# Best-Effort Data Integration

Position Statement

AnHai Doan, University of Wisconsin-Madison

This position statement makes the case that, since "exact" data integration is AI complete, we should seriously consider "best-effort" data integration. I will start by briefly summarizing the development of the field of data integration. Next, I discuss best-effort data integration as one of the next logical research directions, then sketch a simple observation that can be leveraged to examine the topic systematically and to understand current work. Finally, I describe current research on the topics at the University of Wisconsin-Madison, and list open questions that we are considering.

## 1 Data Integration: The Past Thirty Years

The field of data integration can be roughly classified as having gone through four overlapping stages (based strictly on my personal perspectives):

**"Group Grope" (up to 1990):** This was when we realized that data integration is an important problem. Numerous solutions were proposed, and the field started to acquire a "feel" for the various aspects of the problem. A clear distinction was also made between application/process integration and data integration.

**Foundational Development (1990-2000):** A clear foundation was laid down for data integration. The mediator model was proposed and gained widespread acceptance. The various components of the model (e.g., wrapper, schema matching, query reformulation, etc.) were identified, and alternative choices for data integration (e.g., GAV, LAV) were proposed. Seminal works include the mediator model proposed by Gio Wiederhold, the Penn, TSIMMIS, and Information Manifold efforts, among others.

**Building on the Foundation (1998-today):** Problems regarding various integration components are intensively studied. "One hundred flowers bloom" for theoretical development, query reformulation, adaptive query processing, schema matching, entity resolution, managing inconsistent data, integrating XML data, managing provenance and uncertainty, developing P2P systems, among many others. Several communities, most notably databases, AI, and Web, join forces to attack these problems.

**The Rubber Meets the Road (1998-today):** The lessons learned from the above stages are applied in a wave of Internet startups. Data integration "branches" out into many application domains, including bio-informatics, geo-spatial domains, hydrology, intelligence analysis, and the Deep Web. The field takes a step back, to gain perspectives.

## 2 Best-Effort Data Integration

In perspective, one of the key lessons we have gained is that data integration is hard, much harder than we thought ("intractable" or "AI complete", as many have said). A major reason for this, I believe, is that so far we have mostly tried to achieve *exact, precise* data integration. This made much sense in the early days, when like relational data management, data integration is targeted primarily at *business data*. The vast majority of applications involving such data (e.g., payroll) clearly require *exact* data integration, and anything less is *not* usable.

Today, "payroll"-like, exact integration applications continue to play a crucial role. However, there are also many emerging application domains where exact integration may not be necessary. A prime example is citation tracking with *Citeseer* or *Google Scholar*. Other examples include personal information management (PIM), scientific data analysis, intelligence analysis, business intelligence, and many integration scenarios on the Web (e.g., *Froogle*). For these applications, since exact data integration is too hard, we should seriously consider developing best-effort integration solutions, which often incur far less cost and already provide very useful services.

One way to develop such solutions is as follows. Take an exact data integration architecture of the foundational period (e.g., the mediator-based one, see above). Next, remove, simplify, or make "less precise" several components of the architecture. Then study how that affects the rest of the architecture, from both algorithmic and usability points of view. For example, building a *structured* global query interface (over which users can pose queries to the data sources) is often very time consuming. One way to simplify this step, thus saving significant human labor, is to assume just a *keyword* query interface. This leads to a best-effort integration architecture where users can ask keyword queries across multiple heterogeneous structured databases.

As another example, removing the wrapper construction step (but keeping a structured global query interface) leads to an architecture where users can execute structured queries (e.g., SQL) over unstructured, often textual databases.

As yet another example, a Web search engine can be viewed as a best-effort integration architecture where no wrapper is constructed, the global query interface has been simplified to a keyword search interface, and – in response to a user query – the architecture only selects, ranks, and returns relevant data sources (i.e., Web pages in this case).

Thus, *a best-effort data integration system can be viewed as an exact data integration system where certain components have been removed, simplified, or made "less precise"*. As demonstrated, this perspective enables us to understand certain current information processing systems (e.g., search engines) from a data integration viewpoint. It also motivates novel best-effort integration problems (e.g., keyword search over multiple databases, SQL queries over text).

## 3 Current Best-Effort Integration Research at Wisconsin

Our current work on this topic focuses on two main questions:

- What types of best-effort data integration systems can we develop and where can they be useful? We consider systems that perform keyword search over multiple structured databases, or SQL queries over text, as mentioned above. We also consider systems that enables both keyword search and SQL queries over imprecise structured data extracted from text.

- How can we go the "last mile"? Specifically, can we leverage user interaction to continuously improve the quality of the "best effort" provided by the system? Can we leverage the entire user community for this purpose, in a mass collaboration fashion?

Much of the above work are carried out in the context of the Cimple project on building a best-effort data integration system for data-rich Web communities (`www.cs.wisc.edu/~anhai/projects/cimple`). See also `dblife.cs.wisc.edu` for a prototype system that we are building for the database research community. In longer terms, we would like to examine the kinds of guarantees we can provide for the performance of a best-effort integration system, as well as consider alternative best-effort integration models.