EXECUTIVE UPDATE
Business & Enterprise Architecture, Vol. 24, No. 4

Cutter Consortium
Access to the Experts

The Role of Business Architecture in Defining Data Architecture

by William Ulrich, Fellow, Cutter Consortium

Inconsistent, incoherent, or nonexistent business vocabularies undercut strategic planning, impact analysis, business design efforts, requirements definition, and software and data design and deployment. These issues are magnified when planning and executing strategies across business units and deployment teams. This disjointed business vocabulary results in a wide range of challenges. In this *Executive Update*, we explore how business architecture can help define data architecture, delivering transparency across a number of related business domains.

The Art of Clarity

In many organizations, discussing information is like discussing the weather; people talk about it, but few do anything about it. Inattention and inaction on information clarity, consistency, and integrity have significant impacts on an organization's ability to maintain a positive customer experience, compete profitably, and ultimately thrive.

Information clarity is essential to planning, as noted by Prussian General Carl von Clausewitz in his book On War:

> *The first task of any theory is to clarify terms and concepts that are confused.... Only after agreement has been reached regarding terms and concepts can we hope to consider the issues easily and clearly, and expect others to share the same viewpoint....*

General von Clausewitz's quote provides important guidance to business leaders. An organization's ability to execute strategy is undermined when "terms and concepts are confused." Inconsistent, incoherent, or nonexistent business vocabularies undercut strategic planning, impact analysis, business design efforts, requirements definition, and software and data design and deployment. These issues are magnified when planning and executing strategies across business units and deployment teams.

For example, when attending meetings with individuals from other business units, consider how much time attendees spend when trying to gain clarity on certain terms and concepts. Of course, no one wants to make a bad impression, so instead of asking, "What do you mean when you say...?" people simply talk past one another. As a result, meeting attendees walk away with a complete misinterpretation of what is needed to execute a shared, coordinated strategy. This scenario repeats itself multiple times a day at organizations, creating confusion and derailing the most well-conceived strategies.

Business Challenges Stemming from Information Discontinuity

A disjointed business vocabulary results in a wide range of challenges, including:

- Strategy execution failures lead to multiple teams taking conflicting and poorly aligned actions that fall short of expectations.
- Customer discontinuity results in not recognizing the same customer across business units, fueling customer dissatisfaction and losses.
- Initiative scoping issues trigger investments that underdeliver, run overbudget, and deliver solutions that are either rejected or create more problems than they solve.
- Regulatory violations, where regulators cite the inability to produce consistent regulatory reporting results.
- Financial reporting errors proliferate because different business units name the same data using different terminology.
- Poorly defined, misaligned business requirements that use interchangeable terminology confound development teams and delay or scuttle solution deployments.
- Inoperability issues across software deployments force business professionals into manual workarounds and extensive use of desktop solutions.
- Lack of a foundation for AI-based hyperautomation stifles the ability to leverage and scale the use of cognitive computing technologies.

> *When organizations lack clearly defined, business-vetted information concepts, the data they rely on results in many of the aforementioned business challenges.*

Many of the above issues stem from organizations lacking a well-defined, rationalized business vocabulary as a basis for information management. When organizations lack clearly defined, business-vetted information concepts, the data they rely on results in many of the aforementioned business challenges. It is important to differentiate between information and data. For purposes of this discussion, "information is considered a ubiquitous concept that includes human knowledge, sense of mission, and learned behaviors, in addition to more traditional perspectives on information." Data, on the other hand, is defined as "value specifications for qualitative and quantitative variables."

Data represents a subset of information. Consider, for example, that every decision, task, request, inquiry, or message exchanged between parties incorporate aspects of information. Rarely are the totality of these information concepts expressed as data. Yet where data does exist, it has a direct lineage to information. For example, the existence of an information concept called "customer" creates an expectation of having data about that customer. In reality, however, a well-bounded, rationalized perspective on customer and many other information concepts is often misrepresented or missing from operational data.

Various business dialects have rippled into plans, designs, models, operational data, and software systems and, in turn, have magnified software inoperability challenges.

Data incongruity evolved organically as individual business units adopted a variety of language dialects. Dialects form when there is no contact between regions, resulting in words and language evolving independently. Business dialects evolved because many business units and teams within those business units are siloed off from their peers. Various business dialects have rippled into plans, designs, models, operational data, and software systems and, in turn, have magnified software inoperability challenges. The unfortunate aspect of this situation is that even to this day, organizations have made little headway in reversing course on data incongruity and software interoperability challenges.

Make no mistake; there have been major advancements in data science. Consider the work associated with "big data," which analyzes and extracts information from data sets too large or complex to be dealt with by traditional software applications. While organizations should embrace advancements in big data, artificial intelligence, and other areas, it does not eliminate the need to recognize that a customer of one business unit is the customer of a second business unit and that each instance of that customer has a unique agreement and financial account. Incredibly, this and countless other data-challenged scenarios are much more prevalent than business leaders may realize.

When Is an Account an Account?

Data incongruity stems from two factors: synonyms and homonyms. A synonym is a term that means the same as another term. For example, “agreement” and “contract” are synonyms, but only one is acceptable in a formal vocabulary. Homonyms create a more unique challenge, as they represent the same term that can be interpreted to mean multiple things. Incorporating synonyms and homonyms into strategic plans, business requirements, operational data, and software solutions leads to software interoperability and scalability issues. Software interoperability constraints drive up the time, costs, and risks of using and changing software, forcing systems to grow more complex as more software is deployed to feign interoperability.

> *Software interoperability constraints drive up the time, costs, and risks of using and changing software, forcing systems to grow more complex as more software is deployed to feign interoperability.*

For example, if System A thinks an account is a customer and System B thinks an account is an agreement, there is no way for these solutions to interoperate without extensive manual and desktop manipulations. Situations such as this are the norm in the vast installed base of legacy software systems and continue to multiply given overall system complexity and diminishing expertise of large-scale, legacy software systems. Software interoperability creates major business challenges but tends to be ignored. Did anyone ever ask, for example, why it takes six people four weeks to determine whether a customer paid an invoice? One wonders if business leaders have gotten to the point where they assume that these situations are normal?

Interoperability issues force business professionals to engage in costly manual workarounds and to deploy more desktop solutions just to keep up with customer and other demands. Consider a midsized financial institution that had created more than 25,000 spreadsheets to ensure that its systems functioned at a minimal level of performance. One business leader noted that a recent software upgrade forced her team to create 15 new spreadsheets just to deploy the upgrade. Proliferation of desktop “shadow systems” hinders organizational effectiveness and creates countless points of failure, risk, and auditability issues.

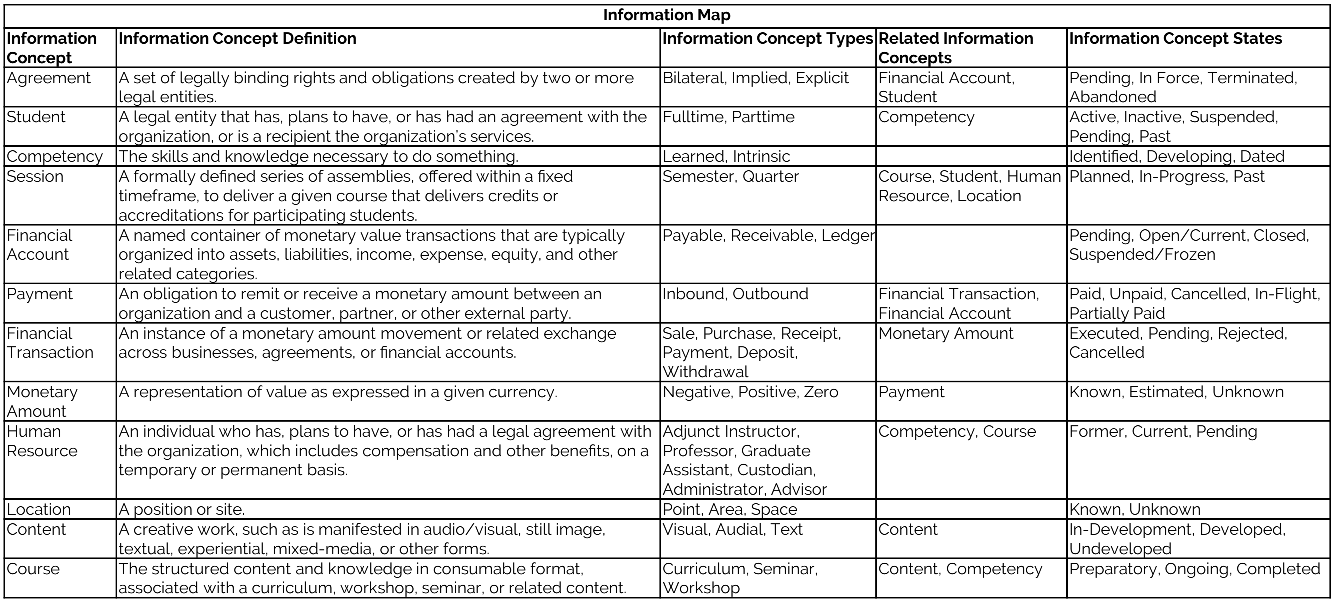
Consider an example of how these situations can take root. A business analyst insisted that everyone knew what people meant when they used the term "account." When business professionals within and across departments were asked to define the term, the variations were striking. Depending on who was interviewed, an account was a customer, an agreement with a customer, an identifier used to identify a customer, or a financial concept used to track monetary balances. This wide disparity in vocabulary usage led multiple deployment teams to pursue four unique data definition and software solution paths. The resulting interoperability issues resulted in the cancelling of final deployments, which would have created more business problems than they resolved. This scenario repeats itself daily at organizations around the globe, all because business professionals initiate and fund high-priced technology investments that span multiple business units, while spending little or no time crafting a common view of their business vocabulary.

Defining Business Architecture Information Maps

Some people might argue that their data architecture team has defined their data. This may be true, but even in the best of cases, existing data definitions rarely reflect the breadth, clarity, and rationalized business perspectives required to fully represent information as it is viewed by the business as a whole. But there is a discipline that if applied properly can help.

Business architecture delivers transparency across a number of related business domains, enabling and expediting strategy execution from planning through deployment. Business architecture in its most widely accepted form encompasses 10 domains, with the four core business architecture domains being capability, value stream, organization, and information. Information in business architecture manifests as a collection of well-defined information concepts captured in an *information map*. The information map represents the complete set of information concepts, definitions, types, states, and cross-concept relationships for a business ecosystem. Business ecosystems extend beyond the legal boundaries of an enterprise to accommodate the full scope of capabilities required and delivered by business units and partners.

Figure 1 depicts a partial information map for a university, which is limited to student, course, location, competency, session, agreement, and select financial information concepts. Information concepts are based on well-delineated business objects that represent real-world "things" within a business ecosystem. Organizations should not conflate business objects, which represent things in the real world, with information concepts, which represent information about those things. Business objects also serve as the basis for forming capabilities, which define the actions applied to those objects. A "session" business object, for example, would have a corresponding Session Management capability that uses and modifies session information.



Information Map

Information Concept	Information Concept Definition	Information Concept Types	Related Information Concepts	Information Concept States
Agreement	A set of legally binding rights and obligations created by two or more legal entities.	Bilateral, Implied, Explicit	Financial Account, Student	Pending, In Force, Terminated, Abandoned
Student	A legal entity that has, plans to have, or has had an agreement with the organization, or is a recipient the organization's services.	Fulltime, Parttime	Competency	Active, Inactive, Suspended, Pending, Past
Competency	The skills and knowledge necessary to do something.	Learned, Intrinsic		Identified, Developing, Dated
Session	A formally defined series of assemblies, offered within a fixed timeframe, to deliver a given course that delivers credits or accreditations for participating students.	Semester, Quarter	Course, Student, Human Resource, Location	Planned, In-Progress, Past
Financial Account	A named container of monetary value transactions that are typically organized into assets, liabilities, income, expense, equity, and other related categories.	Payable, Receivable, Ledger		Pending, Open/Current, Closed, Suspended/Frozen
Payment	An obligation to remit or receive a monetary amount between an organization and a customer, partner, or other external party.	Inbound, Outbound	Financial Transaction, Financial Account	Paid, Unpaid, Cancelled, In-Flight, Partially Paid
Financial Transaction	An instance of a monetary amount movement or related exchange across businesses, agreements, or financial accounts.	Sale, Purchase, Receipt, Payment, Deposit, Withdrawal	Monetary Amount	Executed, Pending, Rejected, Cancelled
Monetary Amount	A representation of value as expressed in a given currency.	Negative, Positive, Zero	Payment	Known, Estimated, Unknown
Human Resource	An individual who has, plans to have, or has had a legal agreement with the organization, which includes compensation and other benefits, on a temporary or permanent basis.	Adjunct Instructor, Professor, Graduate Assistant, Custodian, Administrator, Advisor	Competency, Course	Former, Current, Pending
Location	A position or site.	Point, Area, Space		Known, Unknown
Content	A creative work, such as is manifested in audio/visual, still image, textual, experiential, mixed-media, or other forms.	Visual, Audial, Text	Content	In-Development, Developed, Undeveloped
Course	The structured content and knowledge in consumable format, associated with a curriculum, workshop, seminar, or related content.	Curriculum, Seminar, Workshop	Content, Competency	Preparatory, Ongoing, Completed

Figure 1 — Partial information map for a university.

In addition to the name of each information concept shown in Figure 1, the information map contains a definition for each concept that delineates that concept from other concepts. Information maps also include types, relationships to other information concepts, and a finite set of states. This metadata establishes a business perspective required to understand concept usage in practice. The information map differs from other business-derived data perspectives in several ways, including:

- The information map represents business ecosystems as a whole and must be able to represent and delineate every information concept within that ecosystem.
- Information concepts are clearly and uniquely defined so as not to overlap with any other information concept.
- Each information concept is typed and, as required, subtyped so it is able to apply to as many business scenarios as possible.
- Information concepts are stateless, meaning, for example, that "student" can represent a pending, active, or graduated student, eliminating the need to create applicant, alumni, and enrolled student concepts.

The relationships shown in Figure 1 may also be viewed visually, as shown in Figure 2. Note that unlike a data model, the relationships in Figure 1 do not include cardinalities or number of occurrences in the diagram; these are left to be defined in the data models derived from the information map.

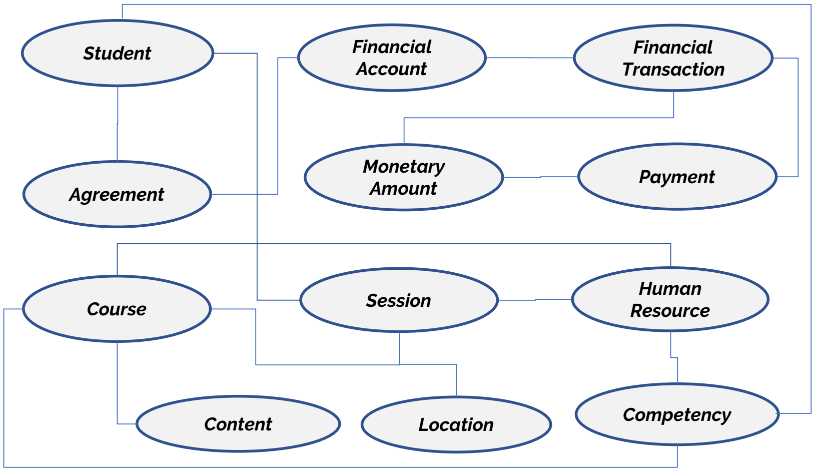


Figure 2 — Information concept relationship diagram.

The information map shown in Figure 2 depicts a subset of information concepts and relationships that accommodate running a course during a session at a given location, assigning an instructor, where human resource represents a professor, adjunct instructor, or graduate assistant, with appropriate competencies required to teach that course. The course may also be associated with content, such as a textbook, that the student must acquire. The information map further accommodates a student with an active agreement with the university, signing up for a course and remitting a specific monetary amount in response to a request for payment from the university. Information maps must accommodate a wide range of business scenarios, but the information concepts themselves remain constant, as they represent an information perspective on the foundational business objects defined across the business ecosystem.

> *Information maps must accommodate a wide range of business scenarios, but the information concepts themselves remain constant, as they represent an information perspective on the foundational business objects defined across the business ecosystem.*

Leveraging Business Architecture to Articulate Data Models

One of the benefits of taking a business-first perspective on information mapping is that information maps are examined and validated across a wide range of scenarios, capabilities, value streams, and related business architecture domains. Because information maps are technology-agnostic and tested against a wide range of business scenarios, they provide a more comprehensive view of the data required to automate those scenarios. For example, does the information map defined in Figures 1 and 2 accommodate enrolling in a course, dropping a course, having the course cancelled, launching a new course, offering student refunds, and switching instructors?

Now compare and contrast the information map to standard data models. Data models can adhere to any number of methodologies, but this discussion uses standard data entity and data attribute modeling concepts. A data entity is something that exists in a business, and a data attribute is a characteristic that identifies the entity, relates an entity to another entity, or describes an entity. Entity relationship modeling is applied through formal data models where:

- Conceptual data models are abstract forms that convey business context but omit details.
- Logical data models represent problem domains independent of a given technology and include tables, data attributes, and associations.
- Physical data models are derived from logical data models and represent data as implemented or to be implemented.

Consider the data models shown in Figure 3. The only data entities shown are student, course, and enrollment (using the British spelling, "enrolments"). "Student" and "course" align to information concepts found in the university information map, but "enrollment" does not have a corresponding information concept.

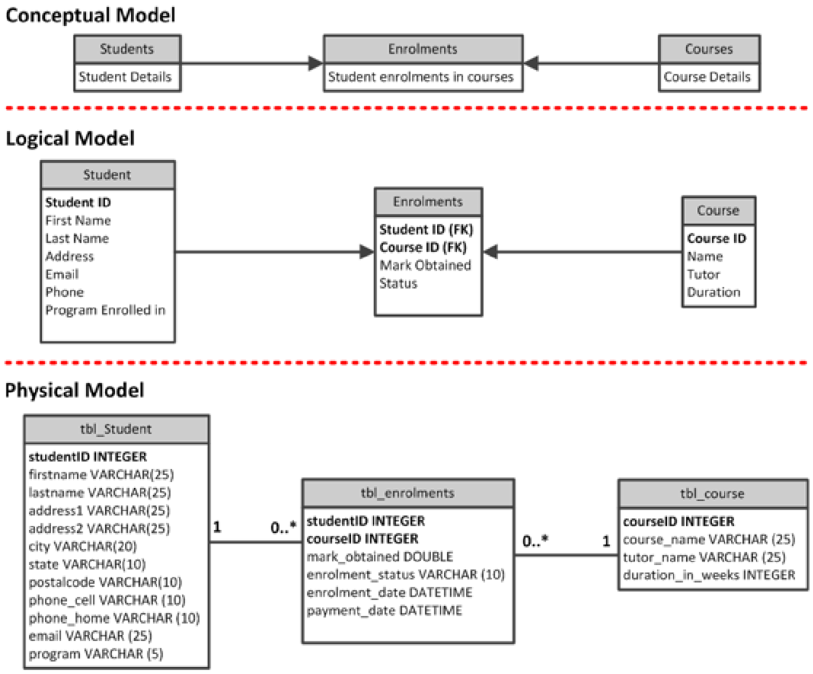


Figure 3 — Sample conceptual, logical, and physical data model for student course registration. (Source: Sri Pakash.)

When viewing the enrollment entity, consider it in contrast to the information concepts defined in the Figure 2 information map. The models shown in Figure 3 omit a good deal of information when compared to the information map. For example, the act of enrolling a student in a course involves the course, the student, student agreement, payment and remitted monetary amount, the session in which the course is slotted, the content associated with the course, student competencies, and assigned instructor. One might argue that perhaps financial concepts are found elsewhere, but could this be due to a siloed perspective on this situation?

This sharp contrast between information maps and data models is not unique in practice. Information maps often cover a broader range of information categories because they are the result of comprehensive scenario walkthroughs developed through holistic discussions with business professionals across various business units. The more ideal approach leverages a top-down, business-driven approach to data entity definition, which involves a three-step approach: (1) data entity derivation, (2) data entity relationship derivation, and (3) data attribute derivation.

Step 1, data entity derivation, is shown in Figure 4. In this example, the agreement information concept is used as a basis to establish a data entity called "agreement." While this may seem simple, it is not the path taken by most organizations, which tend to derive data models from a bottom-up perspective.

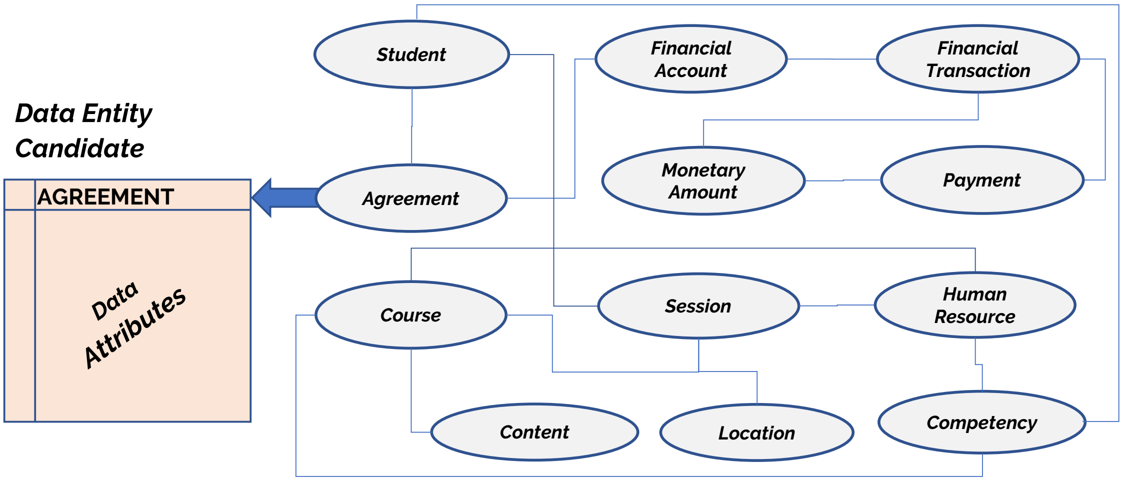


Figure 4 — Data entity derivation from the information map.

Data architects can systematically examine the information concepts in the information map and define corresponding data entities for each of those concepts. There is no assumption that the data model and the information map will be identical. Data architects will apply data modeling techniques to formalize data entities as appropriate. The information map's role is rather to provide business ecosystem transparency, delivering a business-driven perspective to ensure that data models and related deployments enable and do not hinder the organization they are meant to benefit.

As data entities are defined, data architects can leverage information concept relationships to establish corresponding relationships among data entities in the data models. All information maps have a set of relationships that data architects may interrogate to derive their entity relationships.

The next step is to attribute the data entities. Figure 5 depicts data attribute derivation using child capabilities defined under Agreement Management.

The capabilities shown in Figure 5 generally cover agreement identification, term management, preference determination, access constraint determination, risk determination, price determination, and profile, state, type, and history management. These and other capabilities provide data architects with insights into the types of

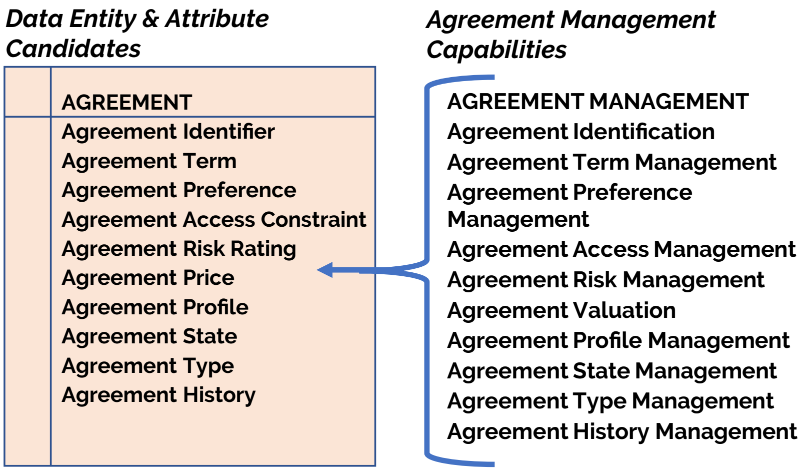

Data Entity & Attribute Candidates
AGREEMENT
Agreement Identifier
Agreement Term
Agreement Preference
Agreement Access Constraint
Agreement Risk Rating
Agreement Price
Agreement Profile
Agreement State
Agreement Type
Agreement History
Agreement Management Capabilities
AGREEMENT MANAGEMENT
Agreement Identification
Agreement Term Management
Agreement Preference Management
Agreement Access Management
Agreement Risk Management
Agreement Valuation
Agreement Profile Management
Agreement State Management
Agreement Type Management
Agreement History Management

Figure 5 — Data attribute derivation from capabilities.

data attributes to be incorporated into the agreement data entity. Figure 5 highlights these attributes, which include agreement identifier, term, preference, access constraint, risk rating, price, profile, state, type, and history. In summary, this three-step data-derivation process involves data entity derivation, data entity relationship derivation, and data attribute derivation.

Modeling the Intersection of Information and Data

Deriving data models from information concepts is an important parallel exercise to software service design because software designers and data architects can leverage information concept/capability relationships as the basis for specifying data requirements for software services. This predefined set of relationships provides a consistent perspective on software service data usage because the information concepts and capabilities are based on a clearly defined, consistent collection of vetted business objects. The metamodel that represents this approach is shown in Figure 6.

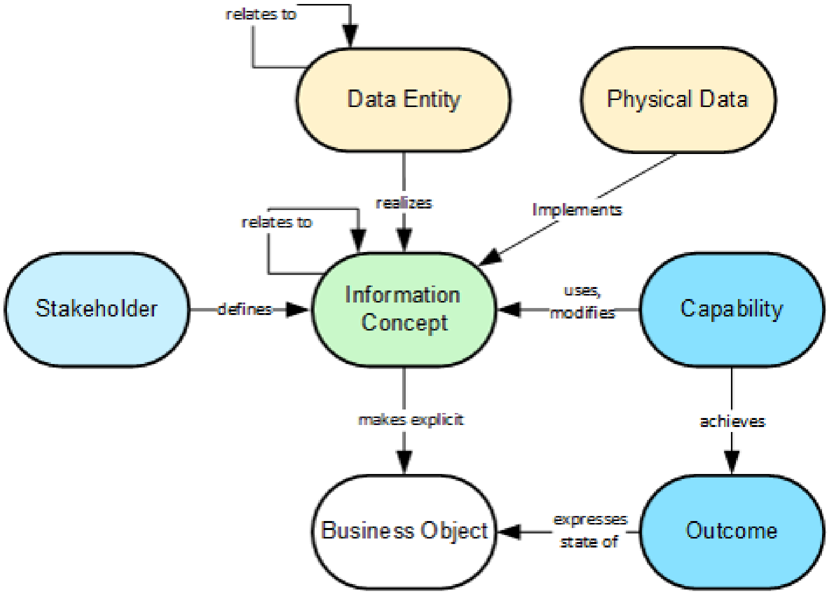


Figure 6 — Information to data metamodel.

The metamodel shown in Figure 6 highlights several associations that underly the overall practice of deriving data models from information maps as well as mapping information concepts directly to physical deployments of the data that corresponds to that information concept. As noted previously, an information concept is derived from or "makes explicit" a business object. The capabilities that define these business objects use and modify the corresponding information concepts that make these objects explicit. Capabilities also establish and manage the relationships among information concepts. The association between capability and information concept is important because it provides context for how information is used to further an organization's capabilities. In this regard, a capability can only modify an information concept it defines and establishes.

> *An information concept is derived from or "makes explicit" a business object.*

For example, Agreement Management capability can only modify agreement information and, correspondingly, Agreement Management cannot modify any other information concept. However, Agreement Management may use agreement as well as customer, policy, product, location, and other information concepts as needed to deliver outcomes.

Figure 6 also highlights the previously discussed association between information concept and data entity and further highlights the relationship between multiple information concepts, an association that is mirrored for the data entity. Not discussed, however, is the association made between information concept to physical data, which supports current-state data architecture assessment efforts. While not the topic of this discussion, organizations seeking to perform a migration or transformation would document these associations as input to targeting certain data structures for migration or other updates.

A Lesson in Top-Down, Business-Driven Data Architecture Specification

The story of two similar organizations highlights the value of using business architecture to derive and define data models and also highlights the pitfalls of not following this approach. The first organization had a business architecture in place, which included a well-defined capability map, value streams, and information map. The second organization did not. Both organizations launched multiyear programs, each of which scaled to include multiple projects running over a period of years and costing eight to nine figures. Both programs adopted and employed user-centered design and Agile deployment techniques.

The first organization leveraged its business architecture prior to starting up development work to create a conceptual data model and a logical data model. The data architecture team used the capability map and information map as input along with consultation with business architects who were also business subject matter experts. These data models formed the foundation for all data design and deployment going forward. As individual deployment teams launched and scaled up, the physical data models designed for those teams were derived from the logical data model created at the onset of the program. As solutions deployed and transformation work ensued, software interoperability was not an issue, enabling more and more deployment teams to come online.

The second organization, while sharing many of the same techniques as the first organization, did not establish business-driven, top-down conceptual and logical data models. As its first deployment team launched and started delivering incremental deliverables, the data architects built physical, bottom-up data models based on the Agile team’s data requirements for a given sprint. As deployment work ensued, multiple physical data models were deployed. At the end of nine months, interoperability issues were becoming too much for the development teams to resolve; multiple data and software deployments would no longer function effectively. The program had to take a pause to address extensive database refactoring and also refactoring of the code as a result of the ripple effects of the data changes. Ultimately, this second organization formed a business architecture and used it to create top-down data models. However, the lack of employing this top-down, business-driven perspective at the onset cost that program an entire calendar year in lost time.

These stories offer a powerful example of why formalizing a business-driven, top-down data architecture is so important to delivering effective data and software solutions to organizations. From a strategy execution perspective, business-driven data architecture work should ideally start up as part of the overall planning effort, ahead of launching any software development work. The first organization's experience highlights the concept of "slowing down to speed things up."

Business leaders should take heed that spending a little bit of time up front to clarify business perspectives can ultimately deliver the solutions they are seeking more quickly, ensuring successful delivery of a wide range of business strategies.

About the Author

William M. Ulrich is a Fellow of Cutter Consortium's Business & Enterprise Architecture practice and President of TSG, Inc. Specializing in business and IT planning and transformation strategies, he has more than 35 years' experience in the business-IT management consulting field. Mr. Ulrich serves as strategic advisor and mentor on business-IT transformation initiatives and also serves as a workshop leader to businesses on a wide range of business-IT transformation topics. He has the unique ability to cross business and IT boundaries to facilitate and streamline business-IT transformation strategies, and his workshops on business-IT architecture transformation have been widely attended by organizations worldwide.

Mr. Ulrich is the cofounder and President of the Business Architecture Guild, cochair of the OMG Architecture-Driven Modernization Task Force, and cofounder and Principal Instructor at Business Architecture Associates, Inc. Previously, he served in a senior management capacity at KPMG, including as Director of Reengineering Strategies, prior to leaving and forming his own company in 1990. Mr. Ulrich's work has been the basis for business and IT architecture transformation projects worldwide. Past clients include a wide range of large-scale corporations and government agencies worldwide.

Mr. Ulrich is a pioneer in the evolution of the practice and discipline of business architecture. As an originating author of *A Guide to the Business Architecture Body of Knowledge*® (*BIZBOK*® *Guide*), he played a pivotal role in framing the now generally accepted approach to business architecture. He is a Certified Business Architect® (CBA®) and continues to play a role in formalizing industry standards around the practice.

As developer of The Systems Redevelopment Methodology (TSRM/USRM), Mr. Ulrich defined many of the standards used in systems modernization within today's IT industry. He has also helped hundreds of organizations capitalize on knowledge built into their existing systems. A forerunner in the IT transformation industry, Mr. Ulrich was on the 1980 alpha test team for the first commercially available systems redevelopment tool and was instrumental in the development of the application reengineering and reverse engineering papers for IBM's Guide Project.

Mr. Ulrich has also served on the faculty of Northeastern Illinois University and facilitated numerous workshops, including sessions for SEI. He has lectured internationally to thousands of business and IT professionals and has testified as an expert witness on the use of IP within the computer field. Mr. Ulrich continues to serve as a software forensic and litigation support expert in technology-related cases. In 2005, he was awarded the Keeping America Strong Award by Rear Admiral Kevin F. Delaney (Ret.).

Mr. Ulrich has authored hundreds of articles appearing in major publications and journals. He is coauthor of *Business Architecture: The Art and Practice of Business Transformation*, *Information Systems Transformation: Architecture-Driven Modernization Case Studies*, and *Legacy Systems: Transformation Strategies*. He can be reached at experts@cutter.com.

About Cutter Consortium

Cutter Consortium is a unique, global business technology advisory firm dedicated to helping organizations leverage emerging technologies and the latest business management thinking to achieve competitive advantage and mission success. Through its research, training, executive education, and consulting, Cutter Consortium enables digital transformation.

Cutter Consortium helps clients address the spectrum of challenges technology change brings — from disruption of business models and the sustainable innovation, change management, and leadership a new order demands, to the creation, implementation, and optimization of software and systems that power newly holistic enterprise and business unit strategies.

Cutter Consortium pushes the thinking in the field by fostering debate and collaboration among its global community of thought leaders. Coupled with its famously objective "no ties to vendors" policy, Cutter Consortium's Access to the Experts approach delivers cutting-edge, objective information and innovative solutions to its clients worldwide.

For more information, visit www.cutter.com or call us at +1 781 648 8700.

CUTTER CONSORTIUM
Access to the Experts