Enhancing the Reagent Selection Workflow via Real-Time Vendor Inventory

David S. Nirschl,*,† Mark A. Hermsmeier,† Greg Silber,‡ Janette V. Logan,† Rick Hammar,§ and Donna L. Young¶

Bristol-Myers Squibb Company R&D, Route 206 and Provinceline Road, Princeton, New Jersey 08543, Intelligent Systems, Inc., 812 Interchange Boulevard, Newark, Delaware 19711, ASDI, Inc., 601 Interchange Boulevard, Newark, Delaware, 19711, and Bristol-Myers Squibb Company R&D, 5 Research Parkway, Wallingford, Connecticut 06492

Received October 3, 2008

In an effort to increase the use and effectiveness of parallel array synthesis in the drug discovery process, we have undertaken a project to evaluate and optimize the array synthesis workflow using Lean techniques[1,2] as a guiding principle. A recent analysis[3] of the workflow, from reagent selection through synthesis, purification, and compound registration, identified reagent sourcing as a major bottleneck, where reagent sourcing is defined as the process of acquiring appropriate array building blocks within a given budget and time frame. Historically, we have identified potential vendors of chemical reagents via utilization of commercially available reagent databases[4−6] that consolidate the catalogs of tens or hundreds of chemical suppliers from around the world. The use of these consolidated databases for reagent sourcing is problematic, however, on two fronts. First, there are a significant number of reagents in these databases for which pricing is not available, and in these cases, it becomes necessary to contact individual vendors either via phone or Internet. Second, for those reagents that do have pricing information in the database, that information may be significantly out of date because of the several months it may take for the database vendor to complete the data aggregation, mail the database to the subscriber, and additional time for the subscriber to update to the latest version. The net result is that the data could easily be sixth months or more out of date.

Considered in terms of Lean manufacturing techniques, the process of attempting to acquire reagents that, for one or more reasons, are not actually commercially available constitutes pure waste. Reagents may be considered unavailable if the manufacturer is out of stock, has discontinued the product, or if the pricing terms are not acceptable. Unfortunately this information is typically only gathered *after* the reagent has been selected for inclusion in an array, potentially via a detailed QSAR analysis or library design, and only at the expense of significant human resource. Rather than address the issue by adding resources to handle reagent sourcing, we chose to focus on the earlier step of reagent selection, with the idea of providing researchers more up-to-date information regarding reagent availability during the reagent selection process.

Common sense would dictate that the most detailed information about reagent availability would come from internal sources such as a local chemical inventory, and indeed arrays for which reagents are chosen exclusively from internal collections are typically completed with the shortest turnaround times.[7] While this might indicate that a potential business solution would be to create a massive internal inventory of reagents, this approach would clearly be unrealistically expensive to implement, especially considering that the universe of reagents is vast. Notably, dealing with the logistics of inventory management is not unique to the chemical sciences, and several examples of companies in other industries are available for study to provide guidance in this area.

One of the hallmarks of companies that excel in direct-to-consumer sales is inventory management: the concept of reducing inventory levels of either parts or final products while still maintaining quantities sufficient to meet current consumer demand. For example, Wal-Mart is well-known for having a sophisticated inventory system that ships replacement products from centralized warehouses immediately following the purchase of the same item by a customer; in recent years Wal-Mart has been creating an even more streamlined system by asking suppliers to tag each pallet of products with a radio frequency ID tag.[8] Dell Computer has worked with their computer parts suppliers such that they have only two-to-four days of parts inventory on hand at any given time.[9] Wal-Mart and Dell have succeeded in part by partnering with their suppliers to create just-in-time inventory management systems, and these partnerships are driven by the sharing of information between the companies. Given the clear importance of supplier relationships in the broader business world, we felt that developing relationships with reagent suppliers should be a central prong in the overall strategy to improve our reagent management capabilities.

We chose to partner initially with a relative newcomer to the reagent supply business, ASDI, Inc., for two primary reasons. First, ASDI had implemented a business model that provided custom-weighed reagents with rapid turnaround (in the U.S., typically 24−48 h from order to delivery) in packaging specified by the customer, rather than providing reagents only in prepackaged amounts. Second, in collaboration with Intelligent Systems, they had developed a sophisticated real-time inventory management system that, at any moment, could provide the exact amount of all ASDI reagents available for purchase, a collection numbering approximately ten thousand. As shown in Figure 1, the overall concept of the collaboration was to funnel inventory

* To whom correspondence should be addressed. E-mail: david.nirschl@bms.com.
† Bristol-Myers Squibb Company, Princeton, New Jersey.
‡ Intelligent Systems, Inc.
§ ASDI, Inc.
¶ Bristol-Myers Squibb Company, Wallingford, Connecticut.

10.1021/cc8001603 CCC: $40.75
Published on Web 11/17/2008

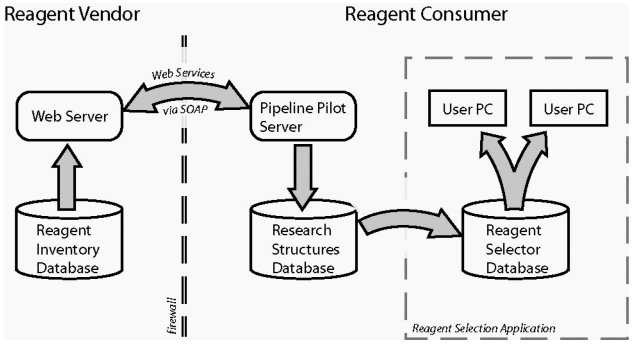

Figure 1. Schematic representing the flow of data from the vendor's reagent inventory database to the end users at the research institution.

information from the vendor into our own databases via Web services.[10] The three main components to the data flow include the transfer of data via web services, the analysis of that information via Pipeline Pilot scripts, and the eventual incorporation of that data into databases, which are searchable by researchers; each of these steps will be discussed in detail below.

The overall system architecture is designed to deliver both aggregated vendor catalogs and real-time inventory information to researchers, who in our implementation access the reagent data via a commercially available reagent searching and selection application (MDL Reagent Selector[11]). As shown in Figure 1, the real-time vendor data is transferred from the reagent vendor's inventory database to an intermediary research structures database, accomplished via a SOAP-based Web service,[12] described in detail below. The aggregated reagent data is periodically transferred to the Reagent Selector database against which the end-user application makes queries. Ultimately, the researcher accesses data that has aged based on three factors: the timeliness of the data in the vendor database, the frequency at which the Web service is polled, and the frequency at which the data is transferred to the Reagent Selector database. In practice, we conservatively estimate that users are accessing data that is less than two hours old.

In essence, the system architecture requires that the vendor's reagent inventory database, including catalog number, structure, price, and amount available, be duplicated at the consumer company. The Web service was designed to accommodate this need with a relatively straightforward programmatic interface, minimizing the amount of data that must be transferred on a daily basis. The Web service essentially consists of three queries: list all available catalog items, list catalog numbers for which information has changed since the last update, and provide detailed information about a specific catalog item. The concept is that only very infrequently[13] is the entire catalog transmitted, accomplished by requesting the full list of catalog numbers followed by a request for detailed information about each individual catalog number. On a daily basis, the databases are synchronized by maintenance of a database "change log", which keeps track of all the changes to the database along with a virtual pointer which tracks which changes have been transmitted to the consumer. See Figure 2 for an overview of the methods and objects defined as part of the Web service.

As an example of how the database synchronization is accomplished, consider the situation in which the vendor adds ten new chemical products to its inventory. The vendor database is updated with the new products, and the database change log is appended with the new catalog numbers. When the consumer queries the Web service for the five most recent database changes, the first five new catalog numbers are returned as part of a complex data type. The catalog numbers can then be used as parameters to query the Web service, one at a time, for the detailed product information such as structure, amount available, and pricing information. The web service keeps track of which records the consumer has seen as part of the update process, so when the consumer requests another batch of change records, the second set of five catalog numbers is returned. To prevent transmission errors or software crashes from corrupting the synchronizing process, the familiar concept of database "transactions" are utilized by the Web service.

From the perspective of the vendor providing the data (ASDI), implementation of the Web service involved adding a layer of software onto a relatively sophisticated real-time inventory system. Real-time inventory data was already available on the vendor's Web site, and there was already a change log in place, thus changes to the underlying data architecture were not necessary. However, the change log items transmitted via the GetPendingUpdates Web service call must be exclusive on a per-user basis, so that each different registered user can maintain a valid, synchronized version of the database. To support this feature, a separate database table was created with the purpose of tracking which database changes had been received by each user.

From the data consumer standpoint, an application was built from scratch using the commercially available data pipelining[14−16] tool Pipeline Pilot.[17] Two example data pipelines are shown below for illustrative purposes. In Figure 3, up to four different SOAP calls are invoked to collect the most recent updates from the web service. The SOAP action which is the heart of this pipeline is *GetPendingUpdates*, which requests the most recent changes to the database. The succeeding pipeline component counts how many items were actually received from the web service, and that count is provided back to the web service as part of the "end transaction" call. If there were any problems with the transmission of the data, the transaction can be rolled back, essentially meaning that the same information will be transmitted during the next iteration of the pipeline. Notably, our production environment log files indicate that it has never been necessary to roll back a transaction.

Figure 4 details the portion of the data pipeline which interrogates the data received from the web service and processes it according to the type of information that was received. If the reagent's catalog number is already in the database, only the amount available needs to be updated. However, if the catalog number is not in the local database, another SOAP call retrieves the structure and related data from the web service, performs error-checking, and finally inserts the new catalog number and structure into the database. This pipeline can be customized as necessary to ensure that the data retrieved from the vendor meets the

Method Details	Complex Type Definitions
GetCompoundListing IN: [none] OUT: array of compoundID (string)	compoundInfoStructure (complex type) catalogNumber (string) compoundName (string) pWght (double) tWght (double) sWght (double) mfcdNumber (string) pricePerGram (double) chemicalFormula (string) structureMDLMol (string) quantityInGrams (double)
GetCompoundInfo IN: compoundID (string) OUT: compoundInfoStructure (complex type)	
GetPendingUpdates IN: count (integer) OUT: array of updateRecord (complex type)	
BeginUpdateTransaction IN: [none] OUT: result (boolean)	updateRecord (complex type) compoundID (string) quantityInGrams (double) updateType (enumeration)
EndUpdateTransaction IN: countReceived (integer) OUT: result (boolean)	updateType (enumeration) inventory (string) quantity (string)
RollbackTransaction IN: [none] OUT: result (boolean)	

Figure 2. Overview of the methods and objects defined as part of the reagent inventory Web service.

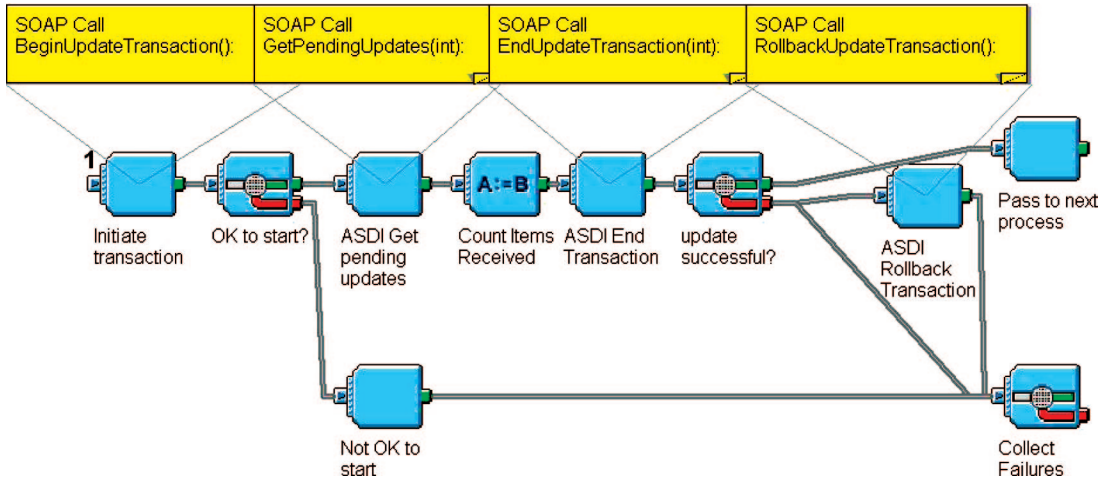

Figure 3. Example Pipeline Pilot script for retrieving inventory information via the Web service.

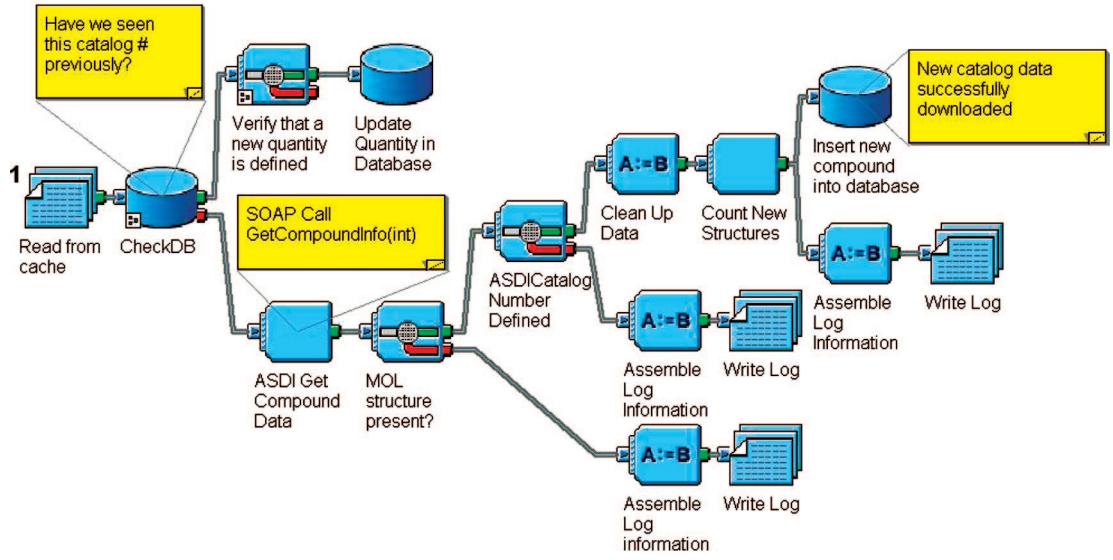

Figure 4. Example Pipeline Pilot script for interrogating the data received from the Web service and depositing it into a local database.

requirements of the local database. For example, salts or solvates can be eliminated from the structure if desired.

Finally, the data accumulated in the Research Structures Database must be copied over to the Reagent Selector

database so that it can be accessed via the Reagent Selector client application. This is accomplished via a repetitive, scheduled Oracle query from the database hosting the Reagent Selector database, run once per hour.[18] The data integration is seamless, with both real-time and nonreal-time data displayed side-by-side in an integrated user interface.

The system architecture as discussed here relies on an intermediary data repository, shown in Figure 1 as the Research Structures Database (RSD). While it would have been possible to implement a similar overall workflow without the use of the RSD, there are several advantages to our finalized design. First, for application development purposes, we were able to modularize the workflow and assign different development teams to each module. Second, we can install periodic updates to the Reagent Selector database and quickly append the ASDI data from the RSD rather than via the web service. The RSD can be used to provide or support convenient functionality such as identifier translation tools (e.g., convert a list of ASDI catalog numbers into a different vendor's catalog number) and library design software. Finally, the modular design will allow us to replace MDL Reagent Selector without needing to rewrite the Pipepine Pilot scripts.

While the commercial software MDL Reagent Selector was utilized in our implementation, it should be noted that the objective of providing integrated real-time vendor inventory directly to chemist end-users can be accomplished using a wide-variety of software tools, both commercial and custom. The best-case scenario is to integrate the inventory information directly into whatever application chemists are already using on a daily basis. One limitation to our current implementation is that chemists utilize MDL Reagent Selector to design their arrays but an entirely different piece of software for nonarray reagent searches and requests. We intend to rectify that limitation in the near future by replacing those packages with software that will minimize the differentiation between array-based and nonarray-based reagent logistics.

Despite the current implementation being limited to a single vendor, we believe that this solution has positively impacted our library design process by providing valuable information to chemists at design time. When real-time inventory data is utilized during the design, the overhead of reagent logistics is reduced. We believe that expanding this solution to include several additional key vendors, including those providing prepackaged reagents, would have a profound impact on our processes. As our organization looks to maintain or increase utilization of library synthesis while at the same time limits resources dedicated to reagent logistics, solutions such as the one described here will become increasingly important.

We have partnered with a key reagent vendor to provide access to real-time reagent inventory. The real-time data integrates seamlessly into a reagent selection application alongside nonreal-time availability information. We have found that access to this real-time data significantly simplifies the presynthesis portion of the array synthesis workflow, as it is no longer necessary to assume that some percentage of selected reagents will not be commercially available or within a desired price range. Looking forward, we hope to either expand the number of vendors who provide us with real-time data or partner with a third-party who can provide aggregated real-time reagent data.

Acknowledgment. The authors thank Harold N. Weller, James Gill, and Steven L. Hoffman from BMS, and Alan Blize and Ron Paloni from ASDI for their support of this project.

References and Notes

(1) Womack, J. P.; Jones, D. T.; Roos, D. *The Machine That Changed the World: The Story of Lean Production*; HarperCollins: New York, 1991.
(2) Womack, J. P.; Jones, D. T. *Lean Thinking*; Free Press: New York, 2003.
(3) Weller, H. N.; Nirschl, D. S.; Petrillo, E. W.; Poss, M. A.; Andres, C. J.; Cavallaro, C. L.; Echols, M. M.; Grant-Young, K. A.; Houston, J. G.; Miller, A. V.; Swann, R. T. *J. Comb. Chem.* **2006**, *8*, 664–669.
(4) MDL Information Systems, Inc., and Symyx Technologies, Inc., Santa Clara, CA.
(5) CambridgeSoft Corportation, Cambridge, MA.
(6) Accelrys Software, Inc., San Diego, CA.
(7) Nirschl, D. S. Unpublished work, 2006.
(8) http://informationweek.com/showArticle.jhtml?articleID=172301246. Accessed October 2007.
(9) Dell, M.; Fredman, C. *Direct from Dell*; HarperCollins: New York, 1999.
(10) Curcin, V.; Ghanem, M.; Guo, Y. *Drug Discovery Today* **2005**, *10*, 865–871.
(11) *MDL Reagent Selector*, client version 1.5; Symyx Technologies, Inc.: Santa Clara, CA, 2004.
(12) For security purposes, the service runs over 128-bit encrypted HTTPS.
(13) We believe that quarterly updates are more than sufficient.
(14) Hassan, M.; Brown, R. D.; Varma-O'Brien, S.; Rogers, D. *Mol. Diversity* **2006**, *10*, 283–299.
(15) Kostrowicki, J.; Peng, Z.; Kuki, A. *Mod. Drug Discovery* **2001**, *4*, 94–97.
(16) Brown, R. D.; Varma-O'Brien, S.; Rogers, D. *QSAR Comb. Sci.* **2006**, *25*, 1181–1191.
(17) *SciTegic Pipeline Pilot*, client version 6.0.2, server version 6.1.1; Accelrys, Inc.: San Diego, CA, 2006.
(18) Young, D. L. Reagent Selector at BMS. Presented at the Elsevier MDL U.S. User Conference, Jersey City, NJ, May 2006.

CC8001603