AN INTRODUCTION TO SEMANTIC MODELING FOR LOGISTICAL SYSTEMS

by

David L. Brock
Massachusetts Institute of Technology

Edmund W. Schuster
Massachusetts Institute of Technology

Stuart J. Allen
Penn State University

and

Pinaki Kar

INTRODUCTION

The underlying success of logistics depends on the flow of data and information for effective management. Since the 1960's, the decreasing cost of computer hardware, the development and application of sophisticated mathematical models, and the advent of low cost data collection methods such as bar codes, all have combined to drastically improve temporal and spatial utility (Coyle, Bardi, and Langley 1992; Simchi-Levi, Kaminsky, and Simchi-Levi 2002). These advances have placed logistics at the forefront of management for many firms.

Recent developments, such as Auto-ID technology (Brock 2000a; Dinning and Schuster 2003; Sarma, Brock, and Ashton 2000) will further increase the amounts of data available for the analytics of business decision-making by using computing systems that sense and interact with the physical world. Such computing systems open new opportunities for logistics management in terms of track and trace (Koh et al. 2003; Schuster and Koh 2004), theft detection (Koh et al. 2003), improved service parts inventory management (Kar, Li, and Schuster 2003), and the control of production and logistics within military and civilian supply chains (Engels et al. 2004). However, analyzing the large volume of raw data (including real-time telemetry) produced by Auto-ID technology in an orderly way requires the additional use of new mathematical models to provide representations and understanding.

The process of building mathematical models often lacks productivity because development seldom follows a linear path (Willemain 1994, 1995) and because separate natural, mathematical, and computer representations are needed for managers, model builders, and computer programmers (Geoffrion 1987). This increases the need for detailed interfacing. As a result, implementing mathematical models is complex, time consuming, and requires advanced technical capabilities and infrastructure. Although there is a strong history of applying models to help managers make decisions about intricate logistical systems, specialists often develop these comprehensive models internally within business organizations or academia. This is commonly an application specific job and the same model building technique must be re-invented afresh for each new situation. Though internal development can lead to significant breakthroughs, this approach depends on trial and error to find what works in practice, combined with mathematical intuition and an extensive knowledge of technical publications.

Beginning in the 1980's, software companies started to embed models into software packages installed on network servers, enabling organizational-wide modeling ability. This approach improved the productivity of modeling by reducing development and implementation time, but limited users to a relatively small set of proprietary methods for problem solving. In all cases, internal development, or packaged software, models have become highly structured with few opportunities for creative applications. Proprietary systems also reduce the possibility of sharing of models between business applications that exist outside the computing environment in which the original model implementation took place.

Part of the problem traces to traditional thinking about information theory. Computers today are faster, memory cheaper, and bandwidths plentiful, yet the tasks performed on these machines such as e-mail, documentation, and data storage, are nearly the same as ten years ago. Computers primarily store, manipulate, and transmit data to people. Unless there is direct human interaction, computers essentially do nothing.

Yet computers have far greater unrealized capability. It is possible to design large-scale Internet systems that might allow computers to store and analyze vast quantities of information and to share these results automatically with other computers throughout the world. Networks of computers have the potential to operate independently or collectively, without human interaction.

The failure to take full advantage of the computer's potential lies not in the hardware or communications technologies, but in lack of languages and standards that allow systems to share data and interface models across multiple applications and domains.

In this article, we discuss a proposed standard for a language and protocol that will enable computers to describe and share models and to assemble new models automatically from a general repository (Brock 2003a, 2003b). This will substantially increase the Clockspeed (Fine 1998) of modeling, and the computational efficiency of applying models to perform the functions of "sense," "understand," and "do" that comprise the underpinning of creating smart objects within supply chains. The new computer language infrastructure we propose includes open standards with two specific purposes: 1) communication of models between computers to create interoperability, and 2) to run distributed models across the Internet. In many ways, this effort challenges the longstanding philosophy of modeling that emphasizes individual effort in formulation and implementation. The ultimate goal is to build an integrated modeling structure for accelerating the development of new applications.

The balance of this article describes our thoughts about designing a network for abstract objects like models. In a sense, this effort is a step beyond linking the physical world, the underlying concept that has made Auto-ID technology successful. Networks, of physical objects or abstractions like models, share the premise that leaps in productivity arise from the free flow of information. Creating an *Intelligent Modeling Network* will accelerate the flow of information to the great advantage of many practitioners in the field of logistics.

SEMANTIC BASED INTERNET SEARCH

The existing standards of the Internet do not provide any semantics[a] to describe models precisely or to interoperate models in a distributed fashion. For the most part, the Internet is a "static repository of unstructured data" that is accessible only through extensive use of search engines (Fensel et al. 2003, p. 377). Though these means of finding data have improved since the inception of the Internet, human interaction is still required and there are substantial problems concerning semantics. In general, "HTML does not provide a means for presenting rich syntax and semantics of data" (Fensel et al. 2003, p. 7).

For example, one of the authors of this article recently did a search for "harvest table, oak" hoping to find suppliers of home furniture. Instead, the search yielded a number of references to forestry and the optimal time to harvest oak trees. Locating the URLs relating to furniture required an extensive review of a number of different web sites. This process of filtering can only be accomplished though human interdiction and is time consuming.

With inaccurate means of doing specific searches based on one semantic interpretation of data, information, or models, it is nearly impossible for the Internet to advance as a productive tool for logistical modeling. Because of this situation, Internet searchers for logistics models that match to the data at hand are impossible to accomplish in practice using the tools currently available.

Several Types of Webs

The problem of semantics arises from the fact that keywords are the means used to describe the content of web pages. Each keyword can have multiple meanings, creating a situation of great difficulty when attempting to accomplish an exact search. The difficulty increases by an order of magnitude when attempting to do phrase-based searches. Without exact search capability, it is impossible to create any sort of machine understandable language for the current *Web of Information*.

---

[a] Semantics refers to the precise meaning of data or models in a machine and human understandable way. Once a semantic is established, it can be used as a descriptor.

Even though the search engine issue has not been resolved, industry forces are pushing for a new type of Internet characterized as the *Web of Things*. Driven by developments in Auto-ID technology and ubiquitous computing, the *Web of Things* aims to link physical objects to the internet using Radio Frequency Identification (RFID) tags as real-time communication devices and to "shift from dedicated computing machinery (that requires user's attention, e.g., PC's) to pervasive computing capabilities embedded in our everyday environments" (Fensel et al. 2003, p. 363).

Aiding this effort is EPCglobal, Inc.,[b] an international standards organization formed by the Uniform Code Council (UCC), and European Article Numbering (EAN) Association (known in the industry as GS1). The group administers the Electronic Product Code (EPC) numbering system, which provides the capability to identify an object uniquely (Brock 2000b). With serial identification for physical objects, searches accomplished through Internet search engines or proprietary IT infrastructures will become much more effective in finding an exact match. This provides the ability to do track and trace across entire supply chains and other computerized functions important to logisticians. Linking the physical world, using Auto-ID technology and ubiquitous computing,[c] will form the basis for a revolution in commerce by providing real-time data and enabling smart objects[d] (Schuster and Brock 2004; Schuster et al. 2004; Schuster, Engels, and Allen 2004). In addition, real-time data also provides the opportunity for improved calculation of such things as supply risk (Allen and Schuster 2004).

As impressive as the effort to create the *Web of Things* has become, it still does not address the question of semantics in describing objects beyond the use of a simple serial number. There exist a large number of abstractions, such as mathematical models, that cannot be characterized by a unique serial number no matter how sophisticated the syntax. Without the ability to provide unique identification of an abstraction, the Internet will serve little useful purpose in linking mathematical models together in a way similar to the manner that the *Web of Things* will eventually link the physical world.

In the future, the definition of a model and the sharing of models though a network will become as important as the model itself. To accomplish this higher goal, the Internet must become a *Web of Abstractions*, in addition to a *Web of Information* and a *Web of Things*.

Creating a *Web of Abstractions* requires a semantic definition of models that is precise and can be machine understandable. Given this capability models can be searched, organized, categorized, and executed – sequentially and in parallel – creating multiple, large-scale interoperable environments. These interoperable modeling environments will exist only in virtual reality and offer the potential for creating a dynamic meta-structure[e] for specific classes of models.

---

[b] EPCGlobal, Inc., http://www.epcglobalinc.org/.

[c] "Ubiquitous computing" refers to the wide spread use of computers for sensing and automatic decision-making. With advances in technology, both the size and cost of computers has been reduced, opening a new range of applications. For an early reference on ubiquitous computing see Weiser (1991).

[d] "Smart objects" are things that can make decisions independently based on external data gathered though sensing technology combined with computer logic imbedded into the object.

[e] A meta-structure for logistics models would show how each model relates to other models. With such a mapping, the ability to search increases dramatically.

Through a *Web of Abstractions*, models can be matched much more quickly to practical problems, along with the available data, and shared beyond single end-user applications. This capability is of great value to both practitioners and researchers who are interested in gaining the maximum value in modeling logistics for practical decision-making.

The Representation of Model Schema

Previous research in computer science consistently states that the missing structure needed to create a *Web of Abstractions* is an ontology. Simply stated, "an ontology specifies what concepts to represent and how they are interrelated" (Fensel et al. 2003, p. 34). This structure provides order when conducting searches and serves the important purpose of creating a crude form of intelligent behavior. For example, one group of researches involved in the early aspects of using computers to create Artificial Intelligence concluded that "…the clue to intelligent behavior whether of men or machines, is highly selective search, the drastic pruning of the tree of possibilities explored" (Feigenbaum and Feldman 1963, p. 6). Properly constructed, the ontology reduces search time for abstractions creating a free flow across a network. With the hundreds of logistics models that do not find widespread application in practice, the capability to conduct a quick and accurate search improves the chances that more applications will occur.

In using an ontology to organize abstractions like mathematical models for machine understandable searches, there are two important aspects to consider.

First, the ontology assumes that a semantically precise definition of an abstraction (model) exists. Absence of this in the current schema presents a problem in that the classification of mathematical models depends on keywords that might have different meanings under different contexts e.g., planning and scheduling.

Second, the ontology also serves an indirect definitional function in that meaning arises by the way one model is connected or related to other models. This is important in visualizing the big picture of the relationships between different logistical models. It also drastically decreases search time by reducing the number of possibilities in reaching an exact semantic match. However, there are significant drawbacks concerning the establishment of an ontology for logistics models.

The Limitations of Representing Models Using Ontologies

By definition, ontologies are rigid and inflexible, and assume one absolute definition exists for each knowledge element. The idea is to establish a set structure of definitions and relationships between different abstractions (models) that are canonical[f] and eternal. This means that the usefulness of an ontology for logistics modeling depends on intensive study of the canon put forth. It is unrealistic to believe that any independent body of academics or logistics practitioners could formulate an all-inclusive canon that would stand the test of time. The ontology approach is a throwback to the

---

[f] Canonical refers to a common set of rules used to classify the relationship between things.

philosophy of Scholasticism that dominated Western thought during the high middle ages.[g] During this time, all problems of physical science and philosophy were thought to be solvable through a rigorous and rigid framework of logic founded in the writings of Aristotle and governed by strict rules of disputation. Lacking an experimental method, Scholasticism often led to false conclusions that persisted for hundreds of years. History has proven that canonical structures, meant to organize and communicate knowledge, often have the unintended outcome of restricting the adoption of further innovations that exist outside the bounds of the canon.

In addition, rigid ontological structures lack the ability to adapt based on inductive reasoning. There is no ability to learn automatically from specific examples that occur through time and generalize to form a new element of knowledge contained in the ontology. This was the major limitation of expert system architectures and a leading reason for the decline in the application of expert systems in practice.

A final major drawback involves the difficulty in merging separate, distinct ontologies into a whole. For all the advantages of a rigid structure in organizing abstractions (models) and reducing search time, there is no easy translation or interface to integrate two different classes of models. We believe that advances will only take place through the free exchange between widely disparate fields of modeling. Without this ability, efforts in establishing computer languages to share and interoperate models will be difficult.

A Relative Approach to Model Representation

To overcome the disadvantages of traditional ontologies in computer science, we advocate the abandonment of a single, unified structure to represent abstractions (models). The reality is that the representation of objects and their interrelation is almost entirely dependent on a person's viewpoint. In other words, as opposed to a single ontological representation for models, we propose a more flexible means of description, so that others may construct their own particular representations and unique ways for connecting them together.

Furthermore, our approach provides the means for building dynamic, "on-the-fly" model taxonomies; that is, hierarchical organizations of models that are generated as a function of an individual's point of view. In our system, there is no one classification scheme (ontology), but multiple. Simply put, several ontologies can exist simultaneously with no contradictions.

With this approach, a model is an atomic element that may subscribe to one or more classification hierarchies. These taxonomies may be mutually agreed industry standards – essentially commercial *data dictionaries*, proprietary schemes or dynamically generated groupings for particular applications. In all cases, the representations, relations, and organization of models will be dynamic and configurable to the task. Later in this article, we provide an example of model representation that is integral to our view of the schema needed to create the *Web of Abstractions*.

---

[g] Scholasticism reached its peak about 1200 AD and was widely taught in the Universities of Western Europe.

In the next two sections, we discuss the practical and theoretical aspects of combining advances in computer science with the existing body of mathematical models that have been developed by logistics researchers over a period of many years. The prospect of doing Semantic Modeling for logistics applications on a large scale draws upon the intersection between computer science and logistics practice.

SEMANTIC MODELING

Most would agree that modeling is a craft industry analogous to the production of automobiles prior to the advent of the assembly line (Willemain 1994). Although models are ubiquitous management tools, they are, for the most part, isolated from one another. In other words, a model from one domain, such as weather forecasting, does not interact with another, such as logistical systems.

The reason for this is obvious. Until very recently humans were the only ones who built, used, and shared models. Our limited cognitive ability naturally restricts the number and diversity of models we can accommodate. Computers, on the other hand, have the ability to execute and communicate models with vast numbers of other computers. With ever increasing processing power, data storage, and networking bandwidth, the computing grid is poised to revolutionize our ability to understand and manage the physical world. The Internet with its standards and languages provides the backbone for communication, but does not provide the mechanism for describing and integrating diverse models. The future is a form of modeling on demand similar to other efforts in establishing a computer grid that resembles electric power distribution (London 2003).

Our goal is to turn modeling into a mass production system based on standardization, scale, and interoperability. In summary, this means that a Semantic Modeling language capable of achieving this functionality must include:

1. "A formal syntax and formal semantics to enable automated processing of their content."
2. "…a standardized vocabulary referring to real world semantics enabling automatic and human agents to share information and knowledge" (Fensel et al. 2003, p. 8).

Achieving this goal will mean that practitioners can produce models in a timely manner with greater productivity and relevance. This anticipates a new era for computers in terms of insight and awareness and it implies the ability to organize data, and define the inputs and outputs of models in a semantically precise way. With a precise semantic, a single descriptor of a model can be formulated that contains no ambiguity concerning meaning.

The mechanism we put forth to mass produce models and create interoperability draws inspiration from current efforts to improve the search capabilities for the *Web of Information*. The World Wide Web Consortium (W3C) is responsible for initiating select efforts to improve overall web search capabilities.[h] Some of the initial work conducted by W3C forms a reference base for our research in developing and implementing a *Web of Abstractions*.

---

[h] W3C Semantic Web, http://www.w3.org/2001/sw/.

Each abstraction (model) has unique elements that can be defined just as a language has a specific syntax and grammar. Defining these elements alone will be of no benefit unless there is a protocol, or computer language, to communicate and execute the elements of models across a large network like the Internet. Our efforts in establishing Semantic Modeling are grounded in the idea of having data and models defined and linked in a way that can be used by machines not just for display purposes, but also for automation, integration, and reuse across various applications. Accelerating the reuse of model elements across vast networks of users will lead to the mass production of models and great benefit to practitioners. In addition, distributed modeling, a set of geographically separated model elements working simultaneously in parallel, adds additional prospects for large-scale parallel computing. This capability will improve the utilization of desktop computers and provide grids of almost unlimited modeling power.

Though the W3C provides something called a Resource Definition Format (RDF) that defines the basics of representing machine processable semantics (Fensel et al. 2003, p. 9), no formal computer language has been put forth that enables the sharing of models or doing large-scale modeling in parallel. The next section gives an overview of our vision for a computer language and protocols that achieves Semantic Modeling.

SYSTEM ARCHITECTURE

The fundamental idea is to design a family of standards that enable the creation of models that integrate automatically into an interoperable environment. In this way, developers can formulate models within their particular areas of expertise and know that the resulting models will interoperate in a shared environment. We believe it is possible, with sufficient care in the definition, to create such a language that is both precise and expressive in its description yet shows constraint in its breadth to ensure compatibility.

The goal is to create interoperable environments that receive data from the physical world (for example through Auto-ID technology) and then produce inferences, interpretations, and predictions about the current and future states of the environment.

These state data are essential for any automated decision system. In other words, the environmental states (obtained from sense data) can feed into networks of decision-making algorithms with the output being informed decisions and deliberate plans that can be adjusted through time based on additional feedback obtained from the physical world. This type of modeling is essentially the underlying basis for automated control, monitoring, management, and planning.

The proposed architecture is composed of five fundamental components: the Data Modeling Language (DML), Data Modeling Protocol (DMP), Automated Control Language (ACL), Automated Control Protocol (ACP), and The Dictionary. The following lists a description of each component.

Data Modeling Language (DML) is a semantic for describing modular, interoperable model components in terms of individual outputs, inputs, and data elements. Models described with DML should automatically assemble into executable model environments. Although a number of ways exist to depict a model component that is interoperable, we choose to focus on data inputs and outputs as the means of describing a model component. This concept is explored in detail as part of the example presented later in this article.

Data Modeling Protocol (DMP), once a connection between models and data is established, the DMP coordinates the communication sequence between the computing machines that host models in terms of outputs and inputs. This can be described as a computational grid.[i] The DMP exists for the sole purpose of coordinating the operation of two or more models running in parallel on different computer platforms. In some cases, this coordination might take the form of an algorithm that communicates the timing of a model run and the timing of information transmission that will be used as an input to another model running on a separate computing platform.

Automated Control Language (ACL) establishes the connection between models and data based on DML (descriptor of inputs, outputs, and data) and the ACP, which locates the appropriate connections.

Automated Control Protocol (ACP) helps model outputs and inputs locate one another within a network, even though the individual models may exist in different host systems and organizations. The ACP identifies potential connections and takes priority over the DMP, which is a coordinating activity after achieving connections through the ACL.

The Dictionary is a common resource containing words with multiple meanings. The dictionary will utilize established sources such as WordNet, and various specialty dictionaries from the medical field, operations, logistics, and other disciplines.

With this array of protocols and languages, model inputs, and outputs are described through DML by using words from the dictionary to express a precise semantic. Because multiple words, akin to a phrase or simple sentence, best provide accurate descriptions of outputs and inputs for models, we envision the use of graphs to express syntax thus giving a precise semantic meaning.

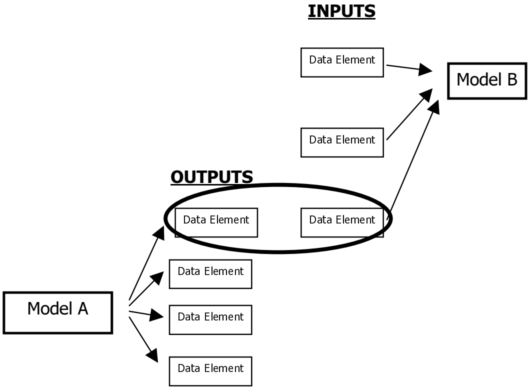
The graphs produced to represent outputs and inputs will need to be of the form that operations, such as sorting, can be applied using computer code. The ACP helps to locate graphs with commonalities that are resident in a network. These commonalities might include: 1) similar structure, 2) an output of one model that might match the input of another model, 3) a connection between a data element and the inputs for a particular model, or 4) a connection between two or more data elements contained within the network. Upon enumeration of appropriate matches, the ACL makes a connection and the DMP coordinates operation in parallel across the separate computing platforms. Figure 1 provides a visualization of the connection between models based on inputs and outputs.

---

[i] As an example, please refer to The Globus Project, http://www.globus.org/.

FIGURE 1

CONNECTING MODELS THROUGH DATA OUTPUTS AND INPUTS



AN EXAMPLE FROM LOGISTICS

Researchers at the 2001 Logistics Educators Conference presented an interesting paper about the implication of advanced planning and scheduling systems (APS) on supply chain performance (Closs and Nair 2001). Even though production scheduling is considered a part of logistics, the authors note that research in this area is somewhat lacking in the context of supply chain management. The article also contained an appraisal of changes needed in academic curriculums to ensure students receive proper education about the role of APS in supply chain management. Based on these comments, we decided to investigate the literature of finite capacity scheduling (FCS), an important sub-segment of APS, to find an initial example for demonstrating the aspects of Semantic Modeling.

In general, there are many solution methods for FCS. A non-exhaustive list includes: mathematical programming, simulation, heuristics, genetic algorithms, neural networks, theory of constraints, and expert systems. Of this list, the first three are frequently found in practice with the most common being heuristics. About 80% of commercial scheduling packages use heuristic solution approaches (Melnyk 1998).

A detailed analysis reveals that each model for FCS exhibits characteristics based on the solution method or algorithms employed (Schuster and Allen 1998). Table 1 summarizes the capabilities of each model in its pure application without modification.

TABLE 1

COMPARISON OF DIFFERENT SCHEDULING APPROACHES

Attribute	Math Programming	Simulation	Heuristic
Hold Time		X	X
Queue Time		X	X
Customer Service		X	
Forecast Bias		X	
Set-up Cost	X		X
Holding Cost	X		X
Overtime Cost	X		X
Capacity	X		X
Production Lot Size	X		X
Production Sequence	X		X
Customer Due Date	X	X	X
Family Structure	X		

X = Functional

Understanding that each model class for FCS listed in Table 1, math programming, simulation, or heuristics, does not fully address all attributes commonly found in commercial FCS problems is important in supporting the belief that future advances will come from combining existing models in new ways to address a wider range of attributes.

A recent article provides substantial background about FCS from the perspective of practical implementation, including several references to a group of models that provide different FCS capabilities (Schuster, Allen, and D'Itri 2000). Essentially the entire group deals with the same scheduling problem. This body of research provides insight for a simple example that highlights how elements from different models can combine to produce new models with better performance, thus demonstrating the importance to practitioners and researchers of developing a computer language and protocols to facilitate this process with some degree of automation.

The example set forth below deals with various types of models used to schedule production for manufacturing lines common to the consumer goods industry. With high demands for customer service, it is important for consumer goods companies to schedule the production of end items with proper consideration given to the risk of being out of stock and the capacity constraints that might limit production in times of peak demand. This activity is a fundamental element of logistics in meeting end-item inventory availability goals set for each manufacturing facility at the lowest possible inventory and set-up cost. To accomplish these goals, the logistical trade-offs are complex and are best decided through the effective use of mathematical models of scheduling. Based on statements made in the literature (Allen, Martin, and Schuster 1997; Allen and Schuster 1994; Schuster and Finch

1990), all of these models were implemented at Welch's, an agricultural cooperative in the Concord grape industry, during a span of fifteen years. The following provides a description of each model:

> **MODEL A – Deterministic Simulation (Schuster and Finch 1990)** – With bias adjusted safety stocks that use customer service levels as an input, production planning occurs for each item independently. Bias refers to the case where the probability distribution of forecast errors is non-normal, a common case in the consumer goods industry where aggressive sales goals sometimes become the forecast. All items run on a production line are summed to give a total capacity load. This model initially assumes infinite capacity is available for production and does not consider set-up or inventory carrying cost. However, the model does provide a method for safety stock planning that considers dynamic forecasts and the impact of forecast bias in planning safety stock levels. For this model, a technique initially developed by Krupp (1982) was subsequently modified for calculating the effect of bias on safety stock levels.
>
> **MODEL B – Mathematical Programming (Allen and Schuster 1994)** – Exploiting the fact that consumer goods have a family structure defined by package size, production can be planned using a two-tier hierarchical structure where product families are sequenced with disaggregation taking place to form end item schedules. This approach provides near optimal solutions based on cost and utilizes an innovative mathematical formulation that yields near instantaneous solutions to mixed integer math programming (MIP) problems. Previously, MIP as a solution method received criticism from practitioners because it was impossible to predict run time of the model. This formulation is a step toward overcoming the run time problem.
>
> **MODEL C – The MODS Heuristic, Sequence Independent (Allen, Martin, and Schuster 1997)** – An approach to scheduling using the Modified Dixon Silver (MODS) method to calculate near optimum production schedules based on inventory and set-up costs, and inventory set-up time.
>
> **MODEL D – The MODS Heuristic, Sequence Dependent (D'Itri, Allen, and Schuster 1999)** – Building on the Modified Dixon Silver method, this approach utilizes the nearest neighbor variable origin (NNVO) heuristic as a second step to sequence production based on a "from-to" table of changeover costs between items.

Relationship to Proposed System Architecture

By looking at working models as an aggregation of interchangeable elements, the possibilities for identifying new combinations becomes very large. Using our system definitions, the DML would describe various elements of models, such as the bias adjusted safety stock method used in MODEL A, that are modular and interoperable. The ACP provides a mechanism for various model elements to locate each other across a network like the Internet. Analyzing the examples of

MODELS A, B, C, and D, it appears that the developers located model elements as a function of many years of study in the FCS area combined with mathematical intuition.

In the situation where distributed modeling takes place, the DMP allows for communication between active models located on separate computing platforms. For example, bias adjusted safety stock (MODEL A) might be calculated on one computing platform with the results being transferred to another platform that contains the MODS heuristic (MODEL C). In this case, the DMP establishes the order to run the models and the timing of data transmissions. The final part of our system architecture is the ACL that would allow communication and location of critical decision-making elements of models (outputs) between different computing platforms. The ACL is needed because the decisions from one model (outputs) might become data (inputs) for another model. This is the case for MODEL A, which can provide safety stocks (output) as an input to MODELS B, C, and D. The ACL matches the outputs of one model to the appropriate inputs for another model.

Establishing Semantics for Logistics Models

The starting point for the goal of building an interoperable system based on DML, ACP, DMP, and ACL is a semantically precise definition of a model. Given that most model descriptions depend on keywords, which might have a number of different meanings, we propose an alternative approach to define a model. The intent of DML is to label models semantically in such a way that common elements can be machine understandable and interoperable.

Our approach to the semantic labeling problem involves forgoing attempts to describe the various algorithms employed in each model and the outputs that relate to decision making. Rather, we focus on the data (inputs) required for each model as a unique base for machine understanding and the grouping of common models together. This assumes that a special, unique relationship exists between a model and its data.

As a practical matter, we believe that definition of a model in terms of data inputs will provide a more precise semantic as compared to definition by attempting to classify the algorithm used for each modular component (model). Keyword definitions for the complex algorithms that comprise models are notorious for having different semantic meanings. In addition, the keyword descriptions often have no meaning at all to business practitioners that do not have extensive formal training in logistics or management science.

Table 2 illustrates how data inputs can become a tool for establishing semantic meaning.

TABLE 2

DATA INPUTS TO MODELS A, B, C, AND D

Data Input	Model A	Model B	Model C	Model D
D1. Beginning Inventory	X	X	X	X
D2. Forecast Demand (by week)	X	X	X	X
D3. Historical Shipments (by week)	X	X	X	X
D4. Historical Forecast (by week)	X	X	X	X
D5. Hold Time (days)[j]	X			
D6. Queue Time (days)	X			
D7. Service Level (% in stock)	X	X	X	X
D8. Set-up Cost ($/changeover)		X	X	X
D9. Set-up Time (hrs/set-up)			X	X
D10. Holding Cost ($/week)		X	X	X
D11. Capacity Limit (hrs/day)		X	X	X
D12. Family Structure (end items per group)		X		
D13. Overtime Cost ($/hr)			X	X
D14. Sequence Dependent Set-up Cost (From–To table of change-over costs)				X

From Table 2 we note that MODELS A, B, C, and D all share the data inputs D1, D2, D3, D4, and D7. This gives a natural way to categorize MODELS A, B, C, and D into the same group. This also implies that models using the same data will deal with the same initial problem (in this case scheduling of production lines for the consumer goods industry) and that all four models are interoperable with respect to the data. Any of the four models could be applied to the same data set to gain the result of a production schedule. The outcome is that by defining a model in terms of its data inputs, a precise semantic results that allows assignment of the model to a common group.

Further, the use of input data as a means of establishing semantics also aids in distinguishing differences between models in a group. Likely, the data inputs for a group of models will not be identical if different solution methods (algorithms) are used. From Table 2 we notice that none of the four models shares the same set of data inputs yet all of these models are capable of producing a schedule for a manufacturing process characteristic of the consumer goods industry. This offers a way to identify differences between models within the same group as categorized by data. This also provides an indirect indication of the solution methods (algorithms) employed.

---

[j] Hold time refers to the period of time a product is quarantined for routine quality checks.

For example, MODELS B, C, and D share the commonality of requiring a capacity limit, inferring that these models belong to a class of FCS systems, and perhaps are interoperable. In another case, Table 2 shows that MODELS A, B, C, and D all have service level as a parameter, implying that this class of models include some aspect of safety stock. Other safety stock models, not mentioned in this example, might offer alternative ways to calculate safety stocks using the same data requirements. Because all of these models share the same set of data inputs they are interoperable with MODELS A, B, C, and D.

The reader must keep in mind that we view models in an atomic elemental way. Taking an example from chemistry, a single element like Calcium (Ca) can become part of many different molecules such as calcium hydroxide (CaOH) or calcium chloride (CaCl) through chemical reactions. In a similar way a single model, for example bias adjusted safety stock (MODEL A), can be combined with MODELS B, C, and D to create entirely new model forms. Data inputs, as part of DML, hold the key for developing an open architecture for models to combine automatically as in chemical reactions.

To summarize, the descriptors we put forth as the basis for DML include data inputs as the primary semantic for grouping models and the initial basis for machine understanding. Model outputs are only important in providing a) general guidance concerning the objective of the modeling effort and b) some definitions of model outputs that may in turn become model inputs in other situations. We do not believe that semantic description of algorithms based on keywords will play a significant role in the design of DML. One important means of classification that we have not mentioned involves the assumptions of the model. The use of assumptions as a precise semantic of a model provides an interesting area for future research.

An Example of Multiple Ontologies

As an illustration of the fact that multiple ontologies exist with respect to the definition of a model and its relationship to other models, we now examine a final example involving MODELS A, B, C, and D.

Depending on viewpoint, the library of models could be used in two different ways:

– From a **production planner** standpoint, the models could provide a computer generated schedule of the timing and amount of production needed at a manufacturing plant given a specific beginning inventory, end item demand forecast and target safety stock levels.

– From a **supply chain manager** standpoint, the models could provide an accurate projection of inventory levels in plant warehouses given a specific beginning inventory, end item demand forecast, and target safety stock levels. This information could be used for space requirements and as a basis for budgeting operational costs.

There is evidence that all four models have been used as a tool for production planning (Allen, Martin, and Schuster 1997; Allen and Schuster 1994; D'Itri, Allen, and Schuster 1999; Schuster and

Finch 1990). In addition, the authors of this article have applied Model A as a means of calculating budgets and operational space requirements for warehousing.

This brief example shows that the same library of models has different meanings and different relationships depending on the viewpoint of end users. This aspect of relative relationships makes the establishment of rigid ontologies difficult to achieve. In practice, there are few proven approaches to merge separate ontologies although some new standards such as Web Ontology Language[k] (OWL) make the claim that merging can take place. Though we have an idea how to handle this obstacle in producing machine understandable semantics, there certainly needs to be more research conducted in this area before totally abandoning the ontology architecture for a relative approach.

It appears that the key to building multiple ontologies depends on the relationships between models. When faced with systems characterized by intricate relationships, engineers sometimes employ graph theory to provide representations for complexity. Using this approach, we believe the links of the graph hold the answer to establishing different ontologies for the same group of models.

CONCLUSION

In this article, we have made an argument for the creation of an interoperable system focusing on models used in logistics. Through a framework of computer languages and protocols, we put forth an approach to achieve Semantic Modeling in practice along with examples using planning and scheduling models from actual situations in industry. Though the prospect for making connections between models is interesting, several important challenges remain concerning practical implementation.

The history of modeling includes a tradition of individual or small team efforts to formulate a single comprehensive model that provides a robust solution for a particular problem. Seldom are elements of other models incorporated into such efforts beyond conducting the standard literature review. To introduce the system we propose in this article will require a culture shift originating in academic institutions that serve as the training centers for the modelers of the future. Developing DML, DMP, ACL, and ACP as a formal set of languages and protocols will make a step forward in changing the culture of model building. Upon experiencing the power of automatically sharing models between computers, we believe there will be acceptance of Semantic Modeling. As more model builders begin to use the languages and protocols, the power of the network will increase resulting in productivity gains in terms of reduced development time and the appropriate application of models in practice.

As a first step in demonstrating the advantages of Semantic Modeling, we are building several prototypes as a means of gaining acceptance from the software industry, practitioners with practical modeling problems to solve, and academic institutions where the most advanced models are devel-

---

[k] For further information see the following link published by the World Wide Web Consortium (W3C), http://www.w3.org/2004/OWL/.

oped. The first prototypes involve logistical modeling issues in the consumer goods, automotive, and healthcare industries.

The prototypes include a search engine interface that resembles an Internet browser to locate model elements residing on a network. The browser uses data inputs and outputs as the semantic for conducting the search. Once the appropriate models are located, another computer interface provides a workspace for visualization that shows how various model elements might fit together to form a practical solution. With this type of interface, the proper matching of a model to data and the interoperability of models becomes clear to the user.

Given the emergence of prototypes within the next year, there remains the question of what incentives will exist for model builders and practitioners to use Semantic Modeling. Our approach focuses on future model building and the establishment of a repository for models. However, the hundreds of logistical models currently in use present a problem in that these will need to be coded in the proper semantic language and protocols. Since many models are run using proprietary systems, the task of coding will be significant unless new methods of interface and translation are developed.

One idea to provide an incentive for driving Semantic Modeling forward involves a new Internet payment technology (Huang 2004). With this scenario, developers could form a representation of their models using semantics for posting to the Internet in machine understandable format. Those (either humans or machines) seeking to find models would do a search to locate the best model for their application. When the user downloads a specific model found by semantic search, the developer would receive a payment determined in advance or by market forces. In the case of simpler models, a smaller "micropayment" might be more appropriate given the volume of downloads. This would provide financial incentive for developers to select older models for coding that have been long forgotten by practitioners.

We envision a new industry forming where specialized firms constantly review old software or journal articles for signs of models having commercial value when semantically coded and distributed using the Internet. In the long term, existing large companies in the business of selling packaged software might yield to a new generation of firms that specialize in producing a repository of models capable of semantic search. With this scenario practitioners benefit in that model applications would more closely match the problem at hand rather than the current situation where many firms must radically redesign organizational processes to meet the demands of commercial packaged software. If nothing else, Semantic Modeling offers the possibility of assessing the true value of a model through the free exchange across a network.

A final hurdle for implementation of Semantic Modeling involves the adherence to standards. With every standards setting opportunity, there is always the chance that adopters will bend standards to meet their own objectives. This was the case in the development of electronic data interchange (EDI) standards as well as others. Good design of the standards along with active industry associations to monitor adherence are the means needed to maintain integrity.

In spite of these challenges, the emergence of search technologies made popular by Google and others as a commercial enterprise will eventually lead to improved capabilities to search and rapidly apply mathematical models in practice. Through such an approach, both model builders and practitioners will benefit from the establishment of an intelligent modeling network.

ACKNOWLEDGEMENT

The authors wish to thank the Editor and two anonymous referees for their valuable comments in the preparation of this article.

NOTES

Allen, Stuart J., Jack L. Martin, and Edmund W. Schuster (1997), "A Simple Method for the Multi-item, Single-level, Capacitated Scheduling Problem with Set-up Times and Costs," *Production and Inventory Management Journal,* Vol. 38, No. 4, pp. 39-47.

Allen, Stuart J. and Edmund W. Schuster (2004), "Controlling the Risk for an Agricultural Harvest," *Manufacturing & Service Operations Management,* Vol. 6, No. 3, pp. 225-236.

Allen, Stuart J. and Edmund W. Schuster (1994), "Practical Production Scheduling with Capacity Constraints and Dynamic Demand: Family Planning and Disaggregation," *Production and Inventory Management Journal,* Vol. 35, No. 4, pp. 15-21.

Brock, David L. (2003a), "The Data Project – Technologies, Infrastructure and Standards for Distributed Interoperable Modeling and Simulation," *MIT Data Project Workshop*, September: Cambridge, MA.

Brock, David L. (2003b), "Beyond the EPC – Making Sense of the Data – Proposals for Engineering the Next Generation Intelligent Data Network," *The MIT Auto-ID Center*: Cambridge, MA.

Brock, David L. (2000a), "Intelligent Infrastructure – A Method for Networking Physical Objects," *MIT Smart World Conference*, Cambridge, MA.

Brock, David L. (2000b), "The Electronic Product Code (EPC) – A Naming Scheme for Physical Objects," *The MIT Auto-ID Center*: Cambridge, MA.

Closs, David and Anand Nair (2001), "Supply Chain Management with Advanced Planning and Scheduling: Implications for Academia," *2001 Logistics Educators Conference*: Kansas City, pp. 317-343.

Coyle, John J., Edward J. Bardi, and C. John Langley (1992), *The Management of Business Logistics*, New York: West Publishing Company.

Dinning, Mark and Edmund W. Schuster (2003), "Fighting Friction," *APICS – The Performance Advantage*, Vol. 13, No. 2, pp. 27-31.

D'Itri, Michael P., Stuart J. Allen, and Edmund W. Schuster (1999), "Capacitated Scheduling of Multiple Products on a Single Processor with Sequence Dependencies," *Production and Inventory Management Journal,* Vol. 40, No. 4, pp. 27-32.

Engels, Daniel W., Robin Koh, Elaine Lai, and Edmund W. Schuster (2004), "Improving Visibility in the DOD Supply Chain," *Army Logistician*, Vol. 36, No. 3, pp. 20-23.

Feigenbaum, Edward A. and Julian Feldman (1963), *Computers and Thought*, McGraw-Hill: New York, pp. 6.

Fensel, Dieter, James Hendler, Henry Lieberman, and Wolfgang Wahlster (2003), *Spinning the Semantic Web*, Cambridge, MA: MIT Press.

Fine, Charles H. (1998), *Clockspeed*, Reading, MA: Perseus Books.

Geoffrion, Arthur M. (1987), "An Introduction to Structured Modeling," *Management Science*, Vol. 33, No. 5, pp. 547-588.

Huang, Gregory T. (2004), "The Web's New Currency," *Technology Review,* Vol. 106, No. 10, pp. 28-36.

Kar, Pinaki, Ming Li, and Edmund W. Schuster (2003), "A Case Study of Computer Service Parts Inventory Management," *Service Parts Resource Guide*, 1st Ed. Falls Church, VA: American Production and Inventory Control Society.

Koh, Robin, Edmund W. Schuster, Nhat-So Lam, and Mark Dinning (2003), "Prediction, Detection and Proof: An Integrated Auto-ID Solution to Retail Theft," *The MIT Auto-ID Center*: Cambridge, MA.

Krupp, James A.C. (1982), "Effective Safety Stock Planning," *Production and Inventory Management*, Vol. 23, No. 3, pp. 35-47.

London, Simon (2003), "Big Blue Reinvents Itself Again," *The Financial Times*, Oct. 9.

Melnyk, Steven A. (1998), “Finite Capacity Scheduling,” *APICS – The Performance Advantage*, Vol. 8, No. 8, pp. 60-69.

Sarma, Sanjay, David L. Brock, and Kevin Ashton (2000), “The Networked Physical World,” *MIT Auto-ID Center*: Cambridge, MA.

Schuster, Edmund W. and Stuart J. Allen (2005), *Raising the Bar (Code): The Value of Auto-ID Technology*, Cincinnati, Ohio: South-Western Educational Publishing, Forthcoming.

Schuster, Edmund W. and Stuart J. Allen (1998), “Visualization and Finite Capacity Planning in the Process Industries,” in *APICS International Conference Proceedings*, *Nashville, TN*, pp. 118-122.

Schuster, Edmund W., Stuart J. Allen, and Michael P. D’Itri (2000), “Capacitated Materials Requirements Planning and its Application in the Process Industries,” *Journal of Business Logistics,* Vol. 21, No. 1, pp. 169-189.

Schuster, Edmund W. and David L. Brock (2004), “Creating an Intelligent Infrastructure for ERP Systems; The Role of RFID Technology,” Falls Church, VA: American Production and Inventory Control Society, White Paper Published on Home Page.

Schuster, Edmund W., David L. Brock, Stuart J. Allen, and Pinaki Kar (2004), *The Role of Auto-ID Technology in Creating an Intelligent Infrastructure for ERP,* Palo Alto, CA: Stanford University Press, Contributed Chapter for Book on ERP Edited by F. Robert Jacobs and Elliot Bendoly, Forthcoming.

Schuster, Edmund W. and Byron J. Finch (1990), “A Deterministic Spreadsheet Simulation Model for Production Scheduling in a Lumpy Demand Environment,” *Production and Inventory Management Journal,* Vol. 31, No. 1, pp. 39-43.

Schuster, Edmund W. and Robin Koh (2004), “To Track and Trace,” *APICS – The Performance Advantage,* Vol. 14, No. 2, pp. 34-38.

Simchi-Levi, David, Philip Kaminsky, and Edith Simchi-Levi (2002), *Designing and Managing the Supply Chain*, McGraw-Hill/Irwin.

Weiser, Mark (1991), “The Computer for the Twenty-First Century,” *Scientific American*, Vol. 265, No. 3, pp. 94-101.

Willemain, Thomas R. (1995), "Model Formulation: What Experts Think about and When," *Operations Research*, Vol. 43, No. 6, pp. 916-932.

Willemain, Thomas R. (1994), "Insights on Modeling from a Dozen Experts," *Operations Research*, Vol. 42, No. 2, pp. 213-222.

ABOUT THE AUTHORS

David L. Brock is Principal Research Scientist at the Massachusetts Institute of Technology, and co-founder and a Director at the Auto-ID Center (now EPCGlobal, Inc. and Auto-ID Laboratories), an international research consortium formed as a partnership among more than 100 global companies and five leading research universities. He is also Assistant Research Professor of Surgery at Tufts University Medical School and Founder and Chief Technology Officer of endoVia Medical, Inc., a manufacturer of computer controlled medical devices. Dr. Brock holds Bachelor degrees in Theoretical Mathematics and Mechanical Engineering, as well as Masters and Ph.D. degrees from MIT.

Edmund W. Schuster has held the appointment of Director, Affiliates Program in Logistics at the MIT Center for Transportation and Logistics and currently works in research and administration at the MIT Data Center. His interests are the application of models to logistical and planning problems experienced in industry. He has a Bachelor of Science in Food Technology from The Ohio State University and a Masters in Public Administration from Gannon University with an emphasis in management science.

Stuart J. Allen is Professor Emeritus, Penn State – Erie, the Behrend College. He works on design of decision aids for application in manufacturing environments. His educational background includes a Bachelor of Science in mechanical engineering from the University of Wisconsin, a Masters of Mechanical Engineering from Seattle University, and a Ph.D. in engineering mechanics from the University of Minnesota. Stuart began his research career in the field of non-Newtonian fluid mechanics and has published over 50 journal articles in engineering and management science. He has also owned and operated three businesses in Wisconsin and New York State.

Pinaki Kar is currently an independent consultant working in the pharmaceutical industry on analysis and modeling to support strategic planning, business development, and marketing. He is interested in the application of operations research and statistical techniques for planning and decision support across a wide range of business issues. His experience spans multiple industries that include pharmaceutical, chemical, high-tech, and insurance. Pinaki's educational background includes a bachelor degree in mechanical engineering from the Indian Institute of Technology, Kanpur and a Masters degree in Logistics from MIT.