosis and treatment of a patient condition (or set of patient conditions). For example, the Clinical Data Interchange Standards Consortium (CDISC) is working to develop a framework and techniques for maintaining all of the data arising from clinical trials in the U.S., and supporting effective query capabilities, to retrieve the information about both the protocols experimented with and the enactments that occurred during the experiments. A significant challenge here is the high variability of the enactments, because participants in the trials sometimes forget to take a medication on time, or miss a required visit to the clinic for testing during the trial. Another application of protocol enactment querying is in decision support for routine medical care. In the vision of Lifecom and others, while diagnosing a patient a doctor could be provided with access to a computerized system that can sift through large numbers of protocols and associated cases that are similar to the symptoms exhibited by a patient (and also the enactments of clinical trials with related protocols). The system could suggest further diagnosis steps to the doctor, and indicate the statistical outcomes associated with different treatment plans. More broadly, support for ad hoc and/or automated data mining across both protocols and cases should reveal a wealth of useful, but currently inaccessible, medical insight.

The challenge of querying health care protocols, enactments of them in clinical trials, and also cases, can be viewed as challenges in *managing provenance in transactional workflows*. Some recent research, complimentary to the work in scientific workflow, is laying some foundations for a principled study in this direction Specifically, works such as [5, 26] are developing techniques for querying large libraries of BPEL specifications, and querying repositories of enactments of a BPEL specification. Another, largely unexplored aspect of managing provenance for transactional workflow concerns the creation of new workflow schemas from existing ones, through local modifications and customizations, or through splicing and other "algebraic" manipulations. Research challenges arising from provenance, for both transformational and transactional workflow, are discussed in Subsection 4.2 below.

A third, very broad challenge area arising in the context of business processes and workflows, termed *workflow analytics*, involves enabling stake holders of a family of processes to understand how they are working, and how to improve them. Much of the work to date in this area has centered around *business intelligence*, which comes down to the effective use of information that is Extracted, Transformed, and Loaded (ETL) from business process logs and related data. A key challenge is to identify which data to gather and how to transform and analyze it, in order to measure how an enterprise is performing against key business objectives and policies. Interestingly, an ETL process can be viewed as a transformational workflow, and so techniques from scientific workflow might fruitfully be applied in this domain. Research challenges in this area are discussed further in Subsection 4.5 below.

The area of *process mining* [2, 34, 68] provides an additional tool in support of workflow analytics. This work starts by looking at large repositories of logs generated by workflows, and attempts to reverse-engineer the specifications of the processes underlying those workflows, that is, the workflow schemas. This area is motivated by two facts: (a) many real-world workflows include large portions that are manual, and so an accurate workflow schema does not exist, and (b) even when the workflows are automated, they are written directly in programming languages such as Java or COBOL, and so the underlying workflow schemas are obscured. In addition to obtaining a specification of workflow schema actually guiding a process, techniques from process mining can help with conformance testing (e.g., is a workflow complying with a policy or regulation). Some research challenges in this area are discussed in Subsection 4.5.

4 Research Challenges

4.1 Unifying, holistic conceptual models for transactional workflow and business processes

Typical in many computer and information management domains, the conceptual model used in a system has a tremendous impact on how easy or hard it is to design and use the system, and as a result the various costs involved with employing the system in applications. As a case in point, the shift in database management systems from the navigational models for data management (network, hierarchical) to the relational model enabled a substantial transformation in how people manage data, greatly simplifying and reducing the cost of designing, deploying, and maintaining database management systems. The goal of developing a new style of conceptual models [2] for workflow, which combines at a fundamental level the key constructs of evolving data, process flows, and performers (human participants or device/systems), is to enable an analogous transformation in the field of workflow management, enabling dramatic improvements around enabling stake holders to specify, deploy, and understand workflows, and for the workflows of different organizations to inter-operate effectively.

The widely accepted approaches for the management of transactional workflow today start with a conceptual model primarily focused on the process (or control) flow. The impact of these processes on surrounding data is either not modeled at all, or considered only in terms of the flow of data objects between activities with little focus on how the objects are being modified in the persistent store. For example, the notions of pools and swimlanes in BPMN can be used to partition a workflow, with the ability to model messages between different pools and control flow transition between swimlanes. However, the (persistent) data being read or written by the processes, and records about impacts made to the outside world, are essentially treated as second-class citizens: they are either not modeled, or modeled implicitly as annotations for human consumption. Because the data aspect is not central to the core conceptual model, the ways that data usage and manipulation are incorporated into an operational workflow system are often *ad hoc*, differing

[2] In this document we follow the traditional terminology of the database and workflow communities, where "(conceptual) model" refers to a framework (e.g., the relational model) that provides the structuring primitives for "schemas", which are used in support of specific applications. The Business Process and UML communities typically use the terms "meta-model" and "model", respectively, for these notions.

from implementation to implementation. Indeed, this emphasis on the process flow, together with the lack of explicit constructs in the core conceptual model to represent the (persistent) data being used and manipulated, is a fundamental barrier to many of the challenges encountered by current technology for transactional workflow, as outlined in Section 2 above. Similarly, the relationship of performers to a workflow is typically incorporated as a layer on top of the processing, and using a simplistic model of performers based on the "roles" that they can play. Notions subordinate relationships, legal and policy constraints, management responsibilities, delegation, training of performers, teams of performers, and collaborative performance of activities are largely overlooked in current approaches.

In contrast, the growing area of Case Management embraces data as a first-class citizen, and focuses on the case object (or folder) and its lifecycle schema as the primary building block. While there are industrial case management products, and some technical articles focused on case management models (e.g., [71, 23]), the area is still in its infancy in comparison with the process-centric approaches. For example, the first serious effort to develop a standard for case management models is underway at the time of writing, under the auspices of the OMG [57].

A *fundamental goal* for the community is to converge on a widely applicable conceptual model for transactional workflow that unifies, at a basic level, the three primary components of *evolving data*, *process*, and *performers*. Achieving this goal will require several years of research effort, developing and refining candidate conceptual models, creating prototype systems and applications that use them, and studying them from a variety of perspectives, including applicability in diverse domains, surrounding capabilities and tools, human factors, and theoretical foundations. The case management approach, because it is already available in substantial products with significant customer bases, has the potential for growing to support the data-aware, expressive, flexible, declarative, modular transactional workflow model.

Transactional workflow systems facilitate the management and coordination of (a) activities within which (b) performers (people, computers, external entities) perform tasks, which generally use and manipulate (c) persistent data, and sometimes include effects on the outside world. The selection and timing of individual activities may be determined by the workflow schema and the current status of an enactment, or by the performers, possibly subject to constraints specified in the schema. The intent of a good model is to provide a useful and efficient basis for essentially all of the functions of transactional workflow management and surrounding capabilities. As such, the ultimate test for these models will be their applicability, their ease of use, the extent to which they can save time and effort as compared with current approaches, their ability to support modularity and reasoning, and the extent to which they can be used in new application areas. Some of the most important characteristics anticipated for the eventual model are now listed.

Understanding and visibility. Workflow and business process schemas are often the primary interface between different groups of involved participants, including at least business managers and software engineers. Partly due to the traditional division of academic disciplines it is often the case that in an application context these two groups of people possess disjoint technical backgrounds. As a result, workflow schemas that are understandable and usable by one community are typically not understandable and usable by the other. This leads to significant communication problems, and a significant cost as the two groups work towards a deployment of the workflow processes that the business managers desire. In some cases the software engineers cannot fully support the desires of the business managers, and so there is inefficient use of the workforce that is guided by the workflows.

The ideal workflow model would be based on constructs that are understandable at the business level, and precise enough (or can be augmented to be precise enough) for implementation and deployment. We note that in the realm of data management the Entity-Relationship (ER) model plays this role. As with the ER model, tools should emerge for translating, in a fairly direct way, the business-level understandable workflow schemas into lower-level models that can be implemented and optimized.

The ideal workflow model should support hierarchical modeling approaches and modularity to allow workflow schemas represented in meaningful chunks that can be aggregated. Modularity can enable in-depth consideration of one part of a workflow (schema or enactment) with little or no detail on other parts. The ability to examine workflow schemas with varying levels of details (or abstraction) is a tremendous help to business process stake holders. For example, this could enable to visualize the progress of the evolving data, process, and performers in individual enactments, and to visualize the aggregate behavior of families of enactments.

Some preliminary candidate models in this direction are the case management proposal of [23] and the Guard-Stage-Milestone (GSM) variant of business artifacts [42]. Reference [71], if extended with some form of hierarchy, might also serve as in important preliminary candidate model.

Flexibility. As discussed in Section 2, transactional workflows in practical settings need be flexible along a number of dimensions. These include (i) explicit support for a generic or global view and understanding of the schema, and for specifying specializations or variations, e.g., for different geographic regions, different kinds of customers, etc.; (ii) support for incorporation of new variations at a dynamic level, during run-time of an enactment; and (iii) support for evolution as business requirements, government regulations, and other aspects change.

By using evolving data rather than process as the backbone for a workflow model, there is a strong potential to use declarative approaches to support the different forms of variation. Research has already emerged on the use of declarative approaches to specifying process-centered workflow [72, 60, 38], but these do not incorporate data as a first-class citizen. As a result, basing variation on the data already accumulated and/or modified by an enactment is not possible. Research into declarative variants of business artifacts (e.g., [1, 27, 19, 42]) focus on verification and basic modeling, but not on support for variation.

Rich model of performers. Incorporating data into the core of a workflow model offers rich opportunities for sophisticated modeling of how performers (human or otherwise) can participate with a workflow enactment. For one thing, the assignment of activities to performers can take into account the activity, the status of the processing, and importantly, information based on the evolving data associated with an enactment. It is natural to include in the evolving data the information about who has performed which activities. This could be used to enable continuity with regards to who is performing activities: for example an accountant who did one activity early in an enactment must be assigned (or avoid) some later activity in the same or a related enactment. The explicit, testable presence of data and the processing status also permits the specification of rich access controls that restrict read and write capabilities of performers based on their role, the activity to be performed, the status of the processing and the status of the evolving data [52].

It is also natural to expand the underlying model of human performers. This can include typically organizational relationships such as subordinates and teams. If there is a hierarchy in the processing model, this would allow assignment of a large cluster of activities to a manager, who in turn might allocate sub-clusters of the activities to his subordinates or other specialists. It would also allow substituting one member of a team for another to perform a given activity. Collaboration on single activities, or clusters of them, should also be supported. The primary research challenge here is to develop intuitive, simple-to-use, yet expressive models for performers and how they can be assigned to activities and clusters of them.

Foundation for rules, regulations, and compliance. The ideal workflow model should provide easy, flexible mechanisms to specify variations, to incorporate new government and other regulations as they arise, new findings and technologies (e.g., about medical treatments or tools to conduct business), and to verify compliance with policies and regulations. The basis for these capabilities will most likely be centered around declarative mechanisms for specifying how workflow enactments must or may progress, for assigning performers to activities, and for querying enactments both in flight and after completion. These mechanisms should be able to refer to both the evolving data in enactments and the progress of the enactments through key stages of activity and key milestones (or, in other words, "business-relevant operational objectives"). The use of a rules-based paradigm, in which the rules can be modified without re-compiling the system, as part of the specification of how activities are sequenced will permit rapid incorporation of new variations and regulations. It should be possible in the model to organize the sets of rules in a modular fashion, e.g., structured around the main constructs of the model as in [71, 42]. A challenge is to ensure that the semantics of large collections of rules is both intuitively natural and unambiguous. Another challenge is to find efficient mechanisms to incorporate new data into enactments, both for individual enactments and as the overall workflow schema evolves.

The raw declarative capabilities incorporated into the base workflow model might be made available to different stake holders through various mechanisms. For IT specialists, solution builders, and business analysts, languages such as SQL, XQuery, or the OMG's Object Constraint Language (OCL) [55] might be used to specify conditions or access execution status and histories. Business-level stake holders might use a language based on "business rules", such as a structured English based on OMG's Semantics of Business Vocabulary and Business Rules (SBVR) [56]. For example, [47, 15] describe how business rules can be incorporated into a procedural variant of the business artifact model, in order to support a generic workflow schema with variations. Visual mechanisms might also be used, at least for simpler forms of rules/queries.

Foundation for provenance and analytics. In typical business process management environments today, the process is managed according to one model (e.g., BPMN), while the evolving data is managed according to another model (e.g., the relational data model). There are often several process specifications that address different aspects of a domain, and the evolving data is distributed across multiple data repositories. This heterogeneity at the conceptual level makes it especially challenging to develop an understanding of the provenance of the workflow enactments, i.e., why processes executed in the way they did, why data was created or updated in the way it was. It also compounds the problem of performing analytics over the workflow data and execution histories. Indeed, a key aspect of "Extract-Transform-Load (ETL)" [22] frameworks is to enable the *extraction* of data from conceptually heterogeneous sources, *transform* them into a coherent form and format, and *load* them into a data warehouse for subsequent analysis.

The ideal workflow model should provide a unified approach to managing the evolving data and processes, so that provenance and analysis can us the same conceptual basis as the workflow model itself. In this aspect, business artifacts provide a natural, holistic basis for maintaining a history of the both key data and processing steps that were performed during workflow executions. An important research direction is to apply techniques from provenance for scientific workflows to the process histories found in business artifacts and other data-aware workflow models.

Data-aware workflow schemas can also be used to provide a conceptually unified view of one or more existing workflows. In this approach, a workflow schema can provide an integrated view of processes that span multiple organizational hierarchies. A data warehouse can be created, that holds a history of the executions (and associated data usage and updates) of the existing workflows, and that is structured according to business artifacts or some other data-aware workflow model. The warehouse might be populated based on events coming from the existing workflows, or based on periodic bulk uploads of data from those workflows. While this vision has been demonstrated in some practical situation, the approach is in its infancy. One research challenge here is to develop systematic approaches for populating the integrated view; ETL is a natural starting point for this direction.

Foundation for the study of theory and fundamentals. The foundations of process-centric workflow models are quite mature, with significant understanding of Petri nets, process algebras, state machines, and their application in connection with verification among other topics. In contrast, our understanding of the foundations of data-aware workflows is still in its infancy.

A basic challenge is to develop crisp formal definitions of the key constructs needed for data-aware workflows and how these constructs can interact, and to provide frameworks that start with a core set of constructs and can build multiple layers around those to provide increased expressive power and applicability. An early work in this direction is [71], which focuses on a formalization of case management. More recent work is [20, 42], which provides a formal foundation for the operational semantics of the GSM variant of business artifacts, including an equivalence theorem that relates a perspective based on forward chaining of ECA-like rules with a fixpoint perspective. Reference [37] develops a quite general formal model for business artifacts, in which the information schemas are full relational schemas, and the lifecycle schema is based on condition/action rules. Additional variations of data-aware workflow models will no doubt arise as such workflows become more commonly used in practice and new features are desired. This will lead to further research into the underlying formalism.

A key challenge for data-aware workflow is to develop effective mechanisms to verify properties about their behaviors. In general the verification problem is undecidable because the presence of data as a first-class citizen makes the set of possible states infinite. There have been several recent research efforts that have found abstractions that enable decidable verification: decidability in PSPACE of properties expressed in FO-LTL concerning artifact-based workflows [27, 19], and application of recent results about data dependencies developed for data exchange [37]. Further research in this area will be motivated by the need to find additional abstractions that permit verification of different classes of temporal properties.

A variety of other research questions can be asked in connection with data-aware workflow. One natural question is to develop a theory for characterizing relative expressive power: when can one data-aware workflow faithfully simulate another one [14]? Another important research area concerns the notion of "views" of data-aware workflows. These are important, for example, in the context of artifact-centric interoperation hubs [43], where different services in a collaboration are provided access to restricted portions of the artifact schema in the hub. It would also be useful to develop an algebra for building data-centric workflow schemas from "fragments" of workflow schemas. This might be useful for enabling the rapid creation of workflow schemas that fit a particular application need.

4.2 Workflow as data

While the evolving data is intimately related to workflow, workflow schemas and their executions (enactments) generate another kind of data, which we name as "workflow as data" or simply "workflow data". Workflow data includes technical specifications of workflow schemas, provenance associated with or logs of workflow executions.

In many application contexts, workflow data may be large to very large. According to [44], the number of workflow schemas is over 3,000 for Haier and 6,000 for SunCorp, and exceeds 200,000 for China CNR Corporation. The large number of workflow schemas makes the management of workflow schemas challenging. Provenance and execution logs are clearly relevant to several management functions in workflow systems, including at least monitoring executions and especially *key performance indictors* (KPIs), support for policy compliance especially auditing, and evolution of workflow schemas. The size of provenance and logs depends the number of workflow executions In the Chinese city of Hangzhou, the number of cases processed by the city's Housing Management Office reaches 300,000 per year. The US National Trauma Data Bank [3] has datasets containing 0.5 to 2 million trauma patients annually from 2006 to 2010. Generally, a significant portion of The Big Data exists in the form of provenance or execution logs.

The provenance (also referred to as audit trail, lineage, and pedigree) in the workflow setting refers to the information about the process and data used to the current state or completion of an execution. It provides important documentation that is key to preserving the original data, to determining the trail of modification on the data for many purposes such as auditing in transactional workflow, and sometimes to

[3] `https://www.ntdbdatacenter.com`.

reproduce as well as validate results (e.g., in transformational workflow).

In many research communities, the need for detailed provenance of scientific experiments has been a major drive for the adoption of workflow technology in many scientific disciplines [21, 32]. To analyze and understand scientific data, complex computational processes must be assembled, often requiring the combination of loosely-coupled resources, specialized libraries, and grid and Web services. Ad-hoc approaches to data exploration (e.g., Perl scripts) have been widely used in the scientific community, but have serious limitations. In particular, scientists and engineers need to expend substantial effort managing data (e.g. scripts that encode computational tasks, raw data, data products, and notes) and recording provenance information so that basic questions can be answered, such as: Who created this data product and when? When was it modified and by whom? What was the process used to create the data product? Were two data products derived from the same raw data? Not only is the process time-consuming, but also error-prone. Workflow systems not only support automation of repetitive tasks, they can also capture complex analysis processes at various levels of detail and systematically capture provenance information for the derived data products.

The need for provenance has also become evident in other areas. In the corporate world, the Sarbanes-Oxley Act [63] instituted auditing rules that require companies to maintain detailed record of financial transactions. In health care, the push for evidence-based medicine requires that detailed provenance be maintained for patients, the treatments they underwent and their outcomes: Only with clinical evidence from systematic research at hand will doctors be able to make evidence-based decisions [62, 66].

While there has been progress on the foundations of provenance and provenance management systems in the context of scientific workflows [33, 32, 21] and databases [13], provenance management is still an incipient area. Though much of the database and workflow-related research can be applied in the application areas that we focused on this workshop, they present several new challenges, both practical and theoretical.

Provenance in workflows is captured as a set of dependencies between entities, which can be artifacts (e.g. data files), processes (e.g. programs or web services) or agents. It is modeled as a *directed graph* which captures causality and explains how an artifact came to be [50]. In research on provenance in databases, the emphasis has been on the propagation of provenance through the operators that make up database views, or on propagation of provenance through copy/cut-and-paste operations within and among databases (see recent tutorial [13]). Since the operators that make up database views have well understood properties—in contrast with the black-box view of modules or processes in workflows—provenance is reasoned about at the level of tuples of the input and output relations rather than at the level of a file (e.g. the database itself). Thus, database provenance is often termed “fine-grained” as opposed to the “coarse-grained” provenance in workflows.

Capturing Provenance. In scientific workflows which represent *in-silico* experiments performed within a single workflow system, capturing provenance is easily automated by logging events such as reading/writing to a database, or sending/receiving messages (see, for example, Kepler [10], Taverna [58], VisTrails [33]). However, when part of the execution falls outside the workflow system (e.g. into the hands of a client, as in the expense reimbursement scenario, or is passed off to a web service or other workflow environment) either part of the provenance will be lost or must be explicitly requested. In either case, provenance management systems must be capable of reasoning with partial information, and potentially at multiple levels of granularity.

Managing Provenance. It has been observed [16] that storing provenance information may entail a ten-fold blow up of the amount of data recorded. Efficiently managing provenance information—storing, indexing, querying and optimizing queries—is therefore essential. Since provenance is a directed graph, prior research on graph databases and graph query languages (e.g., SPARQL[4]) may provide insights, although the ability to view workflows and their provenance at varying levels of granularity adds new challenges.

[4] `http://www.w3.org/TR/rdf-sparql-query/`

Within the context of business workflow, an interesting language for querying workflows of interest (BP-QL) as well as monitoring workflows during their executions (BP-MON) have been proposed [5, 6]. In particular, expression in these languages can not only specify paths in the dependency graph at a single level, but can "zoom" into sub-workflows to expand a path. This represents a very promising approach to take within other application areas.

Integrating Provenance. Since data moves between databases and workflows, and from one workflow to another workflow, the provenance associated with that data should be carried with it. However, there are many challenges in doing this, including: (1) The lack of a common model of provenance: Although standards (such as the Open Provenance Model [50]) can be developed, they must be adopted by the community. To be adopted by the community, there must be incentives. (2) Differences in levels of granularity at which provenance is associated: For example, in a workflow provenance is typically associated at the level of a file whereas in databases it is associated at the level of a tuple. (3) Security: Provenance gives information about the processes and entities that led to the creation of another entity. However, people may be willing to share this information at different levels of detail to different users. For example, scientists may be willing to share all information with their immediate collaborators, but almost no information with people from outside their organization.

Provenance Analytics. The ability to mine provenance information has several promising applications. For example, run-time monitoring of a workflow execution of a Web-based auctioning system may allow the manager to detect fraud attempts and track services usage and performance. Querying and analyzing provenance posteriorly may allow the manager to identify usage trends and optimize the workflow accordingly. In this way, run-time monitoring/provenance analysis can be used to analyze properties that cannot be statically determined by querying the workflow specification. Analyzing provenance may also provide insights into how to design new workflows, an approach that has been explored in [45, 64].

4.3 Design methodology and reasoning

A significant research challenge for data-centric workflow is the integration of reasoning into various stages of the workflow lifecycle. This includes providing support for workflow design and specification, automating the verification and/or synthesis of high-level workflow and regulatory specifications, and automating and providing decision support for aspects of workflow application and/or execution. Due to the lack of effective and efficient tools, workflow management systems in practice are not designed using rigorous techniques nor analyzed with verifiers. This leads to many issues, e.g., some IT trouble tickets may be untouched for a long time, or it is unclear what impact a small change to the data/workflow will have on other workflows and data management systems. In what follows, we elaborate on a subset of specific challenges relating to these objectives.

Modeling for Reasoning. Central to the realization of any form of automated reasoning is the need for effective workflow modeling. Building on the discussion in Section 4.1, automated reasoning prefers a declarative specification of the workflow together with a well-defined semantics. While process modeling techniques such as process algebras [39] and Petri nets [61] have been used for some time to model business processes and workflows, their focus has primarily been on verification specific properties of control flow, and as such do not contain the necessary expressivity to deal with the diversity of reasoning we envisage in next-generation data-centric workflow.

The goal is to develop workflow modeling techniques that support sound and effective automated reasoning techniques, such as the ones identified later in this section. Properties of modeling languages include the ability to model:

- *Data*, *processes*, and *actors* as first-class citizens. Most existing process modeling techniques support the representation of data, processes, and actors, but not necessarily as first-class citizens. As such, one cannot talk about them within the modeling language.
- *Functional* and *non-functional properties* of data, processes, and actors, together with *workflow steps*, *patterns*, and *templates*.
- Multi-stakeholder *objectives*, *views*, *regulations*, *hard and soft constraints*. Such objectives may be procedural in nature and/or temporally extended, potentially requiring linear-temporal logic [28] or some other similarly expressive modeling language.
- *Priorities* over soft constraints, rules and objectives that enable the merging and integration of multiple workflows and/or multi-stakeholder objectives, regulations, and rules.

Reasoning about Compliance with Constraints. In the area of business workflow modeling, it is not uncommon to verify properties of a business workflow specification by appealing to standard software verification techniques. The vision for next-generation data-centric workflow generalizes this along several dimensions. One dimension for generalization is with respect to the type of ***reasoning*** that we are advocating. In particular, rather than focusing solely on *verification* of properties of an existing workflow specification, we are additionally interested in *synthesizing* workflows that comply with constraints either as data-specific customization at execution time, or off-line as a means of customizing generic workflows to specific tasks or environments.

The second dimension for generalization is with respect to the types of ***constraints*** that we would like to reason about. We envision a diversity of constraints including multi-stakeholder goals and and objectives. These goals and objectives may be things that we wish the workflow to achieve following execution, or perhaps things that we would like the workflow to maintain or avoid during execution—such as safety or maintenance constraints. One such example is regulatory constraints such as those found within digital governments.

Within these two dimensions of reasoning about compliance with constraints there are many shared challenges. These include reasoning about rich potentially temporally extended constraints, or reasoning with conflicting constraints as may be the case when dealing with conflicting multi-stakeholder objectives or when trying to merge workflows from multiple interacting parties, jurisdictions, or application areas. We may also wish to reason with workflows at multiple levels of abstraction, whereupon the workflow synthesis problem may be one of instantiating, realizing or customizing an abstract workflow description with respect to a particular situation.

Reasoning about Workflow Integration. While most workflows are designed as stand-alone procedures, there is often a need to compose or integrate such stand-alone workflows as different agencies, jurisdictions are stake holders attempt to combine forces and work together towards some objective that requires a diversity of expertise. A good example is “Incident Command Systems” (ICS) [73] that were the result from the need to manage rapidly moving wild fires in the early 1970s. ICS is used by many agencies including fire agencies, law enforcement agencies, and public safety organizations. Due to the unpredictable nature of fire incidents, ICS needs to quickly integrate workflow/software systems under multiple jurisdiction to direct, control, and manage various activities. A key desirable property is to be able to verify correctness of integrated workflows.

Such merging or integration of workflows requires advanced automated reasoning to ensure that the aggregation of these workflows stills achieves the goals of the individual workflows, to ensure that in places whether there may be conflicting objectives or outcomes that these are identified and resolved, and to ensure that appropriate regulations are respected.

Reasoning about Process Evolution and Improvement. Workflows (schemas) are hardly static entities. They evolve and change over time as the result of changes in stakeholder objectives, goals, and regulations; changes in the environment; and as a result of deeper understanding of what will lead to best practice. Current workflow management systems have provided limited flexibility in handling changes, including unanticipated and just-in-time changes. Providing workflow flexibility to support foreseen and unforeseen changes has been a very active research area. It is widely recognized that runtime process execution flexibility is crucial for managing business process lifecycle and it needs special attention [75, 65, 74]. The real challenge in managing dynamic workflow execution is to provide a systematic and integrated support for the process designer to specify how the process would react to various runtime change s and for the process to evolve gracefully in a controlled, incremental, and predictable manner.

Designing a good workflow is itself an evolutionary process. While a workflow designer attempts to capture best practice at the outset, the optimization of a workflow may require both static analysis of the specification as well as analysis of the operation of the workflow execution in practice. Automated reasoning should support both the automated adaptation and optimization of workflow based on automated analysis. It should also provide a fundamental support for workflow evolution as the result of changes in objectives, regulations, or the environment that the workflow is operating in.

Reasoning in Aid of Decision Support. A final reasoning challenge is that of reasoning in aid of decision support. Automated reasoning can aid in many aspects of decision support that improve activities. For example, in deciding a brain surgery operation for a trauma patient the doctor could be greatly helped with the information about the outcome of the operation on past *similar* patients, where similarity also includes their treatment workflows. The general research area of decision support is broad often drawing on a diversity of techniques from artificial intelligence. Here the focus is on the narrower view of decision support specifically as it services workflow.

Monitoring and Querying Workflow Execution and State. A number, though not all, of the reasoning challenges articulated above focus on offline reasoning as opposed to reasoning about the workflow at execution time. An important component of next-generation data-centric workflow is the ability to monitor and query workflow execution and workflow state at runtime. Concerning monitoring and querying, it is worth pointing out that a key advantage of data-centric workflow modeling approaches. Traditional workflow modeling approaches focuses on tasks and control flows. As a result, languages to query ongoing execution and traces are mostly constrained to tasks (names) and sequences of their executions [5, 25]. Many workflow management systems, e.g., YAWL [70], manage the data involved in the workflow in an ad hoc manner and specific to their systems, developing general querying formalisms is rather difficult if it is possible at all. In contrast, data-centric workflows model data using relational or XML data modeling methods [41, 77] addition to tasks and control flows, such workflow management systems can easily support some adaptation of known formalisms for relational or XML databases [77] to query execution snapshots. A future issue is to integrate formalisms for snapshots and traces.

4.4 System issues (design and runtime)

In a global economy driven by the Web revolution, complex workflows based on the exchange of data by autonomous systems (organizations) are becoming more and more common. The challenge is to be able to provide workflow execution with tools for manipulating data and properly interoperating with data management tools.

With the current technology, the design, deployment and management of complex workflows are becoming extremely costly, error prone and unsatisfactory in terms of quality of service and satisfaction of users. A reason that is regularly brought forward is the separation between workflow systems that mostly ignore data and the logic of the information on one hand, and the database systems that ignore the sequencing of tasks and the logic of the application on the other hand.

Formal models for data-centric workflow. The success of relational data management systems is attributed in a large part to the technical models for data, operations, transactions, evaluation plans, etc. that allowed the development of query optimization and transaction management techniques. From a modeling viewpoint, a first challenge is to develop a rich and flexible data-centric workflow model as a basis for the management of both data and control. The role of the models is to provide understanding and access to all facets of information that are dealt with in the application, including:

- All kinds of data structured or not, domain knowledge, e.g., in the form of ontologies or resource descriptions in RDF, as well as information about data life cycle, time and provenance of data.
- Information about flow of control, sequencing and about the rules guiding them or static or dynamic constraints that are imposed and their level of priority.
- Information about the *performers* or *actors* in the most general sense, e.g., people and groups with their access rights, software resources with their API, available services or machines.

The models should also facilitate the reuse of existing resources and in particular data legacy systems. This is particularly important since data generated by workflows tend to greatly outlast the workflows that generated them and are commonly reused by new workflows and new applications. The models should also be simple and formal to facilitate reasoning about the runs of an application (analysis, verification, diagnosis).

We next consider different key issues related to the use of such a model. The important discussion on reasoning was presented in Section 4.3. To structure the discussion, we distinguish between aspects related to design time and to runtime, although in our view this distinction should be blurred.

Design support: simplify the design of such applications. The goal is to simplify the design of workflow applications for non specialists, e.g., for doctors in the health care domain. This is first a Human Computer Interaction (HCI) problem and it has to rely on HCI Interface technology. Such a design is greatly facilitated if the model is simple, intuitive and combines the different facets of information (workflow, data, performers, rules, requirements, etc.) in a coherent framework. In the same spirit, the model should rely on declarative specifications that can be easily specified and tested at design time. For such tests (e.g., consistency testing) to be effective, it is necessary to be able to automatically reason about applications.

In a typical application scenario, workflow development does not start completely from scratch. The reuse of previously designed workflows as well as the interoperability with already running database, workflow management, and other software systems should be supported (ease of creating, reasoning about, and modifying interoperations). Last but not least, the user should be able to also specify and verify at design time non-functional aspects such as quality of service, reliability, security, or access control.

Runtime support: Efficient and maintainable execution. This is clearly an essential aspects, to some extent the heart of the system. All the information from the application should be accessible at runtime with appropriate search and query facilities for both the data and the workflow execution information (including states, resource uses, correlation with other workflow execution instances, etc.). In particular, the system may also help monitor the application via efficient query subscription mechanisms. Besides direct support for functionalities of the application, such a surveillance has a large spectrum of use including gathering statistics towards optimization, system tuning, error detection. Also, an essential aspect of runtime support is the optimization of resource utilization. In this context, in particular in presence of large volumes of data and/or of intense communications, optimization should combine data management techniques (access structures, query optimization, etc.) and workflow techniques (sequencing, bundling service calls, etc.)

Runtime support: Information gathering capabilities. In many workflow applications notably in scientific workflows, the value of the application resides for a large part in the data it is gathering including provenance and time information. Thus, another key aspect of runtime support is the acquisition of data. We are in particular referring here to the data that will serve to the analytic/mining tools and clearly the quality of the analysis will strongly depend on the quality of the data that was acquired (clean, well-structured, and with appropriate meta-data). We do not necessarily need to distinguish here between data that is available at runtime or data that is simply archived.

Partial traces of runs are also important information for instance to be able to explain why certain situations occurred, e.g., for error diagnosis. In many applications, one should also keep track of previous specifications for both the data, workflow and external services. Such a historical archiving of workflow information may be compulsory for legal reasons. For instance, in government applications, the application changes often typically because of new laws or regulations. We still need to be able to understand the data and its evolution, and sometimes we may even have to retroactively modify it.

Runtime support: more dynamicity and runtime improvements. We want to stress here the importance of very dynamic specifications. We already mentioned that government applications tend to change rapidly. In fact, rapid specification evolution is also commonly found in other application areas. In health care, treatment protocols are often modified to reflect better the changes in population, in understanding of effect of treatment, in technology, etc. Generally, workflow is often at the center of the activity of a community of users and reflects the evolution of their needs (e.g., new financial products in finance, new experiments in science). The structure of the data changes, typically by acquisition of new facets for existing collections or by acquisition of new collections. The structure of the workflow as well. It is important to note that this is defeating the traditional separation between design and runtime phases. The two phases may alternate at a reasonably rapid pace and even possibly overlap.

In some applications, the evolution of the application happens automatically. This may be seen as an adaptation to the environment, e.g., a new Web service is substituted to another non-answering one. This may also result from new knowledge, e.g., acquired by the analytic tools. For instance, the system may integrate a new resource (e.g., a new reported set of experiments) or a new performer (e.g., a new business partner) that has been discovered and is considered relevant.

Finally, we have to consider the management of "rainy day" or "exceptional" situations (yet another form of adaptation to the environment). In workflow application development, typical modeling methodology captures a majority of expected runtime execution cases, i.e., "sunny day" scenarios, in a workflow schema (e.g., typical reimbursement workflow with all receipts supplied). This is primarily due to a combination of factors. (1) Not all possible execution scenarios can be known at the design time. Such a situation may arise because of the system conditions (serious disk failure, network partitioning, external attacks such as denial of service). The exceptional situation may also arise due to a world disaster, natural (e.g., storm, earthquake) or artificial (e.g., war, nuclear accident). (2) In many applications, the frequencies of expected execution scenarios have a skewed distribution with a small number of scenarios occurring very often and a large number of scenarios occurring infrequently or rarely. This is the so called "long tail" phenomenon [4]. In practice, design time decision often chooses to ignore the infrequent and rare scenarios to reduce the design time and cost, and leaving these scenarios to handle at runtime as they occur in ad hoc manner.

When such unplanned situations occur, the software system typically simply stops functioning or behaves erratically and slows down the relief teams. The application should continue functioning as best as possible, tolerating violations of "soft" laws and protecting the "hard" ones, for instance, maintaining a line of command (for the army, government or large organizations in general). It remains to be seen whether and how much workflow system design techniques can help dealing with execution scenarios not specified in workflow schema.

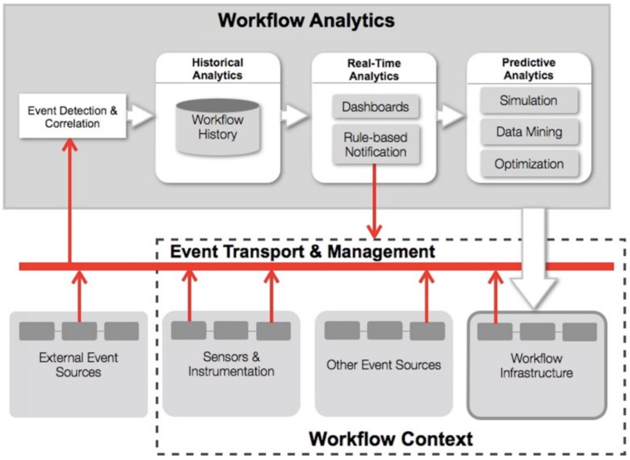

Figure 1: Conceptual elements of analytics and relationship to workflow

4.5 Workflow analytics

Workflow analytics provides the stake holders of a process with information that allows these stake holders to make informed decisions about issues related to the process. As an important class of the issues, workflow improvement focuses on changing the structure or the instrumentation of workflows based on insight gained from the application of workflow analytics and discovery. Workflow analytics has three possible applications: Understanding the past, managing the present, and predicting the future.

- The first application area (historical analytics) is focused on analyzing historical traces of process execution (e.g., audit trails or log files) in order to understand their behavior and what may have influenced this behavior. A typical application would be an audit of a completed process to check whether it complied with applicable rules and regulations. Another application would be the use of data mining techniques to discover patterns that relate the values of certain workflow instance properties (e.g., cycle time or execution paths) to attributes of the data that is being processed by these workflows.

- The second application area (realtime analytics) is focused on studying workflows as they are being enacted with the intention of influencing the course of action should the need arise. In the business domain this area is known as Business Activity Monitoring [49]. A typical application would be the supervision of a data collection workflow that depends on the availability of sensors that are not controlled by the workflow management system itself.

- The third application area (predictive analytics) [54] is focused on discovering pathways to forecast and optimize the behavior of current and future workflow instances.

Figure 1 shows the three analysis tasks and other conceptual components in workflow analytics.

The main sources of analytics information to date are event logs or streams that represent state changes in workflows, data, actors, and their environment. Historically the main source of events was the workflow infrastructure itself (e.g., in form of process logs). However, scientific instrumentation such as sensor systems, medical devices and other information systems constitute an increasingly large source of events, and few of these sources are process-aware.

While events are a raw source of data, different workflow stake holders have different views on events, both in terms of focus and aggregation. For instance, a systems administrator may be focused on infrastructure events such as changes in network latency or server utilization, a laboratory manager may be more interested in changes in inventory levels or the availability of research instrumentation, and an executive may focus on changes to cash levels.

A large portion of the work to-date on workflow analytics has centered around *business intelligence* (or BI), which refers to technologies, tools, and practices for collecting, integrating, analyzing, and presenting large volumes of information to enable better decision making. Today's BI architecture typically consists of a data warehouse (or one or more data marts), which consolidates data from several operational databases, and serves a variety of front-end querying, reporting, and analytic tools. The back-end of the architecture is a data integration pipeline for populating the data warehouse by extracting data from distributed and usually heterogeneous operational sources, cleansing, integrating and transforming the data, and loading it into the data warehouse. Since BI systems have been used primarily for off-line, strategic decision making, the traditional data integration pipeline is a one-way, batch process, usually implemented by extract-transform-load (ETL) tools. The design and implementation of the ETL pipeline is largely a labor-intensive activity, and typically consumes a large fraction of the effort in data warehousing projects. Increasingly, as enterprises become more automated, data-driven, and real-time, the BI architecture is evolving to support operational decision making. This imposes additional requirements and tradeoffs, resulting in even more complexity in the design of data integration flows. These include reducing the latency so that near real-time data can be delivered to the data warehouse, extracting information from a wider variety of data sources, extending the rigidly serial ETL pipeline to more general data flows, and considering alternative physical implementations.

We now briefly discuss several research issues in workflow analytics.

Stakeholder-Aware Analytics. Currently, it is difficult to relate individual events to stakeholder objectives. Most events are technical in nature and need to be interpreted and/or aggregated to provide actionable information. At the same time stakeholder objectives may cut across multiple processes and are rarely formalized in such a way that they can be related to the workflows that may affect these objectives. Since it is currently very difficult to integrate events that originate from the workflow infrastructure with events that originate from non-workflow-aware sources it is very hard to provide stake holders with comprehensive information necessary for informed decision-making. It is desirable to develop languages, methods and tools for workflow analytics that cater to the needs of diverse workflow stake holders.

Inferring Causality from Individual Events. Events only record the results of state changes, but not the reasons why the state changes were initiated in the first place. This is due to the fact that there currently are no standard ways to document the context in which a process was initiated or executed. Causality inference may also be helped by data-centric modeling approaches since workflow specification under these approaches includes data and fairly complete business logic. It is conceivable that incorporation of provenance into data-centric workflow may lead to significantly better ways for causality analysis. In addition, event formats are largely not standardized: Integration of non-workflow events and workflow events require some sort of correlation.

Compliance of workflow to rules and regulations. In the current practice, compliance enforcement is a difficult task. There are three possible causes for this: (1) Rules are typically expressed in a declarative fashion and may apply to several workflows at the same time. (2) Policy statements are typically separated from implementation guidelines, leading to a potentially large number of implementation alternatives. (3) Lack of data in traditional workflow modeling languages significantly limits the ability for meaningful compliance checking. Moreover, once rules and regulations are transformed into implementable guidelines the original intent of the policy may be no longer apparent.

Context-Aware, Open-world Workflow Model. It is hard to understand how context changes affect a workflow. Workflows are typically designed under a closed-world assumption: Workflows are unaware of context-changes that may impact them. For example, when FAA issues new pilot training regulations, airlines need to update training processes. It is needed to find out which processes are affected and how should they be changed. Extensions to data-centric modeling approaches including methods, and tools are needed to discover, model, and integrated the context of a workflow with the workflow specification itself. It is more desirable to develop workflows that are not only context-aware, but also can adjust to changes in their environment.

Predictive Analytics. Predictive analytics arose from insurance and financial industries to assess, e.g., risks. While prediction may employ statistics, machine learning and other techniques, the support for prediction intimately relies on relevant data being available. To this end, techniques and tools are needed to allow us to effectively support accurate prediction of the likely future states of running workflows.

Process Mining. Not all application scenarios may provide process-centric sources of raw analytics data. In these cases workflow discovery techniques can be applied to uncover workflow structures from existing traces. A typical example would be the standardization of an experimental workflow based on the observation of several manually performed instances of this process. The term "process mining" has been used in recent years to refer to the problem of discovering/re-constructing workflow schemas from the workflow execution logs [2, 68]. Existing process mining systems mostly produce activity based workflow schemas such as Petri nets, extensions have been developed to discover, e.g., organizational structures [68]. However, it remains unresolved whether any of these algorithms can be extended to uncover data-centric workflows that can be used to support workflow analytics.

4.6 Workflow interoperation and collaboration

Workflow interoperation remain in high demand in businesses and application domains. Business collaboration among different departments and organizations is an emerging trend in industries to stay competitive in the global market. For example, according to a 2011 Forrester[5] report the online retail sales in US alone reached $176 billions in 2010, many leading retailers such as Amazon.com employ customer shopping workflows that interoperate with workflows from other vendors, warehouses, shipping companies, etc. Enabling interoperation between workflows, and between web services, is highly desired in workflow practice.

Business collaboration may include both internal partners within their organizations and external partners. Typically, workflows interoperate in a distributed environment involving multiple parties with dynamic availability. Two traditional approaches to this challenge are orchestration and choreography [59]. Orchestration tackles interoperation by essentially creating a new workflow schema to fit with and direct the various workflows or services that are to interoperate. Business Process Execution Language (BPEL) [11] is commonly used for programming the orchestrator. A key weakness in this approach is that orchestrators often become the primary controllers of the interoperation, and as a result reduce the autonomy of the different stake holders (individuals and organizations) in achieving their portions of the aggregate goal. Also, the ad hoc nature of orchestrators limits opportunities for re-use. On the other hand, choreography embraces the autonomy of the stake holders, and attempts to enforce the achievement of aggregate goals by restricting how messages can be passed between the stakeholder workflows or services. WS-CDL [76] is a protocol that defines choreography by focusing on message sequence interaction and using connectors to link services across organizational boundary. A weakness of choreography, however, is the lack of a single conceptual point or "rendezvous" where stake holders can go to find current status and information about the aggregate process.

In spite of these advances, tools and support for continue to be a major challenge in current and future enterprise [48]. Interoperating workflows often involve multiple participants, and multiple resources spread

[5] `www.forester.com`

over multiple administrative domains. Typically, these workflows are complex in terms of process logic, relationships among participants and resources, distributed execution, and semantic mismatches between participant data, ontologies, and behaviors. Such complexity is the source of many technical difficulties in design, analysis, realization, execution, and management of interoperations. In the following, we discuss several aspects of the challenge unique to interoperation.

Modeling interoperative workflow. An interoperating workflow models a collaboration among multiple participants. The logic of an orchestrator may be specified with high level programming language (e.g., BPEL). A choreography may be specified as a state machine representing message exchanges between the participants or permissible messages sequences among them with FIFO queues, or as a process algebra expression with sequence, parallel, conditional, and loop constructs. It may be specified in individual pieces using patterns, as a composition of message interactions, or implicitly through participants behaviors. In this end, WS-CDL proposes an XML-based package for specifying choreography through a conventional set of control flow constructs over messaging activities.

Existing languages for specifying orchestration and choreography either have low level of abstraction (a significant obstacle for reasoning) or are particularly weak in modeling participants and data involved. Concerning participants, existing languages assume a fixed number of participants (types) and makes no distinction between a participant type and a participant instance. For example, an ORDER workflow instance may communicate with ***several*** VENDOR workflow instances at runtime; this cannot be effectively captured and managed without making the type and instance distinction explicit. Existing languages either do not support instance level correlations, or lack the ability to reference correlated instances. Therefore, an interaction workflow modeling language must be able to model correlations between workflow instances and manage them at runtime.

Also fundamental to interoperation is the modeling of data. Data for interoperation can be divided into several kinds. The most obvious kind is the data directed needed for the business logic of the interoperation (e.g., shopping cart). The second kind of data is the states of execution by the participants, these may be essential in representing the execution state of the interoperative workflow. The third kind of data is the states of resource usage (e.g., cargo space reserved on a delivery truck). And of course, instance correlations mentioned in the above are also among the data that need to be modeled, accessed, and managed. In data modeling, there are at least two aspects to consider. A piece data may be produced by one participant (a workflow instance) and consumed by another. The workflow model for participants should clearly indicate the data for interoperation. Secondly, the lineage or provenance of the data during execution may sometimes play important role in determining responsibilities. Existing languages pay very little attention to data modeling. WS-CDL models data through variables that only implicitly associated with participants that produce data. Developing suitable modeling frameworks for interoperative workflow is an interesting and challenging problem, data modeling tools (such as semantic data models, dynamic integrity constraints, schema mapping techniques) developed in the database community may find their new use here.

Runtime management. This includes a runtime execution model and functions for managing the executions, including the states and enforcement of constraints (both local to individual participants and global). With the orchestration approach, the orchestrator essentially controls the execution and the execution model is generally clear. When the interaction is specified as a choreography, a first task is to translate the choreography into an execution model. This is called the realization problem in the literature [12], exiting solutions are only for choreographies with no data nor instance correlations and they need to be significantly extended.

Managing executions of interoperative workflow has additional difficulties primarily due to the lack of the ability to see the full snapshot at runtime. For example, when a participant's local resource becomes unavailable or behaves erroneously, the information may not be visible outside of the participant (the orchestrator or other participants). Clearly, an immediate issue is to develop a framework that can capture and manage execution states for interoperative workflow instances. Distributed solutions may have to be developed in cases when, e.g., the government laws/policies prohibit accessibility to portions of the state data.

Interoperative workflows may also have additional constraints for the execution. Effectively enforcing these constraints is necessary, since violations can lead to unexpected outcomes. The constraints may involve multiple participants and/or temporal features. A sound enforcement mechanism could combine constraint checking algorithms with some coordination schemes. This general issue has not been explored extensively.

Service level agreements (SLAs). Non-functional properties (such as the ones concerning execution time, reliability, privacy) are an important aspect of workflow. In the services computing community, the study on QoS (quality of service metrics) for composite services was explored but commonly accepted frameworks are yet to emerge. Formulating and maintaining such properties are more difficult for interoperative workflow due to the nature that they often span cross the boundaries of organizations and/or administrative units.

A SLA provides a contractual guarantee and it mostly concerns non-functional parameters. SLAs are especially important due to the larger number of participants involved. In this case, A SLA should also provide suggestions on distribution of responsibilities for each of the parameters so that both credits and faults can be shared by the participants in proportion to their shares of responsibility. It is also possible that the specific distribution of responsibility is determined at runtime and varies from instances to instances. Modeling constructs and runtime techniques are highly desired to support SLAs for interoperative workflow.

5 Cross Fertilization

Application of workflow management is widely spread in many sectors, including but not limited to: government agencies, business (travel, retail, and others industries), service providers (both traditional and newly arising class of information services), data-intensive sciences, healthcare organizations. The application needs and the availability of rudimentary computing systems (hardware and software for office needs and for managing data and files) have been the driving force for enterprise and organizations to develop their own workflow management systems. A good example of this is the NSF's fastlane system.

However, the workshop finds that

> *The application areas of business, digital government, healthcare delivery, and scientific workflow face many common and overlapping problems, but are developing paradigms, techniques and tools largely in isolation.*

For example, the business process management community needs tools for managing transformational workflows that analyze business process performance, but seems largely unaware of recent advances in scientific workflow. As another example, the notions of "business artifact" and "business object" in the business community, the notion of "record-centric workflow" in digital government community, and the notion of "cases" in healthcare delivery, are highly related and yet the communities are developing frameworks and technologies for these largely in isolation. Furthermore, there is a lack of forums that focus on the overall issues in management of data, workflow, and resources and attract researchers and practitioners from the different communities. For example, related research conferences are typically centered around application topics (e.g., digital governments, health informatics, business enterprise, scientific workflow).

The workshop thus recommends that

> *Cross-fertilization must be fostered between the research communities that are studying workflow in different application areas.*

Since the core technical issues in workflow management concern development of special purpose software systems for managing data and business processes, it is vital for the data management and software engineering research communities to play the central role of building collaborative teams that involve researchers from one or more application areas. This will lead to a faster and deeper understanding of the fundamental issues shared by the application areas, and enable quicker dissemination and adoption of techniques from one domain to another. It is expected that the increase of such cross-area collaboration will naturally lead to forums for exchanging technical knowledge by researchers from different communities.

To foster cross-area collaboration, NSF is well positioned to play a leading role using its existing or new mechanisms. For examples, the GOALI program could be leveraged by supporting match-making between academics and industrial and government applications, starting new joint programs or augment existing ones with the emphasis on workflow management between NSF and other federal agencies (AHRQ, NIH, NIST, DOD, etc.). Workflow research is very active in EU (in the BPM community), fairly active in Australia (collaboration between Australian and European researchers), arising rapidly in China (an annual national conference started in 2009). NSF can steer its international programs to include support for collaborative team building in workflow research, for example, joint programs with EU, Australia Research Council (ARC), and Natural Science Foundation of China (NSFC) to support travel and short-term visitors.

Acknowledgements

NSF grant IIS-0842993 which directly supported this workshop. Hull and Su are also grateful for partial support by NSF grants IIS-0415195, CNS-0613998, and IIS-0812578 in connection with organizing this workshop.

References

[1] S. Abiteboul, L. Segoufin, and V. Vianu. Static analysis of active XML systems. In *Proc. Intl. Symp. on Principles of Database Systems (PODS)*, pages 221–230, 2008.

[2] R. Agrawal, D. Gunopulos, and F. Leymann. Mining process models from workflow logs. In *EDBT*, pages 469–483, 1998.

[3] A. Agrawal et al. WS-BPEL extension for people (BPEL4People), Version 1.0. `http://xml.coverpages.org/BPEL4People-V1-200706.pdf`, 2007.

[4] C. Anderson. The Long Tail. *Wired*. `http://www.wired.com/wired/archive/12.10/tail.html`, Oct. 2004.

[5] C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying business processes with BP-QL. *Information Systems*, 33(6):477–507, 2008.

[6] C. Beeri, A. Eyal, T. Milo, and A. Pilberg. Monitoring business processes with queries. In C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, and E. J. Neuhold, editors, *VLDB*, pages 603–614. ACM, 2007.

[7] K. Bhattacharya, N. Caswell, S. Kumaran, A. Nigam, and F. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems Journal*, 46(4):703–721, 2007.

[8] K. Bhattacharya, R. Hull, and J. Su. A Data-centric Design Methodology for Business Processes. In J. Cardoso and W. van der Aalst, editors, *Handbook of Research on Business Process Management*. Information Science Reference (and imprint of IGI Global), Hershey, PA, USA, 2009.

[9] O. Biton, S. C. Boulakia, S. B. Davidson, and C. S. Hara. Querying and managing provenance through user views in scientific workflows. In *ICDE*, pages 1072–1081, 2008.

[10] S. Bowers and B. Ludäscher. Actor-oriented design of scientific workflows. In *Int. Conf. on Concept. Modeling*, pages 369–384, 2005.

[11] Web Services Business Process Execution Language Version 2.0. `http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html`, April 2007.

[12] T. Bultan, X. Fu, R. Hull, and J. Su. Conversation specification: A new approach to design and analysis of e-service composition. In *Proc. Int. World Wide Web Conf. (WWW)*, May 2003.

[13] P. Buneman and W. C. Tan. Provenance in databases. In *Proc. ACM SIGMOD Conference*, pages 1171–1173, 2007.

[14] D. Calvanese, G. De Giacomo, R. Hull, and J. Su. Artifact-centric workflow dominance. In *Proc. Int. Conf. Service-Oriented Computing (ICSOC)*, pages 130–143, 2009.

[15] T. Chao et al. Artifact-based transformation of IBM Global Financing: A case study. In *Intl. Conf. on Business Process Management (BPM)*, 2009.

[16] A. Chapman, H. V. Jagadish, and P. Ramanan. Efficient provenance storage. In J. T.-L. Wang, editor, *SIGMOD Conference*, pages 993–1006. ACM, 2008.

[17] A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi. Storing and querying scientific workflow provenance metadata using an rdbms. In *E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, pages 611–618, Washington, DC, USA, 2007. IEEE Computer Society.

[18] P.-S. Chen. The entity-relationship model — toward a unified view of data. *ACM TODS*, 1(1):9–36, 1976.

[19] E. Damaggio, A. Deutsch, and V. Vianu. Artifact systems with data dependencies and arithmetic constraints. In *Proc. Intl. Conf. on Database Theory (ICDT)*, 2011.

[20] E. Damaggio, R. Hull, and R. Vaculín. On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles. In *Intl. Conf. Business Process Mgmt. (BPM)*, 2011.

[21] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *Proc. ACM SIGMOD Conference*, pages 1345–1350, 2008.

[22] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson. Data integration flows for business intelligence. In *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, 2009.

[23] H. de Man. Case management: Cordys approach, Feb. 2009, `http://news.bptrends.com/publicationfiles/02-09-ART-BPTrends%20-%20Case%20Management-DeMan%20-final.doc.pdf`.

[24] E. Deelman and Y. Gil. NSF Workshop on Challenges of Scientific Workflows. Technical report, NSF, 2006. `http://vtcpc.isi.edu/wiki/index.php/Main_Page`.

[25] D. Deutch and T. Milo. Type inference and type checking for queries on execution traces. In *Proc. Int. Conf. on Very Large Data Bases (VLDB)*, 2008.

[26] D. Deutch and T. Milo. Top-k projection queries for probabilistic business processes. In *Proc. of Intl. Conf. on Database Theory (ICDT)*, 2009.

[27] A. Deutsch, R. Hull, F. Patrizi, and V. Vianu. Automatic verification of data-centric business processes. In *Proc. of Intl. Conf. on Database Theory (ICDT)*, 2009.

[28] E. A. Emerson. Temporal and modal logic. In J. Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 7, pages 995–1072. North Holland, 1990.

[29] R. T. Fielding and R. N. Taylor. Principled design of the modern web architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, May 2002.

[30] F. Flores and J. Ludlow. Doing and speaking in the office. In *Decision Support Systems: Issues and Challenges*. Pergamon Press, New York, 1980.

[31] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. In *SSDBM 2002: Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, pages 1–10, Washington, DC, USA, July 2002. IEEE Computer Society.

[32] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10(3):11–21, 2008.

[33] J. Freire, C. T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo. Managing rapidly-evolving scientific workflows. In *IPAW*, volume 4145 of *LNCS*, pages 10–18. Springer, 2006.

[34] W. Gaaloul, S. Alaoui, K. Baïna, and C. Godart. Mining workflow patterns through event-data analysis. In *SAINT Workshops*, pages 226–229, 2005.

[35] R. Glushko and T. McGrath. *Document Engineering: Analyzing and Designing Documents for Business Infomratics and Web Services*. MIT Press, Cmabridge, MA, 2005.

[36] A. Goh, Y.-K. Koh, and D. Domazet. ECA rule-based support for workflows. *Artificial Intelligence in Engineering*, 15(1):37–46, Jan. 2001.

[37] B. Hariri, D. Calvanese, G. D. Giacomo, R. D. Masellis, and P. Felli. Foundations of relational artifacts verification. In *Proc. Intl. Conf. on Business Process Management (BPM)*, 2011.

[38] T. Hildebrandt and R. R. Mukkamala. Distributed dynamic condition response structures. In *Pre-proceedings of Intl. Workshop on Programming Language Approaches to Concurrency and Communication Centric Software (PLACES 10)*, 2010.

[39] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[40] R. Hull. Artifact-centric business process models: Brief survey of research results and challenges. In *On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008*, Monterrey, Mexico, 2008.

[41] R. Hull, E. Damaggio, R. D. Masellis, F. Fournier, M. Gupta, F. Heath, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P. Sukaviriya, and R. Vaculn. Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In *Proc. of 5th ACM Int. Conf. on Distributed Event-Based System (DEBS)*, pages 51–62, 2011.

[42] R. Hull, E. Damaggio, R. D. Masellis, F. Fournier, M. Gupta, F. H. III, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P. Sukaviriya, and R. Vaculín. Business artifacts with guard-stage-milestone lifecycles: Managing artifact interactions with conditions and events. In *ACM Intl. Conf. on Distributed Event-based Systems (DEBS)*, 2011.

[43] R. Hull, N. Narendra, and A. Nigam. Facilitating workflow interoperation using artifact-centric hubs. In *Proc. Intl. Conf. on Service Oriented Computing (ICSOC)*, 2009.

[44] T. Jin, J. Wang, and L. Wen. Efficient retrieval of similar business process models based on structure. In *Proc. Int. Conf. Cooperative Information Systems (CoopIS)*, pages 56–63, 2011.

[45] D. Koop, C. E. Scheidegger, S. P. Callahan, J. Freire, and C. T. Silva. Viscomplete: Automating suggestions for visualization pipelines. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1691–1698, 2008.

[46] S. Kumaran, P. Nandi, T. Heath, K. Bhaskaran, and R. Das. ADoc-oriented programming. In *Symp. on Applications and the Internet (SAINT)*, pages 334–343, 2003.

[47] M. Linehan. Ontologies and rules in business models. In *Proc. 3rd Intl. Workshop on Vocabularies, Ontologies and Rules for the Enterprise (VORTE)*, 2007.

[48] C. Liu, Q. Li, and X. Zhao. Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue. *Information Systems Frontiers*, 11(3):201–209, 2009.

[49] D. McCoy. Business activity monitoring: Calm before the storm. Gartner Research Report, April 2002. http://www.gartner.com/resources/105500/105562/105562.pdf.

[50] L. Moreau, J. Freire, J. Futrelle, R. E. McGrath, J. Myers, and P. Paulson. The open provenance model: An overview. In J. Freire, D. Koop, and L. Moreau, editors, *IPAW*, volume 5272 of *Lecture Notes in Computer Science*, pages 323–326. Springer, 2008.

[51] D. Müller, M. Reichert, and J. Herbst. Data-driven modeling and coordination of large process structures. In *On the Move to Meaningful Internet Systems: CoopIS, DOA, ODBASE, GADA and IS*, pages 131–149, Algarve, Portugal, 2007.

[52] P. Nandi and S. Kumaran. Adaptive business objects – a new component model for business integration. In *Proc. Intl. Conf. on Enterprise Information Systems*, pages 179–188, 2005.

[53] A. Nigam and N. S. Caswell. Business artifacts: An approach to operational specification. *IBM-Systems-Journal*, 42(3):428–445, 2003.

[54] C. Nyce. Predictive analytics: White paper. American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America, 2007.

[55] Object Management Group. Object Constraint Language: OMG Available Specification, Version 2.0. `http://www.omg.org/technology/documents/formal/ocl.htm`, May 2006.

[56] Object Management Group. Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.0. `http://www.omg.org/spec/SBVR/1.0/`, January 2008.

[57] Object Management Group. Case Management Process Modeling (CMPM) Request for Proposal, September 2009. OMG document bmi/09-09-23, `http://www.omg.org/cgi-bin/doc?bmi/2009-9-23`.

[58] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, R. Greenwood, K. Carver, M. G. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(1):3045–3054, 2003.

[59] C. Peltz. Web services orchestration and choreography. *IEEE Computer*, 36(10):46–52, 2003.

[60] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst. Declare: Full support for loosely-structured processes. In *IEEE Intl. Enterprise Distributed Object Computing Conference (EDOC)*, pages 287–300, 2007.

[61] C. A. Petri. *Kommunikation mit Automaten*. PhD thesis, University of Bonn, 1962.

[62] D. Sackett. Evidence-based medicine – what it is and what it isn't. *BMJ*, 312(71-72), 1996. (`http://www.bmj.com/cgi/content/full/312/7023/71`).

[63] Sarbanes-oxley act. http://uscode.house.gov/download/pls/15C98.txt, 2002.

[64] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva. Querying and creating visualizations by analogy. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1560–1567, 2007.

[65] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. van der Aalst. Process flexibility: A survey of contemporary approaches. *CIAO! / EOMAS 2008*, pages 16–30, 2008.

[66] S. Straus, W. Richardson, P. Glasziou, and R. Haynes. *Evidence-based Medicine: How to Practice and Teach EBM*. Elsevier, 2005.

[67] J. Strosnider, P. Nandi, S. Kumarn, S. Ghosh, and A. Arsanjani. Model-driven synthesis of SOA solutions. *IBM Systems Journal*, 47(3):415–432, 2008.

[68] W. van der Aalst. *Process Mining*. Springer, 2011.

[69] W. van der Aalst, P. Barthelmess, C. Ellis, and J. Wainer. Proclets: A Framework for Lightweight Interacting Workflow Processes. *International Journal of Cooperative Information Systems*, 10(4):443–482, 2001.

[70] W. van der Aalst and A. ter Hofstede. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, 2005.

[71] W. van der Aalst, M. Weske, and D. Grünbauer. Case Handling: A New Paradigm for Business Process Support. *Data and Knowledge Engineering*, 53(2):129–162, 2005.

[72] W. M. P. van der Aalst and M. Pesic. Decserflow: Towards a truly declarative service flow language. In *The Role of Business Processes in Service Oriented Architectures*, 2006.

[73] J. Wang, D. Rosca, W. Tepfenhart, and A. Milewski. Incident command system workflow modeling and analysis: a case study. In *Proc. of Third Int. Conf. on Information Systems for Crisis Response and Management*, 2006.

[74] B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features – Enhancing flexibility in process-aware information systems. *Data & Knowledge Eng.*, 66(3):438–466, 2008.

[75] B. Weber, S. Sadiq, and M. Reichert. Beyond rigidity - dynamic process lifecycle support. *Computer Science - R&D*, 23(2):47–65, 2009.

[76] Web Services Choreography Description Language (WS-CDL), Version 1.0. `http://www.w3.org/TR/ws-cdl-10/`, 2005.

[77] W. Xu, J. Su, Z. Yan, J. Yang, and L. Zhang. An artifact-centric approach to dynamic modification of workflow execution. In *Proc. Int. Conf. on Cooperative Information Systems (CoopIS)*. Springer, 2011.