34

Personalized Biomedical Data Integration

Xiaoming Wang, Olufunmilayo Olopade and Ian Foster
University of Chicago,
USA

1. Introduction

Translational research is a growing field of science that seeks to discover the molecular underpinnings of diseases and treatment outcomes in any individual patient (Horig, Marincola et al. 2005). The mission has driven researchers out of isolated and discipline-oriented studies into collaborative and trans-disciplinary research efforts known as team science (Guimerà, Uzzi et al. 2005). In this new scientific arena, the ability to search for an individual's biomedical data across various domains and sources via a common computational platform is a vital component for the formulation of sophisticated hypotheses and research decisions.

Biomedical data is composed of records from both clinical practice and basic research. Each sector has distinct data governance policies and database management rules. While basic biological research data sources are open— some 1,230 curated databases are available in the public domain and accessible through the Internet (Cochrane and Galperin 2010), all primary clinical data sources are kept private with rigorous data access controls, due to *Health Insurance Portability and Accountability Act* (HIPAA) regulations (Faddick 1997). Furthermore, while basic biological research data sources frequently make data elements, database schemas, metadata information and *application programming interface* (API) available to the public, the majority of clinical data sources are hosted by proprietary commercial software. The vendors (or developers) of these tools usually disclose little information about schema and metadata to third-parties. Finally, while most basic research (e.g., biological molecule or pathway) data sources must have data integrity at the species level, translational research requires data integrity at the individual patient level. Indeed, integrated and individualized biomedical data sources will need to make a significant contribution to translational research in order to truly achieve personalized medicine. However, generating such data sources is a more difficult task than the already challenging mission of integrating basic biological research data (Stein 2008).

Data integration is the process of combining data from different sources into a unified format with consistent description and logical organization. After more than two decades of research, the topic continues to become more challenging due to increasing demands and persistent obstacles (Batini, Lenzerini et al. 1986; Bernstein and Haas 2008; Agrawal, Ailamaki et al. 2009). In this chapter, we focus on the issues that must be addressed to fulfil the demands for individualized biomedical data integration and introduce a customized warehousing approach for this particular purpose.

2. Background

2.1 Current status of biomedical data

To illustrate the demand of integrating individualized biomedical data, we start with an example: for a cancer translational researcher to assess the association between the genetic background and the occurrence of a particular cancer and its treatment outcomes, she likely needs to:

1) screen family history through medical surveys on a selected cohort; 2) read pathology report about each individual's histological diagnosis; 3) check surgical, chemo, and radiation records in the clinics; 4) follow the outcomes and adverse events of the treatments; 5) record dates and evidences of the cancer recurrence and metastasis; 6) find DNA samples from specimen bank; and 7) conduct genotyping experiments and link the genotype results back to the phonotypical records.

In order to extract meaningful information from these data, the researcher needs to have these data distinguishably aligned to individual persons, but linking these data together, even in a modest number of subjects, often fails due to data heterogeneity and discontinuity. Combining biomedical data with integrity at individual level frequently encounters four distinct challenges.

The first challenge is caused by source heterogeneity. Data elements and/or schemas for the same domain data that are designed by independent parties will normally be semantically different. Such heterogeneity may also exist in different (or the same) versions of software developed by the same party. To further complicate matters, many data sources are subject to dynamic change in all aspects, including data structures, ontology standards, and instance data coding methods. These sources customarily do not provide metadata or mapping information between datasets from previous and newer versions.

The second challenge stems from data descriptor inconsistencies. Many biomedical domains do not have established ontologies and others have more than one set of standard taxonomies. For example, one can find official taxonomies for describing cancers in SNOMED (Cote and Robboy 1980), *International Classification of Disease* (ICD) (Cimino 1996), and the NCI-thesaurus (Sioutos, de Coronado et al. 2007).

The third challenge comes from data source management styles. Most data sources are isolated and autonomously operated. These sources typically neither map nor retain the primary identifiers (of a person or the specimens that originated from the person) created in the other sources. The silo settings of the data sources not only generate segregated datasets but often require repetitive re-entry of the same records (e.g., patient demographic data) by hand into different sources. This practice increases the risk of human error.

The fourth challenge is due to low data source interoperability. The majority of clinical data sources are neither programmatically accessible (syntactic interoperability) nor have metadata available for the source data (semantic interoperability).

Many of these problems have been continual to date and will linger for the foreseeable future. As a consequence, biomedical source data are typically heterogeneous, inconsistent, fragmented, dirty and difficult to process. Valuable information embedded within the data cannot be consumed until the data are cleansed, unified, standardized, and integrated.

2.2 Related data integration regimes

The purpose of data integration is to deliver integrated information. This purpose can be realized via either permanent (physical) or transient (view) data integration. Among the

various information integration approaches, data warehousing, view integration, and information mashup are popular regimes that are actively discussed in IT and informatics publications (Halevy 2001; Jhingran 2006; Goble and Stevens 2008). Each regime has its own distinct design principle and system architecture.

2.2.1 Data warehousing

While describing the architecture of a data warehousing solution, many focus on a multidimensional database (Louie, Mork et al. 2007; Goble and Stevens 2008). This database-centred approach is widely used across multiple fields and has a well-documented history in regards to its evolution. A warehouse delivers integrated information by organizing and storing data from various sources within a physical schema so that the integrated data can be reused for a variety of applications. Since the basic requirement for a database to function is data availability, the warehousing approach appears to be more tolerant to various data source conditions than its counterpart solutions which all require data sources to be interoperable and accessible. Warehousing is generally considered most suitable for historical data accumulation, quality data integration, and post-integration data curation and annotation (Halevy, Ashish et al. 2005). In biomedical informatics, the warehousing approach is considered most suitable for personalized biomedical data integration (Louie, Mork et al. 2007; Wang, Liu et al. 2009).

The major drawbacks of the warehousing approach are its association with stale data and the resource-consuming nature of system maintenance. These negative aspects reveal a simple truth about warehousing: data supply issues are the biggest obstacle and financial drain to the ultimate success of the strategy. These issues demonstrate the need for a data *extraction-transformation-loading* (ETL) process that determines the quality and freshness of integrated data. Thus, we discuss in greater depth of both the database and the ETL process.

2.2.1.1 Multidimensional database modelling

The database in a warehousing strategy must store multidimensional data. Three distinct conceptual modelling methods are often employed for database design: 1) the Entity-Relationship (ER) model (Chen 1976; Kamble 2008); 2) Entity-Attribute-Value representation with classes and relationships (EAV/CR) model (Dinu and Nadkarni 2007; Lowe, Ferris et al. 2009); and 3) Object-Oriented Database (OODB) (Trujillo, Palomar et al. 2001). Although no consensus has been reached concerning modelling standards, we prefer the ER model due to its solid mathematic foundation, data structure semantic clarity, and data presentation transparency. These features not only benefit high-throughput data deployment at the database layer for integration but also support satisfactory query performance at the user-application interface. In addition, since the ER model requires semantic clarity and consistency for each attribute element during cross-source data processing, the delivery of useful data and consumable information is better ensured.

At the implementation level, a data warehouse is suggested to have a star schema (a single large fact table and many smaller dimension tables) to improve application performance by reducing the time required to join multiple tables to answer a query (Kimball, Reeves et al. 1998). However, a highly de-normalized schema, like a star schema, also increases cost of database maintenance. With materialized-view technology being well-developed, the de-normalization of data entities at the primary schema level is no longer critical. Materialized-view technology is capable of flexibly aggregating data into any combination.

2.2.1.2 ETL modelling and tools

A warehouse relies on an ETL process to refill and refresh data from heterogeneous sources into a predefined schema for data integration. The history of developing and optimizing ETL is directly tied to the history of data warehouse systems. Yet, a significant technological breakthrough for ETL remains to be seen. In general, the process remains error-prone and labour-intensive (Rizzi, Abello et al. 2006). The methods related to the different stages of an ETL process can be briefly categorized into the following: 1) schema mapping (Bernstein and Rahm 2001); 2) metadata collection and management (Kolaitis 2005); 3) error detection and data cleansing (Kalashnikov and Mehrotra 2006); and 4) systematic modelling of the entire ETL process (Vassiliadis, Simitsis et al. 2002).

The pressing demand for data integration has spurred the development of commercial ETL software packages. Examples include DataStage (Oracle), Informix (IBM), and products from smaller vendors. To date, all ETL strategies still involve human intervention, albeit at different levels. Many approaches appear to be exhaustive and obsessive in the hierarchical levels of schema mapping while paying insufficient attention to data value integration. Some are inherently brittle, often without a clear measure of success for an ETL workflow in the real world. In the field of biomedical informatics, which fiercely promotes interoperable data integration (Komatsoulis, Warzel et al. 2008), ETL remains an autonomous in-house activity and formal reports on the topic are seldom seen.

2.2.2 View integration

The architecture behind view integration consists of an interoperable data grid with constituent primary databases that are autonomous, disparate, and heterogeneous. Yet, data from these databases need to be aligned to a common virtual schema during integration (Halevy 2001; Stein 2002). In this regime, queries issued by users are posed to the virtual schema and then translated into various customized queries against disparate data sources. Extracted source data is then transformed on the fly to meet the definitions of the common schema. Finally, unified data is presented to users as integrated information.

Because view integration combines data dynamically, without any persistent physical data aggregation, it can provide real time information. However, its effectiveness depends critically on the interoperability and network accessibility of every data source of interest, which (at least at the current time) seems unlikely to occur in practice. In addition, because view integration requires the entire data aggregation process from disparate sources to the view to be fully automated, it provides less control of data quality than data warehousing approaches in which manual curation steps can be included in the ETL process. This lack of sufficient data quality control is a serious concern for biomedical informatics researchers (Goble, Stevens et al. 2008; Galperin and Cochrane 2009).

To date, view integration systems with production level maturity are not commonly seen in the real world. However, the design concepts utilized in view integration, including global-as-view (GAV) and local-as-view (LAV), are valuable principles to all data integration efforts (Lenzerini 2002). In GAV, the virtual schema is made adjustable to accommodate all source data elements. In LAV, each source data is individually managed to be transformed to align to this global view.

2.2.3 Information mashup

The architecture behind the information mashup approach comprises of an open fabric which augments information from different sources that allow their API accessible through the

Internet. Although having some similarities to ETL, mashup is advantageous both in its ability to be light-weight and its utilization of a service model that is easily extensible via Web 2.0 techniques, e.g., *representational state transfer* (REST) technology (Fielding and Taylor 2002). The prerequisite for a mashup fabric to consistently present (overlay) different messages on a web interface is standardized information formats in the atomic data elements that carry instance data. These data feeds are either staged within the sources (or at a middleware) or converted to have consistent data interpretation, e.g., geographic code, and formats on the fly. For example, a (subject-predicate-object) triplet expressed in the W3C-standard *resource description framework* (RDF) carries metadata information for related atomic data elements. Therefore, RDF can be easily converted into an ER model and vice versa.

Unlike view integration, which has a predefined schema, a mashup fabric uses structured basic data elements to form information presentations that are not rigorously structured, e.g., information situational display overlay (Jhingran 2006). Therefore, a mashup strategy becomes more implementable and as a result, has been used in many information integration-*lite* tasks (Franklin, Halevy et al. 2005; Goble and Stevens 2008). Examples include Google maps (Wong and Hong 2007) and Avian Flu maps (Butler 2006).

However, the openness of the mashup fabric along with its integration *lite* nature limits its application as the first layer technology option for personalized biomedical data integration. At this layer, the integration framework must manipulate primary source data in a secluded network because the data contain *protected health information* (PHI). Furthermore, data integration with individualized integrity from longitudinal records is an integration-heavy task, which is better handled by a relational database in a secure environment alleviating HIPAA concerns. However, the mashup strategy may still be suitable for higher-level (anonymous) biomedical data integration for population studies that require evidence from a significant number of individuals in a particular domain or against a geographic background.

2.3 Related human-intensive efforts

In addition to computationally heavy informatics solutions, human-developed information research is also a crucial component in overall information delivery (Bernstam, Smith et al. 2010). These efforts include domain ontology creation (Yu 2006) and mapping for multiple standards in the same domain, e.g., *unified medical language system* (UMLS) (Bodenreider 2004) for SNOMED and ICD. However, despite the challenge of mapping between existing standards, newer standards keep emerging. Mapping concepts between different ontologies in the same domain or different versions within the same ontology requires good mapping references, which are usually either inadequate or not computable, e.g., cancer registry coding schema (Edge, Byrd et al. 2010). Nevertheless, discussions concerning how to best use human effort in the field of informatics are rare. It is evident that these efforts can either benefit or hinder data integration efficiency.

3. Methods

3.1 Conceptual design of the warehousing framework

Based on our analysis of existing data integration regimes, biomedical data statuses, and conditions of data sources in the real world, we adopted a warehousing strategy to address the general need for personalized data integration. The purpose of our approach is to provide a computation framework aimed at data *unification for integration* (UNIFIN). By

design, the UNIFIN framework contains two essential technical components: a multidimensional database, the *Translational Data Mart* (TraM), and a customized ETL process, the *Data Translation Workflow* (TraW). UNIFIN also includes an interoperable information integration mechanism, empowered by REST and SOA technologies for data that do not need manipulation during the integration process. Examples of such data include homogeneous molecular sequence records and binary image datasets.

In this chapter, we focus on describing the semi-automated ETL workflow for heterogeneous text data processing. We explain data processing, integration and ontology issues collectively, with the intent of drawing a systematic picture describing a warehousing regime that is customized for personalized data integration. Although TraM and TraW have been developed by the same team, each is designed to be independently portable, meaning that TraM can readily accept data from other ETL tools and TraW, after customization, can supply data to target databases other than TraM.

3.2 Data warehouse TraM

3.2.1 Domain ontology integrated ER conceptual model

We used the top-down design method to model the TraM database. The modelling is based on our analysis of real world biomedical data. The results of the analysis are depicted in a four-dimensional data space (Fig 1A). The first dimension comprises study objects, which can range from human individuals to biomaterials derived from a human body. Restrictively mapping these objects according to their origins (unique persons) is essential to assure personalized data continuity across domains and sources.

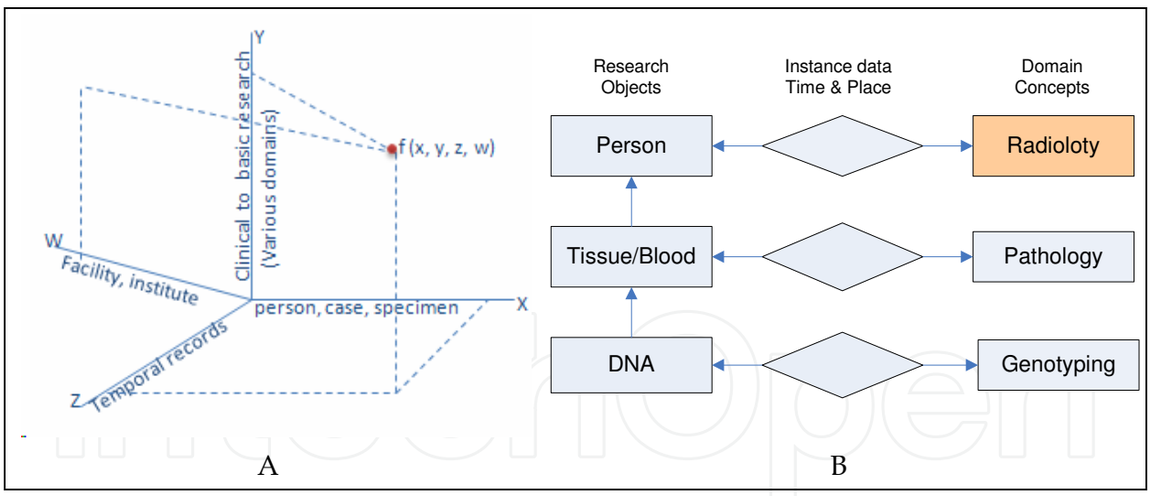


Fig. 1. Data warehouse modelling: A) Biomedical data anatomy; B) DO-ER conceptual model for the TraM database. The square shapes in panel B indicate data entities and the diamond shapes indicate relationships between the entities. The orange colour indicates that the entity can be either a regular dictionary lookup table or a leaf class entity in a domain ontology.

The second dimension is positioned with a variety of domain concepts that are present in an extensive biomedical practice and research field. Domain concepts can either be classified in domain ontologies or remain unclassified until classification takes place during integration. There are no direct logical connections between the different domain ontologies in an

evidence-based data source. The associations among domain concepts will be established when they are applied to study objects that are derived from the same individual.

The other two dimensions concern the times and geographic locations at which research or clinical actions take place, respectively. These data points are consistent variables associated with all instance (fact) data and are useful to track instances longitudinally and geographically. This analysis is the foundation for TraM database modelling.

We then create a unique *domain-ontology integrated entity-relationship* (DO-ER) model to interpret the data anatomy delineated in Fig 1A with the following consistency. (See Fig 1B for a highly simplified DO-ER model.)

1. The instance data generated between DO and study objects are arranged in a many-to-many relationship.
2. Domain concept descriptors are treated as data rather than data elements. Descriptors are either adopted from well-established public ontology data sources or created by domain experts with an ontology data structure provided by TraM, assuming that there is no reputable ontology existing within that domain. Thus, domain concepts are either organized as simply as a single entity for well-defined taxonomies or as complex as a set of entities for classification of concepts in a particular domain.
3. Study objects (a person or biomaterials derived from the person) are forced to be linked to each other according to their origins regardless of the primary data sources from which they come.

There are three fuzzy areas that need to be clarified for the DO-ER model. The first is the difference between an integrated DO in the DO-ER model versus a free-standing DO for sophisticated DO development. The DO development in the DO-ER model needs to be balanced between its integrity in concept classification (meaning the same concept should *not* be described by different words and vice versa) and its historical association with the instance data (meaning some DO terms might have been used to describe instance data). The data values between the DO and ER need to be maintained regularly to align the instance data descriptors to the improved DO terms. For this purpose, we suggest that concept classification be rigorously normalized, meaning to make concept of an attribute not divisible in the leaf class of the DO, because merging data with unified descriptor is always easier than splitting data into two or more different data fields. The advantage of the DO-ER model is that these changes and alignments usually do not affect the database schema. The latter remains stable so there is no need to modify application programs.

The second is the conceptual design underlying the DO structures in the DO-ER model. In fact, the DO under this context is also modelled by ER technique, which is conceptually distinct from the popularly adopted EAV modelling technique in the biomedical informatics field (Lowe, Ferris et al. 2009) . The major difference is that the ER underlying the DO has normalized semantics for each attribute, while the EAV does not.

The third is determining the appropriate extent of DO development that should go into an evidence-based database. We believe that TraM users in general have neither the intention nor the resources to make TraM a DO producer. Thus, our purpose in allowing for DO development within TraM is solely to satisfy the minimal requirement of harmonizing highly heterogeneous data with controlled vocabulary, so that the DO is developed as needed. The resulting standardization of data concept abstractions, classifications, and descriptions will make it easier to merge data with future reputable DO standards as they (hopefully) emerge. Under this context, we further explain a use case underlined with DO-ER transformation-integration mechanism in section 4.2, and detail how an evolving DO may affect data integration workflow in section 3.3.4.

3.2.2 Enforcing personalized data integrity

Personalized data integrity is enforced throughout the entire TraM schema. To achieve this level of quality control, the first required condition is to identify uniqueness of a person in the system. HIPAA regulations categorize a person's demographic information and medical administrative identifiers and dates as PHI that should not be disclosed to researchers or transferred between databases without rigorous legal protections. However, as a person is mobile, an individual's medical records are often entered into multiple databases in more than one medical institution or clinic. Without PHI, it is almost impossible to reliably identify the uniqueness of a person unless 1) the person's identifiers are mapped across all data sources or 2) there is a universal identifier used in all healthcare and research domains. Neither condition currently exists. Therefore, a data warehouse often must be HIPAA-compliant to contain PHI data to verify the uniqueness of a person. This is the case in the TraM operation. Once the uniqueness of a person is identified, TraM has a built-in mechanism that automatically unlinks the PHI records to form the materialized view. Since the materialized view is the schema that answers queries, the application program can only access de-identified data, and therefore, regular users do not see PHI but can still receive reliable individualized information.

3.3 ETL process TraW

The TraM data model reflects one particular interpretation (our interpretation) of biomedical data in the real world. Independent parties always have different opinions about how a warehouse database should be constructed. Different data sources also interpret the same domain data differently both among themselves and from a warehouse. To bridge these gaps, TraW is designed to be configurable to adapt to different sources and targets. Since most medical data sources do not disclose database schema or support interoperability, we have focused in designing TraW on gathering the basic data elements that carry data and performing data extraction from available electronic data forms (Free text is not included in this discussion.). Unlike TraM, which has a relatively free-standing architecture, TraW is an open fabric with four essential highly configurable components:

1. A mechanism to collect metadata—routinely *not* available in source data deliveries. A web-based data element registration interface is required to collect metadata across all sources.
2. A set of systematically designed and relatively stable domain data templates, which serve as a data processing workbench to replace numerous intermediate tables and views that are usually autonomously created by individual engineers in an uncontrolled ETL process.
3. A set of tools that manipulate data structures and descriptions to transform heterogeneous data into an acceptable level of uniformity and consistency as required by the target schema.
4. A set of dynamically evolving domain ontologies and data mapping references which are needed for data structure unification and descriptor standardization.

Behind these components is a relational database schema that supports TraW and records its data processing history.

3.3.1 Metadata collection

TraW treats all data sources as new by collecting their most up-to-date metadata in each batch data collection through a web-based application interface. If the existing sources do

not have any changes since the previous update, the source data managers are required to sign an online confirmation sheet for the current submission. To avoid another level of heterogeneity as generated by metadata description, TraW provides a pre-defined metadata list for data providers to choose from through the registration interface. These metadata are defined based on the TraW domain templates (the differences between domain templates and target schema are detailed in section 3.3.2). These metadata will not completely cover all source data elements, not necessarily because they do not represent the meanings of those data, but because they do not share the same semantic interpretations for the same kinds of data. Thus, TraW allows data providers to create their own metadata for unmatched data fields.

3.3.2 Domain data template

In section 2.2.2, we described the GAV and LAV concepts. The TraW domain template is derived from the GAV concept. The difference is that the GAV in *view* integration is a virtual schema that responds directly to query commands, while the domain template in TraW both carries physical data and serves as a workbench to stage data before integration. Unlike a target schema, which has normalized domain entities and relationships governed by rigorous rules to assure data integrity, domain templates in TraW do not have entity level data structures, nor do they have relationship and redundancy constraints. Instead, since there are no concerns about the user application interface, the templates can simply be frequently edited in order to accommodate the new source data elements. However, these templates must have three essential categories of data elements.

First, a template must contain elements that support *minimal information about data integration* (MIADI). MIADI is presented by a set of primary identifiers from different sources and is required for cross-domain data integration. These identifiers, which come from independent sources, should be capable of being mapped to each other when study objects are derived from the same person. If the mapping linkage is broken, PHI will be required to rebuild data continuity and integrity for one person may have multiple identifiers if served in different medical facilities.

Second, a template must contain the domain *common data element* (CDE), a set of abstracted data concepts that can represent various disciplinary data within a domain. For example, cancer staging data elements are required for all types of cancers so they are the CDE for evidence oncology data. Elements for time stamps and geographic locations are also CDEs for cross-domain incidence data. Domain CDEs are usually defined through numerous discussions between informaticians and domain experts if there is no available CDE that is widely accepted in the public domain.

Third, the template must contain elements that carry data source information, e.g., source database names, owner names of the databases, data versions, submission times, and etc, which are collectively called data provenance information. This information is required for data curation and tracking.

ETL workers continue to debate what exactly constitutes domain CDE, despite significant efforts to seek consensus within or across biomedical domains (Cimino, Hayamizu et al. 2009). Each ETL often has its own distinct semantic interpretation of data. Therefore, TraW should only provide templates with the three specified element categories in order to give ETL workers flexibility in configuring their own workbench.

Generally speaking, domain templates have looser control on CDE formulation than do target schemas because they are intended to cover all source data elements, including those that have

a semantic disparity on the same domain data. For this reason, a domain template actually serves as a data transformation medium which in principle, has a consistent data structure as the target schema while simultaneously containing both an original (O-form) and a standardized (S-form) form for each data element. Data need to be in a semantically consistent structure before they can be standardized. Data in S-form are completely consistent in both structure and description to the target schema. Reaching semantic consistency relies on a set of data transformation algorithms and semantic mapping references.

3.3.3 Data transformation algorithms

Data transformation is a materialized process of LAV (refer to section 2.2.2), meaning that it converts data from each source to a common data schema with consistent semantic interpretations. Since we focus mainly on basic data element transformation, we mashup these elements from different sources and rearrange them into different domain templates. Because each domain template contains a data provenance element, we can trace every single record (per row) by its provenance tag through the entire data manipulation process. The transformation of a data element proceeds in two steps: data structure unification and then data value standardization. The algorithms behind these two steps are generic to all domain data but depend on two kinds of references to perform accurately. These references are the domain ontologies and the mapping media between sources and target schema (more details in 3.3.4). In this section, we mainly explain data transformation algorithms.

In our experience with integrating data from more than 100 sources, we have found that about 50% of source data elements could not find semantic matches among themselves or to the target schema even when they carried the same domain data. Within these semantically unmatched data elements, we consistently found that more than 80% of the elements are generated through hard-coding computation, meaning that data instances are treated as variable carriers, or in other words, as attribute or column names (Fig 2.I). This practice results in data elements with extremely limited information representation and produces an enormous amount of semantically heterogeneous data elements. It is impossible to standardize the data description in such settings unless instance values are released from the name domains of the data elements. The algorithm to transform this kind of data structure is quite straightforward, unambiguous, and powerful. The process is denoted in the formula:

$$f\{x_{(1)}, y_{(i)}\} \Rightarrow f\{x_{(i)}, y_{(\text{specified})}\}$$

In this formula, a data table is treated as a two dimensional data matrix. x represents rows and y represents columns. Column names (left side of arrow) are treated as data values in the first row. They are transposed (repositioned) into rows under properly abstracted variable holders (columns). The associated data with those column names are rearranged accordingly (Fig 2.II). The decision as to which column names are transposed into which rows and under which columns is made with information provided by a set of mapping references. Since this process often transforms data from a wide form data matrix into a long form, we refer to it as a wide-to-long transformation. The fundamental difference between this long form data table versus an EAV long form data table is that the data table in our system is composed with semantically normalized data elements while EAV data table is not.

Once data values are released from data structures under properly abstracted data elements, normalization and standardization of the value expression can take place. This process is

called data descriptor translation and relies on a mapping reference that specifies which specific irregular expressions or piece of vocabulary are to be replaced by standardized ones. At the same time, further annotation to the instance data can be performed based on the metadata. For example, the test values in Fig 2.II are measured from two different test methods for the same testing purpose. In this circumstance, unless the test methods are also annotated, the testing results cannot be normalized for an apple-to-apple comparison. In addition, the assessment (asmt) field is added to justify the score values read from different testing methods (Fig 2.III).

There are other complex data structure issues besides hard-coded data elements requiring significant cognitional analysis to organize data. We discuss these issues in section 5.

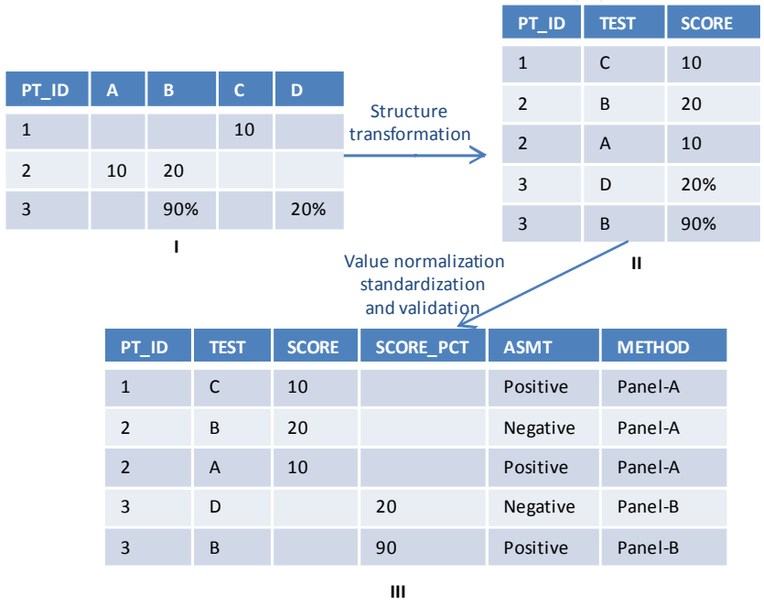


Fig. 2. An example of data transformation: I) Hard-coded data elements; II) Semantically unified data structure; and III) Standardized and normalized data values with additional annotation based on metadata and data mapping references

3.3.4 Domain ontology and mapping references

Domain ontology and mapping references are human-intensive products that need to be designed to be computable and reusable for future data processing. In general, it is simpler to produce mapping references between irregular data descriptors and a well-established domain ontology. The problem is how to align heterogeneous source data elements and their data value descriptors to a domain ontology that is also under constant development. Our solution is to set several rules for domain ontology developers and provide a backbone structure to organize the domain ontology.

1. We outline the hierarchy of a domain ontology structure with root, branch, category and leaf classes, and allow category classes to be further divided into sub-categories.
2. We pre-define attributes for the leaf class, so that the leaf class property will be organized into a set of common data elements for this particular ontology.
3. Although domain concept descriptors are treated as data values in ontology, they should be in unique expressions as each should represent a unique concept.

We train domain experts with these rules before they develop ontologies, as improper classification is difficult to detect automatically. We maintain data mapping references in a key-value table, with the standardized taxonomy as the key and irregular expressions as values. Both in-house domain ontologies and mapping references should be improved, validated, maintained, and reused over time.

3.3.5 Data transformation process

Here, we describe a snapshot of the data transformation process. Typically, this process requires a set of leaf class attributes for a domain ontology, a mapping table that connects the leaf class data elements and the source data elements, a data structure transformation program, and a set of source data (Fig 3). At the end of this process, the source data structures and values are all transformed according to the concept descriptor classification in the domain ontology. The original source data attribute name is now released from the name domain (red boxes in Fig 3) of a data element and becomes a piece of value record (purple box in Fig 3) that is consistent to the taxonomy in the domain ontology.

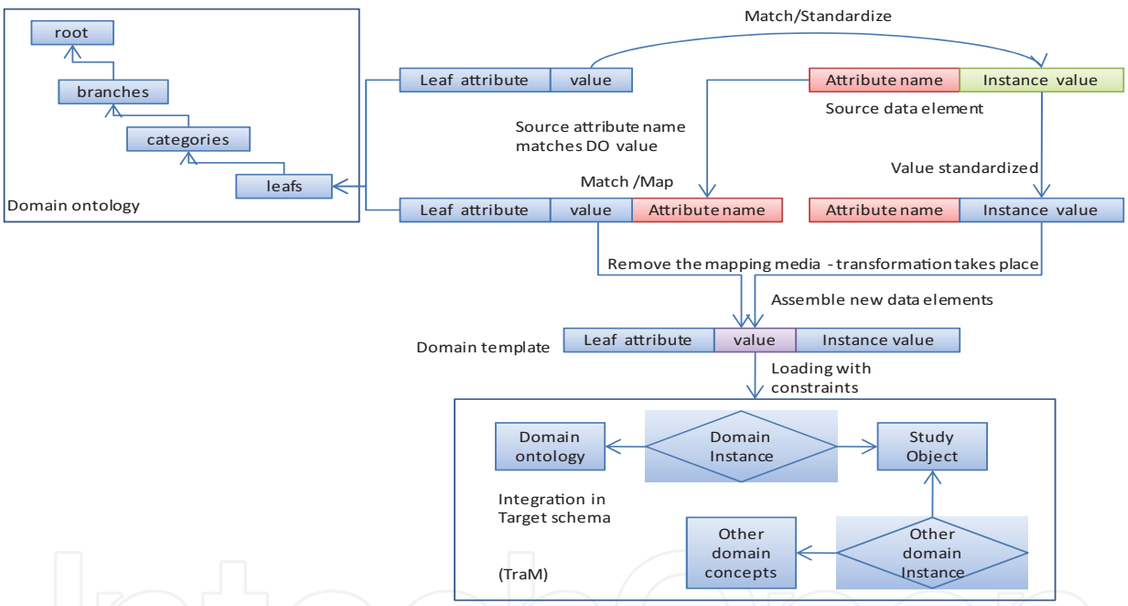


Fig. 3. A generalized data transformation processs using dynamically evolved domain ontology and mapping media.

4. Results

4.1 UNIFIN framework overview

UNIFIN is implemented through the TraM and TraW projects. TraM runs on an Oracle database server and a TomCat web application server. TraW also runs on an Oracle database, but is operated in a secluded intranet because it processes patient PHI records (Fig 4). TraM and TraW do not have software component dependencies, but are functionally interdependent in order to carry out the mission of personalized biomedical data integration. Fig 4 shows the UNIFIN architecture with notations about its basic technical components.

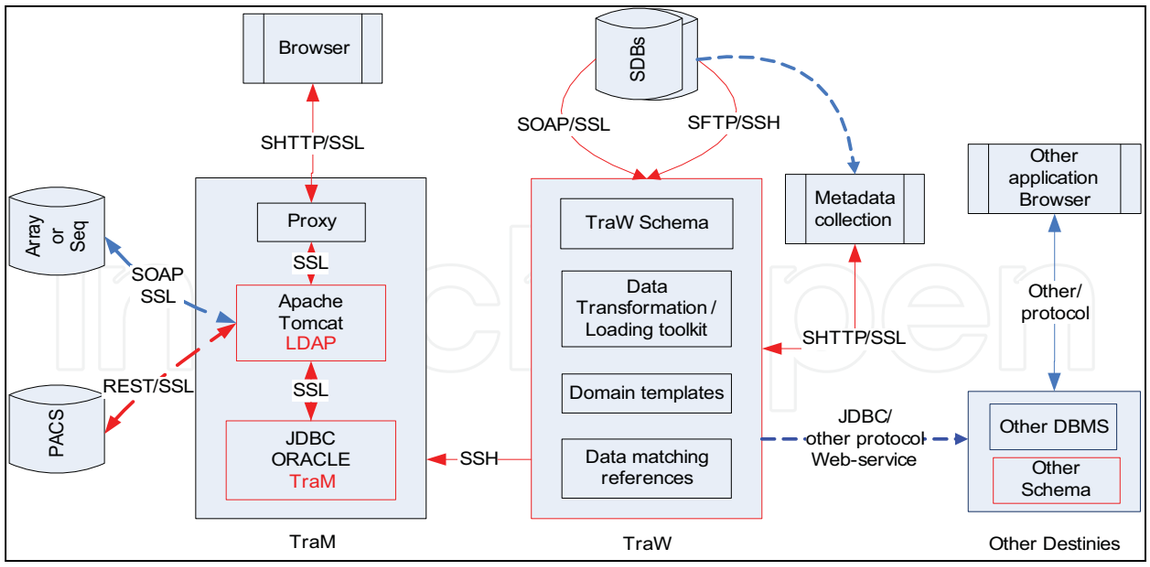


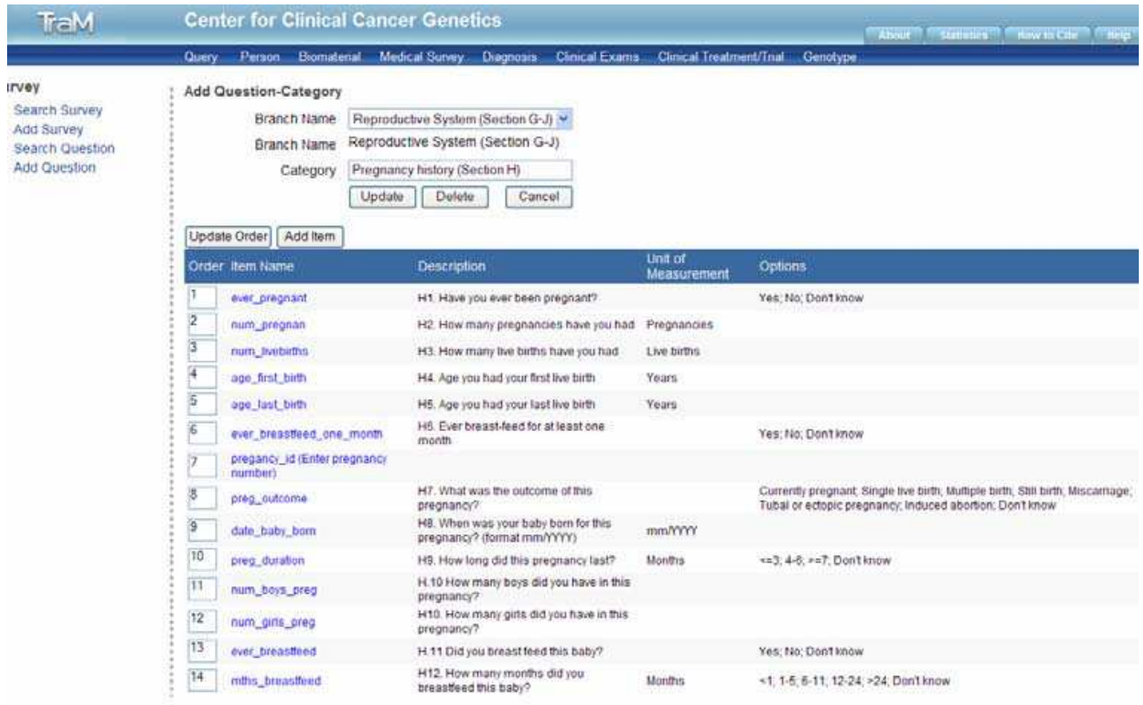
Fig. 4. UNIFIN overview: Dashed lines for the panel on the left indicate the workflow that does not route through TraW but is still a part of UINFIN. Red lines indicate secured file transmission protocols. Areas within the red boxes indicate HIPAA compliant computational environments. The right side panel of TraW indicates other data integration destinations other than TraM.

Whereas the web application interface of TraM provides user friendly data account management, curation, query and retrieving functions for biomedical researchers, TraW is meant for informaticians, who are assumed to have both domain knowledge and computation skills.

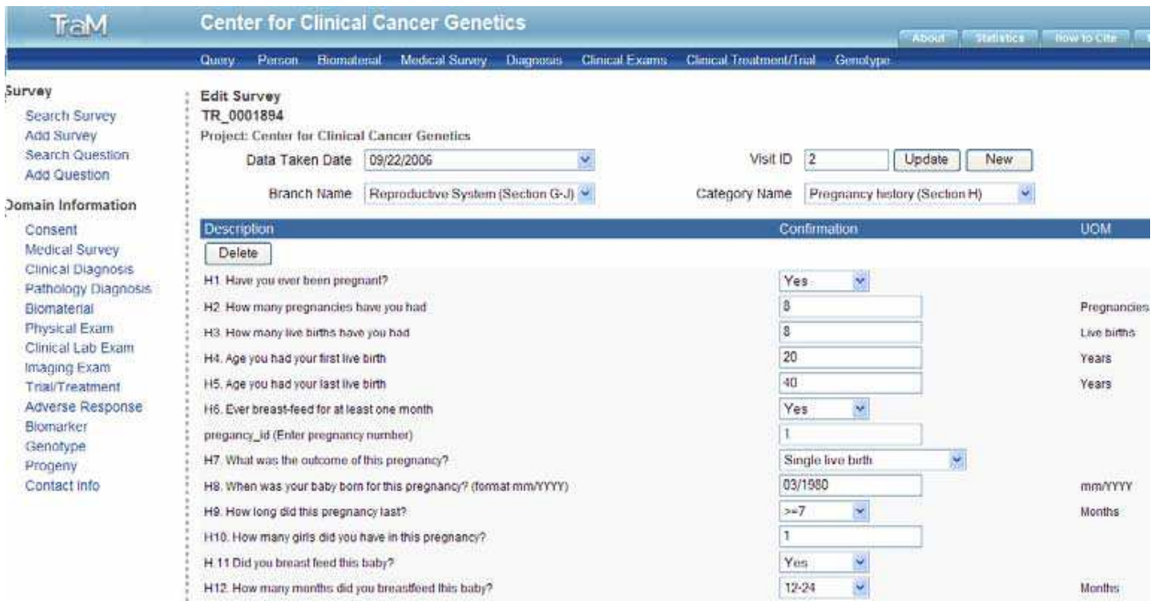
4.2 A use case of TraM data integration

We use the example of medical survey data, one of the least standardized and structured datasets in biomedical studies, to illustrate how domain ontology can play an important role in TraM. It is not uncommon to see the same survey concept (i.e., question) worded differently in several questionnaires and to have the data value (i.e., answer) to the same question expressed in a variety of ways. The number of survey questions per survey subject varies from fewer than ten to hundreds. Survey subject matter changes as research interest shifts and no one can really be certain as to whether a new question will emerge and what the question will look like. Therefore, although medical survey data is commonly required for a translational research plan (refer to the example in 2.1), there is little data integration support for this kind of data and some suggest that survey data does not belong in a clinical conceptual data model (Brazhnik and Jones 2007).

To solve this problem, we proposed an ontology structure to manage the concepts in the questionnaires. Within the DO, the questions are treated as data in the leaf class of the questionnaire and organized under different categories and branches. Each question, such as what, when, how, and why, has a set of properties that define an answer. These properties include data type (number or text), unit of measure (cup/day, pack/day, ug/ml), and predefined answer options (Fig 5A).



A



B

Fig. 5. Domain ontology and data integration: A) Leaf class attributes of the questionnaire; B) Instance data described by domain ontology. Both A and B are screenshots from the TraM *curator interface* (explained in 4.4). The left hand panel of screen B shows hyperlinks to the other domains that can also be associated with the individual (TR_0001894). The data shown is courtesy from the account of centre for clinical cancer genetics

Since a new question and its properties are treated as a new record (a new row in the leaf class table), the overall database structure stays the same. Since the possible answers in the survey are pre-defined with controlled vocabulary, the answers will be recorded in a relationship between the person entity and the question item entity. The survey results are also instantly and seamlessly integrated with the other domain data (Fig 5B). This underlying mechanism is meant to allow for the new question to play a double role: first, as a record (value) in the questionnaire ontology and second, as a question concept to recruit the survey result. In this way, the TraM system gains enormous flexibility in recruiting new concepts and is able to annotate data with controlled vocabularies.

4.3 A life-cycle of TraW process

Running a life-cycle of TraW takes four steps: 1) extract atomic data elements from various source data files; 2) unify (transform) data structure at the atomic data element level and mashup data into proper intermediate domain templates; 3) standardize (translate) and validate data values upon mixed data elements on the domain templates; 4) load and restructure data from domain templates into a target schema and complete integration.

We have yet to gain much experience in extracting data directly from sources since most of the medical data sources that we work with deny programmatic data access. The source data files we obtained are in table or XML formats delivered through a secured file transport process (SFTP).

The person(s) who operates TraW is also responsible for customizing and maintaining the essential constituent components. This responsibility includes editing domain templates, maintaining data mapping references and modifying programs in the toolkit. As a high-throughput computation process, TraW may drop a list of disqualified data at each step and keep moving forward. Disqualified data will be sent to a curator for further validation. Recovered data may rejoin the workflow after verification to be processed with other data (Fig 6). Unified and standardized data on the domain templates (in the S-forms, refer section 3.3.2) are in a mashup status but not integrated. Integration is realized through the loading procedure which resumes ER structures and constraints in the destination. Since domain templates provide a standardized and stabilized data inventory, loading procedures can be highly automated between domain templates and the target schema.

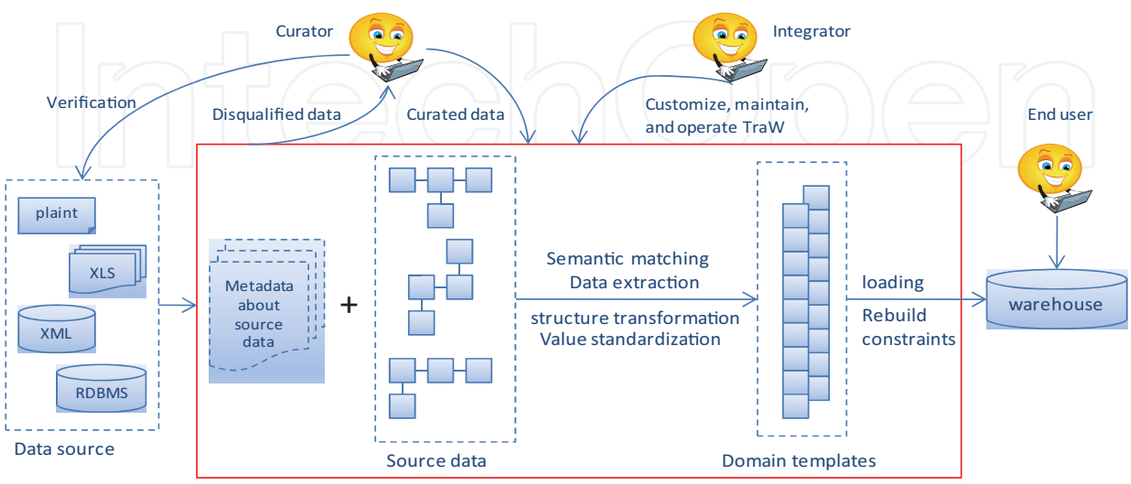


Fig. 6. Semi-automated TraW process

4.4 Information delivery

Evaluation of the success of the UNIFIN approach is assessed by its final products: personalized biomedical data and the applications that deliver these data. The quality of TraM data reflects the integration capacity of the TraM schema and efficiency of the TraW process and is measured by data uniformity, cleanness and integrity. A researcher who demands individualized cross-domain data (as described in the example in 2.1) should be able to query through the TraM application interface to obtain satisfactory information. Specifically, quality data should have the following features: 1) the same domain data should have consistent data elements (e.g., domain CDEs) and data descriptors; 2) all specimens, which are the linkage between clinical and basic science records, should be bar-coded and annotated with at minimum, the person's demographic information; 3) various domain data derived from the same person should be interlinked regardless of their disparate sources; 4) redundancies and discrepancies in the data are rigorously controlled at all hierarchical levels of schema and domains. Fig 7 displays some of these features of the TraM data in a breast cancer translational research account.

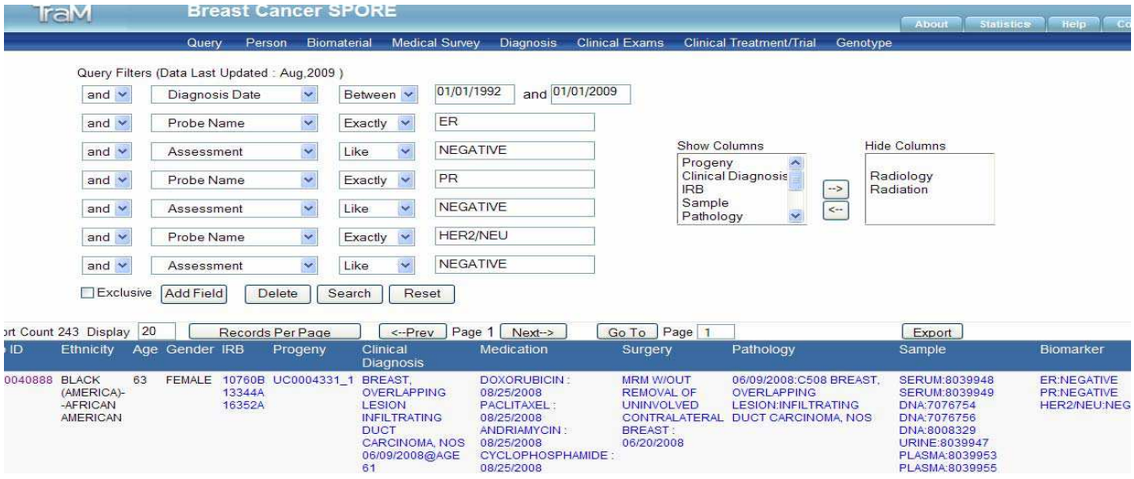


Fig. 7. An example of the TraM data (screenshot on the regular user interface): In this particular query, the user used filters for three negative breast cancer biomarkers (noted as probe names), while other domain data are also displayed on the screen, including specimen samples. Each data field provides hyperlinks to allow for navigation to more detailed domain records. Important time stamps, e.g., the dates of diagnosis, surgery, and medication, are also shown. The export function allows users to retrieve normalized data in any domain, including detailed domain data that are not shown on this screen. The data shown is courtesy of a breast cancer translational research account.

On average, when data are collected from more than 10 independent sources (batch data processing usually contains data from 10-30 disparate sources), around 50% of the distinct individual counts from these sources will be eliminated after data descriptor standardization and error correction, which means that about 50% of the person counts in the raw data are redundant because of heterogeneous descriptions and errors generated by humans in disparate sources. Furthermore, 50%-60% of source data elements do not match the target schema or among themselves because of semantic heterogeneity (detailed in section 3.3), which means that these elements need to go through a data structure transformation in order to retain the data carried by these elements. Although these simple statistics only reflect the

situation of the ETL executed between our data sources and our particular target schema, it delivers a piece of important information: complicated data transformation and curation are required to achieve personalized biomedical data integrity in the real world.

For a biomedical data source that contains detailed patient data with records of time stamps, even de-identified, the data source usually is not freely accessible because of HIPAA concerns. Therefore, the application of the TraM data is quite different from a conventional biological data source in the way how data is accessed. First, the TraM data that contains healthcare administrative dates can only be viewed through project accounts. Each account usually contains at least one *institutional research board* (IRB) protocol (Fig 7). Researchers need to be approved by the same IRB protocol that patients have consented to in order to view the data within the account.

Under each project account, TraM offers four kinds of user interfaces and each allows a specified user role to access data with a uniquely defined privilege. 1) The role of account administrator has the highest privilege within an account and the person manages users within the account by assigning them different roles. 2) The role of curator has write privileges so the person can edit data even after the data have been loaded into the TraM system (Fig 5). 3) The role of power uses, usually a physician, under IRB approval, has privileges to view the patient's medical record number which is considered as PHI, but has no right to edit data. Finally 4) the role of regular user can view all de-identified TraM data (Fig 7). Although unlimited accounts are allowed to access the TraM data based on IRB protocols, there is only one copy of integrated data in the database. If the IRB protocols that a patient has consented to happen to be in different accounts, the patient's records will be shared by these accounts without duplicate records in database.

5. Discussions

Personalized biomedical data integration is a complicated and perpetual challenge in biomedical informatics. Simply utilizing a single method, technology, or system architecture may not solve all of the problems associated with the process. In comparison to the other software products available in this line of work, the focus of UNIFIN is on data processing for integration, a goal that the system has achieved utilizing current real-world data. The architecture of UNIFIN is supported by a highly abstracted data modelling strategy which provides a solid foundation to be modified, extended, and improved for future and presumably improved source data environments without altering its backbone structure. Issues related to UNIFIN are the following:

Ad hoc versus sustainable solutions: Personalized biomedical data integration is on the frontier of scientific challenges on a day-to-day basis. Rapidly evolving research has forced many to adopt ad hoc data capture solutions to keep the records. These solutions usually capture data in as-is formats and the data, along with its descriptors, are not synthesized with concept abstraction, semantic normalization and vocabulary standardization. Some ad hoc approaches are unavoidable, especially when users who create ad hoc data are not computer professionals. However, we believe that the ad hoc solutions should be limited to raw data capture but not be promoted for multiple source data integration or integrated data management. Considering the cost, scope, and endeavour of a warehousing project, making an effort up front in the design stage to separate data concepts (variable carriers) from data values (instance data) will be rewarded with long term software architecture stability and function flexibility. A software product produced with such planning should be suitable to ad-hoc events and sustainable in a constantly evolving data source environment. Here,

sustainability does not mean that software is static and does not need to be modified. Instead, it means that if required, software can be modified at minimal cost to gain a significant improvement in capacity.

Reuse and impact of human efforts: TraW is a workflow that requires human-intervention. The products of human intervention are data mapping references and domain ontologies. After about three years of effort, we have reduced the time needed for a life-cycle of data processing of the same scope of data from several months to a few weeks while gaining much improved quality consistency in the final product. Yet, one of the major reasons for us not able to further significantly reduce the ETL time at present is semantic heterogeneity of source data and high modification frequency of data sources (minimal 3 times a year in some major data sources). These changes often require human cognitive attention to validate data matching, mapping, and transformation processes. In some cases (e.g., source data elements designed with implied or multiple overloaded meanings) off-line human research is required to form a data processing solution before proceeding. Therefore, it is important to design and maintain these mapping references to make the human-intensive products computable, so that they can be reused in data processing with a high-throughput manner. When these mapping references become more sophisticated, improved automation should be possible. At the current stage, the need of human intervention reveals a major limitation of the UNIFIN approach: TraW needs to be operated by experts in biomedical informatics, which not only slows down a process that was intended to be streamlined, but also has the potential to produce data with uneven quality due to the uncertainty of human behaviour.

Position and contribution of warehousing solution in a biomedical data space: If there is a spectrum for data integration strategies based on their product integrity, warehousing solution appears to fall at the top of the spectrum as the most integrated while mashup at a position of less integrated. However, the two can be successfully interlinked when individualized studies need to be transformed into population studies, e.g., medical census. UNIFIN-like approaches will potentially become conduits that allow for significant amounts of information mashup by providing standardized quality data. In order to form a harmonized ecosystem in the data-space, warehouse data sources need to work towards using interchangeable domain ontologies and CDEs to process data and making these data available for interoperable sharing. If this does not occur, the sprawling warehouses, which usually collect regional biomedical data, may contribute to yet another layer of data heterogeneity.

6. Conclusion

We have created and tested a warehousing framework consisting of a unique database conceptual model and a systematically adjusted and enhanced ETL workflow for personalized biomedical data integration. The result is a real-world tested solution that is capable of consistently and efficiently unifying data from multiple sources for integration and delivering consumable information for use by translational researchers. The UNIFIN is a work-in-progress in the field that demands new knowledge and innovative solutions to this line of work.

7. References

Agrawal, R., A. Ailamaki, et al. (2009). The Claremont report on database research. *Commun. ACM* 52(6): 56-65.

Batini, C., M. Lenzerini, et al. (1986). A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv*. 18(4): 323-364.

Bernstam, E. V., J. W. Smith, et al. (2010). What is biomedical informatics? *J Biomed Inform* 43(1): 104-110.

Bernstein, A. P. and E. Rahm (2001). A survey of approaches to automatic schema matching. *The VLCB Journal* 10: 334-350.

Bernstein, P. A. and L. M. Haas (2008). Information integration in the enterprise. *Commun. ACM* 51(9): 72-79.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res* 32(Database issue): D267-70.

Brazhnik, O. and J. F. Jones (2007). Anatomy of data integration. *J Biomed Inform* 40(3): 252-69.

Butler, D. (2006). Mashups mix data into global service. *Nature* 439: 6-7.

Chen, P. (1976). The entity-relationship model-toward a unified view of data. *ACM Traslation on Database Systems* 1(1): 9-36.

Cimino, J. J. (1996). Review paper: coding systems in health care. *Methods Inf Med* 35(4-5): 273-84.

Cimino, J. J., T. F. Hayamizu, et al. (2009). The caBIG terminology review process. *J Biomed Inform* 42(3): 571-580.

Cochrane, G. R. and M. Y. Galperin (2010). The 2010 Nucleic Acids Research Database Issue and online Database Collection: a community of data resources. *Nucleic Acids Res 38(Database issue): D1-4.*

Cote, R. A. and S. Robboy (1980). Progress in medical information management. Systematized nomenclature of medicine (SNOMED). *JAMA* 243(8): 756-62.

Dinu, V. and P. Nadkarni (2007). Guidelines for the effective use of entity-attribute-value modeling for biomedical databases. *Int J Med Inform* 76(11-12): 769-79.

Edge, S. B., D. R. Byrd, et al., Eds. (2010). *AJCC Cancer Staging Handbook.* 7th Edition, Springer.

Faddick, C. M. (1997). Health care fraud and abuse: new weapons, new penalties, and new fears for providers created by the Health Insurance Portability and Accountability Act of 1996 (HIPAA). *Ann Health Law* 6: 77-104.

Fielding, R. T. and R. N. Taylor (2002). Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology 2 (2): 115-150.

Franklin, M., A. Halevy, et al. (2005). From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.* 34(4): 27-33.

Galperin, M. Y. and G. R. Cochrane (2009). Nucleic Acids Research annual Database Issue and the NAR online Molecular Biology Database Collection in 2009. *Nucleic Acids Res* 37(Database issue): D1-4.

Goble, C. and R. Stevens (2008). State of the nation in data integration for bioinformatics. J *Biomed Inform* 41(5): 687-93.

Goble, C., R. Stevens, et al. (2008). Data curation + process curation=data integration + science. *Brief Bioinform* 9(6): 506-517.

Guimerà, R., B. Uzzi, et al. (2005). Team assembly mechanisms determine collaboration network structure and team performance. *Science* 308: 697-702.

Halevy, A. Y. (2001). Answering queries using views: A survey. *The VLDB Journal* 10(4): 270-294.

Halevy, A. Y., N. Ashish, et al. (2005). Enterprise Information Integration: Successes, Challenges and Controversies. *SIGMOD '05* June: 778-787.

Horig, H., E. Marincola, et al. (2005). Obstacles and opportunities in translational research. Nat Med 11(7): 705-8.

Jhingran, A. (2006). Enterprise information mashups: integrating information, simply. *Proceedings of the 32nd international conference on Very large data bases.* Seoul, Korea, VLDB Endowment: 3-4.

Kalashnikov, D. V. and S. Mehrotra (2006). Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Trans. Database Syst*. 31(2): 716-767.

Kamble, A. S. (2008). A conceptual model for multidimensional data. *Proceedings of the fifth on Asia-Pacific conference on conceptual modelling - Volume 79*. Wollongong, NSW, Australia, Australian Computer Society, Inc.

Kimball, R., L. Reeves, et al. (1998). *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses* with CD Rom, John Wiley & amp; Sons, Inc.

Kolaitis, P. G. (2005). Schema mappings, data exchange, and metadata management. P*roceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. Baltimore, Maryland, ACM: 61-75.

Lenzerini, M. (2002). Data Integration: A Theoretical Perspective. ACM PODS June: 233-246.

Louie, B., P. Mork, et al. (2007). Data integration and genomic medicine. *J Biomed Inform* 40(1): 5-16.

Lowe, H. J., T. A. Ferris, et al. (2009). STRIDE--An integrated standards-based translational research informatics platform. *AMIA Annu Symp Proc* 2009(5): 391-5.

Rizzi, S., A. Abello, et al. (2006). Research in Data Warehouse Modeling and Design: Dead or Alive? *DOLAP'06* November 10, 2006: 3-10.

Sioutos, N., S. de Coronado, et al. (2007). NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. *J Biomed Inform* 40(1): 30-43.

Stein, L. (2002). Creating a bioinformatics nation. *Nature* 417(6885): 119-20.

Stein, L. D. (2008). Towards a cyberinfrastructure for the biological sciences: progress, visions and challenges. *Nat Rev Genet* 9(9): 678-88.

Trujillo, J., M. Palomar, et al. (2001). Designing Data Warehouses with OO Conceptual Models. *Computer* 34(12): 66-75.

Vassiliadis, P., A. Simitsis, et al. (2002). Conceptual modeling for ETL processes. *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP.* McLean, Virginia, USA, ACM: 14-21.

Wang, X., L. Liu, et al. (2009). Translational integrity and continuity: Personalized biomedical data integration. *J Biomed Inform* Feb;42(1): 100-12.

Wong, J. and J. I. Hong (2007). Making mashups with marmite: towards end-user programming for the web. *Proceedings of the SIGCHI conference on Human factors in computing systems.* San Jose, California, USA, ACM.

Yu, A. C. (2006). Methods in biomedical ontology. *J Biomed Inform* 39(3): 252-66.

Biomedical Engineering, Trends in Electronics, Communications and Software

Edited by Mr Anthony Laskovski

ISBN 978-953-307-475-7
Hard cover, 736 pages
Publisher InTech
Published online 08, January, 2011
Published in print edition January, 2011

Rapid technological developments in the last century have brought the field of biomedical engineering into a totally new realm. Breakthroughs in materials science, imaging, electronics and, more recently, the information age have improved our understanding of the human body. As a result, the field of biomedical engineering is thriving, with innovations that aim to improve the quality and reduce the cost of medical care. This book is the first in a series of three that will present recent trends in biomedical engineering, with a particular focus on applications in electronics and communications. More specifically: wireless monitoring, sensors, medical imaging and the management of medical information are covered, among other subjects.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen