Welcome to the “Review of Optical Module Management Interface Survey Results” Webex

- Please stay on mute  and turn off your video 
- If you have a question, type your question using the CHAT tool  and send it to “Everyone”
  - Moderator will address the questions during Q&A
- If you have Webex technical questions, contact Morgan Sackett at msackett@oiforum.com

Contribution	oif2020.277.00
Working Group	PLL WG – Optical

Title OIF Management Interface Survey Results

Source Ian Alderdice; Ciena

Date July 30, 2020

Abstract This contribution provides the results from the Management Interface Survey.

For additional information, contact:
OIF
5177 Brandin Ct, Fremont, CA 94538 USA
+1.510.492.4040 – info@oiforum.com

Optical Module Management Interface Survey Results

July 30, 2020

High Level Summary

- Survey press release: July 1, 2020
- Response Deadline: July 21, 2020

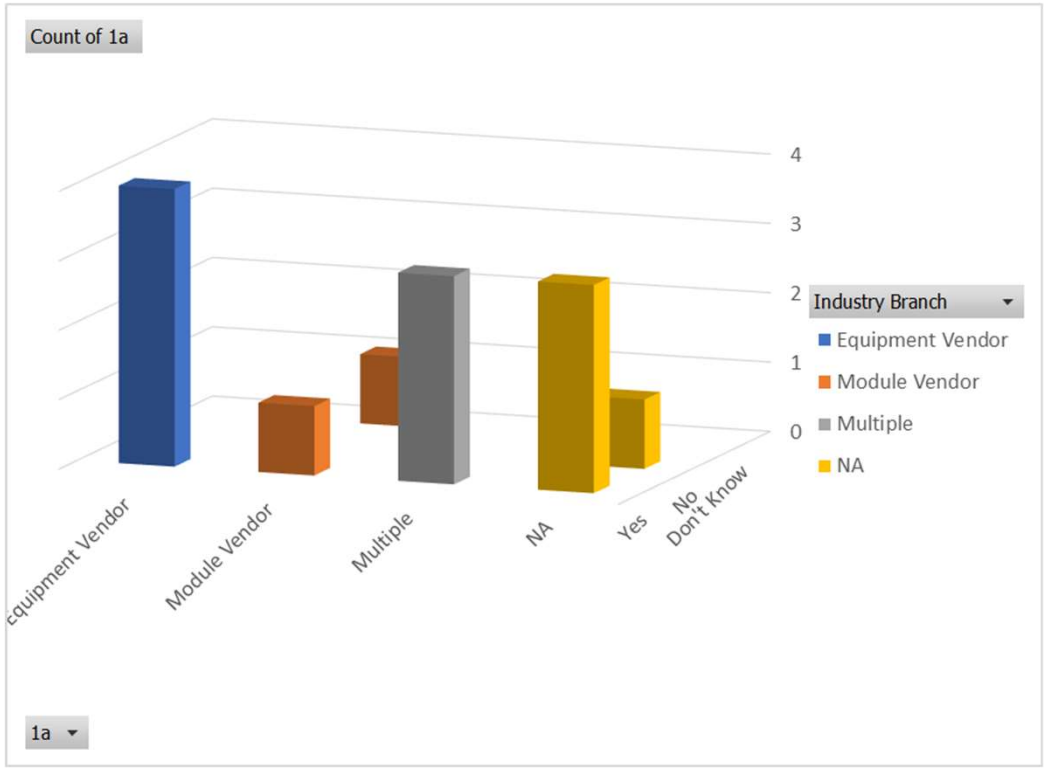
- 13 Respondents
  - 4 Equipment Vendor
  - 3 Multiple
  - 2 Module Vendor
  - 4 Not specified
- 169 verbatim comments

Question 1a - Module Complexity

Background - Complex modules offer more flexibility (i.e. programmable line rates, number of client ports, programmable client rate/protocol) and functionality (i.e. enhanced line and client monitoring, OTN features) but require more complexity to manage.

Question - Do you plan to use or make optical modules that have more complexity than existing modules across the industry?

11 – Yes, 1 – No, 1 – Don't Know

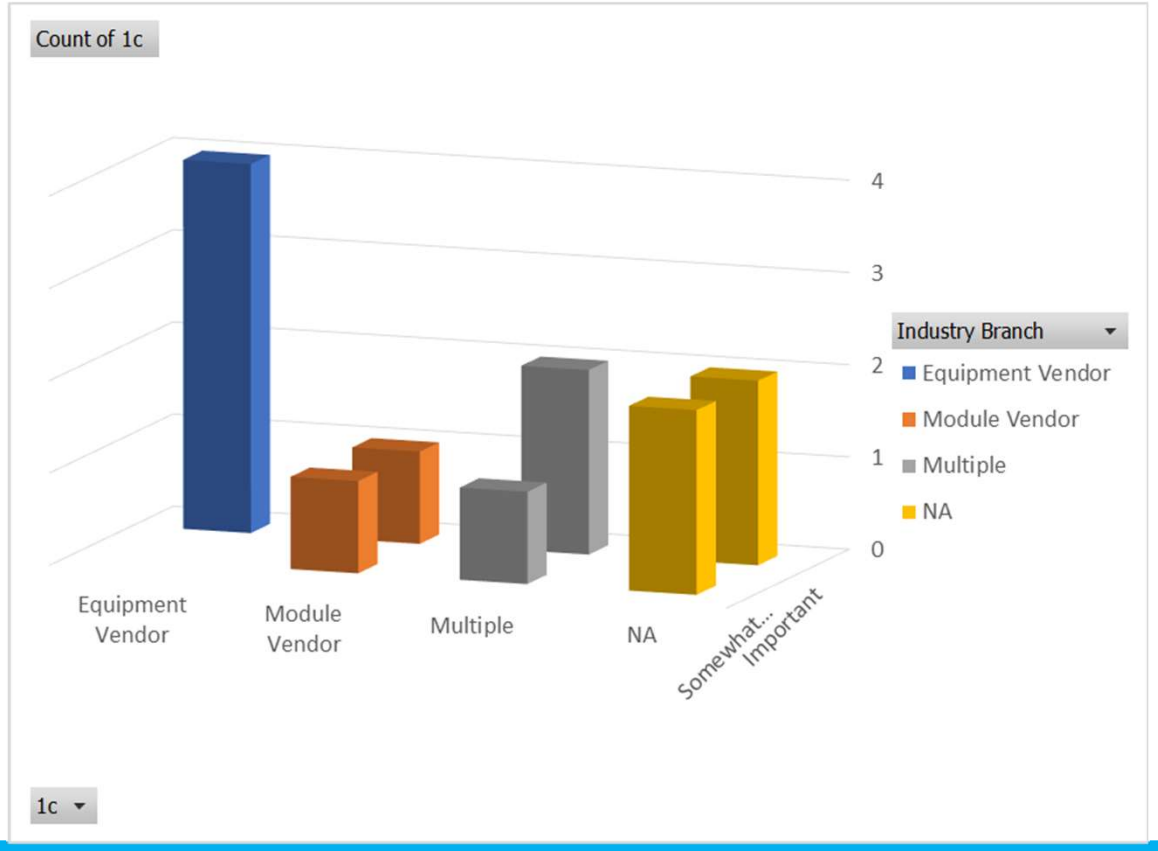


Question 1c - Module Complexity

Background - Complex modules offer more flexibility (i.e. programmable line rates, number of client ports, programmable client rate/protocol) and functionality (i.e. enhanced line and client monitoring, OTN features) but require more complexity to manage.

Importance

9 – Important, 4 – Somewhat important



Question 1 - Module Complexity

1a Comments:

- The minimum complexity required leads to lowest cost module internal sub-components, module firmware/software development and module set-up/testing in production. Complexity must be paid for in development time and per-module costs. Restrict the need for complexity to those markets and applications willing to pay for it. Do not paint with a broad brush.
- Programmability of line and client rates of optical interfaces is an important and interesting feature for operators.
- As a T&M company, we anticipate that modules will hold more processing/analysis/diagnostic power and we are preparing to address the need to test the stability of these transactions with hosts.
- Coherent and CPO modules are 10 to 100x more complicated than SFP's in the past. Based on the fact that optics is replacing copper more complex module design is a cross industry thing before module becomes simple hardware.
- We expect to use CPO when available
- We plan to make and deploy complex modules. We see challenges to standardizing all functionality but would like to see some level of standardization formalized.
- It is anticipated that Standardization drives some common interoperability modes that nevertheless are subject to technology evolution, i.e., CD-tolerance can be improveed compared to "standard/MSA requirements". While such improvements are generally desired, their use cannot be easily adopted and to do so possibly requires a different parameter profile than standard. This is true for same and different manufacturers as well as between product generations. These issues so have been seen with CFP2-DCO.

Importance

1c Comments:

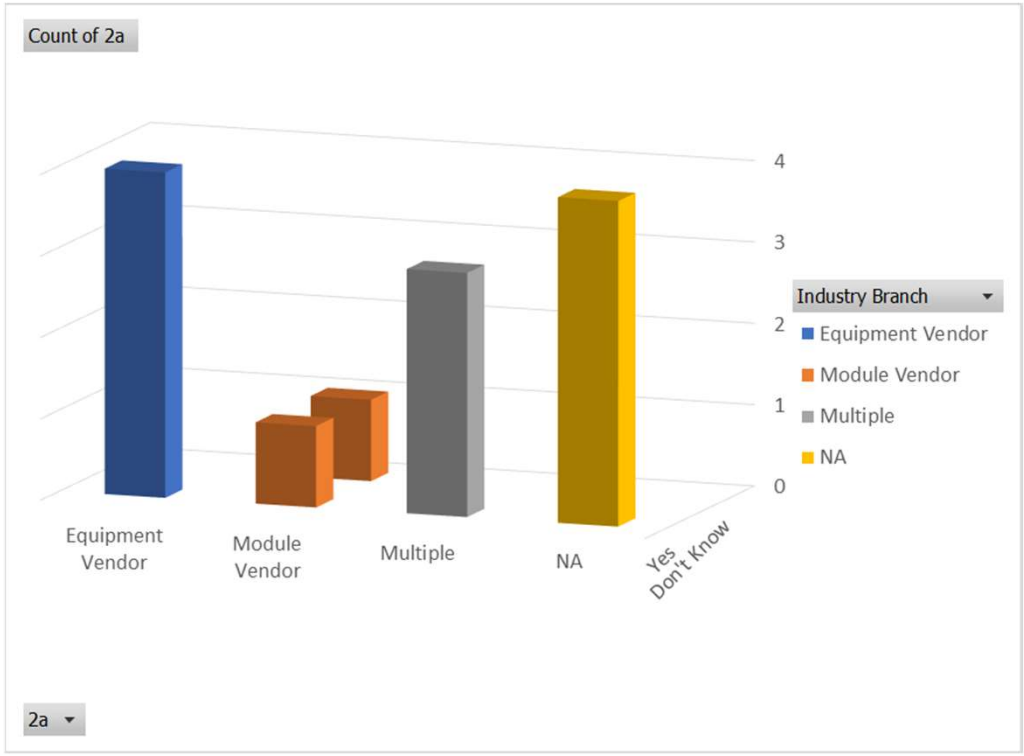
- The most expensive place to put complexity is in the module. The least expensive is in a capable host ASIC. For an industry which values low cost and low power modules, it continually seeks to place additional burden and functionality into these MSA commodity type components, in an apparently effort to reduce the amout of work system companies spend to implement similar functionality. Programmability of line and client rates of optical interfaces is an important and interesting feature for operators
- A standardized approach is required

Question 2a - Module Management Physical Interface Speed

Background - Complex modules will require more management bandwidth, and they may even require non-management data communication to be supported via pin count limited physical interfaces.

Question - As optical modules become more complex do you expect bandwidth intensive features/functionality (enhanced digital diagnostics, constellation, TCM, GCC, degrade signaling, Ethernet monitoring, protection, etc.) to be accessible on optical modules?

11 – Yes, 1 – No, 1 – Don't Know

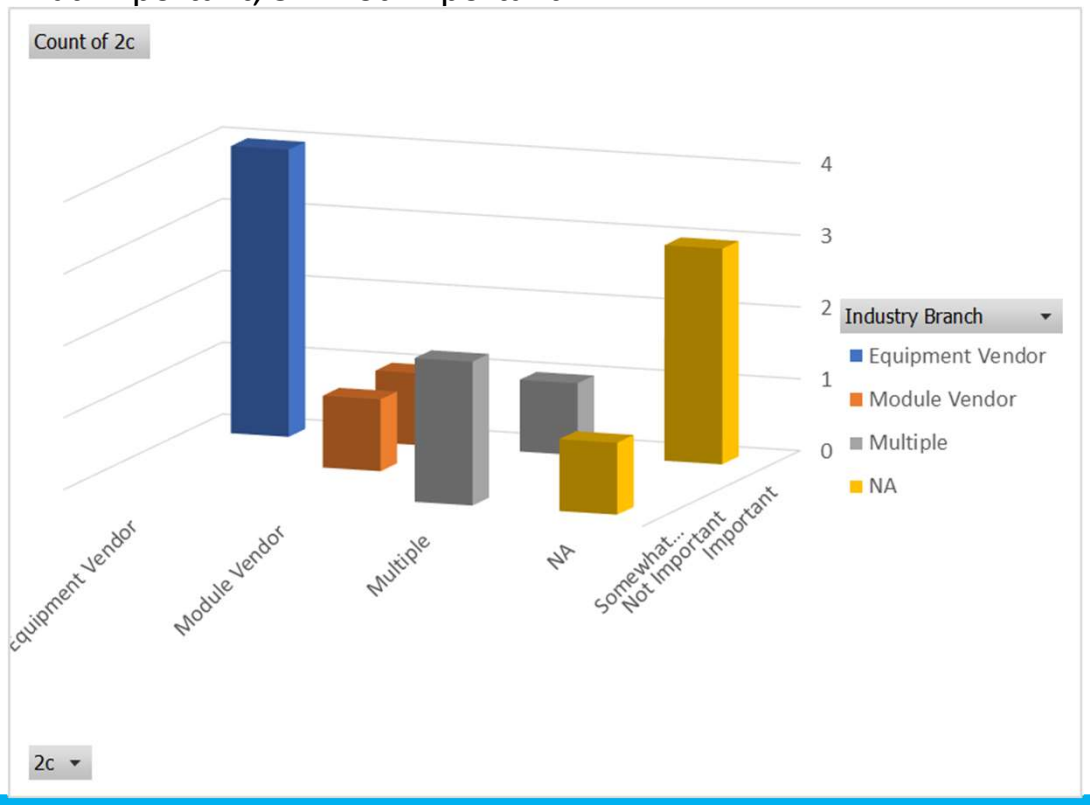


Question 2c - Module Management Physical Interface Speed

Background - Complex modules will require more management bandwidth, and they may even require non-management data communication to be supported via pin count limited physical interfaces.

Importance

9 – Important, 1 – Somewhat important, 3 – Not Important



Question 2 - Module Management Physical Interface Speed

2a Comments:

- data collection and analysis (ex: through AI or other mechanisms) of optical network features, and QAM constellation monitoring (today, rarely accessible) will become extremely relevant in a near future.
- It is evident that consolidating as much complexity into a pluggable device is conducive to lower CapEX for network builds.
- Another complexity is the IoT security requirement that drives the demand of bandwidth.
- Bandwidth needed clearly depends on the type of module with some requiring more than others
- Different parts of our business have different needs functionality. These features are important for some products and not needed for others.
- For complex modules, there needs to be a way to assess the proper operation during workload conditions. We see no need for bandwidth reserved for OTN. Could be that existing communication bandwidth to the module is good enough (tbc!).
- Encryption can be another intensive operation. It is important to mention also the number of pluggable per LC. Having a fast response will boost the CPU capability to manage tens of pluggable in a LC

Importance

2c Comments:

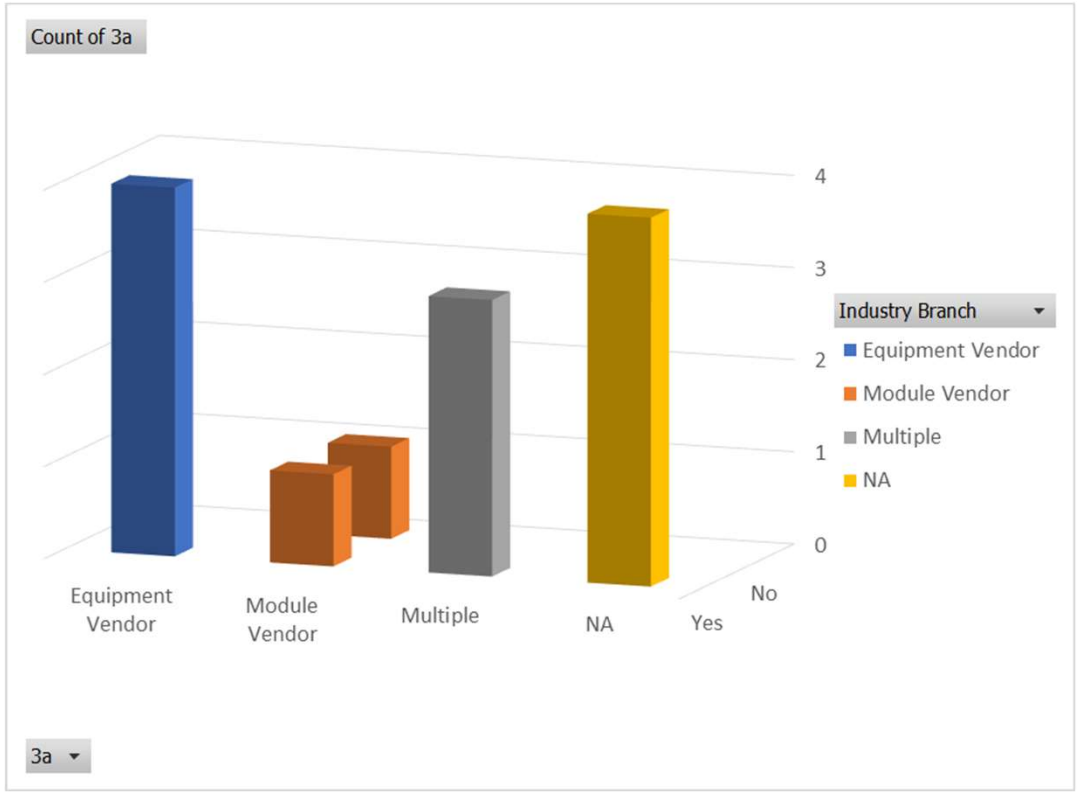
- As T&M access to real time diagnostic information is seen as important
- The Introduction speaks of addressing management interfaces for optical modules. Currently, there would be quite a range of performance to cover. Lumping together survey responses, e.g., 400G SR8 and 400G ZR in the same category of functionality, would not yield reliable results. Different markets, different electrical and optical functions, different granularity of control. Higher speed control interfaces add to cost of module sub-components.
- There are thinkable demands for directly managing traffic through MIS.
- Such functionality is extremely useful for system integration testing too.

Question 3a - Programmable modules (software defined)

Background - Unlike classical PHYs, DSP based high speed modules are often multipurpose programmable pluggables. These DSPs can be programmed to operate in multiple modes (i.e. 200G/400G, OTN/ZR) and they can be used in different form factors (i.e. OSFP, QSFP-DD, CFP2).

Question - Will you deploy / sell / support programmable modules?

12 – Yes, 1 – No, 0 – Don't Know

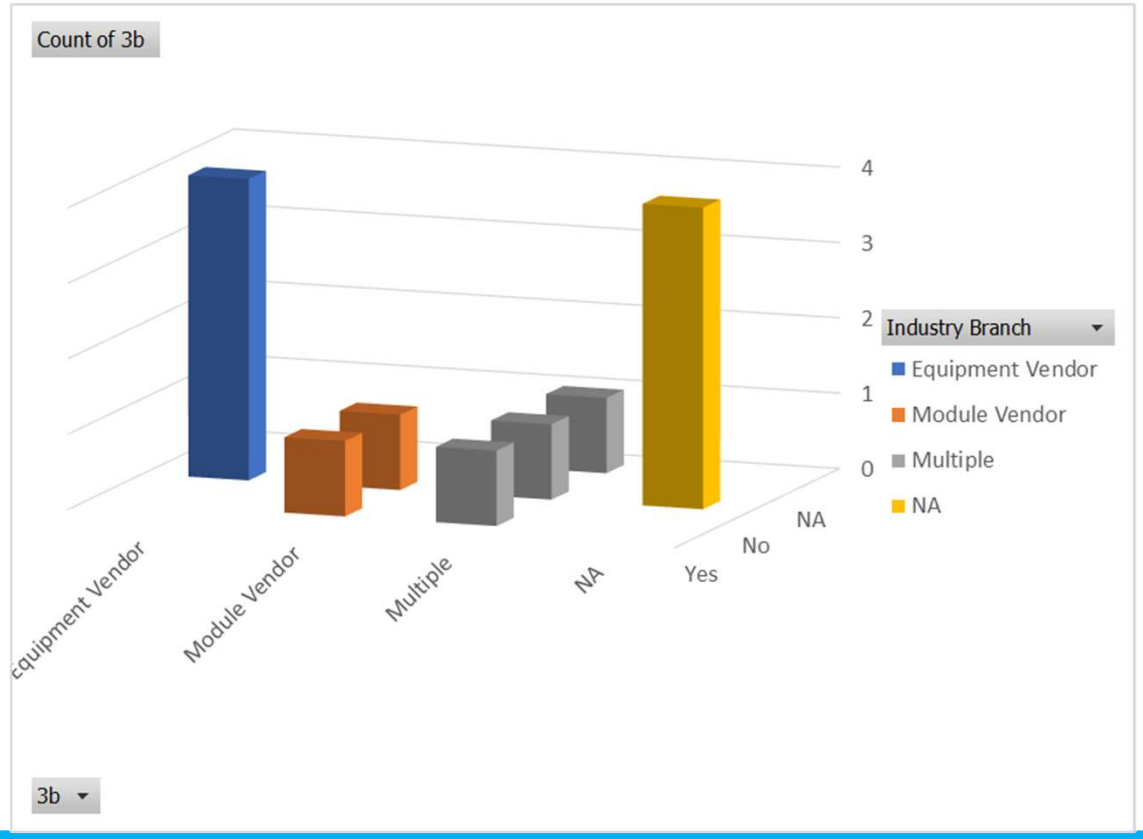


Question 3b - Programmable modules (software defined)

Background - Unlike classical PHYs, DSP based high speed modules are often multipurpose programmable pluggables. These DSPs can be programmed to operate in multiple modes (i.e. 200G/400G, OTN/ZR) and they can be used in different form factors (i.e. OSFP, QSFP-DD, CFP2).

Question - Or are your modules single purpose?

2 – Yes, 10 – No, 1 – NA

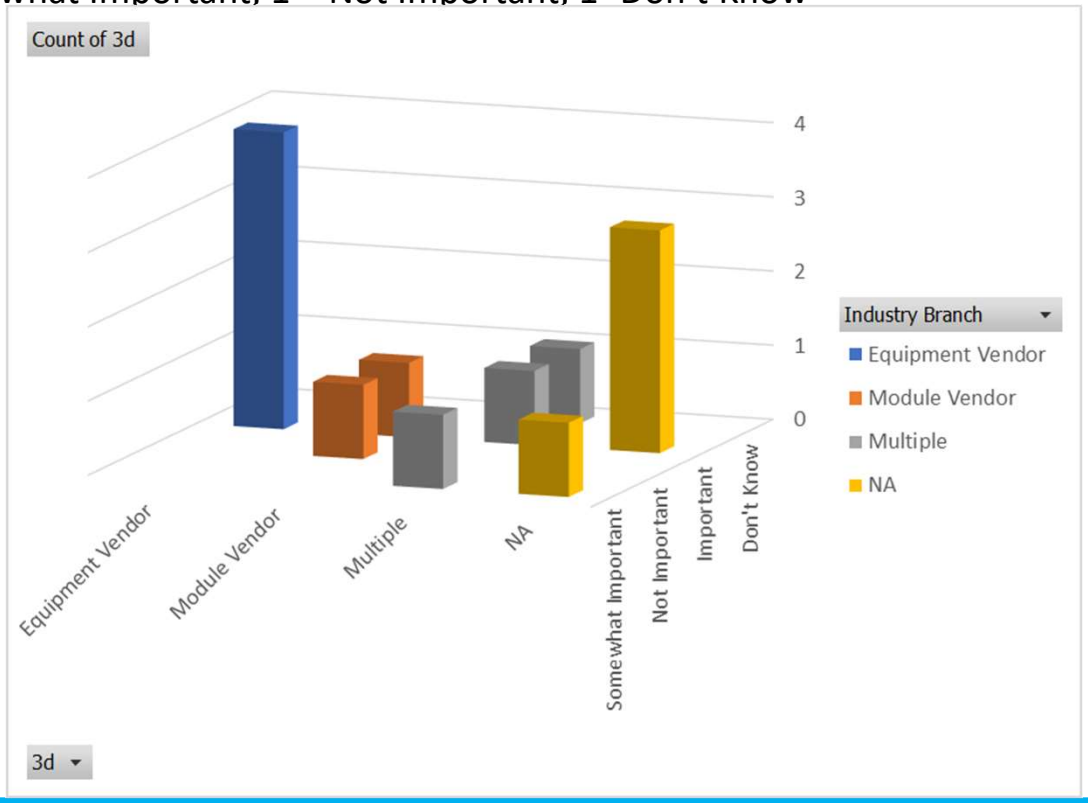


Question 3d - Programmable modules (software defined)

Background - Unlike classical PHYs, DSP based high speed modules are often multipurpose programmable pluggables. These DSPs can be programmed to operate in multiple modes (i.e. 200G/400G, OTN/ZR) and they can be used in different form factors (i.e. OSFP, QSFP-DD, CFP2).

Importance

9 – Important, 2 – Somewhat Important, 1 – Not Important, 1- Don't Know



Question 3 - Programmable modules (software defined)

3a/b Comments:

- High volume, low cost requires the least expensive and most fit-for-purpose component and design choices. Hyperscale and Enterprise data centers demand multiple suppliers, low prices and commodity business models. Multi-purpose pluggables get increasingly expensive to design, manufacture and test to validate that all configurations meet outgoing performance and quality goals. And, modes of operation not utilized by end customers are a waste of development & mfg time/money if under-utilized.
- Such programmable DSPs are already used in our optical transport networks, and will be deployed inside pluggable optic modules in a near future (ex: 400-ZR).
- We typically require 1 generation of backward compatibility in modules when staging upgrades
- This question is worded pretty vaguely - answer depends on what you mean. Modules that are programmable to run at different data rates have been available for many years already, and so are not a new category as seems to be implied in the question. If the question is talking about something beyond that, the "Yes" answer above is less clear - probably more "Maybe"
- We will have applications for both in different applications/products.
- Single purpose modules require more SKUs and are therefore more complex to deal with. Unless there is clear CAPEX advantage for single purpose modules, OPEX-wise it is cheaper to deal with multi-purpose modules. For example, if we do not need OTN features in very high volume, then such features in a multi-purpose module are good for CAPEX and OPEX. Software defined modules provide a method to reduce inventory and operational costs to end customers. This should be beneficial.

Importance

3d Comments:

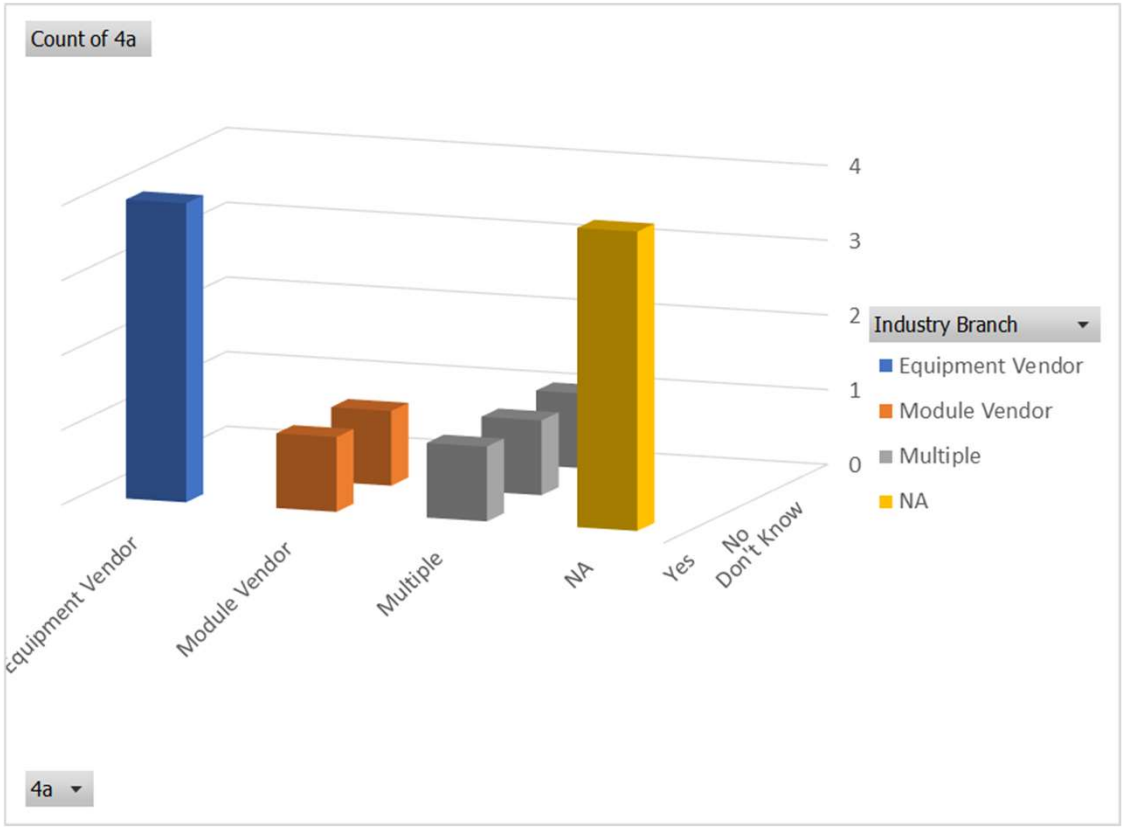
- Successful standard products will be focused on one application. If end customers want multi-functional parts, they can engage and fund their subsequent development.
- Presumably multi-purpose modules could be enabled by licenses, so there may be a need to properly check if a host is allowed to operate a module in a high-value mode.

Question 4a - System on a Module

Background - Even high speed Ethernet modules are no longer simple PHYs. OTN modules servicing Ethernet may even need to include full OTN termination and mapping (MUX on a module)

Question - Do you expect complex Ethernet Modules to be relevant?

10 – Yes, 2 – No, 1 – Don't Know

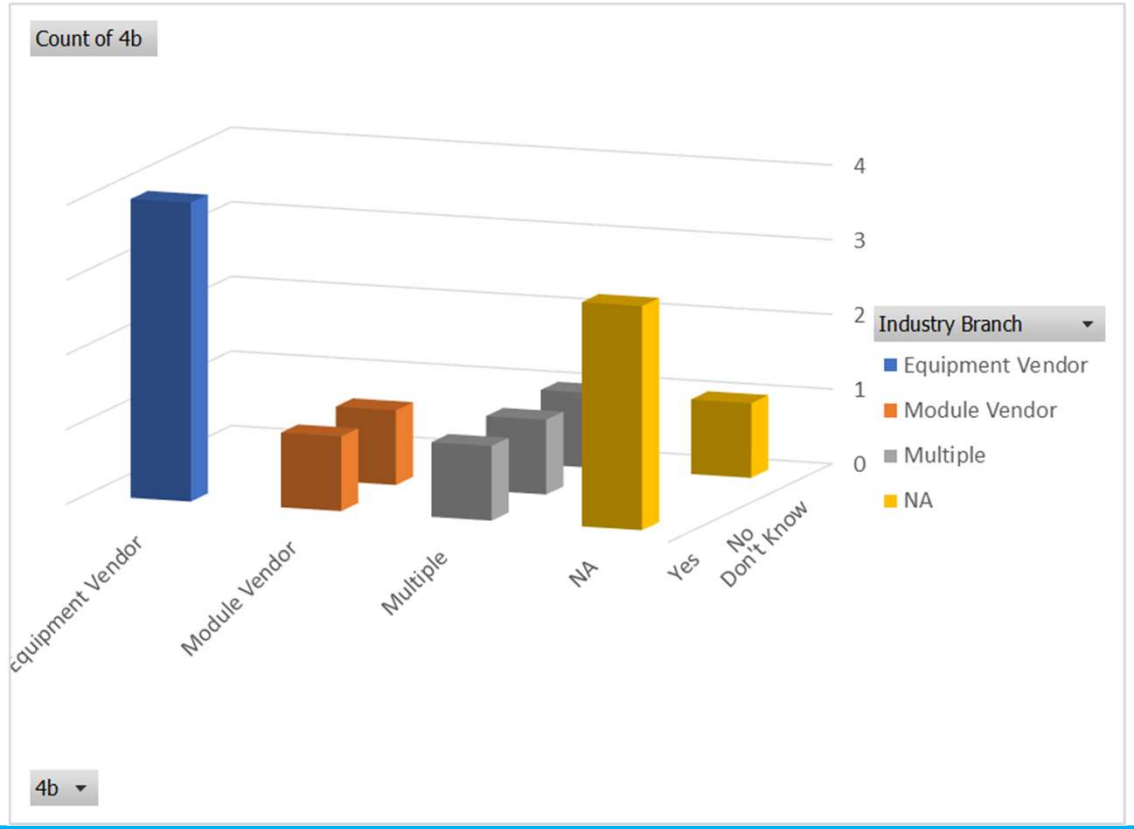


Question 4b - System on a Module

Background - Even high speed Ethernet modules are no longer simple PHYs. OTN modules servicing Ethernet may even need to include full OTN termination and mapping (MUX on a module)

Question - Do you expect "System on a Module" modules to become relevant?

9 – Yes, 2 – No, 2 – Don't Know

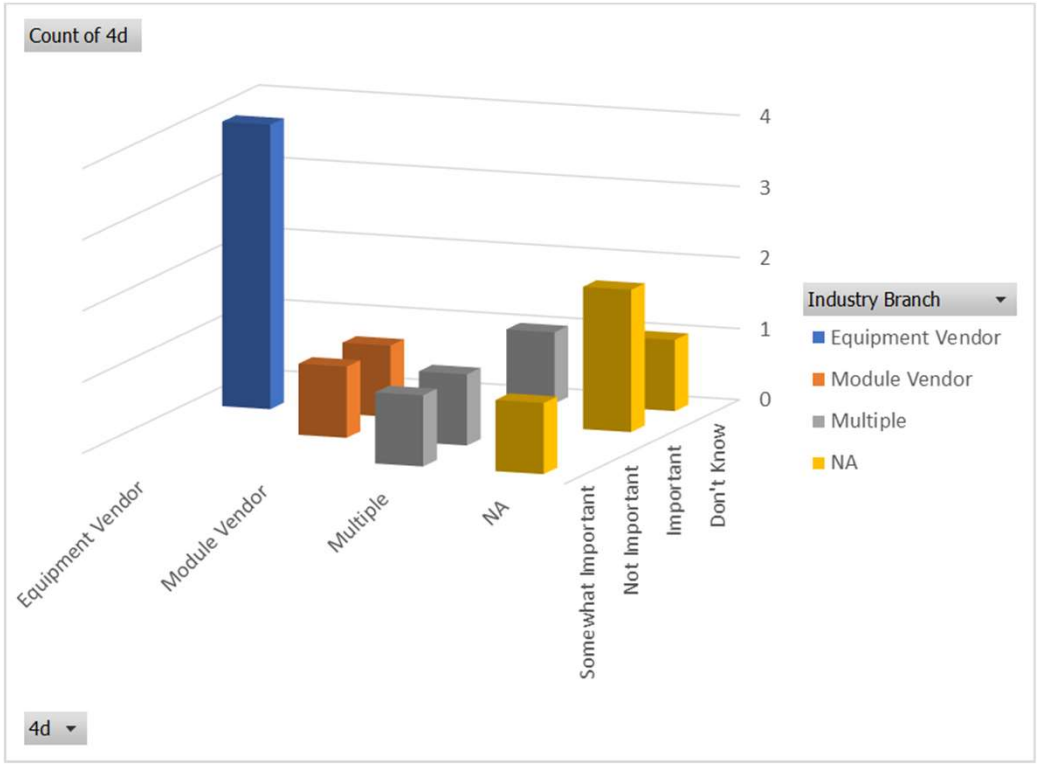


Question 4c - System on a Module

Background - Even high speed Ethernet modules are no longer simple PHYs. OTN modules servicing Ethernet may even need to include full OTN termination and mapping (MUX on a module)

Importance

7 – Important, 2 – Somewhat Important, 2 – Not Important, 2- Don't Know



Question 4 - System on a Module

4a/b Comments:

- For the System on a Module questions, 'relevant' is assumed to mean 'successful in the market'.
- ZR and ZR+ are definitely going to make a big splash and are going to be adopted by all hyperscalers
- The CPO effort can be a system on substrate which integrates conventional host and module into one entity.
- Again, depends what you mean. FEC termination is already required in many modules. Full OTN termination is less relevant from our standpoint.
- We will have applications for both in different applications/products.
- Understand that "system on a module" means a subtended system from a host. Something like a USB-stick plugged in a PC.

Importance

4d Comments:

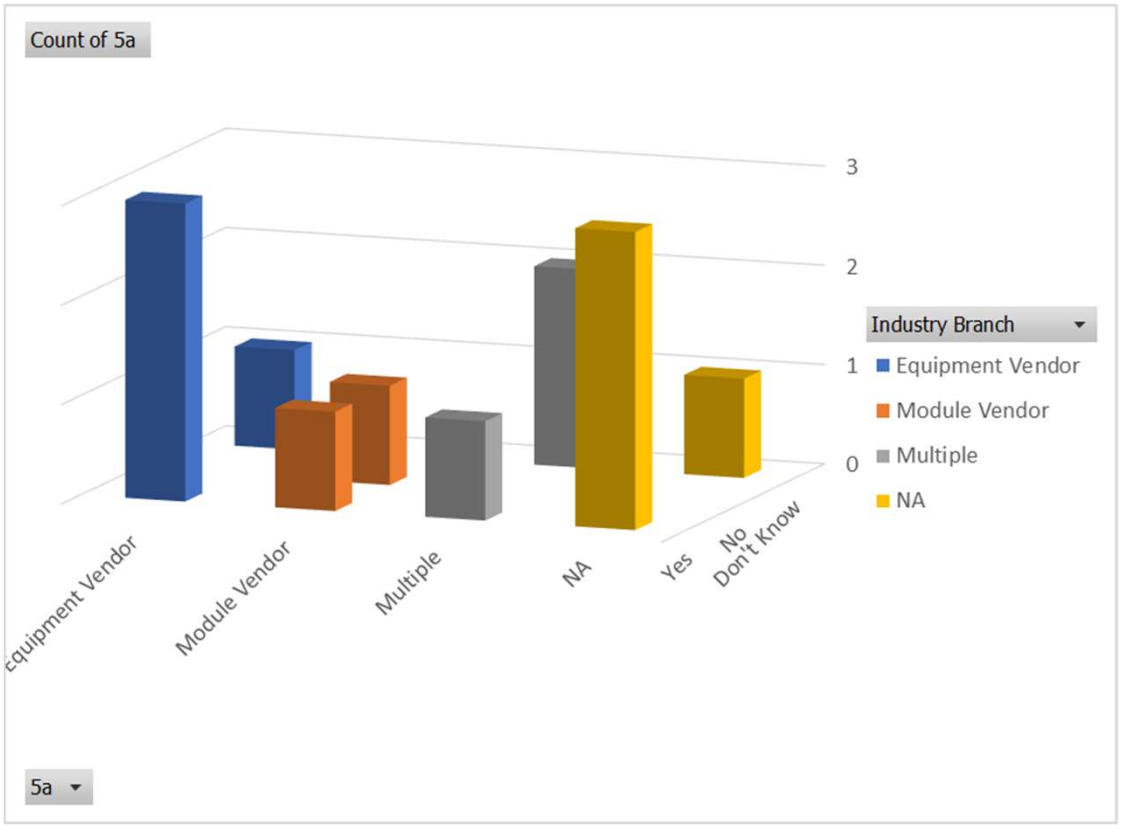
- The associated workflow makes "system on a module" interesting for system vendors as hosts can be fully qualified and incrementally updated with module/driver HW/SW combinations that are kept outside the main SW release. Think about Microsoft-Windows being rolled out and pluggable modules being added independently and validated independently.

Question 5a - Multi Protocol Modules

Background - DSP based high speed modules may offer Ethernet and OTN personalities

Question - Do you plan to use or make such modules?

8 – Yes, 1 – No, 4 – Don't Know

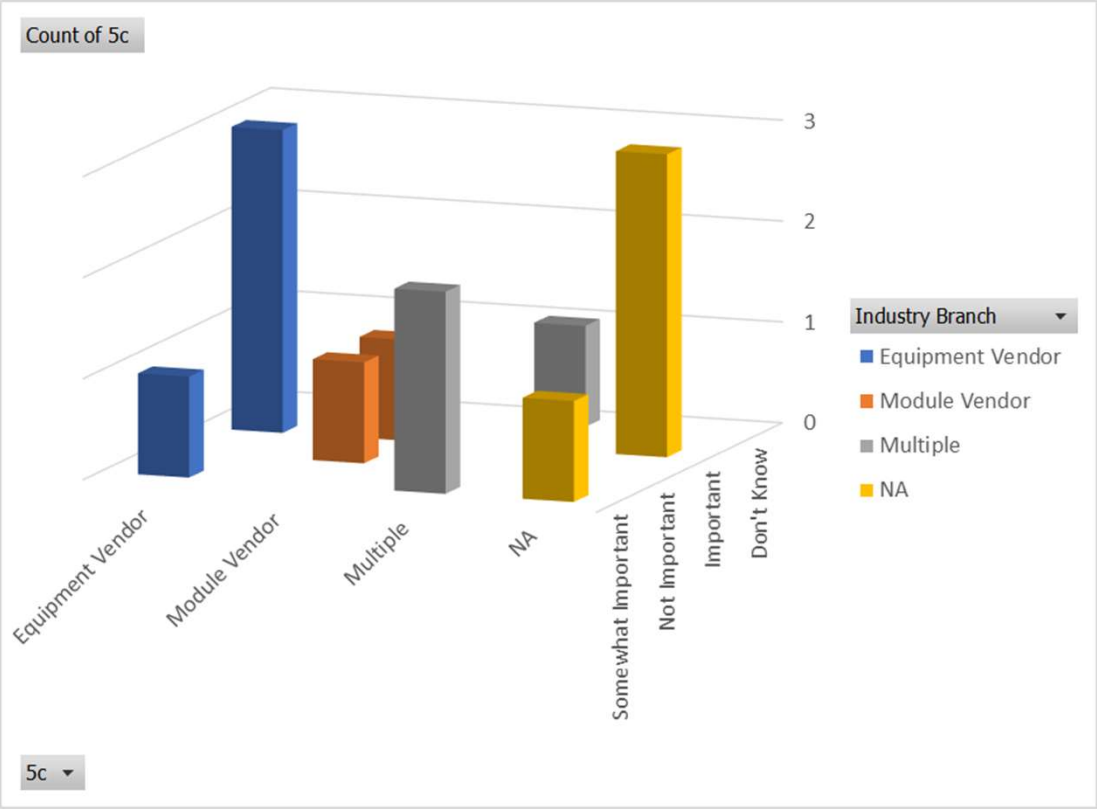


Question 5c - Multi Protocol Modules

Background - DSP based high speed modules may offer Ethernet and OTN personalities

Importance

7 – Important, 4 – Somewhat Important, 1 – Not Important, 1- Don't Know



Question 5 - Multi Protocol Modules

5a Comments:

- Interesting for spare part management and pre-production to respond rapidely to customers' requirements
- We will "use" them in the sense that we will develop test solutions for them.
- We plan to make and deploy both personalities

Importance

5c Comments:

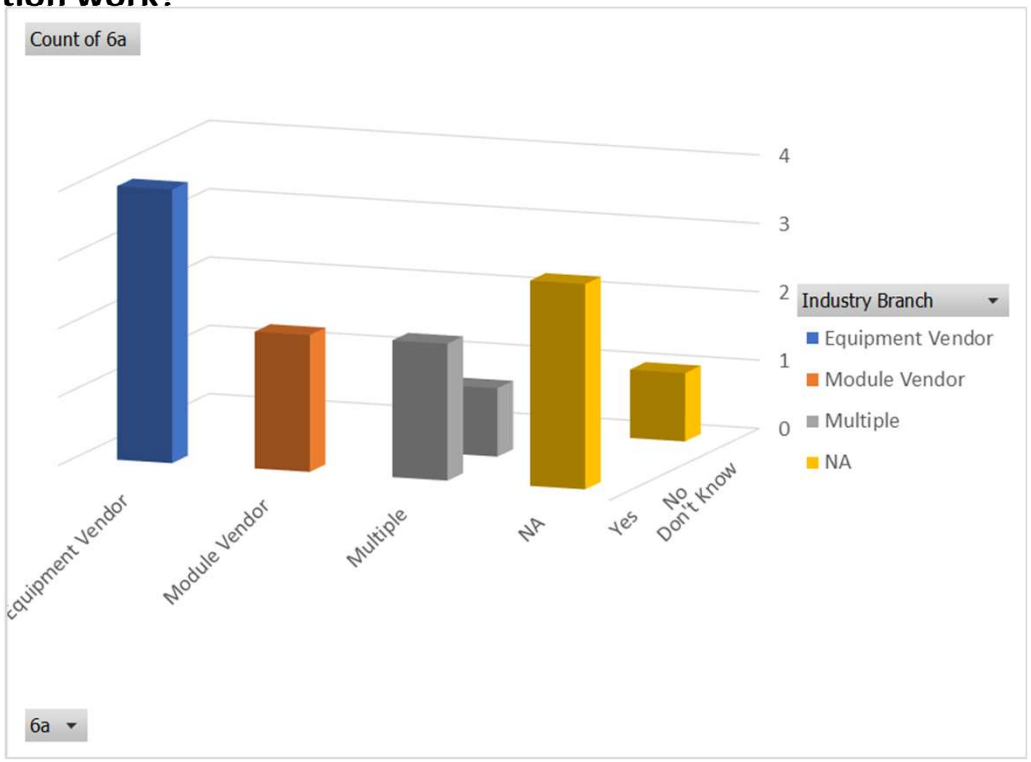
- OTN is nowadays mainly used as "digital wrapper" for long-reach interfaces. It is unclear how much of this is still required when coherent Ethernet interfaces become available. The OTN personality is a nice-to-have.

Question 6a - Timely Updates to MIS Standards When New Modules/Features are Required

Background - New revisions of Management Interface Specification (MIS) register memory map standards that are published to support new features or modules often lag the industry deployment of the new features or modules. This is often the result of serialization of the hardware standardization process followed by the MIS standard.

Question - Have you experienced problems in deploying a new module or new feature due to the MIS standardization process not having completed definition work?

11 – Yes, 1 – No, 1 – Don't Know

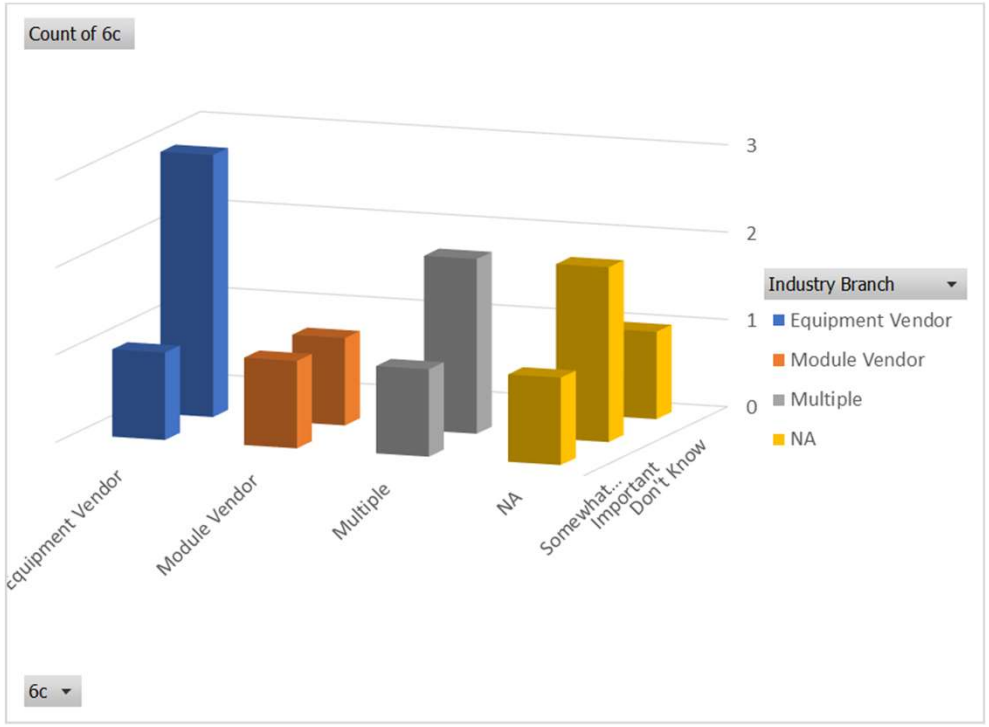


Question 6c - Timely Updates to MIS Standards When New Modules/Features are Required

Background - New revisions of Management Interface Specification (MIS) register memory map standards that are published to support new features or modules often lag the industry deployment of the new features or modules. This is often the result of serialization of the hardware standardization process followed by the MIS standard.

Importance

8 – Important, 4 – Somewhat Important, 1 – Not Important



Question 6 - Timely Updates to MIS Standards When New Modules/Features are Required

6a Comments:

- No clear industry consensus and standard for 400G CFP2-DCO Module Management
- This doesn't seem to be due to form factor, i.e., hardware finalization but by endless requests for new/revised features, e.g. in support of coherent modules. A solution may be to have an annual or biannual MIS release and freeze submissions at the well-publized beginning, take a short predefined period (e.g. 4 -6 weeks) to decide which submissions will be addressed over a predetermined time (e.g. 3-6 months) and then complete the work and publish release 20xx.
- The CMIS standards are constantly changing and evolving due to the turbluent nature of multi-vendor interop with increased module complexity Arranging register map is a huge and non-productive effort in the past. In the curent set up there has no quick way to amend or delete functions. In CFP MIS effort publishing Adanddum was used to add new features but shoehorning new functions into existing register map has been a hassle. A command interface keeps revisioning to include new functinality and to phase obsolete functionality is a good way to go.
- CMIS Rev4.0 development gated our ability to accept new modules
- Importance of backward full compatibility among CMIS revision this was not the case in the transition from 3.0 to 4.0
- This is important, but again, we don't see a significant current problem with CMIS
- The MIS never keeps up with the hardware. 400ZR was close but still behind.
- Common problem in the Industry and with standards in general. Clarity on the availability of modules with newer version from vendors causes unknowns in schedules.

Importance

6c Comments:

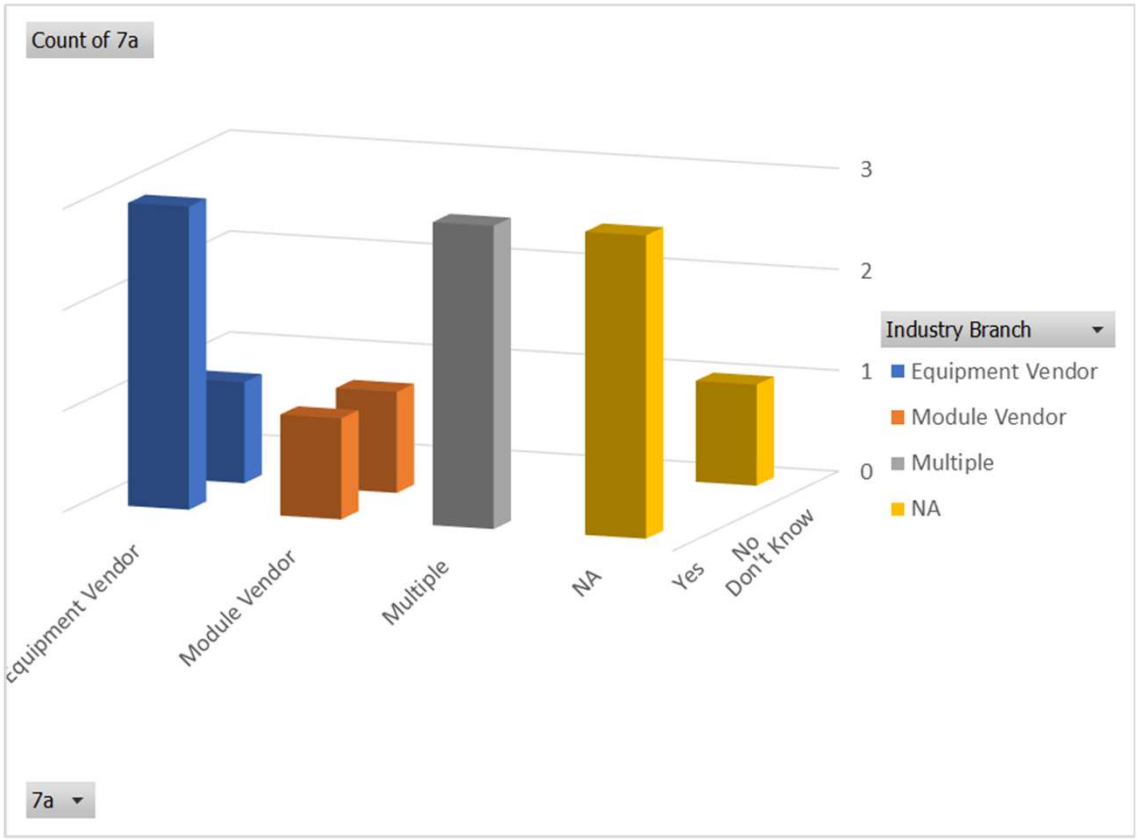
- Agility will likely be required as as new diagnostic features may be wanted and as more form factors are emerging, looking toward CPO as another large step over OBO.
- This is important, but again, we don't see a significant current problem with CMIS
- Other standards provide the capability to add extensions that can be used in a consistent manner, i.e., OpenConfig, IETF YANG. Standard helps to build more reliable module-drivers in-house.

Question 7a - Integration Effort

Background - Adding support for new modules requires software effort on host platforms.

Question - Is software integration effort a factor in selection of a new optical module or the decision to design a new module?

10 – Yes, 2 – No, 1 – Don't Know

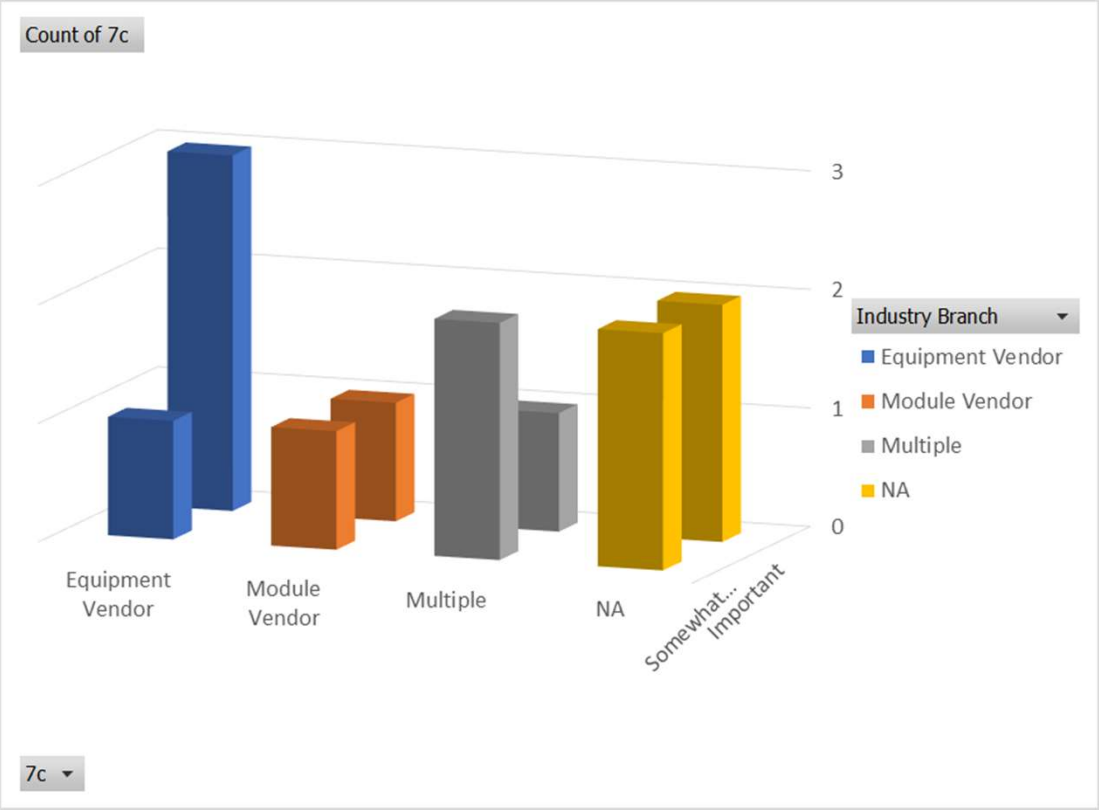


Question 7c - Integration Effort

Background - Adding support for new modules requires software effort on host platforms.

Importance

7 – Important, 6 – Somewhat Important



Question 7 - Integration Effort

7a Comments:

- As T&M we want to support a variety of optical modules and will support whatever the market demands.
- SW integration will inevitably be joined at the hip with Interoperabilty.
- It makes things much easier to have a standard management interface that works across all modules (even if some modules don't support all the features).
- SW integration is a second tier factor behind performance, feature set and cost.
- This is a "weak yes". Industry applies a dual-vendor strategy and selecting the second vendor is eased if it's implementation is similar to the first. The availability of modules based on a matured standard is desirable. Though there are times when some amount of risk is taken when the newer speeds/packages are to be integrated.
- Generally the SW is not really in the long poll and if pluggable is required SW will have to comit resource

Importance

7c Comments:

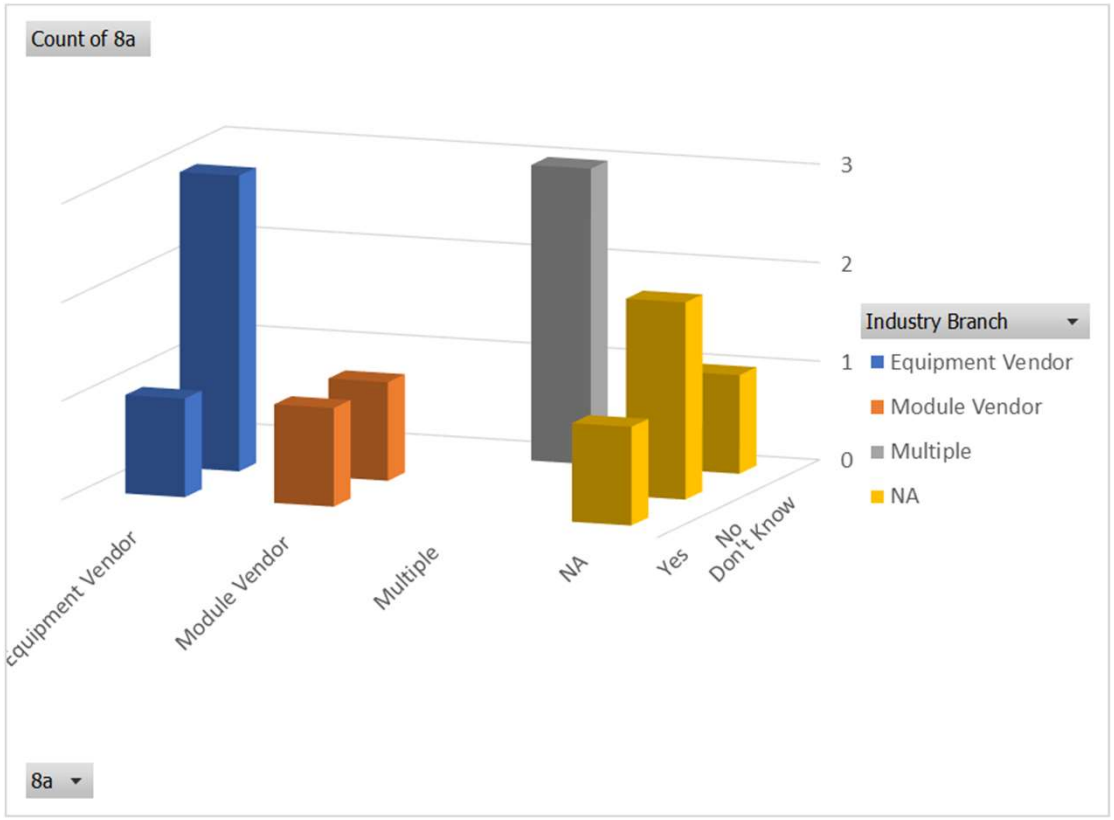
- a good level of standardization helps to modernize networks i.e. through simplifying module upgrade or RMA cases using different variants even of the same vendor.

Question 8a - Limitations in Adding Registers to Standards Based Memory Maps

Background - Port functions (such as OTN) are feature rich in terms of management information. It may be difficult to manage in the fixed memory context of CMIS (TWI) or CFP MSA (MDIO).

Question - Are module makers and/or module consumers willing to accept a loss of functionality if it cannot be scaled into the existing MIS information structures?

3 – Yes, 6 – No, 4 – Don't Know

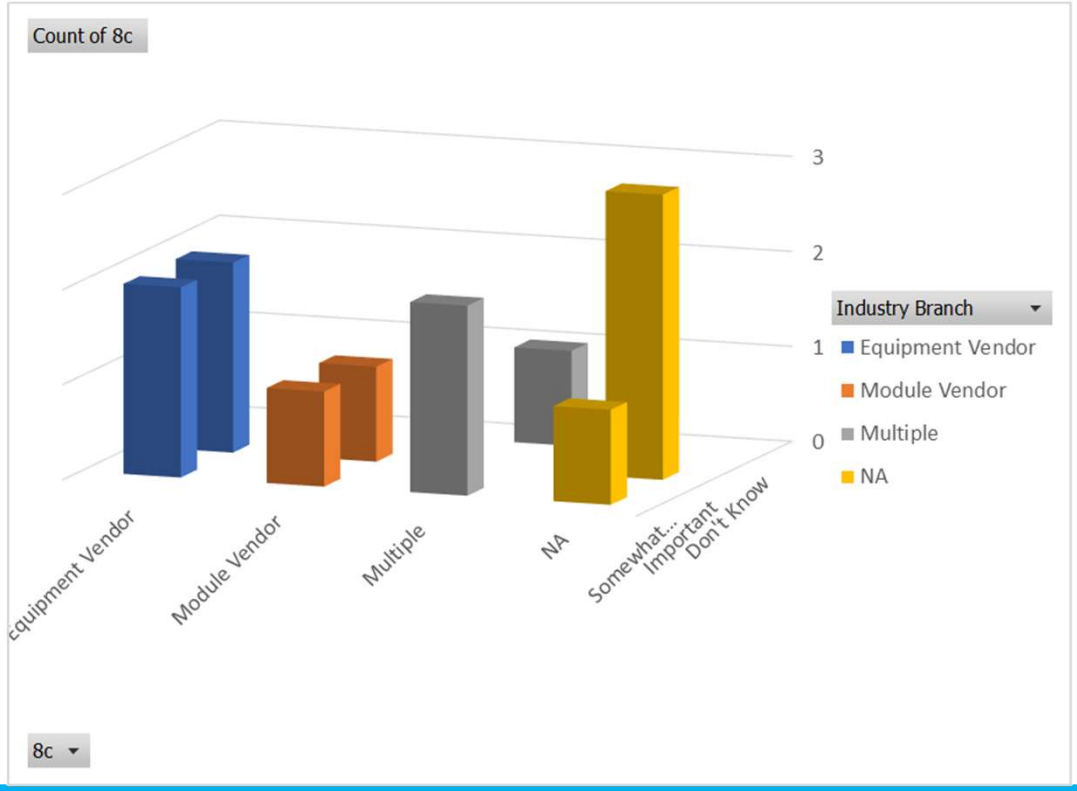


Question 8c - Limitations in Adding Registers to Standards Based Memory Maps

Background - Port functions (such as OTN) are feature rich in terms of management information. It may be difficult to manage in the fixed memory context of CMIS (TWI) or CFP MSA (MDIO).

Importance

6 – Important, 6 – Somewhat Important, 1 – Don't Know



Question 8 - Limitations in Adding Registers to Standards Based Memory Maps

8a Comments:

- The question offers a false choice. CMIS as currently defined can be greatly expanded and means can likely be found for further expansion. Meanwhile, has anyone demonstrated a memory size limit issue?
- There has not many cases that I am aware of in eliminating new functions due to limitation of MIS information structures. Since CFP MIS, we have added CDB (Command Data Block) to mitegate such difficulty. It has been effective.
- This is not directly applicable to me.
- This is an application specific question. Depends on application and supporting equipment.
- Not much of header bytes are controlled directly by applications, so this might be acceptable as long as basic alarms and access to few header fields such as GCC, TTI is available.

Importance

8c Comments:

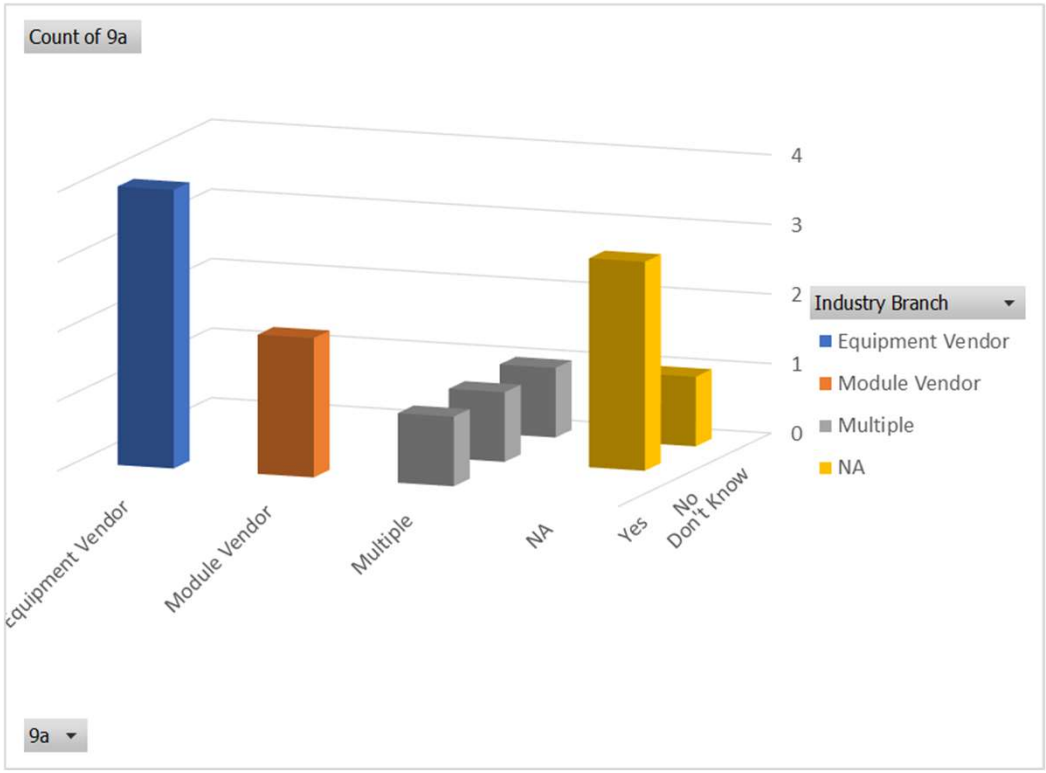
- Not important for datacom / Ethernet, which are already covered by CMIS. If this new (implicitly) proposed effort is scoped to cover only transport/coherent, it may be useful.
- Depends very much on which features are really lost. In general, not every OTN overhead byte needs to be managed and we do not use OTN features

Question 9a - Burden of Module Firmware Supporting a DSP in Form Factors with Different Memory Maps

Background - High-performance coherent DSPs are targeting multiple form factors for deployment. The register memory map depends on the form factor MIS MSA.

Question - Is the need to support multiple memory maps increasing the FW development effort or unnecessarily increasing the complexity of the firmware?

3 – Yes, 6 – No, 4 – Don't Know

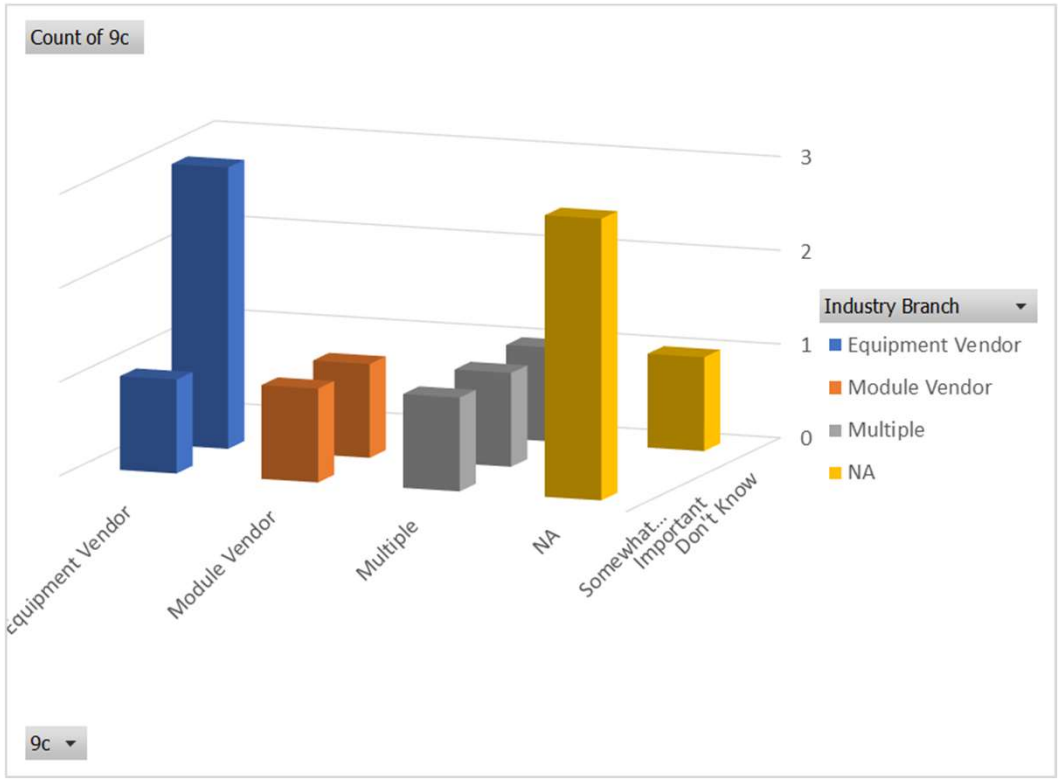


Question 9c - Burden of Module Firmware Supporting a DSP in Form Factors with Different Memory Maps

Background - High-performance coherent DSPs are targeting multiple form factors for deployment. The register memory map depends on the form factor MIS MSA.

Importance

7 – Important, 4 – Somewhat Important, 2 – Don't Know



Question 9 - Burden of Module Firmware Supporting a DSP in Form Factors with Different Memory Maps

9a Comments:

- Because DSP vendors have started to use API based mangement system while the performance and design need improvements
- This is not directly applicable to me.
- Cooperation between relevant datacom module form factors on common CMIS spec has converged on only one spec we need to design to for next-gen datacom optics
- Having the same DSP used in different memory maps increases complexity and effort.

Importance

9c Comments:

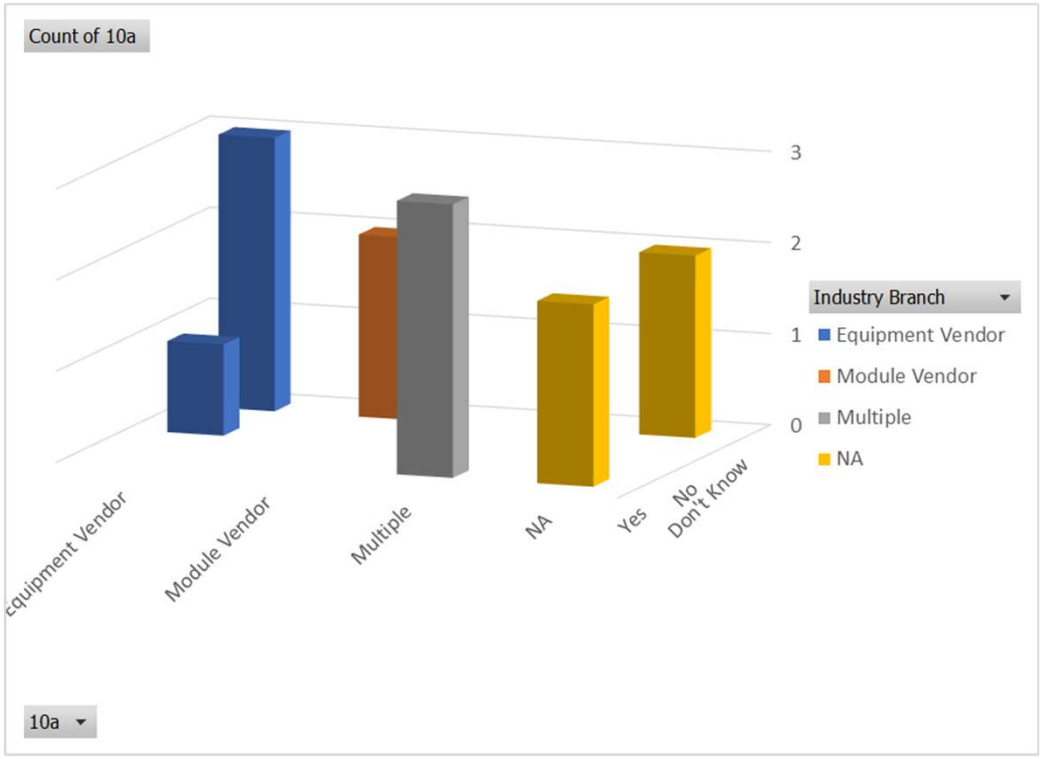
- Unifying such API will increse the compatibility among different DSP vendors. Right now module vendors have to deal with vendor specific API's which vertually perform similar functionalities.
- Important, but as stated above, we don't see this as a current problem
- Developing new hosts accepting new form factors is burdened by developing different SW per form-factor. Ideally the Firmware is largely independent of form-factor and re-useable across platforms.

Question 10a - Beyond CMIS

Background - CMIS was a first attempt to achieve "uniform management" for form factors using TWI for management communication.
Moving forward toward better integration and froth, some propose the goal of "uniform management" even across form factors and across physical communication protocols (TWI, MDIO, SPI, Ethernet)

Question - Do you like the CMIS idea?

5 – Yes, 1 – No, 7 – Don't Know

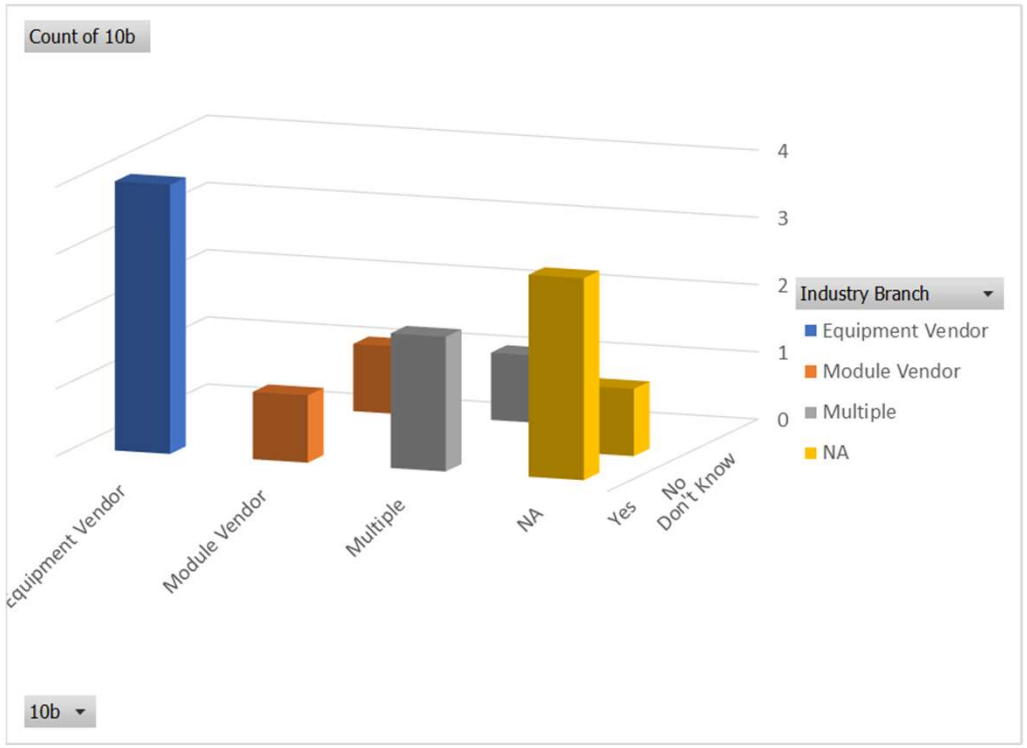


Question 10b - Beyond CMIS

Background - CMIS was a first attempt to achieve "uniform management" for form factors using TWI for management communication.
Moving forward toward better integration and froth, some propose the goal of "uniform management" even across form factors and across physical communication protocols (TWI, MDIO, SPI, Ethernet)

Question - Do you think it was successful?

10 – Yes, 1 – No, 2 – Don't Know

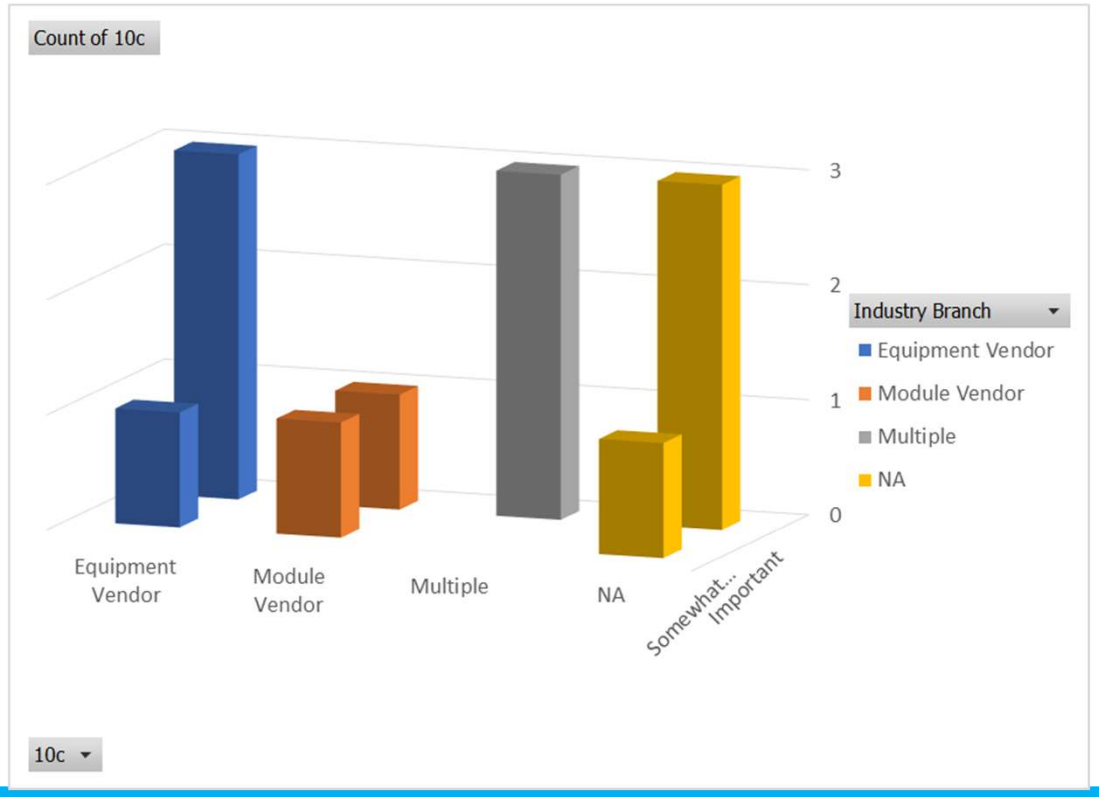


Question 10c - Beyond CMIS

Background - CMIS was a first attempt to achieve "uniform management" for form factors using TWI for management communication.
Moving forward toward better integration and froth, some propose the goal of "uniform management" even across form factors and across physical communication protocols (TWI, MDIO, SPI, Ethernet)

Question - Do you like the idea to carry this further?

10 – Yes, 3 – No, 0 – Don't Know



Question 10e - Beyond CMIS

Background - CMIS was a first attempt to achieve "uniform management" for form factors using TWI for management communication.
Moving forward toward better integration and froth, some propose the goal of "uniform management" even across form factors and across physical communication protocols (TWI, MDIO, SPI, Ethernet)

Importance

10 – Important, 3 – Somewhat Important

Question 10 - Beyond CMIS

10 a/b/c Comments:

- "It's too early to know. To explain, we were deeply involved in CMIS development. These are not ""Yes"" or ""No"" questions ... and whoever wrote this survey may unintentionally be using a leading question based approach towards a desired response. CMIS is a good idea within a channel count (ie. 8 wide QSFP-DD & OSFP), but not so good applied to other channel counts (ie. 1-wide or 2-wide) ... it's way overdesigned (feature creep) for reduced channel width applications. The jury is still out if CMIS is a decent fit for 4-wide, as seen on 200G. Also, many provisions of CMIS are optional and are necessarily ignored for datacom grade components trying to keep functionality fit-for-purpose and avoid bloat in capability and code. Each proper response to Question #10 is ""it depends"", not ""Yes"" or ""No""."
- CMIS is necessary to achieve seamless multivendor interop with DSP based optics, and helps understand the root cause of interoperability failures.
- CMIS has shown limitation of using register maps in add and drop. It took great effort to organize the information and functionalities. It took very long time to develop since many things are tied together. API will somewhat untangle the un-necessary associations due to register map. Register map revision requires industry wide consensus which represent overhead with no gain. API can be added by independant parties and promote innovations. Another limitation of CMIS is its physical interface of TWI and its protocol of 8-bit. This makes the register map organization even more complicated.
- I agree that different form factors may want different physical communication protocols but still allow the same software layer to manage the module. I think CMIS was successful, but will need to be updated in the future.
- Not sure what "carry this further" means. CMIS still exists - is the proposal to form a new group to supersede this, or just to continue CMIS as a working group?
- We are agree with taking it further, possibly further than was intended in CMIS. At a minimum to provide framework for how custom spaces would be used. Interested in discussion of next gen MIS.
- Strong yes! Having a common interface spec is beneficial. There will be many more form factors coming in future. The CMIS idea was successful in getting things started, but unfortunately execution didn't follow for all kinds of modules. We would anticipate that simplicity of integrating high-value modules into host systems would become another decision criteria in selecting module vendors and achieve agility in network roll-out.
- CMIS idea was good but the historical limit of I2C is really a bottle neck

Question 10 - Beyond CMIS

Importance

10e Comments:

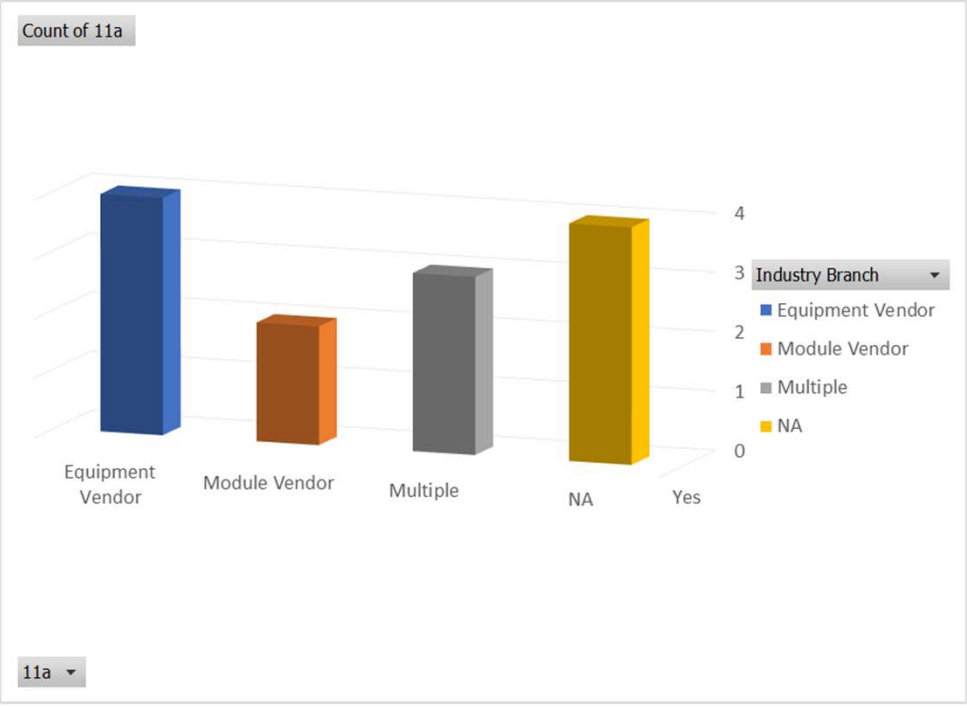
- CMIS is important - whether or not some other industry effort is important depends on what it is.
- See the need to advance managemenet interface for complex applications. CMIS is default candidate but may need more drastic change than updating CMIS.
- Today a Host SW can only be released when SW integration with Module is completed. Would be much quicker to perform a SW integration erly in time (supported by a standard) and add bugfixes/specifics later in the process.

Question 11a - Form-Agnostic Management

Background - Today, management software for a module function is form factor dependent.
For example, a 400G port implemented in a programmable CFP2 module is managed differently from the same 400G port implemented in QSFP-DD.
In the past this was acceptable because form and function of a module were often closely coupled.
In the future the same function may be available in several form factors.

Question - Would you see benefits (e.g. for host software architectures or development) if the management approach (i.e. how a module presents itself to a host) of modules would be more uniform across form factors?

13 – Yes, 0 – No, 0 – Don't Know

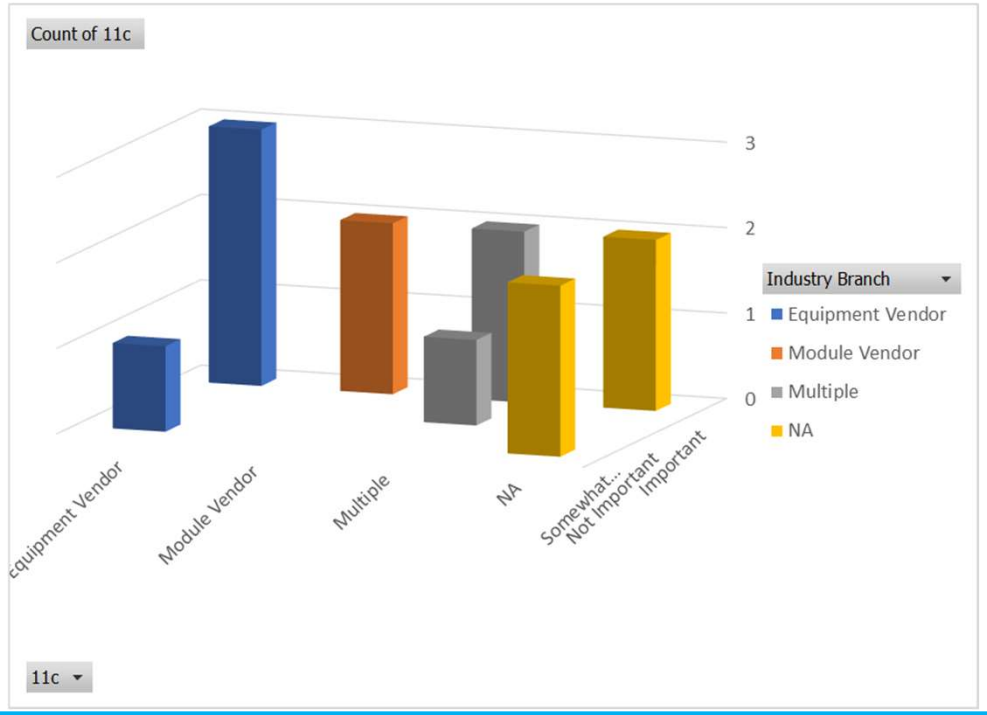


Question 11c - Form-Agnostic Management

Background - Today, management software for a module function is form factor dependent.
For example, a 400G port implemented in a programmable CFP2 module is managed differently from the same 400G port implemented in QSFP-DD.
In the past this was acceptable because form and function of a module were often closely coupled.
In the future the same function may be available in several form factors.

Importance

9 – Important, 3 – Somewhat Important, 1 – Don't Know



Question 11 - Form-Agnostic Management

11a Comments:

- Converging OSFP and QSFP-DD was a good idea. But converging SFP, SFP-DD, QSFP with QSFP-DD is not a great idea, as the simpler devices get bloated, costs go up, development time goes up, and the net value add to the host is rather limited.
- CMIS has started the idea of cross-form-factor interface. Disparity of different modules in management interface is partially attributed to lacking of standard on hardware interface. Currently two HW interfaces are in use, I2C (TWI) and MDIO. Each has its own pros and cons. But neither of them is designed for plugbility. In the view of IoT, ethernet interface shall be considered seriously for optical transceivers.
- We've already begun this migration with putting CMIS4.0 on both QSFP56 and QSFP-DD
- …but not relevant to any product or application we are working on now
- Single management interface across all form factors will save complexity and effort. Can be difficult to predict the future form factor variants and impact on architecture.
- One benefit is to have a much more stable driver set, instead of many different drivers with small differences. It would go a long way in terms of software reuse and having a stable software base. Normally we end up spending considerable effort when a new form factor is available.

Importance

11 c Comments:

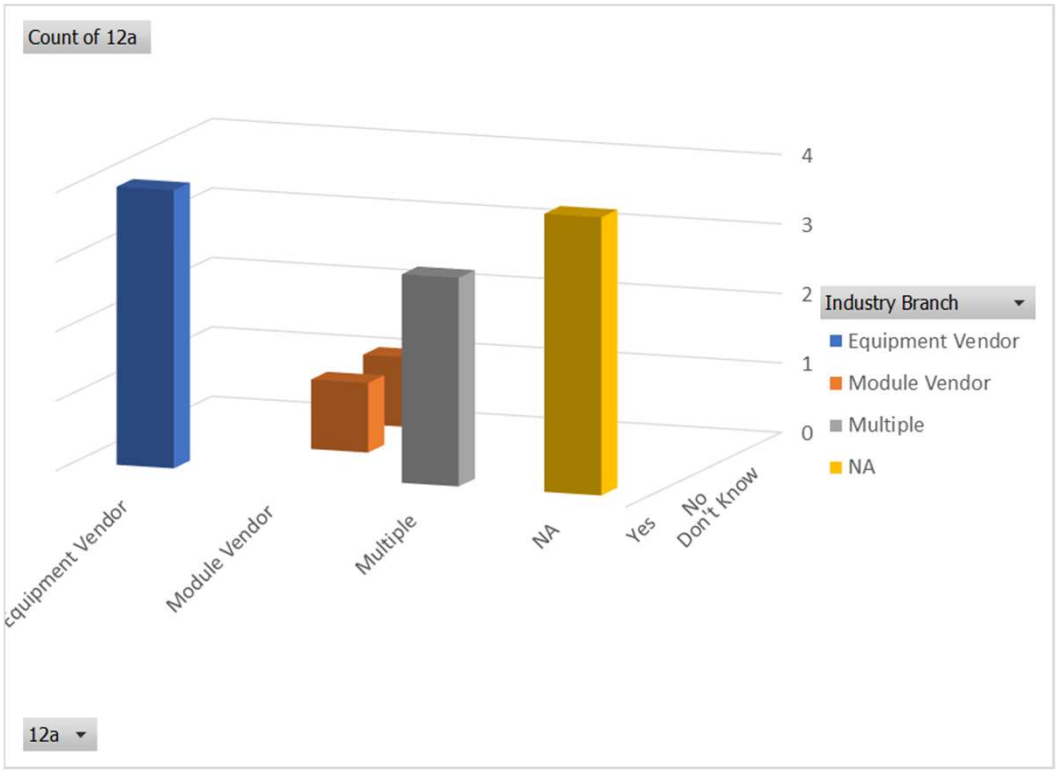
- Very important as T&M
- Each vendor spends significant resources on developing MIS. Quite bit of service calls are related to these trivial issues. It is hence important to have an unified approach to save industry's effort and apply them to more important innovations.
- not an issue that's relevant for any product or application we are working on now
- Every time we desiggn a new host using different form-factors there is effort to change implementation for new form-factors. Obviously we'd like to re-use code that already exists. We are not using different form factors every year, but still developing form-factor-dependent drivers has impact without generating value.

Question 12a - Generic Management

Background - In CMIS evolution it was considered important that less complex and low cost modules are not penalized by the fact that much more complex modules shall be managed via the same core management interfaces.

Question - Should the management of complex modules and of simple modules be unified (from a host view)?

11 – Yes, 1 – No, 1 – Don't Know

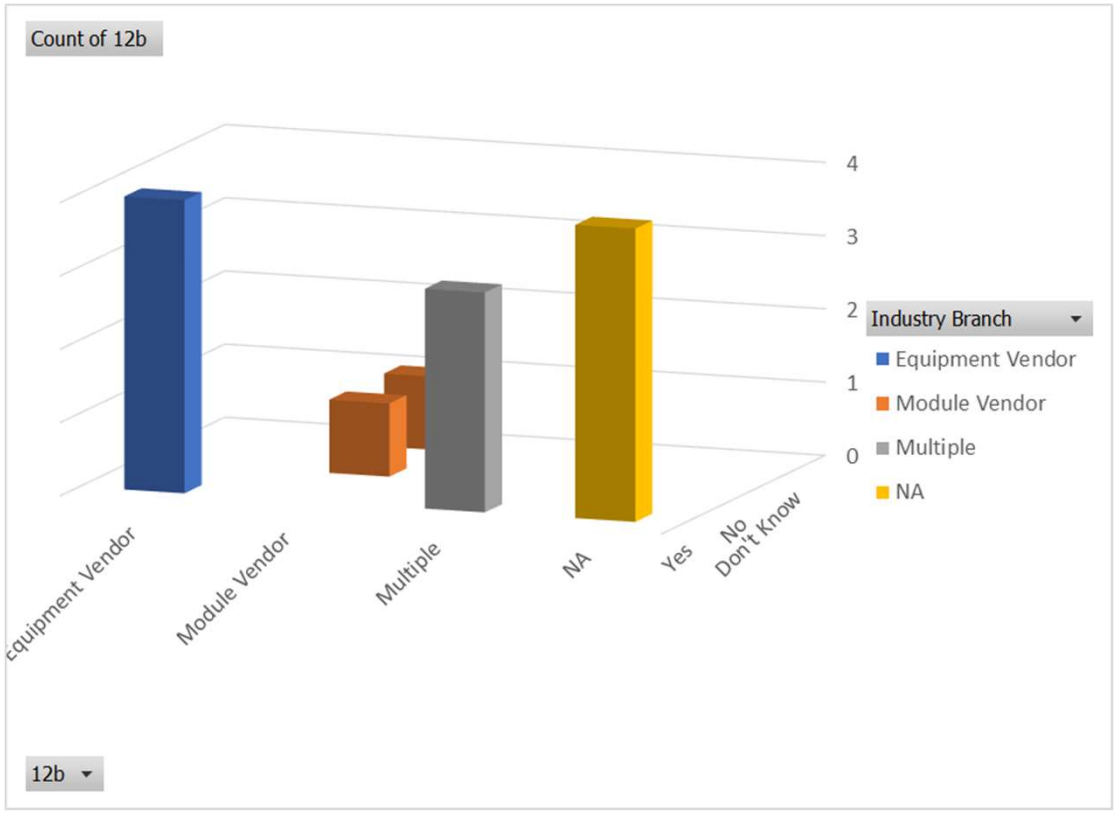


Question 12b - Generic Management

Background - In CMIS evolution it was considered important that less complex and low cost modules are not penalized by the fact that much more complex modules shall be managed via the same core management interfaces.

Question - Generally?

11 – Yes, 1 – No, 1 – Don't Know

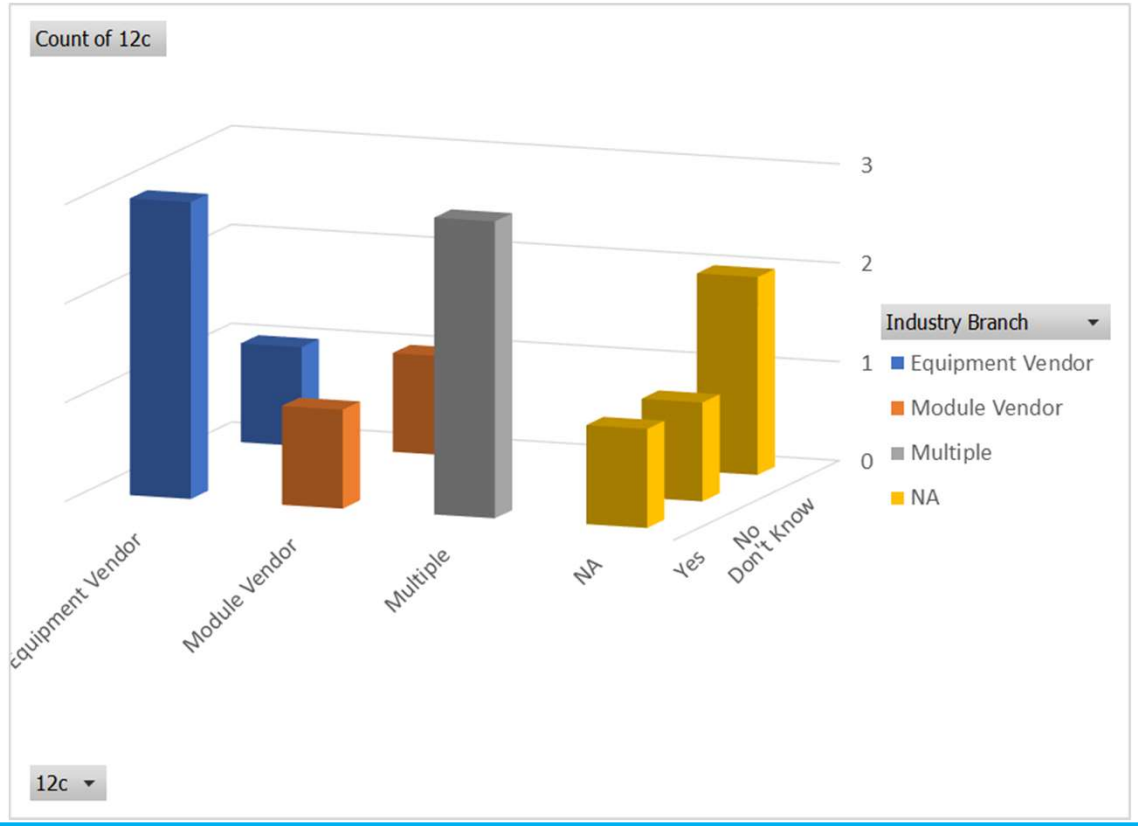


Question 12c - Generic Management

Background - In CMIS evolution it was considered important that less complex and low cost modules are not penalized by the fact that much more complex modules shall be managed via the same core management interfaces.

Question - In the same form factor?

8 – Yes, 1 – No, 4 – Don't Know

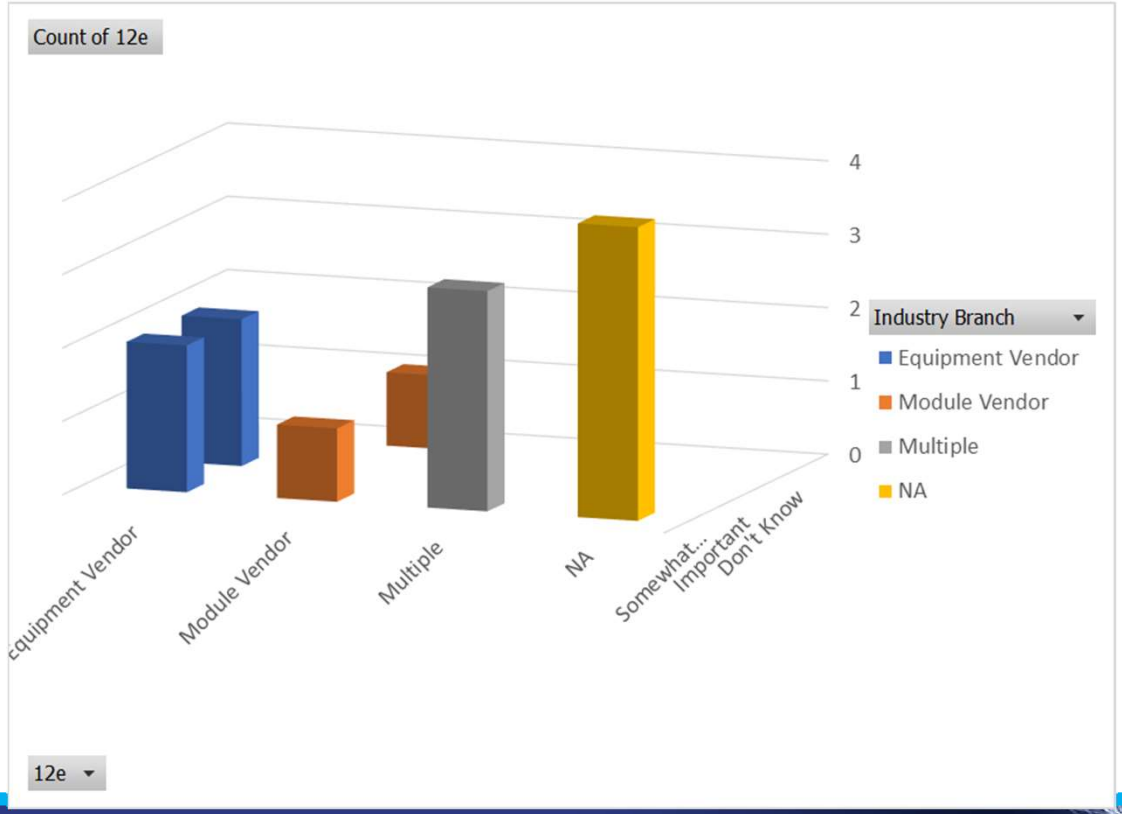


Question 12e - Generic Management

Background - In CMIS evolution it was considered important that less complex and low cost modules are not penalized by the fact that much more complex modules shall be managed via the same core management interfaces.

Importance

2 – Important, 10 – Somewhat Important, 1 – Don't Know



Question 12 - Generic Management

12 a/b/c Comments:

- Ok to converge within duals SFP-DD, NGSFP and DSFP. Ok to converge within octets OSFP and QSFP-DD.
- Innovation will make any advancement obsolete but some core functionalities can be shared by either complex or simple modules.
- The memory map and basic controls should be the same, even if low cost modules don't implement the majority of the features.
- All things being equal, management should be unified. However, will not be worth it if low cost modules have to bear the burden and overhead of functionality and requirements of complex modules
- Multiple opinions within Ciena. Agree on the goal of unified management some discussion of implementation on different size modules.
- Having similar semantics keeps software uniform and the development/qualification cycles can be shorter.

Importance

12e Comments:

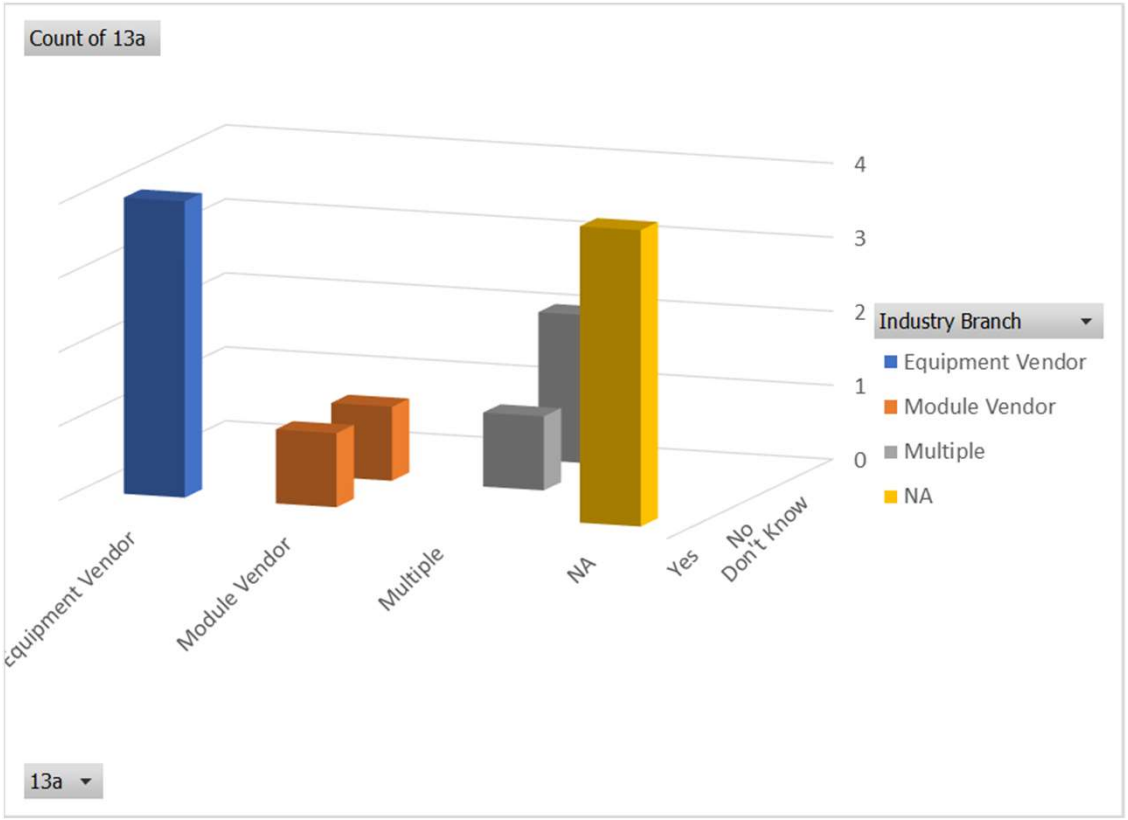
- Although, "it was considered important that less complex and low cost modules are not penalized by the fact that much more complex modules shall be managed via the same core management interfaces" it's not clear that goal was honored or achieved.
- All things being equal, management should be unified. However, will not be worth it if low cost modules have to bear the burden and overhead of functionality and requirements of complex modules
- Ideally, yes. The Model shall be such that simple modules only need to implement a reduced subset and are not burdened by a full implementation, i.e., in YANG all parameters are optional, so vendors can pick&choose what to implement. The key feature of such models is that the module announces which parameters are supported. This does not mean that YANG is the right implementaton for modules. There are possibly other less-intrusive ways that work similarly.

Question 13a - Plug & Play Management (3rd party modules)

Background -CMIS allows vendor agnostic module management of modules not known at design time. This would allow systems to work with arbitrary CMIS compliant modules (not just with a fixed selection of explicitly supported modules)

Question - Is this a use case for your company?

9 – Yes, 2 – No, 2 – Don't Know

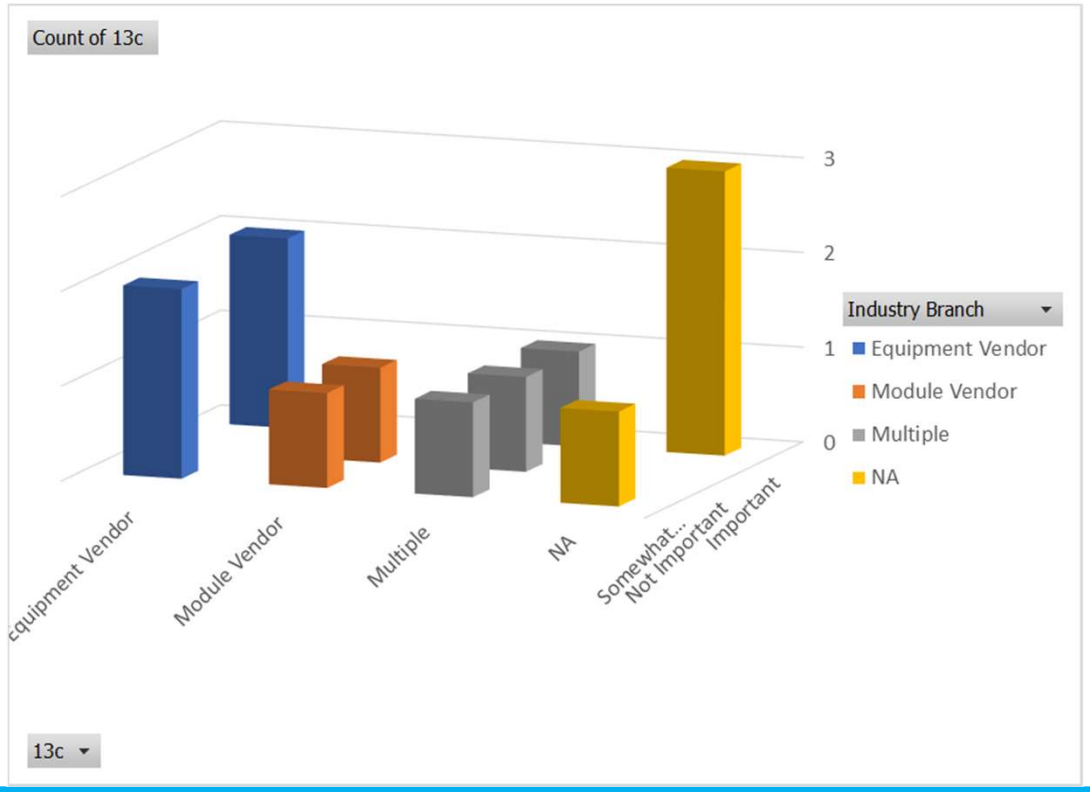


Question 13c - Plug & Play Management (3rd party modules)

Background - CMIS allows vendor agnostic module management of modules not known at design time. This would allow systems to work with arbitrary CMIS compliant modules (not just with a fixed selection of explicitly supported modules)

Importance

6 – Important, 5 – Somewhat Important, 2 – Don't Know



Question 13 - Plug & Play Management (3rd party modules)

13a Comments:

- As a module vendor we find it useful, because it eases fast adoption
- We will validate the modules we deploy
- Agree on the goal, we want maximum re-use between similar and non-similar plugs. We typically update SW for each new plug but like the changes to be minimal.
- A "modern host" would detect an unknown model, seek a DB for validated drivers, and download or reject the module. The same as a Windows PC wuld do upon plugging in a unknown USB device. This is especially useful as end customers choose their preferrable vendors. Also, from the program schedule perspective, the unknowns would be reduced greatly.

Importance

13c Comments:

- Very important for T&M
- This would drive the volume since it is much easier to upsell new modules on deployed hosts.

Question 14a - Communication Speed and Purpose

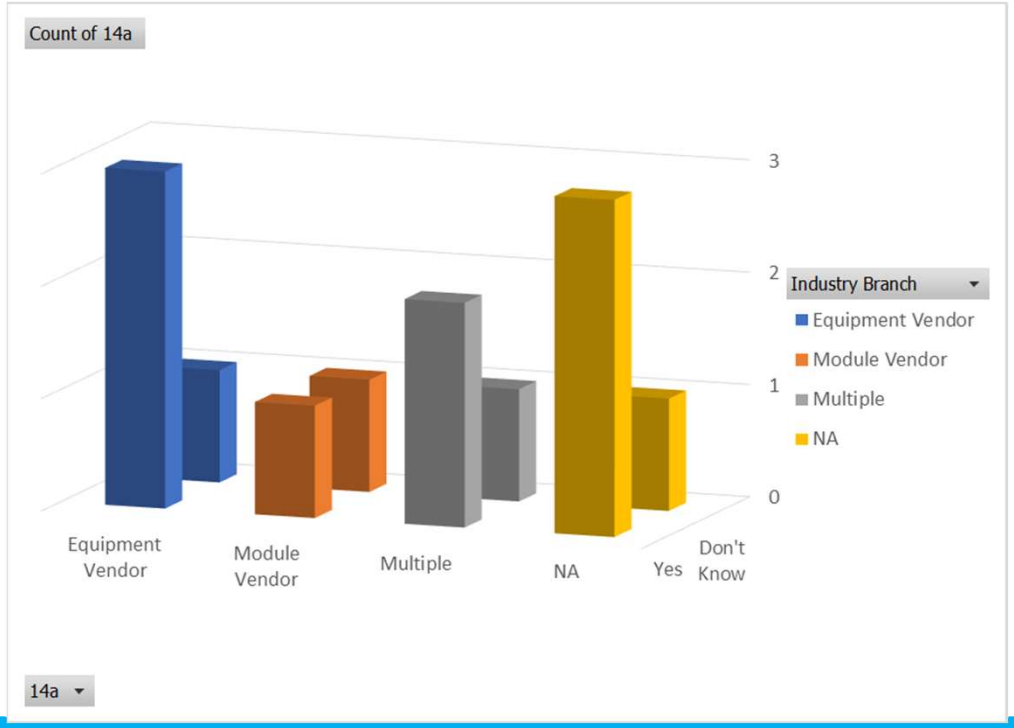
Background -Today, physical management interfaces are slow and do not have headroom to carry other required data that are not management related (e.g. GCC management data in an OTN module).

Future physical interfaces for management communication and for data communication will still be sparse in pins.

Communication interfaces should support multi-purpose high throughput data communication (i.e. tens to hundreds of Mbps) to cover both management and other communication needs

.Question - Is this a use case for your company?

9 – Yes, 4 – No, 0 – Don't Know



Question 14b - Communication Speed and Purpose

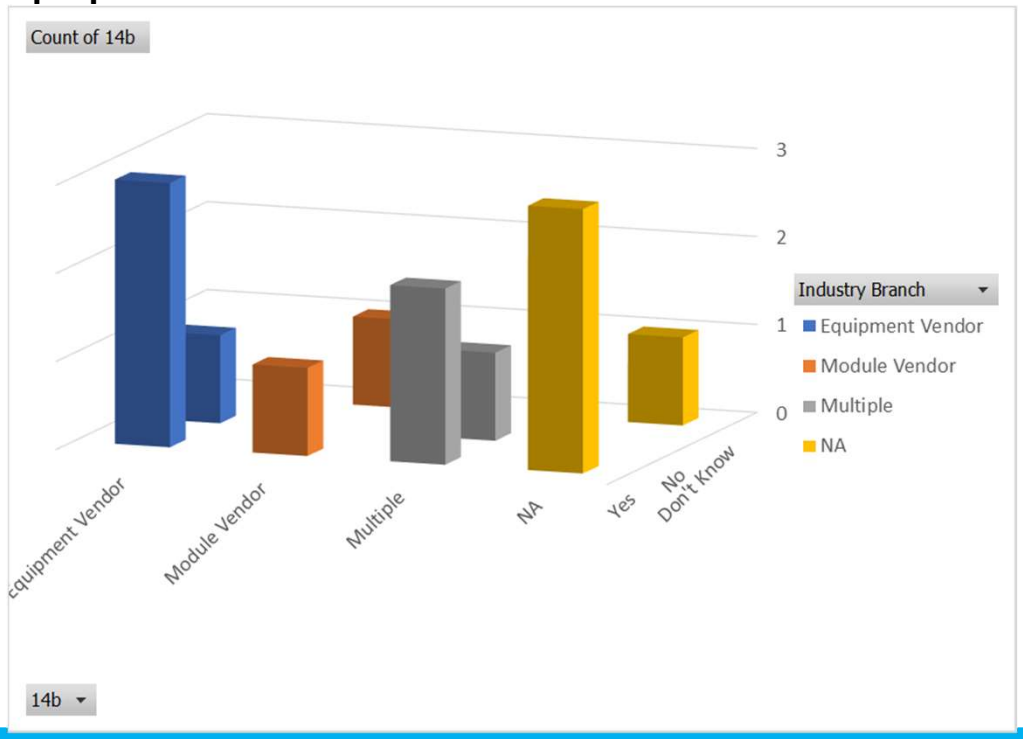
Background -Today, physical management interfaces are slow and do not have headroom to carry other required data that are not management related (e.g. GCC management data in an OTN module).

Future physical interfaces for management communication and for data communication will still be sparse in pins.

Communication interfaces should support multi-purpose high throughput data communication (i.e. tens to hundreds of Mbps) to cover both management and other communication needs.

Question - Would you agree to multi-purpose communication?

9 – Yes, 2 – No, 2 – Don't Know



Question 14c - Communication Speed and Purpose

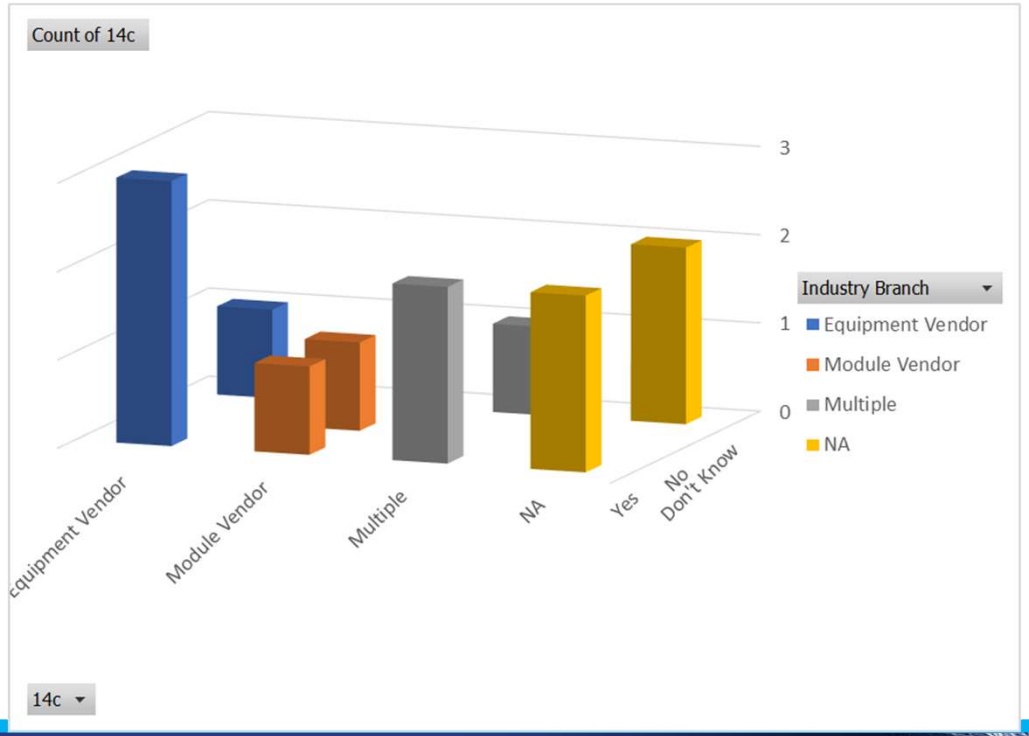
Background -Today, physical management interfaces are slow and do not have headroom to carry other required data that are not management related (e.g. GCC management data in an OTN module).

Future physical interfaces for management communication and for data communication will still be sparse in pins.

Communication interfaces should support multi-purpose high throughput data communication (i.e. tens to hundreds of Mbps) to cover both management and other communication needs.

Question - Would Ethernet be ok as the basis for multi-purpose communication?

8 – Yes, 1 – No, 4 – Don't Know



Question 14e - Communication Speed and Purpose

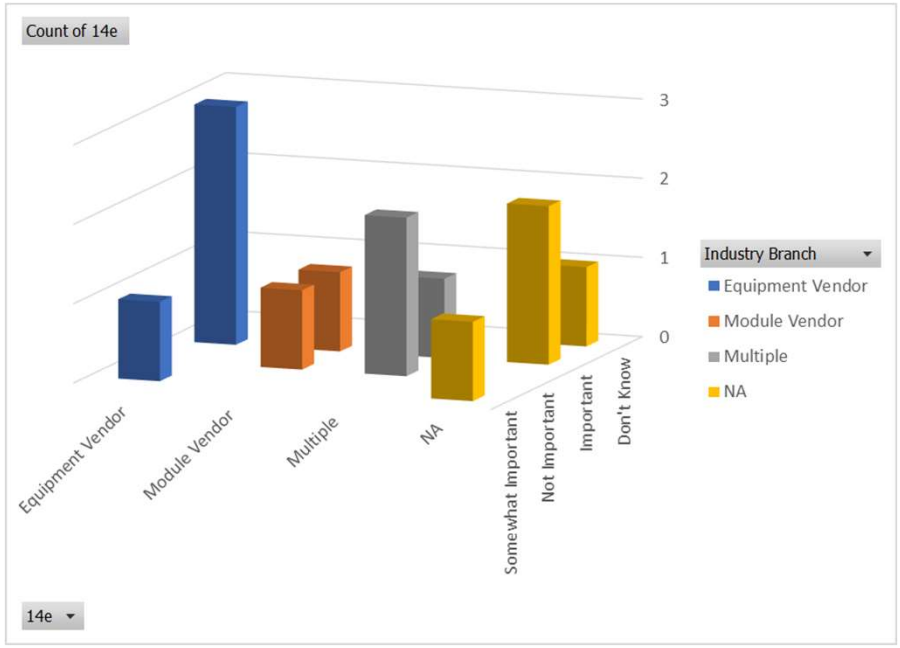
Background - Today, physical management interfaces are slow and do not have headroom to carry other required data that are not management related (e.g. GCC management data in an OTN module).

Future physical interfaces for management communication and for data communication will still be sparse in pins.

Communication interfaces should support multi-purpose high throughput data communication (i.e. tens to hundreds of Mbps) to cover both management and other communication needs.

Importance

7 – Important, 2 – Somewhat Important, 3 – Not Important, 1 – Don't Know



Question 14 - Communication Speed and Purpose

14 a/b/c Comments:

- There would likely be several choices. Just pushing forward a name like Ethernet without providing an estimate of the associated complexity doesn't see appropriate at this time. There may be much better examples found in supercomputers and other high performance clusters.
- data collection and analysis (ex: through AI or other mechanisms) of optical network features, and real-time QAM constellation monitoring require higher speed of multi-purpose communication interfaces.
- Not many low cost microcontrollers have the ethernet port. However, ethernet port implementation is commodity. We should promote MCU vendor to add ethernet port to MCUs.
- As form factors evolve, there will likely be multiple interfaces that make sense depending on the use case. Ethernet is a good option for some modules, but will not work for all.
- SGMII or PCIe
- Goes back to prior comment. This seems like adding significant overhead/effort/cost for modules that do not need this functionality
- May not want high performance interface on all modules but require the option where needed.
- Prception is that much of the "slowness" is not related to the I2C bus, but on internal "settling times" and inefficient ways to access the data needed. If Ethernet is an option, it is certainly valid but it needs a well planned industry adoption cycle. MDIO might be useful as an intermediate solution by begginning with CFP2 form factor.

Question 14 - Communication Speed and Purpose

Importance

14e Comments:

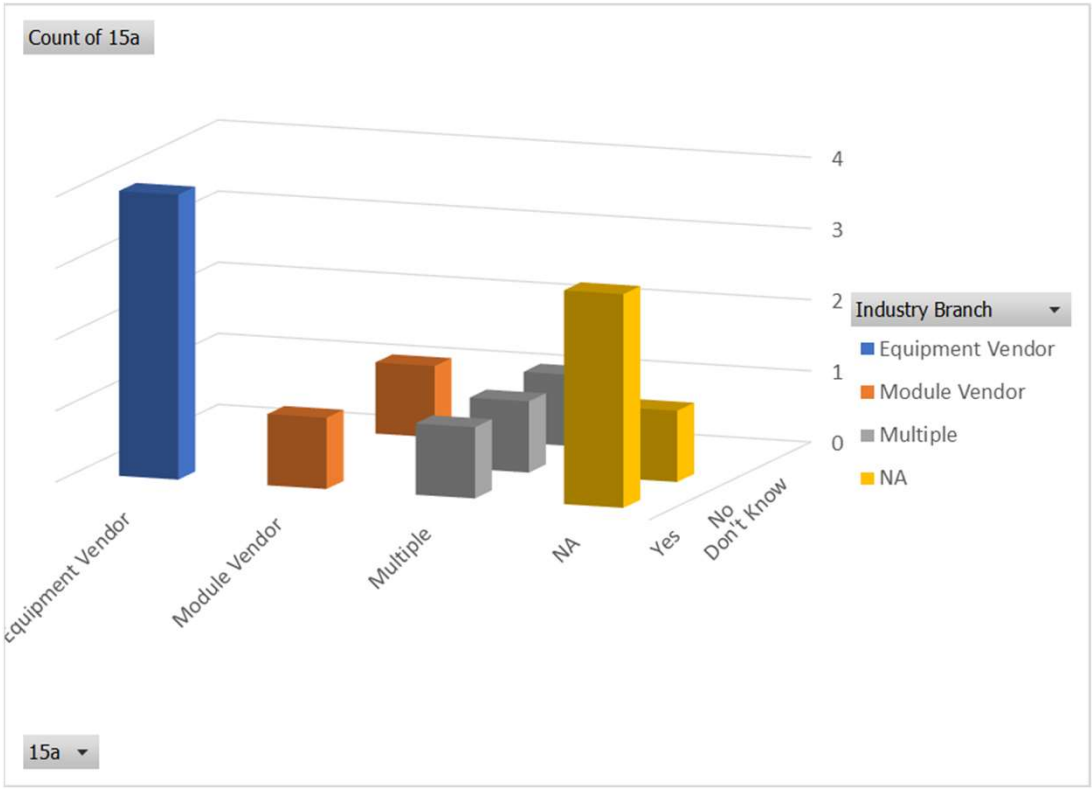
- The profitability of optics is already so low that classic suppliers are exiting the business. The next tier of suppliers is likely less capable in supporting advanced features. Why mandate so much responsibility and cost be entertained by what is considered the most commodity portion of a networking solution (the PHY) ? Are the system priorities to further raise the cost, power dissipation, complexity and development time for what is a media converter (electrical to optical) ?
- Hardware interface right now is an bottleneck in high performance modules and multi-lane modules. This problem becomes more servious for CPO type of modules.
- Goes back to prior comment. This seems like adding significant overhead/effort/cost for modules that do not need this functionality
- Ethernet is certainly future safe. Current DWDM interface is much slower than SW likes it to be. MDIO may be good enough.

Question 15a - Message Based Protocols

Background -One key technique for form factor independent management would be to use functionally oriented message exchange instead of register file access i.e. “setupDataPath(a,b,c, ...)”.

Question - Is this a use case for your company?

9 – Yes, 2 – No, 2 – Don’t Know

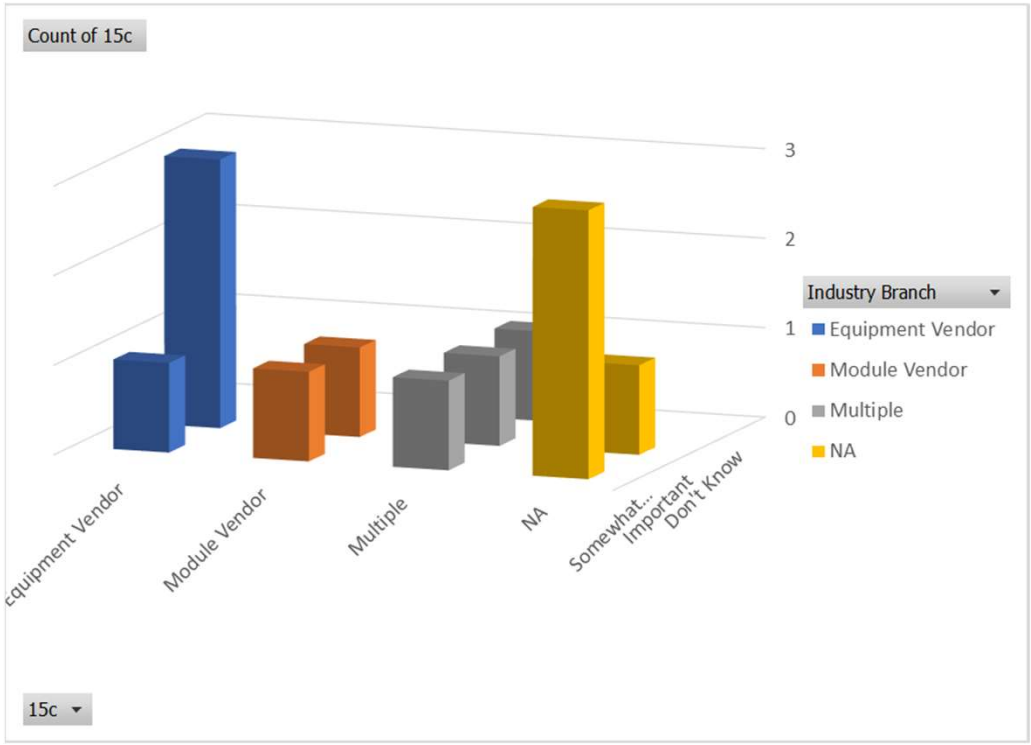


Question 15b - Message Based Protocols

Background -One key technique for form factor independent management would be to use functionally oriented message exchange instead of register file access i.e. “setupDataPath(a,b,c, ...)”.

Importance

6 – Important, 6 – Somewhat Important, 1 – Don’t Know



Question 15 - Message Based Protocols

15a Comments:

- Unless the approach is scalable, i.e. low cost and size for low cost modules and higher cost and size for highly functional modules, the concept of one size fits all should be abandoned and the focus should shift to only management of the highly functional modules.
- A larger automation of tests requires to have more user-friendly management interface and a more functionally-oriented message exchange.
- CDB invented in CFP MIS and used in CMIS is a backward compatible protocol to allow message exchange and register co-exist. It should be considered when industry moves to next stage.
- This is a good technique for access and our preferred option going forward but not the only way to access. These could be designed with models using modelling tools, for example UML.
- Wholeheartly. It would simplify by having the command/response instead of deailing with bits.
- This approach will reduce the behavior difference that generally are difficult to specify

Importance

15c Comments:

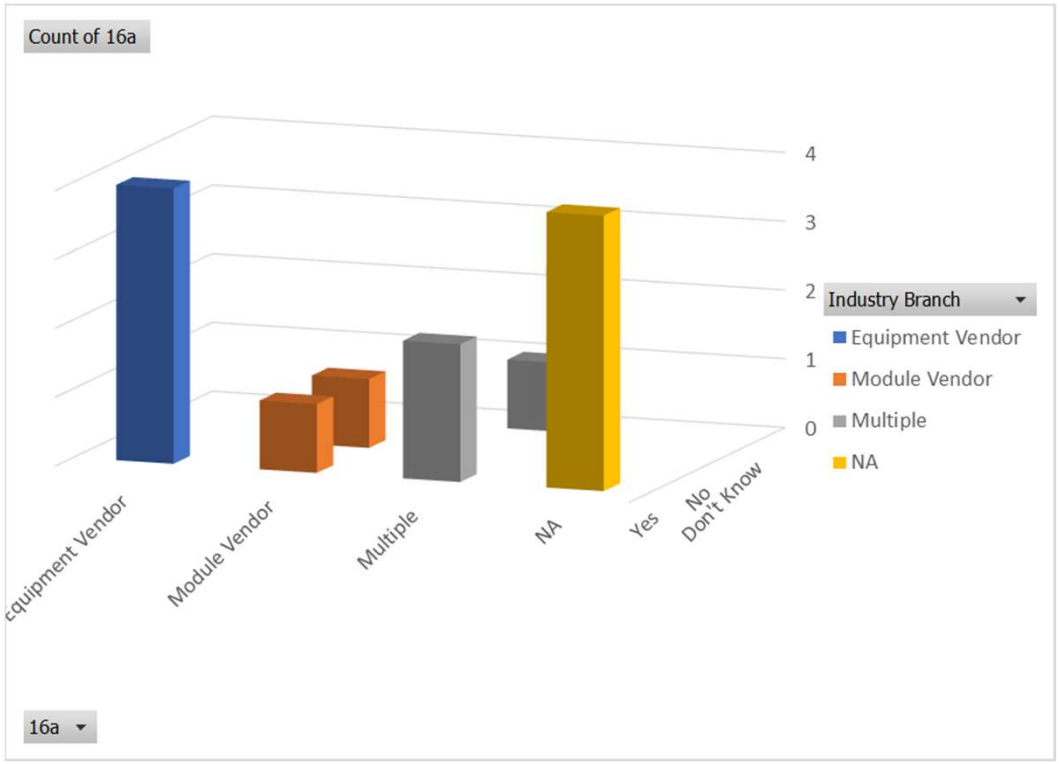
- With more functionalities and new innovations, register map cannot meet the demands. Message exchange is a great way to promote innovation and quick support. It is also a good way to enhance handshaking and security.
- Message based is the way forward for a lot of features: auto-discovery, simplification of integration, access to unique features ...

Question 16a - Standards Based Mechanism that Allows Incorporation of Vendor Specific Module Extensions

Background -Vendor extensions to standard register memory maps such as CMIS require definition of additional registers in address spaces assigned for vendor use. Using an extensible message catalogue or similar approach, a standard approach to defining new vendor specific messages might be possible.

Question - Would you prefer that there is a standard way for a module vendor to "advertise" and define vendor specific messages?

11 – Yes, 1 – No, 1 – Don't Know

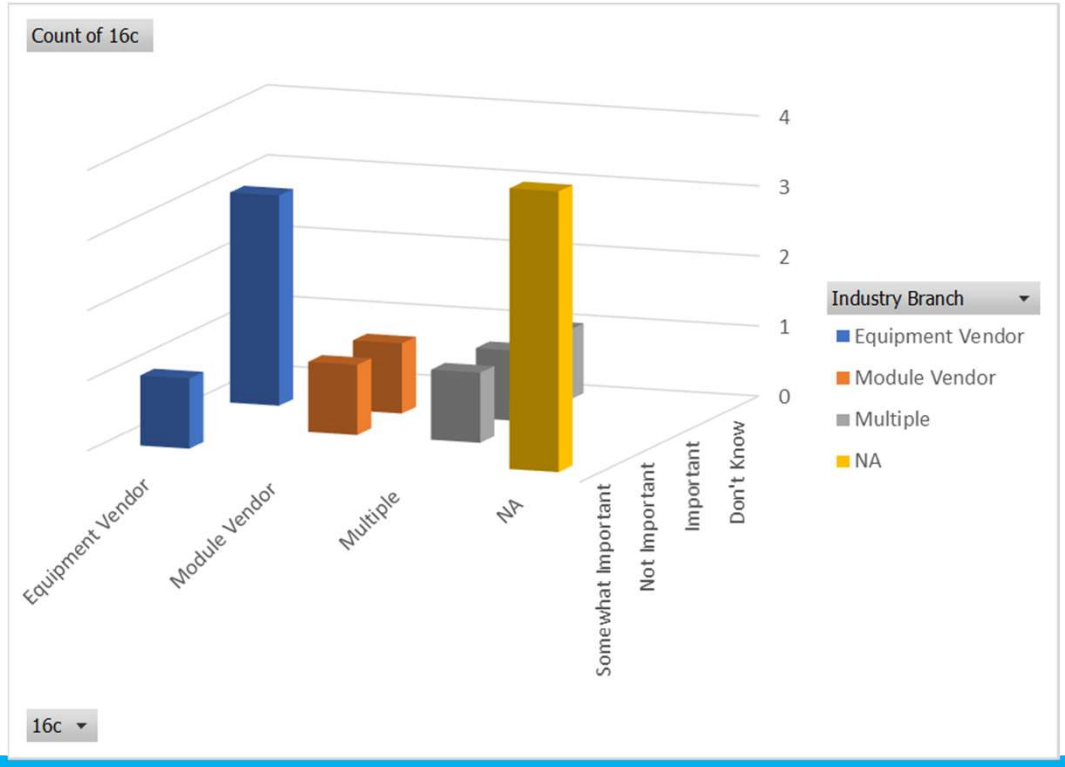


Question 16c - Standards Based Mechanism that Allows Incorporation of Vendor Specific Module Extensions

Background -Vendor extensions to standard register memory maps such as CMIS require definition of additional registers in address spaces assigned for vendor use. Using an extensible message catalogue or similar approach, a standard approach to defining new vendor specific messages might be possible.

Importance

5 – Important, 5 – Somewhat Important, 2 – Not Important, 1 – Don't Know



Question 16 - Standards Based Mechanism that Allows Incorporation of Vendor Specific Module Extensions

16a Comments:

- Vendor specific extension must be exposed using a standard approach and should be easily identified.
- We like the infra-structure idea and agree on the need for vendor specific messages. Different suppliers may have their own set of features that would not be advertised in the generic spec.
- Such capability is State-of-the-art in message based implementations. Vendor extensions allow a very unique feature or additional debug methods to be possible when needed. With messages, this should be much easily achieved as compared to the restricted register spaces.

Importance

16c Comments:

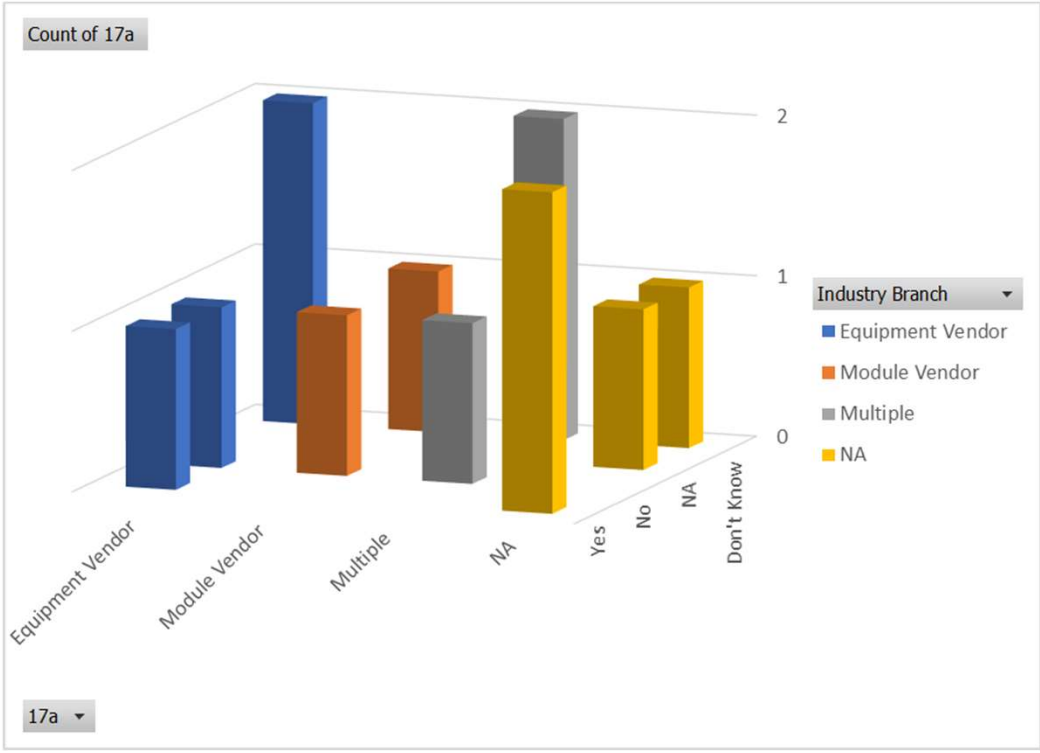
- In general vendor specific should be avoided

Question 17a - Library or Protocol

Background -Host SW development can be alleviated either by an API or SDK (supplied by vendor) running on the host platform, a carefully designed message catalogue (without vendor dependence) or possibly a hybrid

Question - What would you prefer?

3 – Yes, 3 – No, 6 – Don't Know, 1 - NA

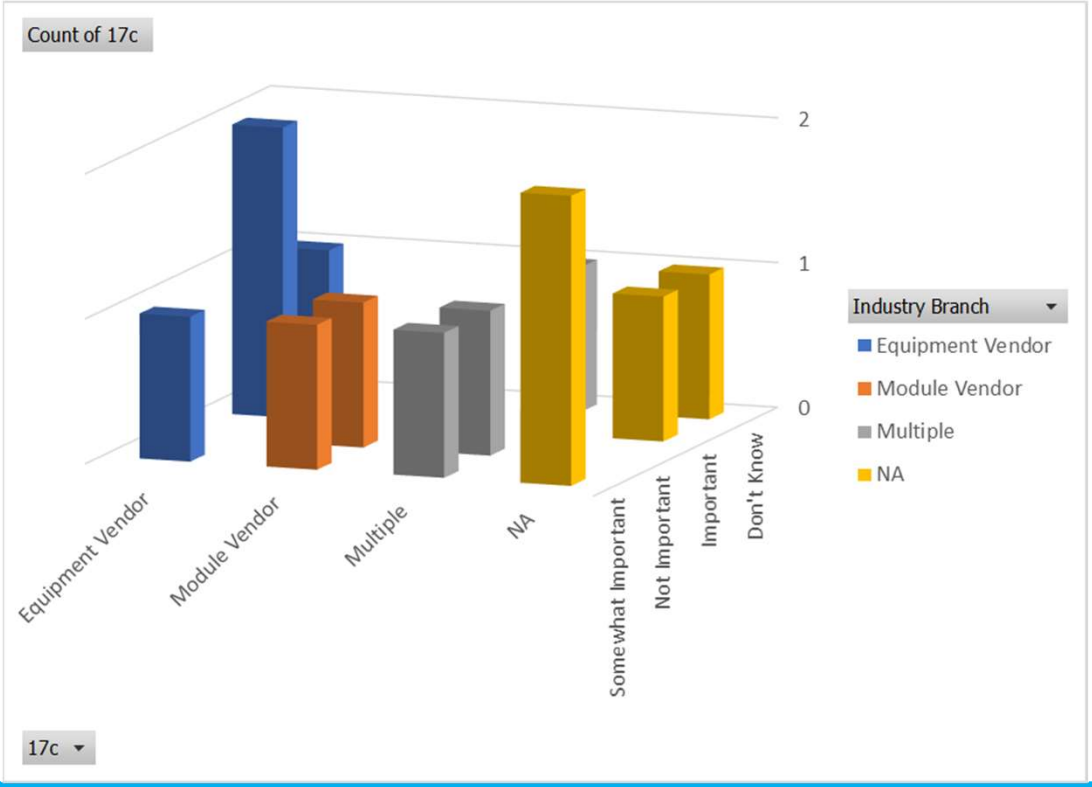


Question 17c - Library or Protocol

Background -Host SW development can be alleviated either by an API or SDK (supplied by vendor) running on the host platform, a carefully designed message catalogue (without vendor dependence) or possibly a hybrid

Importance

3 – Important, 5 – Somewhat Important, 2 – Not Important, 3 – Don't Know



Question 17 - Library or Protocol

17a Comments:

- No appropriate selection possible. We would select the message catalogue
- Carefully designed message catalogue prefered
- The answer choices don't seem to fit the question.
- Standard API should be the good choice for that.
- It saves module vendors time and resource too. Vendor specific API or message exchage promotes innovations.
- we would prefer SW development by an API
- We see benefits to both approaches and want to consider more.
- Difficult to respond with yes/no to this question. Either way could work. Interface to SDK needs to be stable, otherwise we are only shifting the workload. In genera,l we do not have good experience with SDK. Might become a “no” if SDK changes frequently.

Importance

17c Comments:

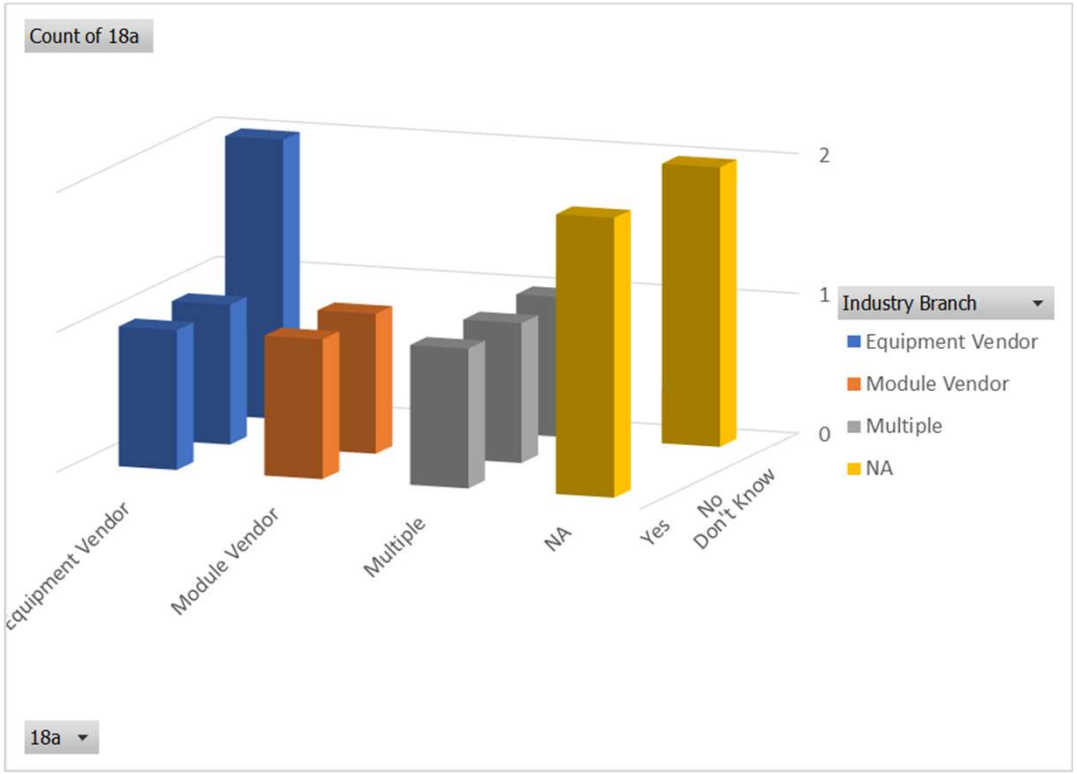
- One way or another is needed. Preference is probably going to be a stable API.

Question 18a - Reference Code for Host/Module Message Parsing

Background -A message protocol based on a standard message catalogue might enable the use of tools that can generate C code to package and parse messages on both sides of the Host/Module interface.

Question -Would you be interested in using generated C/C++ code to package and parse messages exchanged in a message based MIS protocol?

5 – Yes, 3 – No, 5 – Don't Know

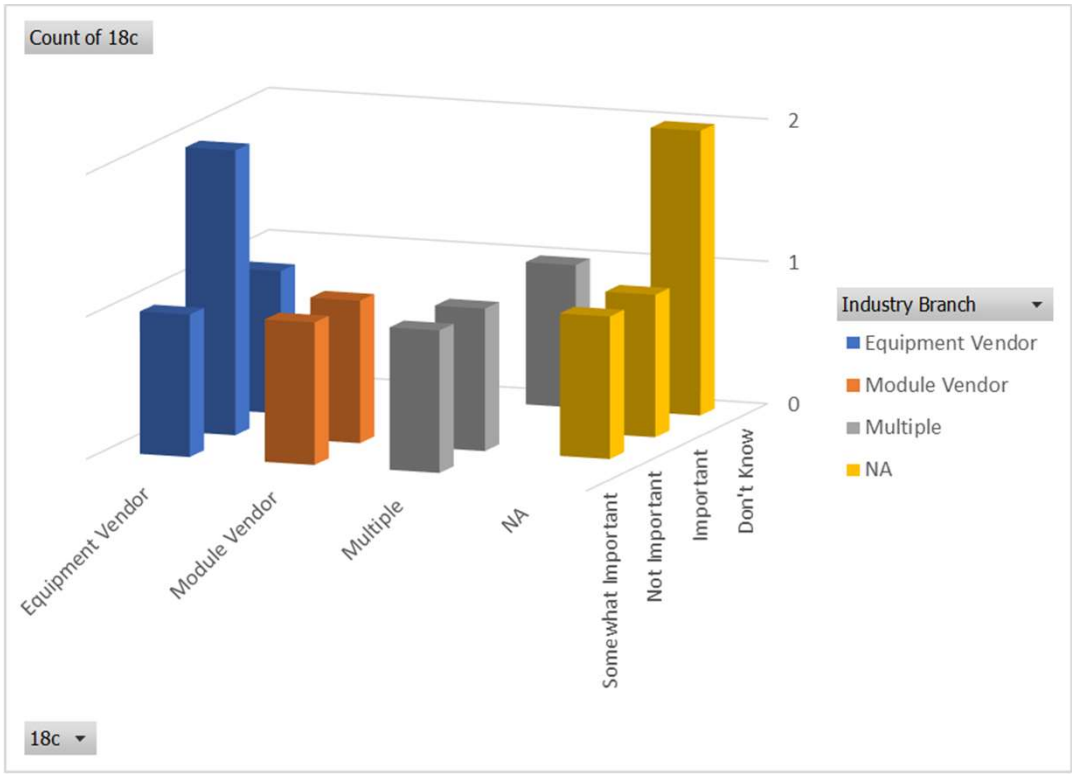


Question 18c - Reference Code for Host/Module Message Parsing

Background -A message protocol based on a standard message catalogue might enable the use of tools that can generate C code to package and parse messages on both sides of the Host/Module interface.

Importance

2 – Important, 3 – Somewhat Important, 5 – Not Important, 3 – Don't Know



Question 18 - Reference Code for Host/Module Message Parsing

18a Comments:

- Depends on licensing terms
- This seems a far stretch. Reference code can be the intermediate step.
- Would like the process to be automated by suppliers as much as possible.
- That's one way, but others would work too. If it is a stable standard and message based CMIS, then there is not much additional value in additional code. Since the quality of the code is not under control of the system vendor, it is harder to integrate in the host software and may result in many different codes to manage.

Importance

18c Comments:

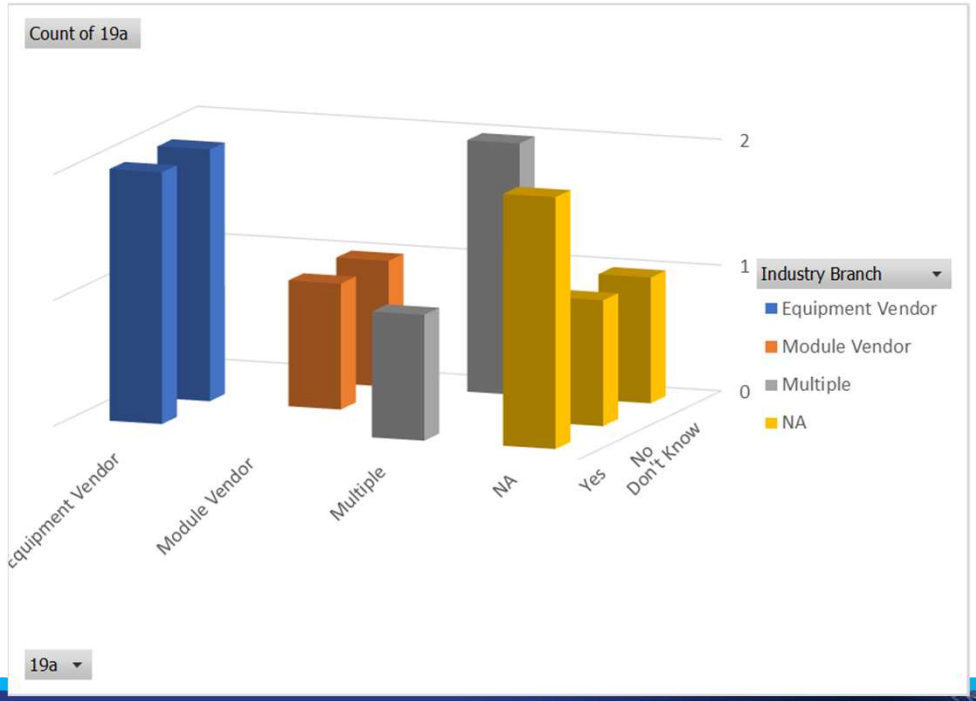
- "There are other industry initiatives working on that subject too:https://github.com/Telecominfraproject/oopt-tai utilizing https://github.com/opencomputeproject/oom/blob/master/optoe/optoe_doc "

Question 19a - Paradigm Shift

Background -Most likely a paradigm shift in the way we create management interface specifications will be required to achieve "One Management" for complex modules. Based on preliminary work, the survey team foresees a complexity-driven paradigm shift away from management data manipulation (in register maps) towards functionality-oriented management commands or notifications (messages). A second, modularity-driven paradigm shift could move us away from hardware-driven fully integrated “management interface specifications” towards well-separated and more software-oriented specifications of application-level management functionality (commands, notifications, data or object model) that can be ported to different and separately specified management communication interfaces as needed. Shifts like these require initial investments for future benefits.

Question -Are you willing to have a larger investment on your first complex module to have a smaller investment beyond that module?

5 – Yes, 4 – No, 4 – Don’t Know

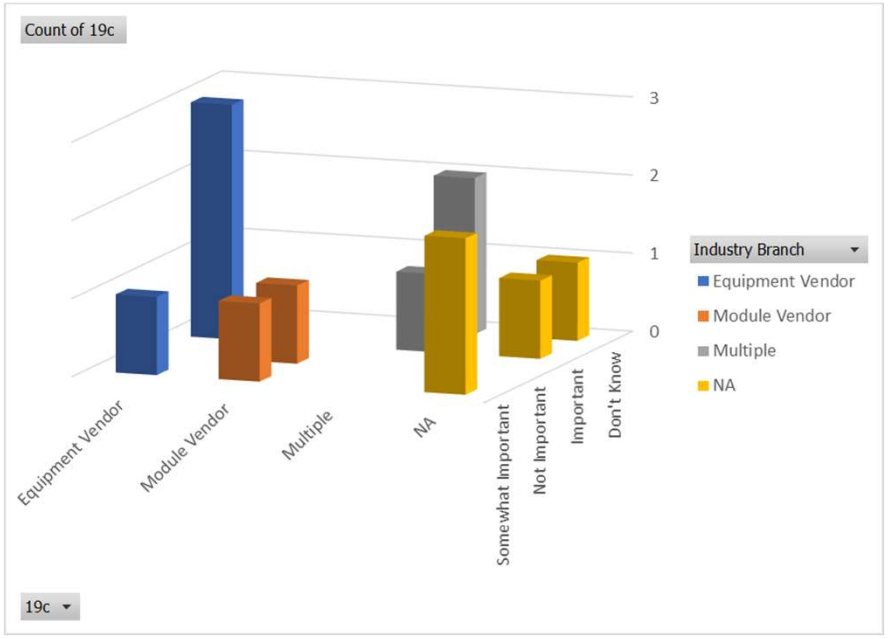


Question 19c - Paradigm Shift

Background -Most likely a paradigm shift in the way we create management interface specifications will be required to achieve "One Management" for complex modules. Based on preliminary work, the survey team foresees a complexity-driven paradigm shift away from management data manipulation (in register maps) towards functionality-oriented management commands or notifications (messages). A second, modularity-driven paradigm shift could move us away from hardware-driven fully integrated “management interface specifications” towards well-separated and more software-oriented specifications of application-level management functionality (commands, notifications, data or object model) that can be ported to different and separately specified management communication interfaces as needed. Shifts like these require initial investments for future benefits.

Importance

5 – Important, 4 – Somewhat Important, 1 – Not Important, 3 – Don’t Know



Question 19 - Paradigm Shift

19a Comments:

- Yes, if there is a common consistent request from the market to be standard compliant over multiple product generations.
- The promise of "a larger investment on your first complex module to have a smaller investment beyond that module" seldom seems to work. It remains a sustained large investment in subsequent efforts.
- We support the first paradigm shift (register map to messages). For the 2nd parameter shift: more information is needed to assess what 'larger investment' signifies.
- The decision to have a paradigm shift will be driven by the application needs. We think that we have reached the point where the complexity of the modules drives the need for change. We think that this will drive down the total cost over several generations of modules.
- It would be a good long term investment to ease things in the future.
- The added complexity also needs some unifying standard before asking suppliers to commit to conversion; if comprehensive enough then the investment may be OK.

Importance

19c Comments:

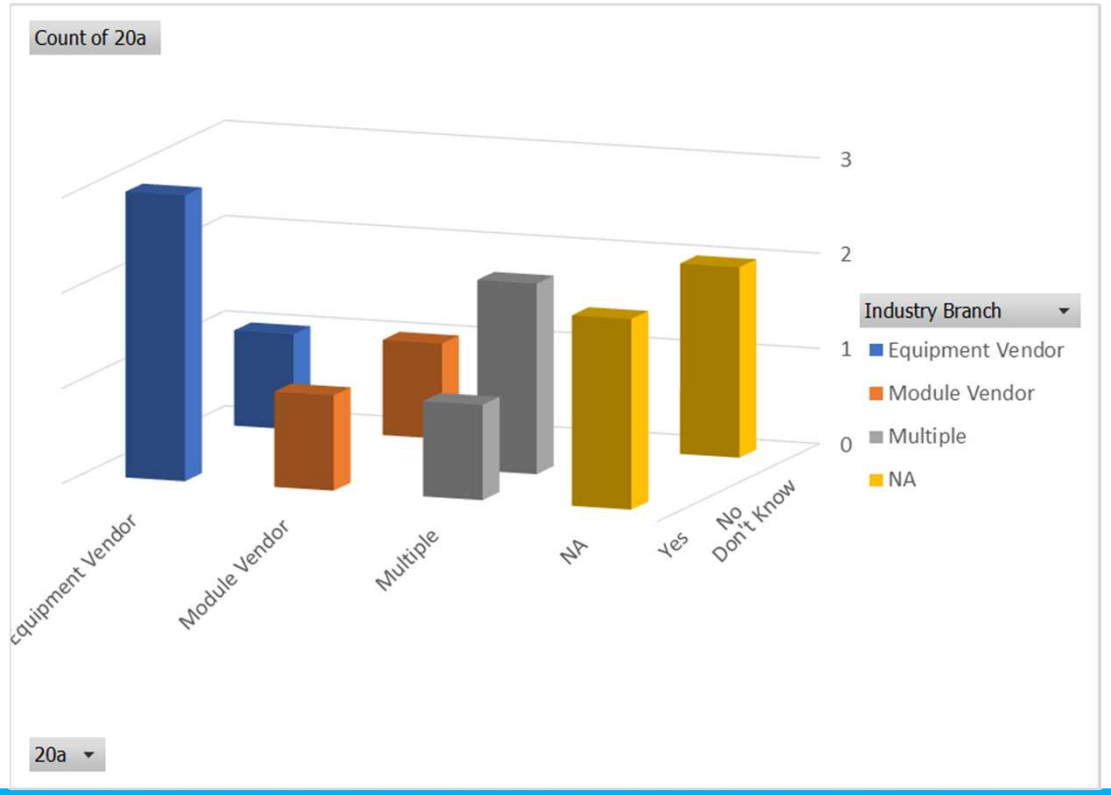
- Note that 1) register map has a long history so backwards compatibility of new protocol is always a winning ticket; 2) optical modules span a huge range of performance, application, and cost structure. A backward compatible new protocol like CDB helps the paradigm shift.
- Shift to messages is important
- Main priority is to avoid additional cost and overhead for "simple" datacom modules, and to avoid having fragmented competing specs for "simple" datacom modules. Creating a brand new framework for management interface and then stating a goal to "unify" across all form factors seems likely to drive additional cost and overhead for "simple" modules. If the intent is for this effort to focus only on coherent modules, that should be clearly stated.
- "Yes, assuming message based solutions will carry on.If not proceeding, there are system burdens: I2C bus is an issue to be used for interrupt.MDIO allows easier interrupt management: SW does not need to get out of the interrupt context to perform a read."

Question 20a - New Direction in MIS Standardization

Background -A new MIS paradigm will involve work from standards bodies, module vendors and systems providers.

Question -Are you willing to work with a separate MIS standardization body that is independent of the form factor hardware MSA standardization bodies? This is similar to the approach with the CMIS, but based on an exchange of functionally oriented messages.

7 – Yes, 2 – No, 4 – Don't Know

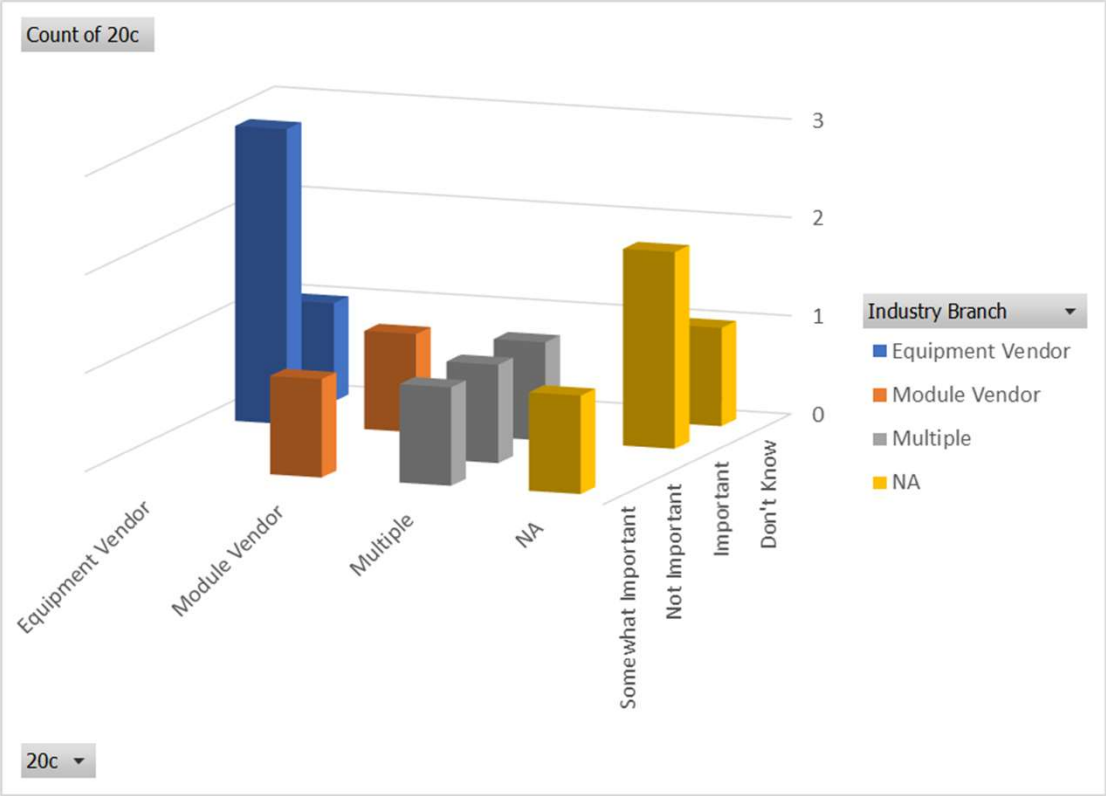


Question 20c - New Direction in MIS Standardization

Background -A new MIS paradigm will involve work from standards bodies, module vendors and systems providers.

Importance

7 – Important, 3 – Somewhat Important, 1 – Not Important, 2 – Don't Know



Question 20 - New Direction in MIS Standardization

20a Comments:

- Since the market is expected to move towards co-packaging of ASICs and optics, there should not be a need for future exotic pluggable module capabilities and protocols, as that management layer will likely disappear altogether.
- Depends - what actually is the scope of this proposed effort? We like what CMIS is doing. The industry does not need another "shadow CMIS" to compete with CMIS and come up with different overlapping specs - this will be highly counterproductive. If the intent is to create a separate effort (similar to the current C-CMIS one already kicked off within OIF) to focus on additional specs for cohernet modules, in full alignment and cooperation with CMIS, that we can support.
- We are already involved and plan to continue this work in the future.
- What's a realistic way forward without boiling the ocean?
- Depends on the nature and scope of the effort

Importance

20c Comments:

- Again, depends on scope - not clear from this survey although we can speculate.

Thank You