Standardizing Data Dimensions of Healthcare Data Warehouses

Richard E. Biehl, Ph.D.[1]

Abstract. Database and storage technologies now enable extremely large and complex data warehouses of very high dimensionality, and design patterns have matured to the point where the design of such data structures is becoming routine. This paper addresses the major remaining problem in dimensional modeling: How to semantically select the optimum number and types of dimensions. Using ontologies, with the Basic Formal Ontology (BFO) as a foundation, the optimum choice for data warehouse dimensions can be made in support of maximum interoperability and semantic reasoning support.

Keywords: Ontology, Dimensional Modeling, Database Design, Data Warehouse, Basic Formal Ontology (BFO), Star Schema, Semantic Web

1 Introduction

Dimensional data warehouses are large aggregate databases that bring together an enormous amount of consolidated subject-oriented enterprise data for aggregation and analysis. The size and scale of today's data warehouses are largely enabled by the advances in database and storage technologies that have taken place over the past thirty years, while the semantics of such warehouses have evolved in parallel under different driving forces.

A central paradigm of dimensional warehousing is the separation of fact data from dimensional data, with facts comprising the quantitative and observational data of interest, and the dimensions comprising the variety of contexts in which those facts are collected and understood. The facts can be said to exist in the center of a cloud of dimensions that was often discussed by drawing a star and labeling the points of the star according to the desired dimensionality of the facts that would be included. (see Figure 1) In those early days of warehousing (e.g., the late 1970s), the five-pointed star seemed appropriate because database technologies were only mature enough to handle about five dimensions in a reasonable design, and the larger warehouses that would require more than five simple dimensions were beyond the cost-effective storage technologies of the day.

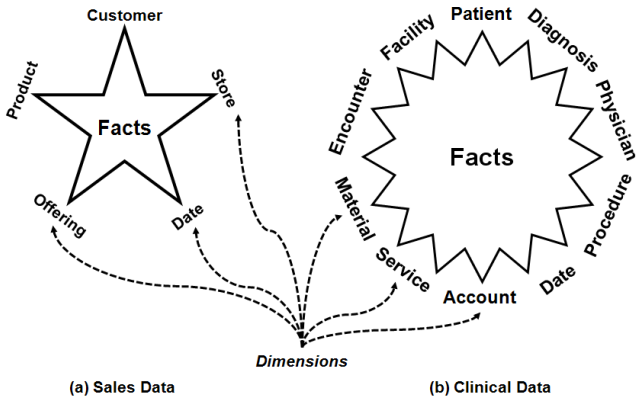

Fig. 1. Evolution of the "star" schema paradigm in data warehousing: (a) A traditional sales data view with a 5-point star (c. 1987), and (b) a much higher dimensional clinical data star with 15-20 dimensions (c. 2007).

[1] Data-Oriented Quality Solutions, 2105 Whitfield Lane, Orlando, Florida 32835-5940 USA, rbiehl@doqs.com

Today's warehouses routinely exceed the 5-dimension limitation of the traditional star, but the vernacular naming of the star schema persists. Figure 2 depicts a 19-dimensional star schema for a clinical data warehouse at a large academic medical center[2]. The figure illustrates a central fact table surrounded by the 19 dimensions, although the simple clarity of the 5-pointed star is less obvious as the dimensions scale up.

Each of the dimension boxes in Figure 2 represents a rich set of data constructs that have been generalized over time to the design pattern depicted in Figure 3. A significant subset of this design pattern is derived from Kimball. [1] The design pattern allows dimensional data warehouses to be designed and implemented very quickly, and the standardization supported by the pattern makes it much easier for data analysts to learn to use these warehouses effectively.

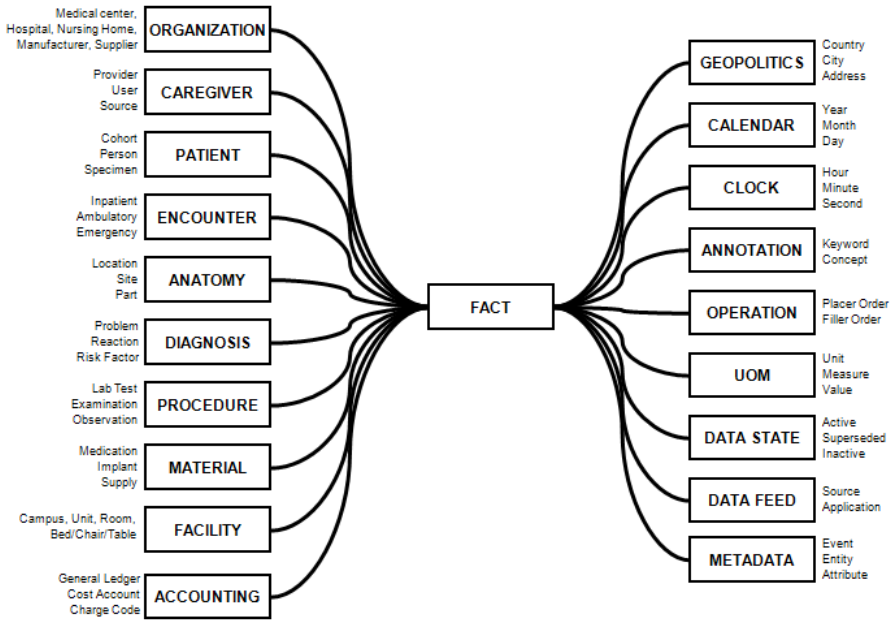

Fig. 2. Clinical Data Warehouse Dimensional Architecture.

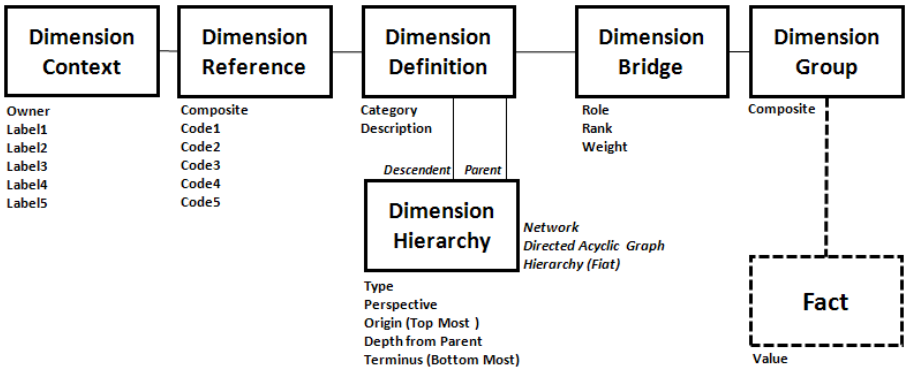

Fig. 3. Dimensional anatomy design pattern. *Bridge-Group* constructs allow individual facts to reference multiple entries in a single dimension, and *Context-Reference* constructs allow entries to be identified by the natural keys in multiple source or destination contexts. The *Hierarchy* construct allows for aggregation, disaggregation, and associations among entries independent of the local dimensionality of the facts.

[2] The examples presented throughout this paper are taken from the author's design work in a variety of clinical settings, but there is nothing about ontologies or these design heuristics that are specific or tailored to the healthcare industry.

Although there are many dimensions in a star schema warehouse, their structures and behaviors are all the same. Because of this, a user of the clinical warehouse depicted in Figure 2 is quickly able to approach and use the research bioinformatics warehouse depicted in Figure 4 because the same design pattern underlies every dimension regardless of the clinical, scientific, or business meaning of the entries in the dimension. In this sense, all star schema data warehouses are the same.

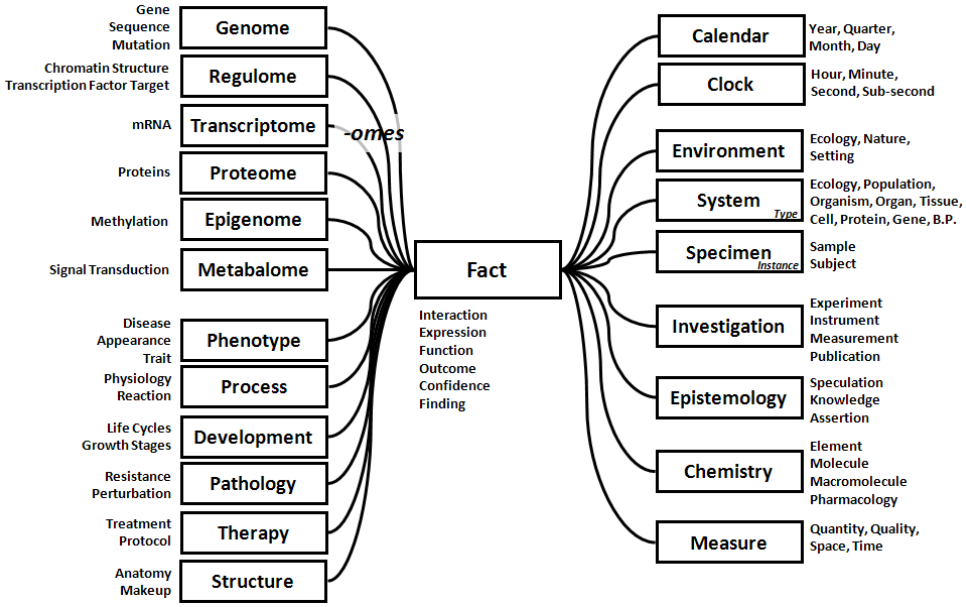

Fig. 4. Bioinformatics Data Warehouse Dimensional Architecture.

While the design pattern determines the structure of each of the various warehouse dimensions, it ultimately says nothing about the content or semantic meaning of those dimensions. The pattern is neutral with respect to defining a grouping of concepts into the dimensions. The expected 15-20 dimensional star schema, with 3-8 subdimensions defined for each dimension, could just as easily have been 50-100 distinct dimensions without violating the design pattern. While a 20-dimensional star is surely preferable to a 100-dimensional star when designing and building a relational database, a technology bias shouldn't be the basis for such a decision. The number of dimensions should be determined as naturally as possible from the problem domain and requirements being addressed by the implementation of the warehouse. Of the permutations of dimensionality that might make sense for a particular data warehouse, one of them should be determinable as the best.

2 Logical vs. Physical Dimensions

The dimensions of the data warehouse embody the set of concepts that provides the semantic context for all of the facts to be stored in the warehouse. Each distinct concept constitutes a logical dimension of the warehouse. Logical dimensions are typically collected into physical dimensions within the database technology that implements the warehouse. The foreign keys within the warehouse Fact table point to the physical dimensions, and within the physical dimensions some form of Category property identifies the appropriate logical dimension, allowing the logical dimension to be often referred to as a subdimension.

An example can illustrate: If a clinical data warehouse needs to store facts about drugs and implants, then these might constitute two logical dimensions of the warehouse. Each could be implemented as a separate physical dimension, but more likely will be consolidated into a single physical dimension (e.g., Material) containing both Drug and Implant logical dimensions (i.e., subdimensions).

The distinction is somewhat arbitrary, but is extremely important to the design of a particular data warehouse. The number of logical dimensions needs to be appropriate to the scope of the entire warehouse and the problem domains it is meant to address. The number of physical dimensions is constrained by the database technology being used to implement the data warehouse. While a typical data warehouse might have 50-100 logical dimensions, no commercially available database technology can effectively handle a Fact table design with 50-100 foreign keys.

One typically sees complex data warehouses today being comprised of 15-20 physical dimensions, across which are spread the 50-100 logical dimensions – subdimensions – needed to properly represent the data. Diagrams of star schema warehouses (see Figures 2 & 4) usually include representative notations about subdimensions outside the boxes that represent the dimensions, although such notations are rarely exhaustive.

The data warehouse designer must determine the optimum number of physical and logical dimensions that will be included in a design. The number of physical dimensions is a *performance* concern, and the number of logical dimensions is a *domain* concern. Fortunately, there are algorithmic mechanisms for changing the number of physical dimensions in a warehouse design so that the logical view of the data remains unchanged. As a result, these two concerns can be treated separately.

Early warehouse designs of low dimensionality can be changed to become higher-dimensional warehouses without losing the domain perspective supported by the logical dimensions contained in those physical dimensions. This provides a pathway (see Figure 5) from early implementations to later implementations as the warehouse domain expands over time and database technologies improve to allow better performance at higher dimensionality. It removes the pressure on the designer to get the first-generation warehouse design perfect.

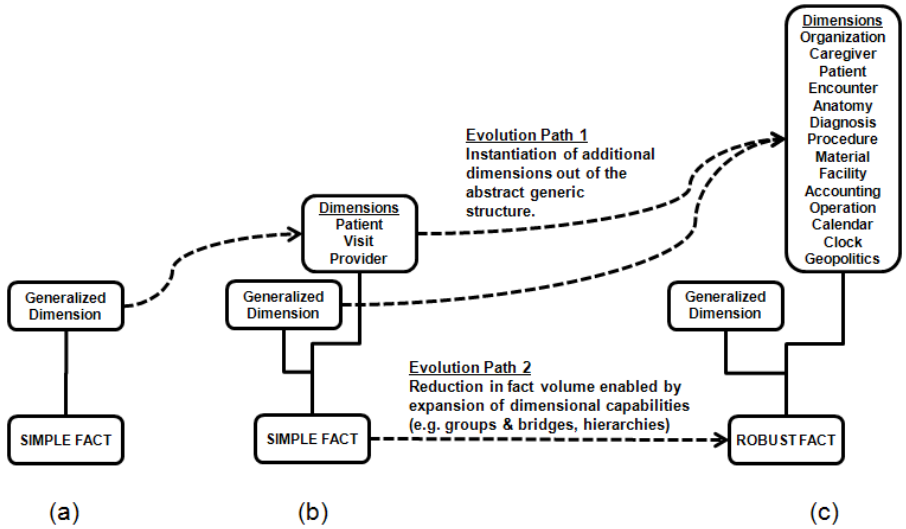

Fig. 5. Dimension Maturity Pathway: (a) 1-dimension warehouse with all logical dimensions generalized into the single physical dimension, (b) the i2b2 design, with three specific logical dimensions moved into their own physical dimensions, and (c) a clinical warehouse with 14 physical dimensions, and everything else left in the generalized dimension. The generalized dimension is always present, and embodies any logical dimensions that don't map into one of the defined physical dimensions. The number of logical dimensions remains constant.

Figure 5 illustrates the pathway from lowest to higher dimensionality. Each warehouse will vary along the continuum from one extremely generalized dimension that can define and store any dimensional reference, to a collection of very specific dimensions that each store definitions that share semantic context and meanings. The Informatics for Integrating Biology and the Bedside (i2b2) warehouse in Figure 5-b is a publicly available warehouse design that specifically instantiates only a few of the dimensions commonly seen in a large-scale warehouse. The clinical warehouse design in Figure 5-c illustrates a typical warehouse star design in which many more dimensions have been instantiated. Viewed correctly, these can all be seen as shared designs along a continuum, and not conflicting alternative designs. The differences are in the number and types of dimensions that are instantiated out of the generic dimension. The challenge is to instantiate the right dimensions correctly.

3 Dimensions: An Ontological Problem

If we designed a 1-dimensional warehouse, then that dimension would need to list all of the necessary concepts; however, the dimensionality would be insufficient to take advantage of the star schema dimensional model espoused here. A simple affinity principle could be used to split the one dimension into many dimensions: Divide up the list of concepts into affinity groupings such that the concepts within each resulting dimension share more in common with each other than any of them do with the concepts placed into the other

affinity groupings. This process will guarantee a multi-dimensional warehouse but will exhibit high variability in the number of dimensions depending upon the conceptual sensitivity used in the analysis. Figure 6 illustrates some clusters of clinically-related data that might be identified through some form of research or brainstorming during the very early stages of a data warehouse development project. Each cluster represents a potential perspective against which a user might want to analyze or aggregate data in the warehouse, and so each is a potential warehouse dimension.

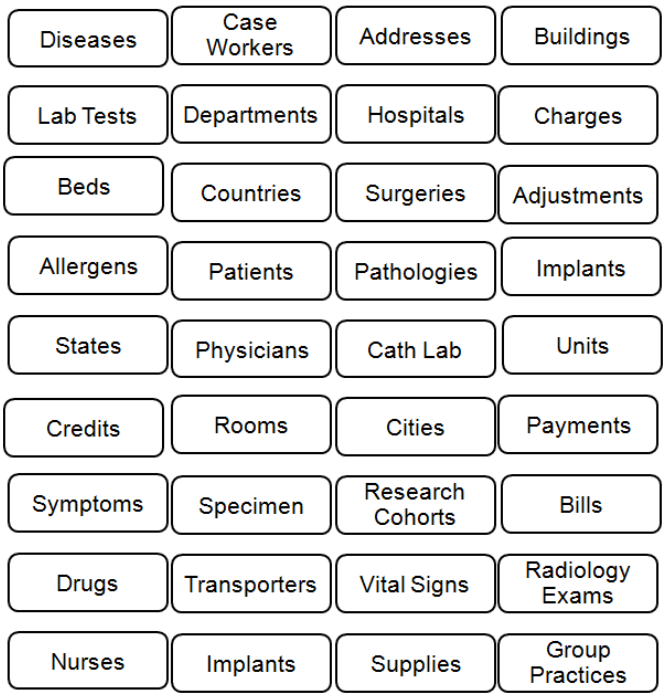

Fig. 6. Examples of logical groupings that might be identified as candidate warehouse dimensions during an affinity analysis of project-relevant data.

Because the number of clusters quickly exceeds the number of physical dimensions anticipated for the warehouse, each cluster is considered to be a logical subdimensions of one or more physical dimensions that now need to be identified. A number of useful heuristics have emerged for helping with the affinity analysis of the logical clusters:

1. Clusters that represent systems of some form will typically reside in different physical dimensions if their represent systems of different scale: societal, organismic, or mechanical. For example, societal systems like *hospitals*, *group practices*, and their associated *departments* can be grouped together as Organizations, while organismic systems like *patients*, *physicians*, and *case workers* would belong in a different dimension.
2. Clusters that differ in focusing on people, places, or things don't typically resolve into the same physical dimension. As a result, *patients*, *countries*, and *drugs* would not be expected to end up as subdimensions of the same dimension.
3. Clusters that represent internal and controllable data don't typically map into the same dimension as external data that might be very noisy and out of our control. For this reason, *patients* and *physicians* aren't typically mapped to the same dimension even though both are clusters of people.
4. Clusters representing physical things aren't usually mapped into the same dimensions as logical or conceptual things. For this reason, it's possible to group physical objects like *drugs*, *supplies*, or *implants*; but it's difficult to also include conceptual constructs like *lab tests* or *vital signs*.
5. Clusters that represent things that one does (typically verbs) are usually mapped into a separate dimension from things that one has or possesses (typically nouns). For this reason, processes like *lab tests*, *radiology exams*, and *surgeries* typically end up in different dimensions that physical objects like *drugs*, *implants*, or *supplies*.

6. Clusters that represent attributes that describe something of interest, like *diagnoses*, typically end up in different dimensions than things that are used or needed for something of interest, like *rooms* and *beds*.

These heuristics are always subject to interpretation, and they won't always produce the same result when practiced by different warehouse development teams; but they go a long way toward reducing the high levels of variability in design that can be seen in dimension identification done using more ad hoc analysis. An example of a dimensional design that might result from these heuristics for the clusters in Figure 6 can be seen in Figure 7.

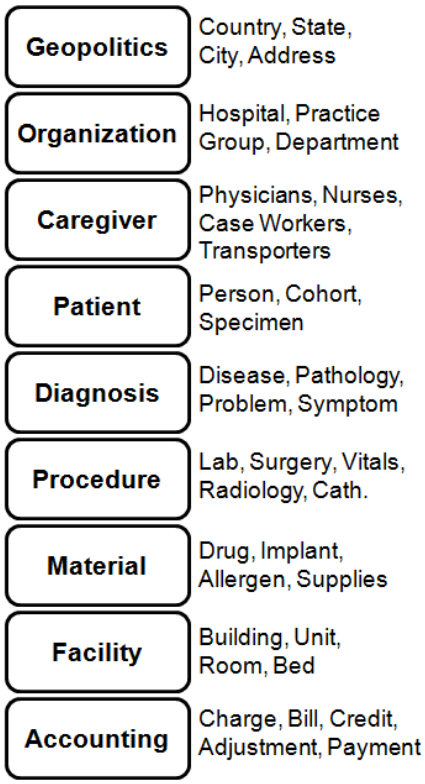

Fig. 7. Examples of physical dimensions with their associated logical subdimensions that might result from an affinity-based and heuristic analysis.

The affinity approach is the technique most used in data warehouse design today. It tends to take an information technology bias because the systems and databases available for loading into the warehouse are often the best known artifacts for identifying and describing what is intended for loading into the warehouse upon completion. Within information technology, affinity analysis is analogous to data model normalization. Identifying dimensions and subdimensions is not unlike trying to identify all of the entity types and subtypes that are needed to model a problem domain.

Many data warehouse design efforts conduct this affinity analysis while looking directly at the data files and databases that constitute their source applications, because those applications lack any form of logical data model documentation. The entity types and subtypes are inferred from the implemented artifacts themselves. The result is that the design of data warehouse dimensionality becomes based on the key structures and data dependencies designed into a few key source systems rather than on the semantics and business rules of the broader problem domain. This is adequate if the identified sources and models are of high quality, and if they truly represent the problem domain. If they don't, then the first generation data warehouse built from them will typically fail to stand the test of time. The combination of dimensions will work well for the facts associated with early warehouse use because they were derived by looking directly at the data sources that would be used to populate the earlier versions of the warehouse. But with the passage of time – usually measured in years – the need to add, change, split, or combine dimensions will place burdens on the warehouse design, the support team, the governance group, and the users who want to query data from that design. However, this approach has

dominated warehouse design because group consensus can typically be achieved in the affinity analysis steps, and an effective alternative has not been available.

The underlying problem is that warehouse dimensions are being discussed and designed only in the context of the facts expected to be stored. This is a backward approach. It is the dimensions that are meant to provide the various contexts in which the stored facts will make semantic sense. It is circular to define those contexts by looking at the facts. The relatively quick progress that is made by reverse engineering database artifacts or logical data models into lists of dimensions might seem satisfying, and will be more than adequate for the first round of data in the warehouse, but it is insufficient for designing and building a true enterprise data warehouse for the long term.

The challenge is to identify, model, and design dimensions without direct reference to the facts or to the source systems that will provide those facts. The dimensions should be designed independently, and then arriving facts can be mapped into those dimensions as they arrive. The data warehouse ingest process is less an information technology function that extracts, transforms, and loads data; and is more an epistemological process of *curation*, where as curators we are taking each data factual artifact that arrives and *annotating* it against our dimensions as best we can. Each fact is *accessioned* into our data collection according to the best dimensionality available at the time, subject to improved annotation in the future as more is learned.

Accession, curation, annotation: These tell the story of a data museum. A museum is entered into by laypeople and scholars looking to learn from the collection of artifacts that have been organized according to bodies of knowledge identified separately from the artifacts themselves. An effective body of knowledge describes what is in the collection, but can also be used to infer what is missing from the collection. To be able to identify the omissions, the dimensions of knowledge need to be defined and complete independent of the particular data artifacts available at the time of their definition.

Ontology is a branch of metaphysics concerned with the nature and relations of being. Understanding what exists and how those things relate to each other is an ontological problem, and the definition of dimensions for a data warehouse is a specific instance of such an ontological problem domain. To the ontologist, an ontology is also the *product* created while exploring a conceptual problem domain. It is a formalized list of concepts, and the relationships among those concepts, which clarifies and defines the conceptual space under study. An ontology is defined independently of the application domains in which it will be referenced and used, and so forms an effective paradigm for data warehouse dimension design. Ontologies can help define the contents of each dimension of the warehouse, and done properly, each dimension can become an *applied* ontology to be used in annotating facts in the warehouse.

4 Basic Formal Ontology

The starting point for using ontologies to define data warehouse dimensions is the Basic Formal Ontology (BFO). [2] The BFO is an upper-level ontology for the classification of real-world artifacts, and serves as an effective general model for defining ontologically-oriented warehouse dimensions. At its highest level, the BFO divides objects in the world into continuants and occurrents (see Figure 6). All instantiated dimensions in a warehouse design should be traceable back to the BFO.

Figures 8 & 9 illustrate a mapping of the clinical warehouse in Figure 5-c against the classifications in the BFO. They highlight the general sequence in which warehouse dimensions are identified and designed.

Important initial dimensions in any warehouse design involve geopolitical locations and time. The Calendar and Clock dimensions represent BFO Temporal Regions (Fig. 8-a), and the Geopolitics dimension defines BFO Spatial Regions (Fig. 8-b). At this point, Geopolitics is just the surface of the Earth and not distinct addresses yet. Individual addresses will eventually end up in this dimension, but as Sites (Independent Continuants) and not simply spatial regions.

Next in the design sequence come the bulk of the clinical dimensions (Fig. 8-c). Note that since the definition of the warehouse dimension is really a list of things that can later be involved with clinical facts, one shouldn't be surprised to see that the dimensions represent BFO Independent Continuants. The dimensions exist before and independently of the facts that will later be stored in the Fact table of the data warehouse.

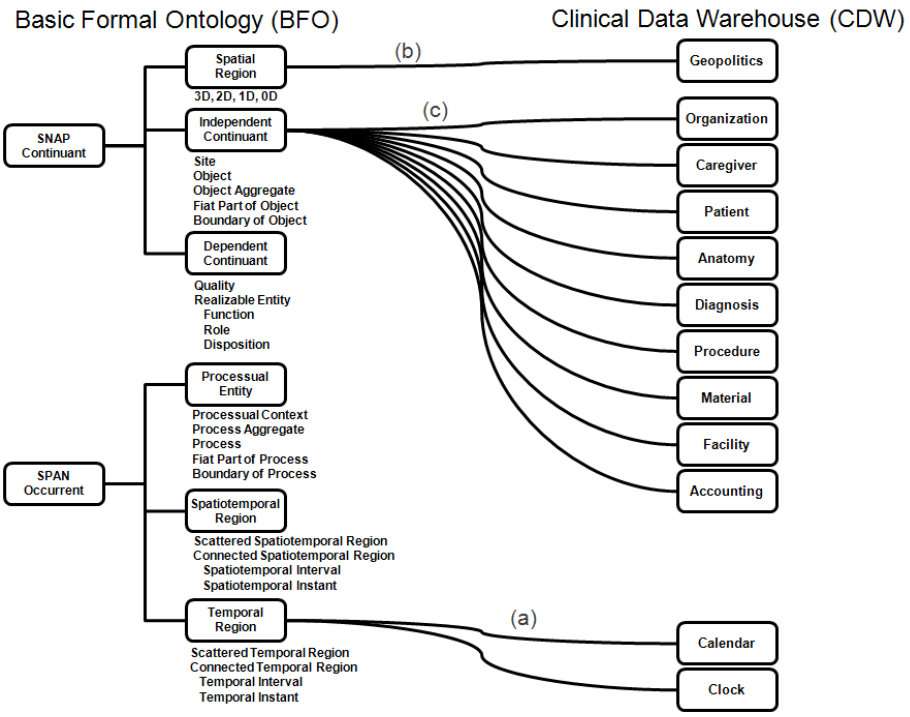

Fig. 8. Mapping of dimensions in the clinical domain against the classifications in the Basic Formal Ontology (BFO): (a) Calendar and Clock dimensions as BFO Temporal Regions, (b) Geopolitics dimension as BFO Spatial Region, and (c) all other clinical dimensions as BFO Independent Continuants.

As the design moves toward the early facts that will arrive in the warehouse (initially in specialized load processes known collectively as Master Data Management, or MDM) the design starts to include properties provided for the various dimension entries as BFO Dependent Continuants (Fig. 9-d). The Fact table also starts to define reference relationships among entries in the various dimensions (e.g., Hospital Organization is at Address Geopolitical, Physician Caregiver assigned to Clinic Facility).

In anticipation of the clinical facts loading into the warehouse, the design needs dimensions for the Encounters and other Operations (e.g., Orders, Administrations, Procedures, Vitals, Notes) occurring within the context of the already defined dimensions. Since these occur in time, they are BFO Processual Entities (Fig. 9-e).

With the process dimensions defined, the design can now store the myriad clinical facts for which the warehouse is being defined (Fig. 9-f). These facts define the integration of all of the dimensionality of the warehouse, and allow it to serve its intended purpose. This storage also continues to deepen and enrich the use of the BFO Dependent Continuant aspects of Realized Entities, as each dimension serves different roles and functions in the context of each separate fact loaded into the Fact tables.

The end-result of all of this warehousing activity – although often not instantiated in actual warehouse databases – is a BFO Spatiotemporal Region for each encounter and operation stored in the clinical warehouse (Fig. 9-g). They form a “world line” for whatever is being analyzed in the clinical data.

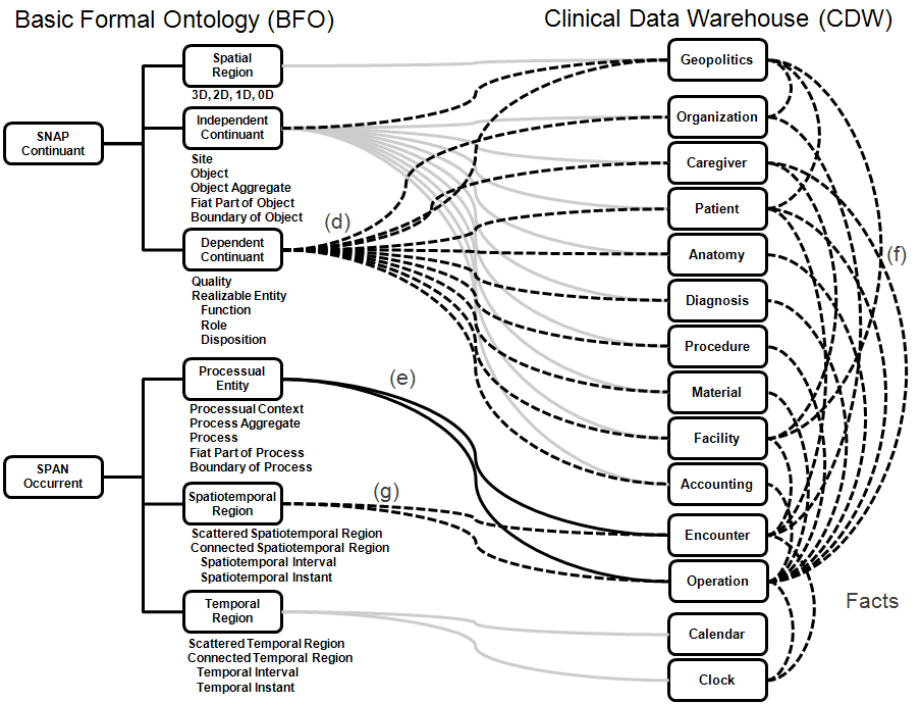

Fig. 9. Continued mapping of dimensions against the Basic Formal Ontology (BFO): (d) Properties of the clinical dimensions as BFO Dependent Continuants during Master Data Management (MDM) activities, (e) introduction of Encounter and Operation dimensions as BFO Processual Entities, (f) loading of facts using all dimensions for context, and (g) resulting interpretation of Encounters and Operations as BFO Spatiotemporal Regions (e.g. worldlines).

5 Incorporating Other Ontologies

Now that the BFO has been used as a model to understand the definition and instantiation of warehouse dimensions, the more concrete question is: How can other ontologies better inform the definition and population of those dimensions? [3] (Note: Table 1 summarizes the following examples.[3])

An example is the Foundational Model of Anatomy (FMA) as a source to instantiate the Anatomy dimension of the clinical warehouse. [4] Because there are three substantive branches in the ontology representing different elements in the BFO, each will constitute a separate subdimension of Anatomy. The dimension design pattern provides standard constructs for representing the relationships defined within the FMA through the standard hierarchy design (e.g., the FMA Attribute Entities {BFO Dependent Continuants} will be related to the FMA Anatomical Entities where relationships are defined {BFO Independent Continuants}).

As detailed facts are ingested into the clinical warehouse from the various source application systems that are providing day-to-day data, the loaded FMA ontology enriches that data. Source applications typically define their master data as lists of instances without complex semantic relationships. For example, source data might provide radiology reports that each identify the anatomical body part imaged. The source data can only be used to aggregate x-rays of left and right arms through fairly rudimentary text-based searches. With the FMA included, such aggregation is easily accomplished through the relationships defined in the FMA. Querying data for *arms*, *legs*, or even *limbs* becomes easy, even though the source systems didn't have these constructs as part of the data loaded.

Another example is the use of the Human Disease Ontology and Human Phenotype Ontology to define semantic entries in the Diagnosis dimension. Note that by using both ontologies in the same dimension, we create opportunities for the clinical Fact data to be used to identify or derive additional relationships between entries in those ontologies. Local ontological extensions can be defined that might then be useful to the broader

[3] The ontologies used in these examples are available through the Open Biomedical Ontologies (OBO) Foundry at www.obofoundry.org.

bioinformatics community. Also, existing extensions obtained from other institutions can then also be tested against local fact data.

The Environmental Ontology provides an example of a single ontology providing entries in two different warehouse dimensions: Material and Facility. It's fine for a single ontology to define multiple dimensions, as long as its individual subdimensions (e.g., Biome) don't split across multiple dimensions or cross BFO categories. Likewise, the Gene Ontology can be used to define Anatomy and Procedure dimension entries.

Table 1. Mapping of ontologies against BFO-based warehouse dimensions.

Ontology	Ontology Element	BFO Element	Dimension
Foundational Model of Anatomy (FMA)	Dimensional Entity	Spatial Region	Anatomy
	Anatomical Entity	Independent Continuant	Anatomy
	Attribute Entity	Dependent Continuant	Anatomy
Human Disease	Disease	Independent Continuant	Diagnosis
Human Phenotype	Inheritance	Independent Continuant	Diagnosis
	Organ Abnormality	Independent Continuant	Diagnosis
	Onset & Clinical Course	Dependent Continuant	Diagnosis
Environmental	Environmental Matter	Independent Continuant	Material
	Food	Independent Continuant	Material
	Biome	Independent Continuant	Facility
	Environmental Feature	Independent Continuant	Facility
Gene	Cellular Component	Independent Continuant	Anatomy
	Biological Process	Independent Continuant	Procedure
	Molecular Function	Independent Continuant	Procedure

The Dimension Anatomy Pattern (Figure 3) prescribes how a database is designed for each dimension of a star schema warehouse. The pattern is used for any warehouse, and is not specific to healthcare. Fortunately for us, the pattern has readily available prescribed positions to place the data needed to implement an ontology-based analysis and definition. Ontological relationships that occur within a single dimension of the warehouse design are implemented as acyclic directed graphs in the general hierarchy table of the design pattern. Relationships that cross between two dimensions are implemented as interdimensional facts in a Fact table, often one reserved for those ontological relationships exclusively. This capability for implementing ontological data within the existing warehouse dimension design pattern means that none of the lessons learned in the software community over the past few decades about how to build good data warehouses needs to be sacrificed in order to adopt an ontology-based design.

Ontology-based dimensional data warehouse design offers an opportunity to improve the maturity and effectiveness of a traditional data warehousing by an order of magnitude or more. Adding the semantic relationships from a variety of ontologies to the data warehouse greatly expands the queries and aggregations that can be easily accomplished. Tying data in the warehouse back to the ontologies offers real-world validation of the ontologies, and allows the data to further suggest semantic relationships not otherwise recognized or defined. The interplay of ontologies and real data creates a symbiosis that can drive improvements in both, and greatly expand the return on investment available from data warehousing.

6 Limitations

The ontology-based design algorithm explored above is not an automatic process that is assured of defining exactly the right set of data warehouse dimensions. Following this approach can produce a range of possible design alternatives, with the less appropriate or incorrect designs largely excluded. But the choices that are required by the design analyst in selecting from among the better alternatives still involves numerous judgment calls that require a good knowledge of the warehouse requirements and design experience in making such selections.

There are numerous such choices embedded in the clinical warehouse example that could have been designed differently using an ontology-based design process. For example, the design includes *Patient* and *Caregiver* dimensions, mapping each as a BFO Independent Continuant. Patients and caregivers are taken to exist independently of the data setting of the warehouse, and so can be managed as defined dimensions. Entries in these dimensions participate in various roles when data is presented to the warehouse. Roles are BFO Dependent

Continuants because they exist only as properties of more independent continuants already defined. These dependent roles are assigned to the independent continuants through their connection to data in the fact table. So, a caregiver is noted to be the *attending physician* on a hospital admission. A patient is noted to be the *organ donor* in an elective transplant procedure. This design correctly represents the heuristics presented above.

An analyst looking at this design only from an ontological position might have chosen a different solution. An ontologist might point out that both patients and caregivers are people, and that *Person* is the BFO Independent Continuant that should have been dimensionalized. The design would then include roles for *patient* and *caregiver* as BFO Dependent Continuants. This choice would be supported by the observation that people who are caregivers are also among the range of possible patients. As database designers, we would have to acknowledge that we'd prefer not to have to store information about a person redundantly in two different dimensions; so we might conclude that the design of a *Person* dimension in the warehouse would be ontologically purer.

However, no clinical warehouse designer would implement a single dimension for both patients and caregivers. This is where ontology meets the realities of software engineering, and the need to build warehouses to meet organizational requirements. Ontologies are tools, not requirements. They need to be used within the normal context of sound requirements definition and database design principles. Patients and caregivers are dimensionalized separately because we keep completely different information about each, have greater control over the data quality of caregiver vs. patient data, and see instances in these dimensions on completely different scales. We would be very reluctant to bury our hundreds of caregivers in the same dimension as our millions of patients.

There are other aspects of the clinical example above that represent design choices that have been made without the support of any specific domain ontology, but that have still been effectively informed by the use of the BFO. The Material dimension includes the definitions of all drugs, surgical implants, and medical supplies – all BFO Independent Continuants. The Procedure dimension includes the definitions of all lab tests, radiology images, vital signs, and nursing actions; again, all BFO Independent Continuants. Why are these clusters of independent continuants grouped into these two dimensions? All can be impacted by the BFO Processual Entities of being ordered, changed, and cancelled. The data for any of these will require other dimensional data for patients, caregivers, facilities, and other supporting data. Why two dimensions and not just one?

The answer lies in the nature of the *instances* of the definitions included in these dimensions. The instances of Material are all physical objects, nouns. The instances of Procedures are all executable processes, verbs. There's a difference between ordering *Acetaminophen* (a Material) for a patient and ordering the *Administration of Acetaminophen* (a Procedure) for a patient. Each actually implies the other clinically, and most clinical information systems store only one of the two.

As the ontology-based design approach yields alternatives, the designer will typically try to select options that result in a less generic data warehouse (Figure 3-c). This usually involves sacrificing some ontological purity since, pushed to the extreme, every concept can be generalized into an ontology that would reduce the dimensionality of the warehouse (Figure 3-a). A 1-dimensional warehouse might label its dimension as *Thing*, with the Fact table storing data about things. The definition of the BFO would argue that there would be a *minimum* of two dimensions in the warehouse: *Continuants* and *Occurrents*, but such a limited distinction would make little practical difference in the design of a major data warehouse containing billions of facts. The richest data warehouse designs today sacrifice some ontological purity to obtain a set of dimensions that represent the problem domain in a way that mimics how the users of the warehouse think about the domain. The challenge remains for designers to select dimensions that can also be integrated across multiple problem domains in multiple organizations.

7 Implications for Software Engineers

The shift toward an ontology-based design paradigm for data warehouse dimensions enables a fundamental shift in the architecture of data warehousing systems – away from an *ad hoc* procedural development framework and toward a standardized and reusable object-oriented framework. [5] The standardization of data enabled through the use of ontologies, along with the standardization of methods driven by the dimensional anatomy design pattern results in a working environment for software engineers that demands the application of some of the best practices that many software engineers look forward to applying but are often constrained from using by the realities of the projects on which they find themselves. [6]

Recognizing that each physical dimension of the warehouse is a polymorphic variant of the generic dimension model, the dimensional functionalities have been standardized so that they can readily be encapsulated within the system architecture, and logical dimensions rely completely on inheritance from the

physical dimension; software engineers can readily develop warehouse systems that are resilient in the face of change, and scalable in the face of growth. Change and growth are central defining tenets of data warehousing.

Data warehouse development projects that used to be measured in years can now be accomplished in months. Development resources are freed up to conduct better analysis at the front end of the project lifecycle, and the automatic side-benefits of the ontology-based development paradigms provide additional end-user value post-development that was not even envisioned just a few years ago.

8 Conclusion

Ontology-based dimensional data warehouse design is a set of heuristics that can greatly improve the quality and effectiveness of data warehouses. While meeting organizational requirements and database design principles needs to be paramount, aligning the dimensions of a data warehouse with good ontology practices can greatly enable a warehouse to meet more extensive requirements for the future, and greatly increase semantic interoperability with other data warehouses in other organizations that are working to align with the same ontologies. Actual data in the warehouse, when analyzed against known ontological alignments, will improve confidence in existing ontology relationships, and suggest new relationships not yet recognized.

For some data warehouse domains, the important ontologies might not yet have been developed, or might not be available to the designer. The BFO provides the ontological foundation to get started today, and our design process must include the flexibility necessary to incorporate future ontology development in our domains. Such incorporation will allow our data warehouses to participate in the anticipated promise of the Semantic Web. [7] [8]

References

1. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. Wiley Computer Publishing, New York (2002)
2. Grenon, P., Smith B., Goldberg, L.: Biodynamic Ontology: Applying BFO in the Biomedical Domain, in Pisanelli, D. M. (ed.), Ontologies in Medicine, Amsterdam: IOS Press, 20-38 (2004)
3. Baclawski, K., Niu, T.: Ontologies for Bioinformatics. MIT Press (2005)
4. Rosse, C., Mejino, J. L. V.: The Foundational Model of Anatomy Ontology, in Burger, A., Davidson, D., Baldock, R. (eds.), Anatomy Ontologies for Bioinformatics: Principles and Practice: Springer, 59-117 (2008)
5. Gašević, D., Djurić, D., Devedžić, V.: Model Driven Architecture and Ontology Development: Springer (2006)
6. Calero, C., Ruiz, F., Piattini, M. (eds.): Ontologies for Software Engineering and Software Technology: Springer (2006)
7. Allemag, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufman (2008)
8. Cheung, K. H., Smith, A. K., Yip, K. L. Y., Baker, C. J. O., Gerstein, M. B.: Semantic Web Approach to Database Integration in Baker C., Cheung, K. (eds.): Life Sciences in Semantic Web: Revolutionizing Knowledge Discovery, The Life Sciences, Springer, 11-30 (2007)