The Practical and Theoretical What's Happening with AI Delta IBT-20260730_095854-Meeting Recording

July 30, 2026, 3:58PM

1h 7m 1s

Emily Malin started transcription

Emily Malin 1:11

Okay, so it is 10, so we'll go ahead and get started. I'm going to go ahead and just paste this guy in the chat here. Okay, hi, everyone. Hello, I am Emily Malin. I am your product marketing specialist with the Delta IBT team. Thank you all for joining these webinars. So far, they've been really well received and we keep getting more and more of you, so we will keep holding them.

If you have any requests for what comes next, as always, please let us know. So just so everybody knows a couple of housekeeping items, this will be recorded. And after in about a week, we will send out all of the materials. So you will get the recording, the transcript, the PowerPoint. We go through and we do all of the Q&A. So if you can't stay until the very end, or if it goes late, you can still read through.

through the question and answer timeframe as well. All of that will be sent out with a survey in about a week. We are going to have all of your mics muted and hold all of the questions until the end. But feel free to type them up and pop them in the chat there. Just that way, when we go through them, we'll be able to just kind of go through one by one.

If you have any questions or like I said, you know, things that you'd like to see next, feel free. My e-mail is in the chat there. As well as in the chat, you will see a registration link to our next webinar. Our next webinar is going to be hosted by Andy Beaveridge and Zach Stevens. It will be on our new embedded graphics. So that will be August 27th.

at 10 A.m. Pacific Time. And then the following one in September will be on September 10th at 11 A.m. And that one will be an introduction to our new kind of product lines in the Delta IBT SBS family, which is Loy Tech. So Luke Barr will run us through kind of an overview of the Loy Tech products, and then we'll start weaving them into this webinar.

Just like that. So without further ado, here we go. The practical and theoretical with

AI. And Jason is our presenter, and I will let him take it from there because I wouldn't do him justice in an intro. So here we go, Jason. Let's kick it off.

Jayson Bursill 3:22

Thank you. Yeah, absolutely. So if I see some really easy questions in the chat, I am moderator on the side. I will try to answer them in line when I'm doing the presentation. So don't hesitate. But if you write me like a paragraph, that's going to have to wait till the end. Just a heads up there. So I guess a bit of background of who I am within Delta, because not everybody
totally knows this. So I'm Delta Intelligibility Technologies data scientist. We have a research group of technically 2 full-time people, but there's a lot of associated universities and labs that we work with, and we have funded students and researchers there. So all this kind of feeds into a
pipeline where we're bringing in kind of cutting-edge research.
manner. We typically work with applied researchers within those spaces. So we're kind of trying to be on the frontier of all these different things and integrate them directly with product. And that's ultimately what I'm responsible for within the research team at Delta.
So a bit of what I'm going to be talking about today. So the latest trends, I'm going to go over some hype cycles and what's hot and what's not and kind of how it's progressed over the last few years. So you understand how much of A truly moving target this is. Some emerging technologies and things that we've put into product as well with real world examples of stuff that's out there.
that's tractable. I don't go into case studies. We have other documentation on that, and Emily can help you with that. But I go over kind of more product usage at a very, very high level. I spent a bit of time, probably a good chunk of the presentation, talking about the emerging ATRA standards, because that's going to be very impactful very quickly. And I've become quite involved with these.
strategically, because I think we're starting to see a world that looks a lot like what BACnet began, but more focused on AI and agentic tooling around that. And then how Canvas is going to help on the AI side of things, and a bit of a sneak peek on some of the cool stuff we're doing to try to interface with the new standards that are coming out.
and be a little bit more AI ready. All right.
So hype cycles, I've been presenting this AI stuff on the road since about 2024, which

is why I pulled this hype cycle up because this is probably the first one that I presented on mass.
Things have moved a lot in that time. Generative AI was still kind of a fresh topic. Computer vision was still something that was really on the radar. And you think of like people counting cameras, that's just kind of a normal off the shelf product these days, right? But it wasn't necessarily in 2024. Things like knowledge graphs, which is where
You use data models to link different assets between the digital and physical realm and digital, digital, physical, physical. Basically, you're linking everything together relationally so you could do analysis on it. This has progressed a bit since then and is more mature. And if you look at the two to five year horizon of hype,
People thought it would be more mature by then, and it kind of is now, especially in the standard space, which I'll get into later. Something notable here. Agentic AI is nowhere on this hype graph, right? This is 2 years ago. I'm sure. Can people out here raise their hands if they have not?
heard of agentic AI. You don't need to define it, but do we have anybody that this is doc come up?
There's a couple. Okay, I'm surprised. I'll define that. But how many people do we have on this right now?
Uh...

Emily Malin 7:02
close to like 150ish.

Jayson Bursill 7:04
Yeah, so we're at about 1% has it heard of agentic AI? Well, 5%. So that's still not bad. So anyway, yeah, thanks guys. So this is still, it's early days for agentic AI, but everybody has heard about it and they didn't hear about it in 2024. So this is kind of a big deal.
Everybody's been talking about quantum, but nobody knows when that's going to happen. Artificial general intelligence, which how do you even define that? I don't totally know. And people I've talked to don't.
2025, you can sort of see how things progressed a bit.
Generative AI people get a bit disillusioned with the foundation models.
You're seeing the vision systems are just normal technology now. Knowledge graphs

are progressing again. Agents appear now, right? And this is becoming a hot topic. Ultimately, I think a lot of it comes off of the foundation model hype is down, but leveraging foundation models with stuff wrapped around it.

Now you can build some pretty interesting things and you start to get into a kind of higher level programming systems that you can work with. And this is where I think it's important to actually define what an agent is. So this is important because I think there's a lot of confusion around what an agent is. And even this is more detailed than what an agent ultimately is.

And that's why I've augmented this with retrieval augmented generation and tools. So really ultimately all that you're doing with an agent is you have a base prompt attached to what you're doing. So you can give it a prompt and then it has another prompt telling it what it's supposed to do. And then it uses a large language model to do other things. And if you don't attach anything else to it, this is really the only path that you're going to have.

where you interact with it, it has a prompt, you give it a prompt, it puts it all together, and then it retrieves things using the large language model. Now where this gets interesting is when you do this agentic retrieval log back to generation and add tools. So now your agent has the capability of, depending on what your prompt is, searching a bunch of descriptions of what these tools are, which are in natural language, and then selecting the right

tools to do things, right? And you'll have APIs that will describe what they do. And if you're interacting with a third party system, this is where you get into model context protocol, which is not a lot different from an API, but it has a really verbosed description so the agent could read it because all of these with these large language models requires an understanding of natural language or some

description of it to connect things, right? You're basically doing a match and retrieval, right? So you have these functions that are wrapped in descriptions and hosted on model context protocol, which is really a nutshell all it is. You have your knowledge bases, which is your data that's squished into a format that could be kind of searched based on language. And then you could just use regular old functions along the way as well, right? And that's ultimately what you're trying to do with model context protocol or APIs, where now you can just basically invoke traditional software, right?

So really what's happening is your agent is acting as a gateway leveraging large language models to connect to a bunch of different stuff and hopefully do what you

want it to do. That's the intent. But getting this to work the way it's supposed to work is not the easiest process. And this is probably the crux of how I'm involved with Canvas and Canvas AI.
So now that we know what agents are, they don't even publish the generative AI hype cycle at the same scale anymore. The focus has all been around agents. Kind of interesting stuff is no code agent builders, right? And no code systems. And I'll get into that in a little bit. Bottle context protocol, that's fairly high up on the hype cycle. And this is all a zoom in on the peak of inflatable.
expectations. So this is full hype from kind of mid 2025 to mid 2026. AI coding agents and think of like your codex, your clod code, multi-agent systems, everything is agents. You get into weird stuff like human agent collaboration workspaces where the human kind of acts like an agent.
I don't know what to do with this yet. I don't know if many people know what to do with this yet. I don't know how we use this in a building quite yet, but it is something that we have on our research roadmap, but it's on kind of a similar horizon to the hype. Well, no, for buildings, I put it at about 5 to 10 years before we need to worry about it too much.
before I need to worry about it too much. You guys probably longer.
But yeah, everything's kind of moving as agent stuff. Arion, I will sort of get into that at the end. Yeah, so bear with me here on the concept of can we use Intel web agent to request room temperature per building and floor report it. I'll explain how we're going to do that and what's in the progress to do that.
on our end. So all of this is centered around emerging technology. No-code applications are becoming a bit of a big deal, and maybe a quick raise of hands in the room. Who here has tried to build apps with no-code applications?
So far, this is less than I expected. Okay, here we go. We got about 5, 6. I would suggest trying this. Frankly, you probably don't even necessarily need to do it in an explicitly no-code application. You could probably push just basic Google AI to write you some code.
But if you want to build something a little more comprehensive and you want to have access to operating system and browser directly, well, I mean, you would have browser access if you're in a browser, but operating system is a little bit different. And that's where you want to get into these types of applications.
Fair, fair, fair. But yeah, no, ultimately, there's a lot of vibe coders out there. I've seen some incredible proofs of concept built or some incredible bespoke solutions that

could be done with these coding environments. And I think a lot of people are familiar with this, some more so than others. There's a lot of different things you can do there now.
And I've used it to build like BACnet crawlers and other odd things that probably don't make too much sense in this context. Skills is kind of the big thing. And I had to throw in a weird Napoleon Dynamite meme here. If anyone's seen that movie from like the 2000s, right, he's talking about skills, right? People like skills, nunchuck skills, bow hunting skills, computer hacking skills.
right? Now you're starting to write these skills that can connect to your agentic systems, and you can describe processes and how it should use a tool. And this really changes the game. And for a bit of insight, actually, when we first started building Canvas AI, skills wasn't even a thing. So we had to contextualize this in other ways. Once skills became a thing, this really ultimately changed how we've been doing things within Canvas AI. That is a bit of an example of some of the emerging technology and how it moves and how within these agentic systems, you can contextualize more processes. All right.
Model context protocol, which I talked about a little bit already. Now you can basically put a wrapper around your API, which is a wrapper around your traditional software. And now suddenly you can do a lot more things and hook it up to agentic systems. More efficient LLMs, and maybe even getting into using transformers that aren't necessarily
large language models for these types of applications. And I think more people are looking at this as we move a little bit more towards on-prem transformers. I'm sure you guys are kind of watching this stuff like I am as well. There is concern out there that we're going to start seeing a real rise in token prices.
as things start to become a little bit more sent out to shareholders and things like that. And they try to return to profitability or not return, enter profitability for a lot of these major companies that do host these models. So we're thinking about this and we're thinking about ways that we can use on-prem transformers. I explicitly say transformers instead of large language models because for some
applications, you might not need a large language model, right? So different ways you can get around costs of large language models, as I assume they will escalate in at least the near future. And then this is a really big one, and this is where I'm going to get into the ASHRAE standard space a bit, as we're really starting to see the solidification of semantic data ontologies.

where the ontologies is ultimately just the relationships of different things. It contains, you know, what's called a taxonomy, which is how everything relates to it. And then it contains definition, which is where the taxonomy extends to an ontology. And this gets quite interesting because now you can provide a lot of context very easily because you have a dictionary for everything. You can classify everything. And I'll give an example of some of those models later.
kind of what this looks like so you get an understanding of the depth of even just a very, very simple setup around this, right, with basic examples like BAB boxes.
So where is Delta delivering value in this already? Kalman filter operative temperature estimation for ceiling edge devices. We've been doing this for a while. I think we were doing this even before we were doing a camera-free occupant estimation. Both of these leverage at least some form of modeling and sensor fusion, and the camera-free has a quantized machine learning ANN model in the
the device to manage the modeling and the forecast, well, the prediction of what the number of people are in the room based on not having that number. There's a lot of power in the Python feature that is now on the Red 5 Edge devices. And I don't mean Red 5 Edge explicitly, I mean the Red 5 controllers.
And this is quite useful. The libraries in there are already pretty useful. It's the base Alpine Linux ones. The plus controllers contain NumPy. So you have a little bit more stuff you can do there in terms of numerical operations. And then of course, you can use Vibe coding to take your favorite functions out of your more
sizable libraries and just put them in there as PGs, right? So there's a lot of power there where you can even use the request library. And if it's connected to the internet, you can now interact with a large language model and build your whole own system on there as a thin client, right? And that's without any of the stuff that we're working on internally in the research space. You could do that out-of-the-box today.
Then of course, we're also continuing to improve the agentic BMS programming in Canvas AI. It's in feature preview right now, where Canvas is in beta. The goal of this is to move beyond just what we consider a greenfield initial programming of systems.
As building Canvas works towards Canvas Deploy, and I'll give a bit of a preview of what we're doing to make this all work later in the presentation. We're working towards being able to interact with existing assets on current systems. So once you have the ability to go from engineering devices to deploying devices,

Our next step is to be able to do an inventory of all these deployed devices and then allow you to interact, much like that question about can you pull the data out of IntelliWeb. Well, that is the intent once we implement the new features that are coming for Canvas 2.0.
So some of this is going to be familiar to people because I presented this a lot before, but I feel the need to go over it because this talks a bit about how our AI systems work in our existing technology. So the camera pre-occupy estimation, this is one of my favorites, where you're using multiple different inputs to get a single output, and then there's an artificial neural network model, and we evaluate it.
with kind of normal metrics, your mean absolute error of the people count, and your person is within two to make sure that you're not too far off that it's useful, which I would argue it's still significantly more useful and reactive than a CO2 sensor. And I think for many people in here, I'm kind of preaching to the choir, so to speak. But you get a lot quicker response because it's using rapid response sensors like passive infrared and infrared temperature, which are two different things. So yeah, inside the system, it's going to show up as AV40 on the O3 device. This is kind of what it looks like inside the teleweb. Ultimately, it gives you a privacy preserving self-contained solution. If you're using the O3 Edge,
You also have a fully programmable controller that you can integrate with this, so it takes the thermostats off the wall.
And a lot of people don't even use their thermostats anyway. They like having the thermostat, but it depends. And then we also found through the studies that we did that 86% of meetings contain six people or less. So that really helped with justifying the, you know, for a small room solution, it's perfectly fine. And then I love to show this graph because it shows you the raw analog to digital conversion count, which shows the real non-linearity of the model.
model inside the system. The green line down here is the estimated count of people within the room. And then the black line is the ground truth. And then plus minus 2 shows that we're generally within the persons within two metric.
And I forget if this slips out once. I think it might slip out once at the end. Yeah, there you go. Yeah. So it's not always within two, but you can't really necessarily expect that for every single room, a static model, and really the intense non-linearity of the signals that go in. And this is just the passive infrared sensor. As I said before, there's luminosity.
There's relative humidity. All these different things go into the algorithm.

And then within the general research space, and I think some people on here have probably been involved in some work that I've done with doing research on buildings in live sites. So when we do go and do research in live buildings, we kind of work around this cloud supervisory control architecture. And I've done this in a few buildings where

We want to be able to roll back to baseline. And this is less of a product thing, and this is more of a research thing that I thought would be useful to explain kind of when we do go and actually test stuff and validate this actually saves energy when we do this, or this doesn't cause wild fluctuations in control performance in algorithms we put together in research.

This is more or less what it looks like. So we have the building automation system. There's processes for sampling and archiving data. We do modeling, and I'll get into some of the modeling that we do around predictive control, which is currently the most promising of the algorithms I have running out in the field in a few different sites. And then we're implementing weather data, whether it's historical or forecasted, right? So all of this kind of latches on, there's kill switches, and it's generally very safe. We haven't seen any issues with running this where people see erratic behavior or get complaint calls, which typically actually we don't, then they could do something about it. The only time we've actually gotten complaint calls is when we've done.

Demand response. So we know there's going to be a peak electricity event coming up. People want to turn off chillers, so we pre-cool the space a bit during the lower cost period of time. We've gotten some complaints because the space has gotten very cold, but they don't really complain once it warms up later. So moving into the predictor control stuff, and bear with me a little bit here.

We've been exploring a lot of different things. Primarily the focus has been looking at the zone level, because ultimately when you're dealing with your HVAC systems, the zone level is really going to be what drives your load. Your air handling unit is going to respond to the requests of the zone and the needs of the zone, and then the plant is going to feed the air handling units in the zones as required.

but the loading at the zone is ultimately what you're treating for in the HVAC system. So this is why this is where we really focus where predictive control goes, because if we can reduce demand there, it'll trickle up. I know that affects boiler curves and things like that and chiller performance and things. A lot of people are looking at that too. So there's a lot of literature out there, but this, there's a little bit less, which

is why I think it's important for us to look at this.
So ultimately, we want to try and reduce zone air flow, optimize discharge air temperature and maintain comfort. And then we also measure within that some of this overlaps, the indoor air temperature and the indoor air temperature set point to validate thermal comfort. We'd like to do more. There are some studies that we're working on.
that are adjacent to this, where we're using the O3 and the infrared temperature and the operative temperature to try to get a better idea of what the feeling is of comfort in the space, because air at the thermostat isn't the only thing. And if the thermal mass is getting a little bit warm or a little bit cold relative to the space, people are going to feel that. And that
ultimately will impact their comfort. So we look at a few different model formats, things like state space, lumped capacitance, which really just looks like a little RC circuit, basic electronic stuff. And we try to keep these really simple because it's easier to troubleshoot. We only have one resistor and one capacitor. And then also some seasonal models in this
gets to have a really long name, seasonal autoregressive integrated moving average with exogenous regressors, which really just ultimately means there's cyclical tendencies of the space and they're daily cyclical, they're seasonally cyclical, and they're also related to what inputs, external inputs, which is exogenous, but go into the system. So it kind of follows a bit of a pattern and you're fitting a model with a bunch of different weights.
So in the particular control space, we're looking at a bunch of different algorithms, and it's a bit of a race to see which is the least frustrating to implement, to the point where we're also interacting with other research groups and kind of trading algorithms between spaces and seeing, okay, how hard is it to do theirs? Does it work better? How hard is it for them to implement ours, right?
So there's a bit of action in this space in the research space with the intent that we'll be able to have an easily deployable Python algorithm in the edge devices sooner rather than later.
Probably more of the focus around AI stuff is this independent data layer. And I'll get a bit more into what Canvas is doing around this later in the presentation. And this is kind of the big piece that enables a lot of other technology. And ultimately, this whole independent data layer concept is around having a bunch of different data sources below

We can call them databases, but like this could be IntelliWeb. This could be your drawing repository. This could be some sort of CMMS system, computerized maintenance management system. And all of this would feed into an independent data layer, which can accept multiple different types of data.
The word here that I like to use is multimodal for the multiple different types of data. And then all of this has a semantic data model attached to the different assets in here, which can be easily leveraged by applications on top. Or you'll have your analytics dashboard, which these types of things, once you have this and once you have the API, are surprisingly easy to code up these days.
Web applications, similar idea. We've had 4th year engineering teams that have put together some pretty cool web applications and mobile applications around just things like air balancing, which I'm sure somebody will talk about this in the near future when we talk about our different capstone teams and different outreach that we do with schools.
which isn't the focus of this presentation, but I could talk more if people are interested. And really anything else you would put on top to pipe out data from this, where you have the raw data coming in, it gets normalized through different processes, which I don't talk about too much in this, but you'll see the tools that attach to it and the ontologies attached to it, and then the applications on top.
So within the North American context, there's been a lot of movement on this lately. You're probably all familiar with BACnet and APIs, and this is one of the puzzle pieces of enabling AI in buildings, in my opinion. Sequence-wise, you guys have been building sequences forever, or at least as long as you've been using our products. There's also ASHRAE guideline 36 that's out in the wild. That's another thing. And we have support of that now. And we're, you know, this close to finishing the CEC part. And from there, now kind of on this side, the blue side, which this is a little bit more new, ASHRAE standard 231 control description language, defining different control algorithms and showing how to encapsulate them. I'll talk about that a little bit today.
today, this normalizes what algorithms look like. It doesn't have coverage over particularly complex algorithms yet, and I'm not going to get too deep into that in this presentation, but I will say that I'm personally trying to work on that through Ashraf interactions. But on the syntactic side as well, I've talked a lot about Brick in the past. Many of you are hopefully familiar with Haystack.
project haystack and putting markers and labels and having a dictionary of things to

put on that. Brick goes a little bit deeper, which I'll show very shortly. And then brick functions very similarly to ASHRAE 223, which technically it's not a standard yet, but it's past second public review. And I would be very surprised if it does not have a 2026
tacked on the end of it and becomes a standard before the end of the calendar year. But don't ever quote me on that. This is just my suspicion. So all of this works together where you can identify what the system is with ontologies. You can identify the algorithm with the structure. The sequences that go into the structure are defined by something.
and then they are transported through the system through something like BACnet or an API or other protocols. So now all of this is well indexed, is well structured, it's defined, it's standardized, and once things are standardized, now suddenly AI and agents can interact with them.
So, semantic data modeling with RDF and Ashraf 223P. This is where it's going to get a little weird, so bear with me and hold some questions to the end. This is ultimately what this stuff looks like. Really, RDF is just a bunch of text content where you'll have a name space that's unique to your project. In this case, we just use
building to describe this. You'll have, for example, a brick start stop command, which this is a supply fan one start stop. That's kind of what this label would mean. Fan statuses and a bunch of other stuff that starts to come out of this that probably looks kind of familiar to you guys. And this would be not an unusual name that you would give to it in BACnet.
But the big important part of this is that you need to ensure that you relate everything, and you need to ensure that you classify everything. And ultimately, this is just the text representation of this model over here. This initial example I'm going to give is just brick and a project.
because 223 gets a bit more complicated, and I'll explain that in a moment. So as we dip into this here, you can sort of see what the model starts to look like. And this is just a basic visualization with an RDF viewer. So you're taking that text that I showed before and turning it into something with, it kind of looks like spaghetti and meat. balls, right? But you go through it and you'll see that you have floor 2 and it's got room 2000, which is a mechanical room. All of these are related together with what's called triples, where you have the subject, the object, and the predicate, or the subject and the object can be reversed depending on what point of reference you're using.

but it can still be traversed as long as these things are linked, right? And typical query systems have pretty good MCP wrappers around them these days. And then when you couple those with agents, your agent can now use the MCP wrapper to traverse these models and quickly find out a lot of information about the system. So
You're classifying a floor as a floor. You're connecting floors to rooms. And then as you go through this, you're seeing all the different pieces. So you'll have like an exhaust fan, different coils for heating, cooling, supply air temperatures, all your air handling unit stuff. And all of this gets classified. Exhaust fans. And then at the end, you have your brick classes for what these different things are.
And you can sort of see here where you have, you'll have a VAV, your VAV will have a thermostat attached to it, which I believe I even neglected in this model to make it simpler. And then you'll see all the different types of sensors that are attached and they'll have a room label to it, which is just your BACnet names. And then that gets stripped out to give it a.
class. And the AI tooling will know what these classes are, and each of these classes in Brick has a definition. So now it has a ton of information on the system just because you've linked this together. And ultimately, in Canvas AI, this is what we're doing with the greenfield side of things. And as I'll show a bit later, we have supporting technology for the brownfield side of things.
for 2.0.
So now we're going to get a 223 a bit. I tried to pick the simplest possible system I could, and I'll start zooming in and flying around this a bit. But this is just a cooling only VAV and how it's represented in ASHRAE 223. So ASHRAE 223, the intent of it is to supplement things like Brick and Project Haystack.
So you're going to see, probably within a model, you're going to see your project has a name space. You're going to see.
Brick would have a name space. Brick is harmonized with real estate core for locations and real estate centric assets. And then 223 really gets down to the more the more finite pieces, right? And this gets a bit smaller and you'll see what I'm talking about in a moment. So this is just the name space for
the model, right? So it's my system. It has a physical space. And then this gets defined as an ASHRAE 223 domain space, right? And then a zone will be defined as a 223 zone. And this has some overlap with brick and real estate core. And you can see that they're conceptually quite similar because they all use triples.
And the spaghetti and meatballs looks kind of the same. You also have ways of

putting labels on things, right? So you can put just whatever text you want in there. And this gets back to having the flexibility like haystack tags. So that's pretty easy. And then now we can look at some of the VAV parts. So this is a guideline 36 cooling only VAV. So you have a whatever you want to call your VAV. I'll get to that question at the end. That's too long. And the guideline 36 VAV, another namespace is added specifically for that. But a cooling only VAV in the 223 namespace is actually called a single duct
terminal unit, right? Because ultimately that kind of is what it is at its most minimal level, right? There's also actuators for the damper actuator, but it doesn't explicitly say that it's for a damper because you have to point to that somewhere else, right? And you can add more labels.
It gets granular down to what the quantities are moving through the sensors. All right? So your discharge airflow sensor is a sensor and it has a, that's not like to the dimensionless ratio. Sorry, let's go to the damper position. The damper position has a unit of percent. It's A dimensionless ratio.
pulling in some more namespaces that come from physics. It's a sensor. And then, yeah, so also actuators tend to be called in the 223 namespace, quantifiable, actuable properties. So other stuff to be aware of is there's a descriptive name for these different things.
that might be not exactly what you would call it in the working space. And then when you get to things like CO2, you'll have a CO2 concentration sensor with the units of PPM. It uses a constituent CO2, and because it is an input sensor, it is a quantifiable observable property, not actual. All right. So there's a lot of levels of detail. It also differentiates between outlet and inlet connection points. So these things that might not have had the granularity in brick or even necessarily haystack without explicitly saying that it's an inlet or outlet, this is intended to be quantified within 223. So together with
brick and 223 and a few other things. You can put together a pretty comprehensive model that's all in human readable and machine readable, more importantly, formats. And then here we get into ducts and stuff and you kind of get the idea. So BACnet's bumping up support for RDF as well.
which RDF, I think I defined it already, but I'll define it again just in case, resource description framework, which is just that format that I showed you that's got the triples, it's got the weird text. And then all of this goes together to build you a large semantic data model that can be ported between projects and buildings. You can

template systems, which I'll give some links to some templated systems that the national labs in the US
have already put together that are freely, openly accessible that you can review. And then you can sort of go from there. And this is basically the technology that we're working with on the back end of Canvas AI right now, because we want to be really well aligned with these standards around semantic data models. And for my European friends, from my understanding, 223, once it gets published,
in Ashraf is then going to go to do ISO. That's what I've been told. So the intent for 223 is for it to have international reach. So it's not just going to be an American thing or a North American thing more specifically. So yeah, it's similar to what's happened with BACnet in a lot of cases, right? And BACnet, this has actually gone out. I took a snip of
the first one that went out for public review, but there's a second one out for public review, which to me implies that it'll probably end up in the next or the one after version of BACnet. But they're trying to make BACnet align with this RDF data structure as well, which starts to get quite interesting. And there's a copy of it out there. And basically the idea is that you should be able to freely go between 223 and 135, which is BACnet.
This is sort of an example of what the BACnet stuff looks like. And if you recall what I showed you with the RDF text for 223, it looks kind of the same. Ah, this is a good question. We are working on this and this is probably the next slide. Oh, no, it's not. It's a couple of slides later. So bear with me, CDL folks.
Is that going to add a standard to include a tag name?
So Chris, it actually technically has an optional property for tag already. And we make use of this in a few cases. I think it's been in there. I forget whether it was in BACnet 2016 or if it was an addendum to 2016, but it is an optional property. Delta supports it in V4 firmware as of version 4.13.
It's not necessarily universal support. And the whole concept of tag is a bit of a, where you put the tag is controversial. And I've seen a lot of debate about that, which is why people have just said, here's RDF, let's figure out what to do with it. And there's been a lot of work actually around taking this BACnet RDF and serializing it into the BACnet tags property.
which we've done a little bit of work in this. I don't think there's a standardized way to do this out in the wild right now, but I'm trying to hopefully find a more comprehensive way to do it that can work between different applications and

systems, right? And I'm thinking putting analytics stuff on top or being able to put systems easily on top. We've just basically been using...
We've been very explicit about the name that we give to it. And part of the issue with the back that tags property, and this is a bit of a tangent, but bear with me here, it's a tuple right now. So you have a key and a value, but this RDF stuff is all based on triples. So you have to have three things, not two. There's some ways to work around it, but this has been a challenge.
kind of the guy that's really behind this, his name's Joel Bender, and he's based out of Cornell. And him and I have had a few talks on this, and I don't know if we've aggregated to something completely yet, but I have ways that we're doing it. And this BACnet RDF addendum actually gives a lot of different options and some examples of how it works. So if you are an ASHRAE member,
This went up for public review last Friday. So I would say give it a review if you're into BACnet stuff. Send some comments in if you have comments, or at least familiarize yourself with the landscape of where this is going, because I think we're only going to see more RDF style stuff. And this kind of bridges the gap between, you know, the BACnet tuple and the RDF triple.
So I'm going to go through one more example and I'll talk about CDL a little bit at a very high level and my thoughts around CDL, which kind of a random reference here. I'm actually now the chair of the Ashraf 231 standing committee. So technically Delta is quite involved in CDL because I'm chairing the CDL standard.
So hopefully we'll see a little bit more integration with that, and I'll get into that in a moment. But here's a very basic example of a VAV3D using BACnet, RDF, and Brick, and this is where we're linking all the assets together. I've stripped out the 223 stuff at this point because the 223 deals more with the physical sensor parts. And the intent of this particular example was to work with
canvas engineer and cloud engineering where you don't really have a physical IO and you don't have any idea where that would be because there's no context to that in the system. So within this, this sort of shows the different layers and what they do. So you have your building and external relationships, your equipment and components and your control points and classification.
And then zooming in on this, you can see you have your very basic components like your damper and reheat valve and a VV with reheat, which are ultimately your two actuators. You define your equipment so that it could be bundled.
And then through the use of what's called in the RDF space a blank node, which is

exactly what it sounds like, you can now have multiple things come out of one spot. And this is where you can have your point classifications, and then you can do a blank node to give your exact BACnet URL, which it doesn't look quite like this at IntelliWeb, you get the idea with a bit of massaging, you get there. But this is just easy to conceptually show, and I show this in a non-commercial manner most of the time. And then you have your BACnet references and your BACnet controllers. So it becomes quite simple when you just look at the brick application layer and 223 adds a lot of detail at the lower levels.

231. What are we doing with 231? So within ASHRAE Standard 231, there's a subset of libraries. I put a link to this at the bottom and its most native format within Open Building Control, which is currently Lawrence Berkeley National Labs distribution. of CDL in the Modelica modeling slash programming language. I could show an example of this after, and it might be worth that. I do a little bit of an example here with the cooling only VAV, or this is the VAV3he, hold on.

Yeah, that was cooling only. So yeah, I do with the cooling only VAV for a very simple example. But there is a strong relationship between CDL, ASHRAE standard 231 and ASHRAE guideline 36, because within standard 231, there are guideline 36 systems, right?

And this gets into those puzzle pieces I showed earlier, where guideline 36 gives you a sequence to then build an algorithm that could be encapsulated in control description language, right? Which ultimately it ends up looking a lot like this. And we'll zoom in here, where you can see very clearly there's function block. This is based on function block.

programming, but as we all know, these function blocks contain, as the name would imply, functions, and there's text code behind these. But here you have your different inputs to the system. So you have your zone temperature, your cooling set point, your heating set point, your window status, occupancy, your operating mode, your CO2 set point, your CO2 reading.

your discharge, temperature, supply. And this is where I get a little bit agitated, to be frank, because the names that are on here, they're based to work with the Modelica, right? And they're based on how Modelica's names are put together. In my opinion, these should all be aligned with Brick and 230

or 223. And in terms of future work, I think that's something that we're going to do within the 231 committee, and there's been a lot of interest in doing that. So I think that for the greater good, we should be doing that anyway, whether it's within the

Modelica realm or whether it's within standards. I know there's certain limitations on what you could do with the names in the Modelica software.
and certain characters that you can and can't have. All right. But here you can see the different outputs of the function blocks. So it's pretty clear how it works. You don't see all the lines going on in the background, which if anyone's played around with our guideline 36 libraries in terms of alignment with what Delta's doing,
We have taken our guideline 36 programming and our GCL programs are aligned and structured to be essentially functional blocks, right? It's not a visual function block, but the IO is very clearly declared at the top and the bottom. There's very clear demarcation of what subsections do within the scripting of the GCL.
So we tried to move towards making our guideline 36 implementation aligned very well with how CDL works. Kind of the next step with that is actually explicitly making it into CDL functional blocks and using, within control description language, there's another thing called control exchange format, CXF, and that's based on using what's called JSON link data.
which is just basically a file that shows you what is linked to what, and you can link your real IO point to the thing in the block algorithm, right? So all this goes together nicely. We're making efforts to align more with this, but I think there's an element of kind of things coming together that we need to be aware of in.
in the process, right? Yeah, and then you also have alarming down here, and it gets into alarms and requests and damper position, flow set point, and how it's doing relative to that, where this would pump upstream to an air handling unit, and then you have a fan, and whether or not you have an alarm based on what's happening with that.
And some other links, if people want to take a screen cap or something, there's a lot of 223 examples, and I don't know if I'll have time to go into them in this, but I'm happy to talk more later if people are interested. I would say peruse the link 1st and see this because the 223 examples are ASHRAE guideline 36 systems.
And then the 231 data, Ashraf has a section specifically for it, and it links you to what I showed and more. And it shows you some of the different systems and some of the diagrams. There's a lot to explore there. All of it's centered around ASHRAE guidelines, 36 algorithms, but you could put anything in there. Just be aware of some of the limitations. 231 is very much tied to Modelica right now.
And 223 technically isn't officially published, so there's a bit of work in progress there. But both of them are coming kind of close, right? And all this feeds into a

semantic data model-based workflow, which I think is very exciting. So once you have all this information, a data model that you built with your system,
Using a tool like Canvas, where you're building your semantic data model alongside your system, or you're able to, once we get what's called brownfield auto-tagging in, retroactively go through your existing assets or your templates and then attach classes and then build systems, much like I showed with the spaghetti and meatballs. balls that you can use as blocks together to build a larger system, right? So now you have all that, you have your semantic metadata, you can rapidly build GUIs with AI. That becomes very simple. And as we add more MCP features to our different products, you'll be able to do that more easily within our environment. Rapidly configure analytics from
semi-public libraries. And I think this becomes almost a trivial thing once you have this entire semantic data model of your system. Everything's classified. AI ages can traverse this system. Single pane of glass, you know, everything's kind of in one location anyway. So you're pulling that out and you can pull all the data into one location.
Normalize context, this gets into the auto-tagging that I was talking about. And it kind of inverts the whole workflow of how people would traditionally program a building, right? So right now you kind of work this way. You open up something that does BACnet, probably, if you're working in BACnet. You can do other things, but I try to...
tie this to the Ashraf Ray examples. Then you pull out your functions and your code that you use from your templates. You give names to these objects that are appropriate to the project, and then you ensure that all of this is saved to Flash and it's out in the system, right? With semantic data models and building them at the source, you kind of flip this whole process upside down. So you start with...
I need a VAV that has these features, right? It pulls the semantics of the IO. You find the appropriate algorithm for what you want, and it pulls that down as well. And then it spits it out in the target environment, which is how we're trying to do things with Canvas AI.
And it's out in feature preview right now. We're continuously trying to improve it, and I'm personally involved with some pretty intense improvements that we're working on right now. So this should be very exciting in the next couple of versions. But it is there if you want to try it out. Basically, you could describe what you want to build.

It'll ask you some questions along the way and then you can apply changes and it spits it out into the workspace so you can play around with the system in a combination of kind of a cloud engineering and a teleweb-like environment.
But this is of course tricky because I feel like we're moving with emerging technology. I like to think of this as kind of building a continental railway, but with nerds and futuristic stuff. So the AI field is always moving. There's always new technology. Suddenly we have things like skills that we didn't have a year ago. And I don't mean like human skills because hopefully we have skills, but I mean like AI skills and the
form of a text file that tells it how to do things. So little things like that that we have to be aware of. I think we're getting close to a bit more stability, right? And this makes it a little bit easier for us to do the development process. So just stuff to be aware of along the way. I wouldn't be surprised if next year's hype cycle looks very different.
Agents might be almost passe, like they're going to have use, right? And people are going to find the full uses for the technology. Maybe passe is the wrong word, but it won't be as much of a hype thing, right? Kind of like LLMs are like, everybody knows what those are now. Everybody knows what AI is sort of now. So this is the big thing. This is how we're going to really improve what Canvas does with the deployment. deploy phase and the people that talk about Canvas Roadmap, it's going to be on there. People talk about this in more detail in a product sense than I do. Remember, this is research. But the whole concept here is really implementing the independent data layer ontologies and multimodal data in one single system, right? And I'm not going to tell you what our internal name is, but you might be able to get from my weird AI-generated meme image here.
But ultimately, we have this multimodal database where you store text, which is going to be your RDF data and your BACnet data. You can store images of drawings and things that you would bring in and scrape text out of to be able to program your systems. Audio, maybe not. I don't know. It's multimodal.
I'm not going to confirm that, Robert. You can guess. That's a bit dystopian, Terry. But yes, so all these different forms of data that can come in there, and you got your time series that can be accessed through links to IntelliVault and other systems. right? So this is really enabled by having stability and resource description framework with Brick and Ashraf 223, model context protocol, which is a very early target for this system because you need to be able to put AI stuff on top of this and rapidly iterate

on your own systems, what we call brownfield auto-tagging, which I've talked about in other presentations.
And I won't get into in this, but you know, labeling your existing digital assets and tying them to physical assets or at least physical asset classes, all of that would natively be in there. And this is stuff that we're targeting for around that time. And then as I said, ingestion of drawings and things. So it gets very exciting because now you're going to be able to have records of things you're making and things you've already made. And you can ask AI about it and you can latch it on with MC.
MCP. I'm very excited about this. So some closing remarks before I open up for a bit of Q&A. The AI space is really moving quickly. The focus has moved to agentic applications, and we're really trying to fine tune those as the technology stabilizes. Semantic data modeling really ties everything together.
And Ashraf and Brick, in my opinion, are leading the way in terms of innovation and making this ready for AI systems. Of course, there is Project Haystack out there, and I know they're doing some good work as well. My personal preference is this, though, but I know that we also have to interact with these systems. And I would also argue that there is some convertibility between these where you can add brick-level context to a haystack system, but that's a whole other.
conversation and presentation. Canvas AI will continue to improve. As I said, skills, we're working on fixing the skills. And I would say unmanaged analytics, I'm very careful about calling it unmanaged analytics. When you have kind of a human somewhere in the loop interacting with the data and kind of
checking the outputs of the AI systems and checking the outputs of the rule-based, I guess you could call them AI systems, heuristic AI, which is a little bit more of an older technology within the AI space around expert systems. That, I think, is going to be a bit commodified by agents, model context protocol, and semantic data modeling.
So I think suddenly we start to get to the point where now you have a data model and it could be accessed by an agent through an API like MCP connection.
You don't necessarily need that unless you like having the person there to help you through the data, which I think there is some value there in having someone to do that. Maybe it's you guys, the partners, that are managing that part of the system, right?
So yeah, I guess opening up to questions now.

I think we'll go through the chat. Emily, have you been tracking which ones I answered in line or not?

Emily Malin 54:37
Yeah, perfect. Yeah, I was just gonna say, yeah.
Well, sort of, it was tough, but I'm pretty sure you did get to the first one is the IntelliWeb agent to request the room temperature. But if you want to say any more in that piece of it.

Jayson Bursill 54:44
Yeah.
Yeah.
That's gonna be Atlas, or that's gonna be Canvas once we do this Atlas thing.
Whoops. Yeah.

Emily Malin 54:56
Yeah.
Well...
Who are you going to blame for that? Blame Robert for that. He guessed it. It's fine. We'll cut that out of the recording. No, it's good. It's good. Yep, that was just in the Eweb MCP. I'm pretty sure you went through that.

Jayson Bursill 55:01
Uh, yeah, I didn't say it. Uh.
The.
Yeah, yeah.

Emily Malin 55:14
The one from Robert was the paragraph one. If you want to look at that one real quick, I don't think we went over that.

Jayson Bursill 55:19
Yeah, yeah, can the standard naming structure be suggested or are you going to be for the future? So the ontology classes exist in parallel to the BACnet object name. So we've intentionally tried to do it this way so that you could continue to conform to whatever naming standard you want. And part of it is I am totally aware that a lot...

of end clients are very particular about the object names and that's still going to be a way that they like to interact with their systems and index their systems.
I don't think I could convince them to change that in the next 10, 20 years. So that's why we've got a parallel system to manage that. I would say in terms of naming structure, you could still continue to use your same naming structure, but the classes behind that that are backed to the back that object names in the semantic data model in the RDF format.
That is going to have to adhere to brick and 223 and all of the other ontologies. And we're also building in validation systems for that so you can assure that you've actually done the correct classification. And in principle, our system should do that correctly, but in terms of manual labeling, things can go a little off the walls.
I think the next one I didn't answer is also Robert.

EM **Emily Malin** 56:29
Delta.
Yes.

JB **Jayson Bursill** 56:33
Yeah, yeah, I talk about CDL. Yep, yep, the answer is yes, yeah.

EM **Emily Malin** 56:33
Yeah, because you did, is Delta doing anything with CDL? Yeah, is BACnet going to add a standard to include a tag name?

JB **Jayson Bursill** 56:42
It technically kind of has already, but it's not required, it's optional.

EM **Emily Malin** 56:42
Mm-mm.
Yep, yep, so then the next two, I don't think we talked about.

JB **Jayson Bursill** 56:52
Alignments of competing brands, technologies that might exclude a brand.
Oh, are you talking about like Delta Controls versus LoytecSupport, Robert?
Is that what you're talking about?
Oh, thank you, Terry.

Emily Malin 57:16
No.

Jayson Bursill 57:18
Ah, that, yeah, yeah, yeah. So internally, I'm doing this, but I haven't shared it externally, and I'm hesitant to throw tomatoes at people. So if you want to have a chat with me at some point, Robert, I'm happy to give you my personal unfiltered opinions, but I don't want to be that guy in a
public forum. Yeah, but I do think that what we're doing, some of the approaches that we do are a little bit different than what other groups are doing. We built most of our, well, pretty much all of our functionality at this point in house as well. We haven't gone and acquired a company, which is something that I think is distinctly different and not controversial.
And in terms of technology, we've attacked it more at the engineering and building out, right? So we can kind of, you know, do it right at the beginning and then do more stuff with it later, right? Rather than putting an agent on top of things, right? So just something to be aware of.
Ah, yes, yes. We'll get, I'll come back to that second, Robert. Sorry. DIBT coin. I won't be doing DIBT Delta IBT coin. Yeah, no, no, no. Yeah, somebody, somebody. I know people that might do that, but it won't be me. Are there any AI agents for BMS platforms from any OEM?

Emily Malin 58:28
Somebody should, though.

Jayson Bursill 58:38
Um...
Ramsey, you're going to need to elaborate on what you mean by that. Do you mean at the BMS activity level or do you mean at like the analytics level? Because those are different things and technically you can say they're for BMS platforms and...
Would they be from OEM? And some OEMs are working on stuff like that, depending on what layer they are. But engineering side, I haven't seen.
too much of this, maybe one or two, but I don't know the details of the technology.

Thank you, Amaan.
Thank you, Emily.

Emily Malin 59:22
Ramsey just said on the platform using for operational reasons about that OEM question.

Jayson Bursill 59:23
Right.
For operational reasons, yeah.

Emily Malin 59:29
Daily, daily operational help agents.

Jayson Bursill 59:33
Yeah, yeah, there's a few companies out there. And actually, if you look at what Robert listed, those guys do have some systems that do that, right? I'm not going to name them. And there's probably a few more out there, right? And I think some of the other platforms have something you can use on top.
Honestly, I think once the data model's there, even if you have a good naming system, you could probably just whip up Google Chrome and WebMCP and build something that does a pretty good job of that with IntelliWeb. And I know of some people that have done similar things to that, right, which is why we've kind of attacked the more core problem of doing it at the programming level first, right, as that technology matures. Our intent is that you'll be able to kind of do that out-of-the-box with
you know, publicly available tools. All right. Yeah, but in terms of differentiation, what we're doing is really at the core of the design and the deployment of the building, rather than doing it after the fact. Because if you want to do this well, you probably need a well-labeled building. And I'm talking about these like ones that are restricted to like 8 characters in the object name, and that's the only thing they have as a tag. And then you go in and you

Emily Malin 1:00:18
Any suggestion?

Jayson Bursill 1:00:38

to pay some company a lot of money to go and label these things, and you probably need someone with a lot of expertise to go and label this system, right? So, yeah, so it's a little bit different. We're trying to make it so you don't have to spend all this time doing data labeling with expensive people on kind of a low value task, right? And that's what I think the differentiation is, is to make integration a lot easier. More cost-effective and just better for everybody, right?

Emily Malin 1:01:03

We have about two more here and I know that some folks have to jump because we are pushing right up to the 11. So just again, everything will be recorded and sent out for about a week. But so if you have to jump, you know, that's okay. The next one, Jason, it was any suggestions on current low hanging fruit pathways for modeling and driving AI value across a large scale portfolio?

Jayson Bursill 1:01:05

Yeah.

Yeah.

Hmm.

Oh.

Honestly, if you have clean data in a spot, and there's a few different applications of the public domain that you can look at to look at a lot of data very easily, right? So if you're in a spot where the data is already clean, there's some pretty low hanging fruits there.

It's pretty easy to build very basic analytical applications and even a BACnet interpreter at this point to do things, right? So I think playing around with that and probably more importantly, learning how to use AI vibe coding systems to build very basic applications, it's come a long way and you don't have to actually know how to do any programming at all.

at this point, as long as you know what you're looking for and how to validate it. It's in a much better spot than it was even a year ago, right? So that'd be my low hanging fruit. Learn how to vibe code well with things you've got, right? If you understand the application, it can get you pretty close as long as you don't need to scale it into a stable product.

Emily Malin 1:02:26
Okay, I think the next one was, can you build up tagging when you engineer the BMS? When putting in sensors and actuator, it builds up the tagging along. That makes sense.

Jayson Bursill 1:02:36
Absolutely. And that is the intent of Canvas AI and that's what it's intended to do today. We have a lot of bugs. We're working on them. But Greenfield, this is the intent where you can start with guideline 36 templates or your own templates. The brick class tags are already built into the template. So as you change names of things, it's irrelevant.
because it is now linked to a class in the background. And unless you change what that sensor does or what that point does, it will still remain known as the same thing to the AI in the back end of the system to make it fully integratable with everything else.
Also, hello, Moose.

Emily Malin 1:03:12
Final question was, Hendra, when are we going to Australia? So...

Jayson Bursill 1:03:16
I hope soon, Hendra. I, yeah, yeah, yeah.

Emily Malin 1:03:19
We need to, yeah. From our marketing perspective as well, I'll e-mail you, Hendra, about that because you're also on my list for Canvas things. So.

Jayson Bursill 1:03:26
Yeah.
Yeah, and Tansy is right. Test and harnesses. That's, yeah, that's the important part. I won't go deeper than that. Yeah.

Emily Malin 1:03:38

Are these templates in Canvas going to be exportable into something we can hand to the customer?

Jayson Bursill 1:03:43

Yeah, that's the intent. They should be fully exportable and the system should be able to export them for you. I think it does now if you get to that part of it, right? But it might not be in the most clean format. And if you have thoughts on what you want as formats, that would be useful to know because we can also tweak reporting skills and things like that.

within the system, which I know everybody likes the reports a little bit different. So that gets a little bit weird. But at least in terms of precise data you would want, a little bit of guidance would be useful. So if you have some stuff you want to send me or Emily, I can send that along to the Canvas team.

Mmh.

Emily Malin 1:04:22

Perfect. Well, I know we're a little bit past, so I appreciate everybody sticking around. Oh, wait, we got one more coming in. The product you showed us from Delta Occupancy Centers in the market for a few years now. When will we see some of these in your DDC controllers and include AI programming from the factory itself?

Jayson Bursill 1:04:24

Mm.

Ohh.

A...

Ah, yes. So Red 5 has the flexibility to do that through Python right now. And hopefully not too much of a tangent, but I have a few graduate students working on building out some algorithms at the controller level. So I got a few guys that have just a Red 5 plus and they're doing a bunch of crazy things and building predictive controllers completely at the device level.

fully constrained within that. The results are pretty radical. So in terms of AI from the factory, my vision for this is that we will, within the next couple of years-ish, I never put timelines to things because I don't own the roadmap at the product level. But we'd like to have like a DLM that's inside a VAV controller that you can now engage a predictive controller which has machine learning in it. In terms of AI

programming,
Yeah, it depends what you consider AI programming, and that needs to be defined a bit better, right? I mean, you could connect to a large language model agent system from a controller today, right? Or you could use systems to make Python programming for a controller also today, right?
Now, device level full AI, that's a bit of a trickier question because I've been looking into fully agentic systems. I think fully agentic systems as an industry is probably 10 years out, probably. It depends where the costs of models and the accuracy of models get to, right? So I can't put a firm timeline to that. But there is some capability in it today.
on a light level that's probably put a bit more work on you guys. And then from the factory, we're working on that in the next couple of years.

Emily Malin 1:06:19
Perfect. Okay. I think that was it. I think we wrapped up not too bad. Only 5 minutes over. So we lost a lot, but a lot stuck through. So we're good. Awesome. Okay, guys, I really appreciate it. Yeah. Yeah. And again, make sure you scroll up and register for our next one. And yeah, let us know what you thought a survey will be coming your way.

Jayson Bursill 1:06:21
Mhm.
Yeah, yeah.
Yeah, well, thank you so much, everybody. Yeah.

Emily Malin 1:06:41
And if you have any other, you know, thoughts, ideas, feedback, always good. Thank you, Jason.

Jayson Bursill 1:06:47
Yeah, thanks so much.

Emily Malin 1:06:49
Bye everyone!
Okay, I'm gonna close down the meeting now.

Emily Malin stopped transcription