Pay Now or Pay Forever: The Design of Control System Software

By Greg Cmar, Bill Gnerre, and Kevin Fuller

The Gerdin Business Building features wireless access and high-tech laboratories so Iowa State's business students and faculty can replicate real-world situations.

PAY FOREVER

The vast majority of buildings, new or old, simply do not operate well. The construction industry works on a "get in, get out, get paid" approach to delivering buildings. 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. This "pay forever" approach dominates our industry today, even when buildings are formally commissioned (or retro-commissioned). The reason for it all starts with the control system's design.

GOOD INTENT, BAD DESIGN

The building highlighted is the Gerdin Business Building at Iowa State University, an 113,000 square-foot building constructed in 2003. Its HVAC systems include three air handlers and 218 VAV boxes with reheat-some of which are fan-poweredplus fan coil units, exhaust fans, pumps, etc.

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. 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.

Through discussions with the facilities staff, and while reviewing the original design documents (and changes), the team was able to trace the building problems back to the original source.

The operations staff is often blamed for "screwing up the building." However, this analysis showed that over 80 percent 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.

SOFTWARE ≠ HARDWARE

Let's refine what part of the design had issues. The mechanical design was fine. 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 available from vendors 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' understanding of control systems, especially DDC systems, is often lacking. Add to that 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 look at an example. Figure A 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 rises and most of the rooms 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.

Why did this happen? The VAV boxes remained in cooling mode while the warm-up command took place. They were trying to cool with hot air—opening their dampers further as the room got hotter. Where did this process go wrong? While the AHU sequence defined the warm-up cycle, there were no instructions for VAV operations during warm-up; no one considered the integration between the two.

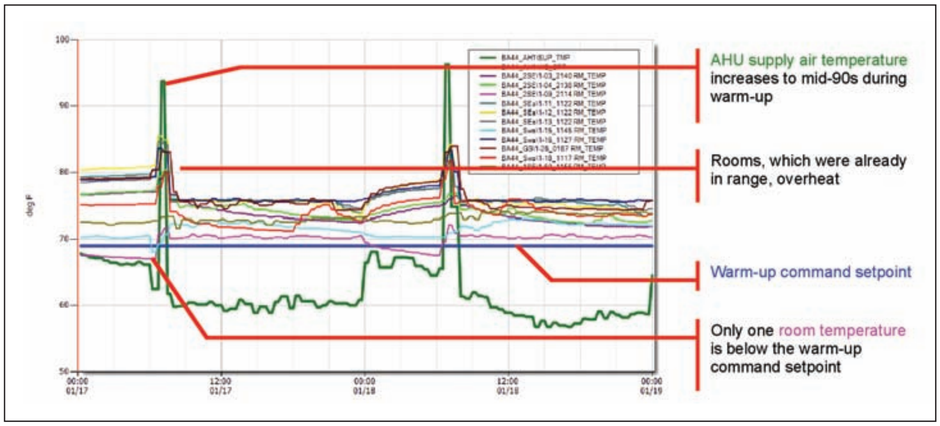

Figure A: Warm-up control gone wrong.

Therefore, the Gerdin building wound up with a warm-up command that worked exactly as specified…and entirely wrong. (OK, not entirely wrong, mostly that pesky detail of the VAV boxes needing to know about the warm-up cycle.) Iowa State was enrolled in the "pay forever" plan, experiencing 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.

While the fix to this warm-up command example is straightforward, it represents a systemic problem. The software layer of control systems, particularly the integration logic, is lacking. It's not just this building, or this school. Every building (even those recently commissioned) our analysts have reviewed has significant operational problems caused by poor integration control programming.

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." The path to fixing the building was to fix the software, and the prerequisite to fixing the software was to create a software specification.

RECOGNIZE THE ISSUE

Software commissioning is not something the industry is automatically qualified to do. Engineers and commissioning professionals commission the hardware and construction aspects of a building. But software engineering is a separate discipline.

AN OPERATIONAL DESIGN (SOFTWARE) SPECIFICATION

Returning to our case study, Iowa State 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 (co-founder of Apple, Inc.)*

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

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

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 the new specification also tackled topics never included in standard specs, such as a description of the operational philosophy; how the university defines and measures comfort and IAQ; and an extensive set of acceptance criteria.

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, the new specification represented a complete redefinition of the control system software.

Existing control specifications tend to be written by engineers, for engineers. The new specification 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 following highlights some of the main sections:

Philosophy: This section provides an overview of how the control system must function in language that both engineers and non-engineers can easily understand. It describes the approach to each aspect of control system operations, but not the detailed sequence of operations.

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 new specification defines comfort and indoor air quality requirements based on the ASHRAE 55-2004 and 62.1-2004 standards. Design intent is not accepted as a proxy for anything—the university measures and verifies conformance, and therefore the method of measurement is detailed as well.

Points: The specification defines a series of naming standards designed to provide users with an understandable, maintainable system. It then defines which points are mapped to the control system user interface, which are trended, and which are collected into an external historical database as a record for the building owner.

Control Strategies: This is the meat of the specification with the most important aspect being to make the result predictable. 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.

Acceptance: Finally, there are acceptance criteria. 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 thermal comfort and ventilation requirements. Component operations checks each piece of equipment to assure it is running properly; and the integration-level tests show that the entire system works as it should.

Operations Manual: Despite only accounting for 17 percent of the issues found, operational errors do cause building performance to degrade over time. Why? Operators are rarely trained on how the system works. They know how equipment works, but they do not often have the background to realize the systemic effects of some of their actions.

SUCCESSFUL APPROACH

Problems in modern buildings nearly always trace back to inadequate control systems programming, typically the result of inadequate software design. Software is as important a component of building controls as hardware, and requires its own specification. The software specification goes well beyond the standard sequence of operations provided as part of system designs today. This approach is required to make DDC systems finally deliver on their potential, and make well-performing buildings as commonplace as they should be.

We are pleased to report that after following this new specification and fixing only sequences (not hardware), Gerdin's energy consumption has dropped (confirmed via the utility bill) by 15 percent when measured directly, and 18 percent when normalized for weather.

RESOURCE

Interval Data Systems, Inc. *Pay Now or Pay Forever, Commissioning the Design of Control System Software*. National Conference on Building Commissioning. May 2007.

Greg Cmar is cofounder and CTO of Interval Data Systems, Inc. in Watertown, MA. E-mail him at *gcmar@intdatsys.com*. Bill Gnerre is cofounder and CEO of Interval Data Systems, Inc. E-mail him at *bgnerre@intdatsys.com*. Kevin Fuller is executive vice president and general manager of utility billing at Interval Data Systems. E-mail him at *kfuller@intdatsys.com*. This is their first article for *Facilities Manager*.