

Research Statement: Zachary G. Ives

September 29, 2008

It is commonly said that we live in the Information Age; however, a more apt description would be the *Data Age*. *Information* implies data that has been made directly relevant to us, whereas in fact we must manually process unfiltered, un-integrated, often irrelevant data. For instance, a Web search leads to sites and database-backed applications that use different terminologies and conventions, based on the requirements and preferences of their sub-communities. Human effort must be expended to “piece together” different facts from different sites. Similarly, financial, business, and scientific organizations often build “vertical” applications with limited interconnectivity.

In general, data is stored in an overwhelming number of warehouses, databases, and text and proprietary formats – each of which has its own local terminology, format, and perspective. This effectively prevents analysts, scientists, or individuals from getting a complete picture of what is known. Moreover, data is not static: it is frequently amended, revised, and replicated – with an extreme of *data streams* being fed from distributed sensors, routers, and servers. Different versions and copies of data inevitably result in inconsistencies. Sometimes such inconsistencies may be resolved through time synchronization, or through curation, cleaning, and refinement; yet at times they are simply irreconcilable due to differences of opinion or local expertise.

My research interests include all of these aspects of data management. My work seeks to make the most relevant data available to users with specific information needs, in a form useful to them. I develop capabilities to integrate, exchange, synchronize, update, and manage versions of heterogeneous, autonomously controlled data on the Web and beyond. I construct systems, architectures, and models that are broadly and generally useful; but whose design is driven by collaboration with users in the sciences (particularly bioinformatics and medicine), where data tends to be rich, readily available, and in need of integration. My four main projects invent new *architectures* for supporting information sharing; new means of *interacting* with users to find the data they need, presented in the form or application in which they need it; and new ways of *learning* how to integrate data from various disparate Web sites and applications.

New Architectures for Data Sharing: ORCHESTRA and Aspen

ORCHESTRA: Exchange of Updates among Collaborating Sites

Scientific collaboration is increasingly reliant on the ability to exchange and integrate data – yet it is also very poorly served by tools today. One problem is that most existing data integration techniques originated in the business world: they focus on providing a single global, consistent, *query-centric* view of data, originating from a central portal. It is assumed that the schema and format of this view can be standardized; that all data can be translated or mapped into this schema; and that the data can be cleaned and unified in a consistent way.

In scientific collaboration, limited resources often prevent centralized community infrastructure such as portals. Standardization is difficult because fields rapidly change and have diverse terminologies and even beliefs about facts. Finally, a great deal of scientific data is uncertain – it consists of *derived* results from curation, cleaning, and analysis, and it is often the result of heuristic algorithms and/or human judgment. Currently, different scientific sub-communities set up their *local* own databases and data warehouses, which present data using their own terminologies, schemas, and curation practices. These warehouses define a representation for data *and* they filter and reconcile data conflicts, according to community standards. The data integration challenge here lies in allowing data to be shared *among and across* these portals, even as this data is constantly updated and corrected.

The ORCHESTRA system [I+08a] (whose architecture was introduced in [IKKC05], and which is being developed with my students Nick Taylor, Todd Green, and Grigoris Karvounarakis, in collaboration with Val Tannen) provides a new, *update- and version-centric* means of sharing data, rather than a query-centric one. Each participant defines

its own database schema and manages the contents of its own data instance; it provides *schema mappings* defining how to translate data and *updates* to and from other participants¹; and it supplements these with *trust mappings*, specifying which sources' data to import, and how to reconcile conflicts. Sites in ORCHESTRA make revisions to their databases, and occasionally publish the revisions in the form of update logs. Other sites *translate* [GIKT07,GI08], assess whether to *trust* (based on the sources), *reconcile* [TI06], and *apply* these published updates. The ORCHESTRA project has developed new techniques for translating and propagating updates *unidirectionally* [GIKT07] as well as *bidirectionally* [GI08]; compactly encoding a record of data's derivation or provenance [GIKT07]²; and redefining concurrency control, transactions, replication, and conflict reconciliation to enable different sites to hold overlapping but different data versions, according to administrators' policies about trust [TI06]. A prototype of the ORCHESTRA system has been implemented and will be released into open source this year. It is, so far as we are aware, the first system to simultaneously address data integration, updates, data versioning, and conflict management.

Ongoing work focuses on improving the performance of our system to scale to hundreds of nodes, leveraging recent work by myself and others on *adaptive* query processing [IHW04, DIR07,IT08] as well as indexing and cost estimation. Todd Green and I are working on how to optimally update data instances when someone changes a mapping. Grigoris Karvounarakis and I are working on indexing data provenance, to support a variety of operations over it, including determining trust, confidence, and derivability. Nick Taylor and I are developing a peer-to-peer query execution architecture that guarantees completeness and recovers from failures.

Aspen: Integrating Heterogeneous Sensor and Stream Data from across the World

In the near future, the spectrum of data sources – and thus the range of data we would like to integrate – will change dramatically, as a plethora of wireless sensing devices are deployed on the Internet. Sensor networks have the potential to create “smart environments,” instrument physical systems, perform monitoring for science, and even revolutionize entertainment. Clearly, information integration and data management will need to expand to incorporate such data; conversely, database queries have shown promise in addressing the challenges of programming large suites of *heterogeneous* sensing devices [MFHH03].

I am leading an effort (in collaboration with my colleagues Sudipto Guha, Boon Thau Loo, and Insup Lee, and students Svilen Mihaylov and Mengmeng Liu) to extend ideas and techniques from data integration and from ORCHESTRA to the realm of *sensor integration*, where streams of complex, heterogeneous sensor data are integrated together and with database data to produce composite results. The main novelty – and most difficult challenge – lies in taking database-style declarative queries (over the data being emitted by each sensor device), and optimizing and executing them across highly distributed, ad hoc wireless networks mixing primitive and sophisticated devices, without up-to-date global knowledge. Our goal is a runtime system that *automatically* partitions computation and communication to meet the needs of the application, while minimizing battery consumption, network congestion, and failure rates. We have begun by implementing and optimizing basic query operators such as joins [MJIG08], using a combination of precomputation, progress monitoring, greedy algorithms, and cost estimation; and fixpoint computation [L+08], which is useful in determining reachability and regions, and which requires a type of data provenance to determine when results should be removed because sensed objects move or disappear. As this work matures, we will move “up the stack” to develop a complete distributed system architecture.

¹ These mappings build upon my past work on peer-to-peer schema mappings for data [HIMT03,HIST03].

² This is based on a formal model proposed by my colleague Val Tannen that was developed in part to our requirements [GKT07].

New Modes of Interacting with Users: The “Q” System

While new data sharing architectures can expand the capabilities available to an expert, a second challenge lies in making a data management system accessible to end users. In many domains – especially those with a multitude of different databases – a typical end-user, e.g., a biologist, can be overwhelmed by the complexity of even individual schemas, and cannot realistically be expected to write SQL queries.

Fortunately, a user with an *information need* for a particular task can often define a *keyword search* describing what he or she is looking for, and can frequently *assess* whether any data returned meets this information need. My work on the Q System [T+08] (in collaboration with Fernando Pereira and Sudipto Guha, and our students Partha Talukdar and Marie Jacob) seeks to “expand” keyword searches into database queries. We observe that in a large domain, each keyword might relate to data from some set of relations; a query might be formulated by choosing one relation from each set, and then joining along foreign keys, schema mappings, and external links to create a *chain query*. However, not every data source or link is equally authoritative or useful: some may be non-authoritative, others might contain dirty data, etc. Q seeks to *learn* which links and sources are most relevant to the user’s information need, and to generalize this for future queries that relate to this information need. We start by matching keywords against schema elements. Then we formulate join queries to “connect” the matching relations. We find the most promising queries, according to a ranking function that will be learned, and provide sample answers to the user. When the user provides *feedback* on individual answers, we use data provenance to learn which underlying queries are of greater interest to this user, and we adjust the rankings of the queries and the relations they use. Our results on bioinformatics data [T+08] demonstrate that this technique is very effective in learning real rankings provided by experts.

Learning to Integrate Data: CopyCat

Finally, one of the greatest challenges in data integration lies in discovering the transformations or operations required to integrate data: learning extractors and schema mappings. Existing extractor generators and schema mapping tools can assist users in creating such extractors and mappings, but the process is highly time-consuming. In some settings – e.g., emergency response, rapid deployments, etc. – this is unacceptable due to time pressures. However, it is often unnecessary to get everything right: some erroneous results are acceptable as long as there is a way to correct them. In a new project with Steve Minton and Craig Knoblock, along with my students Marie Jacob and Partha Talukdar, I am developing a new “best effort” paradigm for learning to rapidly integrate time-critical data, dubbed “Smart Copy and Paste.” In the CopyCat system [I+08b], a non-expert user manually copies data from a variety of Web sites and applications, and pastes it together into an integrated view in a spreadsheet-like “workspace.” As these activities progress, the system attempts to *generalize* user actions and automatically import the remaining data – to “auto-complete.” The system proposes suggested auto-completions, the user provides feedback, and the system refines its generalizations. At the end, data will be exported from the workspace to applications like Google Maps or various logistics systems. An interesting property of this approach, unlike prior work, is that it should never involve more work than manual importing.

Summary

The data integration problem is of pressing importance in today’s data rich world – it has significant ramifications on everything from science to medicine to commerce. By systematically revisiting the techniques developed for the various layers of integration – system architecture, query interface, mapping interface – we can make the process vastly better, helping bridge the gap between data and information. My work continues to make novel contributions in these areas, and I look forward to continuing to explore issues related to scientific applications, inconsistency and conflicts, distribution, and the intersection of learning, approximation, and data management.

References

- [A+06] Madhukar Anand, Eric Cronin, Micah Sherr, Matt Blaze, Zachary Ives. *Security in Sensor Networks: More Interesting than You Think*. Usenix Workshop on Hot Topics in Security, 2006.
- [A+07] Sören Auer, Chris Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *DBpedia: A Nucleus for a Web of Open Data*. International Semantic Web Conference (ISWC), 2007.
- [DIR07] Amol Deshpande, Zachary Ives, Vijayshankar Raman. Adaptive Query Processing. Foundations and Trends in Databases, Vol. 1 No. 1, 2007.
- [GKT07] Todd J. Green, Grigoris Karvounarakis, Val Tannen. *Provenance Semirings*. Principles of Database Systems (PODS), 2007, Beijing, China.
- [GIKT07] Todd J. Green, Zachary G. Ives, Grigoris Karvounarakis, Val Tannen. *Update Exchange with Mappings and Provenance*. Conference on Very Large Databases (VLDB), 2007, Vienna, Austria. Amended version available as University of Pennsylvania Technical Report MS-CIS-07-26.
- [HIMT03] Alon Y. Halevy, Zachary G. Ives, Peter Mork, Igor Tatarinov. *Piazza: Data Management Infrastructure for Semantic Web Applications*. Twelfth International World-Wide Web Conference, Budapest, Hungary, 2003, pp. 556-567.
- [HIST03] Alon Y. Halevy, Zachary G. Ives, Dan Suciu, Igor Tatarinov. *Schema Mediation in Peer Data Management Systems*. International Conference on Data Engineering (ICDE), 2003, Bangalore, India, pp. 505-516.
- [IHW04] Zachary G. Ives, Alon Y. Halevy, Daniel S. Weld. *Adapting to Source Properties in Processing Data Integration Queries*. SIGMOD Conference on Management of Data, Paris, France, June 2004, pp. 395-406.
- [I+08] Zachary G. Ives, Todd J. Green, Grigoris Karvounarakis, Nicholas Taylor, Val Tannen, Partha Pratim Talukdar, Marie Jacob, Fernando Pereira. *The Orchestra Collaborative Data Sharing System*. To appear, ACM SIGMOD Record, 2008.
- [I+08b] Zachary G. Ives, Craig Knoblock, Steve Minton, Partha Pratim Talukdar, Marie Jacob, Rattapoom Tuchinda, Jose Luis Ambite, Maria Muslea, Cenk Gazen. *Interactive Data Integration through Smart Copy and Paste*. Under submission, CIDR 2009.
- [IKKC05] Zachary G. Ives, Nitin Khandelwal, Aneesh Kapur, Murat Cakir. *Orchestra: Rapid, Collaborative Sharing of Dynamic Data*. Conference on Innovative Database Systems Research, Monterey, CA, 2005.
- [IT08] Zachary G. Ives, Nicholas E. Taylor. *Sideways Information Passing for Push-Based Query Processing*. International Conference on Data Engineering (ICDE), 2008, Cancun, Mexico.
- [KI08] Grigoris Karvounarakis, Zachary Ives. *Bidirectional Mappings for Data and Update Exchange*. Accepted for publication, Conference on the Web and Databases (WebDB), 2008, Vancouver, Canada.
- [L+08] Mengmeng Liu, Wenchao Zhou, Nicholas Taylor, Zachary Ives, Boon Thau Loo. Monitoring Regions and Connectivity in Networks. Paper under preparation for ICDE 2009 conference.

- [MFHH03] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong. The Design of an Acquisitional Query Processor for Sensor Networks. SIGMOD Conference on Management of Data, 2003, San Diego, CA.
- [MJIG08] Svilen Mihaylov, Marie Jacob, Zachary Ives, Sudipto Guha. Distributing Stream Joins in Multi-Hop Wireless Networks. Paper under preparation for ICDE 2009 conference.
- [MLIS07] Yun Mao, Boon Thau Loo, Zachary Ives, Jonathan M. Smith. *The Case for a Unified Extensible Data-centric Mobility Infrastructure*. MobiArch 2007.
- [TI06] Nicholas E. Taylor, Zachary G. Ives. *Reconciling while Tolerating Disagreement in Collaborative Data Sharing*. SIGMOD Conference on Management of Data, 2006, Chicago, IL.
- [T+08] Partha Talukdar, Marie Jacob, M. Salman Mehmood, Koby Crammer, Zachary Ives, Fernando Pereira, Sudipto Guha. Learning to Create Data-Integrating Queries. Accepted for publication, Conference on Very Large Databases, 2008.