Simple Tools for Managing Engineering, Purchasing and Assembly Data

Robert Kondner
rkondner@indexdesigns.com
Index Designs
Towson MD. 21286

Abstract
As the assembly of electronic products becomes more automated the collection, storage and retrieval of electronic data demands additional software tools to manage data. The wide spread use of web and email in the delivery of electronic data requires close integration of these data sources to an organization's data system. A number of new and sophisticated functions are now available within current office software and operating systems but many of these features have gone unnoticed by many users. The purpose of this paper is to identify some of these functions and explain how they can be used to help manage engineering data.

Introduction
If you ask most email users what version of email software they use many would respond with dates of 2003 or 2007. If you ask why they don't use a more recent version they will probably answer "I can send and receive emails perfectly fine right now. Why spend the money?" Professional level desktop software tools do cost several hundred dollars and the introduction of new versions along with new required operating systems requires significant user learning time. At the same time many software users maintain thousands of emails in their In Box and searching for items is a serious problem. Even if a personal email systems is maintained by a very organized user any coworker or substitute would find locating or understanding past transactions difficult.

Throughout this paper I will make many references to Microsoft products. As the dominate software vendor a discussion based around these products will target a large audience. Many other vendors exist and they have excellent products. However for the sake of discussion I will use the Microsoft Office suite of tools and operating systems as my major reference point.

Large commercial systems exist for dealing with data management but many are priced in the $10Ks to $100K+ range. While this paper is about using simple low cost tools we want to look at the architecture of many of these commercial systems. These system are characterized by central data storage in a client server environment with multiple users editing data and generating reports.

Client Server Database with Custom Editors
We have all used classic client server environments. A web browser is a perfect example of where requests are submitted to a central machine and screens or forms are rendered back on the users screen. In such a system all data is translated into a format acceptable for storage at the central site. These systems are typically built around a central database system (Oracle or SQL Server for example) while software provides users with screens and forms used to input or review data. If screens and forms are presented to the user using HTML and a web browser we have a web based system. If code on the user machine (client) obtains data using SQL statements over a local area network we have a conventional client server environment.

The most recent trend in "Big System" design is using large web development platforms such as Share Point Server. A major advantage of this approach is most users are fairly experienced at navigating within a web environment. Existing templates simplify system construction and this large base of infrastructure provides the system integrator with a high level to build upon.

In Figure 1 below we see a central data server supporting a number of users. This architecture simplifies design, test and maintenance as most major functional logic is centralized at a single machine. Software licensing is typically on a seat by seat basis and is easily policed through the central server logic. Remote users can connect using conventional Virtual Private Network (VPN) technologies. A big advantage of this approach is company "Business Rules" can be integrated into system software which reduces the likely hood of a user not following through with required practices.

The downside of this approach is both initial expense and maintenance. Integration of custom business rules can be expensive and users often require training courses. For example a system backup of an advanced server is much more complicated than backup of a typical desktop machine. The major downside is it represents a closed system. Vendors

typically do not have access directly to the data and must be sent output reports of files creating the version control issues such a system was designed to solve in the first place. Vendors build to the data they are given, not what is on their customer server. Buy creating a number of business rules a company can expect a more consistent operation but at the expense of some flexibility.

Another issue is each and every user will still have a desktop machine with standard desktop software. The problems of maintaining desktop software still exists. The centralized server does not eliminate the need for internal email or messaging and additional software tools are still required to manage these transactions.

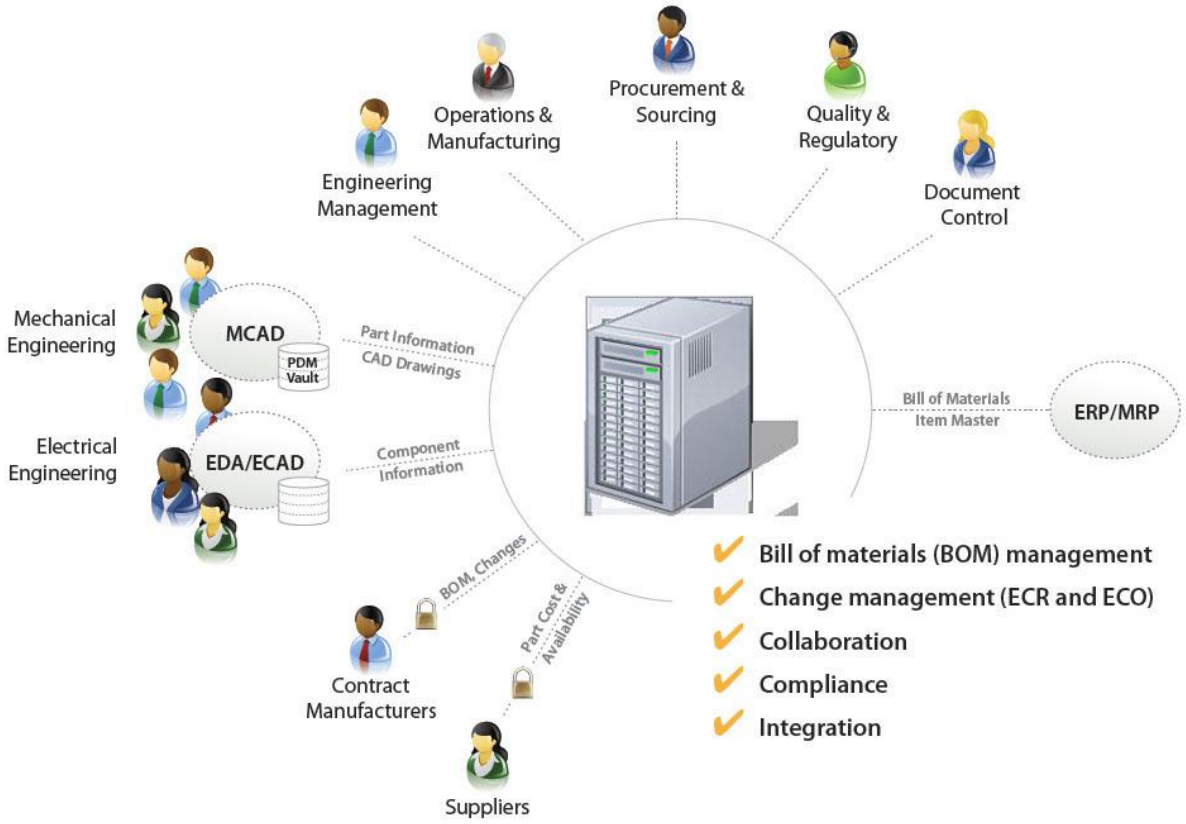


Figure 1 Classic Client Server

If we eliminate the central server and allow users to "Simply Send Emails" then we have a different picture for an Ad Hoc Email System.

Ad Hoc Email System

At first glance the "Ad Hoc Email" system in figure 2 looks like a giant mess. It can easily become a giant mess. The problem is this picture represents what actually happens in most companies. Worse yet this same mess is not significantly reduced even when a centralized data system is provided for engineering and company data. The Ad Hoc drawing is not totally accurate as most "Email Only" approaches also include file shares. Many users think of file servers as a general purpose server but there is a difference. A file server is basically a shared disk. It can reside almost anywhere on a LAN. Compared to a SQL or Oracle servers there is almost zero support logic provided. The control of what data is placed where is totally dependent on where users save it. In a small company where everyone follows some simple rules this can be successful. For a larger organization a central data server that keeps data organized and consistent is required.

The important word here is "Consistent". While the goal of database design is to avoid duplication of data it is often not accomplished. In an Ad Hoc environment with different user storing files in various places as well as keeping their own copies it might be very difficult to keep data consistent. Complete consistency requires complete rules covering all cases which are always enforced, this is difficult to achieve in any environment.

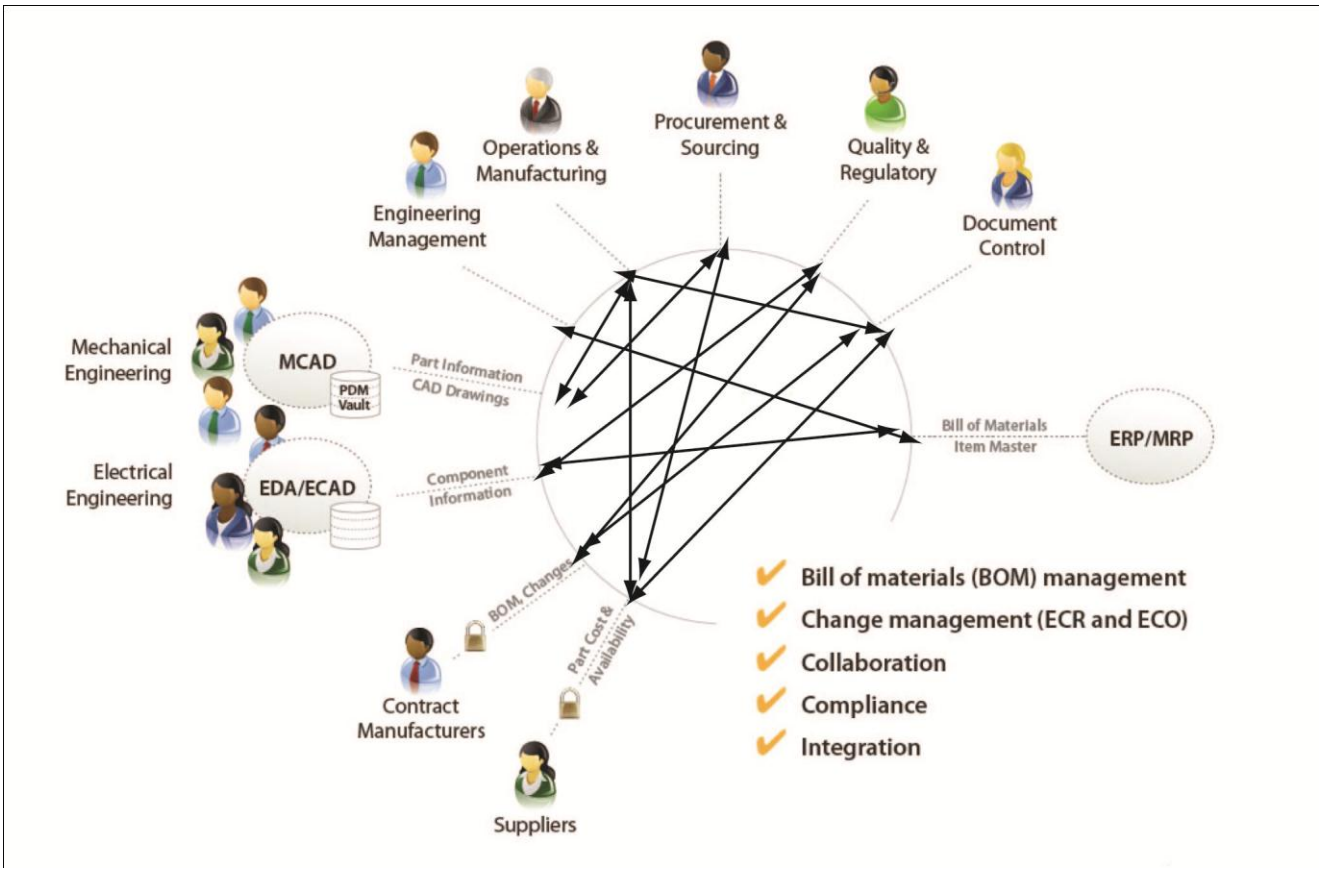


Figure 2 - Ad Hoc Email

A major problem with simple file shares is finding things. In the course of operation a very large array of files will be generated by a large number of users. When an exception occurs and a specific piece of data needs to be found a simple file share system makes finding that data difficult. This is no different than trying to find data on the web. If we need to find something on the web we use a search engine, todays web would be useless without search engines. The same logic applies to file shares.

Not only do users need to locate data in file shares but they also need to locate data on their own machines be it files or emails. Again we see a requirement for the ability to search and find data. Software vendors have not been sleeping. In the next sections we will see some of the new and useful features built into desktop software suite and operating systems.

A useful feature found in Office product is the "Conversation View" of email messages. By using consistent subject fields in messages the history of a set of emails can be easily monitored.

New Windows Search Tools

The latest OS versions from Microsoft (Vista and Win7) have introduced significant improvements in the area of search. A rather elaborate system of scanners and filters is used to index not only file information (names and dates) but also file content. Older XP indexing operation was turned off by default as it was often intrusive of system resources. The "Search Companion" in XP was a slow process and it gave "Search" a bad name.

It is important to understand how searches for information can be very different between various software systems. The slow "Search Companion" in XP where the entire C: drive is read is a perfect worst case example. The use of a "Dog" for the "Search Companion" is all we need to say. Under "Search Companion" the entire drive is often read as a search is performed. Compare this to how Google searches the contents of millions of web sites in a fraction of a second. The difference relates to Google's use of a precompiled index. Goggle uses a large number of "Crawlers" running day and night to read web site contents and then generate indexes. In this case a search is against the index which contains pointers to specific web sites. Win 7 includes additional features such as libraries for housing specific ranges of index collections as well as the ability to index and search remote file shares. The generation of index files within Win 7 can be configured for specific files types, file contents and file locations with the results being used to construct specific search libraries.

Figure 3 below shows a sample search using both file extensions and dates. The .bmb file type is an internal file type used to control BOM and build configuration. The text of "date:>lastweek" restricts the search to finding only those .bmb files with dates in the last week. Notice that Microsoft included a large array of file tag data types and includes an extensive list of date symbolic values which includes "lastweek". There is a large array of possible fields by which to control searching. Users can add custom tags to many file types and these can be used to refine searches. The syntax for a search can be saved and used at a later date. This isolates most users from needing to know the Advanced Query Syntax used to construct searches.

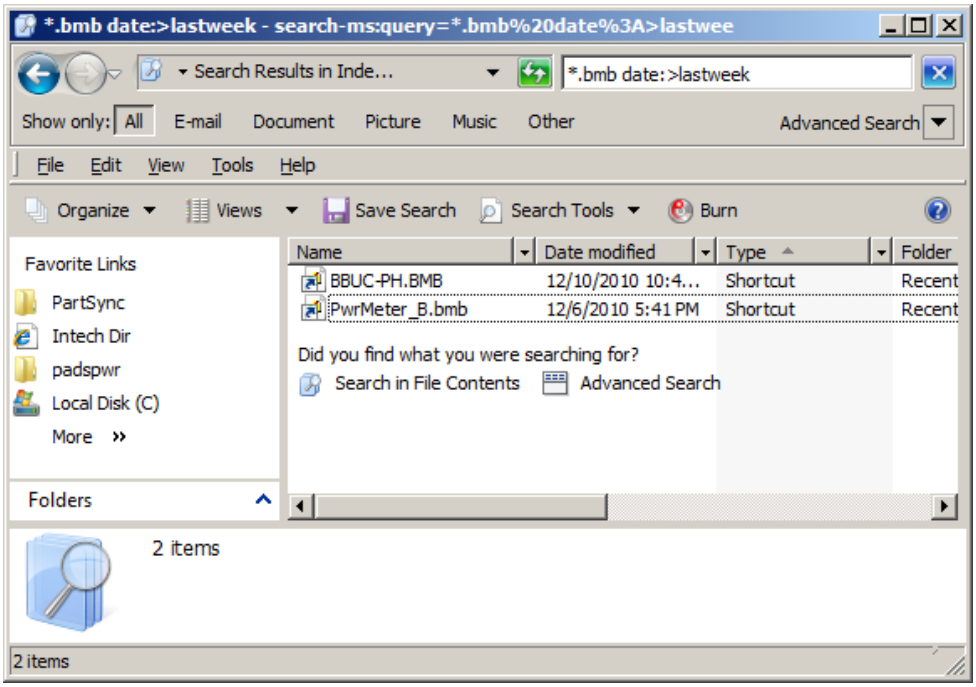


Figure 3 - Windows Search

When a search is save it appears as a new "Folder" in under the search folder within Windows Explorer.

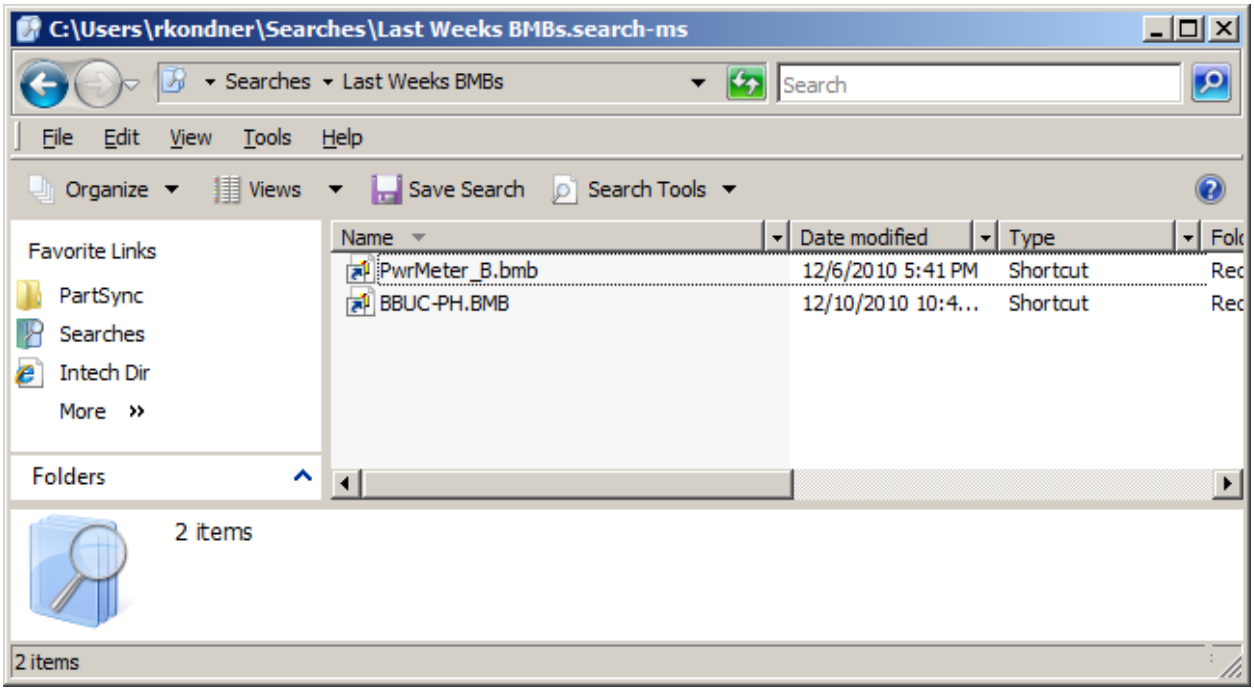


Figure 4 - Saved Search

This saved search "Folder" is automatically updated as files change or days pass

File Tags
Starting in Vista certain file types allow a user to append "Tags" to a file. These are pieces of text are separate from file content that are typically used to help sort and find files. Many, but not all, file types associated with common desktop tools support user tags. Microsoft Word file do support tags but .txt files do not. When supported a tag can be placed or edited during a Save As, within the Windows Explorer Detail Pane or using the File properties dialog box.

In figure 5 below there are a number of file properties that can be used to refine a search. Author and Title are generated by Word when the document is created. (If you are using a recent version of Word!) Also shown in figure 5 is user data that has been entered into the "Tags" property. A search of tag:gamma or tag:Alpha will find this file. Multiple tags can be used along with other properties, file names, file extensions or dates. Further, all this searching is through index files built and maintained by the Windows indexing engine so searches are VERY fast.

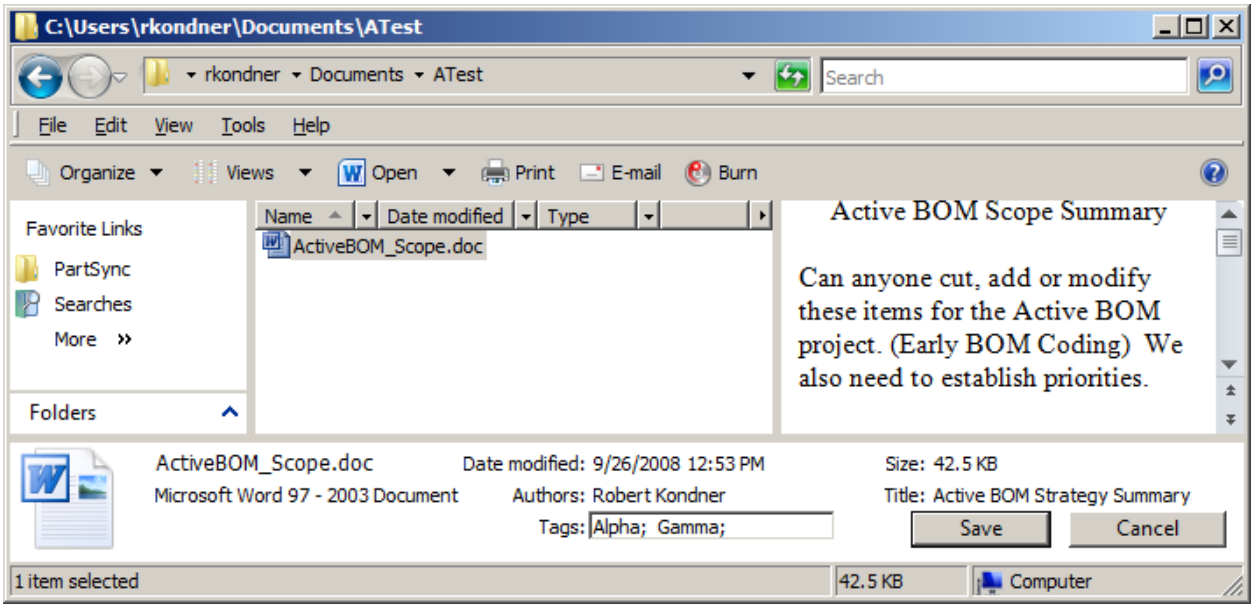


Figure 5 - Custom Tags for User Info

Operating System Resources
The Windows Explorer screen shown above in figure 5 is a very good example of the newer features in Vista and Win 7. What we see is an integrated view of searching, file view, file preview (right side of screen) and file properties. We have come a long way from the "dir *.*" of DOS days.

Microsoft OneNote
A underused applications for the storage of various types of job related information is Microsoft OneNote. Originally designed as a software "3 Ring Binder" OneNote is a graphically oriented container for almost any type of data. Pictures, emails, spread sheets, documents or custom file types all fit onto pages which fit into sections which form notebooks. Individual pages can have subpages for a structured arrangement of data and search capabilities are included. OneNote is well integrated with the Microsoft Office environment.

OneNote definitely qualifies as a significant change in application software. It is as different from previous office apps as Excel is from Word. OneNote abandons the client server approach for managing data and instead uses synchronization techniques on simple file server. Multiple users of the same notebooks are supported using local caches of the notebook contents. Synchronization happens automatically in the background.

A good example of how OneNote can be applied is in the management of assembly tasks. Incoming files, emails and intermediate work and cleansed BOMs can all be placed in notebooks pages. In figure 6 below a sample job has been built using text, graphics, embedded files and spreadsheet data. The page can be shared with other users with access control using passwords. If the file is changed the author and time stamp are recorded. Backup copies are kept in case a file is changed in error.

In general there are many useful features that OneNote offers to the electronic design workplace.

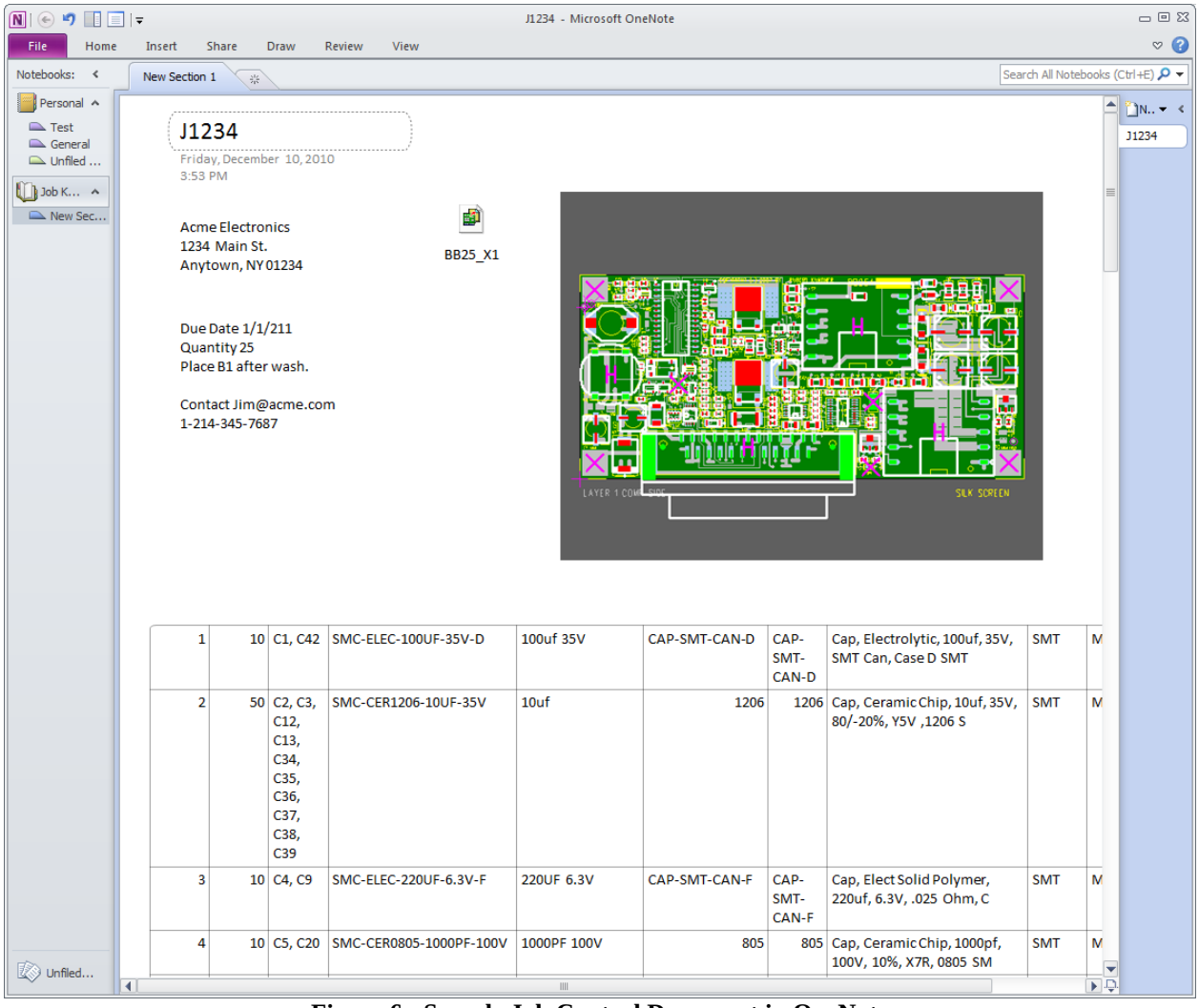


Figure 6 - Sample Job Control Document in OneNote

Synchronizing BOM items with Purchasing and Assembly Data

Any organization with either a purchasing system or automated assembly equipment will already have a significant investment in existing data structures. When a new job or PCB design is delivered for assembly there is a need to convert, or at least match, the materials to be assembled with existing purchasing and assembly equipment libraries. BOM items that cannot be matched to existing or previous used components will require setup within internal databases. Creating new parts represents a significant level of effort.

The process or crossing customer BOM line items to existing internal databases is complex. Customer data will come in many formats and from many CAD tools. While spreadsheets are the most common format they are most likely to contain manual edits and are thus suspect to errors. Even data from CAD system can contain confusing errors. Very often a single CAD file might be a product of multiple designers using different libraries, this can cause confusion during coding. Few internal MRP/ERP/Inventory system include search engine capabilities making the lookup of line items a manual process.

The lookup and crossing of BOM line items into existing components in not a process where normal desktop tools can assist. Spread sheet tools such as Excel often have interfaces to external but some programming and configuration is required. The programming environment in office tools is often weak and professional programming tools are generally much more productive. When new components are identified the existing internal purchasing and assembly machine libraries will require additional components. When coding a BOM the time required to add the new components is a major portion of the effort. There are very few tools to address these issues.

Figure 7 below show the overall flow of data when a customer BOM is coded against internal purchasing and assembly machine libraries. Included in this drawing is a database called "Reuse Data". Having a means to collect data from previously coded customer jobs can significantly reduce the time required to locate customer BOM components within internal libraries. While this is a powerful concept it requires the coding process to involve yet another database source. At the same time customer BOM components are matched to purchasing information there is also need to match BOM components with assembly machine information. Unless these two data sources can be tied together a total of two lookups will be required for each BOM line item.

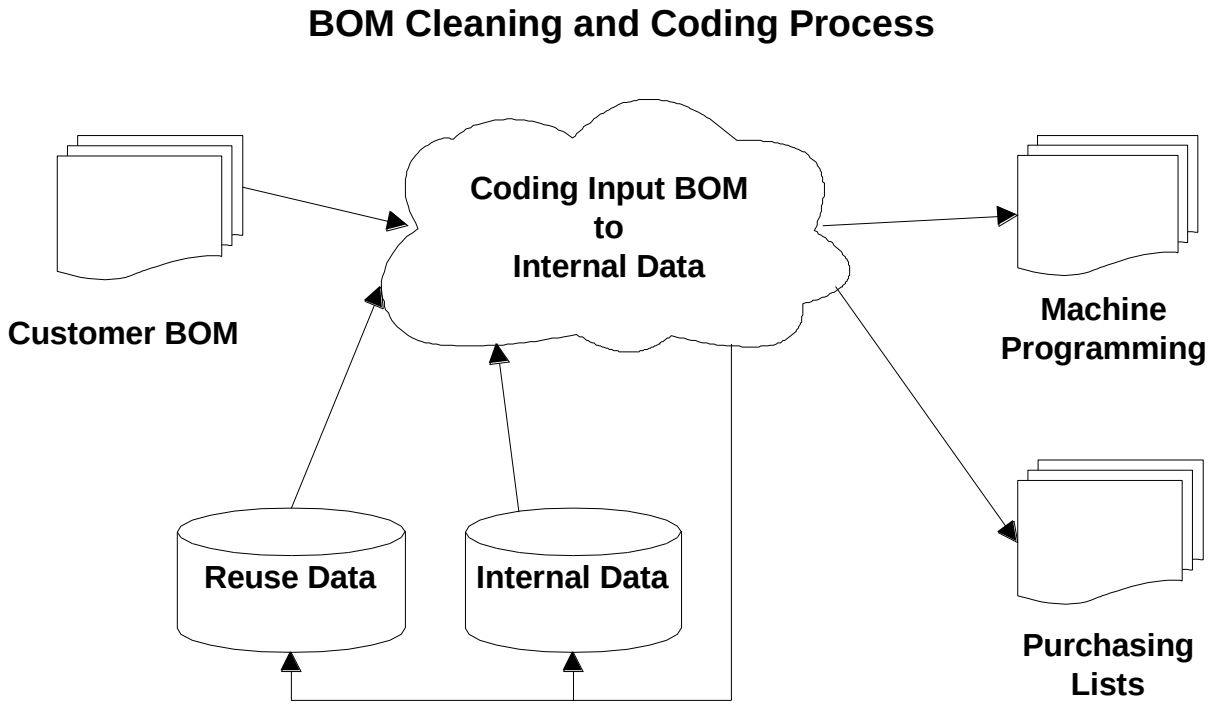


Figure 7 BOM Coding Process

Tools for BOM Coding

Many of the tools in use for coding BOM line items are tools associated with specific Pick and Place (PNP) machines. Given the large array of component package sizes and feeding devices (reels, trays or tubes) a simple in house part number is only a minor concern. In fact most PnP software systems utilize an in house part number as a primary key field in local database tables. While the PnP software might use a in house number internally it is often the purchasing department that generates and maintains in house part number and does so using MRP or ERP software. The result is two sets of data, one in purchasing and another in manufacturing that must be kept synchronized. This synchronization is usually accomplished through manual editing of one database or the other.

When looking at the complete scope of component data systems there are additional requirements for PCB footprints and assembly translation constants. Assignment of PCB footprints is a major source of design errors and the CAD tools that are used to place components use coordinate systems which are different from PnP machines. Rotation angles and spin directions are the most common differences. Additional parameters must be included in a component database if CAD data is to be automatically converted to drive PnP operation. In many organizations this conversion is not done in software, instead conversion is typically done by hand on the PnP machine when it is not being used for assembly.

Figure 8 Shows the BOM coding environment used at Index Designs. On the left is the input BOM where components have been grouped into line items as line items are the bases for both purchasing and PnP setup. The center column shows purchasing data for this line item. This center column is actually the result of a search through the in house master database base The source for the search was the input in the BOM line item. We have included PCB footprint and assembly machine package information within our master database which allows purchasing and assembly coding to take place at one time. Also included within master tables is datasheet information which is often required to support purchasing, engineering and assembly activities.

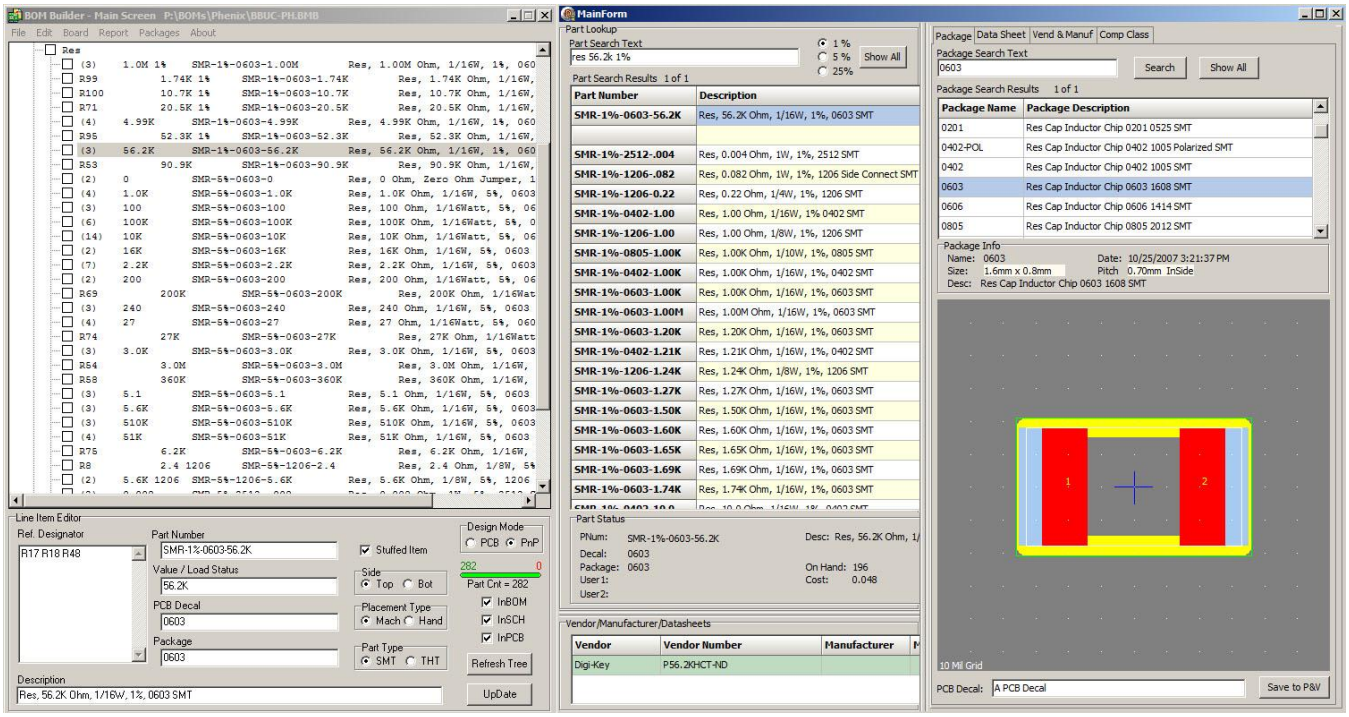


Figure 8 - BOM Coding Tools

While the process of BOM coding is complex the output is fairly simple, plain text files in most cases. Even when driving spread sheet programs text files in plain or .csv formats can be used. OneNote provides a special printer driver by which any application that prints can supply data to be saved on OneNote pages.

Summary

With most customer communications now taking the form of email a desktop office suite is the ideal point to start capturing customer transactions. The problems of finding data can be addresses using advanced search features of operating systems and by using file properties and tags. Search and sort features are also contained in current office tools making large collections of emails more easily read.

OneNote provides a very flexible storage contains for various file types. It has a simple organization format, that of a 3 ring binder, which is easy to learn and use. OneNote contains its own set of search functions and it is integrated with the Microsoft Office suite of desktop software. OneNote is tied to the "To Do" list in Outlook and helps generate tasks and develop schedules.

The task of "BOM Line Item Lookup and Coding" is not a simple task and it can involve multiple databases. Standard applications like Excel do contain a number of database features and the more recent version include "Pivot Tables" which are very much like the data grids found in database applications. Additional products from 3rd party vendors or consultants will be required to further automate BOM coding. It is very important that any coding work done for a job be made available for the coding of future jobs. Experience show that many "New Jobs" are based on previous work and even new jobs often use a number of components found on previous jobs by the same customer.

A suite of desktop tools would not be considered appropriate for running a larger assembly organization but for smaller organizations it might provide the required functions at a very low cost and learning curve.

Simple Tools for Assembly Data

Robert Kondner rkondner@indexdesigns.com

Major Topics:

- Introduction
- Client Server environments
- Ad Hoc environments and Email
- Advances in OS features
- Microsoft OneNote
- BOM coding. Matching for Purchase and Assembly
- Tools for BOM Coding
- Summary

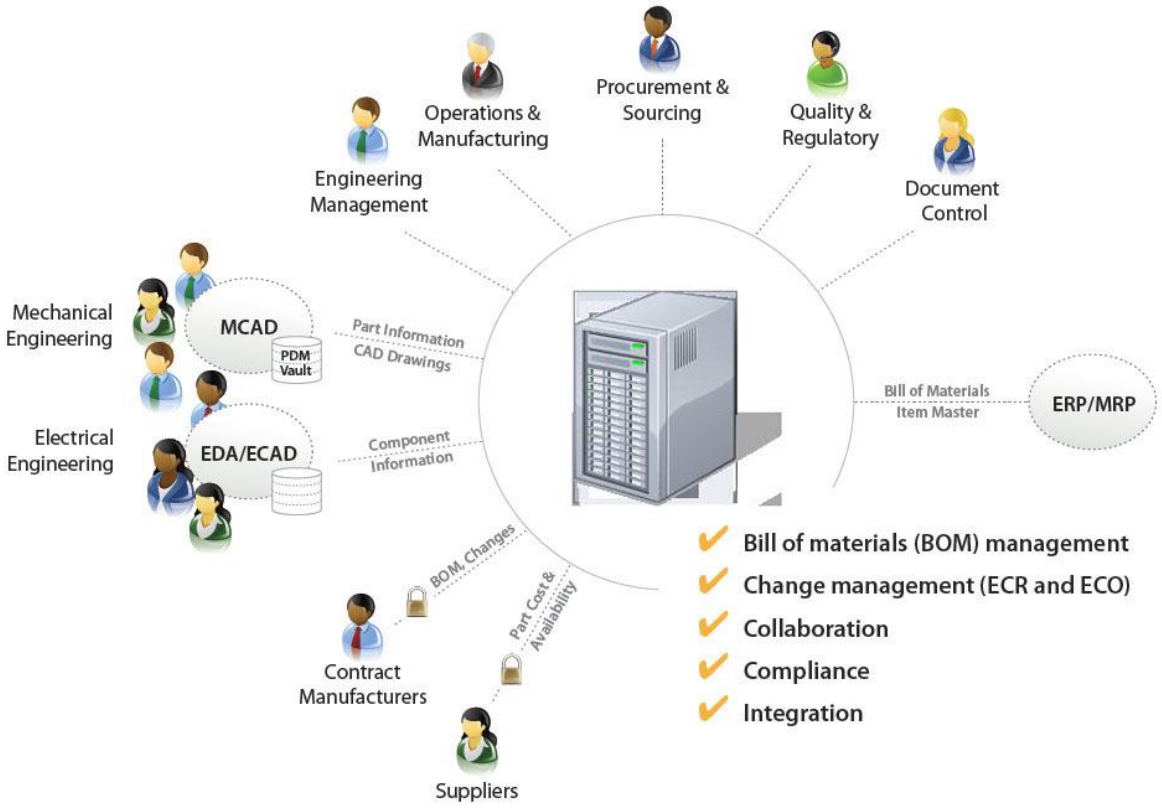
Introduction

- Dual Systems in use:
  - Email based environments for most work.
  - Client servers apps for configuration control.
- Client Server issues:
  - Complex: Used only for final release.
  - Prototype data was “Any Way Possible”.
  - Care and feeding we an effort.
- Can Desktop Tools Help?

Client Server Environments

- Advantages
  - Rules to verify content.
  - Wide range of system reports.
  - Covers wide range of business functions.
- Disadvantages
  - Complex with steep learning curve.
  - Expensive to purchase and maintain.
  - Not appropriated for smaller companies

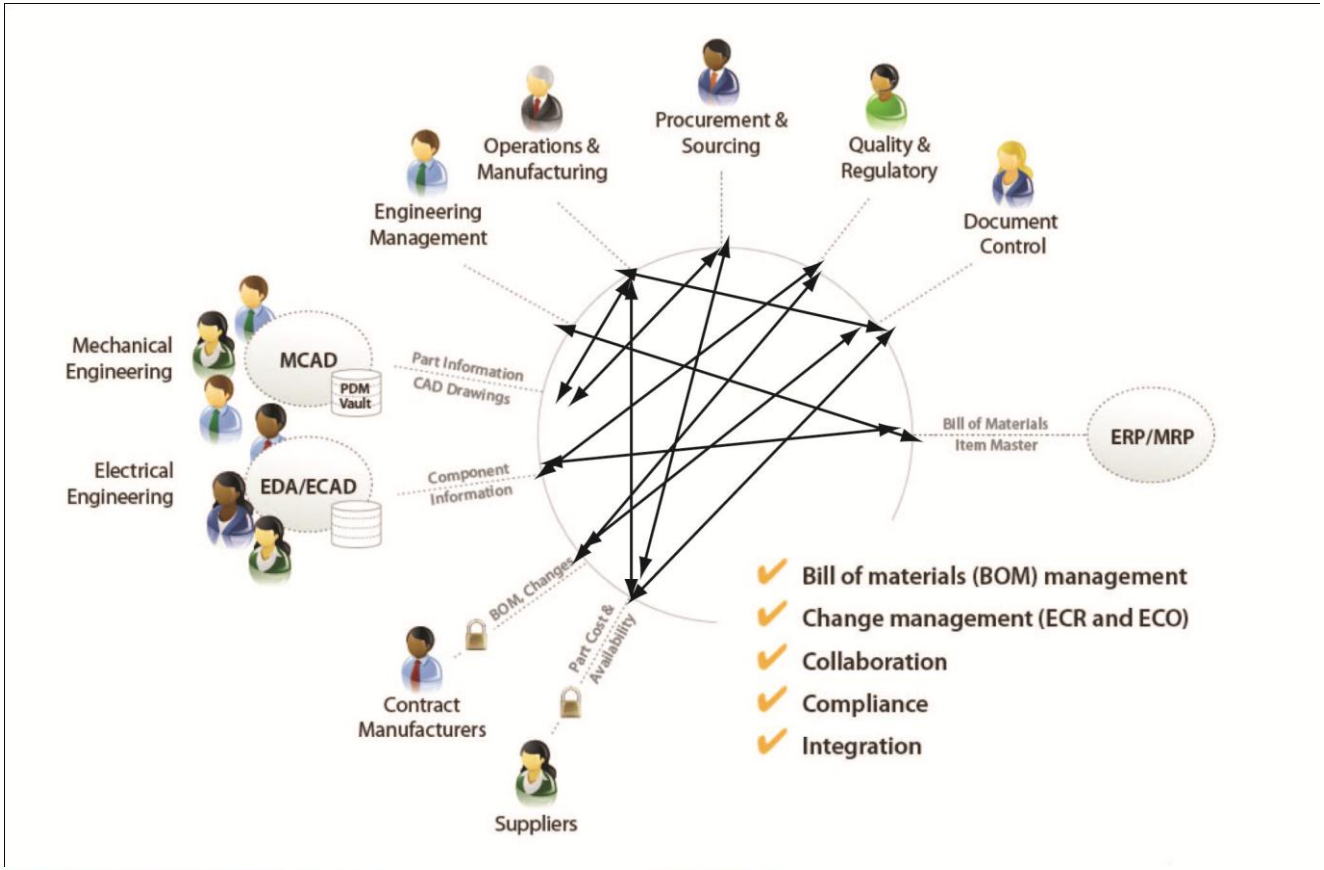
Example of Client Server



Ad Hoc Email

- Advantages
  - Already in place
  - Easy to learn
  - Extend to include customers
- Disadvantages
  - No built in rules
  - No timeout limits

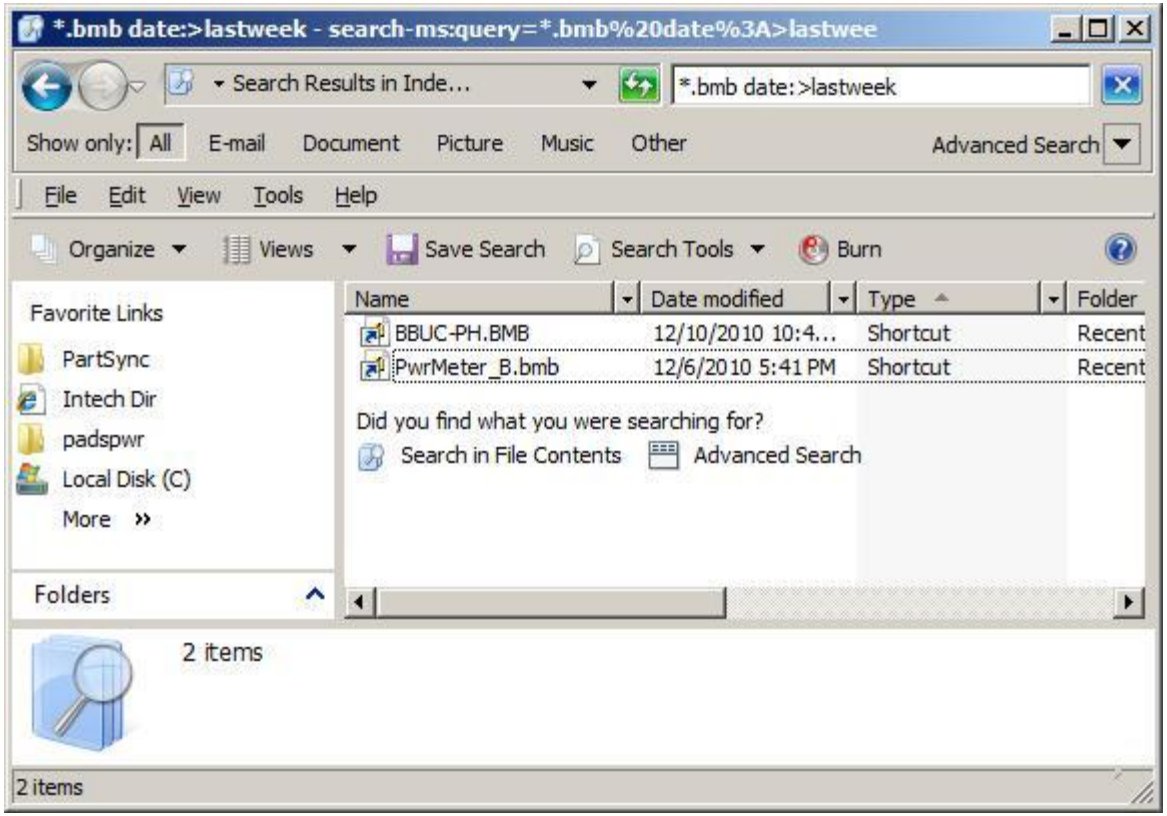
Do You Do This?



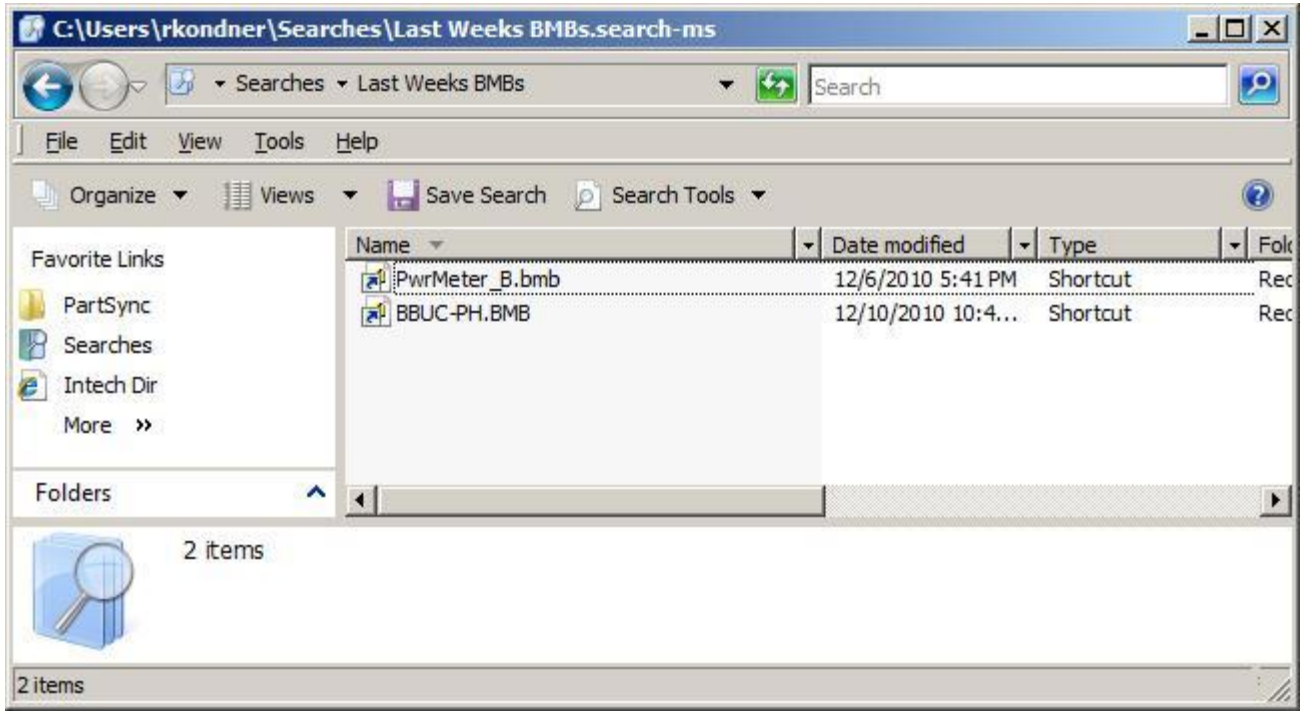
New Windows Features

- Preview Pane
- File Properties
- File Tags
- Indexing and search
  - Local and remote file share
  - Libraries
- Preview Pane

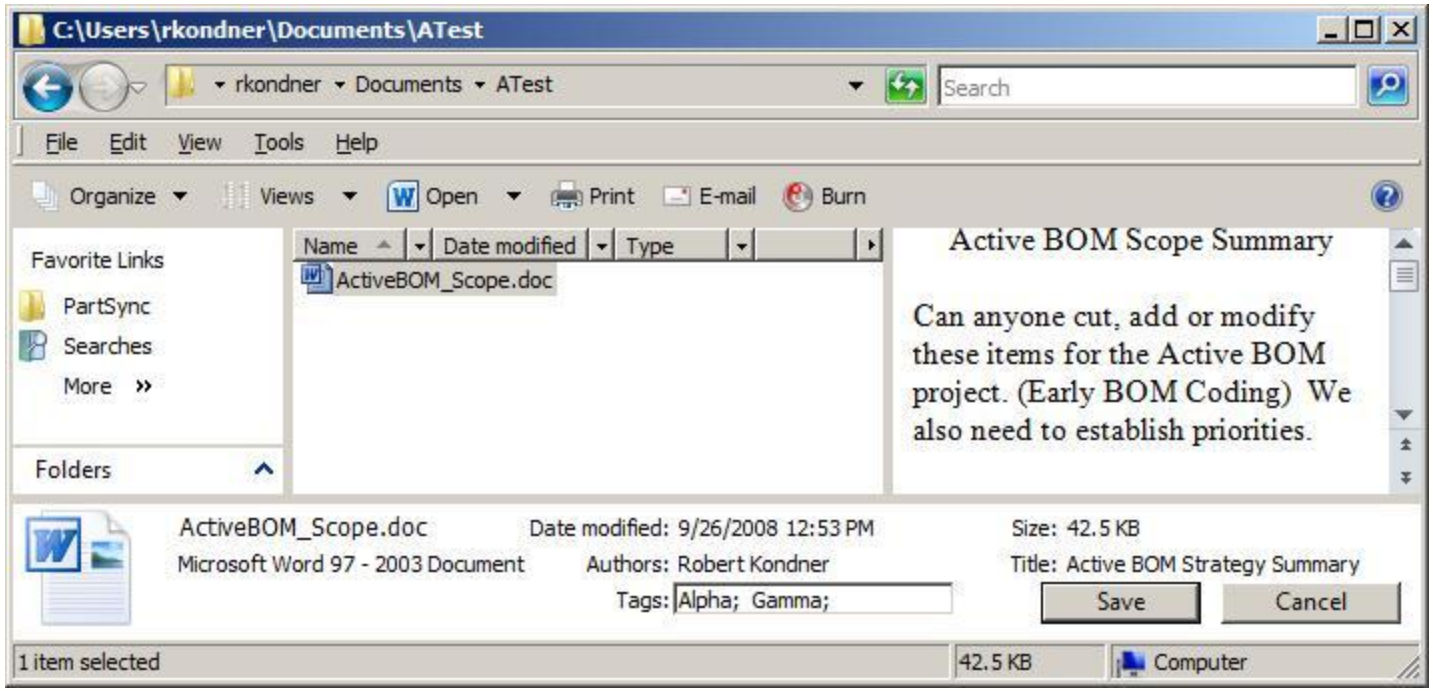
Index and Search



Saved Search Folders



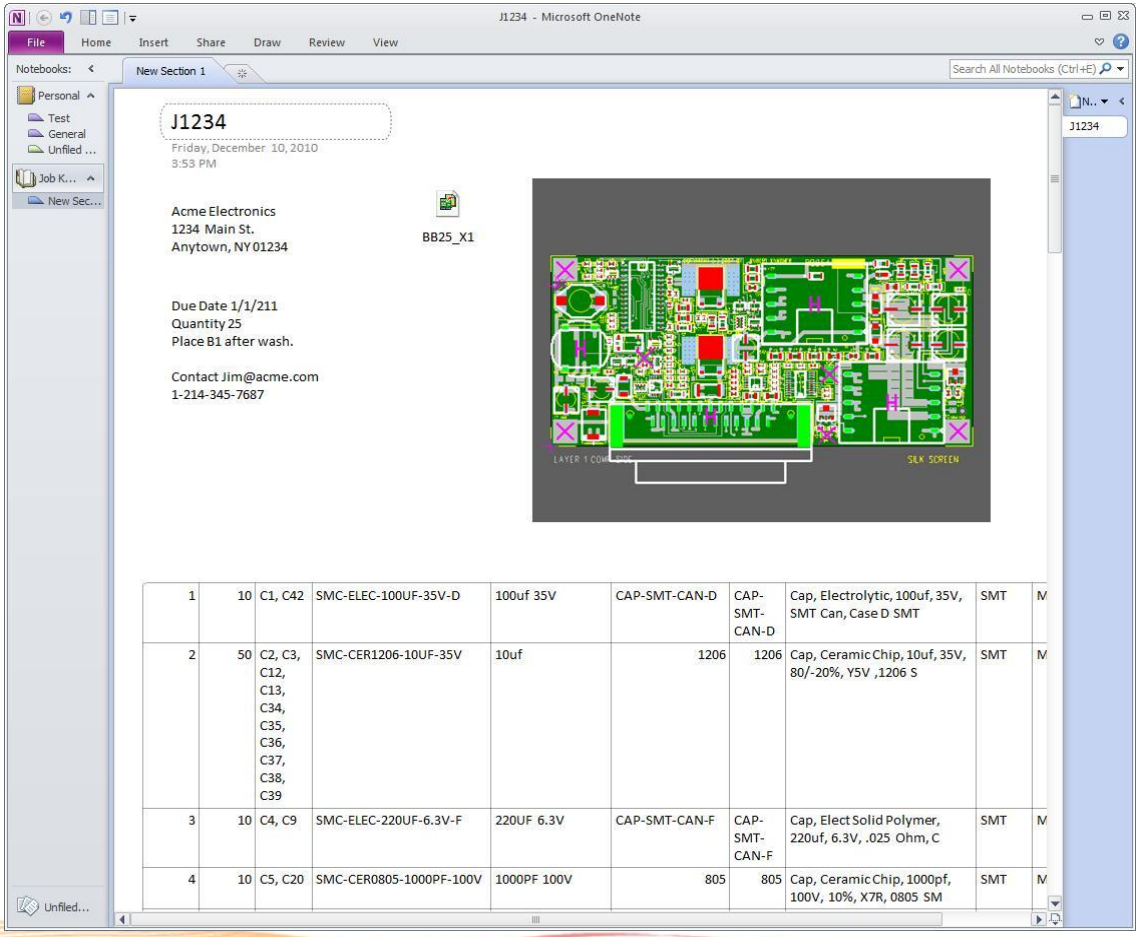
File Tags and Preview



Microsoft OneNote

- Very easy to use, intuitive.
- Add any file types.
- Simple “3 Ring Binder” logic.
- Internal search functions.
- Share with multiple users.
- Self synchronizing.

OneNote as Job Traveler

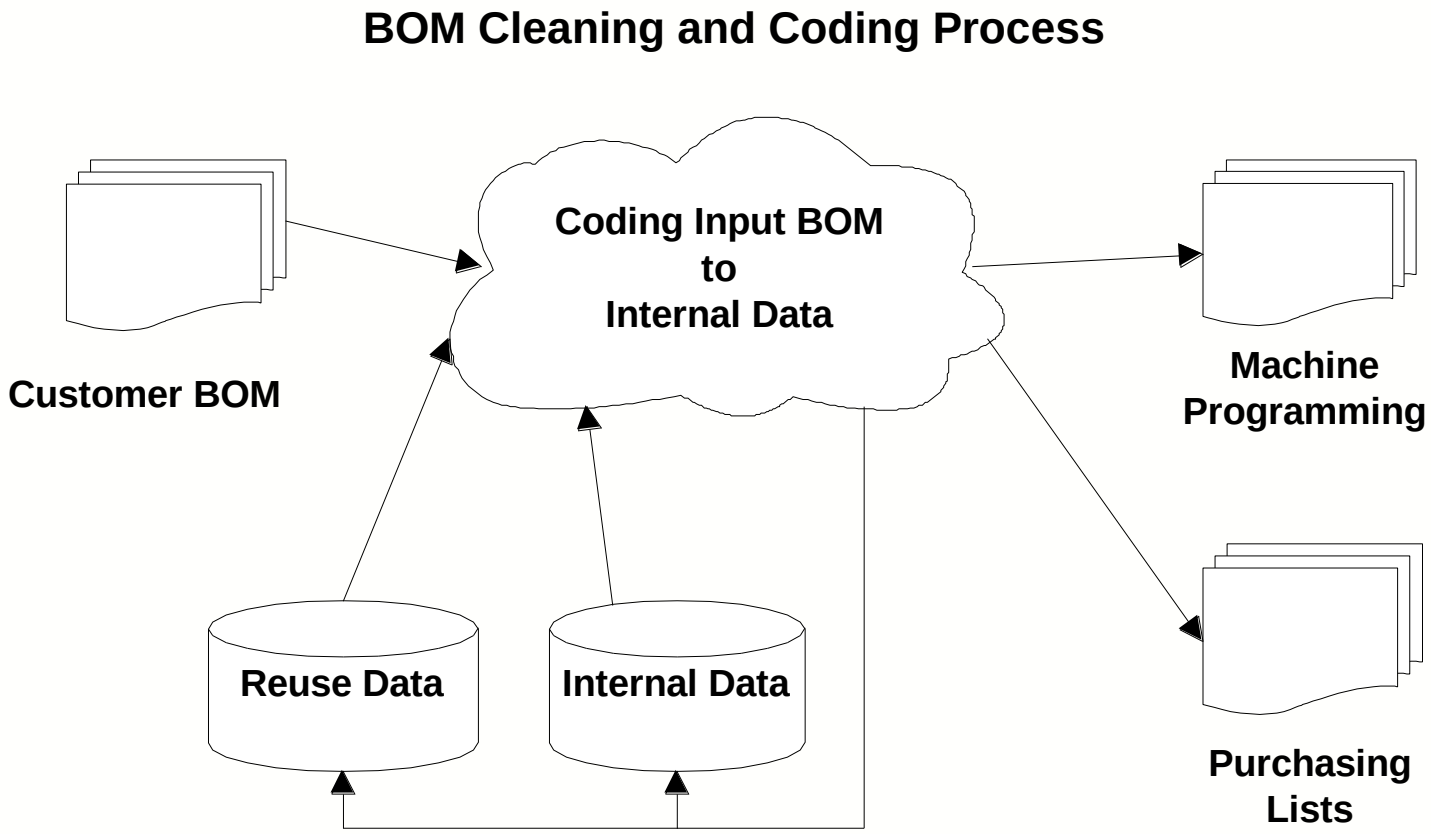


1	10	C1, C42	SMC-ELEC-100UF-35V-D	100uf 35V	CAP-SMT-CAN-D	CAP-SMT-CAN-D	Cap, Electrolytic, 100uf, 35V, SMT Can, Case D SMT	SMT	M
2	50	C2, C3, C12, C13, C34, C35, C36, C37, C38, C39	SMC-CER1206-10UF-35V	10uf	1206	1206	Cap, Ceramic Chip, 10uf, 35V, 80/-20%, Y5V ,1206 S	SMT	M
3	10	C4, C9	SMC-ELEC-220UF-6.3V-F	220UF 6.3V	CAP-SMT-CAN-F	CAP-SMT-CAN-F	Cap, Elect Solid Polymer, 220uf, 6.3V, .025 Ohm, C	SMT	M
4	10	C5, C20	SMC-CER0805-1000PF-100V	1000PF 100V	805	805	Cap, Ceramic Chip, 1000pf, 100V, 10%, X7R, 0805 SM	SMT	M

BOM Cleansing Process

- Decode user input, form line items.
- Search “Reuse Data” for previous items.
- Search internal parts libraries.
- Identify new Parts and Packages
- Pass data to Purchasing & Assembly

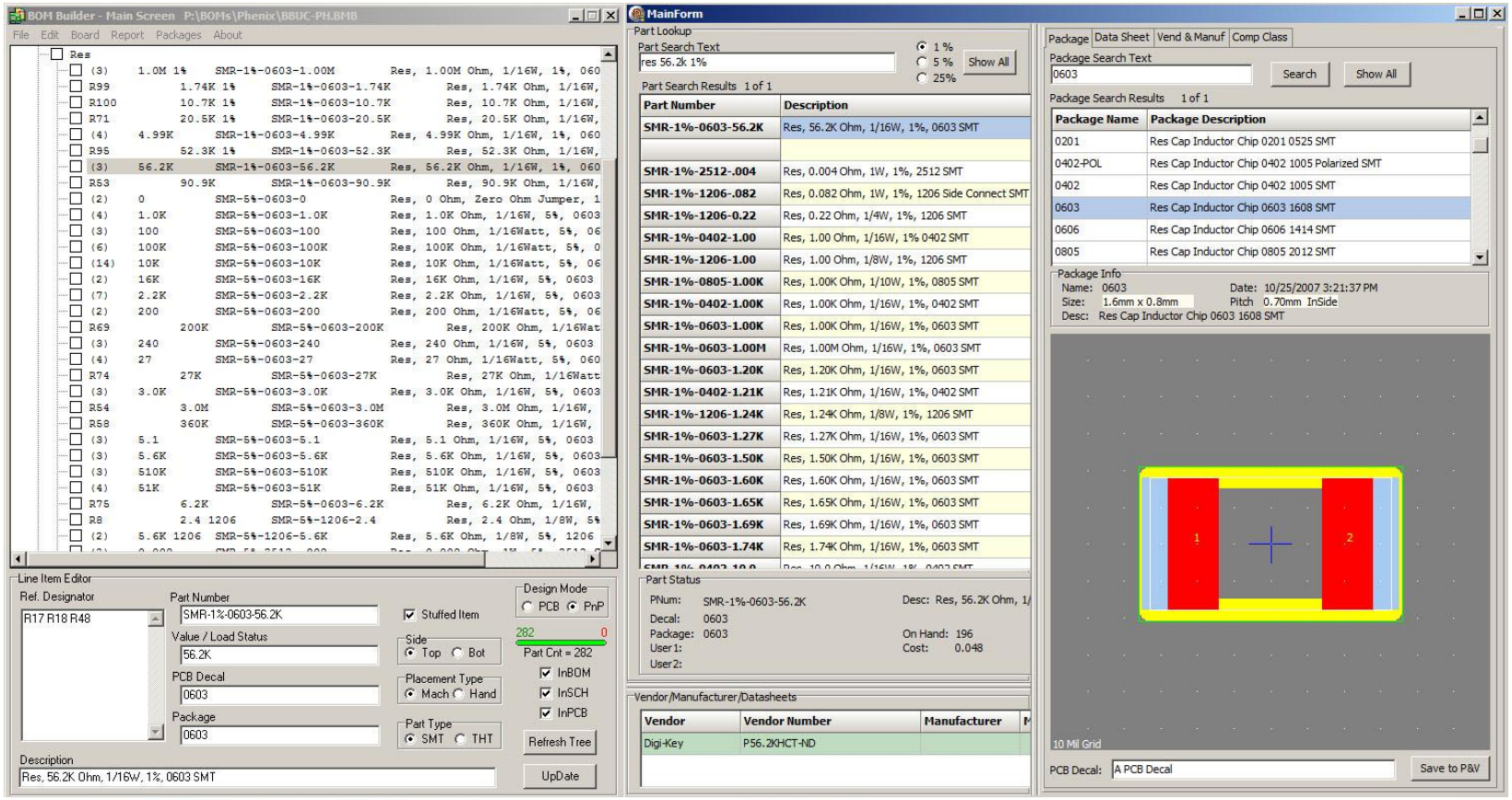
BOM Coding Process



Example Tool for Coding

- Display input BOM as line items.
  - Controls for Stuff and Hand/Machine
- Search of internal libraries.
  - Display component details for selection.
  - Provide data sheet storage and access.
  - Provide vendor information.
- Identify Assembly information.

Sample Tool for BOM Coding



Summary

- Use and enhance existing email tools.
- Define what info is needed where and by who.
- Major decision: Large or Ad Hoc tool?
- Organize PnP data for BOM coding.
- Use a consultant! ☺