Journal of Pharmaceutical Innovation
©2006 ISPE. All rights reserved. | www.ispe.org/jpi

Toward intelligent decision support for pharmaceutical product development

Chunhua Zhao, Ankur Jain, Leaelaf Hailemariam, Pradeep Suresh, Pavankumar Akkisetty, Girish Joglekar, Venkat Venkatasubramanian, Gintaras V. Reklaitis, Ken Morris, and Prabir Basu

School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA
Corresponding author: Venkatasubramanian, V. (venkat@ecn.purdue.edu).

Developing pharmaceutical product formulation in a timely manner and ensuring quality is a complex process that requires a systematic, science-based approach. Information from various categories, including properties of the drug substance and excipients, interactions between materials, unit operations, and equipment is gathered. Knowledge in different forms, including heuristics, decision trees, correlations, and first-principle models is applied. Decisions regarding processing routes, choice of excipients, and equipment sizing are made based on this information and knowledge. In this work, we report on the development of a software infrastructure to assist formulation scientists in managing the information, capturing the knowledge, and providing intelligent decision support for pharmaceutical product formulation.

Introduction

The current emphasis in the development of a pharmaceutical product is on shortening development time, reducing development costs, and improving the process design to ensure higher flexibility. Commercial scale product and process development typically goes through the following stages after the viability of a newly discovered molecule is established: laboratory scale, pilot plant scale, and commercial scale manufacturing. Laboratory scale experiments are used to determine the physical and chemical properties of a drug, the desired dosage form, and critical quality attributes. Selected pilot plant experiments, typically guided by design-of-experiment methodologies, are carried out to provide a detailed understanding of the processing steps in the selected route and to generate the data needed for scale-up to commercial manufacturing. The end-point determination criteria and the preliminary design space are determined at this stage. The next stage involves the testing and revision of the design space. The information related to manufacturing is used in troubleshooting and productivity improvement studies. The three stages are closely related through the information they exchange. For example, information generated at the laboratory scale can be used to improve manufacturing; problems identified in the manufacturing stage are communicated to the laboratory scale to identify root causes and develop measures to avoid similar problems in

Continued on page 24.

future product development.

Pharmaceutical product development is an information and knowledge intensive process. Use of new process analytical technologies (PAT) [1] has enabled scientists to get a better understanding of the underlying physical and chemical phenomena. The knowledge created from the learning process can be in different forms: reports on paper, data in electronic format, and experience gained by scientists. Because more information and knowledge become available, it is clear that more-powerful and intelligent software systems to manage and access them effectively for efficient decision making are needed.

Pharmaceutical product development involves the integration of process modeling tools, effective handling of laboratory generated information and knowledge, development of technical specifications, and an information base to satisfy regulatory requirements. To support the activities and decision-making processes in pharmaceutical product development, a systematic and integrated informatics framework based on formal and explicit modeling of related information is required. These information models should be easily accessible by humans and software tools and should provide a common understanding for information sharing. Various forms of knowledge, including heuristic rules, guidelines, and mathematical models need to be handled in a systematic manner so that the knowledge can be easily created, used independently or in an integrated fashion. Only with such a framework can intelligent decision-support systems be developed to provide decision support proactively.

Several intelligent systems have been developed in the past two decades to capture the heuristic knowledge and support the pre-formulation activity in pharmaceutical product development. During pre-formulation, one or more formulations that meet the product specifications are generated. A formulation comprises the dose of the drug substance along with excipients and their quantities, a manufacturing process to produce the associated dosage form, and key operating conditions of the process. In designing a formulation, the formulator has to consider the properties of the ingredients and possible interactions between the ingredients. This requires navigation through a large and complex design space wherein the relations between the properties are frequently ill defined. Experts are frequently adept at navigating through the design space; however, their knowledge and thought processes are difficult to quantify and explain, and to transmit [2]. Decision-support tools to capture this knowledge, such as rule-based expert systems, have been proposed in the literature and are reviewed in the following section.

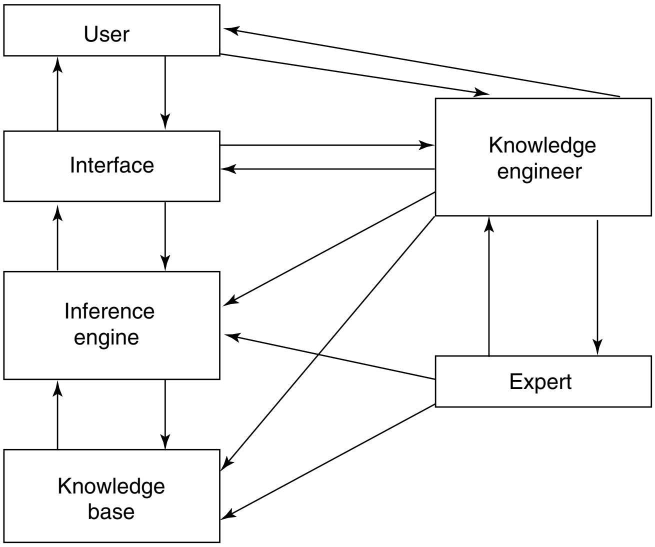


Figure 1. Schematic diagram of a simple expert system showing its interaction with user, developer (knowledge engineer), and domain expert. The three main components are a user interface, an inference engine, and a knowledge base [35].

Decision-support tools

In rule-based systems, knowledge is modeled as a series of rules utilizing items of information: for example, if a system recognizes certain condition (e.g., if the drug substance is insoluble), then it proposes an action (e.g., use a soluble filler). Other examples include: ‘If the functional group of the drug substance is highly acidic, then it needs a moderate binder’ and ‘If the melting point of the drug substance is low (<75°C), it needs a diluent’ [3]. In addition, a user interface and an inference engine are required to provide the decision support (Figure 1). Knowledge about the domain is gathered by interviewing experts or using machine learning technologies such as neuro-fuzzy logic-based learning, which generates rules based on historical data [4]. The effectiveness and span of applications of a rule-based system depend on the amount and validity of the data in the knowledge base.

Rule-based systems are commonly used in tablet and capsule formulation to select excipients based on the properties of the drug substance. The user inputs the relevant physical, chemical, and mechanical properties of the drug substance along with the product specification. Using the knowledge base and inference engine, the system predicts the type and quantity of excipients required to meet the specifications. The formulator tests the conformance of the predicted formulations with product specification and feeds the information back to the expert system, modifying the formulation if necessary. A technique that is often used in knowledge acquisition is rapid prototyping. In this approach, the knowledge engineer quickly builds a prototype, which is then shown to the experts and users, who suggest incremental modifications [5].

Possibly the earliest expert system for pharmaceutical formulation was the one developed by Podczeck [6]. Mixtures of predetermined composition were used in experiments to explore the relationships between the set of dependent and independent factors by multivariate statistics. These relationships were converted to rules, and these rules were used in conjunction with other rules developed independently to determine which mixture should be used for each formulation.

Logica's Product Formulation Expert System (PFES) was developed as a reusable software kernel to support a generic formulation task [7]. The formulation process is driven by a hierarchy of tasks. At the level of a task are the formulation object (defining the current composition of the formulation) and the specification object (defining the current state of knowledge about the formulation problem). The knowledge in the specification object is used to move the formulation process forward. Provision was made for development and maintenance of knowledge via a rule template. In the Boots/SOLTAN System [8], knowledge that was gained from interviews with senior formulators was used with PFES. Existing information sources, such as databases, were presented in a frame-based semantic network that can be manipulated by the problem-solving knowledge of the domain. Bateman *et al.* [9] developed the Sanofi System for the formulation of hard-gelatin capsules based on specific pre-formulation data of the active ingredient. Using PFES, the system generated one formulation with as many subsequent formulations as desired to accommodate an experimental design and the knowledge of the user. Rowe *et al.* [10] implemented a film-coating formulation system using PFES. A later version of PFES, Formulogic [2,9], is comprised of a three-level architecture: physical, task, and control levels. The physical level contains the domain knowledge in several objects with attributes, and is accessed from the task level through a query interface. The control level is tasked with executing the tasks.

Ramani *et al.* [3] developed the Cadila System for tablet formulation. Excipients that are compatible with the drug substance are selected in two steps: selection of the properties that are desirable in excipients for compatibility with the drug substance and selection of the excipients that have the required properties. Typically, several feasible formulations are generated, from which the best is selected.

The Galenical Development System Heidelberg (GSH) was originally developed by Stricker *et al.* [11] and aimed at giving knowledge-based assistance in one phase of the pharmaceutical development, namely the galenical routine development of drug products. Galenical development deals with the development of a recipe for a certain drug and its manufacturing technology. Each development step is broken down into weighted actions, and the objective is to move from action to action based on predefined rules, while improving a scoring function without violating the constraints that are imposed by previous steps or forming incompatible combinations. Rowe *et al.* [12] modified the Galenical Development System to formulate a parenteral product. The system first attempts to optimize the solubility and/or stability of the drug before choosing suitable additives. Frank *et al.* [13] attempted to improve the Galenical System by building a subset of actions associated with a development step that is considered capable of solving the respective problem and using a predetermined ordering of development steps.

Hybrid systems

Rowe [14] suggested the possibility of integrating artificial neural networks (ANN) with expert systems. Neural networks might be used to comb through large databases for patterns that are then converted into rules. Rowe and Colburn [4] developed a way to generate rules from an already available formulation using fuzzy logic and automated rule induction (determination of rules) from experimental knowledge and the NEUfuzzy software.

Case-based reasoning (CBR) systems

Another approach to capture and reuse knowledge, a case-based reasoning (CBR) system, recognizes certain situations and from its case (knowledge) base recalls an action that was taken when similar problems were encountered. The solution is either used directly or adapted to account for differences between the current and previous problems. The proposed solution is then stored for future use. A convenient and concise description of the CBR process is 'retrieve, reuse, revise, and retain' [5].

CBR has been proposed as a response to the difficulties encountered in using rule-based systems, such as the need for an expert to encode the rules. Lai *et al.* [15,16] described an expert system for the formulation of hard-gelatin capsules based on previous formulations, trends in formulation, information on the drug and excipients in the form of rules derived from experimental analysis (the Capsugel system). Rowe *et al.* [17] presented a CBR approach to tablet formulation using the commercial software ReCall, which includes a case library of formulations depicted by an object-oriented representation, an induction engine, and an engine to check for similarity of the retrieved case with the new formulation. Guo *et al.* [18] proposed linking the Capsugel system to an ANN. The ANN predicted dissolution performance based on initial training. The Capsugel Expert System provided candidate formulations, the fitness of which was tested through the ANN against the required dissolution performance until the best formulation is achieved.

Expert systems have several important benefits [2]: (i) the knowledge they store is protected and available; (ii) there is consistency in the formulation process by expert systems, which is an important concern from a regulatory standpoint; (iii) expert systems can be used as training aids for both novices and professionals; (iv) such systems can reduce time of development [8] and thus facilitate cost savings; (v) human experts, freed from training and formulation develop-

Continued on page 26.

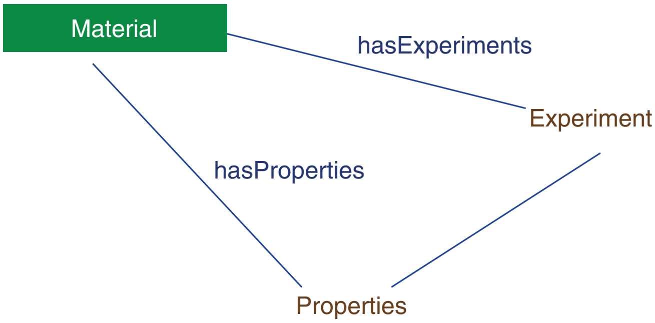


Figure 2. Concepts and their relations. Concepts are associated with a set of properties, and a range of a property might be an instance of another concept, thus defining their interrelationships.

ment, can devote more time to innovation; and (vi) better communication within the company occurs with the expert system being a common platform for discussion and identification of crucial research areas.

Expert systems provide some of the functionality required by a decision-support system. However, to support decision making in product development effectively, several other components are also important. For example, information and knowledge must be shared among all the stages in pharmaceutical product development for improved manufacturing, problem diagnosis, and product design. The above goals make it very important for the decision support system to integrate information, effectively manage information from multiple sources (the developer, experiments, and computer tools), use functionalities between tools, and integrate different types of knowledge (heuristics and mathematical knowledge in the form of equations). There is awareness of the above issues in existing formulation expert systems. Hohne and Houghton [19] provided access to the modeling calculations performed by synthetic chemists with the expert system acting as an interface. In Formulogic, the objects in a knowledge database can be accessed through a query interface [2]. However, these efforts were not comprehensive. Effective product development can only be brought about through a seamless integration of information, knowledge in various forms, and diverse computer tools.

In this work, we adopt ontology as a foundation for developing an integrated framework. Ontology defines and semantically describes the data and information. Ontology is also the basis for modeling different forms of knowledge. We consider knowledge to be organized or contextualized information that can be used to produce new meanings and generate new information. We have concentrated on two different types of knowledge: mathematical knowledge, which is concise, precise, and abstract, and knowledge to guide decision making, including decision trees and heuristic, which can be modeled as guidelines.

The remainder of this article discusses: (i) ontology-based information modeling; (ii) the use of information modeling to support information management during the preformulation stage of drug development (examples are used to compare the proposed approach with existing solutions); (iii) the modeling of knowledge in the form of guidelines and modeling of mathematical knowledge; and (iv) the integrated decision-support system architecture.

Information modeling

Information can be divided into two types: unstructured and structured. Information that can only be processed by humans is categorized as unstructured information; for example, experimental results that are reported as Word documents. This type of information cannot be used directly by software tools for information processing or drawing inferences. For the information to become machine processable, it has to be in a syntax that is semantically rich, and therefore, understandable by machines and by humans. This type of information is called structured information. Examples of structured information are meta-data for files, such as the predefined set of terms to describe the title, subject, author, and other information, and data generated from instruments, which are typically in tabular form with specified meaning to each column. Formal information models are the foundation of structured information.

Several approaches exist to model the information. However, such information models are usually designed for specific applications and only provide a limited view of the information. An information-centric approach has been proposed [20] in which information is modeled using ontology. Ontology is a formal and explicit specification of a shared abstract model of a phenomenon through identification of its relevant concepts [21]. For example, as shown in Figure 2, material has several properties, and the property values can be determined from experiments. The ontology for material properties captures relations such as 'hasProperties' and 'hasExperiments'. The relations are specified by defining ranges for the properties. Thus, the range of the property 'hasProperties' of the class 'Material' is the class 'Properties'. Such relations are usable by both humans and computer tools. Compared with a database, which targets physical data independence, and an XML schema, which targets document structure, ontology targets agreed-upon and explicit semantics of the information, and directly describes the concepts and their relations. Web ontology language (OWL, http://www.w3.org/TR/owl-features/) was used in this work to create ontologies.

Ontology building for pharmaceutical product development

Ontology building is an evolutionary design process that consists of proposing, implementing and refining classes and properties that comprise an ontology [22]. The steps involved include: (i) determining the domain and scope of the ontology; (ii) reusing existing ontologies; (iii) enumerating important terms in the ontology; and (iv) defining the classes and class hierarchy.

The central concept in the abstraction described above is material, which can be a pure substance or a mixture. A material has several properties (e.g., specific heat capacity), can have several roles (e.g., drug substance and flow aid) and can be involved in several experiments (e.g., Hosokawa Tests). The concepts, such as material, roles, and experiments, are defined as classes in ontology. The list of material properties might be classified into engineering properties, compound properties, particle properties, and powder properties. Engineering properties include those that are used in engineering calculations, such as thermal conductivity. Compound properties include molecular properties, such as molecular mass, and a description of chemical stability. Particle properties include crystalline properties and a description of the physical stability (if crystalline, the crystal system). Powder properties describe the behavior of several particles of the material, such as flow characteristics or deformation of the powder when pressed into tablets. Each property is represented by a class with its own set of attributes. A material property value can be measured experimentally, calculated mathematically, or retrieved from the literature. If measured experimentally, the conditions under which an experiment is performed defines its context, which might consist of temperature, pH, relative humidity, and so on. The description of an experiment includes the materials involved, the experimenter, the location of the experiment, the date and time of the experiment, the equipment used, the procedure followed, and the experimental data.

The relations between the experiments, the materials on which they are performed, and the properties that are measured are explicitly described. Modeling of the domain information (i.e., creating the domain ontology) requires an understanding of the ontology-building techniques and of the domain being modeled. In this project, we collaborated with colleagues in the Department of Industrial and Physical Pharmacy at Purdue University and with external experts. The current ontology is the result of several iterations of the propose–discuss–revise steps. The visualization tools provided by the Protégé ontology editor (http://protege.stanford.edu/), such as plug-ins to view class hierarchy graphs and automatically generated forms for information entry, facilitate ontology development.

The concept of 'development state' is defined to characterize systematically the sequence of steps that lead to the final product (i.e., selection of a dosage form, processing route, excipient roles, specific excipients, and excipient compositions). In addition, it contains the description of the drug substance, excipients, and the dose amount.

The ontologies are the foundation for the information repository and are used to provide access to information for various tools, including an engine to execute guidelines and an engine to utilize mathematical knowledge. Ontologies for material, formulation, processing route, experiment, and unit operations have been developed. The top-level concepts of these ontologies and their relations are shown in Figure 3.

Information management

Voluminous information is generated during drug-product development, including raw data from analytical instruments, pictures from scanning electron microscopes (SEMs), experimental set-ups, experiment notes and reports, mass and energy balance results from simulation tools. The information also can be in different formats, such as plain-text files, Word documents, Excel worksheets, JPEG files, MPEG movies, and PDFs. Some of these formats generate unstructured information. Structured information must be generated from unstructured information before it can be used by the decision-making tools. How to gather information effectively from different resources and organize it for its end use are key information-management tasks. The key functionalities and shortcomings of various categories of information-management systems are discussed in the next section.

Current information-management solutions

Laboratory information-management system (LIMS)

Laboratory information-management system (LIMS) consists of database applications that are used to store and manage information in a laboratory setting [23]. Typical LIMS functionalities include sample tracking, data entry, sample scheduling, quality analysis, and quality control (enabling users to generate control charts and trend analysis graphs), automatic electronic data transfer (from analytical equipment to the LIMS), chemical and reagent inventory, personnel and equipment management, and maintenance of the database. A LIMS stores information in relational databases such as Oracle, DB2 or MS SQL [24]. Most LIMS have interfaces that give access to the database for information retrieval or storage. As discussed earlier, the data-

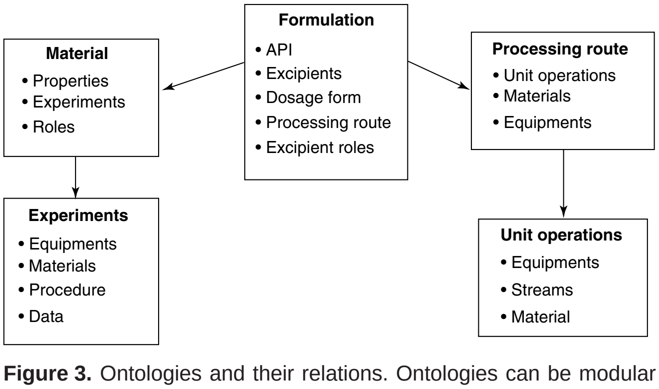


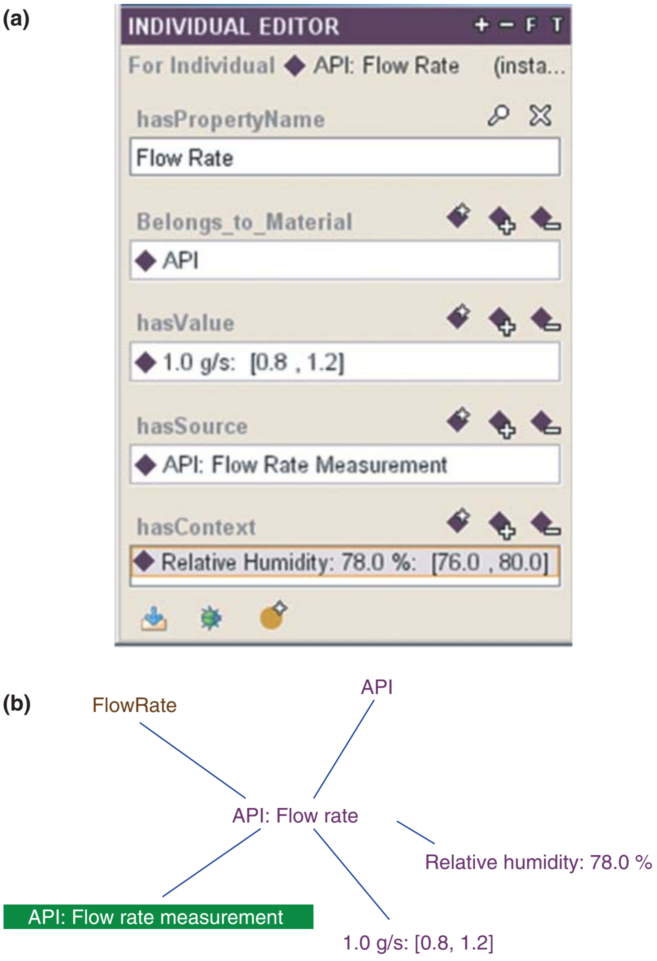
Figure 3. Ontologies and their relations. Ontologies can be modular and imported into other ontologies. For example, formulation ontology can import material ontology.

Continued on page 28.

base schemas provide only limited semantics. Relational database structures also limit the capability for describing complex relations between information.

E-laboratory notebook (ELN)

An E-laboratory notebook (ELN) can provide functionalities, including browsing online libraries, databases, and remote sources, such as the Web, writing documents and datasets, managing data, publishing and sharing information, and creating records [25]. With an electronic notebook, the records are shared by collaborators and reviewers. The utility of an ELN to provide a collaborative environment has proven inadequate for industry, especially when quality assurance and control is important [26]. Quality assurance demands experiments to be performed according to standard operational procedures. This might be accomplished by an automated interface for data entry. However, automated data-entry interfaces and integrated-information systems are not readily available for ELNs.



For Individual ◆ API: Flow Rate (insta...

hasPropertyName: Flow Rate

Belongs_to_Material: ◆ API

hasValue: ◆ 1.0 g/s: [0.8 , 1.2]

hasSource: ◆ API: Flow Rate Measurement

hasContext: ◆ Relative Humidity: 78.0 %: [76.0 , 80.0]

Figure 4. (a) Interface for creation of material properties. **(b)** Relations between concepts.

Content management system (CMS)

Content management system (CMS) supports the process of publishing, maintaining, and disseminating documents [27]. The major components of CMS are the data repository, user interface, workflow scheme, editorial tools, and output utilities. They enable writers to create or update content and track changes, and publish contents to make them available to all users in various configurations.

Ontology-driven information management

Without specifying the semantics of the information, it is difficult to provide functionalities beyond sharing the information among users and keyword-based search on the information. From our experience in implementing information-management systems in this project, we found two major problems with the current systems: organization of related information and lack of an open and systematic way to manage meta-data. These problems are directly related to the lack of the semantics of the information.

Although ontology defines the semantics of the information, an ‘instance’ or ‘individual’ contains information organized according to the semantics. For example, information about each experiment performed by an experimenter is stored as an ‘individual’ of the experiment concept. To manage the information associated with experiments, an experiment instance is created and is linked to the raw files generated from the experiment. Similarly, as shown in Figure 4a, an ‘instance’ of material properties is also created in which the links to relevant experiment instances are specified. Figure 4b shows a graphical view of the concepts and relations with respect to material, property, and context. The system can automatically locate these individuals and provide an integrated view of the information. For example, from a specific material, all the experiments performed with it and the associated properties are summarized.

The information infrastructure described above enables effective management of experimental files, which might exist in different forms with non-descriptive names and folder locations. A system for managing experimental data was developed based on the Alfresco (http://www.alfresco.org/) content management system. Because Alfresco is Web based, it provides ready access to information through a browser. It also enables search by keyword and on the hierarchy. Forms for creating, viewing, and editing instances were developed for the current domain. Several functionalities provided by Alfresco were directly applicable for the current application, such as user management, workflow, and security.

The semantic richness of the information provides several benefits. For example, a user can find all the experiments that have been performed on the micromeritics of the drug substance that are potentially affected by the relative humidity. In the current case study, the micromeritics of 39 materials (including mixtures) were studied, each with 18 micromeritic properties. There were on average five experiments for each property for every material, and each experiment had an average of three files associated with it. In semantic search, identification of relative humidity as an instance of context would lead to all instances of properties in which that context appeared. Through the relations between property and material, instances are further filtered based on the specified drug substance. The instances of these properties, which are subclasses of micromeritic properties, are found given the explicit definition of class–subclass relationships in the ontology. The individual experiments that are performed with these properties are identified through the part–whole relationship with the property. The experiment files that are linked to these experiments are presented as the search results. The semantic search engine found eight experiments conducted to determine the flow rate of a powder through an orifice for the particular drug substance as the micromeritic experiments affected by relative humidity. By contrast, without the ontology to provide the semantics, a keyword-based search would not be able to navigate using the relationships. Search using keywords 'relative humidity micromeritics' did not identify any of the experiments, whereas a similar search using 'relative humidity' identified many documents, most of which had very little to do with the experiments. Although it is acknowledged that such results are not indicative of all experimental work, they point to challenges faced by humans and machines in processing large amounts of information with little or no semantics. Conversely, they serve to illustrate the utility of semantic search made possible through the development of material, property, and experiment ontologies.

Presentation of information

Another important component of information management is the presentation of information to the user. To achieve the goals of accessing information anywhere on the Web and working on the information collaboratively, a Web-browser-based thin-client architecture is preferred. Forms and graphs are the two most widely used views of information and constitute the presentation layer. Typically, presentation format is interspersed with the presented information. However, they should be completely separated so that presentation format can be changed independently of information to satisfy user requirements. In the most widely used HTML forms, presentation and data are closely related. HTML forms are usually created manually and validated using scripts on the client side or code at the server side. Because information represented in OWL has explicit syntax and semantics, forms for the information should be generated and validated automatically. The look and feel of the forms should be controlled separately by a style sheet.

For this purpose, we used the XForms technology to generate forms for data entry and viewing purpose automatically. XForms is an XML-based technology that is considered to be the next generation of Web forms. In XForms, presentation and data (in XML format) are separated. Given an XML schema, an XForms form can be generated automatically using an editor (such as XFormation) and further modified for layout design. Given an XML instance, the form is automatically populated. Validation is handled by the XForms processor on client side, eliminating interactions with the server and screen refreshing. Output of the form is also an XML instance validated by XML schema. To use this technology, the conversion between OWL ontology and XML has to be carried out. OWL ontology has to be pushed down to XML schema, whereas OWL instances are converted into XML instances based on the schema. Bicer *et al.* [28] have developed a tool – OWLmt – that facilitates this conversion step.

In summary, in this ontology-driven information management approach, ontology has been created to model the information related to pharmaceutical drug development. It forms a structured information layer on top of the unstructured information layer. With the help of the structured information layer, tools can and have been developed to use the structured information to provide support to users of the information in the areas of management, access, and search.

Knowledge modeling

Knowledge can be classified into two categories: implicit knowledge, which is in the mind of a domain expert, and hence, usable only by the expert, and explicit knowledge, such as decision trees and procedures, mathematical models of operations, and guidelines from the Food and Drug Administration (FDA) or International Conference on Harmonisation (ICH). Explicit knowledge can be shared and used by all. As the body of knowledge and information grows, its management becomes more difficult. To get maximum benefit, we believe that knowledge should be modeled in the form that can be directly interpreted by computers, which in turn can be used by software tools in decision making. The conventional ways to model knowledge, either programming-based or rule-based, have severe limitations. In programming-based methods, the logic is hard-coded using a suitable programming language, and therefore, it is not accessible to a user. To make any changes, a user needs access to the source code, which sometimes might not be available or the user might not have the understanding of the particular programming language used. In rule-based expert systems, unorganized collections of rules are used to model the pieces of knowledge. Often, it is difficult to determine the purpose of individual rules and to envision potential interactions between rules. These issues greatly limit the

Continued on page 30.

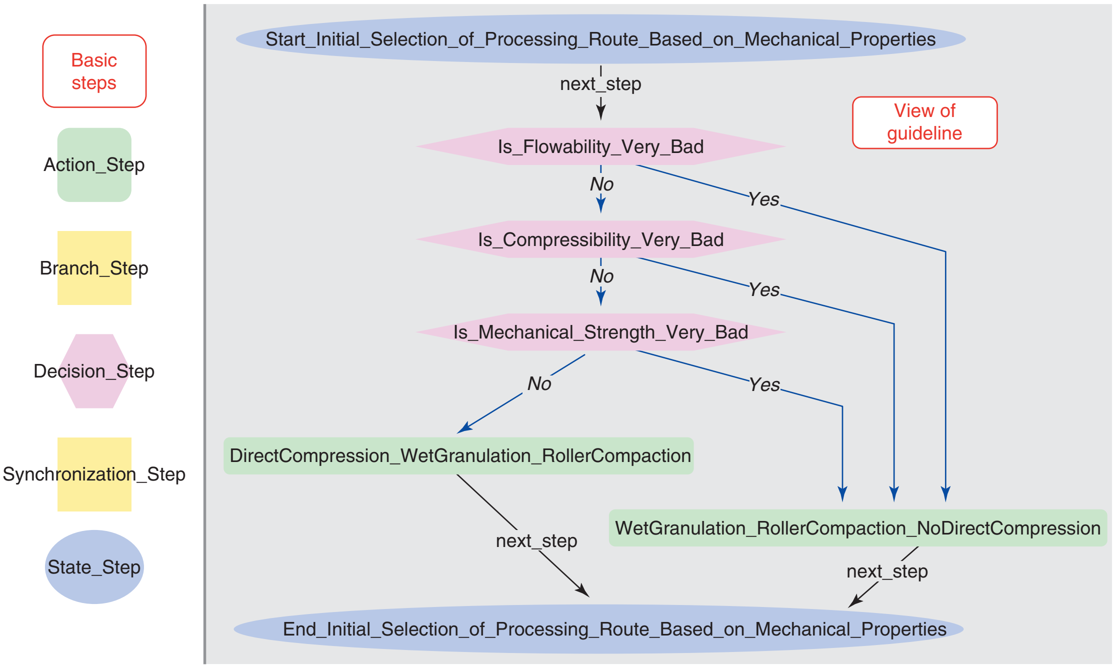

Basic steps
Action_Step
Branch_Step
Decision_Step
Synchronization_Step
State_Step
View of guideline
Start_Initial_Selection_of_Processing_Route_Based_on_Mechanical_Properties
next_step
Is_Flowability_Very_Bad
No
Yes
Is_Compressibility_Very_Bad
No
Yes
Is_Mechanical_Strength_Very_Bad
No
Yes
DirectCompression_WetGranulation_RollerCompaction
WetGranulation_RollerCompaction_NoDirectCompression
next_step
next_step
End_Initial_Selection_of_Processing_Route_Based_on_Mechanical_Properties

Figure 5. Details of a specific guideline. A guideline is encoded using five basic steps: state step, action step, decision step, branch step, and synchronization step. The guideline shown in the figure is used to select the processing route for manufacturing the dosage form based on mechanical properties of drug substance.

scalability and maintainability of rule-based systems.

Regardless of the forms of the knowledge, the users of the knowledge can be generally divided into two categories: the users who rely on the knowledge to make decisions and the experts who develop the knowledge. To support knowledge gathering, the system has to be able to present the knowledge in the most intuitive way, while taking care of most of the details. Furthermore, once the knowledge is gathered, it should be ready to be used in decision making. For users, the knowledge should be readily integrated with the information resources and other forms of knowledge. In the following discussion of modeling various forms of knowledge, we concentrate on approaches for gathering the knowledge, using the knowledge, and developing the engine to handle the underlying complexity to deal with integration issues and to provide decision support in an intelligent and proactive manner.

Guideline-knowledge modeling

A guideline models procedural knowledge, which mainly consists of decision logic, information look-up, and evaluation of decision variables. For example, to determine whether direct compression is appropriate for a particular drug substance, values of several properties, such as flowability and compressibility are examined. Systematically modeling procedural knowledge makes it possible to provide a standardized approach for product development and can be easily reused to ensure the quality of product development.

An ontology based approach is developed to model knowledge in the form of guidelines. A particular set of knowledge, such as how to select a processing route, is modeled as an instance of the guideline ontology. Furthermore, the ontology based approach provides a natural link between the knowledge and the information it utilizes, in addition to the knowledge and tools used. Various guidelines were developed for the selection of excipients and processing routes for pharmaceutical product formulation. The guidelines were based on the knowledge gathered from detailed discussions with the faculty in the department of Industrial and Physical Pharmacy at Purdue University.

Guidelines were created based on GuideLine Interchange Format (GLIF) [29], which is a specification developed mainly for structured representation of clinical guidelines. GLIF was developed to facilitate sharing of clinical knowledge and was designed to support computer-based guideline execution. GLIF guidelines are computer interpretable and human readable, and are independent of computing platforms. In GLIF, each guideline is represented as an instance of the guideline concept. The steps that are involved in decision making are modeled as an algorithm of that guideline. An algorithm is represented as a flowchart of nodes connected by directed links (Figure 5). Each node represents a 'guideline step', and a directed link denotes the order of execution of the steps. A step can be one of the five step types. An 'action step' represents a domain-specific or computational action; a 'decision step' represents a decision point; a 'state step' is used to specify the state of product development in the specific context of a guideline's application; a 'branch step' is used to initiate multiple actions in parallel; and a 'synchronization step' is used to coordinate concurrent steps or steps with arbitrary execution order. The decision-making process can be nested using subguidelines, and thus, multiple views to the process with different granularities can be defined. Figure 5 shows the different steps of a specific guideline.

A state step defines the starting or end point of a guideline. In this work, state step points to an instance of development state that contains the information about the current state of pharmaceutical product development. To use a guideline, first an instance of drug development state is created. A new development state instance contains the name of the active pharmaceutical ingredient (API) in the pharmaceutical product, which is an instance of the material ontology and the dose amount.

Every decision in a guideline is represented by a decision step (Figure 6). The main attribute of a decision step is a logical expression. For example, the decision step 'Is_Flowability_Very_Bad' is based on the expression 'Hausner_Ratio>1.75', where 'Hausner_Ratio' is the property of the material under consideration. Once the material is known, the property value can be retrieved from the material ontology. The value of the decision step can be either true or false. A decision step also specifies the options selected based on the value of the logical expression.

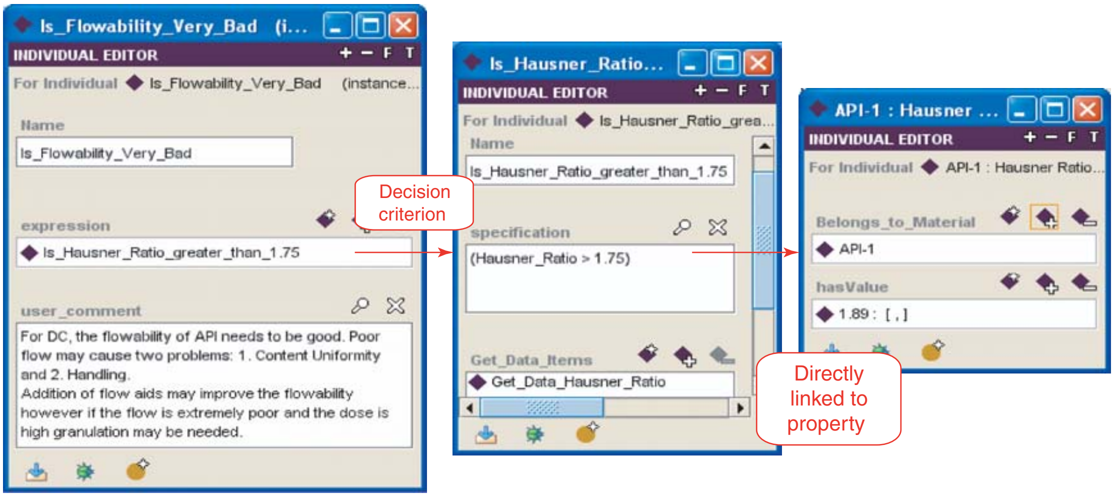
To use the information repository directly in the guidelines, connections have to be established between the guideline knowledge and the information repository. For example, in the decision step 'Is_Flowability_Very_Bad', the Hausner ratio of the API is used. The Universal Resource Identifier (URI) in the ontology provides a mechanism to identify a class, a property and an individual uniquely. For example, the class 'Hausner_Ratio' is identified by the URI http://pharma.rcac.purdue.edu/pharma/material/Property.owl#Hausner_Ratio, whereas the property 'hasAverageValue', which links the 'Hausner_Ratio' with a literal, is identified by http://pharma.rcac.purdue.edu/pharma/material/Property.owl#hasAverageValue.

Execution of guidelines

The decision-support system requires an engine for the execution of guidelines. The engine is linked to the guideline ontology. As a case study, guidelines were developed and used for the product development of a drug to treat multidrug-resistant tuberculosis (MDRTB). The guidelines were used for recommending processing route and route-dependent excipients to manufacture the drug product as immediate release solid oral dosage form.

The properties of the API were experimentally measured and stored in an information repository as instances of classes in the material ontology. Some of the important properties are Hausner ratio, angle of repose, compressibility, density, and stability. Also, instances of excipient properties based on values available in the literature were created in the repository. When applied to the materials in the repository, the 'Selection_of_Processing_Route' guideline selected roller compaction as the feasible processing route. It eliminated direction compression because of poor flowability of the API and wet granulation because of poor chemical stability of the API. As the first step in selecting excipients, the 'Excipient_Selection_for_Roller_ Compaction' guidelines identified the type of excipients needed in this case (flow aid, filler and lubricant).

In summary, the guideline modeling framework provides a systematic approach to gather and present the procedural knowledge. The core-building blocks were defined



For Individual ◆ Is_Flowability_Very_Bad (instance...

Name: Is_Flowability_Very_Bad

expression: ◆ Is_Hausner_Ratio_greater_than_1.75

user_comment: For DC, the flowability of API needs to be good. Poor flow may cause two problems: 1. Content Uniformity and 2. Handling.
Addition of flow aids may improve the flowability however if the flow is extremely poor and the dose is high granulation may be needed.

For Individual ◆ Is_Hausner_Ratio_grea...

Name: Is_Hausner_Ratio_greater_than_1.75

specification: (Hausner_Ratio > 1.75)

Get_Data_Items: ◆ Get_Data_Hausner_Ratio

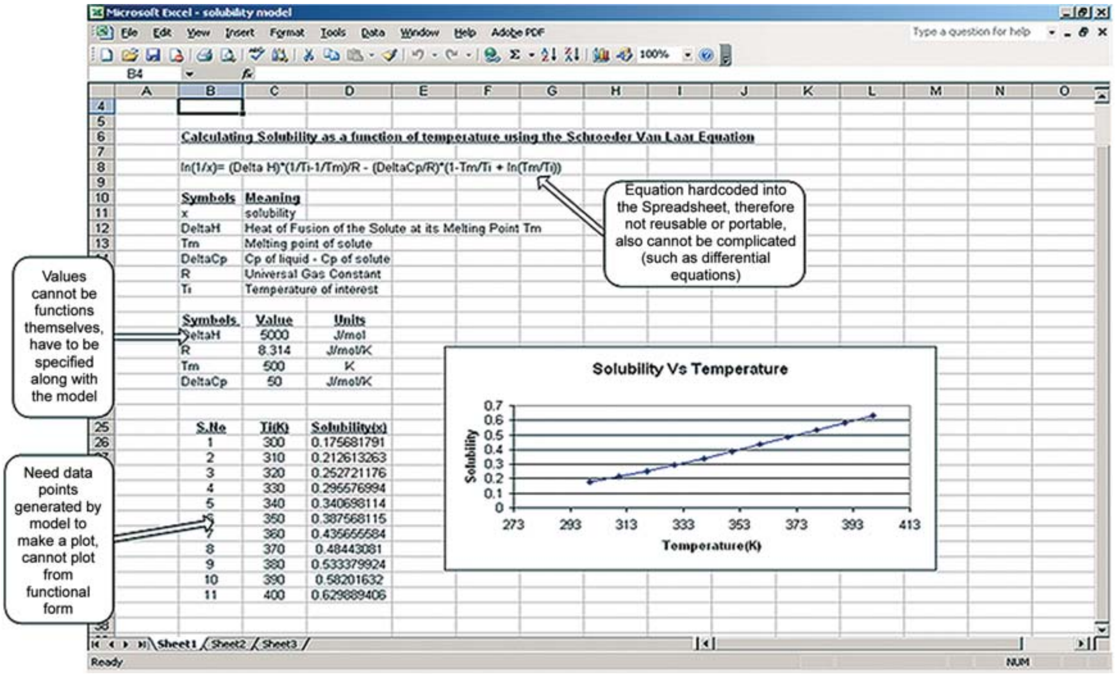
For Individual ◆ API-1 : Hausner Ratio...

Belongs_to_Material: ◆ API-1

hasValue: ◆ 1.89 : [,]

Figure 6. Details of a decision step in a guideline. Decision criterion of decision step is represented as an instance of expression. Also the property value used in expression is directly linked to material ontology.

Continued on page 32.


Microsoft Excel - solubility model
File Edit View Insert Format Tools Data Window Help Adobe PDF
Type a question for help
100%
B4
A B C D E F G H I J K L M N O
Solubility Vs Temperature
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Solubility
273 293 313 333 353 373 393 413
Temperature(K)
Sheet1 Sheet2 Sheet3
Ready
NUM

Calculating Solubility as a function of temperature using the Schroeder Van Laar Equation

ln(1/x)= (Delta H)*(1/Ti-1/Tm)/R - (DeltaCp/R)*(1-Tm/Ti + ln(Tm/Ti))

Symbols	Meaning
x	solubility
DeltaH	Heat of Fusion of the Solute at its Melting Point Tm
Tm	Melting point of solute
DeltaCp	Cp of liquid - Cp of solute
R	Universal Gas Constant
Ti	Temperature of interest

Symbols	Value	Units
DeltaH	5000	J/mol
R	8.314	J/mol/K
Tm	500	K
DeltaCp	50	J/mol/K

S.No	Ti(K)	Solubility(x)
1	300	0.175681791
2	310	0.212613263
3	320	0.252721176
4	330	0.295576994
5	340	0.340698114
6	350	0.387568115
7	360	0.435655584
8	370	0.48443081
9	380	0.533379924
10	390	0.58201632
11	400	0.629889406

Equation hardcoded into the Spreadsheet, therefore not reusable or portable, also cannot be complicated (such as differential equations)

Values cannot be functions themselves, have to be specified along with the model

Need data points generated by model to make a plot, cannot plot from functional form

Figure 7. Mathematical knowledge stored in Excel.

as ontology, and the Protégé ontology editor was used for creating and viewing the guidelines. Compared with human experts, knowledge in the form of guidelines is permanent and transferable. In a large search space, decision making is faster and more consistent. The semantics embedded in the ontology facilitates automatic execution of a guideline through the guideline engine. Apart from helping domain experts in decision making, guidelines also can help them to understand better the development process and training.

Mathematical-knowledge modeling

A large amount of knowledge used in pharmaceutical product development is in the form of mathematical equations. This is referred to as mathematical knowledge. Compared with other forms of knowledge, such as rules and guidelines, mathematical knowledge is more abstract and highly structured [30]. So far, most of the mathematical knowledge is either embedded in specific software tools, such as unit operation models in simulation software, or has to be entered into a more-general mathematical tool following a specific syntax, such as MATLAB or Mathematica. However, much of this knowledge concerns specific applications and is expressed procedurally rather than declaratively.

$$M_t = M_o - kt$$

$$k = \frac{\rho_g}{\rho_s} \frac{C_{pg}(T_{inlet} - T_{exit})}{H_v} \frac{U_o}{(1-\varepsilon_m)L_m}$$

$$M_t = \frac{6}{\pi^2}\left[\sum_{n=1}^{\infty} \frac{1}{n^2} \exp\left(-n\pi^2 \frac{D_m t}{R^2}\right)\right]\left(M_o^{'} - M_\infty\right) + M_\infty$$

Figure 8. Model for fluid-bed drying [32]. First two equations model the first stage of drying process, which is vaporization of moisture on the surface, and the last equation models the second stage, which is diffusion of water out of particles.

As an example to illustrate a widely adopted approach, Figure 7 shows the solubility of a drug substance as a function of temperature using the Schroeder Van Laar equation embedded in Excel [31]. To calculate the solubility, values of properties such as heat of fusion, heat capacity, and melting point of solute are specified in cells, and the Van Laar equation is embedded in each cell that contains the solubility. The only way to use this model is to run it in Excel, by manually changing the values of variables and observing solubility as a function of temperature. In general, it is difficult to use knowledge embedded in a mathematical model for decision support or carry out what-if analysis automatically.

Another common practice is to implement a model in general mathematical tools, such as MATLAB or Mathematica, to use the equation-solving and visualization capability of these tools. For example, Figure 8 shows the equations and variables for a fluid-bed drying model [32]. In this model, the drying process is divided into two stages: the first stage is the vaporization of moisture on the surface, whereas the second stage is the diffusion of water out of the particles. These two equations describe the relations between the moisture content and time for the two stages. The relations also depend on material properties, equipment parameters, and operating conditions.

In this approach, the model equations are written in a syntax that is specific to the solver. Therefore, the users have to be familiar with the use of the syntax designed for the solver being used. In summary, the severe limitation of the two approaches used in modeling mathematical knowledge is the re-usability. It would be difficult for users other than the developers and decision-support systems to utilize the knowledge in these forms.

The information technologies are transforming how mathematical knowledge is modeled, communicated, and applied. Mathematical-knowledge management, a new interdisciplinary field of research, has attracted researchers from mathematics, computer science, library science, and scientific publishing. Marchiori [33] provides a general account of technologies such as XML, RDF, and OWL to foster the integration of mathematical representation and semantic Web. By doing so, it becomes possible to integrate various mathematical sources, to search globally for existing models, to associate metadata as context, and to integrate with other forms of knowledge. Caprotti *et al.* [34] discussed a mechanism for encoding information on mathematical Web services to identify automatically a tool that satisfies the requirement for performing a particular task. This mechanism uses OWL and Description Logic reasoning capability. The management of mathematical knowledge can be divided into four activities [30]:

- Access: searching and making queries, performing deductions and computations.
- Dissemination: the mathematical knowledge needs to be disseminated so that it can be distributed as text in traditional journals and textbooks, or digitally stored and provided on the Web, and incorporated into mathematical software systems such as computer theorem-proving systems and computer-algebra systems.
- Organization: articulated mathematical knowledge needs to be carefully organized to capture connections and avoid redundancy.
- Articulation: using an expression language with the context within that is understood and a representation by which it is conveyed.

The proposed approach

In the proposed approach, the declarative and procedural parts of the mathematical knowledge are separated. The declarative part consists of the information required by the model to run, the information generated from the model, and the model equations. The variables used in the model equations are formally defined consisting of the corresponding symbols in the equations and the link to the information resources. Model ontology is created to describe information related to a model, including input, output, assumptions, and equation sets. A specific model is created as an instance of the model ontology. Mathematical markup language (MathML), which is based on XML, has been created as a standard way of describing mathematical equations. There are two dialects in MathML (http://www.w3.org/TR/

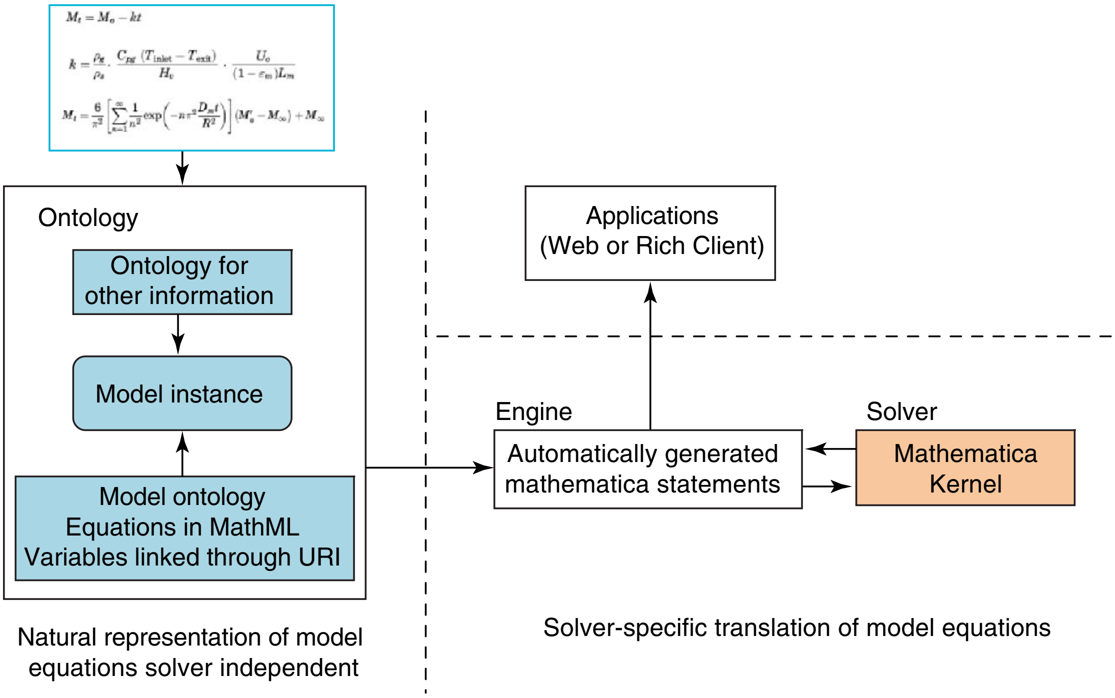

$M_t = M_o - kt$
$k = \frac{\rho_g}{\rho_s} \cdot \frac{C_{pg}(T_{\text{inlet}} - T_{\text{exit}})}{H_v} \cdot \frac{U_o}{(1-\varepsilon_m)L_m}$
$M_t = \frac{6}{\pi^2}\left[\sum_{n=1}^{\infty}\frac{1}{n^2}\exp\left(-n\pi^2\frac{D_m t}{R^2}\right)\right](M_o' - M_\infty) + M_\infty$
Ontology
Ontology for other information
Model instance
Model ontology
Equations in MathML
Variables linked through URI
Applications
(Web or Rich Client)
Engine
Automatically generated mathematica statements
Solver
Mathematica Kernel
Natural representation of model equations solver independent
Solver-specific translation of model equations

Figure 9. The proposed approach for modeling mathematical knowledge. Fluid-bed drying model shown in Figure 8 is represented using model ontology that is independent of solver.

Continued on page 34.

MathML2/): the presentation markup, which concentrates on displaying the equations, and the content markup, which concentrates on the semantics (meaning) of the equations. OpenMath, which is another XML-based language, is developed to extend Content MathML beyond its basic scope, by defining an abstract model for constructing mathematical objects from combinations of these symbols.

The procedural part consists of solving the model equations. Mathematica was used as a general purpose solver. It has several features that can be used directly in the proposed approach: (i) symbolic processing capability, which handles equations in MathML formats without translating them into procedures as in other general mathematical packages; (ii) extensibility with programming languages such as Java, which facilitates communication between the Mathematica kernel and the engine; and (iii) Web enabled, which provides through a Web browser an environment to access the functionalities of the mathematical models.

Each model is an instance of the model class of the model ontology (Figure 9). A model ontology enables representation of mathematical equations in a form that is independent of the solver. The solution of the model equation is governed by the context in which that model is used. For example, suppose that the drying model described earlier is to be solved in a stand-alone mode using Mathematica. An engine must be written to translate all model equations into Mathematica statements, and to retrieve values for parameters describing the given dryer and operating conditions from information repository. With this information, the engine invokes the Mathematica kernel, which solves the equations and returns the values to the engine. The engine then displays the computed values in the desired form. A similar engine must be developed if a different solver is used. In this work, an engine was written as a middle layer to interact with the Mathematica kernel and Web pages from which information on a model instance is presented. Additionally, the output generated by Mathematica in tabular or graphical form is displayed on the Web page.

This approach provides a systematic way for model creators to describe the models in terms of equations with the help of the intuitive and visual equation editor, and the variables that are described using the ontology and linked to the information resources. The MathML description of the equations and the ontologies provide an open and solid foundation for the engine to understand the equations and variables, along with the links to access the values of the variables during execution of the model. The equation solving is performed by Mathematica.

Integrated decision-support system

The scope of the proposed infrastructure and the decision-support system are summarized in Figure 10. The major components deal with unstructured information, structured information, and various forms of knowledge. Only with explicit and formal semantics, which makes the information machine processable, tools can better use the information and provide better functionalities. Ontology was used to model the information. We also have demonstrated an ontology-

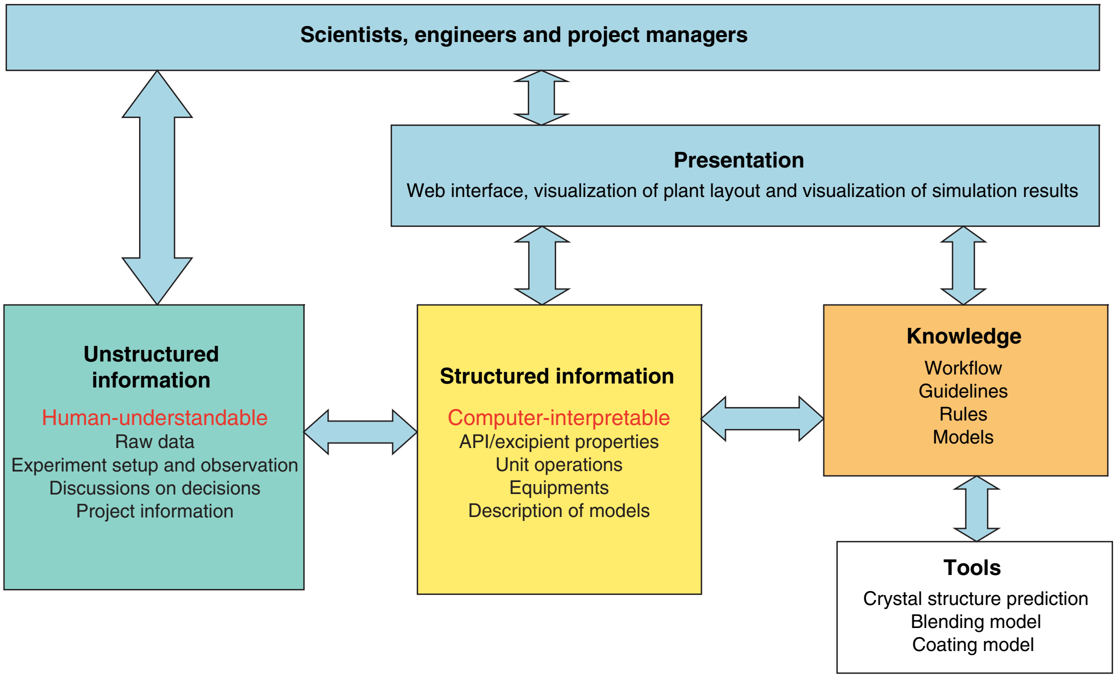


Figure 10. The scope of the proposed integrated decision support system.

driven information management approach based on the developed ontology. A structured information layer, based on the ontologies to model the information in pharmaceutical product development, is created on top of unstructured information to provide the semantics. Structured information can be used directly by tools that assist decision making.

Various forms of knowledge are modeled. Ontology is used for modeling the declarative part of the knowledge. We have demonstrated that ontology based modeling of the guideline knowledge and mathematical knowledge makes creation of the knowledge intuitive, and its utilization convenient for users and by knowledge engines. The proposed intelligent decision-support system has been developed and demonstrated using the reformulation of an MDRTB drug. This systematic approach to model and use the information and knowledge in pharmaceutical product development makes it possible to capture the rationale for the development process. This rationale can support development of new products using case-based reasoning or data-mining technologies.

Conclusions

We have described an integrated framework that facilitates the flow of ideas through information modeling, broadening the search horizon through incorporation of mathematical modeling and heuristics. The proposed system also archives the knowledge from successes and failures through knowledge modeling, thereby accelerating product development and ensuring quality of design. The system that has been developed provides an ontological, information-centric approach to model information and knowledge with Semantic Web providing a general framework for implementing the infrastructure. The ontological informatics infrastructure proposed in this article is the dawn of a new paradigm for representing, analyzing, interpreting, and managing large amounts of complex and varied information for product development and manufacturing. Considerable intellectual and implementation challenges lay ahead but the potential rewards will completely transform how to develop and manufacture pharmaceutical products in the future.

References

1. Yu, L.X. *et al.* (2004) Applications of process analytical technology to crystallization processes. *Advanced Drug Delivery Reviews* 56, 349-369.
2. Rowe, R.C. and Roberts, R.J. (2002) Expert systems in pharmaceutical product development. In *Encyclopedia of Pharmaceutical Technology* (2nd edn), pp 1188-1210, Marcel–Dekker.
3. Ramani, K.V. *et al.* (1992) An expert system for drug preformulation in a pharmaceutical company. *Interfaces* 22, 101-108.
4. Rowe, R.C. and Colburn, E.A. (2000) Generating rules for tablet formulation. *Pharmaceutical Technology International* 12, 24-27.
5. Rowe, R. (2000) The right formula. *Chem. Ind.* 24, 465-467.
6. Podczeck, F. (1992) Knowledge based system for the development of tablets. *Proceedings of the 11th Pharmaceutical Technology Conference*, 1, 240-264.
7. Skingle, B. (1990) An introduction to the PFES Project. In *Proceedings of the 10th International Workshop on Expert Systems and Their applications*, pp 907-922.
8. Wood, M. (1991) Expert systems save formulation time. *Laboratory Equipment Digest*, 17-19.
9. Bateman, S.D. *et al.* (1996) The development and validation of a capsule formulation knowledge-based system. *Pharmaceutical Technology* 20, 174-184.
10. Rowe, R.C. *et al.* (1998) Film coating formulation using an expert system. *Pharm. Tech. Europe* 10, 72-82.
11. Stricker, H. *et al.* (1994) The galenic development system Heidelberg/systematic development of dosage forms. *Pharm. Ind.* 56, 641-647.
12. Rowe, R.C. *et al.* (1995) Expert system for parenteral development. *PDA J. Pharm. Sci. Technol.* 49, 257-261.
13. Frank, J. *et al.* (1997) Knowledge-based assistance for the development of drugs. *IEEE Expert-Intelligent Systems & Their Applications* 12, 40-48.
14. Rowe, R.C. (1996) Applying neural computing to product formulation. *Manufacturing Chemist* 67, 21-23.
15. Lai, S. *et al.* (1995) An expert system for the development of powder filled hard gelatin capsule formulations. *Pharm. Res.* 12, S150.
16. Lai, S. *et al.* (1996) An expert system to aid the development of capsule formulations. *Pharmaceutical Technology Europe* 8, 60-68.
17. Rowe, R.D. *et al.* (1999) Case-based reasoning – a new approach to tablet formulation. *Pharmaceutical Technology Europe* 11, 36-40.
18. Guo, M. *et al.* (2002) A prototype intelligent hybrid system for hard gelatin capsule formulation development. *Pharmaceutical Technology*, 9, 44-60.
19. Hohne, B.A. and Houghton, R.D. (1986) An expert system for the formulation of agricultural chemicals. In *Artificial Intelligence Applications in Chemistry, American Chemical Society*, pp 87-97.
20. Zhao, C. *et al.* (2005) Pharmaceutical informatics: a novel paradigm for pharmaceutical product development and manufacture. *Proceedings of 15th European Symposium on Computer Aided Process Engineering*, 1561-1566.
21. Gruber, T.R. (1993) A translation approach to portable ontology specification. *Knowledge Acquisition* 5, 199-220.
22. Noy, N.F. and McGuinness, D.L. (2001) Ontology development 101: a guide to creating your first ontology. *Stanford knowledge systems laboratory technical report* KSL-01-05.
23. Paszko, C. and Pugsley, C. (2000) Considerations in selecting a laboratory information management system (LIMS). *Am. Lab.* 9, 38-42.
24. Grauer, Z. (2003) Laboratory information management systems and traceability of quality systems. *Am. Lab.* 9, 15-18.
25. Zall, M. (2001) The nascent paperless laboratory. *Chemical Innovation* 31, 15-21.
26. Pavlis, R. (2005) Why doesn't a traditional electronic laboratory notebook work in a QA/QC lab? *Scientific Computing & Instrumentation* 22, 31-32.
27. Noga, M. and Kruper, F. (2002) Optimizing content management system pipelines. *Lecture Notes in Computer Science* 2487, 252-267.
28. Bicer, V. *et al.* (2005) Archetype-based semantic interoperability of Web service messages in the healthcare domain. *Int. Journal of semantic and Web Information Systems* 1, 1-22 .
29. Peleg, M. *et al.* (2004) Guideline interchange format 3.5 technical specification. InterMed Collaboratory.
30. Farmer, W.M. (2004) MKM: a new interdisciplinary field of research. *ACM SIGSAM Bull.* 38, 47-52.
31. Lachman, L. et al. (1986) Liquids. In *The Theory and Practice of Industrial Pharmacy*, 3rd edn, Philadelphia, Lea & Febiger, p 461.
32. Kunni, D. and Levenspiel, O. (1969) Design of Physical Operations. In *Fluidization Engineering*, Wiley, pp 425-426.
33. Marchiori, M. (2003) The Mathematical Semantic Web. *Lecture Notes in Computer Science* 2594, 216-224.
34. Caprotti, O. *et al.* (2004) Mathematics on the (Semantic) NET. *Lecture Notes in Computer Science* 3053, 213-224.
35. Rowe, R.C. (1993) Expert-systems in solid dosage development. *Pharm. Ind.* 55, 1040-1045. JPI