Reprinted from PHARMACEUTICAL ENGINEERING®
The Official Magazine of ISPE
September/October 2012, Vol. 32 No. 5
www.PharmaceuticalEngineering.org

This article presents ideas, concepts, and prototype experience on how to bring products faster to market through a more structured and integrated management of product, process, and analytical data based on proven industrial standards (S88/S95) and data warehouse technology.

Bringing New Products to Market Faster

by Adam Fermier, Paul McKenzie, Terry Murphy, Leif Poulsen, and Gene Schaefer

Introduction

Large pharmaceutical organizations are currently being pressured to increase the efficiency and effectiveness of their business in terms of leveraging internal and external resources to deliver faster on design, execution, analysis, and reporting. Inconsistency and sometimes a complete lack of structure around key business processes has led to intensive allocations of resources spent on last minute efforts to complete regulatory filings and technology transfers on time. Inherent in these efforts is often a misplaced emphasis on gathering primary data rather than its transformation into information and knowledge and its subsequent analysis. Thus, these efforts are typically the result of an information "push" through the corporation as opposed to an information "pull" driven by a well-coordinated knowledge management strategy. The root cause of this push versus pull in the pharmaceutical industry is the fundamental lack of a scalable knowledge management strategy that can handle the lifecycle management of a novel medicine end to end.

Building a solid knowledge management strategy for the industry has many requirements and challenges. Fortunately, other data driven industries have tackled the knowledge management challenge by adopting industrial standards for batch execution and planning/modeling (i.e., ISA S88/S95 compliant).[1-5] However, the problem is how to assemble and contextualize the data scattered throughout many systems. Compounding this problem is that these systems are often a mix of validated and non-validated systems; therefore, it is imperative that the strategy encompasses a modular and scalable approach to the integration of information contained within these systems.

Data warehousing is a common informatics approach that can help meet the requirements set forth above where the data warehouse has a data model conforming to the standards. Bringing these two concepts of a data warehousing strategy in combination with what some have called recipe-based execution will enable the assembly of data rich systems into a common system defined here as a "Recipe data Warehouse" (RW). The RW strategy will allow the organization of data across multi-source execution systems and will drive more data rich decisions for products in a timely manner. This will ultimately lead to increasing the quality, capability, and capacity of the organization to execute our ultimate business deliverables: New Drug Application (NDA), Biologic License Application (BLA), technology transfer, and delivery of therapeutics to patients.

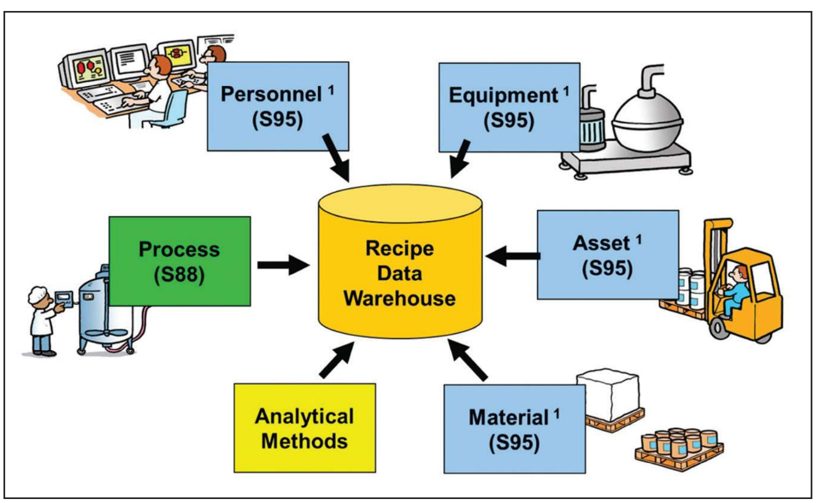

Figure 1. S88 & S95 recipe objects to be managed by recipe data warehouse.

Defining the Strategy

S88/S95 standards provide definitions around people, materials, and equipment as well as procedural models on how these are combined to make products *Figure 1*. Typically, quality monitoring methods are not well defined in these standards; however, the associative resulting data from these methods could be easily stored. The core of the recipe data warehouse is based on a well structured and tested data model which must:

- Support the business objectives/planned system functions (S95)
- Provide modeling of recipes including specification of processes, personnel, equipment/assets, materials, and analytical methods (S88/S95)
- Align with current/best practice in pharmaceutical manufacturing, i.e., development of small and large molecule drugs
- Align with relevant S88/S95 models
- Include modeling of analytical methods/data, which is not well defined in S88/S95
- Adopt the S88/S95 object oriented thinking (use object classes and instances)
- Adopt the S88/S95 expandability/collapsibility concept (use recursive relations)
- Allow for stepwise development of a recipe based on recipe building blocks (use reference or inheritance)
- Enable ad-hoc addition of analytical measures that may initially not have been defined in the recipe
- Provide ability to capture in process or release data (discrete and continuous)

Putting all these requirements into a centralized recipe data warehouse can be daunting, but well defined strategies in data warehousing can help tremendously.[6] The strength of combining these two strategies is the common modular approach. The data warehousing strategy breaks the information management into four unit operations as outlined in Figure 2. Data source systems provide all source data for the recipe warehouse and in this strategy validation and compliance issues, including change control are addressed in these source systems. The data staging area is a complex, yet simplified manner to help conform to the S88/S95 data standard and designed to optimize data writing speeds. The data presentation area now pre-aggregates data from the data staging area designed to optimize read speeds. The data access tools provide a means to deliver standard reports as well as ad-hoc

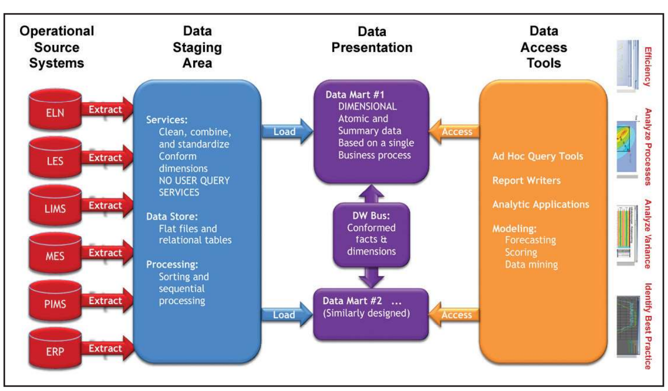

Figure 2. Overview on information management – pulling data from source systems.

to advanced trending/analysis. Like the data source systems, the data access tools are meant to be non-system specific to provide the modularity and flexibility required.

It is important to note that the data structure must be applied and/or understood within these data source systems to effectively leverage this strategy. The transformation from the data source systems into the recipe data warehouse is called Extraction-Translate-Load (ETL). It arguably is the most critical portion of the strategy as it will ultimately be required to handle the diversity of data models in the source systems and conform to one common system independent model.

Recipe Development Process – Driving Standards and Flexibility

The overall business objective is to bring new products faster and more efficiently to the market. To do this, the complete development process from discovery to commercial manufacturing of new drugs must be standardized and based on common recipe data models and tools. A key driving motivator behind this strategy is presented in Figure 3 where the organization prepares in a proactive nature to perform technology transfer between each critical clinical milestone. The strategic modification enables more flexibility to the organization as a whole whereby decisions and priorities can change significantly during the products lifecycle. The recipe data warehouse must support each overall step in the development of new drugs:

- Pre-Clinical Phase
- Clinical Trials Phase I
- Clinical Trials Phase II
- Clinical Trials Phase III
- Product Launch and Manufacturing

The product development process should be managed by QbD principles and include the following steps:

- Quality Target Product Profile (QTPP) development
- Prior knowledge collection and Critical Quality Attributes (CQA) identification
- Product and process development including Critical Process Parameters (CPP) identification
- Design space development, including Design of Experiments (DoE)
- Control strategy development, including real time release testing and process validation
- Continuous improvement supported by, e.g., Process Analytical Technology (PAT)

The recipe warehouse must include the necessary data to perform each of these steps thereby encapsulating the continuum of compliant data - *Figure 3*.

S88 Recipe Objects and S95 Complementary Objects

The recipe data warehouse will be based on a common language for exchange of information about products and recipes for manufacturing of products as described in the ISA standards

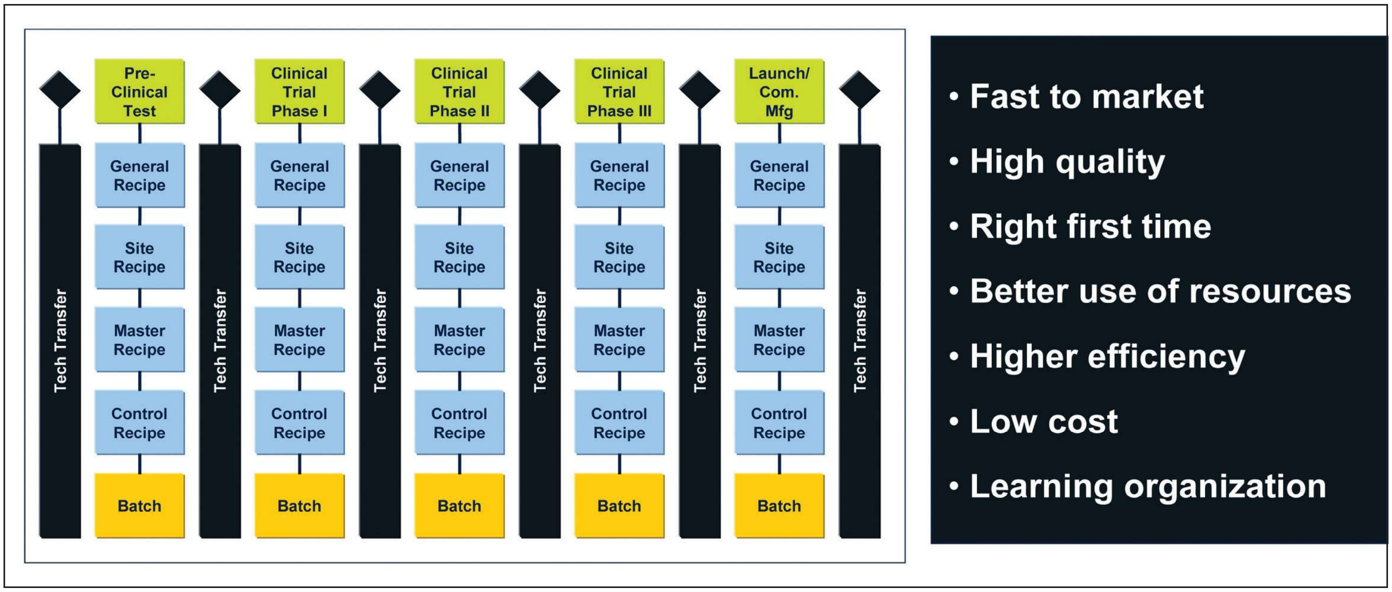

Figure 3. Recipe data warehouse must support tech transfer throughout complete development/manufacturing life cycle management.

S88 Batch Control,[1-3] and S95 Enterprise Control System Integration.[4-5] Alignment on these standards will help provide a common structure over the data which is maintained in multiple source systems.

Figure 4 describes the matrix of models described in S88. The horizontal slices address varying levels of repeatable units, operations, and parameters. The vertical slices then define varying levels of restrictions applied to these models which increase moving from the process model on the left side to the equipment model (physical model) on the right side. It is assumed that the equipment can be controlled by either a paper-based or a computer-based system, which get its product specific input from a recipe (equipment control). Fundamentally, it is important to note that the recipe establishes the link between the process and the equipment in this matrix format to provide for ultimate flexibility.

Implication of Vertical Slices in the Procedural Model

The evolution from a process view to an equipment/execution view is defined as procedural control which is synonymous to a control strategy. Hence, if your regulatory filings are aligned with these overall procedural models, it will help ensure you are indeed providing the most transparent process definitions to the agencies as well as providing flexibility for your commercial manufacturing. For example, a regulatory filing would outline a general recipe and include all Critical Process Parameters (CPPs) and Critical Material Attributes (CMAs) defined, and include a procedural definition as a mean to describe the products control strategy. Certainly, master and control recipes leveraged in development that helped define these CPPs and CMAs would be shared, but only to justify the overlaying general recipe for the product. In such a manner, commercial manufacturing and the agencies are provided clear definitions and processes for the control strategy.

S88 describes how process descriptions may be transformed into similar structure for a recipe. It is important to note again that the information captured in the recipe contains both the process execution as well as the quality testing methods/data. It is through this combination of information in one central location that facilitates effective definition of CPPs and CMAs.

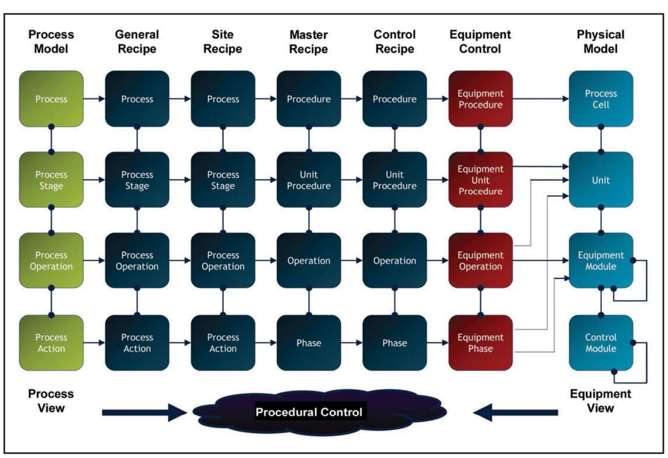

Figure 4. S88 procedural controls and model definitions.

Recipe Definitions

According to S88 Reference,[1-3] a recipe is "an entity that contains the minimum set of information that uniquely defines the manufacturing requirements for a specific product." It is used to describe products and how to produce products. In practice, you need varying degrees of information specificity for different recipients of the information in the organization. That's why S88 operates with four different recipe types as shown in Table A.

Process Models Equivalent to a Platform

Strictly speaking, process models are intended to be independent of product and materials. However, in discussions around alignment of platform definitions and recipe based definitions, we have taken the editorial liberty to enable some material definitions/classes to be defined in these process models as well as equipment parameters and settings. This decision was made to help enforce some further standardization the corporation was looking for in the overall platform discussions.

Product Specific Recipes

Figure 4 describes the evolution from a general recipe to a control recipe. Note the clear equipment independency implied by these recipe definitions. This is important to note and follows on the conversations above around actual filing strategies/recommendations for products. The S88 standard,[1-3] defines four different recipe types:

- General recipe: a type of recipe that expresses equipment and site independent processing requirements.
- Site recipe: a type of recipe that is site specific.
- Master recipe: a type of recipe that accounts for equipment capabilities and may include process cell-specific information.
- Control recipe: a type of recipe which, through its execution, defines the manufacture of a single batch of a specific product.

Each of these recipes is further described in the S88 standard as shown in Table A.

Recipe Types
The ***General recipe*** is an enterprise level recipe that serves as the basis for lower-level recipes. It is created without specific knowledge of the process cell equipment that will be used to manufacture the product. It identifies raw materials, their relative quantities, and required processing, but without specific regard to a particular site or the equipment available at that site. The general recipe provides a means for communicating processing requirements to multiple manufacturing locations. It may be used as a basis for enterprise-wide planning and investment decisions.
The ***Site recipe*** is specific to a particular site. It is the combination of site-specific information and a general recipe. It is usually derived from a general recipe to meet the conditions found at a particular manufacturing location and provides the level of detail necessary for site-level, long-term production scheduling. However, it may also be created directly without the existence of a general recipe. There may be multiple site recipes derived from a general recipe, each covering a part of the general recipe that may be implemented at a specific site
The ***Master recipe*** is that level of recipe that is targeted to a process cell or a subset of the process cell equipment. Some characteristics of master recipes include the following: • The master recipe has to be sufficiently adapted to the properties of the process cell equipment to ensure the correct processing of the batch. • The master recipe may contain product-specific information required for detailed scheduling, such as process-input information or equipment requirements. • The master recipe level is a required recipe level, because without it no control recipes can be created and, therefore, no batch can be produced
The ***Control recipe*** starts as a copy of the master recipe and is then modified as necessary with scheduling and operational information to be specific to a single batch. It contains product-specific process information necessary to manufacture a particular batch of product. It provides the level of detail necessary to initiate and monitor equipment procedural entities in a process cell. It may have been modified to account for actual raw material qualities and actual equipment to be utilized.
Recipe Categories of Information
The ***Header*** in the recipe comprises administrative information. Typical header information may include the recipe and product identification, the version number, the originator, the issue date, approvals, status, and other administrative information.
The ***Formula*** is a category of recipe information that includes process inputs, process parameters and process outputs. A ***process input*** is the identification of a raw material or other resource required to make the product. A process parameter details information such as temperature, pressure, or time that is pertinent to the product but does not fall into the classification of input or output. A ***process output*** is the identification and quantity of a material and/or energy expected to result from one execution of the recipe. ***Equipment requirements*** constrain the choice of the equipment that will eventually be used to implement a specific part of the procedure. In general and site recipes the equipment requirements are typically described in general terms, such as allowable materials and required processing characteristics. At the master recipe level, the equipment requirements may be expressed in any manner that specifies allowable equipment in process cells. At the control recipe level, the equipment requirements are the same as the allowable equipment in the master recipe.

Table A. Glossary of recipe terms as defined in S88.[1]

Implication of Horizontal Slices in the Procedural Model

According to S88, each of these vertical slices is further matrixed to describe in a structured way by splitting the process up into process stages, process operations, and process actions - *Figure 5.* To complete the process description a set of parameters describing required materials, equipment and personnel and specifying process variables may be assigned to each process action.

According the S88 standard, [1-3] the recipes contain the following categories of information: header, formula, equipment requirements, and procedure. Each of these categories is further described in the S88 standard as shown in Table A.

Recipe Data Warehouse Development – S88/S95 Meets Kimball

Combining the S88/S95 data standards with the informatics strategy outlined by Kimball, we have called this system the "recipe data warehouse"[6] recognizing the importance of the relationship between recipes and informatics (i.e., information management) strategies. The combined data model is proposed in Figure 6 recognizing some key staging areas isolating the source systems and target systems. Source systems are exist-

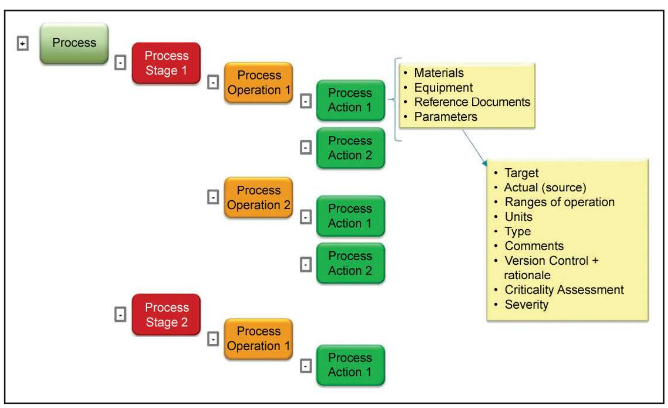

Figure 5. S88 based recipe structure.

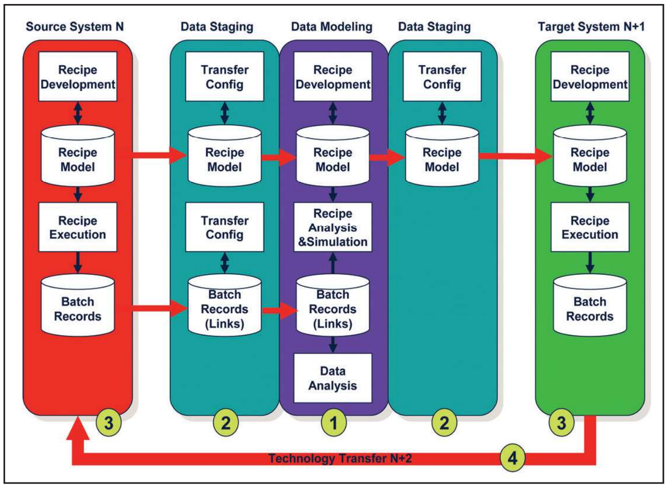

Figure 6. Recipe data warehouse conceptual system architecture.

ing systems that are used to create or modify existing product recipes in the central recipe data warehouse or to produce batches based on existing product recipes where the batch data will be used for data analysis by the central recipe data warehouse. Target systems are existing systems that use existing product recipes to produce batches (experiments, lab production, commercial production, etc.). The numbers in parenthesis below relate to Figure 6.

The conceptual architecture illustrates both the central recipe data warehouse (1) the data staging for connected systems (2), and the connected systems (3) as well as the conceptual workflows (4) related to defined business processes which creates, modifies, uses, or analyzes the recipe data (e.g., material, equipment, people and process definitions, and in process data as well as release/stability data).

It is anticipated in the generic model that all, some, or none of the current systems can act in the role of both being a source system or a target system (3).

Data staging is intended for each connected system to enable standardizing and normalizing on data structures in the central recipe data warehouse and de-coupling these structures from the native data structures used in and by the connected systems themselves.

The core of the architecture is the Central recipe data warehouse (1), which holds the following types of data:

- Standardized/normalized product recipes
- Tools for recipe development, including recipe building blocks
- Meta data for recipe analysis and simulation
- Meta data for linking to batch data in any defined source system (3)
- Tools for analyzing recipe/batch data

It is important to realize that a significant portion, if not all of our current data is stored in a manner that does not comply with recipes and sources range from excel workbooks, custom databases, emails, pdf documents, paper records, etc. So a huge value in building a unified, system independent model is that it helps to capture and contextualize this disparate data today.

The transformation or mapping from/to the specific systems of the generalized data models and structures used in the central recipe data warehouse is done by data staging.

The data staging is intended to be an integrated part or the central recipe data warehouse with centralized configuration of the transformations. This gives a good de-coupling of the connected systems from the central recipe data warehouse and it furthermore gives a robust and consistent basis for managing the data transformations.

For some of the current systems, a full and complete data transformation may not be possible or GMP and other regulations may prevent a direct storing of data into the central recipe data warehouse. In such cases, the data staging could include a user interface component for committing of the transformed data. Data staging has to cover both product recipe data and batch data transformations.

Data staging has for some of the existing systems to be bi-directional to enable business process workflows. This is illustrated by the data staging between the source systems and the data modeling for transforming data that creates or modifies existing data in the recipe data warehouse. Data subsequently leaving the warehouse would go through a similar staging environment where appropriate mapping to system specific definitions would be defined. The inbound and outbound data staging is clearly not identical and must be treated as separate transformations with specific and individual configurations for each of the connected systems.

User interface components shall be included for configuration of the data transformation and eventual data commitment functionality.

Workflows which can be for optimizing, fine tuning, and development of recipes is illustrated by the target system (N+1) which loads an existing production recipe (or recipe component) from the central recipe data warehouse. This recipe is then modified before or during recipe execution and batch data is collected during this execution.

There may now be a desire to update the production recipe to a new "version" in the central recipe data warehouse, and the "arrow" with "technology transfer N+2" indicates that this specific system used for the recipe execution now changes from being a target system to also being a source system. In this way, many of the existing systems can be used as both a source and a target system.

A special methodology which could be used for recipe development is simulation of processes. Simulation of processes is based on Process Models which can be developed based on historical batch data by use of standard software products.

Once such process models have been developed, these can be used by software engines to simulate the modeled process with a variance on inputs (recipe modifications).

In the current concept illustrated above, such simulation is intended to be included as applications in the central recipe data warehouse, but such simulations also could be seen as just another set of source and target systems. In the latter case, this may require that additional data staging components are made for batch data from the central recipe data warehouse to the target system to enable the process model to operate (recipe execution) on real batch data. (This data staging component is not illustrated).

Other Functions

The recipe data warehouse may eventually comprise data for use in quite a number of other applications. Figure 7 shows the envisioned functions.

The core function of the recipe data warehouse is called recipe authoring. Recipe authoring is the stepwise development/refinement of the recipe from the initial idea of the product to commercial manufacturing of the product to be supplied to the patient.

The recipe authoring process will be based on selection and combination of predefined recipe building blocks kept in a library. Different kinds of building blocks will be kept for specification of processes and related resource requirements (personnel, equipment, methods, and materials). The building blocks must represent the best practice in the complete development organization.

The development of new recipes will take advantage of the object oriented approach using object classes and object instances as described in S88, e.g., you may have a class of equipment called fluid bed dryers in your library and based on that class, you may create an instance of a fluid bed dryer called fluid bed dryer 23 linked to a specific recipe operation.

The recipe authoring will be supported by a graphical front end based and recipe representations standards like Sequential Function Charts (SFC) as described in S88.

Parts and bits of a recipe may be developed outside the recipe data warehouse in one of the source systems linked to the recipe data warehouse and then transferred to the recipe data warehouse through a standard based interface (recipe upload).Based on the data kept in the recipe data warehouse, the users may perform a number of different analysis and simulations:

- Simple views/reports, based on SQL queries in the database
- Advanced statistical analysis/reporting, based on statistical analysis methods, like statistical process control, Multivariate Data Analysis (MVA), Principal Component Analysis (PCA), etc.
- Risk assessment/reporting, based on entry of experience based or theoretical risk probability and risk consequence figures
- Advanced process modeling, based on pre-defined rules/equations for material and energy balances
- Advanced what-if analysis/simulation, based on process modeling methods

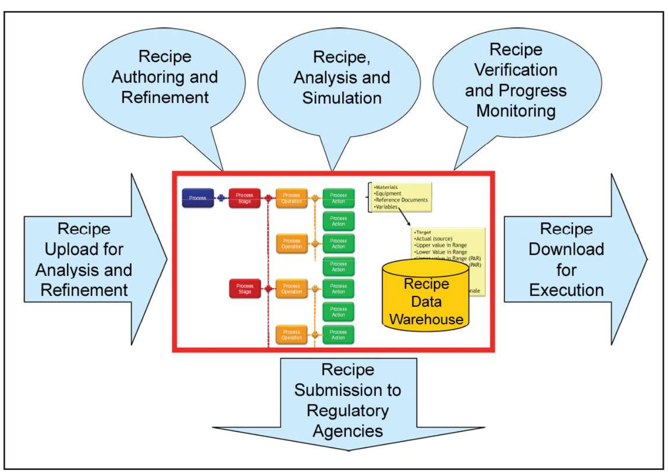

Figure 7. Recipe data warehouse – envisioned functions.

These analysis/simulation/reporting tools will be implemented by linking third party standard software packages to the recipe data warehouse. The recipe authoring process will be supported by various types of recipe verification functions:

- Automatic verification for consistency, based on recipe building rules
- Automatic verification of completeness, based on comparison with pre-defined recipes
- Automatic verification of regulatory/GxP compliance, based on check against pre-defined specification of regulatory/GxP requirements

Further, the recipe authoring process will be supported by a progress monitor describing Key Performance Indices (KPIs) for maturity and readiness for submission/approval:

- Monitor dynamically progress on recipe development
- Compare different versions of recipe and track changes

The progress monitoring tool would provide management with an excellent overview of the product development progress.

Eventually the recipe data warehouse may be used to download recipes for execution in a target system (recipe download). A target system could, for example, be a batch control system in a commercial manufacturing facility. This would require quite detailed modeling of not only the process and the related resource requirements, but also modeling of conditions for transitions and constraints for use of particular equipments.

The recipe data warehouse may store data from execution of recipes or provide on-line links to such data kept by external source systems making it possible to use historical data for the analysis and simulations described above. Historical data may exist in large amounts and may be kept in special historian databases and it may be smart to keep such data in these special databases and just establish links to the data when needed for analysis.

Eventually, the recipe data warehouse may be used for submission of files for approval by regulatory authorities like FDA. Two levels of support may be envisioned for submissions:

- Automatic provision of data for file submissions to regulatory authorities, for example:
  - Collating information over time for a given unit operation would link to S3.2.6 – process development history

  - Information for the current master recipe would link to S3.2.4 process description
  - Specific information gathered during certain instances of the control recipe would link to S3.2.5 process validation and would for the core data set for ongoing process verification, especially useful when combined with the general or site recipe definition(s) for products manufactured at multiple sites
- Manage/track changes related to submitted files

The recipe data warehouse may be used for transfer of recipe data between different systems used for:

- Modeling/specification
- Batch control/execution/reporting
- Quality control/LIMS

Transfer of data between source and target systems would require validation of the recipe data warehouse.

Current Recipe Data Warehouse Experience

A prototype of the recipe data warehouse was built based on S88 and S95 standards and consistent with the published data models.

Standard S88 recipe process models for manufacturing processing and testing thereof was undertaken to drive a common platform of definitions for solid dosage and large molecule synthesis. These process models were loaded into the recipe data warehouse and subsequently used as a framework to abstract general, site, master, and control recipes from previously collected process and analytical data.

Implementation of the concept presented here required a significant amount of data manipulation as the current structure was as diverse as the number of experiments. Thereby, a significant amount of work was undertaken to transform executed batch records and associative analytical data into the recipe structure described above. More than 300 control recipes were converted and stored in the recipe data warehouse. Once the data was loaded, some query tools were developed to help retrieve and visualize the information from the warehouse as depicted in Figure 8.

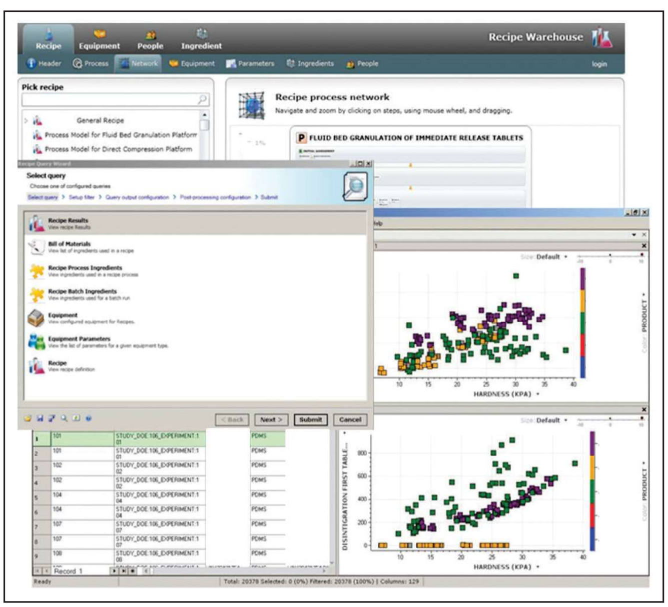

Figure 8. Recipe data warehouse – user interface.

Further Development Plans

Any data warehousing approach requires the organization to look at a continuous improvement of the data model, analysis, and reporting to help ensure learning is leveraged across the lifecycle of the product and across products. As such, the recipe data warehouse future will include expanding the data integration into quality control systems as well as business management tools. Through the combination of product knowledge and resource allocation, the acceleration goals of the organization can be reached, ultimately delivering value to the patients.

Conclusion

The pharmaceutical industry is awash with data, resultant from recipe execution. This data is generated via analytical and process recipe execution, but lacks context to support swift product lifecycle management. As such, recipe based execution requires a strategic approach toward the data management associated with said execution in order to change the current data paradigm from reactive to proactive with respect to releasing the inherent knowledge. To achieve the sustainability and ultimate vision of recipe based execution a system strategy coined "recipe data warehouse" is outlined here. The strategy leverages both external (S88/S95) and internal best practices by which a novel data warehouse was generated to address both. Through the implementation of such a system, the business benefits can be realized via the embracement of scientific and engineering methods. Empowering employees in the corporation will without doubt lead to a sustainable, scalable, and flexible environment to execute on the complex nature of commercializing new medicines for our patients.

References

1. ISA Standard S88.01 "Batch Control – Part 1: Models and Terminology."

2. ISA Standard S88.02 "Batch Control – Part 2: Data Structures and Guidelines for Languages."

3. ISA Standard S88.03 "Batch Control – Part 3: General and Site Recipe Models and Representation."

4. ISA Standard S95.01 "Enterprise-Control System Integration Part 1: Models and Terminology."

5. ISA Standard S95.02 "Enterprise-Control System Integration Part 2: Object Model Attributes."

6. Kimball, R., Ross, M., *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling,* John Wiley and Sons, Inc., New York, NY, USA, 2002.

7. Fermier, A., McKenzie, P., McWeeney, S., Murphy, T., Schaefer, G.,"Transforming Data into Information: An

Engineering Approach," *Pharmaceutical Manufacturing,* Volume 10 (8), p. 20.

Acknowledgements

Dimitris Agrafiotis, Ryan Bass, Arleene Canales, Walter Cedeno, John Cunningham, Nick Dani, John Dingerdissen, Peter Gates, Joel Hanson, Ed Jaeger, Pascal Maes, Shaun McWeeney, Steve Mehrman, Brian Sherry, Andrew Skalkin, John Stong, James Weber, Geert Verreck from Janssen and Frede Vinther, Jorgen Beck, Soren Trostmann, and Anders Magnusson from NNE Pharmaplan. Thanks to colleagues and collaboration partners for contributions.

About the Authors

Adam M. Fermier, PhD, is a Principal Engineer at Janssen, where he has been since 1998. He started his career in early development of small molecules, focused on laboratory automation, and later moved into drug product development, where he led a team focused on providing process analytical support for a pilot plant. In 2010, he joined the Strategic Operations group, where he continued to partner on developing the informatics strategy presented here.

Paul McKenzie, PhD, leads the Global Development Organization (GDO) of Janssen Pharmaceutical Companies of Johnson & Johnson. In this role, McKenzie drives value creation and increases R&D operational effectiveness. He and his teams interact closely with the therapeutic and functional areas leaders, as well as supply chain and external partners to develop and deliver innovative and much needed medicines to patients. The GDO is comprised of the following functions: clinical development operations, small molecule clinical pharmacology, drug safety sciences, external alliances and business models, and pharmaceutical development and manufacturing sciences.In his career, he has worked in both R&D and supply chain activities covering both large and small molecules. McKenzie came to Johnson & Johnson from Bristol-Myers Squibb (BMS), where he was Vice President and General Manager of the BMS large-scale cell culture facility in Massachusetts. Prior to this role, he was Vice President, Technical Transfer Governance Committee, at BMS, where he partnered with key functional areas to create and oversee the technical transfer of biological, chemical, drug product and natural product processes for new drug candidates, and completed multiple global submissions for new pipeline products. Prior to BMS, he worked for Merck in various roles with the company's large-scale organic pilot plant and pharmaceutical development and clinical supply pilot plants. He is a graduate of the University of Pennsylvania (BS in chemical engineering) and Carnegie Mellon University (PhD in chemical engineering). He is currently a member of the Board of Trustees of Illinois Institute of Technology, the board of Society for Biological Engineering. He has served on numerous professional and academic boards. In his spare time, he enjoys supporting his children in their academic and athletic pursuits. All of his kids enjoy competitive swimming and that has led him to become certified as a YMCA and USA swimming coach and official. He still competes himself in both masters swimming and water polo. When not at a pool for a swim meet, McKenzie and his family enjoy spending the summer weekends with their extended family at the New Jersey coast.

Terry Murphy is Janssen's Global Head of Strategic Operations, Pharmaceutical Development and Manufacturing Sciences. He joined Centocor, a Johnson & Johnson company, in 2005, during the construction of their new Biologics facility in Cork, Ireland. Since then, he has held various roles of increasing responsibility, primarily in the manufacturing systems space until moving in 2009 to his current role in pharmaceuticals R&D.

Leif Poulsen, PhD, is a Senior Specialist in automation and IT at NNE Pharmaplan, which provides consulting and engineering services to the pharmaceutical and biotech industry worldwide. Poulsen holds a Masters degree and a PhD in process engineering. Poulsen has specialized in manufacturing execution systems and advanced automation and IT solutions and how they can support current and future business objectives. He has been with NNE Pharmaplan for about 15 years where he is responsible for development of technology, methods and competences within automation and IT. He is member of the ISA SP88 Batch Control Committee, ISA SP95 Enterprise Control System Integration Committee, and a member of the ISPE GAMP Forum. He can be contacted by email: lpou@nnepharmaplan.com.

NNE Pharmaplan – Automation & IT, Nybrovej 80, Gentofte Dk-2820, Denmark.

Gene Schaefer, ScD is currently Senior Director, API Large Molecule Pharmaceutical Development and Manufacturing Sciences at Janssen Pharmaceuticals, in Spring House and Malvern, Pennsylvania. In this role, he is responsible for a number of projects from early-stage process development to commercial product support for protein therapeutics. Previously, he was Director of Process Technologies in the Protein Therapeutics Development group at Bristol-Myers Squibb in Hopewell, New Jersey. He also worked at Schering-Plough in Union, New Jersey, and at Genzyme in Boston and the U.K.