WHITE PAPER

August 2017

Simplifying Industrial Programming in the Classroom

Jon Breen, Owner
Breen Machine Automation Services, LLC.

EXECUTIVE SUMMARY

The world of technology is changing more rapidly than ever before, and many strategies and technologies are being introduced to help manage this change. One of those technologies is virtualization, which can offer fast recovery options, enhanced security, improved team efficiency, and fewer frustrations with disparate software versions. Industry and academia are increasingly leveraging virtualization for these benefits, but when it comes to its application in PLC programming, there's frequently a lot of confusion. Faculty may have little or no experience with virtualization. The IT department usually does, but IT's needs are very different from PLC programmers' and instructors', so solutions developed by IT may be sub-optimal for a programmer or instructor's purposes.

A straightforward and inexpensive process can be employed to take full advantage of the benefits of virtualization, while minimizing confusion. This whitepaper discusses key technical areas, as well as a step-by-step implementation plan for universities and tech schools.

Creating a Proactive Approach to Classroom Technology Management

The world of technology is changing more rapidly than ever before, and many strategies and technologies are being introduced to help manage this change. This can have a double impact for schools who want to stay relevant in industry and best prepare students for the real world, but have classroom needs as well. Many times faculty are no longer working in industry, and may struggle to keep up with changing industry practices. And to further complicate matters, the anecessity of security is increasingly straining relations between IT and end users.

It's not hard to see how the chaos can take time away from course preparation and delivery. Either faculty needs to spend extra time managing lab technology, or risk poor course performance and bad student ratings. Adjunct instructors, making up "more than 70 percent of all college and university faculty," typically aren't paid for prep time, so any extra management burden is likely to make them second guess their choice to teach.[1] In today's landscape of budget cuts and volatile politics, even prestigious universities like UW-Madison are having trouble holding on to faculty, as described by Will Jones: "People are here out of dedication to the institution and it is an institution worth being dedicated to. But that only goes so far."[2]

This is a frustrating scenario for everyone. Will students keep coming if the content isn't relevant or delivery is plagued with technical issues? Will faculty choose to continue teaching in this fashion? Can a school afford to ignore this growing technology management issue? The best approach to classroom technology management is to be proactive - to create a system that supports faculty in delivering content efficiently, keeps courses in line with industry needs, and addresses security concerns. The following will outline a simple, cost-effective approach to doing exactly that using a technology called "virtualization." This will be broken down into the following sections:

- What is virtualization?
- Why is virtualization relevant? How is virtualization used in industry?
- Strong security without interfering with education
- How faculty can save time with virtualization
- Smoothing relations between faculty and IT
- Creating a uniform and manageable student experience
- What does virtualization cost?
- Implementation plan
- A future-proof approach (multiple operating systems and software versions)

After reading this whitepaper, you will be able to answer these questions:

- How can I save time managing classroom technology?
- How can I reduce technical issues to allow faculty to focus on teaching?
- How can I allow students the freedom to explore their software?
- How can I enable smooth communications between faculty and IT?
- Does virtualization fit my budget?

BACKGROUND

What is virtualization?

Virtualization is a broad term that means to simulate a computer. In practice, this usually ends up looking like one computer that pretends to be several computers at the same time. If you've heard of "XP Mode," that's a virtual machine (VM) that allows older programs to be run on newer operating systems (OSs), like Windows 10. In this case, the computer is running two OSs at the same time, where normally a computer is only able to run one at a time. VMs are also commonly used by industrial programmers and educators because they give a user a lot of flexibility in how software is handled - running legacy software, enhancing security, and improving recovery options to name a few.

Figure 1.0 Several Virtual Machines on One Physical Machine

Why is virtualization relevant? How is it used in industry?

In industry, it's common for a manufacturer to have many pieces of automation with varying support software requirements for maintenance. Perhaps a few pieces of automation require a 32-bit computer to program/troubleshoot, while the rest of the plant uses 64-bit software.

Or maybe some automation uses FTView v6.1 and other equipment uses v8.0 (these can't coexist on the same PC). This frequently leads to saving old laptops with specific software installed and keeping multiple programming computers with different versions of the same software to support equipment. Manufacturers are finding they can consolidate all of these programming computers onto one computer using VMs, thus enhancing the lifespan of the software.

From an OEM's perspective, work requires even more software versions from more vendors, further increasing the proliferation of extra programming computers. Imagine a stack of laptops under each programmer's desk. OEMs usually have a team of people that individually install much of the same software - duplicating the effort. In most cases, the stack of laptops can be eliminated and software installation can be done once for the whole team by using VMs.

HELPING EDUCATORS STREAMLINE CLASSROOM TECHNOLOGY

Strong security without interfering with education

IT will typically lock many Windows settings and install or enable things like anti virus and firewall. This makes a lot of sense from a management and security perspective, but it can also prevent many activities important for industrial programming. Common, trivial activities, such as changing an IP address (which industrial programmers and maintenance techs do all the time) are impossible when Windows settings are locked. Antivirus and firewalls can also interfere with industrial software and programming, but they're there for

a reason. How can we be secure and still provide a meaningful lab experience?

There's an easy way to find middle ground on security, and if you haven't already guessed, it's virtualization. A VM can be separate from the school network and assets, meaning there can be fewer security measures and less red tape in the virtual environment. The physical computer can have normal security measures, while the virtual computer allows any configurations, installations, and other activities required for the class.

How faculty can save time with virtualization

Scenario 1

You're teaching a lab course, walking students through a key concept, when someone raises a hand, "I don't have that option," or "I'm getting an error message." You take a walk over to the student's workstation to set things straight, but when you get there, all you can do is scratch your head for a few minutes and ask the student to work with his/her neighbor for the rest of the course. How did the workstation get so different from the rest? You dread the time you know you'll have to spend troubleshooting and reconfiguring.

Scenario 2

All the lab computers need to be updated with Rockwell's latest software. You're dreading it because you know the process takes a whole day or more for one computer and now you need to do it for 20! You've tried working with IT to get an image together, but finicky Rockwell doesn't work well with that, so you accept your fate. You sacrifice a weekend in the lab, installing software and knowing you'll never be compensated for the extra time spent.

The solution to both scenarios is simple. Install the software on one VM. Ask IT to drop that VM on all the lab computers. When you demo in class, you'll be using the same VM, so your screen looks the same as the students are seeing. And if a lab computer ever breaks (OK, we know students like to explore settings), you can just drop a new copy of the same VM on that computer. Problem solved! Now you can focus more on teaching, and less on troubleshooting.

Key Takeaways

- Install it once, works everywhere
- Easy recovery when something breaks

Smoothing relations between faculty and IT

From the perspective of faculty, IT has a few simple responsibilities: install the software, make sure the computers work, then stay out of my way. But IT's responsibilities actually go way past that, most notably into managing servers, networks, and security. Sometimes these two perspectives can be conflicting. Here are some common examples:

"IT should install the software."

IT would generally agree with this statement, but in the context of industrial software, it's not that simple. There can be so many pieces and options that it's not reasonable to expect IT to know which to install. The approach of "install everything" doesn't work here because there's just too much, and even if IT knew all of the important features and settings, normal IT approaches to installing software across multiple computers don't work well with industrial software. So who ends up installing it, and how?

"Make sure the computers work."

Again, IT would agree, but it's never that simple. Normally, if software isn't working correctly, loading a new image would fix it, but images aren't very reliable with industrial software, and IT can't be expected to troubleshoot the strange things that pop up in class. IT typically locks certain settings on lab computers to help keep students from accidentally breaking something, but that gets in the way of industrial programming coursework. The status quo seems to be designed to cause tensions between faculty and IT.

There's a simple solution.

A VM-based approach can reduce the effort of installing software, maintain security while allowing all coursework, and ease the sharing of responsibility in maintaining lab computers. A proactive approach can be rewarding, and time and cost saving for both faculty and IT.

Key Takeaways

- Simplified approach to software management across departments
- Be secure without hindering coursework

Creating a uniform and manageable student experience

Students work at different paces, and the ones that finish lab activities more quickly than others will frequently play with the software in front of them. This is a valuable addition to their experience, but can leave each lab computer in a much different state than the rest. After just one semester of labs, configuration can be different enough on these computers that the instructor has to work individually with a few students just to figure out how to do today's lab. This is time away from the real purpose of the class. Virtual machines offer an easy reset option that can be handled by either the instructor or IT to keep all lab computers functional and uniform.

Key Takeaways

- Give students a uniform starting point
- Allow the fast students to explore, while the slower ones catch up
- An easy reset option lets you focus on what matters

What does virtualization cost?

Virtualization, as described here, can be free in an educational setting. The biggest consideration is effort – effort to learn and effort to maintain. Learning is typically done in a very haphazard fashion, which can make it a painful process. Faculty using VMs may not fully understand what VMs are or how to use them, so they have trouble helping students in class and don't take advantage of time-saving features. This is frustrating for everyone involved. Where maintenance is concerned, a well-thought-out methodology will make this more efficient than the alternative, but again, it's frequently handled haphazardly. In both cases – learning and maintenance – proactively educating faculty and planning ahead can make the process much smoother.

Another part of cost consideration is lab computers. Generally, a computer that's powerful enough to run the automation software can also run virtualization software. However, if a lab computer is a little slow, this will be compounded with virtualization. Upgrading RAM and hard drive for about $300 per computer is usually

what's needed, and this is an easy process. (More information about hardware in the Implementation Plan section.)

Key Takeaways

- VMs in the classroom can be free
- Budget ~$300 per computer if unsure about hardware
- Time upfront training and planning will save time in the long run

Implementation plan

The first step is considering your lab's needs.

What classes use the lab computers, and what software do they need? Get all interested parties together - including faculty and IT - and list the required software. Then discuss who will be responsible for installing the software on the VM. It's good to elect a "champion" - usually a faculty or IT member - to be in charge of the VM and software installation. Then determine how the VM will be copied to the lab computers. IT will usually have a good solution for this.

Training faculty is also important.

They need know how to do all class activities on a VM and how to "reset" a VM if a student manages to break it. BMAS' online course can shorten the learning curve by walking you through step-by-step what a VM is and every aspect of using a VM for industrial programming.

Check it out here.

Let's talk about hardware.

If the lab already has computers, check to see if they meet the minimum requirements below. If they don't, they can usually be upgraded for about $300 - check with IT.

- RAM: 8GB min, 16GB recommended
- Hard drive: >50GB free space, SSD recommended
- CPU: strong enough to run your class software
- GPU: not usually important

Install all needed software on the VM.

It's a good idea to briefly test the software before distributing the VM, especially making sure software licensing is setup correctly. Lab computers will need virtualization software (like VMWare Workstation) installed, then distributing the VM is as simple as copy/paste.

Making things easy for the students enhances the learning experience.

Starting a course with a brief note one what VMs are and how they'll be used can go a long way. To help students get through common tasks, put together a short how-to for common tasks, like "starting up the VM" and "transferring files between the VM and host machine." Examples of these how-tos can be found in BMAS' online course, along with quick videos that can be used in class to introduce VMs and vocabulary.

Maintaining the system.

This is the last piece, usually only needed in two ways: 1) "Reset" a VM to its original state when it gets too different from the rest and at the end of each semester. This is an easy method to make sure students are seeing the same configuration you demonstrate. 2) Periodically build a new VM for the department. This may be every 5 years or so, depending on the department's needs. Software is usually very relevant even if it's 4-5 years old and less frequent upgrading means less work for the VM champion.

A future-proof approach

Software development will only continue to accelerate in the future, in stark contrast to the industry's general resistance to change. Virtualization will continue to be more prevalent for supporting older equipment and managing multiple versions of software, requiring programmers and technicians to be proficient with this technology. Schools can support this need and their own goals by using VMs in the classroom.

REFERENCES

Fruscione, Joseph. "When a college contracts 'adjunctivitis,' it's the students who lose." PBS. Public Broadcasting Service, 25 July 2014. Web. 09 Aug. 2017.

Moorestown Business Association. (n.d.). Retrieved August 16, 2017, from https://it-services.techstarters.com/server-virtualization

Schneider, Pat. "Leaving UW: Reduced funding, politics force fond faculty to say farewell." Madison.com. The Capital Times, 14 May 2016. Web. 09 Aug. 2017.

ABOUT BREEN MACHINE AUTOMATION SERVICES

Breen Machine Automation Services serves the American Manufacturing industry in controls engineering, consulting, and training.

Learn more about our services at www.Breen-Machine.com, or our training programs at courses.Breen-Machine.com