Pay Now or Pay Forever:
Commissioning the Design of Control System Software

Greg Cmar, Bill Gnerre, and Kevin Fuller
Interval Data Systems, Inc.

Synopsis

The importance of starting commissioning at a project's design phase cannot be overstated. To deliver an HVAC system that will run well for the next 20 years, the design must be right to begin with, or the building will never operate properly.

The engineering community has typically done an excellent job advancing the capabilities of the HVAC system. However, it has been a common practice to specify control intent and then to allow the controls contractor to develop the final design documents. All too often, these documents are limited to a set of wiring diagrams and a book full of manufacturer's generic documentation for the devices installed on the job.

A critical, yet often underspecified aspect of the design is the integration software. This paper recognizes that design intent documents must specify, in detail, the sequence of operations at the integration level in order to ensure the building will operate to the desired level of occupant comfort and operating cost.

Commissioning agents engaged at the design phase must be able to recognize and resolve deficient integration software specifications, and then commission the resulting implementation. While this adds to the upfront costs for building owners, the alternative is higher utility and maintenance costs for the life of the building.

This paper uses case study data from a Midwestern university to show the types of operational problems that exist because of improper or insufficient design intent specifications. It defines a set of deliverables ranging from control software logic specifications to the inclusion of an operator's instruction manual specific to the design of the project's HVAC systems. It concludes with examples showing re-specified control sequences and the operational commissioning process.

About the Authors

Bill Gnerre brings twenty-plus years of technology entrepreneurial experience to his role as cofounder and CEO of Interval Data Systems. His previous positions include being a vice president at Circadian Software, cofounder of ChannelWave Software, and various management responsibilities at Formtech and Computervision, vendors of engineering document management and CAD systems. Earlier in his career, Bill held a variety of engineering positions and has a degree in mechanical engineering from Northeastern University.

Greg Cmar, cofounder and CTO of Interval Data Systems, is an expert on energy systems and interval data. He has 30 years of experience in facility operations, energy conservation, energy analytics, monitoring and control systems, and utility billing, as well as database, application and automation technologies. Prior to IDS, Greg was a cofounder and director of engineering at ForPower, an energy conservation consulting firm; and held senior roles at Coneco, Enertech Systems, Johnson Controls, the Massachusetts Energy Office, and Honeywell. Greg holds patent #5,566,084 for the process for identifying patterns of electric energy, effects of proposed changes, and implementing such changes in the facility to conserve energy.

Kevin Fuller, executive vice president, is responsible for marketing and product development for Interval Data Systems. Before joining IDS, he spent 15 years working for vendors of relational database, analytics, and enterprise applications in both marketing and technical roles where he developed a strong understanding of how businesses used data to their advantage. He started his career in programming and systems management roles after graduating with a degree in mathematics from the University of New Hampshire.

Pay Forever

The vast majority of buildings, new or old, are never formally commissioned today. The construction industry works on a "get in, get out, get paid" approach to delivering buildings to the owners. The HVAC and other building systems may work well enough for acceptance and occupancy, but typically don't work well from the viewpoint of the operations and maintenance staff that has to live with the building for the next couple of decades, dealing with high utility and maintenance labor costs. This "pay forever" approach dominates our industry today.

But hey, this paper is for a commissioning conference, and if you're reading it, you're probably attached to the commissioning industry. So you already know all that. The problem is, an unacceptable percentage of buildings that *are* formally commissioned (or retro-commissioned) suffer the same "pay forever" fate. The reason for it all starts with the control system's design.

Good Intent, Bad Design

Let's take a moment to introduce the case study presented in this paper. The site is a Midwestern university with about six million square feet of facilities. The building focused on here is a 110,000 square foot business school building constructed in 2003. Its HVAC systems include three air handlers and 218 VAV boxes with reheat, some of which are fan powered, plus a few fan coil units, exhaust fans, pumps, etc. The campus central plant provides steam and chilled water. The control system exposes approximately 8,150 points, all of which are collected on a 15-minute interval basis and permanently saved in a database.

Upon a thorough analysis of the building operations, the analysts identified a few dozen issues. The individual problems found weren't the most interesting part—we've all seen economizer issues, valves that hunt, and other examples of poor operations. What was interesting were the "bigger picture" issues and identifying the root cause of each problem. We're not talking about the root mechanical or control issues, but back to where the process went astray. In essence, we asked ourselves, "How did we get here?"

Fortunately, there were answers. By interviewing the facilities staff and reviewing the original design documents (and changes), the team was able to trace the building problems back to the original source of what went wrong. This is where it gets interesting.

The operations staff often is blamed for "screwing up the building." However, this analysis showed that over 80% of the issues identified existed the day the school took occupancy of the building—traced to design intent or controls programming implementation errors and omissions. In contrast, operator errors within the building caused less than 20% of the issues identified.

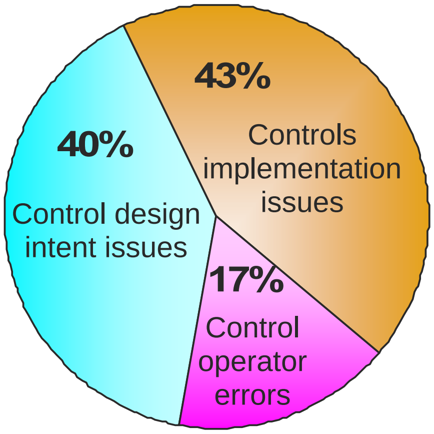


Figure A: Breakdown of root causes of building operational issues.

Designs intend to be good, so why was the design intent so bad?

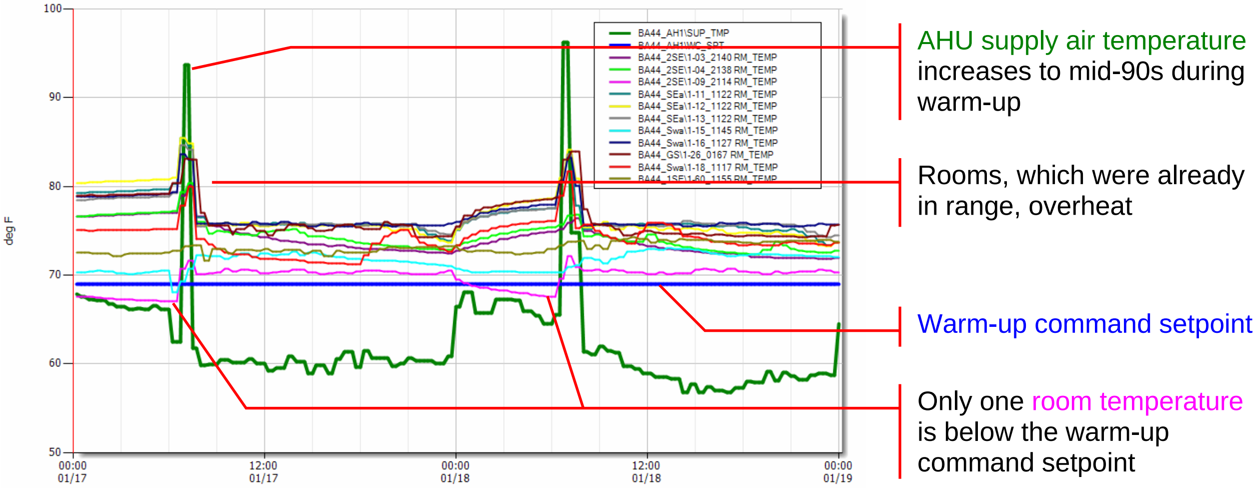
Software ≠ Hardware

Let’s refine what part of the design had issues. The physical/mechanical design was fine, as were the air handler drawings, wiring diagrams, etc. The points list, while underwhelming (many desirable points were omitted), was adequate. The breakdown was in the sequence of operations—a.k.a. the software layer of the control system. More specifically, most issues were at the integration level of the controls programs.

The current controllers from every vendor are highly advanced and capable of sophisticated control strategies. However, the software tools to program them make it difficult (sometimes virtually impossible) to achieve what the hardware is capable of doing. Design engineers do a good job at designing the physical aspects of an HVAC system. But their understanding of control systems, especially DDC systems, is all too often lacking. Add to that the time pressure that always exists, and a copy/paste approach to deliverables, and you have a recipe for a sequence of operations that is littered with vagueness and incomplete instructions.

No Hable Integration

Let’s return to our case study for an example. Figure B shows the air handling system attempting to perform a warm-up command. There is one room operating below the warm-up command setpoint, which triggers the control. What we see is that the supply air temperature is raised and most of the room follow suit. However, the remainder of the rooms don’t need warm-up at all, and many get too warm, getting well above 80°F.



AHU supply air temperature increases to mid-90s during warm-up

Rooms, which were already in range, overheat

Warm-up command setpoint

Only one room temperature is below the warm-up command setpoint

Figure B: warm-up control gone wrong

Why did this happen? It’s easy to diagnose that the VAV boxes remained in cooling mode while the warm-up command took place. As a result, they were trying to cool with hot air—opening their dampers further as the room got hotter. The temperature climbed until the warm-up ended. But why did that happen? Where did this process go wrong?

Let’s look at the sequence of operations for the warm-up command (exactly as written):

MORNING WARMUP CONTROL

33) Morning warmup sequence will be performed based on the lowest room temperature served by this air handler. The AHU will be commanded into “morning warmup” up to an hour (adjustable) before the scheduled occupied time is to occur. The amount of time the unit will be in “morning warmup” will be based on how far the lowest zone temperature is away from 68 degf. That amount of time will be based on the following table.

64 zone temperature (adjustable in GPL)	60 minutes (adjustable in GPL)
68 zone temperature (adjustable in GPL)	0 minutes (adjustable in GPL)

34) The following will occur when unit is commanded into “morning Warmup”

The Differential CFM between the supply and return fan will be adjusted to 0 CFM.
The mixed air dampers will go to 0% ODA
The supply air setpoint will be adjusted to 95 degf (adjustable)
Morning warmup will end if the unit is commanded into the occupied mode or if the zone temperature reaches its target setpoint (adjustable).

Figure C: Sequence of operations for morning warm-up command

The air handler functioned as specified. It took the lowest room temperature, which was below 68°F, and ran a warm-up cycle starting one hour before scheduled occupancy. Although not shown in Figure B, the system adjusted differential CFM and the dampers adjusted to provide 0% outside air. Supply air peaked at about 94°F, and the warm-up ended on schedule.

What about the VAV boxes? Below is the entire documented sequence of operations (again, exactly as written) for the 218 VAV boxes in this building:

VARIABLE AIR VOLUME BOX CONTROL SEQUENCE

System Occupied/Unoccupied Command:

The VMA will operate in the occupied mode whenever the contact from the motion detector for the room lighting is closed or the occupied pushbutton on the room sensor has been pushed.

The VMA will operate in the unoccupied mode whenever the contact from the motion sensor for the room lighting is open and after a 30 minute delay (adj.) after the occupied pushbutton on the room sensor was last pushed.

Occupied Mode Control:

During the Occupied Mode, the VMA Controller will attempt to maintain the space temperature at thermostat xxx_T at its (adjustable) Occupied Cooling Setpoint of 74 Deg F. Should the space temperature at rise above its setpoint, the VMA Controller will drive the box damper open to feed cool conditioned supply air to the space. A sustained space temperature above the Occupied Cooling Setpoint will cause the VMA Controller to drive the box damper to its Cooling Maximum Flow setting. Should the space temperature at fall below the setpoint, the VMA Controller will drive the box damper further closed to feed less cool conditioned supply air to the space. A sustained space temperature below its Occupied Cooling Setpoint will cause the VMA Controller to drive the box damper to its Occupied Cooling Minimum Flow setting. Upon further drop in room temperature, the VMA Controller will open the box damper open to its Occupied Heating Maximum Flow setting while sequencing the reheat coil valve open. Whenever heat is not available at the VAV reheat coils, the VMA will limit the box dampers from opening to Heating Maximum Flow.

(continued)

> **FAN POWERED BOXES: IF REQUIRED**
>
> **Series Fans:**
>
> The Series Fan is off during the Shutdown and Auto Zero modes. The fan is always on during Occupied and Standby modes and is cycled on during the Unoccupied mode whenever heating is on. Before the fan is turned on, the damper is driven closed for the Auto Zero duration time to ensure that the fan is not spinning backward. During warmup the Series Fan is allowed to cycle on. The on/off series is controlled by a single binary output with min on/off timers that can be set in the BO Modify screen.
>
> **Parallel Fans:**
>
> The fan is off during the shutdown mode and Auto Zero modes. The Parallel/temp Fan is cycled on when Warmup is inactive and the heating command is greater than the value and the value of the fan Start Setpoint parameter defaulted to 1%. The Parallel/Flow fan is also cycled on during occupied and Standby mode whenever the flow setpoint is below the “parallel Fan/Flow parameter value. The supply deadband is used as a differential to turn the fan off.

Figure D: Sequence of operations for VAV box

What exactly was the VAV box supposed to do during warm-up? For fan-powered boxes there is some instruction, but that’s a small portion of the number of VAV boxes. For the rest, there’s no indication at all—there aren’t any instructions for VAV operations during unoccupied mode for that matter. No one thought about the integration between the AHU and VAV controllers.

The university wound up with a warm-up command that worked exactly as specified… and entirely wrong. (OK, not entirely wrong. It was mostly that pesky detail of the VAV boxes needing to know about the warm-up cycle and be commanded into warm-up mode.) As a result, the university was actively enrolled in the “pay forever” plan. It experienced daily energy waste during heating season by providing a lot of unnecessary heat and then needing to immediately correct that mistake with cooling. They also paid through hot and cold calls—hot because someone just walked into an 85°F room, and cold because someone else is now sitting under the room supply vent and getting 55 – 60°F air dumped on them as the system cools the room back to the space temperature setpoint.

A Pandemic

While the fix to our warm-up command example is straightforward, it was representative of a systemic problem. The software layer of the control system, particularly the integration logic, was lacking throughout the building. And it’s not just this building; the university had carried this design forward into other buildings during the past four years.

In fact, it is a disease of pandemic proportions. While the individual symptoms vary, every single building our analysts have reviewed has significant operational problems caused by poor integration control programming. It hasn’t mattered if the building was commissioned or not, as some buildings had been formally commissioned shortly before the analysis.

Pay Now

Curing the “pay forever” situation is a difficult issue. It’s not as simple as, “Take two ASHRAE standards and call me in the morning.” Conceptually, it is easily corrected—get it right from the beginning. However, there is an uphill battle with the stakeholders to allocate time and funds to do a better job with operational design and implementation. Despite commissioning findings, programs like LEED, or even just common sense that it’s better to spend a little more to get the design right from the start than to pay multiple times the cost every year for the life of the building, the industry still puts lowest first cost ahead of more rational decisions.

Why is it so hard to get building owners to understand the value of the “pay now” approach? There is the lowest first-cost bid issue, but frankly, that answer is a cop-out. If building owners understood how poorly their buildings actually ran, or how poor the control systems operational designs were, they’d refuse to accept them. Several months ago our university understood that the software specification was missing, and operations were suffering as a result. And to paraphrase Paddy Chayefsky, they were mad as hell and were not going to take this anymore.

The path to fixing the building was to fix the software, and the prerequisite to fixing the software was to create a software specification. Since this was an existing building, the hardware was already in place. It was “just” a matter of creating the detailed software specification.

Recognize the Issue

Before describing the university’s new software specification, let’s bring this back to the role of commissioning. Many industry people (facilities staff, engineers, even commissioning agents) view commissioning as ensuring that a system works “as designed.” The warm-up example given earlier does indeed work as designed. However, it does not approach a level of engineering common sense—clearly, the software needs fixing. Better commissioning professionals believe that buildings should run well, whether designed that way or not, and the commissioning effort should start with the design.

The challenge for commissioning agents is to recognize bad operational design when they see it. The most common example that illustrates the lack of integration specification is open loop controls. Older buildings, with pneumatic systems, were built with open loop controls. The hardware implementation of the control system included the aspects of software. What one sees in the field today is the same approach implemented with DDC technology. That no longer works adequately because hardware configuration and software implementation are two separate things. If you’re looking for good performance out of the building in terms of comfort and energy efficiency, buildings require closed loop system implementations that provide the feedback needed to run well. For example, if the sequence of operations for an air handler has no references to the VAV boxes (or other terminal units) and vice-versa, there’s a problem with the software specification. Commissioning agents dealing with design should ask for the software specification along with other design documents.

Software commissioning is not something the industry is automatically qualified to do. Engineers and commissioning professionals are excellent at commissioning the hardware and construction aspects of a building as well as the engineering aspects of the design. But software engineering is a separate discipline that one should not take for granted. Because computers are so pervasive there is a tendency to assume everyone knows how to deal with the software part of control systems. That is no truer than assuming that, because everyone has home air conditioning and heating, programmers all understand the engineering behind commercial HVAC systems. The commissioning, design engineering, and controls contracting industries must develop the necessary software design and programming skills to make today's control system work well.

An Operational Design (Software) Specification

Returning to our case study, the university opted to reprogram the control system based on a well engineered, detailed, and well documented *operational* design.

> "Design is not just what it looks like and feels like. Design is how it works."
> – Steve Jobs

Now, Steve Jobs may know nothing about HVAC systems, but he knows a lot about design and a lot about software, and what we're talking about is a software design problem. Current designs do well at defining hardware and mechanical specifications, but not at "how it works," as evidenced by the operations of most buildings today.

The university's new specification needed to deliver a fresh approach to communicating the building's operations, and ensure predictability and consistency of the resulting implementation. The design intent is to enable the university to achieve three simple goals:

1. Meet comfort and IAQ requirements in every individual occupied space,
2. Do so at the minimum possible operating cost, and
3. "We don't want to have to dink with the system."

This was no small task. The team creating the new specification included professionals from engineering, software, and communications disciplines. Sure, there are points lists and sequences of operations (with extensive detail), but it also tackled topics never included in standard specifications, such as a description of the operational philosophy, how the university defines and measures comfort and IAQ, and an extensive set of acceptance criteria. The university invested a large sum in a control system to meet its comfort and air quality obligations, and to do so efficiently in terms of both energy and labor. The specification was designed to assure that they actually received what they paid for.

Anatomy of an Operational Specification

Again, the key word here is *operational*. With the exception of adding some sensors, there were no hardware or configuration changes in the physical system. However, it represented a complete redefinition of the control system software.

The specification also represented a complete redefinition in communications. Existing control specifications tend to be written by engineers, for engineers, and simply do not communicate very well what is required for today's complexity of operations. The intended audience for the new specification was much larger than the controls contractor and university control engineers. It informed design engineers, controls engineers, building automation system vendors, mechanical contractors, field mechanics, and other suppliers/contractors of the university's requirements for operations and performance.

The specification has six main sections:

- Philosophy
- Comfort/IAQ
- Hardware
- Points
- Control Strategies
- Acceptance

Philosophy

This Philosophy section provides an overview of how the control system will function in language that both engineers and, as importantly, non-engineers can easily understand. It describes the approach to each aspect of control system operations, but not the detailed sequence of operations—those are provided in the *Control Strategies* section. The Philosophy section provides a framework for the rest of the document, and an organization that other sections follow.

Comfort/IAQ

Comfort expectations are virtually absent from most specifications aside from setpoint specifications or trite statements such as, "The system shall provide a comfortable work/learning environment." The university's new specification defines comfort and indoor air quality requirements in detail. The position at the start of the document reinforces its importance.

The specification provides thermal comfort (temperature and humidity) and ventilation requirements based on the ASHRAE 55-2004 and 62.1-2004 standards respectively. Design intent is not accepted as a proxy for anything—conformance is measured and verified by the university, and therefore the method of measurement is detailed as well.

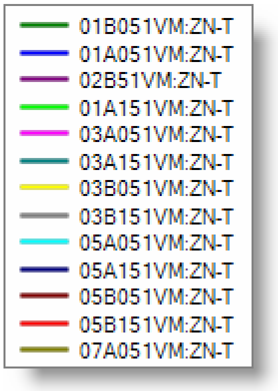
Hardware

The hardware section primarily addresses sensors. As the saying goes, garbage in, garbage out. Every commissioning professional knows the impact of sensors that are out of calibration or poorly located and not providing accurate readings. This section of the specification details the selection, location, and calibration of sensors throughout the building, and explains why they are needed and the value associated with having them.

Points

Before addressing the points list, the specification defines a series of naming standards designed to provide users with an understandable, maintainable system. Point name abbreviations are standardized, so, for example, instead of DAT, SA-T, and SUPTEMP all showing up in different places depending on who set up the system, users can always expect the air handler’s supply air temperature point to be SUP_TMP.

At another level, point names must provide sufficiently qualified name. For example, at another location a building was set up with point names such that the user could not determine where in the building the point referred. Each of 16 separate areas of the building had VAV boxes numbered starting with 51. The resulting point names for the box zone temperatures were 01A051VM:ZN-T, 02B51VM:ZN-T, 01A151VM:ZN-T, etc., all ending with 51VM:ZN-T. The prefixes to the names referenced the controller hardware, forcing the user to know what spaces each box number serviced. 02B and 01A1 are both actually on the second floor while 01A0 is on the first floor. The university’s specification eliminates the possibility of such nonsense.


01B051VM:ZN-T
01A051VM:ZN-T
02B51VM:ZN-T
01A151VM:ZN-T
03A051VM:ZN-T
03A151VM:ZN-T
03B051VM:ZN-T
03B151VM:ZN-T
05A051VM:ZN-T
05A151VM:ZN-T
05B051VM:ZN-T
05B151VM:ZN-T
07A051VM:ZN-T

The new specification goes well beyond just defining point lists with names, point type, display values, and alarm settings. It defines which points are mapped to the control system user interface, which are trended within the control system and at what time interval, and which are collected into an external historical database and at what time interval. This forces the issues of making vital information available to the building owner.

Finally, the points section of the specification explains why all of this matters. It describes the functions of the specified points, what control strategies they participate in, and what calculations they feed. It is not data for data’s sake, but information with business value.

Control Strategies

You can make the analogy that all the portions of the specification prior to this are like creating the foundation and structural frame of a building, and this is where you finish it and make it livable. And, if you tell the finish carpenter that you want hardwood floors, and they install an oak floor when you really wanted cherry, that’s not the carpenter’s fault. You got what you asked for, even if the result was unpredictable.

PECI produced an excellent online document, the *Control System Design Guide* (see citations). It covers the mechanical and hardware portions of designs as well as the operational, in fact making some of the same points as this paper. In particular, section 2.4.1, *System Diagram and Sequence of Operations*, hits the importance of the integration layer of control systems. It begins:

> “Successful HVAC designs hinge on the smooth, integrated interaction between the system’s components and the loads served. It is not just an air handling unit; it is an air handling system made up of an air handling unit, intake system, distribution system, terminal equipment, return system, relief, and exhaust system. It is critical that the control system design reflect this systems-based perspective. It is this systems perspective that guides creation of the system diagram and the detailed sequence of operation.”

That systems perspective and what it means to specify integrated interaction is conceptually understood by many, but implemented by very few. But here’s where it seems to come up short:

> "Often, the difficult part of this transition to the system concept is learning to write out the detailed operating narrative. This detailed narrative is just a written statement of what the designer should already know: the details of how they expect the system to function."

The devil is quite literally in "the details." The software specification must provide sufficient detail to remove the inconsistency and unpredictability from the result. The design engineer, controls contractor, or the design's commissioning agent must properly define the integration software.

In the software industry, a project goes through multiple levels of design before anyone writes code. Early on in the process is a functional description. As the name implies, it describes what functions the program should have—features, user interface, etc. This maps to the phrase used above, "how they expect the system to function." In the software world, if you hand a functional description to a typical programmer, there's no telling what you'll get back. It will be their interpretation of what was asked for. Funny thing, that's what happens in the building controls world too.

There's a step that the software industry embraced decades ago. Software architects/designers translate from the functional description into a software specification. The goal is to ring out all the details and get to a point where, if given to several different programmers, you will get back the same result—not necessarily the same code, but the same result.

Now let's apply this to building controls. If the hardware is still out for bid, then the design engineer can only specify a functional description of the system. Detailed software specifications depend on the hardware involved. The winning controls contractor should develop and submit the software specification so that the implementation can be commissioned against the functional requirements as well as the actual building operation. When the hardware is known, as in the case of an existing building where the hardware is not changing, the options for who develops the software specification get a little broader, but the requirements for a detailed accounting of how the software works are the same.

This is the level of detail put into the university's specification. Because the hardware was not changing, it combines the function description and software specification—the Philosophy, Comfort/IAQ, and Acceptance sections serves up the functional requirements and the Points and Controls Strategies sections provides the specification-level detail.

What's so different about it? Figure E shows the complexity of VAV box integration. (This is for a JCI system with VMA1400-series controllers; other hardware will result in different logic to meet the same requirements.) It is a high-level view showing the major areas of VAV control (inside the red box) and the many integration points where external data or programs affect or interact with the VAV box.

Down the left side there are a variety of data elements (purple) that are inputs at different stages of the VAV control program's operation. For example, the room schedule is an input to occupancy mode determination, and the outdoor air fraction is an input to ventilation control.

At the upper right (yellow) are air handler control strategies, each of which must interact with the VAV boxes to operate properly. Following through on our earlier example, the warm-up routine must integrate at the occupancy level to know when to start and stop operations before setting the box mode to warm-up.

The VAV operation has inputs from local sensors about the room or the box itself (green hexagons). The main flow down through the red box shows the various components of the VAV control program itself.

Finally, the outputs on the lower right (blue) are data points from the VAV controller (e.g., starved box) or the results of its operation (e.g., room temperature). These become inputs to the air handler control, playing their part in creating a closed-loop design.

While this helps explain how the system is designed to function to the controls programmer, and does identify the various points of integration that must be included, it is far from the level of detail necessary for a predictable, repeatable result. The specification takes each area and provides that additional level of detail.

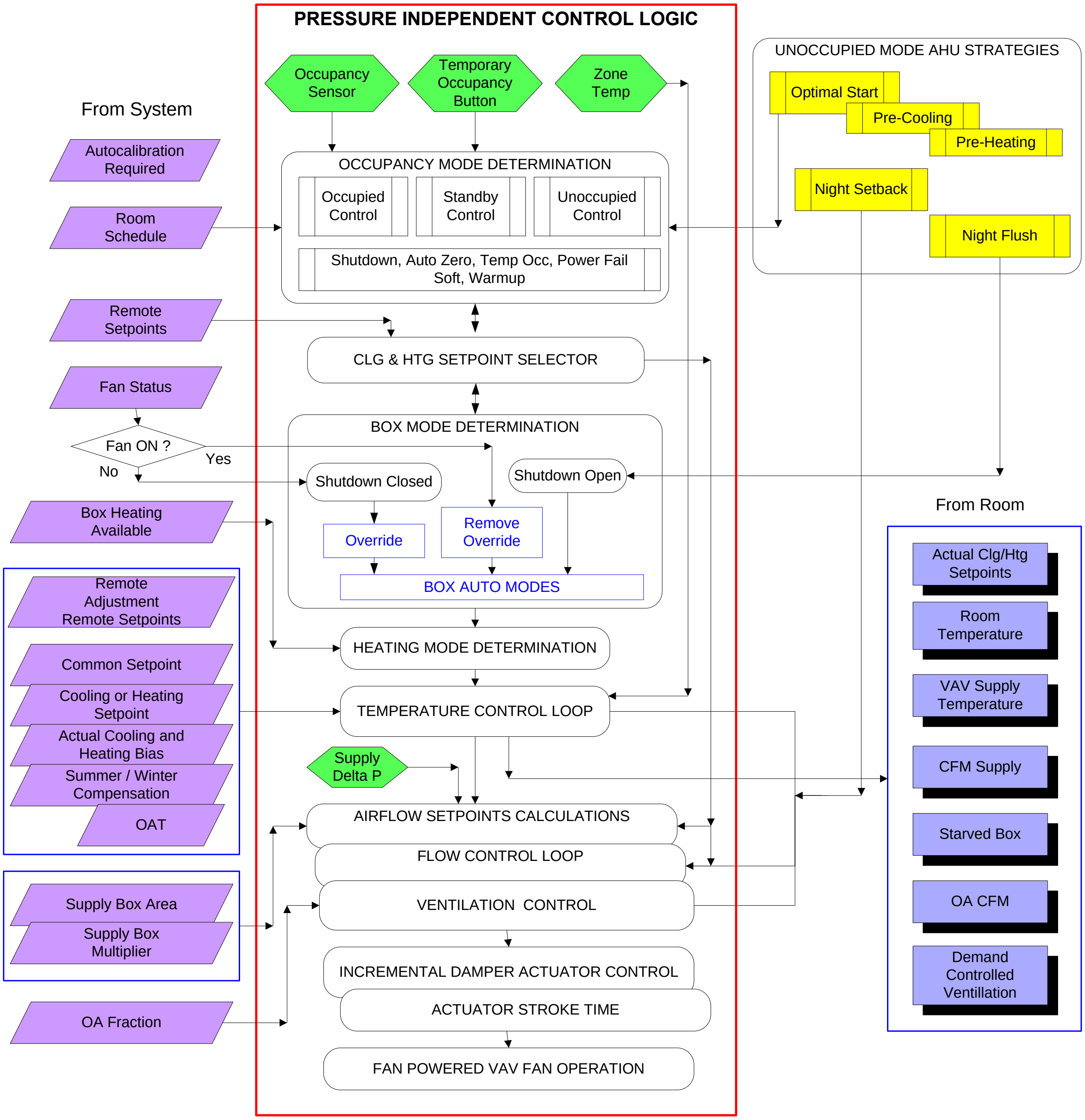


Figure E: VAV box interfaces and integration points

The specification includes written descriptions that accompany each logic diagram to explain fully how the control needs to work and how to program the controller. To show one more example that goes a step deeper into the detail, Figure F is the logic diagram for determining the optimal start time for the warm-up command (or pre-cooling in the summer).

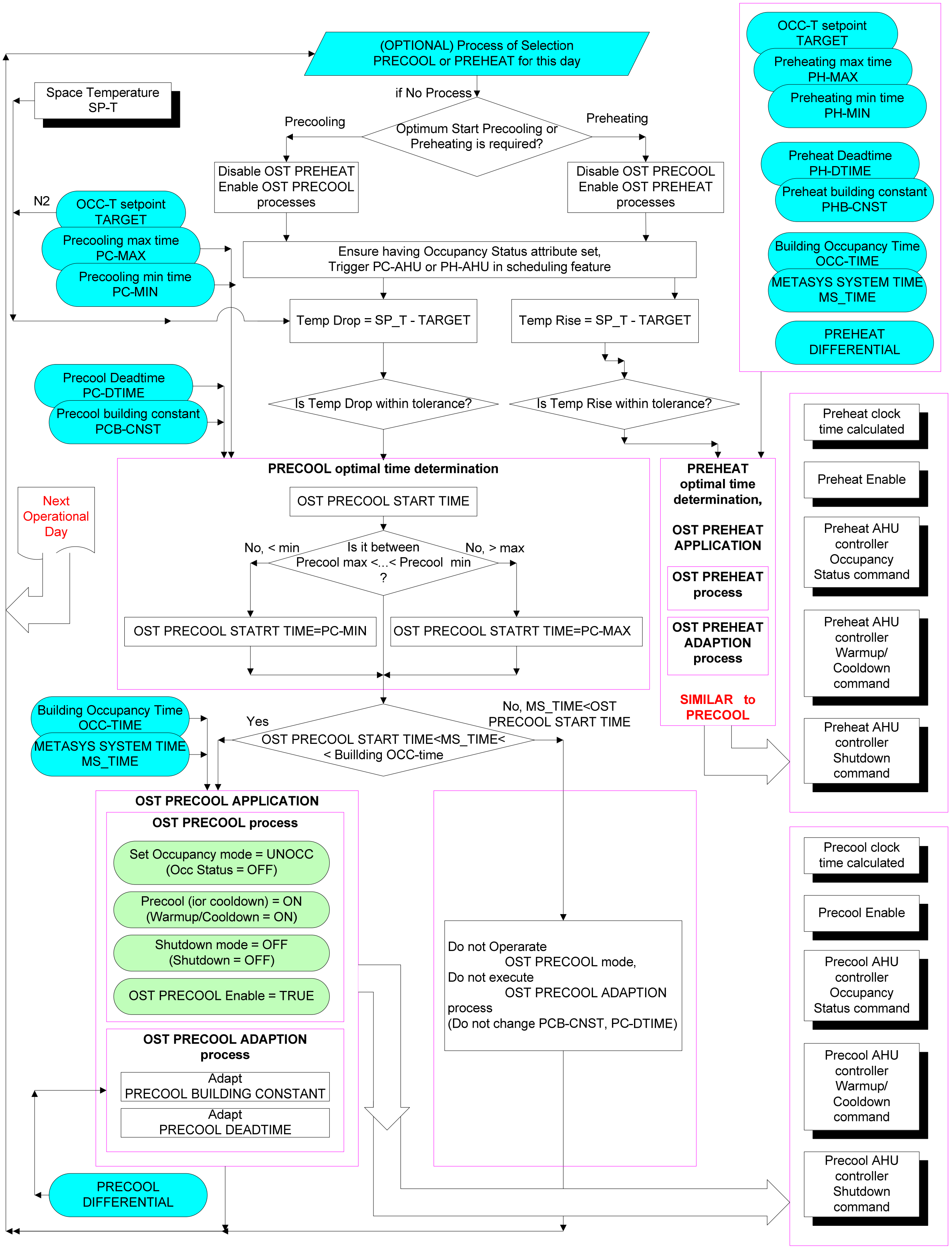


Figure F: Optimal start time logic for pre-cool and pre-heat controls

Acceptance

The last section of the university's specification is the acceptance criteria. Although intended to validate that the controls contractor installed and set up the system properly, it is also a guide for operational commissioning of the building. Again, there is a parallel to commercial software development, which always includes a QA (quality assurance) plan as part of the overall software product design.

The specification defines acceptance criteria at three levels, comfort, component operations, and integration operations. Comfort checks validate that each individual room in the building meets the thermal comfort and ventilation requirements. Component operations checks each piece of equipment (VAV box, fan coil, cooling coil, etc.) to assure it is running properly, which can have a significant effect on cost. And finally, the integration-level tests show that the entire system works as it should.

Figure G shows a sample of the acceptance test for thermal comfort. This check is performed for every room/zone in the building—sampling is not accepted. Each test provides a reference back to where in the specification the proper operation is defined, lists the points used for testing, recaps what is acceptable, and shows a sample test result.

The charts in Figure G show a room that does not pass the criteria. Although it is maintaining thermal comfort quite well, the test shows that the VAV box remains in occupied mode 100% of the time. It has clearly not been set up properly. (Note: The comfort index is a calculation done within the university's EEMS that assigns a comfort measurement based on the ASHRAE 55-2004 standard.)

8.1.1.1 Thermal Comfort

Thermal comfort is defined in section **Error! Reference source not found.**. For each space where temperature control exists, the university verifies operations using the following data points:

- Comfort Index (CI calculation for space)
- Occupancy status (OCC_STS)
- Zone/room temperature (RM_TMP or ZN_TMP)
- Zone/room temperature setpoints (ACLG_SPT and AHTG_SPT)
- Relative humidity (RET_RH (value is from AHU return unless otherwise specified))

The space should maintain a Comfort Index value of eight (8) or higher for all occupant spaces, and seven (7) or higher for all common areas. Room temperature must be maintained within the deadband defined by setpoints. Humidity must be maintained in the 0% – 60% range.

(continued)

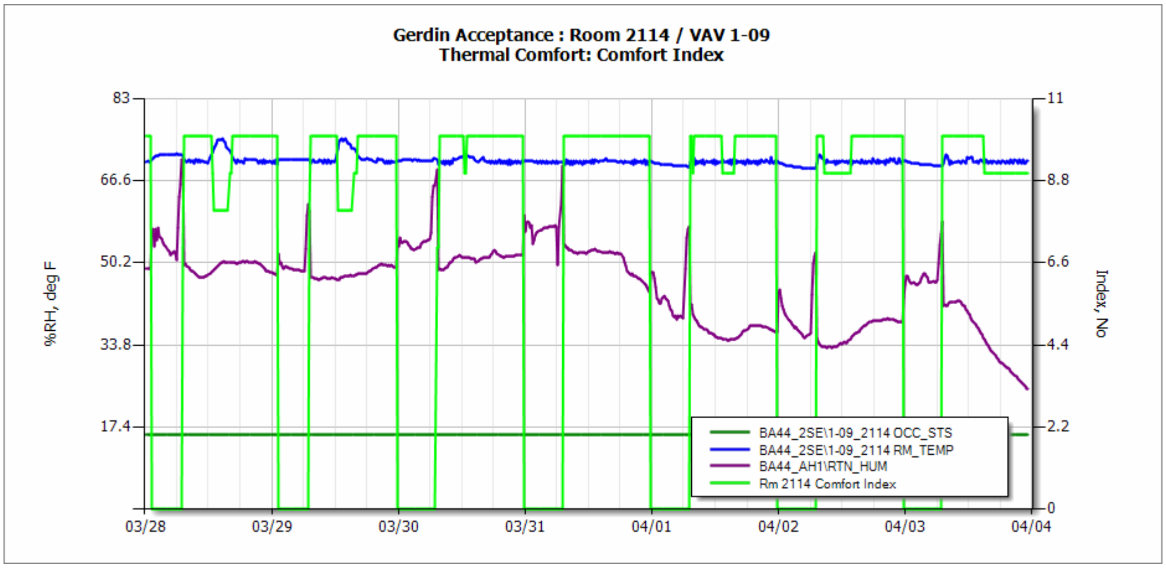


Figure 13: Comfort Index acceptance report showing Comfort Index with related space temperature, humidity, & occupancy status.

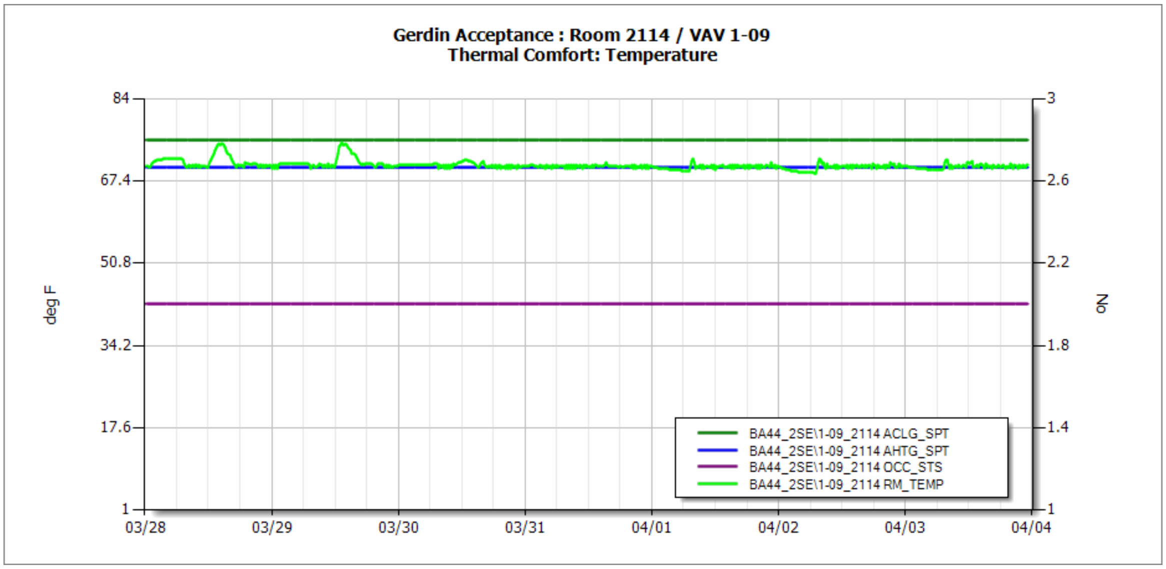


Figure 14: Temperature acceptance report showing space temperature, deadband, & occupancy status.

Figure G: Example of acceptance criteria showing thermal comfort tests

Operations Manual

Despite only accounting for 17% of the issues found during the root cause analysis, operational errors do cause building performance to degrade over time. One cause for that is that the operators are rarely trained on how the system works. They know how equipment works, and have approaches to respond to occupant complaints. But they do not often have the background to realize the side effects of some of their actions.

The last part of this puzzle, separate from the design specification, is an operator's manual. At the time this paper was written, the university had not yet created their operators' manual. However, the plan is to start with the Philosophy section of the design specification so that maintenance staff will understand the bigger picture of how the system runs as well as the basics of each control strategy. It will then go on to instruct readers about the proper actions to take in response to typical scenarios, as well as caution the reader about actions that can cause problems or have unwanted side effects.

Conclusions

Buildings do not run well. Frankly, it's the reason there is a commissioning industry to begin with. Problems nearly always trace back to inadequate control systems programming, typically the result of inadequate software design.

The industry isn't changing, it has changed. Software is as important a component of building controls as is the hardware. It is also separate from the hardware and requires its own specification. The software specification goes well beyond the standard sequence of operations provided as part of system designs today. Along with control drawings, wiring diagrams, and points lists, if the software specification isn't included, the design is not complete and not ready for implementation.

This creates an opportunity for commissioning professionals to commission the design before implementation. Assure that the software component exists and that it will result in the intended operation. Use the vast array of experience to show owners that catching problems at this phase will save them many thousands of dollars in ongoing operational costs. Far better to pay now than to pay forever.

There is also a challenge for commissioning professionals (and design engineers/controls contractors) to learn to deal with control systems software, and its specification. It used to be sufficient to know how to do a bit of programming, but now controls software specifications require real software engineering expertise, which is a major learning curve for many.

Those willing to make the investment have the chance to lead the industry in the years to come as the software, integration, and design issues continue to get more complex, and building owners continue to suffer with mediocrity. It's a chance to make DDC systems finally deliver on their potential, and make well performing buildings as commonplace as they should be.

Citations

Interval Data Systems, Inc. and Iowa State University; *Control System Operational Design Specification, Gerdin Business Building;* draft version 0.9; 2007.

Johnson Controls, Inc.; *Variable Air Volume Room Control with Reheat – Flow Diagram and Module* and *Mixed Air Ventilation System, AH1, College of Business Building, Building Automation (Johnson Controls Metasys) Sequence of Operation;* 2003.

Portland Energy Conservation, Inc.; *Control System Design Guide*, version 1.0; www.peci.org/ftguide/csdg/CSDG.htm.