HealthCatalyst
ignite change

white paper

The Late-Binding™ Data Warehouse Explained

Designing for Analytic Agility and Adaptability

By Dale Sanders, SVP, Health Catalyst

HealthCatalyst
ignite change

The Concept of Data Binding

Data can be "bound" to business rules that are implemented as algorithms, calculations, and inferences acting upon that data. Examples of binding data to business rules in healthcare include

- Calculating length of stay (LOS)
- Attributing a primary care provider to a particular patient with a chronic disease
- Calculating revenue (or expense) allocation and projections to a department or physician
- Data definitions of general disease states for patient registries
- Defining patient exclusion criteria for disease/population management
- Defining patient admission, discharge, and transfer rules

Data can also be bound to vocabulary terms, for both local and industry standards. Examples of vocabulary binding include

- Patient identifier
- Provider identifier
- Location of service
- Gender
- Diagnosis code
- Procedure code

Knowing when and how tightly to bind data to rules and vocabularies is critical to the agility and success—or failure—of a data warehouse. In healthcare, the risks of binding data too tightly to rules or vocabularies are particularly high because of the volatility of change in the industry. Business rules and vocabulary standards in healthcare are among the most complex in any industry, and they undergo almost constant change.

Lessons From Software Engineering

The idea of late binding in data warehousing borrows from the lessons learned in the early years of software engineering. In those early years, very large software programs characterized software development—it was very common to program hundreds of thousands of lines of code in a single module, supporting numerous and widely different business functions. The code for these varied business functions was tightly bound (also known as coupled)

Knowing when and how tightly to bind data rules and vocabularies is critical to the agility and success–or failure–of a data warehouse.

together all at once, at compile time. It was a time-consuming process to write and troubleshoot these large programs. If one piece of the program failed at compile time, the entire program failed. It was all-or-nothing programming. Also, if the program required changes or modifications because of new business rules and requirements after compile time, the entire program had to be modified, re-compiled, and placed back in service, often with significant downtime. Agility suffered enormously.

Object Oriented Programming and Late Binding™

In the 1980s, software engineering practices would change significantly, moving away from large, tightly coupled, early binding programs. Alan Kay from the Universities of Colorado and Utah and Xerox/PARC introduced the concept of late binding and object-oriented programming. This new approach was based upon two radically new concepts: (1) Writing code in smaller modules or objects that were modeled after processes and services in the real world that the software was designed to support, and (2) binding these software objects at run time, not compile time, and **only when those objects were needed** to support the services they reflect.

Alan Kay's new concepts for software engineering sat underutilized and largely unknown, confined to PARC and academic circles, until Steve Jobs founded NeXT. Jobs was not a programmer, but he instinctively understood the elegance of Kay's concepts. Object-oriented, late-binding software engineering became the standard practice at NeXT and paved the way to commercial, large-scale adoption of Kay's philosophies. Steve Jobs receives due credit for his innovation and leadership at Apple, but by making object-oriented, late-binding software a new commercial norm at NeXT, he paved the way for the entire software revolution in Silicon Valley. The agility, scalability, and performance of platforms such as Amazon, Google, Facebook, and Salesforce were enabled by this new approach to software engineering.

Data Engineering and Late Binding™

After witnessing and reflecting upon the failure of several multimillion-dollar data warehousing projects in the US military, Dale Sanders, Senior Vice President for Strategy at Health Catalyst, saw the same patterns in data engineering as those in software engineering prior to object oriented programming. Early and tight binding of data to rules, models, and vocabulary led to unnecessary complexity that delayed time-to-value and led to a very fragile and inflexible data warehouse infrastructure that could not adapt to rapidly changing analytic use cases or new data content.

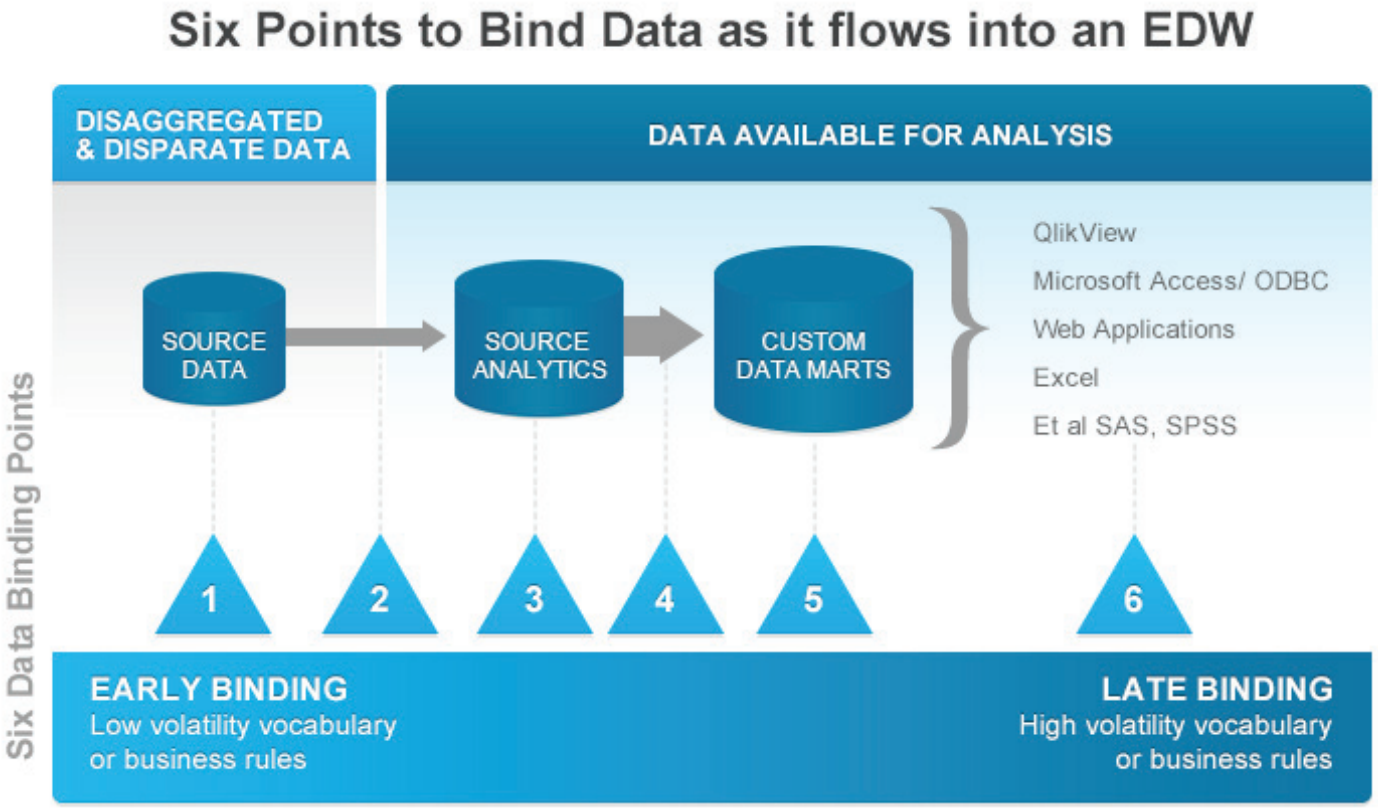
In the late 1990s, while Sanders was employed by TRW Inc., he was sponsored by the Pentagon to study advanced decision support in nuclear warfare operations—a project called the Stratgic Execution Decision Aid (SEDA). He turned to the healthcare industry for what he expected to be role-model examples of computer-aided analytics to drive better decisions in time-critical, life-critical situations but instead found almost no examples, with the notable exception of a scattered few at Intermountain Healthcare in Salt Lake City. Intermountain clearly possessed the culture and willingness to fully leverage data for improving care, but the industry at large was many years behind. Anticipating the eventual demand for analytics in the industry, he made a career transition from the military, national intelligence, and manufacturing sectors into healthcare.

Sanders's late-binding data engineering concept is now fundamental to Health Catalyst's data warehouse platform. The Late-Binding™ Data Warehouse enables

time-to-value that is measured in days and weeks, not months and years, and has proven many times more scalable and adaptable to new analytic use cases and data content than the methodologies that utilize early binding, tightly coupled data models and vocabulary management.

Data Binding Points in an Enterprise Data Warehouse

There are six points in a data warehouse at which data can be bound to rules and vocabularies. As the data flows from left to right in the diagram below, points 1 and 2 are appropriate for binding to rules and vocabularies that exhibit low volatility; that is, those rules and vocabularies that change infrequently, such as patient identifiers and provider identifiers. Late binding—at points 5 and 6—is appropriate for rules and vocabulary that are likely to change on a regular basis, or for which no standard rule or vocabulary exists. For example, binding in the visualization layer is appropriate for what-if scenario analysis that is associated with modeling different reimbursement models or defining disease states. Once that exploratory what-if phase is complete, the new models and definitions can be locked down and bound in points 3, 4, or 5.



It is a best practice to retain a record of the bindings in the data warehouse. This record will allow analysts to quickly run models on rules and vocabulary (e.g., ICD-9 to ICD-10) that change over time, which is helpful for forecasting and predictive analytics. Catalyst recommends embedding the history of vocabulary and business rule binding into the data structures of the data warehouse so that they become a self-contained configuration control library that can easily be used to retrace analytic history when necessary.

Knowing what to bind and when in the flow of data in a data warehouse requires more than technical skills. Data engineers and architects who work in a Late-Binding™ Data Warehouse environment must possess a strategic understanding of the short and long-term evolution of the entire industry. They must appreciate the historic volatility of vocabulary and business rules as well as an ability to predict the velocity and the specifics of volatility in the future. Healthcare is undergoing changes to business rules and vocabulary at an unprecedented rate. Data warehouses must

be designed to keep pace with the market, and the Late-Binding™ architecture has a proven track record of agility and adaptability to new rules, vocabularies, and data content that other designs have not matched.

Data Modeling and Data Binding

Relational data models are inherently binding—they bind data to business rules and relationships. When developing transaction-based applications for capturing data, a data model is an important aspect of the application design and data integrity strategy. When designing a data warehouse, data models can inhibit adaptability to new analytic use cases. Below are the current options for modeling data in a data warehouse, listed in order of progression, from early to late binding.

The Inmon, Kimball, and I2B2 approaches to data modeling are inherently early binding. They require all source system data to be mapped into predefined data models, a process called conformance and normalization. The terms imply exactly what is required—data that was modeled and captured in disparate source transaction systems must conform to a new data model in the data warehouse. While at first this approach might appear reasonable, in practice, it leads to major problems when applied to the healthcare industry.

In analytic environments where data content, use cases, data rules, and vocabulary change infrequently—such as the retail industry where the data model is largely reflected in the simplicity of a transaction receipt—the Kimball and Inmon approaches are adequate. In the healthcare industry where the data environment is much more complicated than a sales receipt and the analytic use cases are constantly changing, these early binding data models can be disastrous in their consequences to agility and initial time-to-value. The process of mapping and conforming data to these early binding models in a healthcare delivery data warehouse typically takes 18–24 months or longer. When new data sources are added to the data warehouse—as occurs in mergers, acquisitions, and ACO partnerships—this lengthy time-to-value is repeated again and again. Likewise, as the complexity of analytic use cases inevitably matures in an organization, the early binding data model must be modified and the source system data must be conformed and mapped again. These early binding data models cannot keep pace with the changes in the analytic environment and the data warehouse subsequently fails to deliver its initial appeal.

File structure association, popularized first by IBM mainframes some 60 years ago, is reappearing in the form of Hadoop, MapReduce, PIG, and NoSQL. Data warehouses based upon this technology exhibit the lowest degree of data binding and coupling. In fact, since there is no data model in this methodology, there is no data binding until the binding is declared between the files of data through associative programming. Data warehouses based upon this technology are very adept at quickly loading data into the warehouse, but the benefit of time gained in loading is more than negated by the complexity of the programming that is required to minimally bind and declare associations between the files and data.

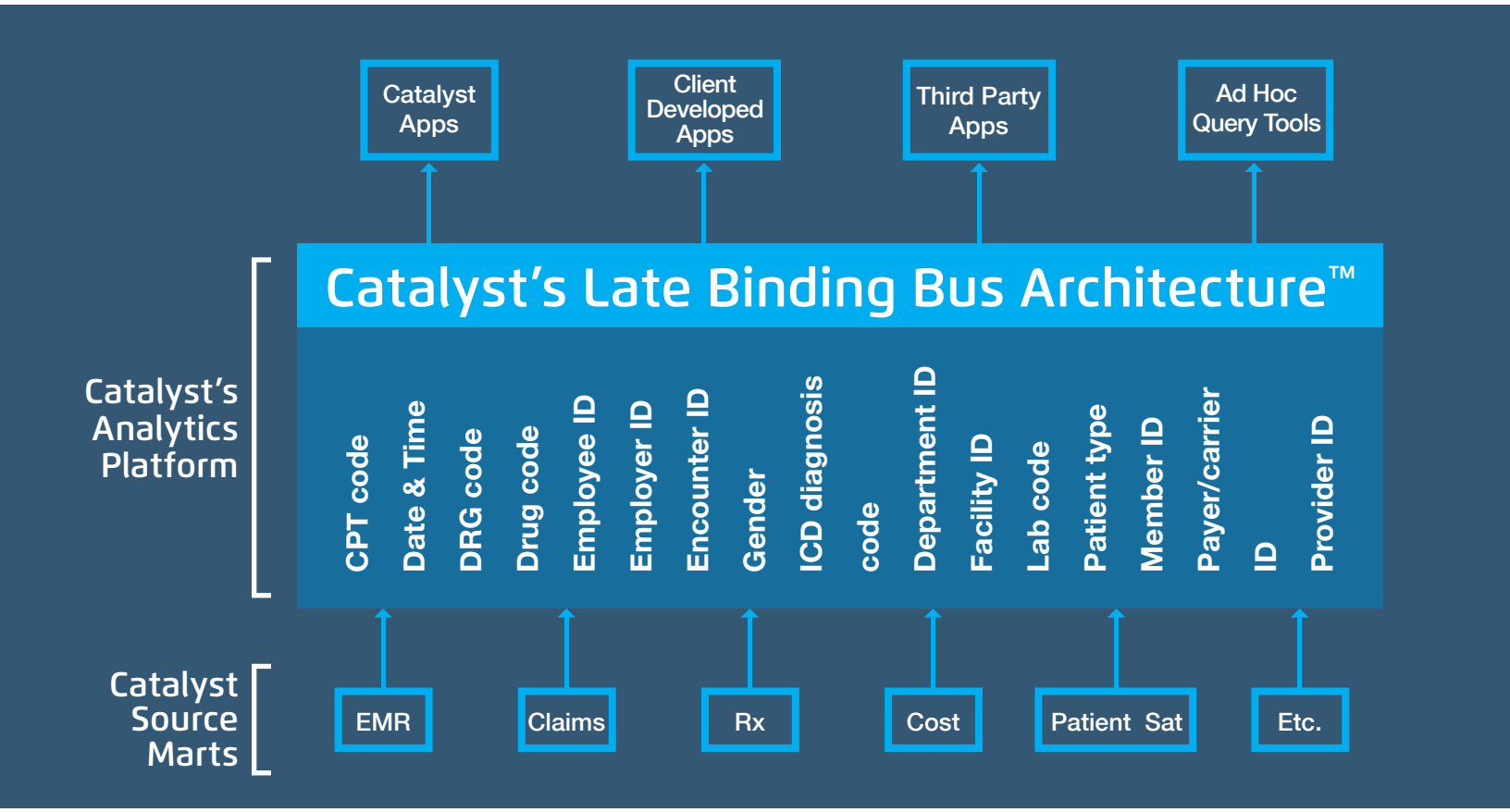
Five basic approaches

- **Corporate Information Model**
  - Advocated by Bill Inmom and Claudia Imhoff
- **I2B2**
  - Advocated by Academic Medicine
- **Star Schema**
  - Advocated by Ralph Kimball
- **Late Binding Bus Architecture**
  - Advocated by Dale Sanders
- **File Structure Association**
  - Popularized by IBM mainframes in 1960s
  - Reappearing in Hadoop & NoSQL
  - No traditional relational data model

The Late-Binding™ Data Warehouse is balanced between the extremes of early binding in Inmon, Kimball, and I2B2 with the no-binding environment of Hadoop. The Late-Binding™ Data Warehouse emphasizes the following fundamental principles related to data modeling:

1. The key to success for data warehouses is relating data, not modeling data. When in doubt, model less, not more.
2. Minimize the use of new conformed data models in the data warehouse by instead leveraging the data models used in the source systems.
3. Apply data models to subsets of data—in data marts—when binding formerly disparate data in new contexts to support new analytic use cases.

4. Approximately 20 core data elements are fundamental to almost all analytic use cases in the healthcare industry. Early binding to these core data elements is a best practice. Bind to other terms and vocabularies later and only when required by analytic use cases.



The core data elements are shown above, illustrating how those data elements leverage the data models of the source systems to act as a data bus for Catalyst's Late-Binding™ Data Warehouse platform. This approach allows queries across disparate source system content in the data warehouse in exactly the same fashion as the theoretical benefits of an enterprise data model but does not require development of, and conformance to, an enterprise data model. The diagram also illustrates how the Catalyst analytics platform can easily feed data to non-Catalyst applications.

Late Binding to Other Vocabulary Terms and Rules

As organizations progress in analytic maturity and sophistication, the need to bind to new and more complex vocabularies and rules will follow. By focusing first on the core data elements and then binding to additional rules and vocabulary when a clear analytic use case requires it, data engineers can deliver rapid time-to-value initially as well as later, when adaptability to new analytic use cases arises. Some of the additional vocabularies that are typically appropriate for later binding in healthcare include LOINC, RxNorm, SNOMED, and HCPCS. Business and clinical rules about data are even more complex and volatile. Catalyst's Analytic Adoption Model, below, illustrates the relationship between progressively higher levels of analytic capability and the need to bind to more complex rules and vocabularies. The important concept to reemphasize is not to bind data to rules or vocabularies until the analytic use case requires it. Too often, data warehouse projects in healthcare attempt to bind data to rules and vocabularies in anticipation of functioning at Level 8 of this model when the organization is still operating at Level 0. It takes years to progress to Level 8, and during that time, rules and vocabularies in healthcare will undoubtedly change. As the

old saying goes, don't drive beyond your headlights. It is a dangerous waste of resources and time to bind to rules and vocabularies that are far beyond the current analytic use cases of the organization.

Summary of Principles in the Catalyst Late-Binding™ Data Warehouse

Below is a summary of the principles that underlie the Catalyst approach to analytics. These principles enabled data warehouses in the military, manufacturing, and healthcare that have been operating and adapting for over 20 years with an unparalleled track record for proven results.

1. Minimize remodeling data in the data warehouse until the analytic use case requires it. Leverage the natural data models of the source systems by reflecting much of the same data modeling in the data warehouse.
2. Delay binding to rules and vocabulary as long as possible until a clear analytic use case requires it.
3. Earlier binding is appropriate for business rules or vocabularies that change infrequently or that the organization wants to lock down for consistent analytics.
4. Late binding in the visualization layer is appropriate for what-if scenario analysis.
5. Retain a record of the changes to vocabulary and rule bindings in the data models of the data warehouse. This will provide a self-contained configuration control history that can be invaluable for conducting retrospective analysis that feeds forecasting and predictive analytics.

About Health Catalyst

Health Catalyst provides data warehousing solutions that actually work in today's rapidly changing healthcare environment. Health Catalyst is on a mission to transform healthcare in the U.S. by utilizing its next- generation data warehousing solutions to accelerate care improvement for all types of healthcare systems. Helping hospitals and health systems to create a data-driven approach to care, Health Catalyst provides clinical, IT and financial executives with the tools and technologies necessary to improve care by reducing costs. Clients include Allina Hospitals and Clinics, MultiCare Health Systems, North Memorial Health Care, Stanford Hospital and Clinics, Texas Children's Hospital, and Providence Health & Services.

Health Catalyst, LLC
3165 East Millrock Drive, Suite 450
Salt Lake City, Utah 84121
ph. (801) 708-6800
healthcatalyst.com

HealthCatalyst
ignite change