The Open System for Master Production Scheduling:

Information Technology for Semantic Connections between Data and Mathematical Models

Edmund W. Schuster†
Field Intelligence Lab and Data Center Program
Laboratory for Manufacturing and Productivity
Massachusetts Institute of Technology

edmund_w@mit.edu

Hyoung-Gon Lee
Laboratory for Manufacturing and Productivity – Data Center Program
Massachusetts Institute of Technology

and

u-Computing Innovation Center (uCIC)
Seoul National University, South Korea

Stuart J. Allen
Penn State – Erie, The Behrend College
Erie, PA

Pinaki Kar
New York, NY

Ping Wang
Beijing, China

† Corresponding author.

The Open System for Master Production Scheduling:

Information Technology for Semantic Connections between Data and Mathematical Models

ABSTRACT

This paper examines the general case of an open system architecture to deliver a specific master scheduling model to end-users. The architecture is flexible and can accommodate the delivery of any mathematical model written in structured computer code such as Java or C++ resident on a remote server, to end-users located anywhere using a Microsoft Excel spreadsheet as the interface. The approach employs the M Dictionary for machine understandable semantics, eliminating confusion concerning word definition and improving the precision of Internet search.

INTRODUCTION

Customer service plays a central role in achieving marketing objectives for firms in the consumer goods industry. The most important element of customer service is product availability (Coyle, Bardi, and Langley 1992, p. 81). Commonly measured as the fill rate for incoming orders, product availability depends on the amount of end-item inventory in situations where a make-to-stock policy exists. Manufacturing firms in the consumer goods industry adopt the make-to-stock policy because the manufacturing lead-time for end-items is often longer than the cycle time for taking and shipping an order.

The main tool to control product availability is the master production schedule (MPS). By using the beginning inventory and the sales forecast for a particular end item, a planner can calculate the amount of production needed per period to meet anticipated customer demand. This calculation becomes more complex in a multi-product environment where forecast errors and capacity constraints can add a great deal of uncertainty to the planning process. As firms continue to integrate the MPS into supply chain planning, it is becoming increasingly clear that the MPS plays a major role in managing the trade-off between costs and product availability. A previous paper presented at the Council for Supply Chain Management Professionals Educators' Conference has noted that even though master production scheduling is considered a part of logistics, research in this area is somewhat lacking in the evolving context of supply chain management (Closs and Nair 2001).

This paper examines a new direction for master production scheduling that combines a proven heuristic to find a near optimal solution along with an open system approach to deliver sophisticated mathematical modeling capabilities to end-users via the Internet. Specifically, the M Language (mlanguage.mit.edu) in conjunction with other web standards enables end-users direct access to the Modified Dixon-Silver Heuristic (MODS) through an information technology called Software as a Service (SaaS). This approach allows anyone access to the MODS algorithm located on a remote server using only a Microsoft Excel spreadsheet.[1] There is no implementation of MODS on local computing systems and access is immediate. Essentially, the algorithm serves as a calculator and does not store any data from the spreadsheet on the server.

The open system approach fulfills a goal to standardize and speed the process of modeling in practice by essentially creating a supply chain for mathematical models that is searchable across the Internet with precision. The M Language dictionary provides machine-understandable semantics to describe data fields that are inputs, outputs, and attributes of MODS. In this way, there can be no misunderstanding regarding the type of data needed to run the model or the meaning of various calculations required for input data. The M Language is also the standard for XML data transfers between the spreadsheet and the algorithm or any other target. This makes the data interoperable. Overall, the SaaS approach, combined with the M Language, quickly puts state-of-the-art modeling in the hands of many users with no local computer implementation other than downloading an Excel spreadsheet.

The MODS model is a useful way to calculate the MPS for situations relating to make-to-stock manufacturing, commonly experienced in the food, consumer goods, and repetitive manufacturing industries. Tested for more than ten years, MODS consistently finds feasible solutions involving the

[1] To download the spreadsheet and connect to MODS, click on the following link (it is best to save the file to disk, and then re-open to execute MODS):

http://web.mit.edu/lmp/news/news_03_07_08.html

or

www.ed-w.info/osmps6.xls

Please note that users of the spreadsheet must enable the macros to allow access to the MODS algorithm located on a remote server.

trade-off between choice of products to produce and capacity, inventory carrying cost, setup time, and setup cost. Solve times for large planning problems involving 32 products and 52 weeks are less than ten seconds. MODS is robust enough to become the standard for make-to-stock manufacturing worldwide.

The general name for the computer architecture described in this paper is the Open System for Master Production Scheduling (OSMPS). With the open system approach, there are no costs for users to run a model other than the time needed to organize data or to modify the Excel spreadsheet. This computer architecture is not limited to the production planning and scheduling domain. The open system approach also applies to other types of mathematical models that might occur in industries as diverse as agriculture or finance.

For example in agriculture, a complex model for harvest risk (Allen and Schuster 2004) is a candidate for the SaaS approach using the M Language. This model will give growers in the fruit and vegetable industry instant access to a way of calculating the rate of picking during harvest when the start time (full maturity for the crop), end-time (frost that stops photosynthesis), and crop size are random variables. Optimizing the trade-off between the cost of harvesting capacity and the risk of losing crops because of over maturity or poor weather conditions is fundamental to creating economic efficiencies in the fruit and vegetable industry, especially in the Northern United States fruit belt. Achieving this optimization under conditions of uncertainty is also an advance in the application of mathematics to calculate supply chain risk.

Further, each year the United States Department of Agriculture funds mathematical modeling research for a wide a range of topics involving grain crops like corn, soybeans, and wheat. These models provide guidance to farmers regarding important decisions such as when to apply agricultural chemicals based on sampling the frequency of pests in a field. Currently, there exists no mechanism to organize and implement these models for individual farms using a simple interface like an Excel spreadsheet. The open system computer architecture described below allows for rapid updating of programming code, eliminating many of the problems associated with version control. This is helpful to practical implementation of agricultural models.

To summarize, any class of mathematical model can use the open system architecture and the M Language for delivery to end-users through an Excel spreadsheet. The balance of this paper provides details about the OSMPS and important trends in the software industry.

SOFTWARE AS A SERVICE

For nearly all supply chains, Enterprise Resource Planning (ERP) plays an important role in coordinating the various activities and business processes inside individual firms. An enormously complex system, ERP depends on data to accomplish planning and scheduling tasks of importance to the management of modern manufacturing. During the past ten years, ERP systems have changed significantly in response to greater sophistication in consumer markets. Globalization, outsourcing, proliferation of stock keeping units, and shorter life cycles (McFarlane and Sheffi 2003) have brought customers expanded product variety at low cost, along with new types of supply chain complexity for manufacturing firms.

As markets and consumer needs change, cost of purchasing and implementing large- scale ERP systems has increased dramatically (Davenport, Harris, and Cantrell 2005, p. 71). It often takes several years to complete an ERP installation and financial results in terms of increased efficiencies are sometimes elusive. In addition, ERP systems that are software packages delivered by large firms such as Oracle or SAP can limit opportunities for replacing individual modules quickly when technology, business conditions, or markets change. This is especially the case for MPS systems. The offerings of

MPS systems by ERP vendors strive to meet the needs of a large user base by limiting functionality to the lowest common denominator. Business process re-engineering becomes the means for firms to adapt to the limited functionality of MPS software packages contained in large-scale ERP systems. Because there are no universal software solutions for calculating the MPS across all industries, a more flexible approach for MPS delivery is valuable to many firms.

Software as a Service (SaaS) is an alternative worth considering for companies that face the high costs of enterprise-wide implementations and extensive re-engineering efforts to enhance existing business processes. In particular, small and medium sized companies often do not have the financial resources to purchase "on-premise" systems that require significant investment in packaged software and dedicated hardware. However, it is also true that large companies with extensive installations are beginning to consider SaaS as a cost cutting measure. In recognition of this trend, SAP has begun shifting their basic strategy to "on-demand" services as a means of offering individual software modules with a target market of small and medium sized firms. This is a major development in the delivery of various types of software to customers.

Some analysts estimate that virtually all new software will eventually use the SaaS architecture. (Ray 2008). The recent successes of Apple's iTune's and Google Apps have paved the way for more software and content to be offered in this way. For example, Salesforce.com has introduced AppExchange, an innovative approach to establishing a software-sharing platform. The vendor no longer distributes software using CD-ROM as a media; rather it sells access to applications located on internet. With this approach, Salesforce.com is becoming the eBay of hosted software by supporting about 1,000 developers specializing in Customer Relationship Management (CRM) applications. Another company doing something similar is NetSuite, financed by Oracle founder Larry Ellison. The firm accomplished a successful initial public stock offering in 2007.

With the current interest in SaaS, it is a reasonable extension that master production scheduling will become an Internet-based service rather than a dedicated software application hosted on local computers. Eventually, it might become obsolete to use packaged software for management of manufacturing processes and other applications in business. While packaged software will be around for many years into the future, the end of the 2002 economic expansion might mark the peak in sales. For the United States market, where the growth of manufacturing has slowed considerably, SaaS represents an innovative way of reducing the cost of operation for existing and prospective users of ERP systems.

With the advent of the Internet, there is no fundamental reason why MPS calculations for a firm need to done using dedicated software and computers located on-site. Further, more advanced mathematical models that consider capacity or bias adjusted safety stock (Schuster, Unahabhokha, and Allen 2005), also do not need to be hosted on local computing systems. Given the substantial literature on various approaches for calculating the MPS, the Internet has potential to become a vehicle to match and deliver specific models to real-world problems experienced in industry.

Establishing MPS-as-a-service, however, requires the resolution of research issues in computer science and information technology relating to the connection of data and mathematical models, which both have complex hierarchies and semantics. Creating an open system architecture for delivery of mathematical models via the Internet with the end-user in mind is the challenge of expanding SaaS as a force in the software industry and a new opportunity to improve decision-making in supply chain management.

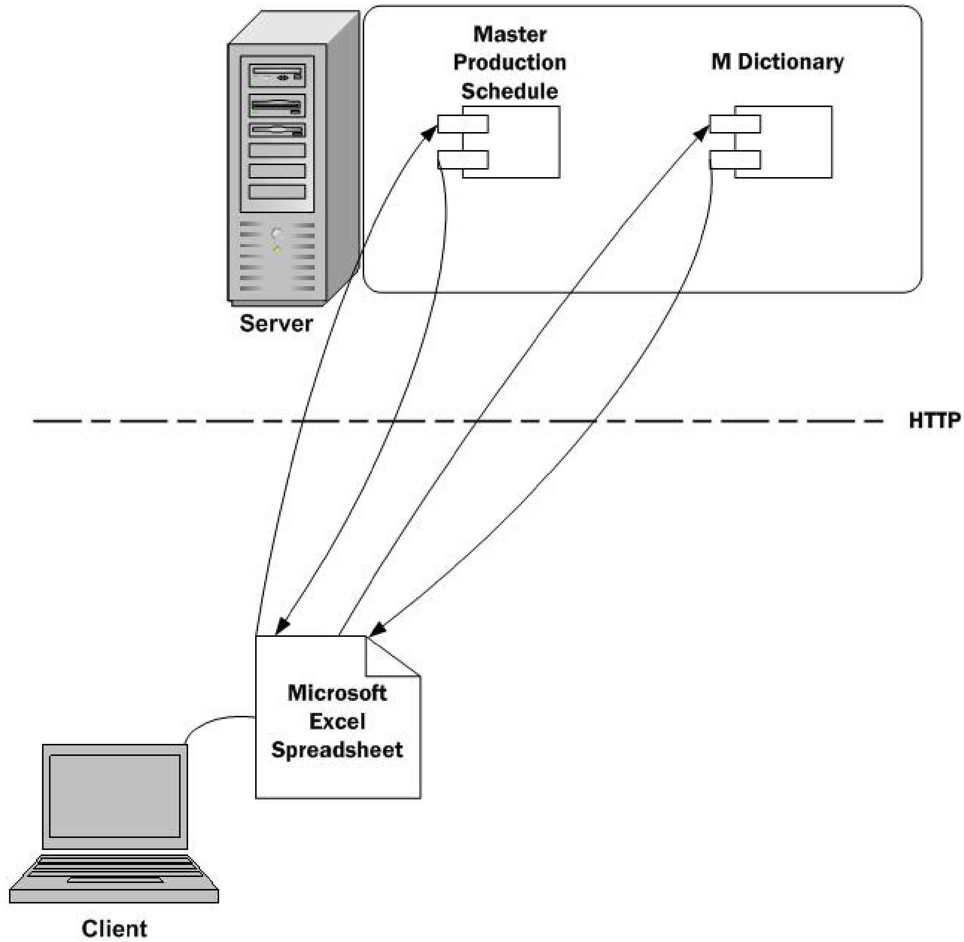
AN OPEN SYSTEM ARCHITECTURE FOR SaaS

The problem of creating an open system for MPS calculations, and mathematical models in general, has two aspects. First, there must be a simple interface for the user. Second, a means must exist to connect the interface to computer code located on a remote server. Separating the interface from the computer code offers several advantages in rapid delivery of complex mathematical models to end-users and the control of versioning. The central idea is to host a MPS model, written in a structured computer language like Java or C++, on a single server with an interface that can be downloaded onto any computer using the Internet. The interface then connects to the central server when running the model. Such a system allows users located anywhere in the world to use a particular MPS approach with little implementation and at no cost.

An Excel spreadsheet serves as the end-user interface for OSMPS. Spreadsheets are easy to understand and many firms already do master production scheduling in Excel using custom approaches developed internally. Enhancements in Excel 2003 and 2007 allow for direct interaction with a remote server that contains computer code such as Java.

Creating an open system for modeling using Excel spreadsheets also requires a robust way to treat semantics. The OSMPS uses the M Dictionary to provide consistent semantics for words and noun phrases contained in the spreadsheet interface that are elements of MODS. In this way, Internet search becomes precise and there is no ambiguity regarding the definition of terms used to describe the data fields, and Internet connections between and Excel spreadsheet and the code for the MPS. Figure 1 shows the high-level diagram of how the Excel spreadsheet interacts with the remote server and the M Language dictionary.

Figure 1 – The Overall Architecture of OSMPS



This SaaS architecture depends on a standard called Web Services[2] to carry out communications across the Internet. As defined by the World Wide Web Consortium, a Web Service is "a software system designed to support interoperable machine to machine interaction over a network (Booth 2004)." Typically, these interactions take place via messages that flow between a server and a client. The messages use the Uniform Resource Indicator (URI) for location, the extensible markup language (XML) for content, and the hypertext transfer protocol (HTTP) for communication. As a means of

[2] World Wide Web consortium, W3C

describing the operations supported by a particular server, the Web Services Description Language (WSDL) identifies specific software and connections available in a way that allows discovery by other systems using simple object access protocol (SOAP). The Web Services approach as a way to expose software located on a specific server to the Internet is now a widely accepted practice. It is becoming the standard for integration of business applications including various types of software (Vinoski 2003).

Web Services provide the means of connecting computing machines together across the Internet and is an important component of the OSMPS. The final element needed to create a robust open system is a common semantic to describe data, namely the inputs, outputs, and attributes of MODS, or any other mathematical model. As shown in Figure 1, the M Dictionary provides precise semantics for the open system by interacting directly with the Excel spreadsheet. What follows is a description of the important functional aspects of the M Dictionary in relation to the open system.

UNIFORM SEMANTICS

At the most basic level of communication, words are the glue that connects nearly everything together (Schuster, Allen, and Brock 2007). The power of words can give descriptive meaning to the most complex physical objects existing in business or nature, and to the most diffuse ideas that exist only in the mind. Data is described with words, information is word based, and computer code uses words as a way of communicating the various operators available to programmers. Words establish not only the limits of human imagination and intellect, but also the possibilities for the computing systems of the future.

Every word has at least one definition and, when used in conjunction with other words, it is possible to create a countless number of sentences. For example, a simple cartoon shown to twenty-five different people will generate twenty-five individually unique perspectives if each writes a single sentence about what they see (Lederer 1991). Dictionaries organize and define words used to make sentences, and a weak ontology[3] based on groupings such as synonyms and antonyms provides the relationships that the human mind can fathom into linguistic communication.

In spite of the almost limitless capacity of human language, describing data or mathematical models in a consistent machine understandable way remains a challenging objective for both computer scientists and practitioners within business. Although English is a powerful tool for communicating meaning through words, noun phrases, and sentences of varying patterns and complexity, this ability is under-utilized by modern computing systems because meaning is a combination of syntax, semantics, and context that is beyond the cognitive abilities of computers.

The fundamental problem with employing words as a descriptor is that a single word can have several different definitions and multiple words can have the same definition. This paradox means that natural language often does not have the internal consistency required for straightforward application as an identifier or a unit of meaning within computer systems.

Complicating matters, the intricacy of meaning increases dramatically when dealing with the noun phrases and sentences needed to describe data and mathematical models. Given this property of English, it is impossible with current technology to conduct a semantically precise, computer-based search of information contained in web pages, quantitative data tagged with words, news feeds comprised of text files, complex mathematical models, or any other situation where words describe physical or abstract objects. Achieving the goal of word descriptions that are machine understandable requires a deeper appreciation of the role of semantics in computing systems, and especially the Internet.

[3] In computer science, an ontology represents the relationships between things.

The M Dictionary

The words used in the M Language are slightly different from English words. In M, every word has only one definition. This is an extremely important characteristic because computers that communicate using M do not need to understand the context or usage of a word to know its meaning.

English words are ambiguous. For example, the word “cell” might mean “cellular phone,” “biological cell,” “jail cell,” or “fuel cell.” Without some idea of the context, it is impossible to know the meaning of the word “cell.”

To overcome this issue, the M language includes a number to denote individual words as in the following example:

```
cell.1
```

To account for multiple definitions, the M Language allows numeric extensions, one for each definition. Thus, `cell.1` is a word in M and `cell.2` is a different word. With this method, every word has only one meaning.

Figure 2 shows a screen shot of the M Dictionary (mlanguage.mit.edu) for the word “forecast.” In this example, there are five different definitions. Depending on usage, some compound words are also part of the M Dictionary. An example is “operations research,” represented as `operations_research.1`.

Figure 2 – An example from the M Dictionary

In addition to the definition, the dictionary entry also contains three other pieces of important information. These include (1) word relations, (2) data format, and (3) language translations. This information is available through Web Service interfaces.

Word relations are simply the connections between words. These relationships include synonyms, antonyms, types, and parts. Synonyms and antonyms are the same as in English.

Types refer to word generalizations. For example, `automobile.1` is a *type of* `motor_vehicle.1`. Parts are words that are components of another word. This is often the case when thinking about physical objects, although this could also be the case with abstractions. For example, a `wing.4` is a *part of* `airplane.1`.

Both types and parts establish a hierarchy within the dictionary through making connections between entries. These word connections are valuable in a number of different ways, including improved Internet search.

Data format provides guidance concerning the forms and patterns of data values that are associated with a particular word. In many situations, computer-to-computer communication might contain a word such as `first_name.1` that has an associated data value such as "John." Other common situations include words like `telephone_number.1`, `account_balance.1`, or `postal_code.1`. In all of these cases, a word in the dictionary has a particular format or pattern for associated data.

Finally, the language translation portion is simply the representation of the word in M as a word (or phrase) in another human language besides English. In most situations, computer-based language translation is very difficult because a lack of context exists for the specific communication. Since in M each word has only one definition, the word `cell.1` (biological), for example, cannot be confused with `cell.2` (telephone). Words with a single definition allow users to specify exact meaning independent of context. This eliminates ambiguity in translation.

The M dictionary uses the wiki approach with several important modifications including improved security through user registration, maintenance of the integrity of word relations, a monitoring function to reduce the chances of near identical definitions, and administrative controls to ensure accuracy. The dictionary also has various statistical features that measure usage.

The "wiki" process has emerged as an innovative application of Internet technology to knowledge management and consensus building. A "wiki" is a type of website that allows users to add and edit content and is especially suited to collaborative authoring. It is remarkably accurate (Associated Press 2005) and several companies have begun to use the process internally (Wessel 2005).

Since 2001, Wikipedia has become the largest encyclopedia ever created with over 2.3 million articles in English. Currently, there are 210,000 words in the M Dictionary and over 700,000 ontological relationships.

Precise Semantics for Internet Search

Perhaps the most important aspect of the M Dictionary is the ability to create a precise semantic for Internet search. Since words in M have only one meaning, a key-word search in M yields matches based on the definition, not on the character string. This allows for search based on meaning rather than keyword. Further, the M Dictionary provides additional search capabilities as compared with current approaches.

Every definition in the M dictionary includes provision for word relations. From the previous example in this paper, `automobile.1` is a *type-of* `motor_vehicle.1` and `wing.4` is a *part-of* an `airplane.1`.

Combined with definition-based search, these relations provide a powerful tool. For example, search can span types of a word. A query for types-of `flower.2` would return `rose.4`, `violet.3`, or `marigold.1`. A search for parts-of `automobile.1` would return `fender.1`, `muffler.3`, or `engine.1`. Web Services provide a means of query for the various ontologies listed in the M Dictionary.

Searching on a definition and using the word relations from the dictionary greatly increases the precision of Internet search. In addition, words and phrases from the M dictionary can become part of meta-tags used in HTML and Excel spreadsheets, significantly increasing the accuracy and performance of web-based search tools such as Google, MSN Search, and Yahoo. In one prototype, an interface that facilitates the selection of M words for entry into search engines has accomplished superior search results.

DETAILS ABOUT THE OSMPS

The final section of the paper examines the SaaS delivery of OSMPS using web services, the M Dictionary, and MODS. As a first step, the Excel spreadsheet partially shown in Figure 3 must be downloaded from the following link:

http://web.mit.edu/lmp/news/news_03_07_08.html

or

www.ed-w.info/osmps6.xls

It is best to save the spreadsheet to disk, and then open. Immediately after opening, the spreadsheet will connect to the remote server where the Java code for MODS heuristic is located. Java Web Services enable the connection.

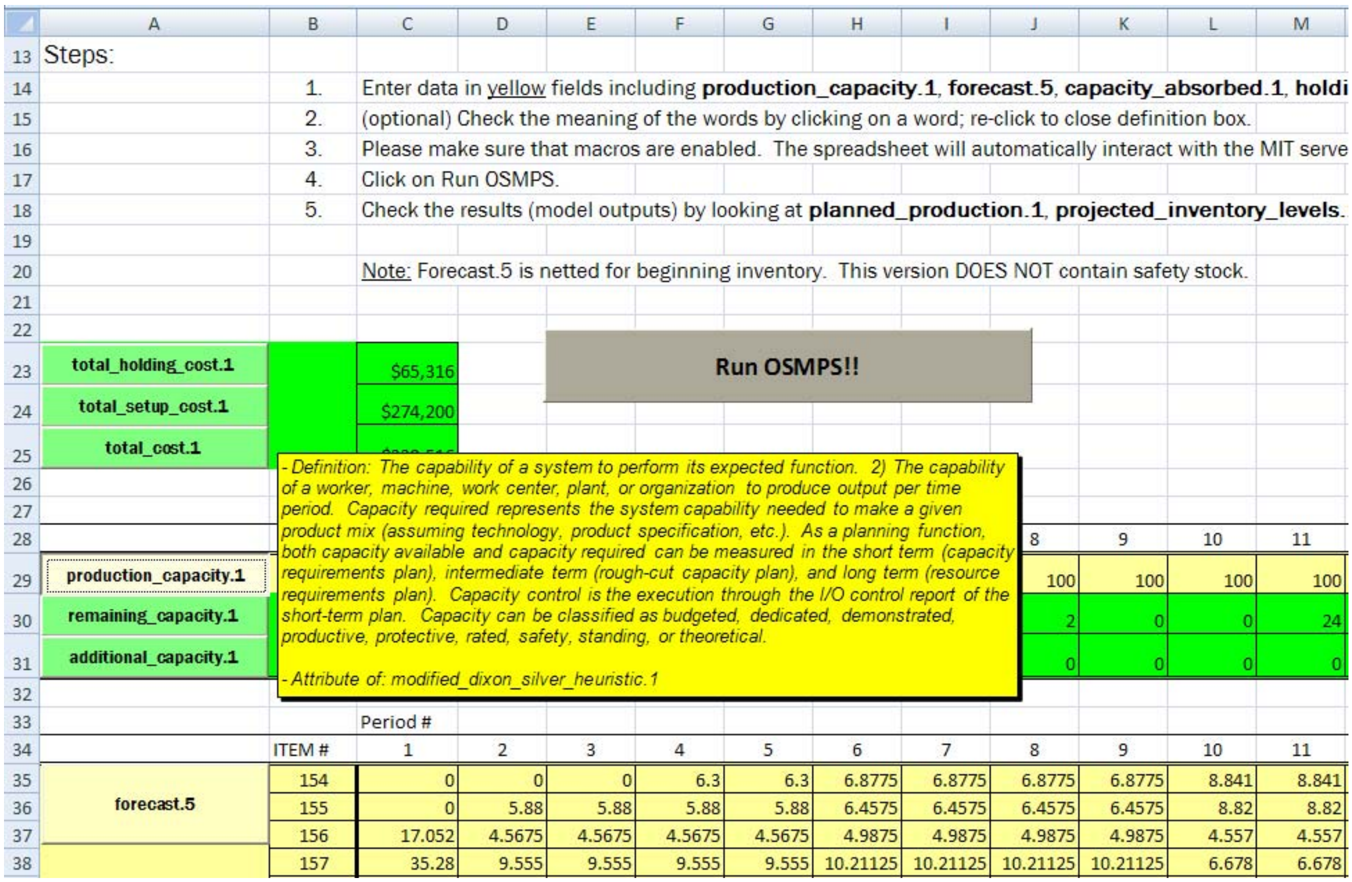
First, a dialog box will appear that will ask the user if they wish to update the M words contained in the spreadsheet. If yes, then the spreadsheet connects to the M Dictionary and all word definitions are updated. This process takes about 30 seconds.

Next, a dialog box appears asking the user if it is ok to clear the model outputs in preparation for a new run of MODS. Clicking yes clears these cell values so that the user can see new values appear as MODS runs. The spreadsheet is now open for entry of new data. The final step is to run the model by clicking RUN OSMPS.

Figure 3 shows that words from the M Dictionary that label various data fields within the spreadsheet. Clicking on an M word opens a yellow box that provides a detailed definition of the word.

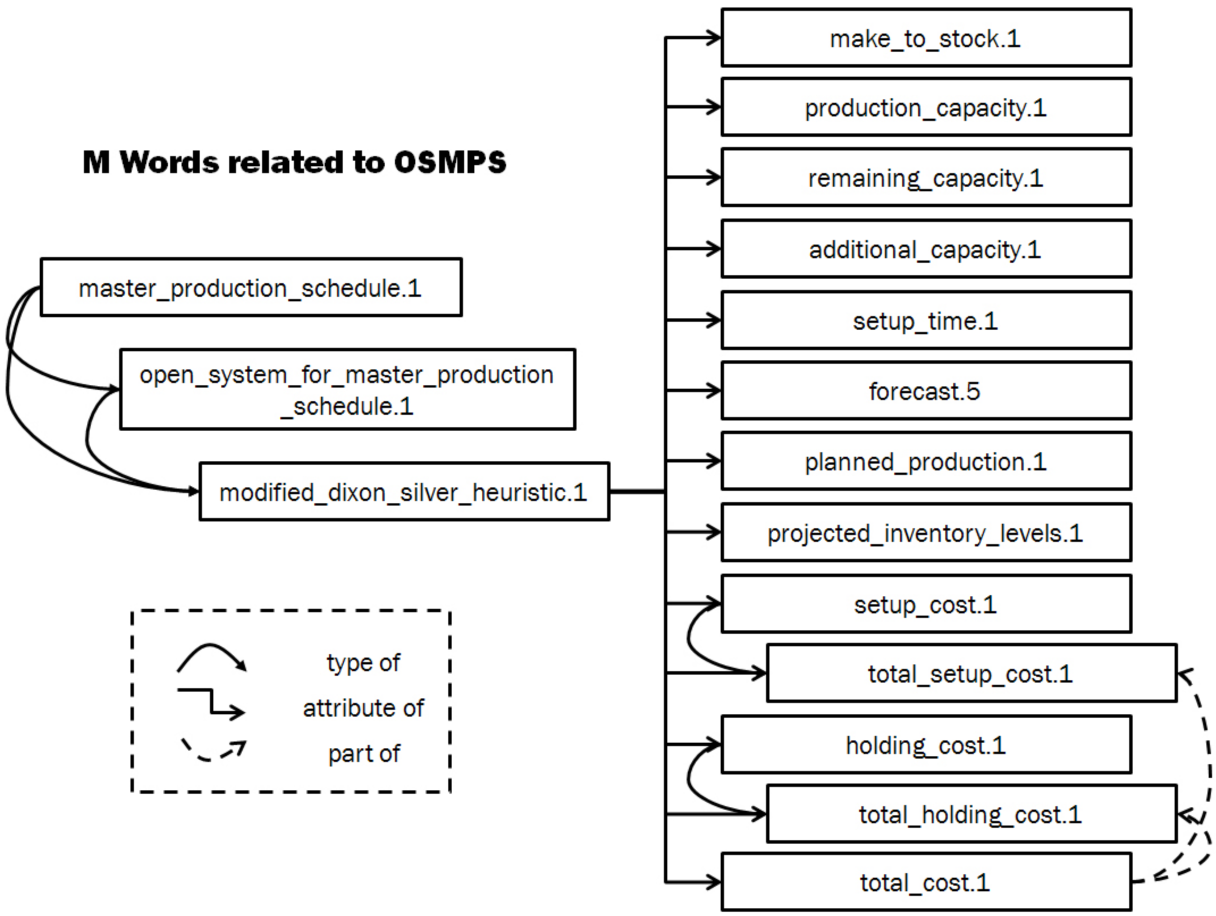
In Figure 3, the definition of `production_capacity.1` appears. For the OSMPS, nearly all of the M words take the form of phrases with an underscore separating each word and the numeric extension appearing at the end of the phrase. The ability to use phrases that have exact semantic meaning is an advantage of using the M Language.

Figure 3 – The OSMPS Excel Spreadsheet Interface



The relationship, or ontology, for the various M words used to describe the data inputs, outputs, and attributes of MODS appear in Figure 4. For instance, the `total_holding_cost.1` is a "type of" `holding_cost.1` (inventory carrying cost per unit, per time), which is also a "part of" total_cost.1 one of the outputs from a MODS run. Though this ontology is very specific, flexibility exists to add words describing other, non OSMPS approaches linked to master_production_schedule.1.

Figure 4 – The OSMPS related ontology



The M words used to describe data inputs, outputs, and attributes of MODS offer a great advantage in conducting Internet searches. By placing M words into the "properties" of an Excel workbook, they are exposed to search engines like Google, Yahoo, or MSN if the .xls file is saved on a server. This is essentially a meta-tag for a workbook. Typing M words like total_cost.1 or planned_production.1 into a search engine will lead to the exact Internet location of the file, enabling users to find a model such as MODS that has an M word to describe a data input, model output, or attribute very quickly.

In summary, the OSMPS is an effective, low cost way to deliver mathematical models to users. Implementation costs are low and the M Dictionary gives users a precise definition for data inputs, outputs, and attributes. Since most practitioners are familiar with spreadsheets, the interface is simple and allows for flexibility in customization. The open system approach has the potential to offer users a variety of models for a specific business process such as master production scheduling.

The final part of paper provides background information about MODS in the context of the OSMPS.

THE MODIFIED DIXON SILVER HEURISTIC (MODS)

Designed for a specific set of operational conditions, MODS is a heuristic that finds solutions using "rules of thumb." Heuristics are intended to converge quickly in finding solutions that are close to, or even may match, the optimal solution. For MODS, the specific conditions are:

- A make-to-stock manufacturing environment with no stock-outs or backorders permitted.
- Multi-item, single level, dedicated production lines with finite capacity
- Setup times and cost are nonzero and sequence independent
- Sequencing of multiple items to be produced within a specific time period is not considered
- Safety stocks (buffers) are determined "outside" of the scheduling system. As an additional note concerning safety stock, the MODS application described in this paper does not include safety stock, although this can be included if needed.

These conditions are typical for many different industries, including food, chemical, pharmaceutical, plastics, paper, and biotechnology. In addition, MODS is an appropriate application for repetitive manufacturing such as high volume metal forming where single machines do “runs” of different products. Collectively, these industries comprise a large segment of worldwide manufacturing.

The MODS heuristic is the outcome of many years of research tracing to early work on single level, lot sizing under infinite and finite capacity situations (Silver and Meal 1973; Dixon and Silver 1981). Others made improvements by adding setup time and by using an Excel spreadsheet combined with visual basic programs to calculate the MPS (Allen Martin, and Schuster 1997). Using design of experiments, the performance of MODS has been tested over a wide range of real conditions.

The MODS spreadsheet is configured for 52 time periods and 32 items. Though not transparent to the end-user, special functions exist to expand the time periods and number of items for scheduling. The data input requirements for MODS are straightforward and include:

<u>Data Inputs</u>

- `Forecast.5`: by item: the demand for each period netted for beginning inventory, by item (cell C35 to BB36 – anticipated units sold per week).
- `Production_capacity.1`: units of capacity available (cell C29 to BB29 – total hours available for the manufacturing line or machine)
- `Capacity_absorbed.1`: units of capacity required for production, by item (BH35 to BH66 – hours to produce 1,000 units)
- `Holding_cost.1`: the cost of holding inventory, by item (BJ35 to BJ66 – Dollars per 1000 units per month)
- `Setup_cost.1`: the cost of a setup, by item (BL35 to BL66 – Dollars per setup)
- `Setup_time.1`: the time to setup, by item (BN35 to BN66 – hours per setup)

<u>MODS Outputs</u>

- `Remaining_capacity.1`: the amount of surplus capacity per week (C30 – BB30, hours)

- `Additional_capacity.1`: the amount of capacity needed over standard capacity (C31 – BB31, hours)

  NOTE: the MODS algorithm makes every effort to fit production into available capacity, however, sometimes an over capacity situation exists.

- `Planned_production.1`: the production schedule by week (C35 – BB66, units per week by item)

- `Projected_Inventory_Levels.1`: the amount of inventory remaining at the end of each week (C105 – BB136, units per week by item)

- `Total_holding_cost.1`: the sum of the holding cost for the 52 week period, Dollars

- `Total_setup_cost.1`: the sum of the setup cost for the 52 week period, Dollars

- `Total_cost.1`: total holding cost plus total setup cost, Dollars

CONCLUSTION

This paper examines the general case of an open system architecture to deliver a specific master scheduling model (MODS) to end-users. The architecture is flexible and can accommodate the delivery of any mathematical model written in structured computer code such as Java or C++ resident on a remote server, to end-users located anywhere using an Excel spreadsheet as the interface. The approach employs the M Dictionary for machine understandable semantics, eliminating confusion concerning word definition and improving the precision of Internet search. The MODS heuristic is robust and is relevant to a large segment of make-to-stock manufacturing firms. Combined with the open system architecture, MODS can become a standard for solving specific scheduling problems in addition to an educational tool for demonstrating commercial grade, supply chain applications to university students and industrial practitioners.

Further enhancements to MODS will include safety stock models and specific capabilities to study master production schedule stability. Plans are in place to expand the open system architecture with other MPS models and models from other disciplines such as agriculture.

REFERENCES

Allen, Stuart, J., Jack L. Martin, and Edmund W. Schuster (1997), "A Simple Method for the Multi-item, Single-level, Capacitated Scheduling Problem with Setup Times and Costs," *Production and Inventory Management Journal* 38:4, pp. 39-47.

Allen, Stuart J. and Edmund W. Schuster (2004), "Controlling the Risk for an Agricultural Harvest," *Manufacturing & Service Operations Management* 6:3, pp. 225-236.

Booth, David, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard (2004), Web Services Architecture, *W3C Working Group Note*, Available: http://www.w3.org/TR/ws-arch/

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.

Davenport, Thomas H., Jeanne G. Harris, and Susan Cantrell (2005), "Getting More Results from Enterprise Systems," in *Strategic ERP: Extension and Use*, Palo Alto, CA: Stanford University Press, Edited by F. Robert Jacobs and Elliot Bendoly.

Dixon, Paul S. and Edward A. Silver (1981), "A Heuristic Solution for the Multi-Item, Single- Level, Limited Capacity Lot Sizing Problem," *Journal of Operations Management* 2:1: p. 23-39.

Lederer, Richard (1991), *The Miracle of Language*, New York: Simon & Schuster, p. 16.

McFarlane, Duncan and Yossi Sheffi (2003), "The Impact of Automatic Identification on Supply Chain Operations," *International Journal of Logistics Management* 14:1, p. 2.

Ray, Tiernan (2008), "Salesforce.com's New Rivals," *Barron's*, Jan. 7.

Schuster, Edmund W., Chatchai Unahabhokha, and Stuart J. Allen (2005), "Master Production Schedule Stability Under Conditions of Finite Capacity," unpublished manuscript.

Schuster, Edmund W., Stuart J. Allen, and David L. Brock (2007), *Global RFID: The Value of the EPCglobal Network for Supply Chain Management*, New York and Berlin: Springer: pp. 229 – 247.

Silver, Edward A. and Harland A. Meal (1973), "A Heuristic for Selecting Lot-Size Quantities for the Case of a Deterministic Time-Varying Demand Rate and discrete Opportunities for Replenishment," *Production and Inventory Management Journal* 14:2: p. 64-74.

Vinoski, Steve (2003), "Integration with Web Services," *IEEE Internet Computing* 7:6, pp. 75-77.

Unauthored (2005), "Science Journal Says Wikipedia is Fairly Accurate," *Associated Press*, December 15.

Wessel, Rhea (2005), "Office Technology Opening Up," *The Wall Street Journal*, October 24.