ARTICLES

Advanced Biological and Chemical Discovery (ABCD): Centralizing Discovery Knowledge in an Inherently Decentralized World

Dimitris K. Agrafiotis,*,† Simson Alex,† Heng Dai,‡ An Derkinderen,§ Michael Farnum,† Peter Gates,† Sergei Izrailev,† Edward P. Jaeger,† Paul Konstant,† Albert Leung,‡ Victor S. Lobanov,† Patrick Marichal,§ Douglas Martin,‡ Dmitrii N. Rassokhin,† Maxim Shemanarev,† Andrew Skalkin,† John Stong,† Tom Tabruyn,§ Marleen Vermeiren,§ Jackson Wan,‡ Xiang Yang Xu,† and Xiang Yao‡

Information Technology, Johnson & Johnson Pharmaceutical Research & Development, L.L.C., 665 Stockton Drive, Exton, Pennsylvania 19341, Information Technology, Johnson & Johnson Pharmaceutical Research & Development division of Janssen Pharmaceutica N.V., Turnhoutsweg 30, 2340 Beerse, Belgium, and Bioinformatics, Johnson & Johnson Pharmaceutical Research & Development, L.L.C., 3210 Merryfield Row, San Diego, California 92121

Received July 24, 2007

We present ABCD, an integrated drug discovery informatics platform developed at Johnson & Johnson Pharmaceutical Research & Development, L.L.C. ABCD is an attempt to bridge multiple continents, data systems, and cultures using modern information technology and to provide scientists with tools that allow them to analyze multifactorial SAR and make informed, data-driven decisions. The system consists of three major components: (1) a data warehouse, which combines data from multiple chemical and pharmacological transactional databases, designed for supreme query performance; (2) a state-of-the-art application suite, which facilitates data upload, retrieval, mining, and reporting, and (3) a workspace, which facilitates collaboration and data sharing by allowing users to share queries, templates, results, and reports across project teams, campuses, and other organizational units. Chemical intelligence, performance, and analytical sophistication lie at the heart of the new system, which was developed entirely in-house. ABCD is used routinely by more than 1000 scientists around the world and is rapidly expanding into other functional areas within the J&J organization.

INTRODUCTION

Connecting disparate pieces of data into a unified whole is one of the pharmaceutical industry's most pressing needs.[1−4] Scientists working in modern pharmaceutical research organizations are usually confronted with an intricate web of databases and end-user applications providing access to a wealth of information and data analysis and visualization capabilities. Although some integrated systems have been reported in the recent literature,[5−7] in most organizations integrated access to all the data needed to advance a discovery project is often impossible, forcing users to maintain their own personal "databases", typically in the form of Excel spreadsheets.

The situation at Johnson & Johnson Pharmaceutical Research & Development, L.L.C. (J&JPRD) in ca. 2002 was very similar. Owing in part to J&J's decentralized organization, several independent data systems had been implemented by J&JPRD and its affiliated pharmaceutical companies for tracking chemical and pharmacological data. These systems had very little in common. Each came with its own ontology, architecture, interface, security model, quality control standards, and degree of completeness. While this environment allowed affiliated companies to be productive at a local level, convenient and consistent access to data generated at different sites and deposited at different source systems became an objective to facilitate moving compounds rapidly into full development.

More importantly, it became apparent that scientists consumed a significant amount of their time moving data from application to application, in order to gain access to specific data retrieval, analysis, and reporting capabilities (Figure 1). The rampant popularity of the Internet led to a proliferation of underpowered web-based systems, establishing the HTML table as the second most popular interface for scientific data delivery and exchange, next to Excel.

In the absence of a comprehensive data integration platform, data pipelining tools like SciTegic's Pipeline Pilot[8] found significant utility as a means to integrate disparate data scattered in ad hoc documents and data sources into more manageable data sets. Pipelining tools "wrap" data manipulation components into processing elements that implement specific input/output interfaces, which allow the output of each element to be used as input to another. These elements

* Corresponding author phone: (610) 458-6045; fax: (610) 458-8249; e-mail: dagrafio@prdus.jnj.com.
† Exton, PA.
‡ La Jolla, CA.
§ Beerse, Belgium.

10.1021/ci700267w CCC: $37.00
Published on Web 11/01/2007

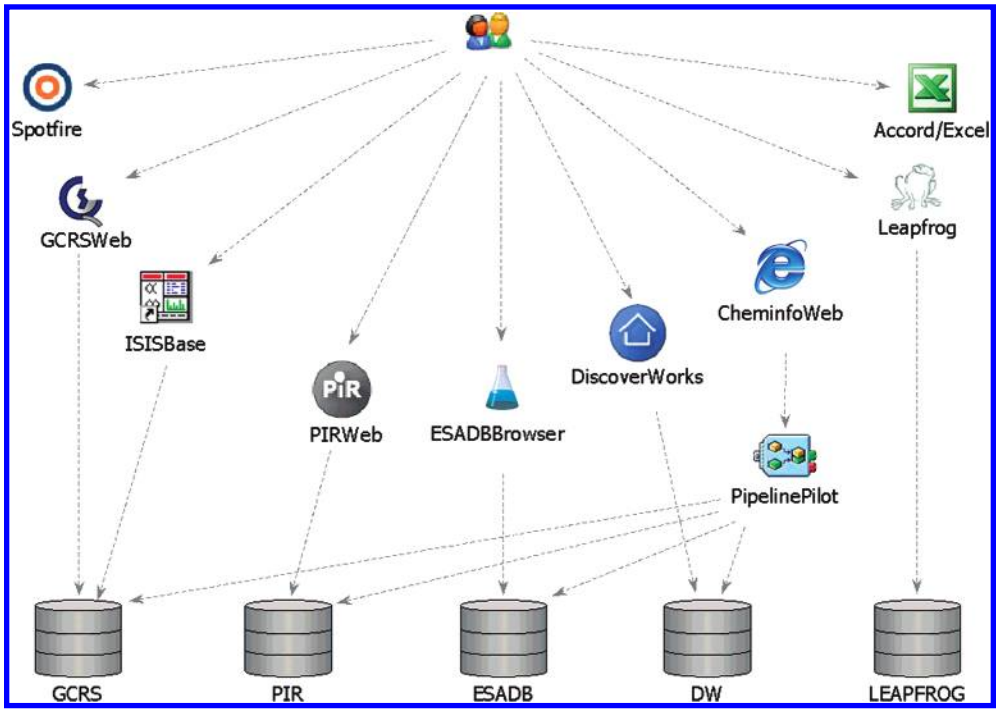

Figure 1. Simplified, schematic view of the discovery informatics landscape at J&JPRD prior to ABCD.

can then be assembled into complex data processing workflows using an intuitive graphical user interface. This environment has allowed computational chemists, IT professionals, and IT-savvy end-users to create and persist their own workflows with minimal programmatic effort and deploy them for enterprise use. Although mature platforms such as Pipeline Pilot offer a multitude of configurable components, the most commonly used functionality involves data import, merging, and filtering.

The need to combine heterogeneous data sources under a single query interface has been around since the widespread adoption of databases in the 1970s. The two most popular strategies are data federation and data warehousing. Both aim at providing the user with a unified view of the data, and both have found utility in the commercial as well as the scientific domains.

A federated database management system (FDBMS) is a meta-database, which transparently integrates multiple autonomous databases into a single conceptual unit. It can be thought of as a virtual database that represents the logical, rather than the physical, union of the constituent data sources. Users retrieve the data by posing a query to a virtual, mediated schema. The query is internally decomposed into specialized subqueries submitted to the original source systems, and the results are combined to reconstitute the final result set. Since different DBMSs often employ different query interfaces, federated systems make use of "wrappers" or adapters that translate the subqueries into the appropriate query languages and transform the subquery results into a form that can be easily processed by the integration engine. This approach allows new data sources to be incorporated by simply constructing a suitable adapter for them. This contrasts with data warehousing solutions, where the new data must be physically copied into the system (vide infra).

Although on the surface federated systems appear to be easier to construct, there are several obstacles that hinder their effective implementation. Besides the need to accommodate multiple query languages, if the overall system is not planned from the top-down (as is most often the case), incompatible data types (e.g., an attribute represented as a string in one system and an integer in another) and semantically equivalent entities, which are named differently in the various source schemas, pose additional complications. A common solution to the latter problem is the use of ontologies, which resolve semantic conflicts through explicitly defined schema terms. To circumvent the problem of constructing and maintaining all possible $n(n-1)/2$ equivalence mapping rules between a set of n attributes, most FDBMSs employ a global schema that encompasses the relevant parts of all the source schemas and provides mappings in the form of database views.

Just as with data pipelining, the most important limitation of federated systems is lackluster performance. In most practical applications, the source systems tend to be transactional in nature and are not tuned for the kind of query performance that is required from a decision support system. The designer of a federated database typically has no control over the source systems and cannot optimize the speed at which the various subqueries execute. For decision support systems where performance is essential (as in ABCD), data warehousing is a more attractive option.

In data warehousing, data from several source systems are extracted, transformed, and loaded (ETL) into a new repository, which can be queried from a single schema. This is a tightly coupled approach, in the sense that the data reside together in a single repository at query time. The three obvious shortcomings of data warehouses is that they are difficult to construct when some of the source systems are only accessible through a query interface, that the data need to be replicated, and that the warehouse is never "current". Updates to the original data sources are not reflected at the warehouse level until the next ETL cycle. Fortunately, for many applications including the one described herein, this is not a serious impediment; indeed, the 24-h ETL cycle

Table 1. Key Differences between Online Transactional Processing Systems (OLTP, or Operational Systems) and Online Analytical Processing Systems (OLAP, or Data Warehouses)

	OLTP system	OLAP system
source of data	operational data; OLTPs are the original source of the data	consolidated data; OLAP data come from the OLTP source systems
purpose of data	to control and run fundamental business tasks	to help with planning, problem solving, and decision support
what the data reveal	a snapshot of ongoing business processes	multidimensional views of various kinds of business activities
inserts and updates	short and fast inserts and updates initiated by end users	periodic long-running batch jobs refresh the data
queries	relatively standardized and simple queries returning relatively few records	often complex queries involving aggregations
processing speed	typically very fast	depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
space requirements	can be relatively small if historical data are archived	larger due to the existence of aggregation structures and historical data; requires more indexes than OLTP
database design	highly normalized with many tables	typically denormalized with fewer tables; use of star and/or snowflake schemas
backup and recovery	backup religiously; operational data are critical to run the business, data loss is likely to entail significant monetary loss and legal liability	desired but not essential, some environments may consider simply reloading the OLTP data as a recovery method

adopted in ABCD has proven acceptable by the vast majority of our users.

At this point, it is important to note that the design of databases intended for online analytical processing (OLAP) is fundamentally different from those intended for online transactional processing (OLTP). OLTP systems (also referred to as operational or transactional systems) are characterized by a high volume of small transactions such as the registration of a new chemical compound synthesized by a chemist or a new set of results obtained in a biological assay. These are structured and repetitive operations that are short, atomic, and isolated and require a detailed and up-to-date view of the data. The expectation is that each transaction will leave the database in a consistent state. A transactional database is always current, and consistency and recoverability are absolutely critical for the success of the enterprise.

In contrast, OLAP systems such as federated databases and data warehouses are designed to facilitate multidimensional data analysis and decision support and are primarily read-only. An OLAP system analyzes the results of daily business in order to make tactical and strategic decisions that can change the course of the business and gain competitive advantage. OLAP systems extract historical data that have accumulated over a long period of time and provide flexible views of that data from different perspectives and at different levels of abstraction. The key differences between OLTP and OLAP system design are summarized in Table 1.[9]

While effective organization and querying of data is a prerequisite for building an integrated system, success is ultimately determined by the overall end-user experience. We believe that this experience starts and ends with the graphical user interface. From a scientist's point of view, the organization of the data on the back-end is of little intrinsic interest. What is important is the ability to pose relevant questions, retrieve the results in an expedited manner, and present it in a way that offers insight into the underlying biological phenomena. Indeed, the ultimate goal of data integration is not to present the user with endless arrays of numbers but to convert them into knowledge and insight. In drug discovery, that insight can only come through effective analysis and visualization of SAR. This requires a dramatic shift in the way end-user applications are designed and integrated.

For decades, applications have been built to serve a single purpose for a single set of users without a central vision or strategy and without sufficient thought as to how they could be extended and leveraged across different problem domains. Much of the legacy software that is in use today was developed using arcane and proprietary technologies that did not enable reuse of functionality and seamless movement of information. Today's researcher is confronted with a myriad of different applications, each with its own design principles, input/output protocols, performance standards, user interface conventions, and aesthetic preferences. The end result is a confused and frustrated user.

While recent advances in enterprise application integration have enabled more effective communication between applications, the undercurrent of such efforts seems to be the desire to share data and processes without having to make sweeping changes to the applications and/or data structures. We felt that such a sweeping change was both timely and more sustainable in the long-term. Our strategy was to develop a multifunctional application that was endlessly modular and extensible, yet powerful beyond comparison (Figure 2). This application, known as Third Dimension Explorer (3DX), was developed on Microsoft's .NET framework and capitalized on our huge prior investment in high-performance C++ code.[10]

In the remaining sections, we provide an overview of the key transactional systems that serve as data feeds for ABCD and provide a brief description of the major components of the new system.[11]

LEGACY SOURCE SYSTEMS

Global Compound Registration System (GCRS). The purpose of the compound registration system is to provide a

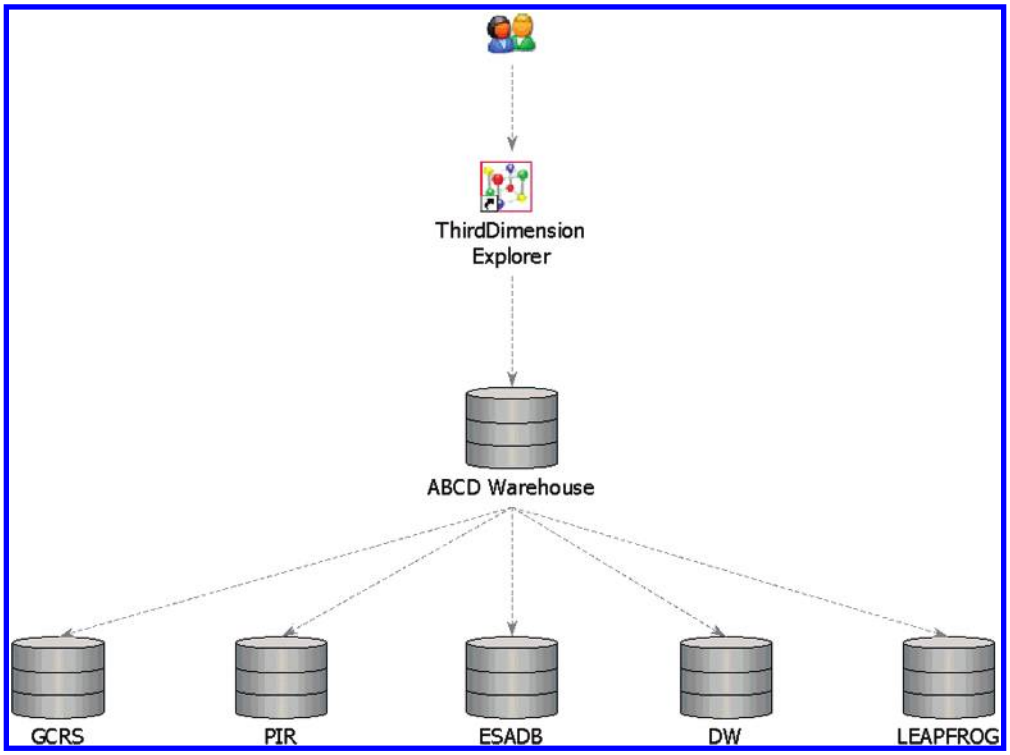

Figure 2. Schematic view of the discovery informatics landscape at J&JPRD after ABCD.

common platform where compounds, batches, salt forms, and aliases are consistently represented and serve as the authoritative source for this information. It supports a consistent set of business rules that define the representation, identification, and depiction of chemical structures, their salt forms (unique forms of structures with salts, solvents, and adducts), and their synthesis batches. These unique identifiers, the JNJ numbers, are used to identify chemical entities throughout the discovery and development process. The system is built on the MDL Isis technology[12] and consists of an Oracle backend database with the Isis Direct chemical cartridge and a custom middle tier registration service that supports individual and bulk registrations, enforces business rules, and provides validation services. The system uses three databases to ensure performance and reliability. A single master database serves as the registration database. Two synchronized replicate database instances, deployed in Europe and the United States, provide the searching capacity. The main user interface is a registration client that supports the chemist's preparation of a registration job, including sketching of the compound structure. Searching and reporting functionalities are provided by a custom application built on MDL Draw technology and a standard IsisBase database form. Simple reporting via various internal Web portals is also supported. A key component to the success of the registration system is the thought and effort placed into the harmonization process. As the authoritative source of chemical identity for the enterprise, GCRS is synchronized with dependent systems such as the compound inventory, biological data registration systems, and downstream development processes.

PIR. The PIR (Pharmacological Information Retrieval) system has been supporting the European discovery community of Janssen Pharmaceutica (a subsidiary of Johnson & Johnson) for the past three decades. The PIR database contains nearly 40 million aggregated biological results from more than 3500 assays and represents the cumulative corporate knowledge of 25 years of pharmacology research at that site. PIR's longevity can be attributed in part to the site's structured approach to data capturing and uploading. Its key strengths are fast query performance, consistency in data processing, and the use of scoring. Assays with similar data processing requirements are supported by a limited number of standard curve-fitting routines, resulting in standardized, content-rich, aggregated estimates of activity. In addition, the assayer, as subject-matter expert, assigns a score to classify compounds as highly active, active, or inactive in that particular assay system. This score not only encapsulates valuable additional contextual information and data interpretation from the subject-matter expert but also serves to normalize the results on a common scale, which simplifies comparison of compound activities across multiple unrelated assays. Compound availability and computed molecular properties are also included, allowing the user to filter the results by any of these additional attributes. A particularly popular feature is the ability to create and manage lists of compounds and assays at a personal, departmental, or discovery project level. Users interact with the system through a tabular Web interface that provides drill-down capabilities to visualize dose-response curves and offer the ability to export the data in text or Excel format for further analysis. The strong standardization has allowed the system to run with relatively minimal application support.

While PIR is perceived to be very effective, it is limiting in some important respects. From a user's perspective, the system lacks chemical searching capabilities and integrated data analysis and visualization tools and is too tightly coupled to a specific, highly structured way of conducting research. Support for nonstandard data processing is rather weak, as is the ability to accommodate more dynamic discovery processes. From an architectural point of view, perhaps the most significant limitation is the use of a proprietary database

platform and a nonstandard query language that offers only a limited subset of SQL's capabilities. Given the state of database and hardware technology at the time of its inception, the designers of PIR opted for an architecture where all the data were resident in the computer's main memory (RAM). While this permitted fast execution of complex Boolean queries, it became restrictive as advances in commercial technology obviated the need for caching the entire database in memory and offered a number of advanced services such as cost-based optimization, replication, clustering, integrated data access programming interfaces, and many others.

ESADB. The Extended Structure Activity Database is the primary repository of biological data in the three major North American sites of J&JPRD. The system was originally designed as a way of recording and sharing high throughput screening (HTS) results and was first used for that purpose in ca. 1996. Later, its scope was extended to include lower throughput secondary assays, both in vitro and in vivo. As of 2006, its size and scope were comparable to the PIR system, holding over 30 million separate results taken from more than 2000 different biological assays. ESADB as a whole consists of four main parts: a central Oracle data store and three end-user applications, including a protocol registration system, a data uploader, and a query front-end.

ESADB's central database is primarily transactional in structure. However the database also exhibits some characteristics typical of data warehouses. The data measures, or "facts", from all result types, are gathered into a single highly indexed table. Moreover, some of the important tables joined to these measures, including protocol and result type, are partially denormalized in a manner suggestive of data warehouse "dimensions". The pharmaceutical process flow begins with the user entering information through ESADB's protocol registration system. When experiments are complete, data are loaded through an interface that makes use of standard Excel templates, which map results to database fields exposed through the loader's interface. Data are viewable through the a Web-based browsing tool designed to allow users to make common queries, without requiring customization.

The ESADB system was a reasonable solution in its prime, providing a single central storage for pharmacological data, with a hybrid database structure, which had the potential for good performance. A range of supporting interfaces, originally written in Delphi, provided a reasonable client/server experience at the time. But perhaps the greatest virtue of ESADB has been its flexibility. The database can house pharmacological data of many different types and allows users to handle and interpret data according to standards developed within their own teams, without enforcing a fixed methodology. The structure of result types, in particular, has a good balance of flexibility and comprehensiveness, allowing basic names to be reused and splitting off measurement details into secondary fields. Furthermore, the protocol registration interface was an important step toward the orderly classification of different protocols and toward a more comprehensive registration of biological detail to support searches and analysis. Its versioning structure also allowed for the quick generation of similar protocols and fostered the possibility of protocol reuse.

The ESADB system, however, did not always exhibit the promise of its early design, and certain limitations have been exacerbated over time. The performance of its various front-ends is far from optimal, particularly when used across the WAN, though the subsequent recoding of the data loader in Java improved the tool markedly. Some of the original code was developed externally, and significant internal expertise was lost over time; these facts have made it difficult to make adjustments in both the user interface and the database back-end. The incremental nature of ESADB's expansion in scope, from HTS to in vivo assays, has also resulted in inconsistencies and awkward compromises at the design and data representation levels. Even its flexibility proved to be problematic. Certain key fields were not carefully controlled, with the result that over time many trivially different versions of the same conceptual entities have been entered, or, conversely, identical fields have become used for significantly different purposes. These infelicities compromised searches and comparative analyses. Finally, the front-end query facilities native to the system are limited in their scope. If users wish to make sophisticated and graphical analyses of SAR, they have to export their results into external tools.

DiscoverWorks. DiscoverWorks is an integrated informatics platform developed at 3-Dimensional Pharmaceuticals, Inc., a technology-based biotechnology company founded in 1993, acquired by Johnson & Johnson in 2003, and later merged into J&JPRD. It was built to support a variety of informatics needs, including compound and sample management, reagent and inventory management, plate management, high-throughput chemistry, protein expression, preparation, and purification, X-ray crystallography, and registration of in vitro biological protocols and results (including those obtained with 3DP's proprietary biophysical assay platform known as ThermoFluor[13]). The system was built on Oracle technology with a first generation chemical indexing engine. The user interface is Web-based and built with custom controls that provide sophisticated chemistry awareness and searching glued together with Java scripting. A key strength of DiscoverWorks is its ability to integrate information on compounds, genes, and crystal structures with biochemical, biophysical, cellular, and pharmacokinetic data. While DiscoverWorks can handle a variety of biological results types, it has predominately been used for in vitro data. As a transactional system, it excels in its ability to represent data consistently. The Web-based user interface, however, does not permit the large-scale data mining that was a critical requirement for ABCD, and the development environment leaves much to be desired in terms of programmability and ease of maintenance.

Leapfrog. Leapfrog is a target and project tracking system based on a sophisticated gene id integration system called GeneView. GeneView allows a user to register a target using a sequence identifier from any major sequence databank, such as Genbank, Embl, etc. as well as proprietary systems. Once registered, the system links all the information regarding a gene together even though a multitude of sequence ids could be used to represent the gene. GeneView accomplishes this by prematching all the sequence ids using text, id, and sequence-based comparisons. This allows Leapfrog to inform the user when a gene is already involved in another project.

Although Leapfrog was created to manage mainly drug discovery projects, its design allows it to handle any projects that can be described in terms of stages and steps. For drug discovery, the typical stages include target identification,

assay development, hit to lead, lead optimization, and drug evaluation. An important benefit of Leapfrog is the standardization of names for projects, stages, and steps across all sites, which simplifies global views, reports, and comparisons.

THE ABCD SYSTEM

Our team set out three major goals: (1) establish a coherent, unifying framework for organizing chemical and biological data; (2) eliminate the multitude of applications and interfaces for extracting, analyzing, and reporting that data; and (3) create a collaborative framework that would support the project-team-based organizational structure of J&J's pharmaceutical R&D sector. These goals resulted in three key deliverables: (1) the ABCD data warehouse, (2) the Third Dimension Explorer application suite (also referred to as 3DX), and (3) the Workspace.

1. Data Warehouse. The ABCD project required a central storage for data from multiple legacy source systems, each with different conventions for organizing the variety of experimental results generated in the drug discovery process. The set of source systems, moreover, was not stable; new legacy sources from affiliated partners had to be incorporated at any time, and new consolidated transactional sources were also planned. In addition, the project required a high performance query engine in order to support intensive and large-scale scientific analysis. All these requirements immediately suggested a data warehouse approach, following the guidelines pioneered by the work of Ralph Kimball.[14]

The most notable characteristic of data warehouses is the use of redundant or denormalized data. Normalization is the process of restructuring the logical data model of a database to eliminate redundancy, organize data efficiently, and reduce the potential for anomalies during data operations. Data warehouses are constructed using a design technique known as dimensional modeling (DM). DM introduces redundancy to provide for simplicity in query construction and high-performance access. Every dimensional model is composed of one table with a multipart key, called the fact table, and a set of smaller tables called dimension tables. Each dimension table has a single-part primary key that corresponds to exactly one of the components of the multipart key in the fact table, resulting in a characteristic "starlike" structure, also referred to as a star schema or a star join. (The normalized alternative to the star schema is the snowflake schema.) Dimensional models are usually contrasted with conventional entity-relationship (ER) models, which aim at removing redundancy in the data. While ER modeling is highly beneficial in OLTP applications because transactions become simple and deterministic, it should be avoided for end-user delivery because it leads to complex and inefficient queries. In DM, data denormalization must be carefully controlled during ETL processing (vide infra), and users should not be permitted to see the data until they are in a consistent state.

Most data warehouses, however, have been built to support the consolidation and analysis of conventional business data, such as sales and marketing results. Scientific research data are different in many important respects: in the nature of and relationship between dimensions, in the nature of fact aggregations, and in the relatively nonstandard and continuously evolving character of the dimensional attributes themselves. Different research sources would inevitably describe "dimensions" with idiosyncratic sets of attributes, and conceptually identical choices within attributes would often be captured with highly variable names and descriptions. In the pharmacological research context, for instance, attributes variously named as "gene," "protein," "target," and "project" may well point to the same entity or to entities confusedly interrelated. To be successful, the ABCD data warehouse had to find the means to allow for all these differences without losing the key virtues of the dimensional approach.

The first step in any data warehouse project is to determine the nature and granularity of a warehouse fact and then to determine the overall structure of the dimensions that describe those facts. One way to look at the distinction between them, in a scientific context, is through the central concepts of independent and dependent variables. Independent variables are those physical aspects of an experiment controlled by the experimenter, and they correspond well with warehouse dimensions. Dependent variables are observations made during the course of an experiment and can be seen as facts.

The drug discovery research process can be viewed as the assembly of results from experiments testing the biological activity or the physical properties of small organic molecules. The goal of this assembly is to facilitate SAR analysis, i.e., to understand the relationship between the structure of the small molecules tested and their associated activity and properties, with the ultimate aim of designing a drug useful for treating a particular disease or indication. Early in this process the primary goal is to identify those molecules with a significant threshold of biological activity and, using the structure of these hits as a guide, to then synthesize additional molecules of greater potency. This process suggests a dimensional structure that would separate out the following major independent variables or groups of variables: (1) the identity of the molecule being tested; (2) the concentration at which the molecule was tested; (3) the identity of the protein target against which the molecule was tested; and (4) a description of the way in which the test was performed.

Item 1 reflects the ultimate aim of discovering a novel compound or drug. In ABCD, this is implemented in the chemistry dimension, a single table exhibiting the denormalized hierarchical structure typical of warehouse design. The level directly connected to the facts is the individual batch of a synthesized compound; above that are the various salt forms used in a compound's batches; and above that is the abstract molecular structure which represents the compound itself.

Item 2 rests on the recognition that potency is a primary consideration for drug activity. This is implemented in the concentration dimension, which consists primarily of a set of discrete numeric values, representing the intended or the rounded measured concentrations of compounds used in experiments. Since concentration can be recorded using units of distinct type, such as log molar, molar, or mg/mL, a unit field is also included.

Item 3 reflects the need to understand drug activity in the context of particular indications, which themselves depend on the pathways in which protein targets participate. This is implemented in the target dimension, which like chemistry

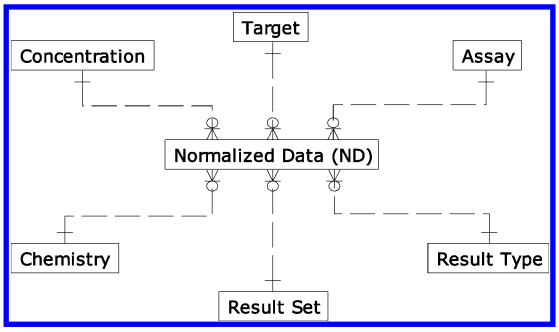

Figure 3. ABCD dimensional model, forming a type of star schema typically associated with warehouse design.

is hierarchical in structure. The lowest level identifies the particular sequence of the expressed protein used in the assay. Above that is a common identifier for a "generic" version of the protein, inclusive of variations within and across related species. The highest levels include protein family information derived from public sources.

Finally, item 4 is captured in the assay dimension, a kind of catchall, gathering together a myriad of independent variables documenting experimental conditions. Reusable sets of independent variables are, in effect, captured as distinct protocols, identified at the top of the assay hierarchy, which also points to the complete protocol documents, separately stored. At the bottom of the hierarchy is the assay run, which includes attributes associated with the individual experiments as they are performed, including notebook references and name and team of the assayer as well as experiment and load date information.

The dimensions outlined above are derived from an analysis of in vitro assays, which was sufficient for ABCD's initial scope, but the system is in the process of being expanded to include in vivo assays as well. This will necessarily require the addition or separate expansion of several other dimensions, including structures representing biological subject detail, dose delivery attributes, and the time intervals at which observations are taken. Two further dimensions are also required by the particular nature of the dependent variables (facts) in drug discovery. The typical sales or marketing warehouse can take advantage of the abstraction associated with monetary systems and, by assigning each fact a shared monetary measure, allows for the direct comparison of a variety of economic activities. In contrast, scientific research involves observations of many more different kinds and with measurement units that cannot be simply resolved and compared. This suggests the utility of a fifth, separate dimension, in addition to the four major ones outlined above that clearly identifies the kind of observation or *result type*. It is also true that in scientific research multiple observations or result types can be gathered at the same time, under a common set of conditions (independent variables), and that the interrelationship between these observations is often crucial to scientific understanding. This suggests the usefulness of a sixth dimension clearly identifying these *result sets*. Figure 3 shows how these six dimensions, along with the experimental observations, form a type of star schema typically associated with warehouse design.

The fact that a scientific warehouse cannot make use of a single shared abstraction, such as the dollar, in the design of its fact tables, is also related to its particular requirements

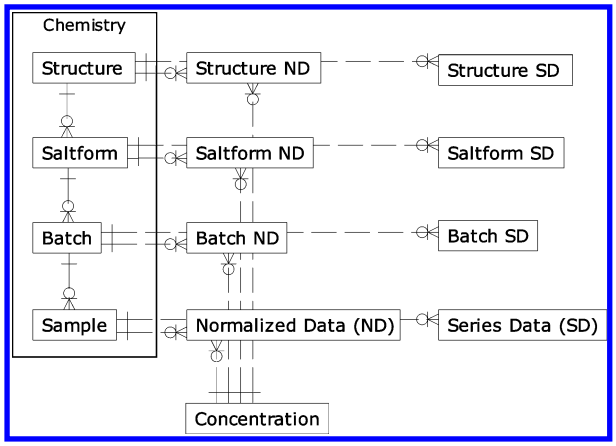

Figure 4. Fact tables supporting multiple aggregation levels at the compound (vertical) and parameter estimation (horizontal) levels.

for the aggregation of these facts. In a business warehouse by far the most frequent aggregation is summation; total dollar amounts are summed and compared across various dimensional slices, comparing total sales, for instance, across sales regions or across products. In ABCD, on the other hand, there are two major types of aggregations, neither of which involve summation. The first of these involves means or medians of the various measures of compound activity, typically moving up and down from a replicate or a compound hierarchy. Scientists may sometimes wish to consider results at the individual batch level, but often they wish to get an indication of the overall performance of an abstract compound structure, averaging over all the batches and salt form manifestations of that structure (e.g., when generating quantitative models of biological activity or QSAR). The second type of aggregation involves parameter estimation over individual data points that do not vary randomly, but according to a planned perturbation of independent variables, in order to determine a higher order scientific measure that is more useful, and indeed more meaningful, than the individual results underneath. A very common example of this would be an IC50 determination, a parameter based on fitting a curve to a series of measured inhibitions of a biological activity at known compound concentrations. The result is a very useful estimate of the compound concentration necessary to inhibit a biological activity by 50%, an estimate that allows an easy comparison of efficacy across many different compounds.

It is possible that both of these types of aggregations can be performed in real time, based upon a single fact table, and to some extent the ABCD system supports just that. But because of the very large cardinality of the compound dimension, on the one hand, and the complexity of parameter estimations, on the other, we have optimized the warehouse performance by precalculating these aggregates and storing them in several tables, that are in effect projections of both types of aggregation, as can be seen in Figure 4.

Structuring the dimensions and the facts of our pharmaceutical warehouse in this way ensures that we can enjoy the two major advantages that star schemas have over fully normalized, transactional designs. The first of these is performance. In the star schema above, there is never more than one join required to get to a fact. And since facts are numerous, the tables housing them should be highly normal-

ized and compact. The physical storage for a fact row is kept to a minimum by moving context information out to the dimensions, accessible through a single foreign key per dimension. The second major advantage of the star schema design is simplicity. Queries are easy to construct due to the simple topology of the star design. In effect, where possible many-to-one relationships are prejoined into a hierarchical structure within each dimension. The net result is wide short tables for dimensions and narrow long tables for facts.

2. ETL. Perhaps the greatest challenge to creating a scalable and sustainable scientific data warehousing project comes from determining the rules for populating the integrated database. Whatever the design, normalized or dimensional, changes in scientific understanding and ever-increasing demands for functionality lead to changes in the design of both the source systems and the warehouse. Further, changes in personnel implementing these algorithms introduce knowledge transfer issues. Other documented biological data warehousing efforts have failed after a relatively short period of time due to the ongoing effort required to keep up with these changes.[15]

In order to address these challenges, we leveraged commercial data migration tools for extraction, transformation, and loading (ETL) widely used in the data warehousing industry. With basic database connectivity, integration functions, and workflow scheduling built-in, we were able to focus on data profiling to determine the integration rules. With these rules documented, implementation within the GUI framework became a substantially streamlined process.

Effective ETL algorithms require several vital components and feature a number of seemingly conflicting requirements. Every new, modified, or removed result must be reflected in the data warehouse as soon as possible, so there must be an error-proof method of Change Data Capture (CDC) from every source database. However, the CDC algorithm must not place an undue burden on the source system. Within the ABCD project, a single logging table recording every change in the source system was populated through triggers on each table. Considering the assay dimension described earlier, the protocol is entered into the system once and then reused to perform experiments over many days, weeks, and months. The independence of the CDC algorithm allows each record to be loaded into the warehouse as the records are created. The CDC algorithm then faces the challenge of knowing the exact time of the last successful execution. With the exact start time of the algorithm written back to a database entry, this limit can be applied correctly to a fraction of a second.

The ETL queries must have little impact on the performance of the source system, which may be in active use by other users at the same time. Within the global environment of our company, these algorithms also must tread lightly on the wide-area network. Get in, get it, and get out is the guiding principle in the design of queries against production systems. Each of these changes must be captured within the warehouse environment. However, as shown in the dimensional design section above, performing all the denormalization within the source query ETL is too likely to create a performance burden on these algorithms. To solve this issue, we have decided to create an intermediate copy within so-called staging areas, which allowed us to add many other features required for successful ETL systems.

Within the staging area, the ETL algorithms must capture the type of change, generate surrogate keys for new records, and prepare the new records for final loading into the data warehouse. As the data are written by the source-to-staging algorithm, the ETL tool provides for automated "update else insert" logic based on unique fields defined in the staging table. Once again, table triggers, this time based on database sequences, generate data warehouse surrogate keys. Looking again at the assay records, the staging table for experiments assigns the surrogate key that will be used on the dimension records. The source system provides the unique identifiers used to identify a new or updated record. To accommodate deleted data from the source system, a separate component captures the identifiers of these records from the logging table and updates the staging records to reflect the deleted status. These logically deleted records in the staging area are then physically deleted in the warehouse dimensions and facts.

Loading the data into the warehouse requires denormalizing the records for dimensions and resolving every dimension key for the results. These algorithms do the "heavy-lifting" of the ETL effort, which is then performed entirely within the data warehouse server. Since this is entirely local, these processes run much more efficiently and have no impact on the performance of the source systems. Within the drug discovery data warehouse environment, if a result is changed for any reason, only the current result must be maintained. This logic allows for a direct update of the record within the warehouse. For the assay dimension record, when additional information such as the lab notebook page is provided, the record is simply updated, and all results are immediately associated with the added information. Further, some changes may be applied to multiple records for this dimension. When a user's name changes, for example, all experiments can be modified from a single, separate ETL algorithm.

Data quality, complete and correct reflection of every scientific result, is a crucial goal for the ABCD project. If a single record is lost, there is a chance that the opportunity for the next blockbuster drug is lost with it. If the scientist who generated the data discovers missing data within their assay, they ask how much data are missing for other assays with no way of having a definitive answer. However, with millions of results being generated from the global discovery organization every day, the chance that some record will not find every component needed—a compound from the registration system may be delayed getting to the warehouse, for example—becomes increasingly likely.

Late arriving dimension records are managed through exception handling ETL algorithms. When the calculated results—which comprise the fact table records of the dimensional model—are loaded, any record that fails to find the correct dimension entry is diverted to a rejected data table. Another ETL component re-evaluates these rejected records within each batch loading process. When the record is successfully loaded, the rejected record is updated to show the resolved status. These rejected data tables are also further evaluated periodically to audit the process. How long did the record remain rejected? What caused the record to be rejected?

Commercial ETL tools also assist the ongoing maintenance efforts. The incremental ETL processes require continuous monitoring to ensure that the continual stream of exceptions

to any rules generated by the scientific community are integrated correctly. Further automation is integrated into the process with automated reporting of each batch execution sent via e-mail to support personnel. These reports reflect both the successes and the failures from the previous execution so a complete audit is maintained.

Since ABCD integrates data across multiple systems developed in different generations of technology and in different locations that supported different processes, there were dramatic inconsistencies in terminology. A protocol in one system was a procedure in another and an assay in a third. Mapping all of these concepts into a common ontology quickly became a necessity.

With an integrated data warehouse comprising thousands of protocols, the need to provide multiple ways of finding protocols of interest to a particular project is of paramount importance. Again, diversity reigns across sites, time zones, and countries. Each of the existing systems may have provided one such mechanism, but no two were the same. With the assistance of the scientific community, a curation process was established, which allows the curators to manually update one record with the pertinent classification data. These curated records are then integrated into the ETL processes to apply these values in all appropriate places within the ABCD warehouse.

3. Ontologies and Curation. Data cleansing, curation, and the establishment of useful domain ontologies are necessary parts of any data warehousing project, but the nature of pharmacological research presents challenges of a scope not commonly seen in other business contexts. A marketing or sales warehouse may require extensive cleansing of fields subject to common data variation, such as names of people, sales offices, products, and the like. But the meaning of the individual choices is usually clear enough, and it is generally straightforward to decide the canonical form of an individual choice to guide the cleansing effort. But in the fluid and continuously evolving world of science there is often a bewildering variety of significantly different terms, and terminological systems, for the same, or for overlapping, concepts—one scientist's "platelet" may be another's "thrombocyte"—and it is not always easy to determine what should be considered canonical. Furthermore, in the research context it is not just the choices within fields, but the dimensional attributes themselves, and their interrelationships, that can be problematically dynamic. It is reasonably easy to resolve the potential overlaps between a region and a state for marketing purposes, but it is much more difficult in a research database to decide unambiguously where a cell type should be distinguished from a tissue type or a tissue type from an anatomical structure.

One approach to these difficulties is to base the warehouse's scientific ontologies, where possible, on authoritative public sources, sources that have already devoted considerable resources to such problems and will continue to do so. But these public sources do not always exist, and when they do, they often exist in confusing and competitive superfluity. In such cases one must either make a difficult judgment between multiple suitors, cobble together a serviceable monster from competing candidates, or attempt to construct a workable ontology of one's own. In ABCD, all the above approaches have proved appropriate at times, depending on the domains where curation was required, which included molecular targets, biological subjects, and the classification of assays and experiment types.

Since the structure of business processes in modern drug discovery is built upon the centrality of a particular protein target, and since the names in common usage for these proteins remain variable, a stable identification and coherent classification of molecular targets in the warehouse were vital. Fortunately, in this area we could depend on one major authoritative source, the Gene Nomenclature Committee of HUGO,[16] the Human Gene Organization, to provide us with canonical names and symbols to serve most of our needs. However, in HUGO these names apply to a little more than half of human genes, while the assays in ABCD make use of the translated protein products, products that are often not human, and products that are sometimes protein complexes. Therefore, all existing targets were manually curated to ensure appropriate identification when possible, and all new targets are subject to review at registration. The ABCD warehouse also provides a simplified three level protein family classification for targets. This classification is ultimately based on external sources such as Gene Ontology,[17] but the multilevel networks from these sources had to be adjusted, and flattened into a simple hierarchy, to highlight "druggable" protein families.

Unlike protein targets, the biological subjects used in an assay are not generally used as starting points for queries or as pivots of scientific analysis; nevertheless, quick access to biological detail beyond target is very useful when a scientist is assessing the relevance of an unfamiliar protocol or making initial analyses of data. The main problem has always been how to organize this data, which is highly variable in structure and detail, and which also presents multiple candidates among public sources for possible orderings. The common distinction between in vitro, in vivo, and ex vivo experiments provides a convenient, but overly broad, starting point for organization, so in ABCD we sharpened this distinction by developing an initial classification based on five hierarchical levels of biological organization: *whole animal*, *tissue/organ*, *whole cell*, *subcellular*, and *physical chemical properties*. This initial classification is then connected, where appropriate, to more focused subtype domains, such as *taxonomy*, *anatomical structure*, *cell type*, and *cell line*. It is at this subtype level that we have found prudent choices among authoritative external ontologies to be useful. The classification of whole animals, for instance, is derived from the taxonomy in NCBI's *Entrez*,[18] our cell line table is based on information in the *ATCC* cell line repository,[19] and our list of anatomical structures and cell types is derived, with appropriate modifications, from *MeSH*,[20] the *Medical Subject Headings* controlled vocabulary provided by the National Library of Medicine.

The third area requiring significant effort, both in the design of ontologies and in data curation, was the classification of assays as a whole and of the common experiment types used by these assays. A simple, intuitive protocol classification would make it much easier to find one's own assays or assays similar in structure designed by other teams. But since in this case no obvious external source was available, we chose to construct our ontology from the ground up. The top level is founded on the same five high level classifications used for Biological Subject above, which

we found was also a practical entry point into the assay as a whole. Underneath each of these levels we constructed a set of subtypes based on what one might call the primary "focus" of the assay. This focus could be derived from states or phenotypes associated directly with the higher level of biological organization or alternatively on measurable states or activities associated with lower levels of organization. Under *Whole Cell*, for instance, are a number of subtypes, such as *Aggregation*, *Apoptosis*, or *Proliferation*, that are based on phenotypes observable in the activity of the intact cell as a whole. Others, such as *Ion Channel Activity*, *Receptor Activity: cAMP*, or *Receptor Competitive Binding*, focus on states or activities of specific proteins within the cell that might be measured through assessing the concentration of ligands or second messengers. And below the Assay Subtype level we have constructed a set of shareable and reusable experiment types, based on common technologies and experimental methodologies. Scientists can now exploit these familiar structures to help them quickly understand and assess unfamiliar protocols. And what might be even more gratifying to the individual scientist, the documentation, for newly registered protocols, of the sometimes highly complex interrelationships between result types can now be accomplished with a simple choice of experiment type.

The above discussion makes clear the considerable effort involved, both in design of ontologies and in curation, that must be assumed at the beginning of any data warehouse project in pharmaceutical research. Much of this effort must be completed before the project can be put into production. But it is also wise to design one's warehouse procedures to support continuing curation, partly because initial efforts tend to remain unfinished even as a warehouse is being used, and also because new sources of legacy data may be folded into the project at any time. Moreover, this being science, the structure of the ontologies themselves will sometimes require adaptation. We have tried to aid this continuing effort by appointing regular curators, developing notification systems, and building ETL scripts that will treat curation tables like they would any other source, automatically updating warehouse dimensions as curation is performed. The most important strategy, however, for simplifying the classification process has been the creation of a new protocol registration system that incorporates the new ontologies within it. Now scientists supply canonical classifications when protocols are first registered, with minimum effort needed by either curator or supervisor. Assays must be typed appropriately before new protocols are accepted; biological information must be supplied; and targets must be chosen from a canonical list. If new proteins must be added to the system, then the scientists exploit a registration process that is fed by data directly from NCBI's Entrez; if genes must be connected to heteromeric protein complexes, then the scientists themselves are prompted to make this connection where possible. In those occasional cases where new experiment types must be created, the registrar meets with technical personnel directly to ensure that this is completed in a timely and accurate manner. Certainly curators will never run out of work, but this labor may become an interesting diversion, rather than a burden.

4. Third Dimension Explorer (3DX). Third Dimension Explorer (3DX) is our response to the stovepipe integration of custom-made, single-purpose applications built on a mishmash of technology platforms and development paradigms. It was designed to be a "Swiss-army knife", aimed at bringing coherence not only in the way the data are stored but also in the way they are processed, uploaded, retrieved, analyzed, and reported. Our goal was to develop a multifunctional application that was modular, extensible, flexible, programmable, customizable, elegant, and fast. In addition, the application not only had to target first and foremost the practicing scientists but also support advanced functionality for the experts.

To fulfill its intended purpose, 3DX had to support all conceivable actions associated with data: (1) retrieve; (2) collate; (3) edit; (4) process; (5) visualize; (6) model; (7) design; (8) report; and (9) upload. But to be effective in a drug discovery setting, it also had to provide "native" support for chemistry and biology. For most generic data analysis tools, this "chemical" awareness is either lacking completely or is at best an afterthought, a foreign add-on, inconveniently integrated into the main application. Although in this review we focus exclusively on the data retrieval and analysis capabilities of 3DX, it is important to note that the application can be used for tasks as diverse as sample tracking, HTS data processing, 3D modeling and simulation, and even as an electronic lab notebook. These uses are beyond the scope of this paper and will be described in subsequent publications.

Taking into account the requirement for interactive data mining and visualization and the scientists' longing for application responsiveness and performance, 3DX was developed as a Windows client using Microsoft's .NET framework.[21] Windows is the de facto industry standard for personal computing, and .NET is the platform of choice for building Windows applications, as it provides many language, framework, and integrated development environment (IDE) features that enable faster development cycles. More importantly to us, the .NET framework offered the best integration strategy for our massive body of pre-existing high-performance C++ code via a C++/CLI layer. .NET is supported by a vast development community, and there is a wealth of public and commercially available components that can be readily integrated into an application.

3DX is a table-oriented application, similar in concept to ubiquitous spreadsheet processors, most notably, Microsoft Excel. A high-level design of a 3DX document structure and file format is straightforward. A 3DX document (or project), which can be saved to and loaded from a disk file, contains a collection of tables. Each table, in turn, contains a collection of columns that represents the table's schema. A table also contains a collection of rows, representing the data held by the table. Such table-oriented data organization is very common, and there exist many file formats and software libraries supporting it. However, the specialized nature of 3DX imposed a set of requirements that severely limited our choices. The requirements include the following: (1) ability to handle documents containing a very large number of records; (2) extremely fast data access for interactive data analysis; and (3) ability to change the table schema on the fly without the user experiencing a noticeable delay (e.g., by adding, deleting, or changing columns, both programmatically and from the user interface).

With these requirements in mind, we converged on an embedded database that combines some features typically found in relational, object-oriented, and hierarchical data-

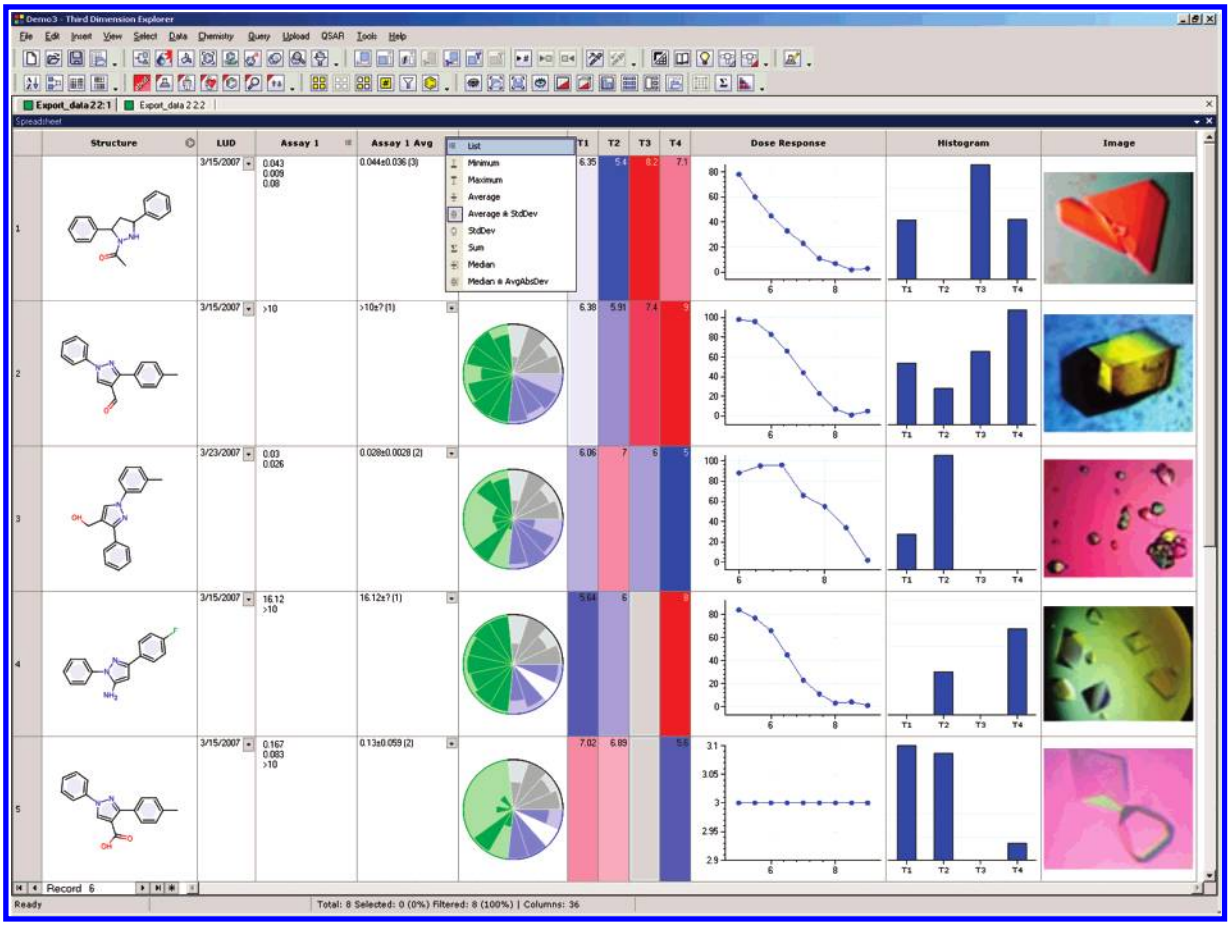

Figure 5. Representative custom cell renderers in Third Dimension Explorer (3DX). Several data types are highlighted, including chemical structures, ids, qualified (fuzzy) number lists with on-the-fly aggregation, charts, histograms, pie bar charts, and images.

bases, including support for relational joins, aggregation, serialization, and nested structures. The database engine makes use of so-called memory-mapped files to dynamically load segments of data from the supporting disk file when needed and to discard unused segments, which significantly decreases the amount of physical RAM required to work with 3DX documents. Unlike most other database systems, which are row-oriented, 3DX employs a column-oriented architecture, where all the data associated with each column are stored in contiguous memory. This minimizes the overhead from internal data rearrangements when changing table schemas and provides very fast access to data when iterating over the elements of a column. The practical limit to database size is around 1 GB, which roughly corresponds to 10 million small molecule structures encoded as SMILES strings.

In 3DX, each column must contain data of the same type. Besides conventional data types such as strings, bytes, 32- and 64-bit integers, single- and double-precision floating point numbers, and variable-size byte arrays, 3DX supports a very large and potentially infinitely extensible set of custom types, including chemical structures and substructures, "fuzzy" or qualified numbers (floating point numbers with range or uncertainty qualifiers), number lists, dates, time intervals, images, graphs, charts, histograms, forms, and many others. Column types are usually inferred automatically when a new data set is imported or can be explicitly specified. Data can be imported in a variety of ways, including input from a flat file (Excel, CSV, SD, SMILES, and SAS), direct query to a database, or action by a specific plug-in (vide infra). Much of the power of 3DX comes from its ability to associate custom cell renderers for each data type in the spreadsheet (Figure 5). For example, the cell renderer associated with chemical structures and substructures is an intelligent sketcher, which allows on-the-fly depiction of chemical graphs at extremely fast rates. The data can be edited either directly by typing in the spreadsheet or by copying-and-pasting from the clipboard. Editing of chemical structures and substructures is enabled through the chemical sketcher (Figure 6) or through copying-and-pasting from ISIS Draw[12] and ChemDraw.[22]

The application provides a number of graphical components for rendering the data in each table, including spreadsheets, 2D and 3D scatter plots, histograms, forms, heatmaps, correlation maps, SAR maps,[23] radial clustergrams,[24] pie bar charts,[25] ternary phase diagrams, and many others. An example of the richness of the UI is illustrated in Figure 7 (2D scatterplot). These viewers can be laid out and docked on the main form in virtually any conceivable arrangement and can be persistent (along with the data) in a 3DX project file (Figure 8). This file preserves the current state of the application, including its visual appearance, and can be restored at a later time or shared with colleagues across the organization. In addition, the layout of the viewers itself can be persisted as an XML file and reapplied to any table with the same schema.

3DX offers a full gamut of navigation and selection options, augmented through linked visualizations and interactive querying (filters).[26] Shneiderman[27] identified seven

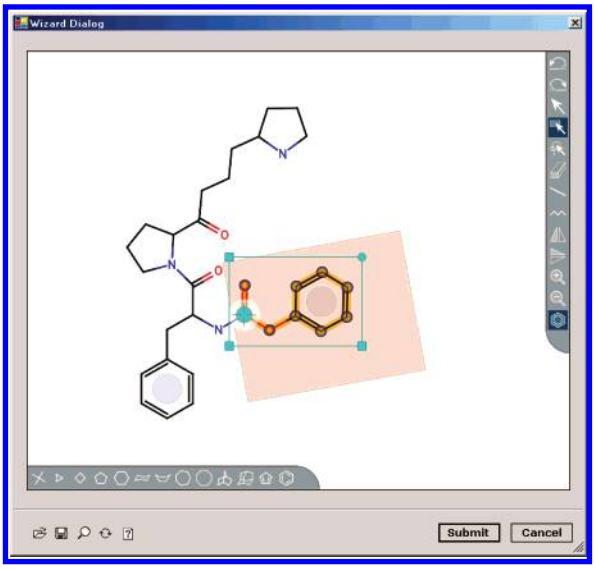

Figure 6. Chemical exact structure and substructure sketcher.

critical tasks that users try to perform on information visualization systems: (1) overview (provide an overview of the entire information); (2) zooming (zoom in on a point of interest); (3) filtering (filter out uninteresting information); (4) detail-on-demand (get details on a piece of information only when needed); (5) relationships (view relationships among pieces of information); (6) history (provide undo capabilities); and (7) extraction (extract information on a domain of interest). 3DX supports these functions through a highly finessed UI and a superb rendering engine that delivers fast antialiased graphics with subpixel accuracy.[28]

3DX was designed to be infinitely extensible. This is accomplished through a plug-in architecture that allows new functionality to be developed independently of the main application and delivered to the user either automatically or on a per-need basis. Plug-ins can be UI or non-UI driven and have full programmatic access to the 3DX core and the data, allowing them to create and remove tables, insert and remove columns, edit data, create, and (re)arrange viewers, etc. Their functionality and implementation can be extremely diverse, bringing a wealth of data retrieval, processing, analysis, visualization, and reporting capabilities to the end users, without requiring them to leave the application (Figure 9). An impressive array of powerful, chemically aware data mining tools were introduced in this fashion, including exact structure, substructure and similarity searching, structure alignment, maximum common substructure detection, chemotype classification, R-group analysis, physicochemical property calculation, combinatorial library generation, diversity analysis, SAR visualization, and many others.

The plug-in architecture was also used to provide seamless integration with the ABCD warehouse. The most widely used and technically challenging plug-in was the ABCD wizard, a graphical query builder that allows users to mine the ABCD database without requiring knowledge of SQL or its relational schema and to retrieve the results in a variety of tabular formats. Prior to ABCD, access to data was provided mostly through custom-built Pipeline Pilot protocols, which lacked a convenient user interface and were very restrictive in the types of queries that could be performed. Changes to a protocol were beyond what most scientists would be willing to master and had to be implemented by support personnel.

The ABCD query wizard was designed to be straightforward for simple needs (e.g., retrieving all compounds that contain a particular substructure or tested against a particular assay) yet powerful enough to support queries of arbitrary complexity. To achieve that goal, the query interface is separated into several tabs. The first tab is used to select assays and result types of interest, the second allows filtering of assay results by value and date and provides support for Boolean logic, the third supports various forms of chemical searching (exact structure, substructure, compound identifier, etc.), the fourth is for searching across organizational dimensions (retrieve data by team, project, etc.), and the fifth is for specifying the format of the report (e.g., assay selectivity, assay detail, compound profile, etc.), the level of aggregation for replicate measurements, and various other miscellaneous options.

This last topic, namely the aggregation of biological assay results, is a complex problem that required special consideration on the part of the software and database development teams. In the course of a discovery project, compounds may be tested multiple times in the same assay(s). This is done to confirm observed activity, to evaluate the response at different concentration levels, and to examine variability across different batches of the same substance and for a variety of other reasons. Furthermore, many activity measurements are imprecise but still useful, because repeating experiments is expensive and not always practical. For example, an IC50 of $<10\ \mu$M or $>10\ \mu$M is often used to indicate that the compound was active/inactive at 10 μM, but the range of concentrations tested were insufficient to model the full dose−response curve and establish a reliable activity value. All these factors greatly complicate the presentation and aggregation of the biological assay results, and there is no single solution that would be satisfactory in all cases. In some situations, users may wish to see all recorded measurements associated with a compound of interest, whereas in others they may wish to use a single aggregated value per compound per assay (e.g., in order to rationalize SAR across a chemical series). Both the ABCD wizard and the 3DX application itself allow the user to choose whether to use aggregation and to specify whether that aggregation should be applied at the compound, salt form, or batch level. In addition, 3DX allows the display of imprecise (or "qualified") results (e.g., $< 10\ \mu$M) in the same column as the exact values, with full support for numerical sorting, searching, and plotting. To enable this functionality, we devised a special data type that supports storing qualifiers such as $<$ (less than), $>$ (greater than), $\sim$ (approximately), and $\pm$ (plus/minus) along with the numeric values and introduced special rules for performing arithmetic operations on such qualified values.

A number of additional ABCD convenience plug-ins have been developed, which allow the user to augment an existing table by retrieving additional information from the database (e.g., structures, activities, etc.). The query used to create a table is actually stored in the table itself and can be extracted and resubmitted at a later time. This also enables one of the most striking features in 3DX, namely the ability of tables to self-correct and self-update when new information is added to the database.

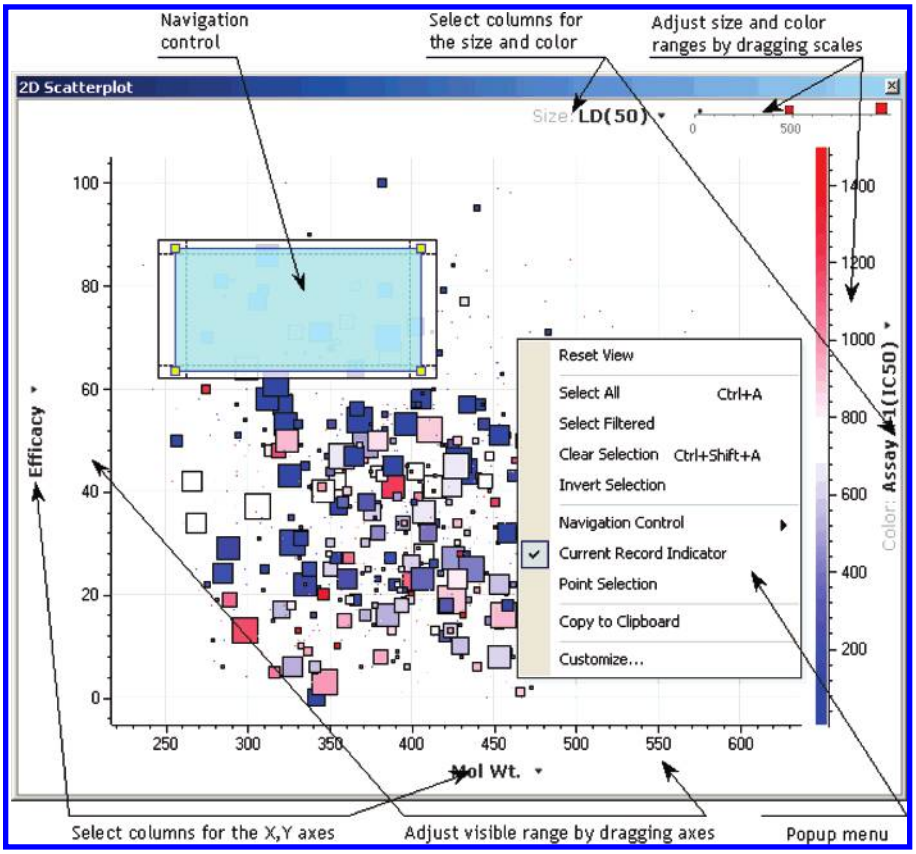

Figure 7. Graphical user interface of 2D scatterplot viewer.

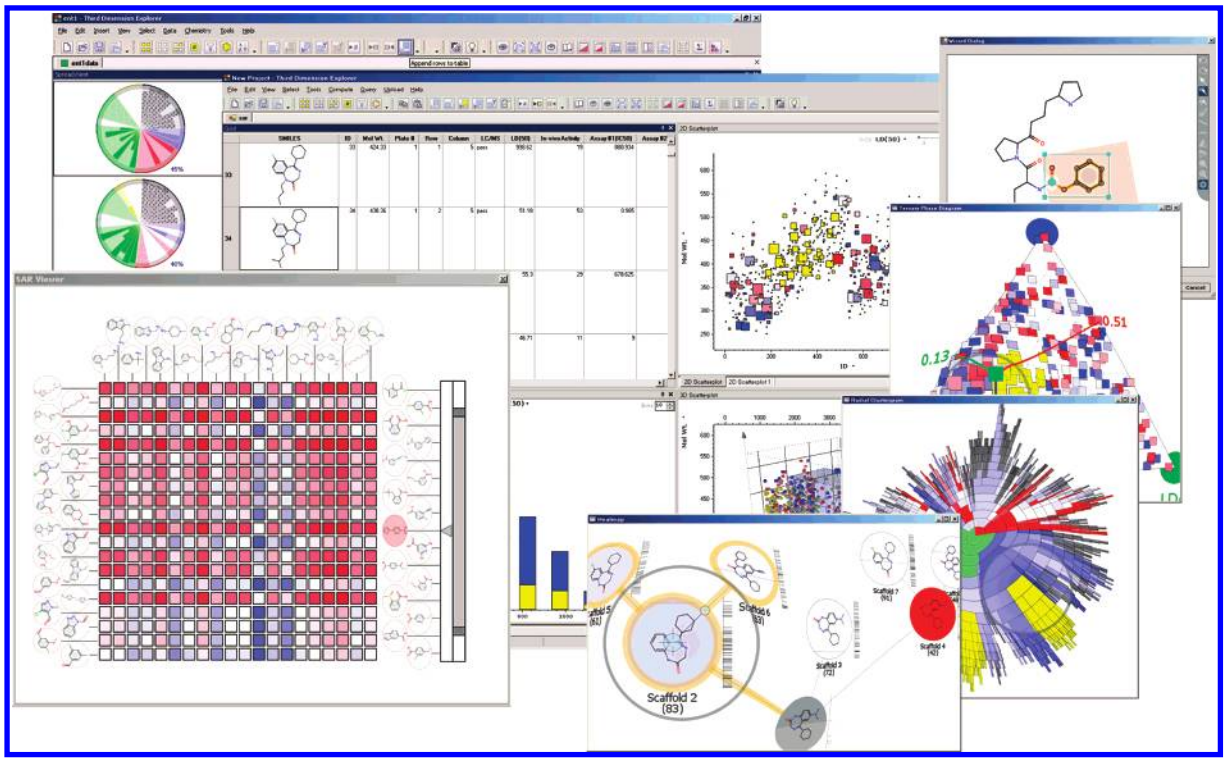

Figure 8. 3DX representative viewers and layout management.

Numerous other functionalities are provided, including the ability of the application to self-update, collect usage statistics, and allow control from external applications. Finally, 3DX provides transparent access to the ABCD Workspace (vide infra), a collaboration environment which allows users to share queries, documents, layouts, form, and reports and provides a discussion forum for asking questions, exchanging ideas, and requesting support.

In addition to providing good usability and relevant functionality, we placed particular emphasis on performance and aesthetic appeal and paid meticulous attention to design patterns, color schemes, and the overall consistency of the

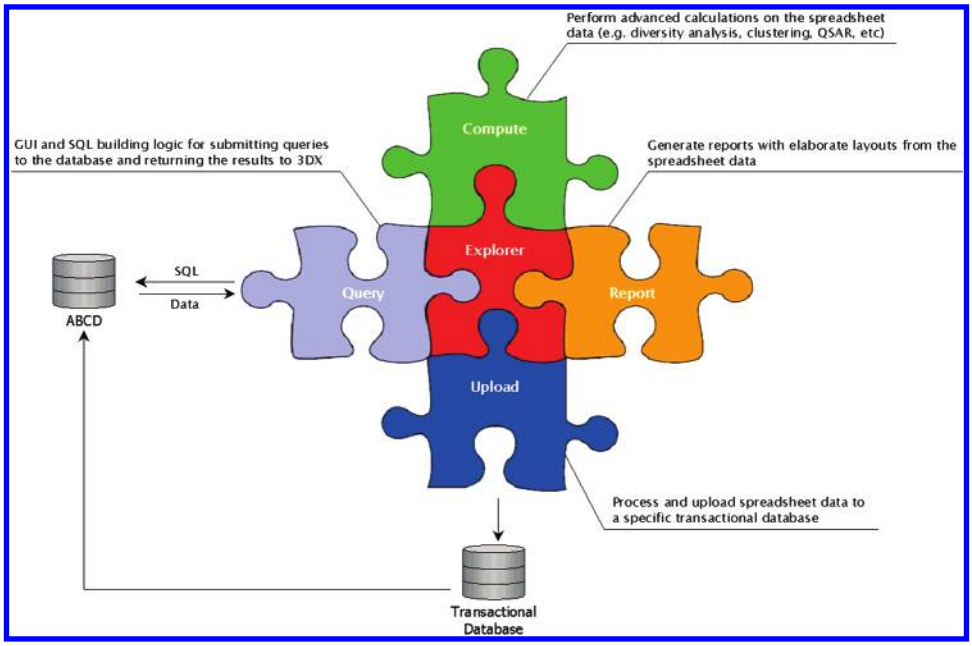

Figure 9. Plug-in architecture of 3DX.

look-and-feel. Drawing from the extensive experience in developing large-scale chemoinformatics and visualization software and capitalizing on the DirectedDiversity[10] platform developed at 3-Dimensional Pharmaceuticals, Inc., we were able to build the first prototype version of 3DX in ~6 months and deliver the first production version to J&J users ~12 months later.

5. Workspace. The third component comprising the ABCD solution is the Workspace, a Web portal dedicated to delivering and managing ABCD information. The site provides news, announcements, useful links, training materials, documentation, and other support documents related to the project. It is also designed to manage files such as queries, forms, layouts, and reports generated by 3DX. Users can save these files directly into the workspace from within the 3DX application through a modified Windows file dialog. These files can then be shared with other members of the project team, the research site, or the entire organization. By using Windows authentication, the workspace enforces security automatically, so that users can access only the documents that they are authorized to see.

The most popular feature of the Workspace is the discussion forum. Because the users of ABCD are scattered across different time zones, the forum provides a convenient mechanism for users to communicate with each other at their leisure. The forum is open and accessible to everyone at J&J. Users can search for questions and answers already posted and discussed, thus minimizing the volume of work for the support group. They can also subscribe to any topics that they are interested in and receive e-mail updates when new posts are available. The forum has been extremely valuable in collecting feature requests and bug reports, discussing technical and scientific issues, and sharing experiences and ideas.

PROJECT MANAGEMENT

ABCD was designed to be generally useful and comprehensive yet flexible enough to accommodate each site's individual needs. Its successful and timely deployment required sound execution in every aspect of the project, including design, development, testing, piloting, installation, training, and support. To facilitate project management, the project was divided into three main work streams: (1) data, (2) applications, and (3) project management.

The application stream was responsible for the development of Third Dimension Explorer (3DX), the Protocol Registration System, and the Workspace. The team maintained a high degree of agility, placing less emphasis on collecting detailed user specifications and relying much more heavily on fast prototyping and rapid response to user feedback, a strategy that proved highly successful in practice.

The database stream was responsible for establishing an efficient dimensional schema for the data warehouse, the historical and incremental ETL processes for extracting, transforming, and loading the data from the primary data sources, the technology for database versioning, and the framework for deploying and managing multiple warehouse interfaces. This process required careful study and understanding of the data models of the primary data sources and the scientists' interaction with and use of those systems. A particularly challenging aspect of the project was dealing with the legacy data and out-of-date models while simultaneously advancing the design of the new database. The construction of the data warehouse involved the curation of thousands of different biological assays, result types, molecular subjects, biological subjects, and many other data dimensions, a tedious and labor intensive job that required frequent communication between technologists and data owners across multiple time zones, often in late night or early morning hours.

Finally, the project required extensive user training and support. The ABCD team prepared training materials, trained more than 900 scientists at all J&JPRD sites on how to use the new system, organized brown-bag sessions, and answered countless questions from users through face-to-face contacts, e-mail, and the ABCD forum.

BENEFITS

ABCD is now used routinely by more than 1000 research scientists around the world. In the ongoing quest for efficiency and savings, ABCD will allow a more effective selection of complementary external software solutions and the retirement of several internally developed legacy systems, resulting in significant savings in maintenance and integration costs. Most importantly, the research community was given a toolset that is actively supported and rapidly expanded and offers capabilities that are not available in any commercial software.

EARLY OBSERVATIONS AND USAGE PATTERNS

3DX and ABCD were designed to capture usage statistics, allowing us to understand usage patterns down to the organization, site, user, time, and function levels. As of this writing, there have been at least 980 recorded users of the system spread across 16 sites and/or J&J operating companies, with a median of 171 distinct users per day (measured over the first quarter of 2007). More specifically, there have been 690 distinct users over the past 30 days, 821 over the past 60 days, and 906 during the last quarter, indicating both broad and repetitive usage. The number of application sessions in a typical day range from 300 to 600. Success has also been validated through the actions of our computational scientists who are rewriting their pipelining scripts to use the ABCD data warehouse as the primary data source.

Qualitatively, we observe that the most active users fall into three distinct categories: computational and enabling technology scientists, project champions and data managers, and technology-savvy scientists. General, exploratory querying is practiced relatively rarely and is typically used to identify strategic opportunities and assess research productivity. By far the greatest use of ABCD is in managing project data. The most common querying and reporting is focused on accessing compounds and assay results associated with a given project and searching the chemistry space around active molecules. A small number of individuals within each project tend to assume responsibility for accumulating and nurturing this holistic view of project data. Indeed, our European project teams have developed a "data manager" function and have appointed key scientists to that role to ensure that the most complete and appropriate view of the data is used to drive project decisions.

Another observation is that most users expect *all* the data to be available in the system, otherwise they will resort to the old-fashioned methods for creating consolidated views. The first release of ABCD focused on in vitro data and made it broadly and readily accessible. However, the lack of in vivo results has slowed broader adoption of the system. In vivo data are one of the foci of the upcoming release of ABCD.

Training and usage obviously go hand-in-hand. Our release strategy has included a variety of training tactics: general introductory sessions, hands-on training sessions, birds-of-a-feather sessions, and individualized training. The usage statistics has also allowed us to identify pockets of low usage and tailor training and support processes to address specific needs and requirements. While all these methods are both useful and effective, we have found that there is no substitute to working directly with the discovery teams. Helping scientists see how these tools can be used for their specific needs emboldens them to explore them more broadly.

CONCLUSIONS

ABCD represents a new vision for discovery informatics that goes far beyond the loose "plumbing" of data systems and applications that one typically encounters in the pharmaceutical industry. The program's widely acknowledged success can be attributed to several factors: (1) driven, developed, and managed by scientists; (2) built by J&JPRD for J&JPRD (the product was made to fit the company and not the other way around); (3) close partnership between technologists and the customers; (4) rapid feedback cycles; (5) strong technical leadership and coordination; (6) strong scientific direction and oversight; (7) strong management support; and (8) outstanding technical talent with domain expertise in life sciences and drug discovery. Principal among those factors was the dynamic interplay between scientists and technologists and the ability to influence each other. In the words of Lajiness: "what is asked for is often not what is wanted, which is often not what is needed".[29] Indeed, the ultimate success of such projects rests upon the ability of technologists to take rough ideas and convert them into elegant, workable systems. Such systems rarely resemble their initial specifications. They undergo dynamic adaptation. This adaptation requires a development team with advanced scientific training, embedded in a living, breathing research organization. The ABCD "experiment" reinforces the emerging belief that the best informatics systems are developed by in-house experts, who have a vision and the ability to implement it.

ACKNOWLEDGMENT

We thank Michael Jackson, Roger Bone, Didier De Chaffoy, John Barbano, Todd Jones, Jan Hoflack, and David Neilson for their unwavering support of the ABCD initiative. We also thank the Scientific Advisory Team led by Dennis Hlasta (initially) and Peter Connolly (currently) for their sage counsel, scientific and user validation, and strong advocacy for and to their colleagues. We thank Serge Masyn for driving the ABCD assessment phase and envisioning how our design aligned with the original ABCD requirements and for providing, along with Brian Johnson, program management oversight. Joseph Brenner deserves special acknowledgment for bringing right-sized program management to an unruly group of scientific developers. Greg Rusin was indispensable in his ability to build and support the novel and continually evolving ABCD server architecture. Keith McCormick, Jim Gainor, and Brian Wegner are acknowledged for their efforts in designing and executing a successful training and support strategy. We are particularly grateful to Mark Seierstad from the La Jolla CADD group for evangelizing the use of ABCD and providing hands-on training and support. We also thank our colleagues in Infrastructure Services, especially Jim Fortunato, Francis Heylen, Joel Kaufman, Pam La Rocca, and Greg Plante, for listening to where we wanted to go technologically, understanding where we needed to go, and guiding and driving us towards that goal. We had a number of contractors who contributed substantially to the project but would like to specifically mention Guy Mahieu, whose work on forms had a significant impact on their usability and widespread

adoption. All views expressed herein are those of the individual authors.

REFERENCES AND NOTES

(1) Claus, B. L.; Underwood, D. J. Discovery informatics: its evolving role in drug discovery. *Drug Discovery Today* **2002**, *7*, 957−965.

(2) Kreusel, D. From raw data in the laboratory to information availability in the enterprise. *Drug Discovery World* **2001/2002**, Winter, 70−74.

(3) Hagadone, T. R.; Lajiness, M. S. Capturing chemical information in an extended relational database system. *Tet. Comp. Meth.* **1988**, *1* (3), 219−230.

(4) Hagadone, T. R.; Lajiness, M. S. Integrating chemical structures into an extended relational database system. In *Chemical Structures 2: The International Language of Chemistry*. Proceedings of the 2nd International Conference, June 3−7 1990, Noordwijkerhout, The Netherlands; Warr, W. A., Ed.; Springer-Verlag: Berlin, 1993; pp 257−269.

(5) Rojnuckarin, A.; Gschwend, D. A.; Rotstein, S. H.; Hartsough, D. S. ArQiologist: An integrated decision support tool for lead optimization. *J. Chem. Inf. Model.* **2005**, *45*, 2−9.

(6) Cho, S. J.; Sun, Y.; Harte, W. ADAAPT: Amgen's data access, analysis, and prediction tools. *J. Comput.-Aided Mol. Des*. **2006**, *20* (4), 249−261.

(7) Walters, P. VERDI: An extensible cheminformatics system. *Daylight User Group Meeting '02*, 2/27/2003.

(8) Hassan, M.; Brown, R. D.; Varma-O'Brien, S.; Rogers, D. Cheminformatics analysis and learning in a data pipelining environment. *Mol. Diversity* **2006**, *10* (3), 283−299.

(9) http://blogs.netindonesia.net/kiki/archive/2006/02/28/8822.aspx (accessed month year).

(10) Agrafiotis, D. K.; Lobanov, V. S.; Salemme, F. R. Combinatorial informatics in the post-genomics era. *Nat. Rev. Drug Discovery* **2002**, *1*, 337−346.

(11) Uehling, M. D. ABCD: How to spell discovery. *BioIT World Magazine* **2004**, *6*. Available online at http://www.bio-itworld-.com/archive/061704/discovery.html (accessed June 20, 2006).

(12) http://www.mdli.com (accessed month year).

(13) Pantoliano, M. W.; Petrella, E. C.; Kwasnoski, J. D.; Lobanov, V. S.; Myslik, J.; Graf, E.; Carver, T.; Asel, E.; Springer, B. A.; Lane, P.; Salemme, F. R. High-density miniaturized thermal shift assays as a general strategy for drug discovery. *J. Biomol. Screen.* **2001**, *6* (6), 429−440.

(14) Kimball, R.; Ross, M. *The data warehouse toolkit: the complete guide to dimensional modeling*, 2nd ed.; Wiley Computer Publishing: 2002.

(15) Stein, L. D. Integrating biological databases. *Nature Rev. Genet.* **2003**, *4*, 337−345.

(16) http://www.gene.ucl.ac.uk/nomenclature/ (accessed month year).

(17) http://www.geneontology.org/ (accessed month year).

(18) http://www.ncbi.nlm.nih.gov/Entrez/ (accessed month year).

(19) http://www.atcc.org/common/technicalInfo/TechLit.cfm (accessed month year).

(20) http://www.nlm.nih.gov/mesh/ (accessed month year).

(21) http://www.microsoft.com/net/basics.mspx/ (accessed month year).

(22) http://www.cambridgesoft.com (accessed month year).

(23) Agrafiotis, D. K.; Shemanarev, M.; Connolly, P. J.; Farnum, M.; Lobanov, V. S. SAR maps: A new SAR visualization technique for medicinal chemists. *J. Chem. Med.*, in press.

(24) Agrafiotis, D. K.; Bandyopadhyay, D.; Farnum, M. Radial clustergrams: visualizing the aggregate properties of hierarchical clusters. *J. Chem. Inf. Model*. **2007**, *47*, 69−75.

(25) Howe, T.; Mahieu, G.; Tabruyn, T. *Drug Discovery Today* **2007**, *12*, 45−53.

(26) Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software* **1994**, *11* (6), 70−77.

(27) Shneiderman, B. *Designing the user interface: strategies for effective human-computer interaction*, 3rd ed.; Addison-Wesley Publishing Co.: Reading, MA, 1998.

(28) Shemanarev, M. The Anti-Grain Geometry Project. http://www.antigrain.com (accessed June 1, 2006).

(29) Lajiness, M. S. Cheminformatics integration to support drug discovery. 9th CHI Conference on Cheminformatics: Data Driven Decisions, Philadelphia, PA, 2005.

CI700267W