vqa-nwxg-jkt (2024-02-28 16:05 GMT+1) - Transcript

Attendees

Aarsh, Alexis Guijarro, Daniel Agar, Gael Gatera, Jaeyoung Lim, Khaled Elmadawi, Kimberly McGuire, Kimberly McGuire's Presentation, Mayank Joneja, nerellasureshkumar 7, Peter van der Perk (NXP), Rafı S.M, Ramon Roche, Rikin Ramachandran, Ryan Friedman, Samuel Chew, sayed Ayman habib, Tully Foote, Youssef Madany

Bullet-point Summary Meeting

- Meeting starts. Announcement about Diversity scholarship and the start of the indoor navigation subcommittee. Also Aarsh shared that ROScon France was announced to be held in June.
- Kimberly McGuire provides background information on REPS and Rep 147 (ROS Enhancement Proposal 147), mentioning its different types and a previous discussion meeting in August.
- Mentions the need for improvement of Rep 147 and proposes discussing specific topics during the meeting.
- Tully Foote shares a bit of history regarding Rep 147, mentioning funding to visit Zurich and discuss with Lorenz from Willow Garage.
- Discusses exploring the Dronecode project and the need to think about less optimized but more generic solutions.
- Describes Rep 147 as an effort to define standard messages agnostic to specific types of vehicles, focusing on making messages vehicle-agnostic.
- Emphasizes the importance of standard data types to simplify data transfer between systems, making it easier to integrate different components.
- Suggests focusing on agreed-upon pieces of Rep 147 to solidify standards incrementally rather than trying to cover all aspects at once.
- Discusses the flexibility of Rep 147, allowing for optimization based on specific use cases while still maintaining compatibility with standard interfaces.
- Kimberly McGuire clarifies that Rep 147 is more of an informational guideline than a strict standard, emphasizing the need for community agreement on certain aspects.
- Mayank Joneja suggests prioritizing the most common mapping messages or functionalities in Rep 147 to ensure completeness for common use cases.
- Ryan Friedman proposes adding a behavior section to Rep 147 to define specific use cases and behaviors, facilitating automation of compliance checking.
- Ramon Roche highlights challenges with MAVlink governance and suggests starting from MAVlink to identify what works before incorporating it into Rep 147.
- Raises concerns about the limitations and challenges autopilot developers face with MAVlink, especially regarding custom message implementations and behavior definitions.
- Discusses the need to balance freedom for developers with the need for standardization to ensure interoperability and maintainability.
- Ryan Friedman discusses the advantages of Rep 147 in terms of ABI compatibility and the ability to break API for necessary changes, contrasting with MAVlink's lack of API flexibility.
- Ramon Roche expresses skepticism about MAVlink as a long-term solution and discusses the challenges of managing dependencies on third-party message sets.

- Tully Foote suggests maintaining control over dependencies by keeping forks or clones of necessary repositories to ensure compatibility with product timelines.
- Confirms that companies are using ROS for product development and recommends managing dependencies carefully to maintain control over the software stack.
- Jaeyoung Lim proposes starting with a simple minimum application or message, such as defining a basic vehicle status message, before expanding Rep 147 further.
- Discussion revolves around incremental steps in developing a standard message set for ROS (Robot Operating System).
- Emphasis on solving small, real problems rather than tackling everything at once.
- Proposed approach involves creating Rep (ROS Enhancement Proposal) documents for standardization.
- Consideration for maintaining practicality and avoiding overly detailed specifications hindering progress.
- Proposal to slim down Rep 147 and iterate through incremental changes.
- Proposal to create compliance test suite for certifying autopilots against the standard.
- Recognition of the need for involvement of MAVlink developers in defining message sets and use cases.
- Agreement to continue discussions on Discourse or Discord channels.
- Next meeting is scheduled for March 13th.

Chat Content

- Tully Foote: https://roscon.fr/
- Daniel Agar: you build and deploy it atomically
- Alexis Guijarro: Thanks, have a good one!

Transcript

This editable transcript was computer generated and might contain errors. People can also change the text after it was created.

Kimberly McGuire: Yeah, let's get started. The recording has started so welcome everybody at the errorable text Community working group for February 28th. It's going to be A good old discussion meeting we used to have only a few months ago until it kind of changed to a new format. You see we have a presentation meeting around this time, but we didn't really had a topic lined up and somebody had a post to discuss So we're going to have a nice discussion this time. So a little bit of the agenda I'm going to start with some announcements to share with you all then some background information about but Ross enhancement proposals probably can be enhanced by others as well in this meeting. I also kind of a summary of last meeting because we have a similar discussion meeting in August.

Kimberly McGuire: So just a bit of a summary on that's and I proposal for improvement of the rep 147 by Ryan and maybe On some topics we can see if we have time for the end of this meeting. All right. So first of the announcements so to few weeks ago, the first scholarship application of rochecon has opened up which is in odense Denmark. I've been asked to share for diversity. So that's why I'm also sharing with you guys. So if you have nobody that's disabled to apply for this. or what qualify for it, please share the application form of them. Also, I'm happy to say that the internal navigation subcommittee has kicked off.

Kimberly McGuire: My uncle is going to help out the facilitating. I don't know if you remember but I've Sunday let's say participation form for the arabletics to kind of see what the interests are and where we can kind of start ship topics or should committees it starts making some specific development goals. So there's now some one for internal navigation. My young has started a discourse threat on it. So if anybody has any experience in this, please respond to that this course threat. Yeah, and is there anybody else that would like to share? Anything at this moment for instance a Meetup or some other announcements? Now is the time.

Aarsh: So there is a Roscoe in France if anyone is interested visiting France.

Kimberly McGuire: Yeah, that's also a good one.

Aarsh: Yeah.

Kimberly McGuire: What date is it?

Aarsh: Just a second. I will share the website.

Aarsh: So it's on the to 20 June in a small but beautiful town of North. Yeah, that's it.

Kimberly McGuire: All right. Thanks. Everybody else wants to share anything else or is it safe for me to continue?

Mayank Joneja: but Roscoe in India videos are slowly trickling out. So mine is I'll share it if anybody's interested on why we use loss where we don't make robots. But yeah, there's a one I'm looking forward to or somebody who made a beagleone flapping Wing implementation as a part of a g-soc. So at the moment that videos out eye drop it on the agent of what extract as well on Discord.

Kimberly McGuire: All Might be able to add something for the flapping Wing MVP here. So that's something for that later. This to be my old lab a cup of I will just go ahead we'll be very very short introductions. I'm just hoping as people will actually read a button rep is it's like I said before it's like a rose enhancement proposal and it's kind of like a documents to how do you describe it's kind of either to propose some kind of Standards. Some guidelines are some kind of form of implementation. That's a work for different topics within the Roche infrastructure. very shortly since the full description is of course, you can find it in the link right here, but I guess what I thought was interesting to know that there are actually different types of rules enhancement proposals.

Kimberly McGuire: They have the standards informational rep and process At least to that, kind of the backgrounds. Let's say hi everyone. let's say before I go into the rep one four seven, which is specific for aerial politics and kind of the workflow is that So many writes a draft for it that can be let's say finalized together with the input of the community that can either be accepted and can be final or it will stay in draft until nothing happens and That's apparently the workflow. So that is all what I shamefully copy paste from this web URL right there. Just have a bit of a background information. So we had our last meeting on messaging standard was in August 16th.

00:05:00

Kimberly McGuire: 2023 and you kind of came to the cushions that the current state of the rep one four seven doesn't really fully aligned it needs of variabletics. it's good to kind of consider existing standards a real world use cases. It's also kind of the advice that we had to really Define semic semantic data types and behavior and also kind of really try to be even more specific on what kind of messages and the content that it needs to be done. Also advice I think was much fertility is to Start small have an incremental build or Marathon an organic growth of the drop document that the discussion was one hour long. So probably these couple of conclusions doesn't really capture the whole thing. But luckily we have to recordings and transcripts for anybody to miss that's meeting On the figure Ryan you missed the meeting, right? Yeah, okay. Yeah.

Ryan Friedman: Yeah, I was sick. I'm sad but I watched the recording.

Kimberly McGuire: Yeah, So that's why we have recordings and happy that you're here now.

Kimberly McGuire: So let's kind of look against the rep 147 147. That's the standard interface for Ariel Vehicles, which was started in 2016 as drafts. And then it The last big update was in 2018, but unfortunately still it was given in the first state is in 2022. So a little bit unclear exactly. What's that state is kind of practically means is it still a draft or let's say it's not rejected, but it's more of like indicating that it's a bit stale. I guess right.

Kimberly McGuire: There were some tiny updates since that Ryan has implemented mostly and the state is still in the first status. Until this moment and that's a bit too bad. So, okay. I'm going to give this kind of the word to Ryan for now because this is your slides that you've proposed with all the changes that you want for so. You could. spinos

Ryan Friedman: Yeah, So basically over the past bunch of months. I've tried using rev147 as is an implementing it on order pilot. We got a bunch of stuff merch but a couple of the things just don't make sense that we did. So I wanted to bring those up as potential. what we can do to improve rep 147 if that's correct the direction to go. So the first one was the sensory messages, there's that sensor messages IMU. So there's a field in IMU to report a quaternion orientation and it doesn't really say which frame ID you could put that in but Can do it Ned because There's a way to do that. But the thing is that I need to actually report orientation. That's more of An autopilot, it's called aarsh system that does that.

Ryan Friedman: So it's a bit confusing to try to populate this for Save picks hawk or not a pick sockets, a px. So one X I can't remember all names one of the modern ones with three amus that sensor messages I need doesn't really make sense anymore. When you try populating that with the data is that data that the inu's actually give so that's one thing is that I got a pull request and all our pilot devs are like why is that and you have this information and you message from Russ have this information in and then if you try sending three different Imus, then you end up with a lot of duplicated data for an eight-har system. That's not you. And then another one of the biggest problems that I ran into is the covarian. It's a race for some of the messages or mandatory, even though you can just set them. A lot of systems especially embedded don't want to pass on that kind of variances or don't have them available.

Ryan Friedman: Or a slam system like cartographer doesn't provide covariances for its odometry that it gives you. Google cryptographer by the way, so it's a little bit unclear on why roths decided to Mandate covariances of hardly anyone uses them and what this does is basically explode the bandwidth that we use between a companion computer and the embedded flight controller. so in a perfect world, we could just change the message definitions to make the covariances optional, but that's a breaking change and there's nothing to do about humble. And that's what both PX4 and art of pilot are using for the Ross version right now and that's going to be the same for a while.

00:10:00

Ryan Friedman: Especially with embedded companion computers people are going to stay on Ubuntu 22 for a long time until Nvidia can support a newer version and they just released but Ubuntu 22 is not even officially supported and now Ross is about to release about to 24. So people are gonna be on humble for a while and I think we need to be conscious of the fact that a lot of companies are going to be humble too for a while because of this reason so we need to find a way basically To not have mandatory covarianism messages for humble, and I'm not sure how to do that because it's breaking API change.

Ryan Friedman: So I guess if anyone has feedback on those two, we could talk about those before we move on. Or I could just go through the list and then we could pick what we want to discuss.

Ramon Roche: Sorry, I got a question how we already discussed the…

Ryan Friedman: Yeah.

Ramon Roche: if the viability of the project and where are the goals behind Reviving red one four seven? And what are we supposed to?

Ramon Roche: What is the common layer between the autopilot flight Stacks here? And how can we help? I don't know that we've discussed this before before and…

Ryan Friedman: Yeah.

Ramon Roche: Before we jump into specific details of Route 1147 what we want to improve we like to discuss governance and…

Ryan Friedman: exactly Yeah,…

Ramon Roche: a lot of other things first

Ryan Friedman: I think I could cover that now. Yeah, that's cool. Jail said a question if you want to dig in a specifics or we could wait on that Jay.

Ryan Friedman: Yeah, so yeah, basically why rep147 …

Ramon Roche: It's crazy.

Ryan Friedman: so the goal of using rep147. For me was to try to find To integrate some of the common autopilots in a way that you can write agnostic Ross autonomy code.

Ryan Friedman: That's the goal and rep 147 is a means to an end because it's the closest thing. to something standardized other than mavlink

Ryan Friedman:

Ryan Friedman: Some of the key reasons for rep147 to me or the tooling integration. So foxglove has support for an absent fix. So the tools that Ross developers are commonly already support a lot of the messages and rep 147 and the alternative.

Ryan Friedman: For obviously these MAVlink. That's hugely popular. already supported by all the flight controllers

Ryan Friedman: But the problem for mavlink, is that the Ross tooling for it isn't there? And it's a big ask if I submit a foxglove ticket right to go ask to support the link position packet. Not going to do that.

Ryan Friedman: There's very limited resources and tooling Arvest same thing. there's no core developers maintaining artists necessarily. so by using rap 147, it was really how do we integrate with all the existing Ross tooling and The next things harder on the autopilot because reporter 47 doesn't exist for PX4. that's something that the main thing for me was tooling but the other thing was also that mavling was designed as a communication between a grad station And the date drawn an error or very limited bandwidth. And with new flight controllers these days having ethernet and what doesn't actually that much of a concern. between a companion computer and the autopilot so the optimizations made for math have significant costs like Certain messages assume certain coordinate systems that you want.

Ryan Friedman: And it's not exactly clear in the documentation on which ones they are or sometimes. They're autopilot specific just A bit scary trying to write autopilot agnostic code with unknown frame IDs and I just saw a bug in order to Pilot two days ago. About how a specific combination of the ground station and autopilot would result in terrain following using the wrong vertical datum. so I think there's a lot of assumptions made that that can be cleared up in mavling, but Ross is nice because all the messages have a frame ID.

00:15:00

Ryan Friedman: and so that was another key Advantage for me is Because there's rep one of 103 105 and…

Tully Foote: I can give a little.

Ryan Friedman: a frame ID. It's much more difficult to mix up your datums when you follow messages with a header.

Kimberly McGuire: Okay.

Ryan Friedman: So for me, That's the reasons but I think there's obviously other solutions to this problem than rep 147. So I'm interested to hear what people think on. Is rep147 even the right way to go.

Ramon Roche: Hello, Jay. You want to go ahead?

Jaeyoung Lim: Through this I think for me at least red one for seven is met it's quite specific and quite broad at the same time. And if we're talking about sort of this autopilot agnostic. this implies that we are looking for compatibility between Autopilots,…

Ramon Roche: not

Jaeyoung Lim: So if you follow this rep one for seven or improved version of it, you will expect also behavioral compatibility on the autopilot. For example, if you Listen to a certain topic you expect that to be the global position estimate of fmu, for example, or if you command the body rates at point, it would execute it. Except I think at least for me.

Jaeyoung Lim: the current standard doesn't Define Optics of which topics it's mostly message types and formats, which I think it's probably part of the standard we need but it's lacking a lot of this kind of what is the behavior that we want to what we would expect if it is a come compatible autopilot to this rep. standard and the second or I think I'm a bit missing. Is that if we try to make a standard that's autopilot agnostic and sort of what audience are looking for and how would we actually check? If it's fully compatible or the coverage or at least this message is supported to this do we leave it?

Jaeyoung Lim: to each autopilot implementation or delete with the Aerial Work will be responsible for Sort of keeping a list of compatible autopilots. How would this be maintained? Once we establish a standard. I think these two things are quite unclear for me from the current Red. And if we actually want a compatible order be able to write autopilot diagnostic code by being complied to this rep, I think. This kind of Behavioral definitions should be included. Otherwise, I don't think we could make autopilots competitive.

Ryan Friedman: Absolutely has a great points. one of the things I've done actually and…

Ramon Roche: All right.

Ryan Friedman: this was the first time I was able to actually write a whole Behavior test was Once art pilot had enough of the messages. I was able to send global because I think it's a global position command. basically just a waypoint a loiter. So an arm command to take off command and a lawyerweight plant and…

Ramon Roche: Okay.

Ryan Friedman: then you could write tests and check the behavior that the Drone eventually got to the loader wave point and then check that it's turned command was going In a left turn rather than a right turn because we asked for a left turn. And so I built up this coal contest and so

Ryan Friedman: I think One path to standardization is to write is that the organization could maintain a test suite for certifying something according to rep 147 and we distributed a series of tests that behaviorally regardless of which simulator you have you send these messages and over some time you expect this behavior. And that assumes that you can Implement enough for the standard to do things like that.

Kimberly McGuire: Seems that the Tully you have.

Ramon Roche: totally

Kimberly McGuire: Your hands up going.

Tully Foote: yeah, so I want to suggest that building up to a full standard would be awesome. But I suggest I think

Tully Foote: it was mentioned earlier that grab forty seven is both very Broad and very specific. and I think that if we try to take this to be fully Broad and a full standard all up front, we get bogged down and sort of basically never get there. What I suggest is we do sort of incremental processes along the way I think.

00:20:00

Tully Foote: a little bit of History rep 147 I got some funding was able to fly out to Zurich and spend some time hanging out with. Lorenz and it's coming from Willow garage. I haven't done the pr too and looking at the Drone code project and seeing the different autopilots the different. things that I was looking at mablink, and I wanted to Look forward a little bit to the point where we start thinking about as Ryan mentioned less optimized things but a little bit more generic. And the rep 147 was basically the sit down what can I take from the general robotics community and bring it there the aerial vehicles and how can we clearly Define those?

Tully Foote: Things because I want to provide what the vision for this was that we say, okay, here's a bunch of standard messages. How can we make them be truly vehicle agnostic and not just vehicle agnostic. It't these things don't there's no encoding in here of whether it's a quadcopter or a fixed Wing. I mean, it doesn't even know that it's a

Tully Foote: these things could be generalized to work on the asterby on the ISS because we use the same messages as we use for ground vehicles and some ground vehicles have multiple Heights. they're limited between six and eight inches

Tully Foote: and thinking about these in the abstraction and providing these standard data types between them allows us to Make it it's just an engineering effort to get data from A to B. So you're mentioning we don't have the topics to find. But the first step is as long as we have the same data types. It's like somebody has to set up a launch file that remaps the topic. Or even if you're not using the Transport layer you just have to write a little glue layer that says equals c between system a and system and it's easy to put things together. So I would encourage us not to try and go creep this up and instead take little.

Tully Foote: Little Steps along the way and I think one of the challenges rep 147 became very Broad and I would actually encourage us to

Tully Foote: potentially just take the pieces that we agree on. and solidify those and then leave the ones that we can agree on out and do that in another iteration because the value of having The battery message or something if we can all agree on that just having that would be valuable and stepping forward from there. in the ways that the

Tully Foote: But we agree on this let's ratify that and move forward. The other thing I want to go back to was Some of the topics earlier especially the optimization ones. I think that If we don't make this a standard. It's fine. If you have an application where you can't afford to send the full covariance Matrix, you can do something a little bit different and then you can just document I'm doing all the standards here. But this one I need an optimize one. I'm going to Use an optimized one and maybe at some later point in time. I will take the optimized one and expose it as the standard one.

Tully Foote: On the companion computer. So I'll just up sample it in you can do things like, maybe the covariance doesn't change at all on your system. So you can just publish the raw data and then have some node at some point later in the pipeline that will take the low level one that you've implemented the specific to your system and up sample it to the generic one. And that's something that we don't necessarily have to.

Tully Foote: Don't think of this as saying that these are constraining you preventing you from using other data types. I think that's the big sort of power of thinking about using things more in Ross where these are standard interfaces that you use for compatibility, but it doesn't prevent you from using other ones.

00:25:00

Kimberly McGuire: Yeah, I guess as I understood for another indeed that the current rep it's an information one instead of a standards that are process one, which means is that it's not as fixed or it's not meant that everybody should follow that exactly it's more of a general guideline and even at least in the explanation on that's the first page of the rep that's kind of explains not all everybody in the community should agree, but there should be at least something that's Most maybe can agree on. There were some people that had their hands of my oncuber first.

Mayank Joneja: Hey, I just had a question. I don't know if this is already factored into how the rep one 47 is written down. But as a strategy in terms of scoping out what we Implement what it makes sense to First Target the most common mappling messages or marbling based functionalities considering it. As far as I know it is the de facto standard between let's say at least for autopilot and artify it and then go from there so that if you're doing a completeness check for the most common use cases would it make sense to take the subset of marveling messages which are there and make sure that 147 in capsulates those use cases in some fashion. Maybe not those exact message types, but whatever messages we offer those are included or is that something which is already on the document? I'm not that familiar with that as it sensitive.

Ryan Friedman: so It doesn't from a message perspective, but it doesn't do it from our behaviors perspective. And I think that's the part that's missing. I think it'd be really helpful from the sounds of it to add a behavior section to say.

Mayank Joneja: Okay.

Ryan Friedman: I want to go launch a plane fly to away point in orbit it in this direction. What are the messages? I to do that. And which Fields do I need to set to Nan or whatever and write that as here the list of messages we can agree on that and then the next thing would be to write a test to check the behavior and then we can start to automate the compliance checking but doing it from a behavior perspective and start from some really basic use cases like Waypoint control my idea way to approach this and just to make sure that

Kimberly McGuire: or at least aiming for Yeah.

Ryan Friedman: Yeah.

Kimberly McGuire: Ramon you had Your hands up.

Ramon Roche: Hey, so two things I think mappling has Been a tremendous learning experience for everyone involved. First of all, I think that if we wanted to optimize for Rosa developers to interface with autopilots, I think we really need to look into MAVlink and start from MAVlink and not from Brad poem for 7. and take out what we doesn't work and see if we can fix that but in the other hand the governance for maplin hasn't been great, but it's useful and it's still going so no critics there, but from an autoplay perspective and this is

Ramon Roche: my personal perspective and I see Daniel agars here. So maybe he can chime in but it's been severely limiting for an autopilot in for the developers of out about it's to have to deal with MAVlink. Let's say you want to introduce any future that it's not implemented in a different flight stack. And you need to introduce a new message. So you have to go through the whole cycle and then you are blocked on the end of MAVlink wanting to make this Upstream message set instead of having it defined as an auto body specific thing. For your use case, so that's one it becomes. awkward when you need to define the behavior for a future that the other Auto Part doesn't have or is still implementing and then you have to sort of work together and work backwards from there to try to

Ramon Roche: have same interface and that's being causing a lot of issues. So if you look into mappling and implementing projects sort of like a map SDK or qround control your grand control being the historic one. Has a lot of spaghetti code everywhere and it's a very hard to maintain because of those differences between implementations that were not catched on the standard level. And then on the map SDK level there was an effort to try to correct that and it's starting great, but it's becoming a little bit more involved. So

00:30:00

Ramon Roche: In order to have a successful standard in my opinion. we need to make sure we think about those things and try to optimize for developers. But also make sure that the autopilots are not in a position where they need. the rest of the community to be able to move forward in any given Direction which would severely limiting for any flight stack that is implementing this standard and then Also from the autopilot perspective. What would be the reason for one Ros developer to choose wine flies like versus the other of the interface is the same so it would be basically a black box for everyone and you wouldn't really need to deal with the idlepod at that point. So those are my thoughts not necessarily answers or anything. So you just wanted to throw it out there.

Kimberly McGuire: If I could add to the fact it's like, the muffling. there was a lot of trouble with versioning and things like that it was kind of one of the main reasons why at least from bitcoin's point of view. They haven't implemented a mufflink at all. They actually have implemented something For good and bad. I would say that's fully Dynamic on what's messages are defined within the firmware. Which means is that there's versioning whatsoever. Whatever is defined as a message in the firmware. that is the one that's being communicated to all the ground stations and versus version. Which means is that actually for developers? That's great. But mesos also they have maybe a little bit too much freedom.

Kimberly McGuire: so, yeah, so maybe that is something that's And yeah, we can also kind of think about it's like how much Freedom you want to give developer that or maybe there's some messages that you can say this is an aerate that you can just put information and mayank. Yeah.

Mayank Joneja: To that point about future proofing or custom message Styles. I wanted to ask click fundamentally Ross message types are kind of account for that future proofing or scaling where in their arbitrarily composable as long as you have support for serialization and DC realization of the Primitive types, right? if we treat what comes out of their 147 or this message standard as the supported messages, but we don't block off this custom messages that any autopilot vendor could develop and as long as the lower layer supported we are not very limiting anybody from having their own custom implementation as long as the handle both ends of the interface, So, I mean in that sense isn't lost messages fundamentally different from avalink and already sort of allowing us to do that customization in the first place.

Kimberly McGuire: Yes, I don't know …

Ramon Roche: My book already has this you can do common ad messages to the comment set or…

Kimberly McGuire: if go for it.

Ramon Roche: messages to a specific autopilot set. So we already have that on MAVlink.

Ramon Roche: Yeah, I think the problem is mainly when something is generic enough that we need to move it from a custom autopilot to actual common message set. Where we need to actually work together to standardize that interface.

Ryan Friedman: Yeah, the custom mode stuff. I think is a really great example, that's obviously the right way to do things using guided mode for doing all sorts of different stuff is just really hard and hard to Pilot and I've been struggling with the dev team on actually implementing different behaviors with a single node, and I think The Explorer proposed this really awesome way of handling this. And it's just a ton of work to implement an order Pilots, which doesn't exist yet. But And I think that's a really good example of we found the right way to do things. It's just harder in one place stacked than the other so they can't support it yet. And that makes it hard right if I want to base a autopilot agnostic stuff, I can't use the custom nodes interface even though it's the right thing to do right now. So as a developer that puts me in a tough position. To try to choose one of the other.

Ryan Friedman: So I wanted to actually bring up something. I'm interesting that hasn't really been talked about as ABI compiled ability. So from what I've seen there's three options right now one is rep. 147. Which is ABI compatible within a roster release and you can break API once a year if you really need to so there's ability to change message definitions. If something is obviously wrong and on the other hand in math link. From what I can tell there's no way to break API and message definitions ever. Even if it was a mistake and sometimes those mistakes have to break API to fix them. And doesn't seem like Madeline has a process to break Avi ever. So those mistakes get propagated for eternity until Madeline 2 came out, but from what I understand that didn't actually change the message definitions. so I think Ross has a unique advantage.

00:35:00

Ryan Friedman: and that sense that there already is an Avi breaking process and I'm not saying that it's a good thing to break It's really knowing what it does change but oftentimes they're really good reasons to do that. And then the other flip side of this is your message is which are an implementation detail for PX4 and those can change whenever the Explorer wants. But the trouble I would have of using the PX4 messages and a commercial application and I'm not sure how companies are dealing with. This is what if the PX4 Dev team tomorrow decides change ABI on their messages. Does that break companies developing software and then they're blocked on upgrading PX4 for bug fix because they changed the message definitions that So I know that there's currently a lot of tooling and effort to support using your messages in Ross. but they're

Ryan Friedman: doesn't seem to be any promise of BI stability. So I was wondering how some more support commercial product built on that. Maybe someone could elaborate on how that's currently handled. And if that's not intended for companies to be using your messages on the raw stacks.

Ramon Roche: Them able to respond to that answer to that question. Sorry.

Ramon Roche: For my opinion. I don't think Europe is a long-term solution.

Ramon Roche: But Daniel says you build and deployed atomically.

Tully Foote: yeah, I think in terms of that also be my suggestion you want to be able to sync to whatever the PX4 Dev team is, but if it's in your build and it's something that you're gonna be relying on you should have a fork or a clone that you're making sure to keep in sync so that If your timeline and the other other developers timelines are different. you should have your snapshot and make sure to build it yourself. the

Tully Foote: No, any companies using Rossi this all the time where in general you're using the mainline open source one for your development cycles and testing but as soon as you want to go to a product and ship it. You should really be snapshotting and building everything yourself so that you can make sure it's there and if there's a critical bug. In something that you have to fix right now you have that ability. Or more commonly. there's a bug that fixes something but that's not a vulnerability for you. But you don't want to incur the cost to potentially have upgrading as well and you'll do a alternative work around which might be like I'm just gonna all firewall that forward or I do something else. but if you're building a product yourself you want to be able to control that Pipeline and

Tully Foote: Monitor changes that come in and pick them. Generally you want to keep up but you want to know that you have the ability to choose not to and do an alternative solution.

Ryan Friedman: So is it recommended to been a product with Ross right now or not?

Ramon Roche: Yes, it's recommended. And in fact,…

Ryan Friedman: Okay.

Ramon Roche: a lot of companies are doing that right now.

Ryan Friedman: Okay, I had a comment

Jaeyoung Lim: And also maybe then we should not treat this as either a standard or a full-blown API, but try to have a very simple minimum application that we think is Maybe it can be even just one message. that we can try to find useful specific to area robots and then try to establish this and then we can try to think about the next steps after we To this…

Ramon Roche: Okay.

Jaeyoung Lim: because I think anyway most stuff can be done dealt with standard rough messages in general if we don't think about compatibility.

Jaeyoung Lim: and maybe there is something that's missing for the aerial robots for example, vehicle status and then maybe we can have a common message definition and reports like basic Vehicles status that we actually need for auto maybe when we talk to different autopilots that it already may report something like maybe the arms status or the autopilot ID maybe or what autopilot it's running for example, and then we can try to establish this as a very minimum thing that solves something in the Robotics domain and then we can try to think about the next steps after. Try to do this.

00:40:00

Ramon Roche: Yeah, that would go off as far as say that maybe look at the mapping common message set and see if there's anything that we can bring over from there.

Ryan Friedman: Do you think a good example would be like hey, I really want Fox Club to show my vehicle on the map. So can we establish navasat fix is the right way to do this or if we have issues with it work with foxglove and our Vis to get a different message supported because what you were thinking and do a one message at a time with some certain tools and behaviors that I want to see where my vehicles on the map. But by the way Fox Club doesn't show the orientation so you don't know which way it's facing. the map panel

Jaeyoung Lim: But maybe yeah, but I was more talking about not trying to do this kind of all or nothing but try to solve one very small problem that we like a real problem. We have an area Robotics and then think about I guess the grand dream is I can just write an obstacle avoidance framework and it works for every flying robot in the world. But I think that's something that will take time, but we can try to make some very incremental baby steps first. That's also real problem with this rep that makes it actually useful and then I think more and more people will actually see value in this rep that encourages people to adapt it more and more.

Tully Foote: just yeah,...

Kimberly McGuire: Right, perhaps...

Tully Foote: I mean that's

Kimberly McGuire: then it's robots. let's say then we have to even have steak take even a step back and it's like, not even to looking at the rep but just kind of determining what is the main issue that we're trying to solve here and perhaps Start some kind of discussion on this course to kind of figure that one out because yeah. Other people have a lot of opinions on that. Of course, I hope but really you were saying.

Tully Foote: Yeah. Yeah, I was gonna say I think that taking the small incremental steps is going to be a much more productive if we can get the little things out of the way that are easy and can make them happen. That's great. it's say naps that fix that's moderated mature and Ross and it does the things that you want to do and it meets the use case of being able to draw it in RVs or Fox Club. if we need to say okay that's there and I think what I want to point out is that the reason that rep 147 is in deferred. is because we don't to standardize messages in Ross. What we really want to have is one and preferably. one full implementation and preferably to

Tully Foote: so we can actually be navsat fix works for both PS4 and already pilot we expect there's nothing in there that's specifically if PX4 or if our new pilot which will block a third party one coming in. we try not to embed those and just say okay, this is standard. we put that into a sensor messages or common messages somewhere and we say we've got two reference in limitations. It's fine. We believe this will be solid and won't need to change down the line. a lot of rep 147 was basically me sitting down and being like if I go back to the underlying physics, what do I want to represent which it's coming from a very different direction than I think a lot of what modeling questions like I need to set the setting on this motor.

Tully Foote: And I guess it's sort of the different approach but a lot of the reason it's ended up in deferred is because I looked out I tried to take everything that was. Potentially representable and abstracted away and I'm not day to day using drones. I don't have time to push on this hard and actually do an implementation and what I'd like to see is basically find the things that are Implement them like Ryan whatever you see is valuable you try it out no sat fixed. Get somebody else. To try out it on their system.

And once you get a couple of you to agree, hey naps, this is working fine for these use case. Let's just ratify that. and then we'll go forward and maybe the battery state or maybe the

00:45:00

Tully Foote: What's the next one and basically get to get one full implementation or two and then because we really want to avoid the design by committee process where I sit in a room I come up with this very great abstract standard and then Ryan comes back and he's like, you added covariances here and that overflows my microcontroller buffer where I'm actually running this and I can't use it.

Tully Foote: and that's the challenge the spec is very very generic and sometimes we need to make sure it's practical too.

Kimberly McGuire: So yeah, maybe we should try to aim. it's to be closer to standards, but cutting away a lot of let's say the ones that are now very detailed in there that's perps kind of like stopping the progress. Of actually us implementing it because it's too specific for it.

Kimberly McGuire: So we have about 10 minutes left and we actually haven't really talked about Ryan's proposal. Of what's to improve? But apparently we also need to really talk about what is the going to be the purpose of rep and what needs to be in the first place? So

Kimberly McGuire: So Ryan, do you still want to kind of pick out one of these topics or shall we perhaps Use these last 10 minutes to maybe have a bit more of practical.

Kimberly McGuire: Talk about selective practical ability to actually take this to the next level to actually get something done here.

Ryan Friedman: I think I'm happy to follow up and discourse on those specific items. I really appreciate the people that we have right now or we have a lot of good subject matter experts. I think it might be more useful when we're together to talk about bigger plans. So I guess whatever Ramon had to say might probably be important.

Ramon Roche: Thanks, Ryan. basically as you just want to see we could agree at least on…

Ryan Friedman: Yeah.

Ramon Roche: what would be the goal of doing this and in my opinion, what will be really value is to list from the PX4 perspective?

Ramon Roche: Optimize for Roz developers one into talk to flight stacks and see if there's any thing that we could do to help them and…

Ryan Friedman: Great.

Ramon Roche: to help the Ross package developers to be able to integrate with flightstacks easily and in more the generic things. I see it is looking like we have a lot of things the final already in MAVlink. And this in the comments said so I would highly suggest that we look into the messages. I know that My link is a lot of things. So if we ripped everything from mapling and we just keep the messages. Maybe we have a path forward there. but yeah, I'm happy to discuss at least we should Define the goal for this effort and if we can agree on that, I think we can start moving slowly Tully you suggesting.

Ramon Roche: Because I do agree that trying to fight on the whole thing. It might not work out. But if we start small we might be able to get far.

Kimberly McGuire: I guess from a practicality point of view.

Tully Foote: Yeah plus.

Kimberly McGuire: I'm wondering in the current state, it's okay for us to give a full total new.

Kimberly McGuire: Let's say rep147. Let's say that we're a proposing completely slim down version of that or is it better to maybe propose something new?

Tully Foote: I would suggest that if we can agree on one or two messages and rep 147. Go ahead. Delete the rest of them and make that rep 147 and say okay, we've got two implementations. We've confirmed these work. Go ahead and then we make rep 140. X

Tully Foote: and say, okay we deleted all these things over here. We want to come back. We think we can optimize this one. I mean we could write a new one. the number doesn't really matter. they can move rep 147 to withdrawn and make a new one. but

00:50:00

Ryan Friedman: So…

Kimberly McGuire: Because it's nice to work.

Ryan Friedman: what if we start you I was just asking …

Kimberly McGuire: Sorry.

Ryan Friedman: if we want to write we do this process of incremental changes. What do you expect? We have a rep.

Ryan Friedman: What's going to be not one fork? Because this rail taken but let's just call it 199. We have a rep 199 and…

Tully Foote: Yeah.

Ryan Friedman: We get implementations of net we choose an outside fix to solve the use case if I want my thing to work with foxglove and artists and we compare it to math link and make sure that has all the data that Maverick exposes or modify an apps that fix and make a new one. then we get that message approved agree that At least two flight stocks are going to go implement this a couple, months later. We come back and say At least two flight Stacks support this message. Do we make an amendment to the rep get a list stakeholders to approve the pr and then merge that one change and then just keep it in deferred until we have, enough of the messages added. Would that be the process?

Tully Foote: So I'd actually roughly that I mean the nice thing informational reps can be evolved more in there. we can revolve them in place. If we get the right buy-in from people, I think that in this case though, but we can make it active with just one or two. It's like okay, here we are and then there's a pull request that'll extend it. If we're going to go into something completely different or new messages we can make new ones. you can make a parallel numbers are approximately free. So yeah, is it more helpful for somebody if we put them in the same document? Yeah.

Tully Foote: To read if it makes sense for them to be side by side in the same document this information. We should add it there. if it's going to be something that we want to sort of Set in stone and move forward. Because a lot of this is standardizing messages. Doesn't have to go through the Reps if we just have a we sort of push it through this more public process for the standard messages to slow it down and make sure that we don't.

Tully Foote: break things because of the

Tully Foote: As you say if we push it into umble, it's in Humble and it's gonna stay there basically. And we don't want to in the core packages. The other approach is to actually iterate quickly in a little temporary package Test your implementation of that and then once that's valid Move it into the core because a lot of the gating and the slowness of the deployment is from the reviews and deployment side. Not from the rep.

Ryan Friedman: So you're saying attempting to revise napstat fix? Would be very difficult. but if we just had a revenue 199 / navs.fix temporarily that we could use and humble and break API when we want that might be a

Tully Foote: Yeah, and then you have a migration path which says in Iron it'll go to this and we recommend all people deploy. This one said script that'll be Acts to capital X or something. Or at least have some sort of migration path, but yeah, I think a rep specific package might make sense where it's like this will land in Rolling after Jazzy's fort. And everyone after that has to do the migration, but as of Jazzy is static.

Ryan Friedman: Yeah, I mean, I'm fine with that. I still do want to support humble. I'd be curious from Ramon's perspective.

Tully Foote: Yeah.

Ryan Friedman: You probably work with a lot of different people using Ross. It maybe have some idea but from the developer survey the most used Ross Distribution right now in Ross 2 is humble by a wide margin. And I'm not sure when that's going to change. But maybe what people are using. I know that the Explorers based on humble, so doesn't even work on iron or rolling our pilot doesn't.

Ramon Roche: I don't know.

Ryan Friedman: We don't have enough developers to maintain that but yeah. It is yeah.

Ramon Roche: Wait, we are at this point operating you've been to dependency as we update RAW support. So right now we're supporting whatever humble supports. yeah, so we're trying to push people towards the latest you want to always. It's just hard to keep up. Just the tooling the CI and…

Ryan Friedman: definitely

Ramon Roche: all that you have to keep updating. I mean, it's not that hard. It's just you need someone to keep to be on top of that.

00:55:00

Kimberly McGuire: So we're hitting.

Ryan Friedman: What do we? I was just curious. Just one more question.

Kimberly McGuire: Yeah go. Yeah.

Ryan Friedman: All right, can we do thumbs down on making a compliance test Suite that Allows you to certify an autopilot against the standard. Do we find Value in that?

Ramon Roche: I mean I mean if we are gonna be doing this I would say yes, so that cross package developers can attest that their interface Works. They need to have the test Suite to be able to develop. So if we're gonna make a standard, I think we should have as test Suite to not certify self validate that you can do this.

Ryan Friedman: Yeah, I mean it's not an official certification, but how do you know which Maverick messages work on which Good luck, right That's a hard question to answer. it'd be great. If you could just go look at a Jenkins job and see green check boxes on this Madeline message works on a solid pilot.

Ramon Roche: Yes, but I think that's more partly followed the government structure than the actual. Tech stack on MAVlink and that's something that could be fixed if that's something that is raised by both autopilots.

Ryan Friedman: We're out of time here. So. just

Kimberly McGuire: Yeah, that's what I was the so there's definitely a lot of questions, but I think you suggested. Ryan please make a discourse threats and share it on the Like and there was aerial and Discord so we can kind of get some input on that as well. So yeah, but I think we might need some more discussion meetings to talk about this. I don't eventually we can look at the Improvement proposal so I also had a lot of questions,…

Ryan Friedman: yeah. I'll make it this first.

Kimberly McGuire: but I don't think we're even close to even covering those so

Ryan Friedman: Node, I'd like to maybe just enumerator list of use cases and see if we can get some agreement on which one we want to tackle first and I'll just propose navsafix. And see what people think but if people want to do Waypoint people really fan of doing Waypoint control or rate control instead then so be it, I don't care but I just want to start on something easy. let's not try to do a mission yet.

Kimberly McGuire: No. Perhaps that's a like yeah,…

Ryan Friedman: Richard actually Yeah.

Kimberly McGuire: but yeah, that sounds like a great plan. If you can propose something or at least propose and…

Ryan Friedman: All Yeah, and…

Kimberly McGuire: list of options and if people don't like that, they get proposed their own so she…

Ryan Friedman: how do we involve the MAVlink developers and…

Kimberly McGuire: how it works.

Ryan Friedman: the governance model obviously have a huge amount of experience doing this. Is there a way we could involve them for doing this message definition and use cases to make sure that if there's anything that they learned from the Napoli messages that really needed to be fixed that we make sure we address that.

Ramon Roche: Yeah, we can bring in Hamas really. He's got limited time in working in open source, but he's available if we schedule the recording it.

Ryan Friedman: Yeah. Sounds good. Thanks.

Kimberly McGuire: All right, so I'll just try to I think we have came to the conclusion now. So let's keep all the discussion for this course or Discord. So the next meeting will be on the 13th of March. So that will be a developer meeting. and hopefully with more subcommittees to talk about like I said before internification, we kind of had the startup and the simulation hopefully we will have some kind of idea of how the facilitate that so

Kimberly McGuire: sooner or gum of more information about that. Yeah. So here's also to get the organization for the working group. Which get to welcome this Roche Ariel. Also, usually we have presentation meetings during this time. So if you would like to give a presentation about your specific particular, please let me know or Ramon or I will start emailing People based on the area of participation for it because some of you have said that you would like to give a presentation. I would like to thank you guys and also telling Ryan Ramon and everybody to that was able to during this interesting discussion, and we're probably going to have more about the specific.

Ryan Friedman: Yeah, really? Appreciate everyone for coming. I know a lot of you are busy and you know that I think we got to start somewhere and appreciate everyone being a fan of just wanting to move forward on autopilot agnostic code that works with multiple flight Stacks. I think it's going to be a big asset for the Ross can I needed to be able to do this?

01:00:00

Kimberly McGuire: All right. I'll just stop to recording no.

Meeting ended after 01:00:06 👋