



**The
Data
Administration
Newsletter**

Since 1997

Published by Robert S. Seiner

[Home](#)
[Articles](#)
[Columns](#)
[Features](#)
[Blogs](#)
[News](#)
[Events](#)
[Resources ▾](#)
[Archives](#)
[Strategy](#)
[Governance](#)
[Architecture](#)
[Data Mgmt](#)
[Quality](#)
[Modeling](#)
[Operations](#)
[Metadata](#)
[MDM](#)
[BI](#)
[Applications](#)
[Big Data](#)
[Subscribe to The Data Administration Newsletter Today - it's FREE!](#)
[REGISTER HERE!](#)

[Home](#) / [Data Topics](#) / [Data Architecture News, Articles, & Education](#) / [Data Architecture Articles](#) / [The Unfolding of the Data-Centric Paradigm – Part 1](#)

The Unfolding of the Data-Centric Paradigm – Part 1

Posted on September 18, 2019 by Derek Coursen with 1 Comment



There is a movement to upend traditional thinking about information systems by putting data and meaning at the center of strategy, architecture, and system development sequencing. It is the most recent of a series of data-centric waves which, over several decades, have attempted to remedy the entrenched application-centric paradigm. Past waves have receded, largely because data modeling theory has lacked solid foundations.

The current movement brings a new strength by focusing on meaning and offering the possibility of ontologies that can disentangle semantics from structure and technology. Yet the problems inherited from data modeling will still need to be resolved.

To establish a complete data-centric paradigm, there must be foundational principles. Those must highlight a clear break with application-centric thinking; must define the relationship

between the semantic core of an information system, the real world, functional requirements and business language; and must point toward criteria for assessing the quality of the semantics.

Paradigm Shift in Information Systems

There is a movement to upend traditional thinking about information systems by putting data and meaning at the center of strategy, architecture, and system development sequencing. [The Data-Centric Manifesto](#) affirms that "Data is the center of the universe; applications are ephemeral." [7] The Manifesto identifies the current information system paradigm as *application-centric* and outlines its consequences: information footprints that become more and more fractured into silos; the failure of a high proportion of application development projects; and exorbitant costs for changing or integrating information systems. And the Manifesto advocates a data-centric architecture with shared data stores interpreted by an ontology, or high-level data model. In the same spirit, [The Data Doctrine](#) offers a counterpoint to the Agile Manifesto:

- Data Programs Preceding Software Projects;
- Stable Data Structures Preceding Stable Code;
- Shared Data Preceding Completed Software;
- Reusable Data Preceding Reusable Code. [1]

Paradigm is a big word and often misused. But the current data-centric movement is advocating change so broad that it requires a paradigm shift in the formal sense. It needs to be explored in those terms.

"One thing about which fish know exactly nothing is water," wrote Marshall McLuhan, "since they have no anti-environment which would enable them to perceive the element they live in." [13] In the social sciences, *paradigm* means, roughly, a framework of broad assumptions that organizes people's thought and action. People often take the paradigm in which they operate for granted; they may not even notice that it exists, and may be unable to imagine that any alternative is possible. People working within one paradigm may be unable to understand the assumptions of an opposing one; a person defecting from one to another may experience something like a religious conversion. [3]

This article begins from the premise that the information systems profession has always operated within an application-centric paradigm which, for long stretches of time, can go largely unnoticed and unquestioned. There is, however, an opposing data-centric impulse which periodically emerges when application-centric approaches are seen to be failing; some new data-centric practice then becomes popular.

Search TDAN.com

Search ...	Search
------------	--------

Follow TDAN.com:



Advertisements

So far, data-centric waves have always receded without washing away the application-centric paradigm. The letdown is then attributed to a knot of methodological, technological, and other factors.

But there is a more comprehensive explanation. The application-centric paradigm has survived so far because data-centric ways of thinking and practicing have not yet coalesced into a complete paradigm that can replace it. The most basic assumptions of the application-centric paradigm have not yet been noticed and challenged. Fundamental questions which thwarted past data-centric approaches have not yet been dealt with. And alternative data-centric principles—the foundations for a new paradigm—have not yet been articulated.

This article traces how the possibility of a data-centric paradigm has slowly unfolded over time. There is a fundamental difference between application-centric and data-centric ways of perceiving the information system. The data-centric mindset offers a holistic understanding of the common root of the problems that afflict information systems; the application-centric mindset cannot. In the past, though, data-centric approaches for resolving those problems came to grief because data modeling theory has lacked solid foundations. A crisis of confidence in data modeling has, for two decades, allowed application-centric approaches to dominate the profession. Now, a new and ontological wave of data-centric thought is emerging. It focuses more explicitly on semantics, and offers the promise of disentangling meaning from the constraints of structure and technology. Yet as the heir of earlier approaches, it will inherit unresolved issues from the history of data modeling. The possibility of advancing a complete data-centric paradigm hinges on establishing principles that both resolve those problems and that help practitioners reframe their most basic understanding of information systems.

The Data-Centric Way of Seeing

The essence of the application-centric paradigm is to view the information system as a set of functionalities which happen to need the support of data structures. This is a habit of perception. Current professional training, literature, and methods pervasively reinforce it.

Yet some information system practitioners, usually without openly or deeply challenging that paradigm, have a data-centric mindset. It is an alternative way of seeing. When they look at the information system, they primarily see the substructure that organizes data, upon which functionalities may be built. They are able to perceive that the data structures constitute a meaningful artifact in their own right.

From that vantage point, data-centric practitioners can then focus on the information system's potential affordances and limitations in a way that extends beyond the horizon of current intended functionality. While the application-centric perspective is tightly focused on what the system must do for its stakeholders in the present, the data-centric perspective includes a broader dimension of concern— how will different options for the design of data structures impact the system's future ability to meet needs that are still unknown?

As a result, when data-centric practitioners look at the wide range of apparently different problems afflicting information systems, they see one overarching pattern: *After data structures have been established, unanticipated changes in their environment, arcane shortcomings in their design, and emerging expectations in relation to other data sources all create pressures which are costly to alleviate.*

The problems arise first during system design. The cost of changing a specification increases exponentially the later the need is discovered. [2] The data-centric perspective notices that changes at the level of data structures are expensive to propagate upward through the design; it attributes a large proportion of project failure to that fact. Data-centric practitioners therefore aim to figure out good data structures as early as possible. By contrast, the application-centric perspective tends to focus on the complexities of gathering and validating users' requirements.

The same problem continues after the information system has been rolled out. The requirements change or become more completely understood, or limitations in the data structures rear their heads. Modifying the system from the bottom up is expensive. Foreseeing this, data-centric practitioners wish to aim, from the design stage, to reduce long-term costs by building stability and evolvability into the data structures. [6] But that requires looking beyond the immediate project, an approach which clashes with the application-centric convention of focusing on users' requirements within project boundaries.

Finally, the existence of other information systems impacts stakeholders' expectations. They want to reduce inconsistency and redundant data entry, and they want more panoramic and deeply analytical views of the enterprise. Within the application-centric paradigm, various fixes have been attempted: developing point-to-point communication between multiple smaller systems; replacing them with larger enterprise systems; and bringing data from multiple sources together in data warehouses or lakes. [11] Data-centric practitioners notice that disparate data structures are the primary factor limiting integration of business processes and data, whether operationally or for analytics. They therefore advocate for addressing those upstream differences.

Though data-centric practitioners discern the underlying unity in all these problems, their efforts to advance a holistic solution have so far been frustrated. The application-centric way of doing things directs attention away from data structures as a common factor. And it is difficult to demonstrate the impact of decisions in that area to anyone who doesn't have a background in data modeling. But in addition, the data-centric impulse today is being held back by its own history.

Crisis of Confidence

The strongest waves of data-centric practice so far have been information engineering in the last decades of the twentieth century and enterprise data modeling thereafter. Both are widely considered to have failed to live up to their promises. There are various accounts of why.

The most far-reaching explanation is an inconvenient one: *Data-centric practices have treated data models as the foundations of information systems, yet the data modeling community has little agreement on the theoretical foundations of its own craft.*

Upcoming Events

- Data Architecture Summit 2019 - Tutorial - 10/15/2019
- RWDG Webinar: Build Your Own Data Governance Tools - 10/17/2019
- RWDG Webinar: How can Everybody be a Data Steward? - 11/21/2019
- RWDG Webinar: Utilize Governance Working Teams to Improve Data Quality - 12/19/2019

Quote of the Week

Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time. --
Thomas A. Edison

Popular Posts



Universal Data Vault: Case Study in Combining "Universal" Data Model Patterns with Data Vault Architecture – Part 1
November 1, 2015



Data Modeling is Data Governance
August 1, 2015



Data Warehouse Design – Inmon versus Kimball
September 1, 2016

Tags

agile big data data data governance
data management data modeling data quality
Metadata non-invasive data governance seiner

It turns out that thought leaders hold a wide and incompatible range of views on what a data model is, how it relates to the real world, how to characterize the data modeler's work, and how that work should relate to business requirements. Graeme Simsion frames these convoluted debates in terms of two poles: whether data modeling is considered an activity of description or design. Description is related to science, analysis, and knowledge of the world; it often implicitly or explicitly suggests that there is a single correct solution to represent a single objective reality. By contrast, the idea of design incorporates the role of subjective interpretation and posits an inexhaustible number of different solutions. [16]

Furthermore, data modeling is recognized as a difficult skill to learn, and there is little agreement on what constitutes high quality. Daniel Moody identifies thirty-nine different proposals for assessing the quality of conceptual data models, of which less than a fifth have been tested empirically; the proliferation, he notes, is counterproductive to establishing a cumulative tradition and a common paradigm. [14]

This miasma of uncertainty has contributed to a crisis of confidence in data modeling. It has weakened the attempt to assert the importance of data models. It has allowed application-centric approaches to dominate. Many people now recognize that those are making matters worse and are therefore open to considering new data-centric possibilities. But the history, and the lack of a coherent theoretical foundation, stand in the way.

Analogically, one can imagine a world in which a circle of experienced building contractors had realized, from observation and intuition, that a building needs a solid foundation. They were skilled at laying good foundations, but they hadn't yet agreed on how to explain or measure the goodness, and it was hard to teach the craft to others. Nonetheless, they spearheaded a movement promising better, safer, longer-lasting buildings, if only people would pay for laying solid foundations. Over the objections of other builders (who preferred to think about what the building would do for its inhabitants), the foundations movement became popular. But inadequate theory, oversimplified hype, and poor education led to faulty planning and execution of many foundations.

In some, stones were not properly placed; in others, people skimped on the cement; and in others they tried to make mortar without sand. Many buildings collapsed, but people could not agree on the cause. Then there was a construction boom. Owners tried to add new wings and levels that hadn't been anticipated, and to connect their buildings with tunnels and walkways. More and more of the foundations fell apart, flooded or sank into the ground. After years of this, it became unfashionable to talk about laying solid foundations. And yet the original insight about their importance had been correct.

In the last decade, the sharpest edge of data-centric thinking has shifted to semantics (meaning) and ontologies. But the underlying issues that have blocked a common understanding of data models will create problems there as well. They must still be faced.

Disentangling Meaning from Structure and Technology

Information systems contain meaning. By focusing on functionality, the application-centric paradigm loses sight of that. Data-centric practitioners, though, have always paid close attention to meaning. They look at data rather than functionality because it is a more direct route toward understanding meaning.

Meaning appears under many guises. For example: A child can observe that *Dad's Mazda has license plate 1234; Mom's Fiat has license plate 5687*. A data modeler can note the general pattern that *every vehicle has an owner, brand, and license plate* and can create entity and attribute types on an ERD. A clerk at the DMV can enter those data elements into an interface, and a manager can run a report displaying those columns. The agency can maintain a comprehensive catalog of brands including the values *Mazda* and *Fiat*. To transport data between information systems, someone can create an XML format with the tags *vehicle owner*, *vehicle brand* and *license plate* and embed it in a SOAP envelope. Someone else can develop a JSON schema using the broader terms *asset owner*, *manufacturer* and *identification number*.

People perceive the differences between these situations, and they tend to talk about the stakeholders' purposes and the technologies used; but that is an application-centric viewpoint. A data-centric perspective notices the common core meaning, and it asks how best to represent the concrete phenomena using abstract concepts. The practical challenge is in choosing concepts, and relating them to each other, accurately and at favorable levels of abstraction.

A few decades ago, data modeling was celebrated as a powerful way to clarify and represent meaning. But then—at the same time that the weak theoretical foundations of data modeling were being recognized—people began to see that in data models, the representation of meaning is bound up with and limited by considerations of structure based on technology. That, in turn, pointed to a new challenge: Could meaning be represented in as pure a form as possible, separate from the structural considerations of organizing data?

Ontologies offer a way to do that. Yet although there is a break between the craft of data modeling and that of creating ontologies, there is an essential continuity as well. It is apparent in the way in which data models shade into taxonomies.

Where Data Modeling and Taxonomy Converge

Information systems have always had lookup tables, which are in effect taxonomies. Yet, the formal discipline of taxonomy, which developed in traditional logic and library science, has gotten short shrift. As Malcolm Chisholm has lamented, the result is that "taxonomies often seem to be little more than grab bags of concepts thrown together, neither dividing up a general concept, nor aggregating distinct concepts for a specific purpose." [4] Poor taxonomies then make nonsense of the attribute definitions that they purport to exemplify. This is a strange historical blind spot in data modeling. Its literature shows great awareness of how changing the meaning of an attribute can change the meaning of an entity; yet there's much less acknowledgement of the interdependence between the meaning of an attribute and the coherence of its domain values.

Within data modeling, though, a semantically aware strand of practice does embrace taxonomic techniques. Graeme Shanks has shown that expert modelers often seek high quality through *innovation*, the introduction of concepts which were not mentioned by the user. [15] Innovation often aims at attaining flexibility by choosing higher degrees of abstraction than are present in everyday business language. The higher abstraction then pushes the semantics out of the model's static structures (entity and attribute types) into data itself. For example, a data model might include the entity type *asset* instead

of *vehicle*, with *vehicle* then becoming a value in a lookup table *asset type*. At that point, the meaning must be managed by managing data records, and that calls for using methods appropriate for taxonomies.

This has been a recognized staple of advanced data modeling practice since David C. Hay published *Data Model Patterns*. [5] Dave McComb calls this a metadata design approach, and notes that many good designers are not even aware that they are doing it. [8] What happens is that the designer, searching for a good representation of meaning, shifts his or her awareness fluidly between the two places where meaning might be stored: static structure and dynamic data. That mental movement incorporates taxonomy into the act of data modeling.

Ontologies explicitly integrate the discipline of taxonomy with the semantic craft used in advanced data modeling practice. Their essential continuity with data models is that they embody a choice of concepts, relationships, and levels of abstraction in order to represent concrete phenomena. Their advantages over traditional data models are—in addition to integrating taxonomy—that they are pure conceptual artifacts which can be maintained separately from the structures in which data is stored, and that they offer more opportunity to clarify and consolidate vague or idiosyncratic terminology across the enterprise. Dave McComb estimates that the data of most enterprises can be represented by an elegant core semantic model of a few hundred concepts and a few thousand taxonomic modifiers. [10]

Ontologies point toward a future in which information systems can be designed to represent meaning in a way that is both stable and can flexibly accommodate unanticipated changes in the environment and emerging expectations of the data. Dave McComb offers a vision in which “the data model precedes the implementation of any given application and will be around and valid long after it is gone” [9] and outlines the elements of a complete information system architecture that can make this possible. [12]

The advantages of such an approach are clear. However, the creation of these core semantic models will inherit some of the same questions that have dogged data modeling. In order to advance a data-centric paradigm, it is necessary to proactively surface and resolve them.

The Need For Foundational Principles

As the Data-Centric Manifesto puts it, “...the main barrier to changing [the application-centric] paradigm is not technical, but mental and inertial.” [7] Some of the mental barriers come from the dominance of that paradigm, while others come from the history of data-centric approaches.

Most information systems practitioners—including many of those who primarily work with data and even with data models—have been trained and immersed in the application-centric paradigm for their entire careers. Focusing on functionality first has become second nature. This socially reinforced way of seeing makes it difficult for practitioners to learn to work in data-centric ways. At the same time, there’s a lingering hangover from earlier data-centric practices. Advocating for new ones can provoke the automatic response that previous data-centric efforts stumbled decades ago.

The solution is to declare and develop a new paradigm. The application-centric paradigm is bankrupt and must be replaced. But replacing one paradigm with another requires articulating principles which sharply and publicly distinguish the new from the old. Those principles must encapsulate and justify the data-centric way of seeing and designing information systems. They must highlight the limitations of application-centric approaches. They must represent a common denominator of understanding that is sufficient to unite people who currently profess divergent data-centric positions while attracting others who so far do not. They must provide methodological support for the practical work of designing core semantic models that can organize the data of entire enterprises.

To do all that, the principles must establish common ground on the unresolved issues inherited from the history of data modeling. Those can be boiled down and rephrased as questions about the semantic core of an information system, meaning the logical arrangement of concepts into entity types, relationships between them, attribute types, and taxonomies that constrain attributes’ values. (The term *semantic core* is used to refer to the formal ontologies that could organize information systems of the future or, equally well, to the data models and associated taxonomies of existing information systems built on current platforms.) The questions are:

- *How does the semantic core of an information system relate to the real world, to functional requirements, and to the language by which the business describes itself?*
- *And what characteristics comprise high quality in the semantic core of an information system?*

The sequel to this article will present three principles which can help resolve these questions.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

REFERENCES

- [1] Aiken, P., & Harbour, T. (n.d.). The Data Doctrine. Retrieved from <https://www.thedatadoctrine.com/>
- [2] Boehm, B. (1981). *Software engineering economics*. Prentice-Hall.
- [3] Burrell, G., & Morgan, G. (1979). *Sociological paradigms and organisational analysis: elements of the sociology of corporate life*. Heinemann Educational Books.
- [4] Chisholm, M. (2012). The celestial emporium of benevolent knowledge: taxonomies are everywhere in information management, but they are hardly ever formally acknowledged and managed. *Information Management*. (February 17)
- [5] Hay, D. C. (1996). *Data model patterns: conventions of thought*. Dorset House Publishing.
- [6] Marche, S. (1993). Measuring the stability of data models. *European Journal of Information Systems*, 2(1), 37-47.

[7] McComb, D. (n.d.). The data-centric manifesto. Retrieved from <http://datacentricmanifesto.org/>

[8] McComb, D. (2004). *Semantics in business systems: The savvy manager's guide*. Morgan Kaufmann.

[9] McComb, D. (2016). The data-centric revolution: data-centric vs. data-driven. *The Data Administration Newsletter*. (September 21)

[10] McComb, D. (2017). The data-centric revolution: the core model at the heart of your architecture. *The Data Administration Newsletter*. (September 6)

[11] McComb, D. (2018). *Software wasteland: how the application-centric mindset is hobbling our enterprises*. Technics Publications.

[12] McComb, D. (2019). The data-centric revolution: the 1st annual data centric architecture forum. *The Data Administration Newsletter*. (March 6)

[13] McLuhan, M., Fiore, Q., & Agel, J. (1968). *War and peace in the global village*. Bantam Books.

[14] Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3), 243-276.

[15] Shanks, G. (1997). Conceptual data modelling: an empirical study of expert and novice data modelers. *Australasian Journal of Information Systems*, 4(2).

[16] Simsion, G. (2007). *Data modeling theory and practice*. Technics Publications.

Tagged architecturecoursendata centricdata modelingParadigmstrategytaxonomy

Share

Like 8

Tweet

Share

+ reddit this!



About Derek Coursen

Derek Coursen develops information systems strategy and data architecture for public service organizations. He has led informatics departments at two major nonprofit agencies in NYC and has been adjunct faculty at NYU's Wagner School of Public Service. Derek holds master's degrees in information science (Pratt Institute), information systems (Pace University) and management (NYU). He can be contacted via [Derek Coursen Consulting LLC](#) or [LinkedIn](#).

View all posts by Derek Coursen →

Related Posts

Announcing The Wilshire Conferences' Award for Metadata Best Practices 2006 WILSHIRE AWARD WINNER Bank of Montreal



Data Quality and the Project Life Cycle

Pick Your Partners:

It's All in the Data: It was 20 Years Ago Today
