

Scientific Workflows: Towards a New Synthesis for Information Integration

Bertram Ludäscher*

Dept. of Computer Science & Genome Center

University of California, Davis

ludaesch@ucdavis.edu

Abstract

We argue that the notion of information integration (II) should be broadened to include the emerging paradigm of scientific workflows. To the domain scientist, II often means an “end-to-end” process, consisting of data acquisition, transformation, analysis, visualization, and other steps. Scientific II then requires a new paradigm that combines a data-oriented approach with a suitable process-oriented workflow modeling approach. Our experiences over the last couple of years indicate that scientists have a real need for this type of “holistic information integration”. To deliver on this need, we have to (i) look at scientific data and workflow management issues with the domain scientist’s eyes, and (ii) solve the quite interesting, multi-disciplinary research issues arising from this new perspective.

1 Background

The database research community has a long-standing, impressive research record in techniques and methods relating to data management and information integration (II): From data models (hierarchical, relational, object-oriented, semistructured, XML, . . .), extensions (*e.g.*, active, temporal, spatial, constraint-based) to query languages for those models and their extensions (*e.g.*, SQL, OQL, Datalog, ECA languages, semistructured QLS, graph QLS, XQuery), to schema evolution, schema mappings, transformations, and—last not least—data integration architectures such as data warehouses and mediators, including their related query rewriting and answering algorithms (*e.g.*, for GAV and/or LAV rules), just to name a few. In addition, over several decades now, the DB community also has been able to bring together researchers and research results from different computer science sub-disciplines. Based on this cross-fertilization, the scope of DB research has expanded

*I would like to thank Shawn Bowers, Michael Gertz, Timothy McPhillips, Norbert Podhorszki, and Daniel Zinn for fruitful and lively discussions on information integration and scientific workflows.

and new solutions to data management problems have been developed. DB researchers with at least one foot in another field—whether it be programming languages (*e.g.*, functional programming), logic-based knowledge representation and AI (*e.g.*, automated deduction, logic programming), mathematical logic or complexity theory (*e.g.*, finite model theory), formal languages (*e.g.*, compilers, automata theory), decision support systems (*e.g.*, OLAP, data mining and knowledge discovery), distributed computing (*e.g.*, parallel and distributed databases), or business process modeling (*e.g.*, workflows)—all have contributed to the field and advanced the state-of-the-art and/or scope of DB research. You name it, chances are the database community can rightfully and proudly claim: *Been there, done that!*

The Road Not (Yet) Taken. However, when measuring the real-world impact on various *scientific* communities, *e.g.*, life sciences (say bioinformatics), and other “dash-informatics” disciplines (geoinformatics, ecoinformatics, cheminformatics *etc.*) and computational sciences (computational biology, chemistry, physics, fluid dynamics, *etc.*), then the record is a bit more mixed and there is a sense that more could be done, given the right incentives. While there have been significant research efforts in many areas, *e.g.*, special kinds of data (say spatio-temporal or multimedia), query processing and special access methods (*e.g.*, for streaming data in sensor networks), the practical impact seems to be mainly in databases becoming a commodity in many science disciplines. As computer scientists we often focus our attention on optimizing machine performance, *i.e.*, execution time or memory usage. Ironically, we are not that focused on optimizing the most precious resource in scientific data management and computing: human time. As Ben Shneiderman put it [13]:

“The old computing was about what computers could do; the new computing is about what users can do. Successful technologies are those that are in harmony with users’ needs. They must support relationships and activities that enrich the users’ experiences.”

2 Information Integration – Take Two

The current experience of many scientific users with data management is: It is too hard. Data preparation and “massaging” into the right format is time consuming and tedious at best (although ETL tools help); schema mappings, data integration, and query rewriting all sound good to us, and maybe even to some (diehard) biologists’ ears, but when pressed, none of the latter seem eager to do a second PhD in database theory or work through PODS or ICDT proceedings. There are of course some specialized tools and DBMS-extensions out there for information integration (e.g., GAV/LAV-style); also recent work in model/schema management to get serious about treating mappings as first-class objects might lead to further practical impacts. But from a scientist’s perspective, the problem remains: there are many bits and pieces, some tools, many sophisticated techniques, and a couple of “big solutions” (commercial II-suites come to mind). But no simple, coherent paradigms (much less: adequate tools) that capture the full spectrum of scientific information integration have emerged yet. Indeed, domain scientists often do *not* think of II as having to do with schema mappings, query rewriting *etc.*, as these types of II essentially only restructure the *same* information. Instead, scientists are typically interested in integrating fundamentally *different* kinds of information for answering scientific questions, *i.e.*, such scientific II scenarios don’t even enter the purview of traditional data integration approaches.

Consider, *e.g.*, a systematist who wishes to use both genomic sequence data and morphological data in the process of inferring the evolutionary relationships among organisms [11]: Rather than mapping DNA sequences and morphological data into a common data format (say XML) or common schema (not that useful), different *analysis steps* may be applied to each data source to infer phylogenetic trees. The systematist then may use the assumption that the organisms have only one true set of evolutionary relationships, and that the phylogenetic trees inferred from integrating genomic and morphological data approximate these true relationships. Thus, for scientists, II is less about traditional data integration (though it also plays a role), but about information integration through *analytical data processing pipelines*, so-called *scientific workflows* [10].

For another typical example, consider a bioinformatics workflow, chaining together the steps:

$$d \xrightarrow{0} \text{BLAST} \xrightarrow{1} \text{Motif_search} \xrightarrow{2} \text{TF_lookup} \xrightarrow{3} \text{Fn_lookup} \xrightarrow{4}$$

Starting with a DNA sequence d (0), BLAST finds a list of similar sequences (1); for each of these, a list of recognized sequence motifs is found (2); for each motif, a list of transcription factors (TF) is returned (3); and finally, associated biological functions are determined for each TF (4).

Again, data is “integrated” by applying computational steps (here: sequence alignment) to the input data, searching external databases, *etc.*, while “gluing” the overall process together in the form of an analysis pipeline. Note that the type signature of the above pipeline can be given as

$$\delta \rightarrow [\delta] \rightarrow [[\mu]] \rightarrow [[[\tau]]] \rightarrow [[[[\phi]]]],$$

where δ stands for the type of a DNS sequence d ; $[\delta]$ for a list of those; $[[\mu]]$ for a list of lists of motifs (resulting from `Motif_search`) *etc.* Thus, scientific II via scientific workflows often means chaining together computational/analysis steps, possibly interleaved with conventional data integration and restructuring steps. The resulting data streams correspond to tagged, nested data collections which in turn can be naturally represented as XML streams [11].

3 Scientific Workflows: A New Synthesis of Process and Information Integration

Scientific workflows [10, 14, 9] are now recognized as a crucial top-layer through which scientists interact with the (ideally invisible) middle and lower layers of the cyberinfrastructure “stack”.¹ They facilitate eScience by allowing scientists to model, design, execute, debug, re-configure and re-run their scientific data analysis, management, and visualization pipelines. Scientific workflows can be seen as a new modeling and design paradigm, creating a particular synthesis of data-driven and process-oriented computing. Instead of forcing scientists to use Perl or Python for gluing computational/analytical steps with data transformation and querying steps (e.g., XQuery, SQL), thus proliferating another “impedance mismatch” with databases, in our vision, an integrated framework is created: Individual workflow steps are modeled as *actors* (components), which are chained together in dataflow process networks [8], where actors are seen as independent processes, communicating by exchanging token streams over uni-directional channels. This dataflow/process-oriented view provides domain scientists and workflow designers with a simple, high-level model for thinking about their analysis pipelines. The model is suited for conceptual-level workflow design [2, 12], exposing both task and pipeline parallelism². Complex subworkflows can be nested and different computation models can be combined [6]. We envision (and have partially implemented) the use of *semantic types* (logic constraints in a suitable ontology language) to facilitate and combine “smart” actor discovery, workflow composition, and data integration [1, 2, 3]. Another advantage of workflows over script-based approaches is the ability to automatically record and then query *provenance* information (e.g.,

¹Scientific workflows are the subject of various new workshops, conferences, special journal issues, and research programs (UK eScience, NSF Cyberinfrastructure, NIH Roadmap, BIRN, CaBIG, DOE SciDAC)

²... and data parallelism, via higher-order functions `map`, `fold` *etc.*

token and object dependency graphs) [5], which can then be used by scientists and workflow engineers to “debug” workflow runs, explain results, produces re-runs, *etc.*

Research Questions. From a scientist’s “end-to-end” perspective, II often involves workflow and experiment design, execution, and management (actor reuse, workflow and schema evolution, archival, provenance). Ever more powerful databases can be used to accommodate scientific computing needs [7], but a new synthesis of process and information integration via scientific workflows seems inevitable to satisfy the needs in eScience. Many research questions arise: What are useful computation models? Are Petri nets useful here or too low-level? Should instead Kahn process networks be used and extended to streams of nested data collections? How can we help scientists/workflow designers to create flexible, reusable workflows, yet have optimization potential, *e.g.*, for static analysis (debugging during design), scheduling of parallel and distributed workflows over clusters or grids? Can hybrid types [2] (combining structural and semantic information) be exploited during workflow design and execution (optimizing machine and/or human performance)? How can we do static analysis of scientific workflows that involve both white box actors (query and transformation actors), black box actors (*e.g.*, external services), and intermediate, gray actors (*e.g.*, when structural and semantic types of actor ports are given, but the overall input/output function is left unspecified). Another question is: How to model, comprehensible control-flow intensive workflows (*e.g.*, for exception handling, fault tolerance, *etc.*) in a dataflow model [4]?

For many of these challenges we can build on the numerous results from the database community; but we also need to look at scientific information integration with new eyes.

References

- [1] S. Bowers and B. Ludäscher. An Ontology Driven Framework for Data Transformation in Scientific Workflows. In *International Workshop on Data Integration in the Life Sciences (DILS)*, LNCS 2994, Leipzig, Germany, March 2004.
- [2] S. Bowers and B. Ludäscher. Actor-Oriented Design of Scientific Workflows. In *24th Intl. Conference on Conceptual Modeling (ER)*, Klagenfurt, Austria, October 2005. Springer.
- [3] S. Bowers and B. Ludäscher. A Calculus for Propagating Semantic Annotations through Scientific Workflow Queries. In *Query Languages and Query Processing (QLQP): 11th Intl. Workshop on Foundations of Models and Languages for Data and Objects*, LNCS, Munich, Germany, March 2006.
- [4] S. Bowers, B. Ludäscher, A. H. Ngu, and T. Critchlow. Enabling Scientific Workflow Reuse through Structured Composition of Dataflow and Control-Flow. In *Post-ICDE Workshop on Workflow and Data Flow for Scientific Applications (SciFlow)*, Atlanta, GA, April 2006.
- [5] S. Bowers, T. McPhillips, B. Ludäscher, S. Cohen, and S. B. Davidson. A Model for User-Oriented Data Provenance in Pipelined Scientific Workflows. In *Intl. Provenance and Annotation Workshop (IPAW)*, Chicago, May 2006.
- [6] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludwig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming Heterogeneity – the Ptolemy Approach. In *Proceedings of the IEEE*, volume 91(1), January 2003.
- [7] G. Heber and J. Gray. Supporting Finite Element Analysis with a Relational Database Backend (Parts I–III). Technical Report MSR-TR-2005-49, MSR-TR-2006-21, MSR-TR-2005-151, Microsoft Research, 2005.
- [8] E. A. Lee and T. Parks. Dataflow Process Networks. *Proceedings of the IEEE*, 83(5):773–799, May 1995.
- [9] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience*, 2006.
- [10] B. Ludäscher and C. A. Goble. Guest Editors’ Introduction to the Special Section on Scientific Workflows. *SIGMOD Record*, 34(3), 2005.
- [11] T. McPhillips, S. Bowers, and B. Ludäscher. Collection-Oriented Scientific Workflows for Integrating and Analyzing Biological Data. In *3rd Intl. Workshop on Data Integration in the Life Sciences (DILS)*, LNCS, European Bioinformatics Institute, Hinxton, UK, July 2006. Springer.
- [12] J. P. Morrison. *Flow-Based Programming – A New Approach to Application Development*. Van Nostrand Reinhold, 1994.
- [13] B. Shneiderman. *Leonardo’s Laptop*. MIT Press, 2003.
- [14] I. J. Taylor, E. Deelman, D. Gannon, and M. S. Shields, editors. *Workflows for eScience*. Springer, 2006.