

Using Information Technology to Enhance Commissioning

E. Lon Brightbill, P.E

Facility Dynamics Engineering

Synopsis

Essential to an effective commissioning process is effective management of and expedient access to the myriad of types of information required in the design, construction, commissioning, operation and maintenance of a facility. A software tool named ComIT (Commissioning Information Tool) designed for this purpose integrates all this information and provides a common graphic interface for accessing this information. The tool is designed to allow the project team (designers, contractors, Cx authority, project managers, operators, etc.) to collaborate on generating this data electronically and to synchronize their efforts via the internet. An "object oriented" method is used to associate requirements with typical components of a building. This allows rapid generation of repetitive information and allows the "knowledge base" to mature and incorporate the expertise of all involved. It allows you to efficiently track action items that arise throughout the process. Having access to all this information increases the quality of the Cx process. Being able to expediently query the project database not only increases efficiency but ensures a more thorough process. The electronic products of the tool are valuable source of information for the operators who can use and maintain the tool throughout the life of the building.

About the Author

E. Lon Brightbill, P.E., is a principal of Facility Dynamics, a company specializing in commissioning and controls consulting. After leading the mechanical designs of many large projects with a design consulting firm and managing large projects for an energy performance contractor, Mr. Brightbill co-founded Facility Dynamics. Mr. Brightbill has been commissioning projects for 11 years and has used his knowledge of database programming to facilitate his commissioning endeavors throughout this time. Mr. Brightbill has also developed software for the continuous commissioning of facilities using data from digital control systems. He has also developed Cx standards for large institutions.

Stating the Opportunity

Status Quo

The goal of the Commissioning Authority (CxA) is to ensure the overall functionality of a building. This requires that he be involved in all aspects of the buildings evolution from design, through construction, and start up and into occupancy. Therefore the CxA must not only be well versed in all these aspects, but he must manage the myriad of types of information required to get through the process. The status quo is generally to manage all this information via reams of hard copy forms, memos, manuals, etc..

In the generally accepted commissioning process, the responsibility for generating/providing the required information lies with many different parties. Designers generate design intent information and most of the contract documents, contractors and vendors generate performance, nameplate, operation and maintenance data, some of the training documentation and sometimes startup procedures and documentation thereof. The CxA typically will generate the, Cx requirements, functional performance testing procedures and documentation thereof, and track Cx related issues. The fact that the information emanates from multiple unrelated parties, is perhaps the primary reason that the Cx process has not kept up with current information technology. The common platform eventually boils down to the printed (and copied, and copied, and copied...) page. This also forces the person responsible for that information to make any modification to it required by any other party. The process is typically submittal of a number of iterations until an approved version is accepted.

In the course of generating the information, much is done directly on paper perhaps using a word processor or spreadsheet computer application to create store and print hard copy forms or templates. For instance, nameplate data templates, or start up checklists, might typically be managed this way. A master nameplate data template and a start up checklist form will be saved for a typical air handling unit. Copies of these templates will be printed and information will be completed by hand on hard copy and submitted as the final deliverable. Perhaps, to make the deliverable more presentable, this information might then be entered into the master application and saved specifically for any given unit. The result is that there are many individual files with information entered "electronically" with little practical means for using that information for other purposes (ie: transferring the nameplate data information to a maintenance management system)

The Cx provider will typically save a "master" version of these templates and checklists. As the provider gains more experience, the variety of systems (and the variations on each of those systems) for which they have dealt with begins to multiply. Therefore the number of "template" files begin to multiply. At the same time, the quality or depth of the processes will expand as different applications arise requiring more advanced procedures and as their personnel gain more experience. When it comes time to add a procedure or bit of information to the default, the Cx basically has to go back to all the template files and incorporate the additional information into all the templates. When you factor in that many different people with varying expertise have input to this information, it becomes quite a management challenge to keep a comprehensive set of masters. Keep

in mind that, since the information is best presented at a system or unit level, this is typically the level at which the templates are saved. Therefore the options exponentiate as the "knowledge base" expands and matures. This process is not conducive to collaborative input from many people and generally does not result in the most mature set of standards.

Throughout the Cx process, the commissioners in the field need access to much of this information. They need the performance information, the nameplate data, start up procedures, contract requirements, functional performance tests, etc.. If a Cx process is optimally integrated with the construction process, it is not always easy to predict exactly which information you will need on a given day of commissioning. For instance, if you uncover a problem with the first of a sample that obviously needs to be addressed on all the other units, you notify the contractor of the required resolution and move on to something else as opposed to completing a sample that you know you will fail. Another instance might be that you are asked to assist in resolution of an issue that is affecting the schedule so it becomes a priority. Because of these realities, you either have to drag reams of information along with you into the field, or you keep going back to the office to get the required information.

When managing and/or tracking issues that arise throughout the construction and commissioning process, here the status quo varies. This is one area that information technology has started infiltrating the construction process. Issues are typically tracked via two mechanisms; a "Request for Information" (RFI) form and some kind of punchlist. The most basic form of tracking issues is conventionally handled via hard copy forms. These are stored in file cabinet somewhere. When an issue involves a multi-party, multi-iteration discussion, individual responses can be inserted throughout the file. Reconstructing the discussion at that point can be time consuming. Ensuring every issue is closed becomes a cumbersome manual task inviting things to "fall through the cracks". Many construction managers have realized the power of information technology and have either been applying software packages designed for issue tracking or developing their own based on a spreadsheet or a database application. This facilitates the construction phase of the project, but rarely is the application provided to the owner for his use during early occupancy.

As one of the key aspects of commissioning is ensuring adequate training and turnover of the facility, much of the information generated throughout the design and construction process must be turned over to the owners and operators (this typically happens between start up and the time of initial occupancy depending on the specific Cx protocols). Many operators experience information overload during this time as the construction manager pushes the schedule. Hence the operators must absorb a lot of information quickly. Of course any credible commissioning process will require that this information be documented for future reference. However, with the complexity of modern day buildings, this hard copy documentation gets put in one of those notebooks that fill the bookshelf. When it comes time to refer to that information, finding it can become a daunting challenge. Besides that, if you aren't where the bookshelf is located, the information is of little value.

The Opportunity

Of course, there is a better way. In case you haven't heard we are in the midst of an information revolution. The power readily available from this information technology has revolutionized many industries. The opportunity to fundamentally enhance management of Cx information lies in applying this information technology to commissioning. Imagine if all parties to the Cx process could collaborate on generating data electronically that is either in a common standard format (where practical) or custom to an application but is in an accessible electronic format. Imagine if all this information were accessible through a common graphic interface that is related to specific building elements or processes. Imagine if one set of defaults could be maintained as a byproduct of the Cx process that realistically allows multiple parties to collaborate on developing a knowledge base that can be easily applied to older projects. Imagine being able to apply the knowledge base and generate a comprehensive set of requirements at the click of a button even on custom or unique equipment. Imagine all information required in the field being readily accessible through a field computer. Imagine automatically populating an asset or maintenance management system from data generated during the Cx process. Imagine delivering to the Owner an interactive application that provides access to all this building information. Such an application could fundamentally enhance the Cx process.

Why should the CxA be concerned with this? Because, as stated previously, the CxA is involved at all phases and has to deal with and manage all this information anyway. The CxA is (or should be anyway) solely an agent of the Owner and should endeavor to provide the best product he can and make the turn over of the building as smooth as possible. The CxA can solicit participation of all the parties in the common goal of providing a better deliverable to the Owner.

One Solution

ComIT

An application has been developed applying information technology to the Cx process to enhance the management of Cx information. The application entitled ComIT (COMmissioning Information Tool) uses a relational database engine to generate, record and/or link to this information. It establishes a common convention for representing building elements and commissioning processes. It uses an "object oriented" methodology to reference building elements and store/generate applicable requirements that make up the knowledge base. It adds a graphical interface to ease access to information.

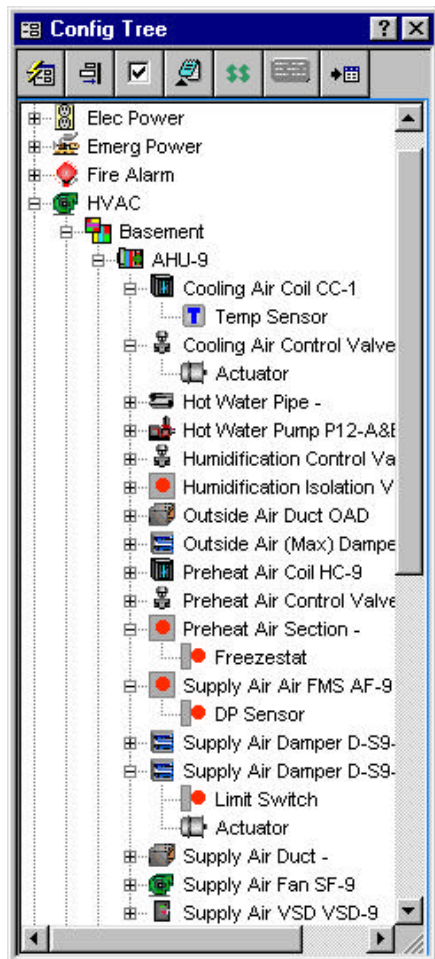
To allow other parties to participate in the process, ComIT comes with a shareware viewer application that gives whoever wants to view or enter information the ability to do

so without having the entire application. This is typically distributed to all contractors and/or entities that will be participating in the process.

The fundamental concept behind ComIT is that common and definitive relationships exist between elements of the systems and between the elements and their requirements. This allows a consistent and repeatable way of visually referencing elements of a building. Furthermore, by predefining these relationships and associations, once you configure your systems and equipment for a given building, you can quickly compile a comprehensive set of requirements. Once the building elements are configured for a particular project, this provides a convenient roadmap for storing, relating, and accessing information associated with those elements whether that be contract requirements, "action list" items, nameplate data, performance and submittal information, start up and testing procedures, operation and maintenance data, contact information, etc..

ComIT uses the *power of suggestion* and couples that with the concept of "never author something twice" to allow multiple parties to build, mature, and use a single comprehensive knowledge base. By the power of suggestion I mean that at any point in the project, the user can ask ComIT to suggest any piece of information from the knowledge base that does not exist in the current project. You can then either accept or reject the suggestion. Therefore if someone has added to the knowledge base since the current project was configured, you can easily apply it.

If you find you need to add something to the knowledge base, you do that directly to knowledge base at the lowest applicable level and let ComIT suggest it back to you. Once units are configured, they can be added to the database of standard configurations.



Configuration Elements

The object oriented concept is well established in the IT fields such as programming. For instance BACNet uses an object oriented approach to define control system elements. Objects are generally small reusable entities that when coupled with other objects, can build to a large complex application. Multiple objects can be packaged into a composite object. Expanding on the object oriented concept and relating building elements in a hierarchical fashion provides an intuitive and comprehensive way of representing building elements. Each object has a set of properties that represent key bits of information applicable to that object. These properties can then be grouped into collections for context sensitive access (nameplate data parameters, performance data parameters, typical start up procedures, tests required, etc.).

Objects in ComIT are typical building elements. Taking an air handling unit as an example, the air handling unit and its controls are made up of a variety of discrete elements. The air handling unit is represented as one item at the highest level. It will typically be composed of fans, coils, filters, dampers, etc. These "components" are generic objects and generally independent of the function they serve. For instance a fan that moves supply air is basically the same as the fan that moves the return air. The water coil that heats the air is basically the same as the water coil that cools the air. They may be of different sizes and capacities but the properties that define them and the procedures to start them up are basically the same. Objects at the second level referred to as components may be composed of other objects, for instance the fan will include a motor and possibly a belt drive, etc.. The control system elements are objects that are typically at the third level of the hierarchy and are associated with the elements that they control. For instance an averaging temperature sensor sensing the temperature downstream of a coil would be an object that serves that coil.

This hierarchical object oriented convention is key to ComIT and provides the object tree that represents all the elements in the building. This configuration tree becomes the roadmap for relating and accessing information throughout the building. Navigating this tree to the applicable object provides immediate access to all associated information. Units such as air handling units, or systems such as a secondary chilled water system, or zones such as a BL3 Vivarium are objects at the first level. Since each of these are assigned to both system types and location groups, these provide two other dimensions for summarizing, filtering and accessing information.

The buttons across the top of the configuration tree represent property collections such as associated action items, start up and check out procedures, functional performance tests, nameplate and performance data, and "info links". The last button exposes a very functional set of properties that basically record links to any other type of information accessible by the computer running ComIT. Again these are generally linked to contract document requirements (contract drawings and details and specific spec paragraphs), performance data, manufacturer's web sites, O&M information, manufacturer's specific installation and start up instructions, and anything you can imagine.

Graphic Interface

For the very novice or infrequent users of ComIT, ComIT provides an optional penetrating graphic interface typical of those you would find on the GUI of most modern day control systems. This allows navigating graphics, floor plans, and schematics, or whatever to find the object you want. When you click on the hot spots defined for applicable objects, ComIT opens the configuration tree with the applicable object highlighted. Again from there you can access any related information. The following two figures are screen shots that represent typical graphics.



Figure A: Typical Splash Graphic

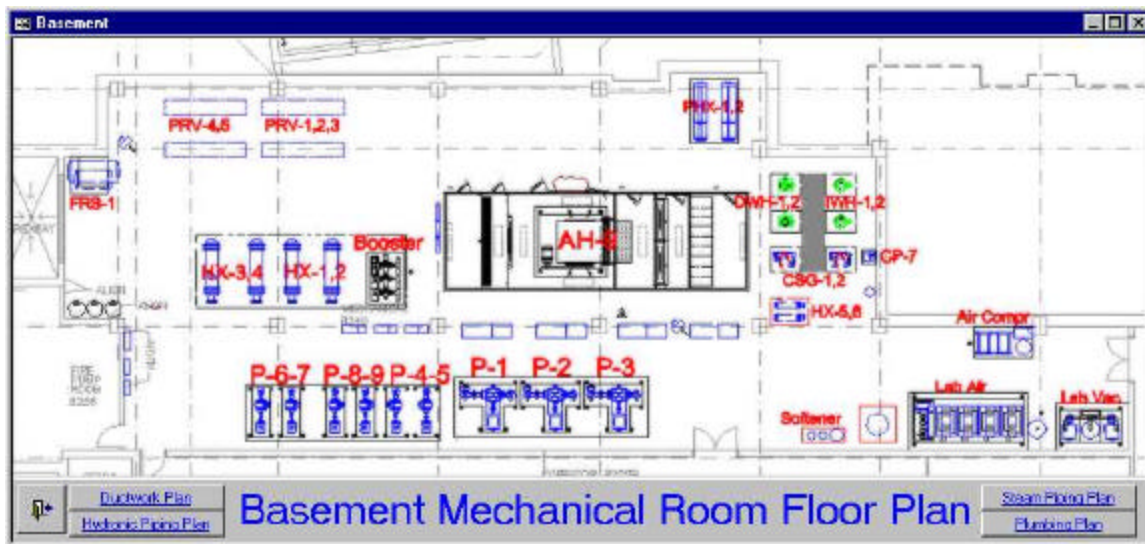
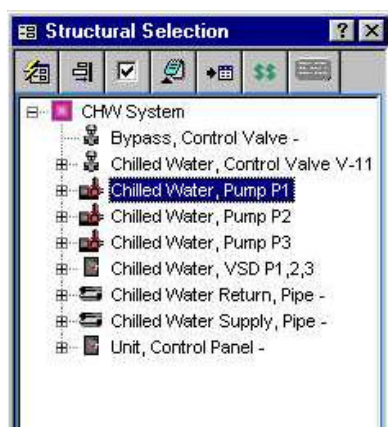


Figure B: Typical Floor Plan Graphic



When you click on P-1 for instance, the adjacent tree will open .

Note that besides hot spots on the floor plans, there are also links to the contract drawings that cover the applicable systems.

Accessing Information

As stated previously, the configuration tree is the primary method for accessing information. However, since ComIT is based on a relational database, your access to information can be via practically any delimited bit of information. You can generally sort and filter information and reports based on axes other than configuration. For instance you can sort and filter for party responsible, dates, status, location, system type, etc.. This is supported across all types of information wherever the parameter is applicable.

There are two basic ways of storing information; storing it within the project database, and storing it external to the database with a reference to that information stored in the project database. Both types of information are accessible via the same menu. The only requirement for accessing information external to the project database is that the computer being used to run ComIT has to have the application software that the file requires. Therefore we typically agree on the family of software that will be used and stick to it. An example of applications that would be used include; AutoCad for drawings; Microsoft Office applications (Word for specs, Excel for calculations, Power Point for training); Adobe for scanned images, submittals and cut sheets, and a lot of manufacturer's information (.pdf format has become a defacto standard for providing information electronically via the internet); and of course any browser for html documents.

Obviously the easiest way to integrate the diverse types of information is if you can get the information electronically in the first place. Different sectors of the industry are at different points on the "electrization curve". Most design entities produce all documents electronically using a CAD package for drawings and a word processor for specs. Over the past few years, manufacturer's are rapidly providing their cut sheets and typical submittals either as html documents or in .pdf format. The provision of O&M manuals electronically has been a slower process but is beginning to gain momentum. Of course the last resort is to scan the printed image.

Database Model

The database model that we chose to support this is the distributed database with synchronization. With this model, each person/entity has their own copy of the project database resident on their computer. That person/entity periodically synchronizes with a hub replica of the database that is stored on an internet server. Any time a synchronization is performed, assuming permission is set to send and receive changes, all changes in the hub are transferred to the remote database and all changes to the remote database are uploaded to the hub. Both databases are now identical replicas. This model permits using the application in the field when you can not be connected to any server. This is when much of the information is generated.

Supporting Either Paper Or Electronic

While we generally encourage the entire process to be done electronically and use field laptops as essential equipment during commissioning, we realize that this can be just a little too much of a culture shock for many of the parties collaborating in the process. Therefore ComIT supports printing templates for various types of information that allow the information to be initially completed on paper then transposed to the database. Templates are provided for start up checklist, start up tests, and functional performance tests, nameplate and performance data, and facility specific numbers.

Contrasting Specific Tasks

The following illustrates some specific tasks and contrasts the conventional approach with the automated approach.

Planning for a Days Commissioning

With hard copy documentation and all the channels information has to go through, information tends to be somewhat old. Commitments made by some contractors relative to scheduling tend to include assumptions that are sometimes out of their control so periodically you come to test something and it is not ready. Sometimes the paper work had already been submitted indicating there was an issue that arose but it arrives on your desk after you've already departed for the field. If there are known issues surrounding these systems on which you are testing, you hope you remember all of them and/or had read all those memo's because it is impractical to search through the files to find any open issues relative to system your testing that day. This means you can show up to commission something and find that it is not ready or you may be uninformed of an issue surrounding that system that needs to be checked during commissioning.

Automation helps here in a number of ways. If everyone is participating electronically, you can check the status of that particular element "near real time". For instance from your office at the beginning of the day or from home the night before you can synchronize and check the start up status of that particular element. You pull up and read only the issues that are associated with that element and with each issue read all the dialogue from all parties. You can also filter all start up and testing procedures for which the contractor is responsible and filter for any that are incomplete or failed.

Another issue related to planning is that you can not bring all the reams of paper with you for every system every time you go into the field. Therefore you take with you what you think you will need. If you get diverted, you may lack the required information. If everything is electronic, it is all in the computer and immediately accessible.

Performing and Tracking a Failed Test

One important aspect to a quality Cx process is to follow up on the many issues that arise and need resolution throughout the process. In the conventional scenario, when testing a complex system, you will typically be going through a many-page testing scenario. The more room you leave for field notes the more pages you have. As you begin testing you typically identify many little things that may be an easy quick fix. If you working with,

(as opposed to against) the contractor, you will typically send him off to address some of those little issues as opposed to failing the test. You move onto something else and come back to the previous issues after that they are resolved. Therefore you are not always waltzing through a sequential process. Therefore you find yourself paging back and forth through the procedures to locate the tests. You are also scribbling in any available margin space, some cryptic note to document the findings and associated issues. Alternatively you are using a lot of reference codes to other pages of notes. When you fail something, obviously, something needs resolved so an action item needs to be generated. Hopefully you will document that immediately and throw it in the file of action items. Alternatively, you might wait until you get back to the office and try and reconstruct the action items which leaves room for things getting overlooked. Typically, all the action items are then periodically submitted in groups, again with all of these needing to go through the proper channels. This submission process again will typically cause slight delays. Then, if the contractor happens to disagree or wants to raise an issue on it (perhaps calling it a design issue) this will spawn a bunch of memos from everyone involved over the next few weeks. You may or may not get copied on all these. If you do they will typically be scattered through the correspondence file. As it may be your role to advise on who is responsible, collecting all the input to the issue can be a time consuming task. If the issue is not a significant one, the action item will be filed with all the other action items in the middle of some thick file somewhere. Anyway, after all of this testing and chicken scratching all over the test forms, you have to get back to the office and make it more presentable before submitting it for record.

Automation can help these scenarios. With information related with specific objects and those objects accessible via a graphic interface, you can easily navigate to any specific information such as a test you want to perform and document. When that test fails, you immediately jump from that test form to the action list to enter the required action. This automatically associates the item with that particular element. Therefore when you come back to close out this system, you can immediately access the status of the action items associated with it. Furthermore, at any point, the contractor can filter for all the items for which he is responsible. Again this will be much more near real time. Therefore if you identify an issue one day and he is sending his technician out to wrap up a system the next day, he will know about it. Otherwise this item may be put on the back burner as the contractor moves onto other system. From your perspective, when you leave the field your task is done. No transposing information to make it presentable. If someone wants a hard copy, you simply print it whenever it is requested.

Collaborating on Information

A fundamental difference between the conventional hard copy approach and the IT approach is the basic ability to collaborate on information. The typical approach for developing draft checklists for instance would be to have one party generate them, distribute them to multiple parties, return comments from multiple parties, incorporate the comments, redistribute, and around again as required until a final set is approved. With ComIT, parties simply add to the procedures directly, which can typically take less time than generating the comments.

Preparing Information, Accessing it the Field and Using It to Create a Directive

Throughout Cx, you need to access a lot of information in the field such as pump curves, details, manufacture specific installation requirements, etc.. This information may typically be included in the shop drawing submittal or other information that comes across your desk early in the construction process. This is when I would typically assess what I would need in the field and get it copied and filed for later commissioning because I did not want to drag my record copy into the field and mark it up. Then, as illustrated above, I would take specific pieces of information into the field. When in the field, if I was lucky enough to have what I needed, I would dig through the files, drawings or specs to see what I need.

Now instead of getting something copied, I get it scanned. The software we use scans and performs ocular character recognition so, especially when a lot of text is present, the file sizes are relatively small. Furthermore it usually saves me typing when creating the procedures as I can simply copy and paste. Instead of a piece of paper, my clerical person then gives me back the file and I associate this with the project configuration. Then when I am in the field, I can pull up any design drawing, detail, specification paragraph, performance curve, etc. directly associated with the item I am working on.

Often I would want to issue a sketch to clarify a directive. I would want to use the design drawings as a background so I'd pull the top off the copier and wrestle the drawings (which by now are ripped, smudged, and faded) onto it trying to position it in the right area for the copy; usually adding another tear to the collection in the process. I then add my edits to the copy and fax it. If who I faxed it to was lucky enough to be able to read the fax, who they in turn faxed it to usually couldn't.

Now I simply pull up the design drawing, clip the applicable portion, add my edits in cad, print it to a *.pdf file and email it. That email can then be relayed indefinitely without loss of quality.

Maintaining A Master Set Of Procedures

Maintaining a master set of procedures with many people involved and many diverse systems included in the master can be a daunting challenge. Since here there are probably as many ways to manage this as there are companies attempting it, I will simply indicate how the process works with ComIT. Say someone in the company comes across a new test that he has to perform on a piece of equipment that is not in our standards but should be. Whoever has to do that, simply adds this test as another test associated with that specific element to the master database. He then asks ComIT to suggest it back to him to be inserted in the project file. If I had configured a system using the same component on another job prior to this and later start testing that job, this test will be suggested to me even though I might not be aware that it was even added. At that point I can either accept or reject it for my current project. If I had already taken a master Word form for instance and customized it for this job, it is unlikely that this new information would get incorporated in the older job.

Presenting Turnover Documentation to the Owner

One of the most valuable byproducts of all of this is the tool this gives the owner for accessing information. As Cx already requires training to be documented for future reference, having this provided electronically and linked to applicable building elements provides yet another enhancement to what you are delivering to the owner. While most of the wall full of notebooks is still provided, this provides a more efficient means for accessing the information. Furthermore, it then becomes portable. For the typical building engineer who might respond to an alarm using his laptop, he now has all access to all this information wherever he is. If the owner has a maintenance management system, information can be transferred to that system without retyping it. I see this as an important asset to the overall commissioning deliverable.

Lessons Learned from a Field Trial

While we have been using the application for many years for our scope in Commissioning, we have only recently begun requesting, then requiring participation from the other entities on the design and construction team. Internally the process is easy and smooth as we obviously keep current with automation tools and are very computerized so we deal with this stuff routinely. This section really focuses on the lessons learned from applying this type of management tool across the entire construction team.

The first application of multi-company collaboration was the Louis Stokes Laboratory at the National Institutes of Health. This is a six story state of the art research laboratory. We had developed the commissioning procedures before the Viewer application was ready for distribution. Therefore the Cx requirements and protocols were basically the traditional approach. Prior to the construction phase we had already configured the project in CACEA (Controls and Commissioning Engineering Application) as we were asked to negotiate the sole source controls cost for this project. (CACEA is the parent application from which ComIT is the commissioning related modules. CACEA further supports controls design and estimating using the same object oriented approach.) We had at this point anticipated using it internally and proceeding with our conventional approach to Cx, part of which requires the contractor to customize start up checklists (with manufacturer specific requirements) with nameplate data included. Early in the Cx phase we offered ComIT to the various contractors as a source of extensive generic checklists. That was presented as purely their option and their use of it relative to the testing and checkout was purely voluntary. We did ask nicely that they at least use it to complete nameplate data electronically as we consider the hard copy submission of that information now-a-days simply foolish. I was actually surprised when all parties (CM, AE, GC, MC, EC, and ATC) agreed to participate in the full functionality. Considering that the GC, MC, and EC had one computer on site between them and none had an email address that could be accessed at the site, I figured they were in for some culture shock.

Never-the-less, largely in order to support ComIT participation, they all got computerized and got site internet access. Personnel were assigned from each company to be the team representative. We conducted a one day training session and we were off and running (or

perhaps crawling might be more appropriate). The road was a bit bumpy at first as I struggled to relate to their learning curve always over-estimating the speed at which everyone was digesting all this. Many were digesting basic computer concepts as well as the Cx process in general simultaneously with all the features of ComIT. So there is the first lesson learned. When introducing this type of automation, do not over estimate where many of the parties in the construction process are relative to basic computer literacy let alone the concepts of automation.

Another early bump was what we eventually found to be conflict between some installations of America Online and Microsoft's ftp services which prohibited a few computers from synchronizing via the internet. We set up a site work around for what turned out to be one computer. This specific issue is not the point here, it is that these types of issues are sure to arise with complex automation. Lesson number 2 therefore is that, with any complex automation technology there is always a technical support burden when you try to deal with many different platforms and configurations.

Another issue that became very evident is that it takes a while to change a mind set that is built on paper exchange. The current mind set is to generate and file a piece of paper on everything. That piece of paper seems to make people feel more secure or something! It took a while for participants to see this as other than just another way of generating that piece of paper. ComIT lets you skip most of the paper production, yet at first some parties insisted that things be passed around in hard copy even though everyone could view the information more precisely, efficiently, and sooner electronically. To document interim versions when that is necessary, I advocated simply printing to a file and saving that file. Over time my perception is that people begin to catch on and see the power of the automation and slowly start taking advantage of it. So lesson number 3 is allow people time to digest the basic change in the way you can do business.

You must also acknowledge that there is an investment cost for all of this. Although this clearly pays back many times, the up front coordination and training effort may make you a little uneasy at first and you must build that into cost distribution.

For the ultimate application of all of this you can eventually be dealing with a lot of information. If you intend to take full advantage of passing this information around via the internet, another lesson is you can never have too much bandwidth! While this will soon be a non-issue, there is still a cost and nuisance factor in acquiring optimal bandwidth depending you location. A typical 384 DSL connection is more than adequate but not everyone is willing to bite the bullet even if it is available. Most everyone on the NIH-LSL job had 56K dial up access to the internet except the Owner (NIH) and the CM who were on the NIH LAN where the hub was hosted.

Summary

The power of information technology can greatly enhance the Cx process. It can make you more efficient at generating procedures and maintaining a collective knowledge base to support your commissioning. If you are a large institution that has many different companies performing your commissioning, it can help you establish the standard of care that you expect and ensure you have an effective information access tool for your

operators. It can help you more tightly integrate the Cx and the construction process to minimize the burden Cx places on the construction. It can foster collaboration amongst the Cx Team in generating and processing data. It can also provide a powerful tool to the Owner for getting to know his building and tracking that information throughout its life.