

Integration Architectures Group

Phil Bernstein

Howard Bilofsky

Michael Carey

Barbara Eckman

Todd Hughes

H. V. Jagadish

Anupam Joshi

Hilmar Lapp

Tom Lee

Bertram Ludaescher

Louisa Raschid

Arnie Rosenthal

A Taxonomy of data integration:
Multiple axes of interest
Systems issues
Privacy, legal, social, economic issues

Engineered vs Ad Hoc

Heavyweight vs lightweight.

Longevity: Long-lived versus one-time.
Long-lived Schema vs Long-lived Data

Deep vs. superficial

Complexity: Simple v. complex
E.g. Column-match versus expensive function

Comprehensive vs Incremental

⇒ heavyweight vs Lightweight

Union v. join
Similarity of data sources, domain overlap

Multimedia integration – dimension on datatypes.
Also streams, and process integration.

Precise vs. approximate/probabilistic

Do we need many technologies or one?

Systems researchers should be looking at the convergence of all these technologies.

SOA is fine as far as it goes, but it does **not** solve the data problem.

Declarative specifications are very good (see Carey talk).

What is the abstract model of the language/system used.

Workflows provide a sequential model of data integration.

They are the “upper ware” of cyber-infrastructure.

Query is a one-step workflow.

Declarative workflow specification leading to optimizations/rewritings

Data integration can be a piece of a larger workflow.

Business process integration is a high level view of integration.

CS Curricula should teach issues of integration, workflow, etc.

Integration in the very large – multi-step from islands.

Federating of federations.

Standards can help – domain-specific data standards.

Declarative specifications can help.

Provenance is crucial.

Spiral model/incremental integration can be multi-step.

Possible exploratory nature – step at a time and see how you are doing.

Workflow evolution

Establishing standards to minimize diversity

Given that there are many ontologies that (vaguely) relate to what I want to say, how do I choose? What do I do?

Overlapping communities of interest.

Schema standards

Create ontologies to query through – need customized ontology “view”

Also ontology/schema summaries for user understandability.

Reusable components (composing/merging/mapping schemas).

Reuse specification

Metadata management system.

What is an artifact of integration? Could be a workflow.
Could be relationship knowledge. Not just the integrated data result.

Integration appliance as a model for imprecise integration.
May lead to easier deployment, adoption; lower price.

CAD framework – development environment for data architects
Methodologies and usage scenarios
Scalable Data and mapping visualizations
Declarative query language
Change Management
Explanations
Debugger
Tools to manipulate artifacts of integration.
UML does not have everything we need

Just-in-time to pre-integrating only links to fully materialized
EII vs ETL (and things in the middle)
Queries are engineered, but the integration is dynamic.
Choice should be autonomic.

Identification issues (globally unique Ids)
Who is responsible, who assigns, ...
Need ids for not just for data objects, but also for vocabulary, ontology, etc.
Managing Evolution/change (for IDs, and for schema/data/spec)
Versioning of data and versioning of transformations.
Particularly hard for provenance.

.
Deal with exceptions to assumptions made
Techniques must be robust, and handle errors gracefully.
Data Cleaning
If equivalent of placing constraint violations in an exceptions table.
Fault-tolerant integration. Run-time vs compile time.
(Logical vs physical faults)
Robustness.

Security/Access control/authorization have to be propagated through integration views.

Making effects understandable.

Benchmark

Repository of large schemas and large mappings that could be used as benchmark.

Sample problems with sample solutions.

Metrics for integration May need an expert evaluation – at least for deep semantics. So usual validation is through community of users.

In x hours how many entities/attributes can you get across to your partner.

Synthetic data sets with schema perturbations to create easier internal evals.

PRIVACY:

There is leakage of data through integration. Detecting this is hard.

May allow temporary leakage if doing adhoc integration.

Complexity of fine grain specification

Specification of policies is really hard

Enforcement is easier (though still challenging)

Audits are challenging over integrated data

May need to know provenance of queries that have been answered

“Double blind” integration when join ID is to be kept private.

LEGAL ISSUES:

Ownership rights over information – what about results of integration products.

Data that is multiply owned when put together.

Liability for errors and incompleteness in integration.

Can corporations certify that they are complying with their stated privacy policies.

SOCIAL ISSUES:

Privacy is theoretically important to people but give it up for small rewards.

Track usage to pay.

Difficulty of citation on the web and with integration.

Archiving of derived products – how to make reproducible.

Automatic tracking of provenance to provide credit.

Integration of multiple viewpoints/policies.

Understanding implications of (privacy/ownership) policies after integration transformations.

Turf battles.

ECONOMIC ISSUES/INCENTIVES:

Who will do what, and make sure they have the incentives to do it.

Lower economic barrier for incremental (and ad hoc??) integration.

Productivity paradox – what is the value of information (integration)? Obtain ROI.

Technologies to reduce risk of failure – support incremental.

Who is funding the cost of building to share?

Fame as incentive.