

# Business Information Modeling: A Methodology for Data-Intensive Projects, Data Science and Big Data Governance

Torsten Priebe, Stefan Markus

Simplity s.r.o.

Vienna, Austria

{torsten.priebe,stefan.markus}@simplity.eu

**Abstract**—This paper discusses an integrated methodology to structure and formalize business requirements in large data-intensive projects, e.g. data warehouses implementations, turning them into precise and unambiguous data definitions suitable to facilitate harmonization and assignment of data governance responsibilities. We place a business information model in the center – used end-to-end from analysis, design, development, testing to data quality checks by data stewards. In addition, we show that the approach is suitable beyond traditional data warehouse environments, applying it also to big data landscapes and data science initiatives – where business requirements analysis is often neglected. As proper tool support has turned out to be inevitable in many real-world settings, we also discuss software requirements and their implementation in the Accuracy Glossary tool. The approach is evaluated based on a large banking data warehouse project the authors are currently involved in.

**Keywords**—Data Modeling, Project Methodology, Data Governance, Metadata, Information Catalog.

## I. INTRODUCTION

The data landscape of large, multi-national corporations is often historically grown and characterized by isolated stand-alone solutions. This results in inconsistent data definitions and reports. There is a lack of understanding what data exists and where it comes from. Communication problems lead to increased development costs and long project cycles. In this context, often data consolidation (e.g. data warehouse) initiatives are launched to facilitate harmonization. These projects are – especially in the financial services industry – supported by demands of regulators, who increasingly expect traceability and consistency of the data reported to them [2].

Often, another challenge is to start a data-intensive project with a limited implementation scope without compromising the broad applicability of the solution for the future. Experience shows that there is a large overlap in the data requirements, e.g. between risk and finance. However, to leverage this synergy, it must be recognized. It is therefore necessary to abstract the information needs from specific reporting requirements or IT systems. A clear language for communication between business and IT must be found.

Recent big data environments, also called “data lakes” [21] tend to be built in a less structured way than traditional ones like data warehouses (DWH). The strength of file- or

document-based data stores like Hadoop<sup>1</sup> or MongoDB<sup>2</sup> is that data models do not need to be defined upfront (so-called “late binding” or “schema on read”), but that does not mean that requirements analysis and modeling should be neglected [25,27]. In fact, the need of an information catalog to organize the data in a data lake has frequently been raised [13,21,24].

The Business Information Modeling (BIM) methodology presented in this paper addresses both challenges – agile, business-driven, stepwise design and development of data-intensive IT solutions as well as governing data within and beyond those solutions – by introducing a semantic business information model<sup>3</sup> as a central point of reference.

This work follows a design science research paradigm. Based on the methodology of Peffers et al. [11], we have above identified the problem. The rest of the paper is organized as follows: Section II discusses related work. Section III defines the objectives of the solution by introducing the BIM methodology, which is the main artifact designed and developed in our design science research. In section IV an extension to the Entity-Relationship model [9] is made, which builds the core of the BIM methodology. Sections V and VI apply the methodology to data-intensive (e.g. DWH) projects, (big) data governance and data science initiatives respectively. Early evaluations in various projects have shown that the BIM methodology can only be successfully applied if accompanied by proper tool support [3]. We therefore developed a tool called Accuracy Glossary, which is presented as a second artifact in section VII. The BIM methodology has been applied in a number real-life projects. The most recent one, where also the Accuracy Glossary tool is used, is presented in section VIII, providing the demonstration and evaluation of our design science research artifacts. Finally, section IX concludes this paper and discusses future work.

## II. RELATED WORK

### A. Methodology

Winter & Strauch [12] propose – similar to ours – a demand-driven approach to capture information

<sup>1</sup> <http://hadoop.apache.org>

<sup>2</sup> <http://www.mongodb.org>

<sup>3</sup> In [3] we used term “business data model” rather than “business information model”, however, as we clearly leave the semiotic level of syntax towards semantics [1], we amended the terminology.

requirements in DWH projects. They also identify the need to “homogenize” information requirements, but to do not cover how exactly this step should be performed and how the captured requirements are then carried over to the technical design and development. Our work picks up at that point with our detailed modeling (section IV) and according model-driven development approach (section V).

The application of models in the development process is a well-known concept in software engineering, normally referred to as model driven engineering (MDE) [5]. A prominent example is the model driven architecture (MDA) approach published by the Object Management Group (OMG)<sup>4</sup>. They are describing an automatic way of code generation by using conceptual models in combination with so-called query/view/transformation (QVT) rules.

Mazón & Trujillo [6] present an integration of the MDA approach into data warehousing. They emphasize the importance of a conceptual model for the underlying data model layer and illustrate how the derivation from one model to the other can be realized with a range of transformation steps. In contrary to this paper they focus on dimensional data models rather than an Inmon [7] style integrated DWH.

### B. Tool Support

Soares [4] provides a coverage of the data governance tool market and provides a categorization. He also identifies a business glossary as a key component. However, he does not go as deep as evaluating the tools against a particular design and development methodology.

Many vendors provide business glossary browsers as part their DWH solution stack. IBM InfoSphere Business Glossary<sup>5</sup> or ASG metaGlossary<sup>6</sup> for example allow collecting business terms and categories and linking them to physical data elements. Also data lineage is supported by commercial products, usually by utilizing metadata repositories, however focused on transformations extracted from an ETL tool (implementation lineage). An example is the IBM InfoSphere Information Governance Catalog<sup>7</sup>, which allows data tracing within IBM InfoSphere DataStage.

Unlike BIM and Accuracy Glossary, the available tools are rather limited in combining mapping specifications with business models (glossaries) and data requirements (specification lineage). Furthermore they are agnostic to a particular modeling approach and hence based on generic terms and categories rather than providing modeling guidance through functionality like attribute definitions, explicit inheritance or composite attributes (section IV).

The integrated software suite Kalido<sup>8</sup> includes its own fully automated business-model-driven approach. Within Kalido, the so-called Dynamic Information Warehouse is generated based on a business model created with the Business Information Modeler. BI tools are integrated via the Universal Information Director. Unlike traditional DWH

<sup>4</sup> <http://www.omg.org/mda/>

<sup>5</sup> <http://www.ibm.com/software/data/infosphere/business-glossary/>

<sup>6</sup> <http://www.asg.com/Portfolio/Content/BUSINESS-GLOSSARY.aspx>

<sup>7</sup> <http://www.ibm.com/software/products/en/infosphere-information-governance-catalog/>

<sup>8</sup> <http://www.kalido.com>

systems, Kalido stores data in a proprietary data store. Hence, Kalido tends to be suitable only for smaller scale data marts rather than enterprise settings.

## III. BUSINESS INFORMATION MODELING (BIM) METHODOLOGY

In [3] we have shown that a logical or physical data model alone is not suitable to enable harmonization and reuse in data-intensive environments. In order to address those goals adequately, the technical data models have to be accompanied by a semantic business information model, capturing the definitions and business rules plus the mappings to the different representations of the corresponding data artifacts.

As an answer to that need, we have developed a methodology called Business Information Modeling (BIM). The business information model represents unambiguous, robust definitions of business concepts and the linking of data to these concepts. The model is an organization-wide and unique catalog of information pieces that represent the requirements of all relevant user communities.

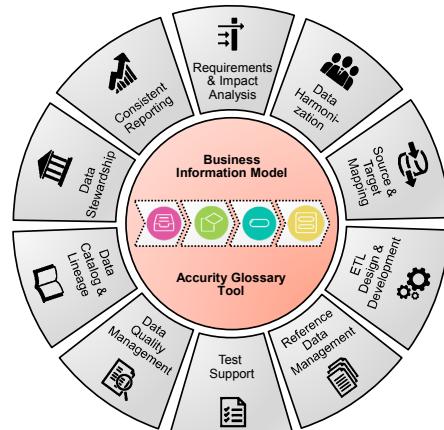


Figure 1. Business Information Modeling (BIM) wheel.

Figure 1 shows the idea of BIM as a wheel, with the business model in the center, surrounded by the ways it can be exploited. The colored symbols in the center of the wheel represent the main elements of the BIM approach, *Subject Areas*, *Entities*, *Attribute Definitions* and *Attributes*. The modeling approach itself is covered in detail in section IV, examples for BIM attributes are the family name of a customer or the nominal interest rate of a loan.

### A. Representation as Business Glossary

Our methodology ultimately breaks everything down into business attributes. However, due to the potentially high number of attributes, a static graphical representation as provided by most data modeling tools is not suitable. Sophisticated search and filtering functionality is needed to facilitate the harmonization of data.

Figure 2 shows a screenshot of the Accuracy Glossary tool (covered in detail in section VII), showing a searchable list of attributes and their descriptions.

Inherited	Composite	Entity Name	Attribute Definition Description	Type	Status
%		Deposit Account	Total Account Gr. Sum of balances i.	Amount	Approved
%		Threshold Rate P	Rate Valid For To	Indicator	Draft
%		Account	Account Balance	Balance on an acc.	Draft
Q		Deposit Account	Account Balance	Balance on an acc.	Draft
Q		Current Account	Account Balance	Balance on an acc.	Draft
Q		Term Deposit Acc	Account Balance	Balance on an acc.	Draft
Q		Notice Deposit Ac	Account Balance	Balance on an acc.	Draft
Q		Demand Deposit	Account Balance	Balance on an acc.	Draft

Figure 2. Attributes in Accurity Glossary.

### B. Linking to (Business) Data Requirements

Besides the elements of the business model *Subject Areas*, *Entities*, *Attribute Definitions* and *Attributes* it is core of the methodology to also maintain business requirements (usually more coarse grained and less formalized than attributes) and map them to elements of the business model. This is particularly relevant for data-intensive projects and will be discussed in detail in section V.

### C. Catalog Physical Data Representations

The more complex and diverse the data architectures are, the more important it is to keep track of what is there. The business information model turns out to be the perfect structure to tag various kinds of data items. A more in-depth coverage will be given in section VI, but let us note already here, that tagging of data items addresses not only cataloging and governance aspects. In particular, source systems or tables in DWH projects get linked to business entities or attributes as a high-level initial source mapping step.

### D. Detailed Mapping of Physical Data Representations

In our methodology the business information model acts as the central anchor point for mappings to various data layers (e.g. sources in DWH projects). In order to facilitate a model-driven development approach, more detailed mapping information is needed than just tagging an attribute to a certain source data field. Joins or selection criteria may be needed. We use the initial assignment (“tagging”, see above) of attributes to source systems or tables as an indication for the mapper, which detailed mappings he needs to add. For a more detailed coverage of mappings in data-intensive projects, see section V.

### E. Data Lineage – Putting the Pieces Together

If various layers in a data architecture are tagged with (or mapped to) elements of the business model, this information can be used to perform pragmatic data lineage queries. The business information model is defined in a way that the attributes represent the most granular atomic pieces of relevant information. If the physical data representations are linked on this level of granularity, the lineage information is quite precise. Of course we are speaking about specification rather than implementation lineage (see section II), so the development approach needs to ensure the implementation strictly follows the specification.

## IV. AN EXTENDED ENTITY RELATIONSHIP APPROACH

Now, how can a business information model be created? Unlike logical and physical data models, which can be acquired as pre-built industry reference models, a business information model is always organization specific, as the terminology and KPIs used will differ from organization to organization. Nevertheless, there are a number of sources that can help to define the organization-specific model. Depending on the setting, one source may be a reference data model of the respective industry (e.g. Teradata FSDM<sup>9</sup> or IBM BDW<sup>10</sup>). Further inputs that add content specific to the organization are corporate standards like a product catalog and existing legacy data models [3].

We base our business information model on the Entity-Relationship (ER) approach as originally proposed by Chen [9] with some extensions. As depicted by the four colored

<sup>9</sup> <http://www.teradata.com/logical-data-models/financial-services/>

<sup>10</sup> <http://www.ibm.com/software/products/en/banking/>

symbols in the center of the BIM wheel in figure 1, the main elements of our modeling approach are *Subject Areas*, *Entities*, *Attribute Definitions* and *Attributes*.

Entities are assigned to subject areas just to structure project scope and modeling work (and potentially to define governance responsibilities); they do not affect the formal model itself. Relationships are captured as entities and/or reference attributes (see below). The concept of attribute definitions is covered in the next subsection.

Like in every modeling exercise, you need a graphical representation to discuss entities, their relationships and key attributes. As we want to keep the notation easy to understand for business users, we adopt the notation proposed by Chen [9]. Figure 3 shows a (simplified) excerpt of an example banking business information model using the main structures *Customer*, *Employee* and *Account* (with substructures based on the products the bank offers, in this case *Current Account*, *Deposit Account* and *Loan Account*). As you can see, entities are represented as boxes, relationships as diamonds and attributes as bubbles.

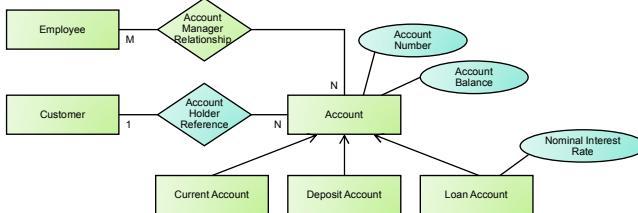


Figure 3. Simple example model in ER notation.

Please note two things: Firstly, we would never show all attributes in a graphical diagram, as you may have over a hundred attributes on one entity (see also section III on using a glossary representation). Secondly, note the different color and naming used for the two types of relationships, *Account Holder Reference* and *Account Manager Relationship*. For the sake of the example, we assume that an account can have only one account holder, but more than one account manager. As we in our methodology break everything down into entities and attributes, the former (a one-to-many relationship) becomes just a reference attribute, while the latter actually becomes an entity.

Reference attributes are a special construct of our methodology, similar to a foreign key, but without having to specify what the actual natural key attributes (in this case of the customer probably something like a customer number).

#### A. Attribute Definition Concept

As mentioned above, the main elements of the BIM methodology are entities (assigned to subject areas) and their attributes. The main issue with this classic approach is that you may use the same attribute name in different entities with a different description and hence inconsistent business meaning. For example, assume you want to capture the risk exposure of a banking customer both on account and on counterparty (customer) level. According attributes would be *Account.Risk Exposure Amount* and *Customer.Risk Exposure Amount*. You want to ensure that the definition of Risk

Exposure Amount is consistent, while having an additional description for the *Account* and *Customer* level.

In order to achieve this, we amended the classic ER methodology by what we call the “attribute definition” concept. Basically, we uniquely define an attribute name and its description as an “attribute definition”. This “attribute definition” can then be attached to multiple entities, making sure the description is consistent. This approach is actually similar to the way properties are handled in ontology languages like OWL [10], where the same (uniquely identified) property can be attached to more than one class.

Figure 11 (at the end of this paper) shows the whole metamodel of our methodology in UML notation (as implemented in the Accuracy Glossary tool). We will get to further details later, but for now note that an *Attribute* is basically just a combination of an *Attribute Definition* (which holds name and description) and an *Entity*. By doing so we ensure consistency of attributes across entities even if they are not in an inheritance hierarchy. An attribute can of course have an additional entity-specific description on top of the one of the attribute definition.

#### B. Explicit Inheritance

Many modeling tools support basic inheritance, i.e. the fact that an entity can have another entity as a parent. However, attributes assigned to the parent are only implicitly considered for the child. In the example in figure 3 *Account Balance* is defined as an attribute of *Account*, which implies it is also valid for *Deposit Account*, *Current Account* and *Loan Account*. While this implicit view may be sufficient for logical and physical data modeling, it is not suitable for detailed mappings. For instance, you commonly find core banking systems that manage only certain types of accounts. The inherited attributes need to be explicitly visible for a mapper’s view (see a deeper coverage in section V), while they may be overcomplicating for a modeler’s view. Recall the screenshot in figure 2. Inherited attributes are explicitly shown and can be conveniently hidden as indicated by the hierarchy icon.

#### C. Derived Attribute Definitions

Besides inheritance on entity level, we found it useful to also provide something similar on attribute definition level. Basically, we want to be able to define a specialized version of a more general attribute definition, e.g. *Account Balance Including EIR Adjustment*, which is based on the general *Account Balance*. The idea is that the description of *Account Balance* becomes an integral part of the description of *Account Balance Including EIR Adjustment* as shown in figure 4. Again this idea is borrowed from OWL [10], where a property can have a parent property.

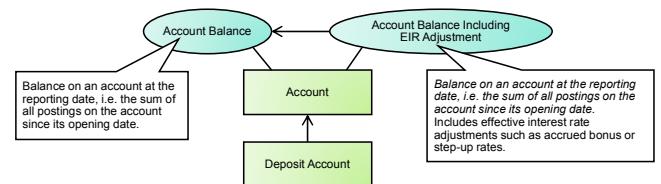


Figure 4. Derived attribute definition.

#### D. Composite Attributes

Linked to the idea of derived attribute definitions, we also often found the need to model non-atomic (composite) attributes. The classic example from modeling literature is an address being broken down into street, city, etc. In practice we found very real needs for being able to represent amounts in different currencies or according to different GAAP regimes (e.g. IFRS vs. local GAAP). Another example is time period attributes, which in fact consist of a unit of measure (days, weeks, months, years) and a count.

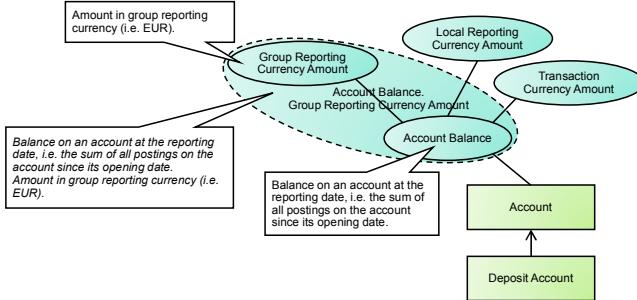


Figure 5. Composite attribute.

Figure 5 shows an example of a multi-currency amount as a composite attribute. The challenge from a tool perspective (as implemented in Accuracy Glossary) is that broken down (component) attributes are automatically created and descriptions are kept consistent. Again, as stated earlier for inheritance, while it may be sufficient for a modeler's view to have the component attributes implicit in the model (and hence not explicitly shown), they are explicitly needed for technical mappers.

#### E. Modeling Rules and Guidelines

In addition to the above main concepts a number of modeling rules and guidelines have been defined. A detailed coverage would go beyond the scope of this paper, but main topics covered are:

- Naming conventions for entities and attribute definitions
- Rules for descriptions, in particular general (attribute definition) vs. specific (attribute) descriptions
- Rules for handling relationships (one-to-many vs. many-to-many) using relationship entities and reference attributes
- Best practices for useful composite attributes
- Best practices on requirements assignment, status tracking and mapping in combination with explicit inheritance and composite attributes

### V. BIM FOR DATA-INTENSIVE PROJECTS

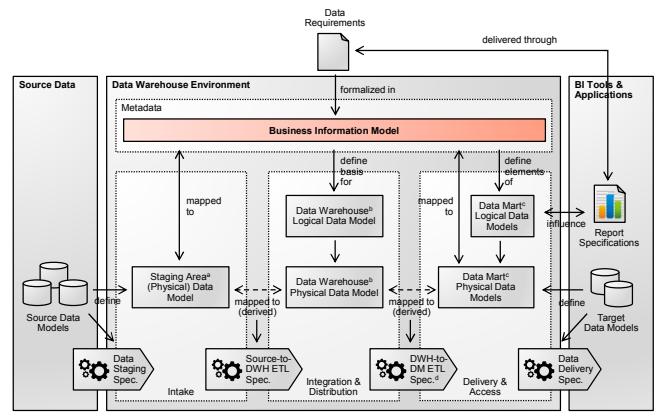
By data-intensive projects we consider all projects where the business requirements are largely data requirements and where large amounts of data need to be collected, moved, integrated and/or queried. Typical examples of such projects are of course data warehouse (DWH) implementations but also customer relationship management (CRM) initiatives or

data migration projects. We use a DWH implementation project as a role model throughout this section – this is also the type of project we have so far been able to demonstrate and evaluate our approach with as discussed in section VIII.

#### A. DWH Architecture and Model-Driven Approach

We base our architecture on Inmon [7] where the DWH is the central point of data integration and is modeled in third normal form (3NF). The DWH acts as the hub in a so-called “hub-and-spoke” setting and consists of integrated and historical data that is the basis for connected data marts resembling the spokes. Data marts can be physical as well as “virtual” based on views. In recent projects we have also seen “quick-win” interim solutions, temporarily feeding a data mart directly and introducing the robust DWH core layer only in a subsequent step. The DWH is fed by various heterogeneous operational source data stores. Between the different layers ETL processes extract the data and perform required transformations and load activities.

As discussed in section III the business data model provides a comprehensive and unique list of business entities with their attributes required to satisfy the information needs of all relevant user communities. As shown in figure 6, it is intended as a central anchor point for all mappings.



a. The staging area is sometimes also referred to as the acquisition layer.

b. The (core) data warehouse is sometimes also referred to as the integrated data layer.

c. Data marts within the DWH environment are sometimes also referred to as access layer.

Figure 6. Business information model as central mapping anchor point.

The requirements are connected to corresponding attributes in the business information model. In addition the attributes get mapped to all their physical representations in source systems, the DWH and data marts. The attribute acts as the central point of reference to enable an at least partly automated creation of source-to-target matrices (STMs) and as a result automated development of ETL jobs.

#### B. DWH Implementation Process

In the style of Kimball et al. [8], Winter & Strauch [12] and the Teradata Solution Methodology (TSM) [15] we base our methodology on seven main phases, *Plan*, *Analyze*, *Design*, *Build*, *Test*, *Deploy* and *Run*. Please note that this paper focusses on the *Analyze* and *Design* phases, which is why *Test*, *Deploy* and *Run* are omitted in figure 7.

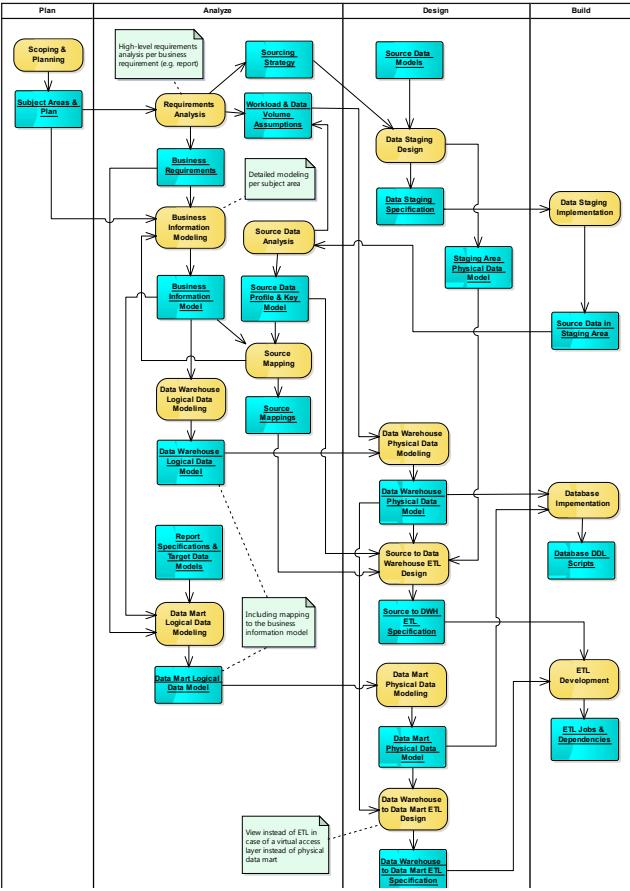


Figure 7. Data warehouse implementation process.

For our discussion the following process steps are of particular importance:

- The *Business Information Modeling* step creating the business information model itself has been covered in detail in section IV.
- The DWH logical data model is created in the *Data Warehouse Logical Data Modeling* step. Often, this step is based on an industry reference model like Teradata FSDM. In this case the main task is to select the relevant subset of the reference model needed for the implementation scope. The logical modeling is driven by the attributes in the business model. Hence, this step provides also a mapping from the business to the DWH logical model.
- The physical data model of the DWH is derived from the logical one in the *Data Warehouse Physical Data Modeling* step. Again, the mapping to the business information model is included in this task.
- Data marts are modeled in the *Data Mart Logical Data Modeling* step. This step can be omitted if the target model is defined by a third-party application and hence the physical model is externally given.
- Like for the DWH, the physical data model is derived from the logical one in the *Data Mart Physical Data Modeling* step.

*Physical Data Modeling* step. Again, mapping to the business information model is included in this task.

- Elements of the business information model are mapped to source (or rather staging area) physical data models in the *Source Data Mapping* step.
- Source-to-target matrices (STMs) are created in *Source to Data Warehouse* and *Data Warehouse to Data Mart ETL Design*. They serve as the specification for ETL jobs (or views).
- Last not least, the ETL jobs or views are developed based on the STMs in the *ETL Development* step.

Please note that we are for this paper focusing on the data management rather than the reporting parts of the process, report specification and development are therefore not shown. Although figure 7 implies a waterfall approach, it is in fact much more parallelized and agile in real-life settings. We split the business information modeling work into subject areas. As data harmonization discussions tend not to find a clear end, but the marginal improvement of the results decreases significantly after a 4-week timeframe for a certain topic, a time-boxing approach with a 4-week cycle and up to 4 subject areas in parallel has turned out to be most suitable.

If feasible it is also desirable to define smaller chunks of scope (like “sprints”) that can go into design and development. Source mapping can usually start by subject area following the same 4-week cycle as the modeling. For ETL design and development it may be more suitable to have slightly bigger chunks of self-contained scope that allow end-to-end testing.

Extensions after the initial implementation project will allow skipping parts of the flow. If a certain BIM attribute is already present in the DWH, it only needs to be added to the corresponding data mart and report (skipping DWH modeling, source mapping and corresponding ETL work). If a new BIM attribute is not present in the DWH, but already in the staging area, at least the data staging can be skipped.

#### C. Source and Target Mappings with BIM

The key to the approach is that all data models, sources, the DWH itself and data marts, are mapped to BIM attributes. Only then source-to-target mappings (ETL specifications) can be derived following a model-driven development paradigm. The challenge is to find the right balance between structured and unstructured mapping definitions. As shown in the metamodel in figure 11, we support entity and attribute-level mappings. Besides the payload attribute containing the actual value, a mapping can contain one or more joins (to define how to reach the target table) and one or more selections (i.e. filter criteria).

Consider a BIM attribute *Deposit Account.Interest Rate Type* mapped to a field *CLAS\_VAL.CLAS\_VAL\_CD* joining via *CLAS\_VAL.CLAS\_VAL\_ID = AGMT\_CLAS\_XREF.CLAS\_VAL\_ID* with a selection on *AGMT\_CLAS\_XREF.CLAS\_SCHM\_CD = 'INT\_RATE\_TYP'* (following a generic modeling approach like in industry models such as Teradata FSDM). A screenshot of Security Glossary is shown in figure 8. If source mappings for are done accordingly, source-to-DWH ETL specifications can be derived.

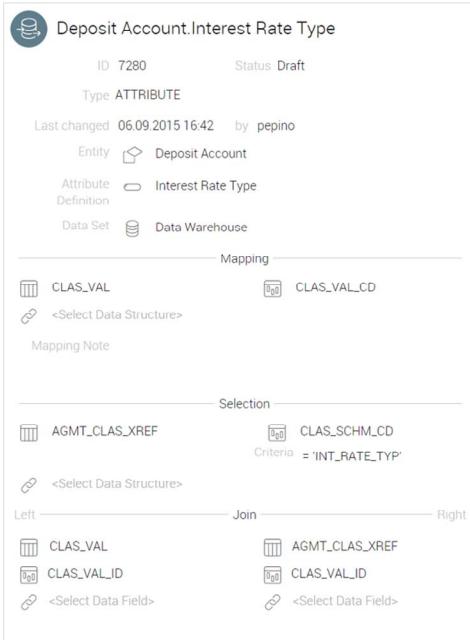


Figure 8. Mapping in Security Glossary.

## VI. (BIG) DATA GOVERNANCE AND DATA SCIENCE

### A. BIM as Information Catalog

So far we have discussed the use of BIM in the environment it was originally designed for – data-intensive projects like data warehouse (DWH) implementations. As discussed in section VIII, this area has been well studied and evaluated in real-life projects. In organizations, where BIM has been applied within DWH project cycles, it has proven also very useful for governance. Data stewards and owners have been assigned on subject area, entity or even attribute level and stored as additional metadata. The tagging of data layers within the DWH environment is used not only during development, but also in production, e.g. to perform data lineage and impact analyses. In this section, we want to take this approach further and extend it to big data environments beyond traditional relational databases. Please note that this section covers work in progress, which still needs proper demonstration and evaluation in real-life settings.

With the emergence of the big data hype, an old problem of IT, the integration of structured and unstructured information, is experiencing a renaissance [22,23]. The “variety” aspect extends this to various data formats and stores, sometimes referred to as a “data lake” [21]. Ferguson [13] takes up on this and coins the term “data reservoir” arguing that data needs to be “refined” to be of business value. In particular, he argues that an “information catalog” is needed in order to:

- Document where data resides and came from
- View data lineage
- Name and describe data
- Define shared business vocabulary terms
- Classify data

Picking up on this idea of an information catalog, we developed the big data landscape scheme shown in figure 9.

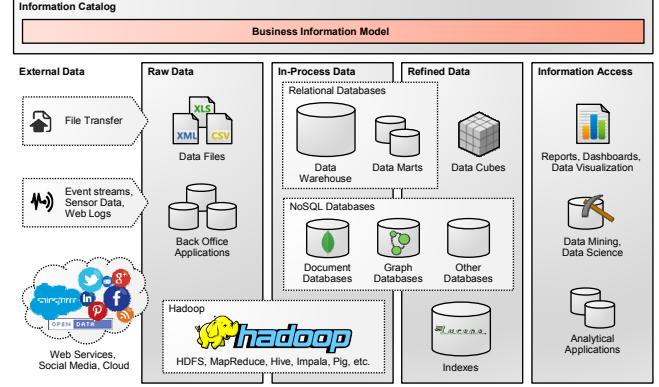


Figure 9. Big data landscape with BIM as information catalog (based on [13,14]).

Simply listing the various data stores in a repository is not sufficient, a proper tagging or categorization is needed. While most existing approaches are agnostic to what kind of tags or categories are used, we argue that the BIM elements (subject areas, entities and attributes) build the perfect basis for a cross-system categorization. For example, imagine a use case where externally acquired prospect data (in Hadoop) needs to be matched against internal customer data (in a relational DWH). Both data sets in fact contain representations of an *Individual.Name* BIM attribute.

Information items in the big data era are of both structured and unstructured nature. In section V we have discussed tagging (and mapping) of relational databases. The tagging of textual documents based on an ontology (resp. business information model) has already been proposed in [19] using semi-automatic text categorization [20].

TABLE I. BIM VS. DATA STORE TERMINOLOGY

BIM Terminology	Data Store Terminology			
	Relational Database	File System, Hadoop	Document Database <sup>b</sup>	Graph Database <sup>d</sup>
Data Set	Database, Schema	Directory <sup>a</sup>	Database	Database
Data Structure	Table, View	File <sup>a</sup> , Document	Collection	Node, Relationship
Data Element	Column	Field <sup>a</sup>	Field <sup>c</sup>	Property

a. Hive [<https://hive.apache.org>] or Impala [<http://impala.io>] also use terms like Table and Column.

b. Based on MongoDB [<http://www.mongodb.org>].

c. Other terms exist for other file formats (e.g. Column for Parquet [<http://parquet.apache.org>]).

d. Based on Neo4j [<http://neo4j.com>].

As you see in figure 11 (at the end of this paper), our BIM methodology and Security Glossary capture physical data representations using three concepts, *Data Set* (corresponding to a relational database or schema), *Data Structure* (table or view) and *Data Element* (column within a table or view). We argue that this approach is general enough to capture a large variety of data stores you may find in a big data landscape as shown in table I.

Raw data files follow a similar paradigm as tables in a relational database, given their format allows an assignment

of fields to a file structure. Depending on whether this structure is known the lowest granularity for tagging with BIM elements is the file (represented as *Data Structure*) or the individual field (represented as *Data Element*).

Unstructured files (e.g. text documents) will be handled differently from structured ones. Rather than trying to assign a schema, those documents will rather be indexed using technologies like Apache Lucene<sup>11</sup>. The lowest granularity for tagging with BIM elements then is the individual document (represented as *Data Structure*).

Document databases such as MongoDB<sup>12</sup> again need to be handled slightly differently. The term document here refers to a JSON document, which basically represents a single data record. A document may contain sub-documents, i.e. there may be a field *contact.phone* (like composite attributes in BIM). The item corresponding to a database table is a collection. Collections do not enforce a schema, i.e. the assignment of fields to collections may be ambiguous. Hence, the lowest granularity for BIM tagging is the collection (as *Data Structure*) or field (as *Data Element*, if we restrict a collection to a fixed document structure).

Last not least, graph databases such as Neo4j<sup>13</sup> use nodes and relationships (which can be captured as *Data Structures*) and their properties (*Data Elements*).

Bottom line, we are convinced that the tagging approach and structure we have introduced and implemented in the Security Glossary tool is suitable to create an end-to-end information catalog for a diverse big data environment. Of course, interfaces are needed to make all those schemas available in Security Glossary for tagging. This would create various cross-system data lineage and impact analysis capabilities and provide also the perfect basis to support data acquisition and preparation in data science initiatives.

### B. BIM for Data Mining and Data Science

Data science is commonly understood as an interdisciplinary field about extracting knowledge or insights from large volumes of data in various (structured or unstructured) forms [16]. It can be seen as a continuation of data mining and knowledge discovery in databases (KDD). As a data science initiative is often open ended and iterative it may seem inappropriate to define a process for it. However, the fact that it is open ended does not contradict the value of a process [27]. In fact, there are a number of proposals for key steps of a data science workflow [16,17,18], which we integrate into our proposal in figure 10:

- Raw data is collected in the *Data Acquisition* step.
- This data is pre-processed in the *Data Preparation* step, which involves understanding the format of the data (“schema on read”) and data cleansing.
- In the *Exploration & Visualization* step, also known as exploratory data analysis (EDA) [16], it may be realized that the data is not actually providing what is needed or in the needed quality. It hence may be

necessary to collect more data or to spend more time on data cleansing.

- *Statistical Modeling* represents the application of algorithms like k-nearest neighbors or alike [20]. Obviously, big data technologies may imply new implementation approaches (e.g. MapReduce) of those algorithms.
- The results are interpreted, reported or otherwise communicated in the *Reporting* step. This may take place within the organization or by publishing a paper in an academic conference or journal [16].
- The goal may be to build or prototype a “data product”, e.g. a recommendation system. The key here is that this data product then gets incorporated back into the real world in the *Deployment* step, and users interact with that product [16].

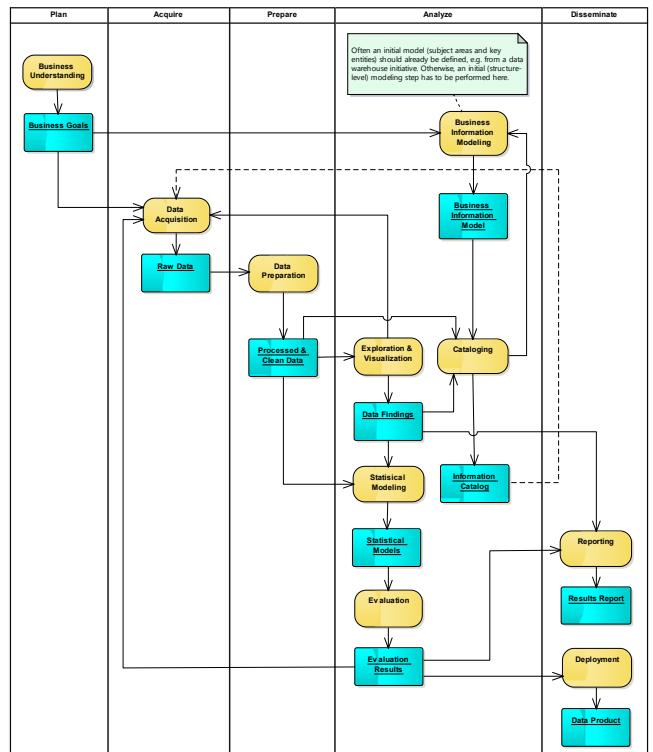


Figure 10. Proposed data science workflow.

The proposals in [16,17,18] neglect one in our opinion inevitable step, which we also see missing in many data science attempts in practice: the definition of business requirements, or (in a broader sense) the setting of clear business goals [27]. Organizations often feel the need to participate in the big data and data science hype and start collecting large volumes of data in a data lake. Those activities fail, if no clear analysis objectives are defined beforehand. We hence amend the mentioned data science workflow proposals by an initial step, which had already been introduced for data mining in CRISP-DM, *Business Understanding*, focusing on understanding the project objectives and requirements from a business perspective.

<sup>11</sup> <http://lucene.apache.org>

<sup>12</sup> <http://www.mongodb.org>

<sup>13</sup> <http://neo4j.com>

Accordingly, we add an *Evaluation* step, in which the analysis results are thoroughly reviewed to be certain they properly achieve those objectives [26].

We argue that a business information model provides the perfect means of defining the data needs for fulfilling the defined analysis objectives. Furthermore, a BIM-based information catalog will then help in identifying the right data set for the analysis (*Data Acquisition* step). As we are talking about “late binding” type data sets, the *Data Preparation* and *Exploration & Visualization* results will

facilitate a refinement of the information catalog in a *Cataloging* step.

## VII. ACCURACY GLOSSARY TOOL

We have implemented the modeling functionality in a tool called Accuracy Glossary. The implementation is a Java-based web application using a relational database as a repository, based on the metamodel in figure 11.

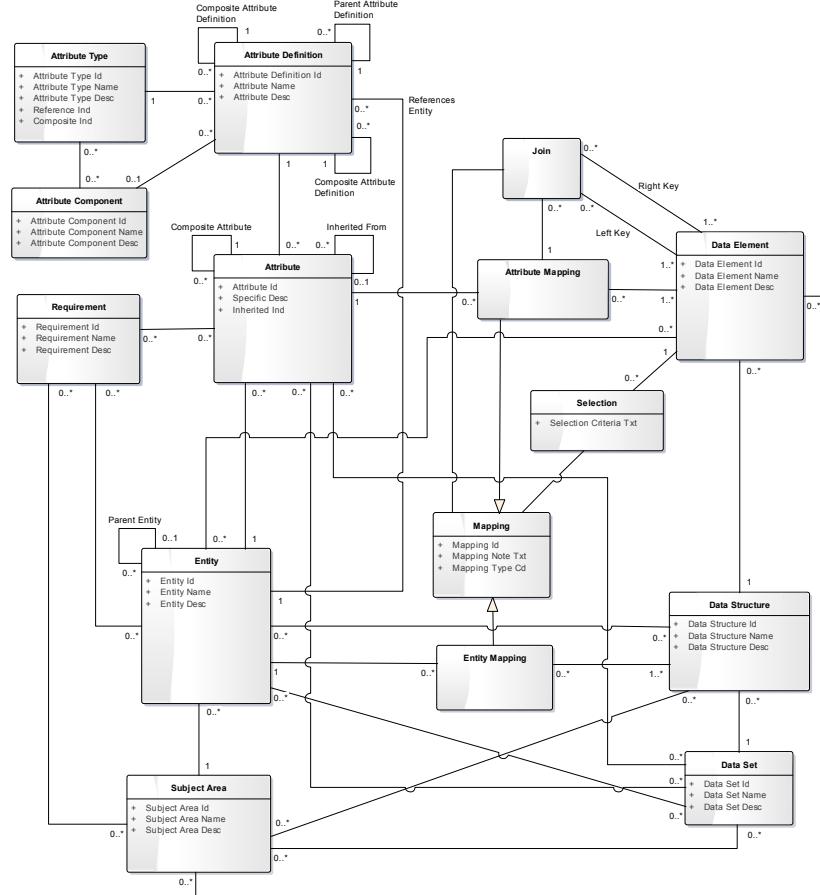


Figure 11. BIM metamodel (as implemented in Accuracy Gossary).

In addition to this relational implementation, we continue to use our research prototype presented in [3] for prototyping. Semantic Web technologies help both to perform data lineage and impact analyses as well as to support inheritance and composite attributes by inferring certain information automatically. We have represented our metamodel in OWL [10]. This representation is imported into Apache Jena<sup>14</sup>, containing a rule-based inference engine capable of processing custom inference rules and mechanisms to store, manage and query OWL ontologies. We plan to evaluate also other graph database technology (e.g. Neo4j<sup>15</sup>).

## VIII. EVALUATION

The BIM methodology and according model-driven development approach has been applied in various DWH implementations. In particular the most recent one at Bank of Ireland in Dublin, where also Accuracy Glossary is in use, provides a suitable setting to demonstrate and evaluate our methodology in accordance with the design science research paradigm [11]. The current version of the business information model stored in Accuracy Glossary at Bank of Ireland consists of 285 entities in 15 subject areas with 11,724 attributes using 1,483 distinct attribute definitions.

<sup>14</sup> <http://jena.apache.org>

<sup>15</sup> <http://neo4j.com>

The BIM methodology and Accuracy Glossary tool have been gradually evaluated and improved between and within the projects where it has been applied:

- The modeling work has been structured into subject areas and is performed using a time-boxing approach with 4-week cycles as discussed in section V.
- Explicit inheritance, derived and composite attributes have been introduced in Accuracy Glossary as discussed in section IV (was done manually before).
- Mapping functionality was added to Accuracy Glossary as discussed in section V (earlier projects used a separate mapping tool requiring time-consuming exports of the model)
- Simple workflow and backlog tracking functionality has been identified as further useful for Accuracy Glossary (not yet implemented).

As mentioned, the application of BIM to big data environments and data science is work in progress. We did not yet have the chance to evaluate our ideas presented in section VI in a real-life setting.

## IX. CONCLUSIONS AND FUTURE WORK

We have presented a methodology to gather and structure data requirements to improve data-intensive projects and enable data governance. The methodology facilitates data harmonization by introducing a semantic business information model as a central point of reference on top of physical and logical data models. We have extended the BIM approach beyond traditional relational databases and positioned the business information model also as an information catalog in big data environments.

Based on the mapping functionality in Accuracy Glossary (see section V) we intend to work on a semi-automatic creation of STMs (source-to-target matrices, i.e. ETL specifications) which are then translated into ETL code. Next steps are to move towards full automation and ETL (or view) code generation. The challenge seen in our projects is to find the trade-off between preciseness and formal correctness of the models (needed for automatic code generation) and model readability and modeling effort spent.

Further future work is devoted to applying the BIM methodology to real-life big data environments and data science initiatives. As discussed in section VI, the business information model as an information catalog should significantly improve the data acquisition and preparation.

## ACKNOWLEDGMENTS

We would like to thank our project mates at Bank of Ireland in Dublin for their support in evaluating and fine-tuning both the BIM methodology and the Accuracy Glossary tool in a large-scale data warehouse project.

## REFERENCES

- [1] P.E. Wisse, Semiosis & Sign Exchange: Design for a Subjective Situationism, Including Conceptual Grounds of Business Information Modeling. *Information Dynamics*, 2002.
- [2] Basel Committee on Banking Supervision, Principles for Effective Risk Data Aggregation and Risk Reporting (BCBS 239), Bank for International Settlements, 2013.
- [3] T. Priebe, A. Reisser, D.T.A. Duong, "Reinventing the Wheel?! Why Harmonization and Reuse Fail in Complex Data Warehouse Environments and a Proposed Solution to the Problem", 10th International Conf. on Business Informatics (WI 2011), Zurich, 2011.
- [4] S. Soares, Data Governance Tools: Evaluation Criteria, Big Data Governance, and Alignment with Enterprise Data Management. Mc Press, 2015.
- [5] D.C. Schmidt, "Guest Editor's Introduction: Model-Driven Engineering", *IEEE Computer*, 39(2), pp. 25-31, IEEE Computer Society, 2006.
- [6] J. Mazón, J. Trujillo, "An MDA Approach for the Development of Data Warehouses", *Decision Support Systems* 45(1), pp.41-58, 2008.
- [7] W.H. Inmon, *Building the Data Warehouse*, 4th edition, Wiley, 2005.
- [8] R. Kimball, M. Ross, W. Thorntwaite, J. Mundy, B. Becker, *The Data Warehouse Lifecycle Toolkit*, 2nd edition, Wiley, 2011.
- [9] P. Chen, "The Entity-Relationship Model – Toward a Unified View of Data", *ACM Transactions on Database Systems* 1(1), 1976.
- [10] M. Dean, G. Schreiber, OWL Web Ontology Language Reference, W3C Recommendation, 2004. <http://www.w3.org/TR/2004/RECowlr-20040210/>
- [11] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, "A Design Science Research Methodology for Information Systems Research", *Journal of Management Information Systems*, 24(3), pp. 45-77, 2007.
- [12] R. Winter, B. Strauch, A Method for Demand-driven Information Requirements Analysis in Data Warehousing Projects, 36th Hawaii International Conference on System Sciences (HICSS'03), 2003.
- [13] M. Ferguson, "Building an Enterprise Data Reservoir and Data Refinery", European TDWI Conference 2015, Munich, 2015.
- [14] C. Ballard, C. Compert, T. Jesionowski, I. Milman, B. Plants, B. Rosen, H. Smith, *Information Governance Principles and Practices for a Big Data Landscape*, IBM Redbooks, 2014.
- [15] Teradata Solutions Methodology (TSM), Version 7.0.7, Teradata Corporation, Dayton, OH, 2012
- [16] C. O'Neil, R. Schutt, *Doing Data Science: Straight Talk from the Frontline*, O'Reilly, 2013.
- [17] M. Lang, *Data Preparation in the Hadoop Data Lake*, Teradata, EB8458, 2014.
- [18] P. Guo, "Data Science Workflow: Overview and Challenges", *blog@CACM*, Communications of the ACM, 2013
- [19] T. Priebe, J. Kolter, C. Kiss, "Semiautomatische Annotation von Textdokumenten mit semantischen Metadaten." *Wirtschaftsinformatik* 2005, pp. 1309-1328, Physica, 2005.
- [20] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys* 34(1), 2002.
- [21] T.H. Davenport, *Big Data at Work: Dispelling the Myths, Uncovering the Opportunities*, Harvard Business Review Press, 2014.
- [22] T. Priebe, G. Pernul, "Towards Integrative Enterprise Knowledge Portals", Twelfth International Conference on Information and Knowledge Management, pp. 216-223. ACM, 2003.
- [23] H. Baars, H.-G. Kemper, "Management Support with Structured and Unstructured Data – An Integrated Business Intelligence Framework", *Information Systems Management* 25(2), 2008.
- [24] S. Soares, *Big Data Governance: An Emerging Imperative*, Mc Press, 2012.
- [25] S. Hoberman, *Data Modeling for MongoDB: Building Well-Designed and Supportable MongoDB Databases*, Technics Publications, 2014.
- [26] C. Shearer, "The CRISP-DM Model: The New Blueprint for Data Mining," *Journal of Data Warehousing*, 5(4), 2000.
- [27] J.S. Saltz, "The Need for New Processes, Methodologies and Tools to Support Big Data Teams and Improve Big Data Project Effectiveness", 2015 IEEE International Conf. on Big Data, Santa Clara, CA, 2015.