Analysis of MPAI-MMC V2 Draft

Stephane H. Maes[1]

September 16, 2023

Abstract

This short paper provides a short list of comments in answer to the request for public comments for the MPAI MMC (Multi-modal conversations) V2.

Our concerns can be grouped in terms of questions on business value, on the architecture assumptions, the standardized artefacts, and the scope of the MMC use cases. Except for the latter, these comments can probably read, and apply to other drafts published by MPAI (MOVING PICTURE, AUDIO AND DATA CODING BY ARTIFICIAL INTELLIGENCE) and on-going activities.

---

1. Introduction

The paper was also published as [6].

The comments are based on the publicly available overview of the standard draft for MPAI MMC V2 [1] and the presentation requesting for public comments [2]. We have not been able to examine the actual detailed draft specifications if such documents exist.

The goal with these comments is to help, not to discourage the effort. Yet we have some concerns about what is actually being standardized, and its value.

In our view, it is unclear what exactly is standardized, and if it is well aligned with software development, AI and cloud services as encountered today. In particular we are concerned that the work tries to standardize systems and system architecture, the old way of networks functions and hardware components, instead of protocols, schemas format, or interaction/data formats, and flows to use them as is encountered in Software development, internet/web/cloud; with APIs or services being sometimes the way to go and sometimes not.

We will also argue that some functionality, and use cases, may be missing, something that may also result from problematic architectural choices. Aspects may also come from a more archaic view of the way that AI capabilities are made available versus the approaches that have emerged lately.

Although we have not reviewed the other standards, we suspect some of the issues discussed below apply to some of the other standards ready for comments and on-going activities.

2. Standardizing Systems or Format, Protocol and Associated Processes

[1] IFS, s.h.m.pub.2@gmail.com.

Cite as: Stephane H. Maes, (2023), "Comments to MPAI-MMC V2 Draft – Request for Public Comments", https://osf.io/hj6pw, https://shmaes.wordpress.com/2023/09/16/comments-to-mpai-mmc-v2-draft-request-for-public-comments/, September 16, 2023. Also published as: Stephane H. Maes, (2023), "Analysis of MPAI-MMC V2 Draft", https://shmaes.wordpress.com/2023/09/16/comments-to-mpai-mmc-v2-draft-request-for-public-comments/, September 16, 2023

Standardizing all aspects of the functions involved in a system, was and is still encountered in telecom standard activities, and hardware standardization. Indeed making two systems, e.g. network equipment or chips / hardware, from different vendor interoperate / interwork requires such complete standardization. If the two systems must plug into one another, the plugs must be fully specified and agreed upon. Any discrepancy may require building a new system.

It is not how software works. In software, once APIs are published for a suitable technology binding (e.g., HTTPS protocol of REST for example), it is very simple for a developer to "plug into", and rely on the service. Any misunderstanding, or evolution, is corrected, or caught up, by say transforming the API (at the client, the server or in between). Functionality differences may have to be compensated for by calling or composing other services.

Key examples pf the challenges encountered by this type of standardization followed by MPAI include the tensions and lack of adoptions of the Open Mobile Alliance (OMA) standards, or telco 2.0 network function exposure to developer failures vs. adoption of on device enablers and their APIs with Apple apps store and Google play store, or the tensions between 3GPP standards and IETF standards.

Standardizing protocols rather than AI or services is how software is developed these days, and it is how one typically rely on cloud services provided by cloud providers.

Note also recent examples where open source activities attempted to de facto standardize services. OpenStack [4] and related network initiatives are good examples. Despite the great achievements, and a huge community involvement, the projects have failed on main of their main objectives and have not dented or 'standardized" the clous services. Recent buzz words like super clouds will most probably go the same way, because victim of related ambitions, that, unfortunately, typically amount. to standardize a lowest common denominator sets of cloud services across cloud providers [5]; it is inefficient, impossible to maintain up to date, and often not what developers seek in cloud services. Note that some companies are doing the same to define a cloud platform for their own services or provide simple AI subscriptions to their customers who are adding AI to the application feature they use. We, IFS, are a good example of it. But that is quite different from standardizing. It is now about selecting and curating interfaces to a subset of AI services capabilities worth abstracting across a few selected vendors or versions and capabilities. That is not standardizing, and therefore it may be viable with totally different value propositions.

3. The new AI era

In fact, AI has been around for a long time. The AI cloud development, besides technology fads like LLM with ChatGPT and consort, is the democratization of data storage, training and executing infrastructure etc. so that people can build AI capabilities that would require huge investment in the past.

And now, the latest evolution with ChatGPT has taken things one step further with now the ability to build AI systems and AI applications with simple tools (AI (cloud) services) that are integrated, or composed with their APIs. With this developers, or even laypeople, with say prompt engineering, can build applications with simple tools, CS common practices, or even just low code no code tools.

Examples are Azure / Microsoft cognitive services, AWS AI, and many other AI (cloud) services and service framework from cloud providers. The services may run on specialized hardware servers provided by AI chip / hardware vendors, or involved 3rd party services like OpenAI services.

For the rest, some services can run on end user devices, including some of the latest cell phones, typically with predefined models.

4. Business value of such standards

Based on the considerations above, it is worth asking who will adopt the MPAI / MCC standard, and who are usefully served by it. It is not immediately obvious that the stated ecosystem will benefit with what is proposed.

As mentioned earlier, e.g., with super cloud ideas, there is little chance that cloud providers who are the engines of AI growth, and betting heavily on it for their own renewed growth, will standardize their services details, let alone their APIs. It's just what it is… and it hasn't happened with cloud services. Fat chance that it will happen with AI (cloud)services.

Most AI platforms are either hardware accelerated / optimized servers that can be used to provide AI services, or applications using these services. They are not exactly served by the proposed standards, as hidden / abstracted beyond the provide service-layer AI services. Again developers of AI based apps will rather only use the APIs of these services.

Sure a framework could implement the MPAI proposal, but it is questionable that multiple such frameworks and services would develop, as it would not be adopted by say a cloud providers or an AI vendors. Also, most have already their framework, why use this. There is little need of interoperability. As is the specification provided by MPAI, and MPAI-MMC, the specs may be suitable for a AI service provider or hardware vendor, as a design specification for implementation and for its exposed API and services. But that is not a standard, at least as we would have expected and as we read in [1,2].

We recommend therefore that some thought be put into reassessing the goal of the activity, and the assumptions of the architecture, controller, notions of / assumptions about components are their life cycle, etc.

5. The controller, the lack of orchestration specification or component API specifications

Although we may have missed it, it seems that there is no specification of a programmatical way to compose or orchestrate the modules/components, as are their lifecycle. Such role is left to, so far and to knowledge, unspecified controlled, and we assume maybe some magic configurations. It may seem ok for cloud services where that is done by the cloud provider. But even then such services may require further management to be useful / tuned to a problem or context.

Components from an store are not enough. Without programmatic composition, installation should be associated to configuration specifying how its input and output are connected / piped together. It should be something that developers can access and change.

More importantly, we believe that one wants a programmatic way to chain / compose the components with the ability to dynamically change the piping possibly based on the conversation status, detected intent, context, or use case. That is missing so far.

Installation and configuration is not specified, as far as we know, and it should be, at least in terms of assumptions and what can be done. Again, if we follow the cloud service trend, a model like terraform or Yaml/helm could be considered, and in any case supported.

Until this is addressed, the standards proposed in MPAI MMC and MPAI in general are probably not sufficient for any relevant interoperable solution to be built on them.

6. Pipelines or APIs

Following the arguments above, we would argue that the cloud service model exposes API that allows the requester to query particular actions on streamed/piped data or payload, instead of just chained default processing that seems proposed so far.

Yes, there are ways to do that inline with the data, as envisaged with some models for example for MRCP[3], but that does not seem to be specified by MPAI so far either.

Until this is address the proposed specification do not seem aligned with current trends in the space, making adoption of the specifications unlikely.

7. MLOps, and service management / training, etc.

While it could be argued to be a backend considerations, instead of user facing concept, the specs do not allow selection of a models, training, tuning or say reinforcement of the models etc. We believe that it is a significant weakness of the standard. Developers using AI, may have to perform such (re)training, tuning and optimization. If the standard is aimed at developer these capabilities are missing.

For many applications, developers / app providers may need to adapt/train/tune their models, e.g., to content of documents, accents, context, domain of usage, audio channel, noise etc. Selecting such models (possibly based on incoming requests) is not part of the standard as far as we can see. APIs or guidance to do this is missing as are also model management, transfer learning, MLOps etc. Assuming that ever trained model is a different component may be a questionable and rapidly unmanageable approach, in particular if such models are to be private, built on public services or base models.

Until this is addressed, the standards proposed in MPAI MMC and MPAI in general is not sufficient for any interoperable solution built on them.

8. Intent / context gaps

In the domain specific of MMC, we believe that the specification is missing the idea of multiple systems cooperating to determine input and mixed initiative next step. There is no discussion or concept of intent or

information needed to fully define an intent (e.g. sets of forms defining what is needed for an intent to be fully defined and executable/processable). The same holds for context.

AI NL conversation developers would need to directly be able to define such concepts and possibly to exchange so far determined context and intent across multiple components. It is a major limitation for a standard aiming at helping developers to build conversational applications!

Until it is added, we question the value offered by the standard for the development of true AI-based conversational applications.

9. Conclusions

In this short paper, we provide comments to MPAI MMC V2 specifications request for comments, as well as the larger MAPI project. We would strongly recommend that these be considered before formal release of the specifications as a viable standard.

As we indicated there are issues with the objectives of the initiative, and associated business values and in our view challenges with the architecture approach taken a well as functionality gaps. A lot of the issues relate to how software is meaningfully standardized, it often isn't, and how AI services are currently offered, especially by cloud providers.

Gaps should be addressed before publication of the specifications. The larger MPAI framework concerns should be separately seriously addressed and discussed by MPAI.

We are certainly willing to expand on the comments, and help considers way to mitigate the challenges identified.

References

[1]: MPAI, (2023), "An overview of Multimodal Conversation V2", https://mpai.community/2023/09/04/an-overview-of-multimodal-conversation-v2/. Retrieved on September 5, 2023.

[2]: Leonardo Chiariglione, "MMC V2 presentation" as part of "MPAI published 3 Calls for Technologies and 5 published standards for Community Comments", Zoom call, September 5 2023.

[3]: Daniel C. Burnett , Saravanan Shanmugham, (2014), "Media Resource Control Protocol Version 2 (MRCPv2), RFC 6787", IETF, https://datatracker.ietf.org/doc/rfc6787/.

[4]: "OpenStack: Open Source Cloud Computing Infrastructure", https://www.openstack.org/. Retrieved on September 16, 2023.

[5]: Stephane H. Maes, (2022), "Super Cloud or Meta Cloud: A Problematic Approach", https://shmaes.wordpress.com/2022/11/25/super-cloud-or-meta-cloud-a-problematic-approach/, viXra:2211.0150v1, November 25, 2022.

[6]: Stephane H. Maes, (2023), "Comments to MPAI-MMC V2 Draft – Request for Public Comments", https://osf.io/hj6pw, https://shmaes.wordpress.com/2023/09/16/comments-to-mpai-mmc-v2-draft-request-for-public-comments/, September 16, 2023.