OMNI: Open Mind Neuromodulation Interface for accelerated research and discovery

Bradford N. Roarr, Randy S. Perrone, Fawad Jamshed, *Member, IEEE*,
Ro'ee Gilron, Timothy J. Denison, *Member, IEEE*, Philip A. Starr,
Jeffrey A. Herron*, *Member, IEEE,* David A. Borton*, *Member, IEEE*

Abstract— **Electrical neuromodulation is an approved therapy for a number of neurologic disease states, including Parkinson's disease (PD), Obsessive Compulsive Disorder, Essential Tremor, epilepsy and neuropathic pain. Neuromodulatory strategies are also being piloted for an increasing number of additional indications, including Major Depressive Disorder, Dystonia, and addiction. The development of implantable devices capable of both neural sensing and adaptive stimulation may prove essential for both improving therapeutic outcomes and expanding the neuromodulation indication space. Nevertheless, an increasingly fragmented device ecosystem forces researchers and therapy developers to customize and reinvent data visualization, clinician engagement, and device control software to support individual clinical studies. Each hardware platform provides a unique software interface to the implanted neurostimulator, making pre-existing code from prior studies difficult to leverage for future work -- a hindrance that will expand as device technology diversifies. Here, we envision, detail, and demonstrate the use of a novel software architecture, OMNI, that accelerates neuromodulation research by providing a flexible, platform- and device-agnostic interface for clinical research and therapy development.**

I. Introduction

Over the last decade, the field of neuromodulation research has been dramatically accelerated by the rising availability of human-use implantable research device platforms. Deep Brain Stimulation (DBS) research, in particular, is seeing a proliferation of new investigational devices for use in the field, with more on the way, including the Activa PC+S [1] and Summit RC+S [2], the DyNeuMo Mk-1 [3], and the Brain InterChange System [4]. These research-focused Implantable Neuro-Stimulators (INSs) allow researchers to develop custom computer software for device interaction using manufacturer-provided software development kits (SDKs). Published research enabled through this "SDK-driven" strategy includes studies exploring adaptive deep-brain stimulation (aDBS) for PD [5], [6], essential tremor [7], epilepsy [8], and neuropsychiatric illnesses [9]. However, these first-generation research SDKs are tightly integrated with the INS hardware, inhibiting code reuse across studies. The result of this tightly coupled software/hardware approach is that research software applications are written as monolithic applications, with the device-specific research SDKs called directly from the software, making it difficult to separate reusable code from protocol-specific implementations.

The proliferation of investigational devices creates a fragmented landscape of programming languages and operating-system support, posing significant barriers to future research support. The Medtronic family of INS hardware is illustrative of this problem: whereas the Activa PC+S platform provided a Java SDK for Windows, the next generation hardware, Summit RC+S, moved from Java to C# for the SDK. This demonstrates that even within the same family of INS devices from the same manufacturer, there is limited opportunity for interoperability - each new device requires rewriting most software from scratch.

The landscape of INS hardware devices is further complicated by their unique capabilities. To perform research with new devices, researchers must often translate configuration parameters from a device used to pilot prior work, as INS platforms frequently lack one-to-one mappings for configuration options. There are long-term impacts of these limitations as well. As the field of adaptive neuromodulation matures, replication studies will become increasingly important -- but owing to INS hardware evolution these studies may require rewriting research software to target new device hardware. We anticipate such rewrites will often necessitate a translation between programming languages, as well as a translation of INS configuration options. Previous discussion of the scientific and technical concerns presented by a fragmented research tools space have highlighted the need for

B. N. Roarr is a Senior Research Software Engineer in the Center for Computation and Visualization at Brown University, Providence, RI (e-mail: bradford_roarr@brown.edu).

R. S. Perrone is a Software Engineer in the Department of Neurological Surgery and the Weill Institute for Neurosciences at the University of California, San Francisco, CA (email: randy.perrone@ucsf.edu)

F. Jamshed is a Lead System and Software Engineer in the Department of Engineering Science and the Nuffield Department of Clinical Neurosciences at Oxford University, Oxford, UK (email: fawad.jamshed@eng.ox.ac.uk)

R. Gilron is a Postdoctoral Scholar in the Department of Neurological Surgery and the Weill Institute for Neurosciences at the University of California, San Francisco, CA (email: roee.gilron@ucsf.edu)

T. J. Denison is the Royal Academy of Engineering Chair in Emerging Technologies and Professor in the Departments of Engineering Science and Clinical Neuroscience at the University of Oxford, Oxford, UK (email: timothy.denison@eng.ox.ac.uk)

P. A. Starr is a Professor in the Department of Neurological Surgery and the Weill Institute for Neurosciences at the University of California, San Francisco, CA (email: philip.starr@ucsf.edu)

J. A. Herron is an Assistant Professor in the Department of Neurological Surgery at the University of Washington, Seattle, WA (email: jeffherr@uw.edu)

D. A. Borton is an Assistant Professor in the School of Engineering at Brown University, Providence, RI (corresponding author, email: david_borton@brown.edu)

*Co-Senior Authors who contributed equally to this work.

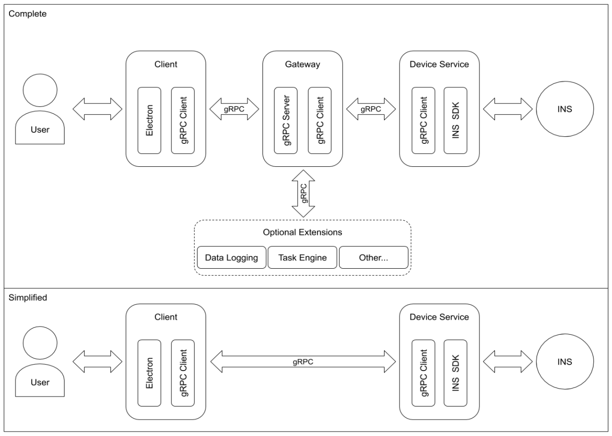

Figure 1. OMNI accelerates DBS research by providing the software for designing rich user interfaces (Client), and interacting with the INS hardware (Device Service). The optional gateway exposes futher functionality via additional gRPC APIs. The complete architecture integrates the optional gateway component. The simplified architecture, where a system developer connects the client directly to the device service, also provides interoperability and software-reuse capabilities beyond the scope of current research system's APIs.

a platform-based approach to solving what will become a key barrier for future work [10].

To mitigate the barriers caused by INS hardware fragmentation, our team is developing OMNI, an INS hardware-agnostic software framework to accelerate research in the electrical neuromodulation space. Our framework relies on gRPC, an industry-developed remote procedure call framework that facilitates software interoperability via networked application programming interfaces (APIs) [11]. gRPC uses the Protobuf interface description language (IDL) to define services, endpoints, and messages used across interfacing software. Protobuf comes with a variety of tools to generate both client and server code for 10 supported programming languages (with more languages supported through the open-source community). Further, by developing OMNI under a design control process and providing this common interoperability framework, independent research sites can efficiently bootstrap their own research protocols in whichever programming language they are comfortable with. OMNI provides an architecture for future investigational device manufacturers to use in new systems, as it provides a robust method for allowing system interoperability without requiring the development of SDKs in varying languages.

In this paper, we present an overview of the architectural design and features of OMNI for use in neuromodulation research. We then describe the current state of software development for OMNI as well as future development strategy, demonstrating feasibility of the platform to allow applications to interface with INS devices through a common interface. Lastly, we show how using OMNI mitigates the software development burden for sites starting DBS research and existing protocols migrating to new INS hardware.

II. THE OMNI ARCHITECTURE

In response to the growing fragmentation of neuromodulation research hardware devices [10], we designed OMNI to be INS device- and programming language-agnostic. OMNI facilitates onboarding of new study teams and implementation of new research protocols by providing generic device APIs that abstract common functionality of INS research systems. OMNI provides turn-key solutions for common needs associated with DBS research protocols, such as neural sensing and data logging.

Our framework, shown in Figure 1, makes use of a singular gRPC-accessible Gateway to provide client applications written in any gRPC-supported language a singular interface for communicating with implanted devices. The client connects to the gateway and the gateway passes messages downstream to the device service and any other auxiliary services. The Gateway is responsible for managing "Device Services," which translates client requests into device-specific API calls, as well as additional standardized functionality via extensions. Each component of OMNI is implemented as a stand-alone process. Implementing components as their own processes makes OMNI more robust against crashes. A crash caused by the user interface in the client cannot crash the device service. In contrast, when the user interface causes a crash in a monolithic application, the whole application crashes, often leaving the INS in an unknown state.

Device Services are the only component in the OMNI architecture that interact with the INS device. The network interface code is auto-generated by gRPC based on common protobuf files to allow the Device Service to be written in any language for which the INS manufacturer provides an API, allowing the body of the device service to be written once per INS device. Device services are responsible for exposing the research functionality of an INS device API as well as ensuring the INS connection remains stable. The "Generic API" provided by OMNI is designed to help researchers answer real questions while maintaining INS device compatibility. By necessity, the Generic APIs obfuscate some of the platform specific features of a given INS device in order to provide a unified interface across INS devices. The protobuf definitions allow device-specific parameters to be appended to Generic API calls. This allows researchers to unlock the full functionality of their chosen INS device. Furthermore, to adhere to industry recommended resource-oriented design principles [12], [13] we have separated the Generic API into the Bridge API and Device API. The Bridge API, representing the hardware responsible for communicating with the implanted device (e.g., a Summit CTM [2]), provides a set of functionality that ensures the continuity of INS device accessibility even in the face of connection errors. The device API exposes the configuration and capabilities of the INS hardware. Together, the Bridge and Device APIs create a standard interface for connecting to, configuring and using an INS device.

While the gateway component is largely optional (as an application developer can program directly to the Device Service), it exposes additional functionality to the client application by providing access to extension modules. The gateway provides a single point of entry to OMNI, easing the burden of accessing varied APIs via different hosts and ports. All API traffic to the device service flows through the gateway; this allows the gateway to intercept traffic from the device service and perform computation on those data. The data logging service logs the sense configuration and sense data of the INS flowing through the gateway in a standardized

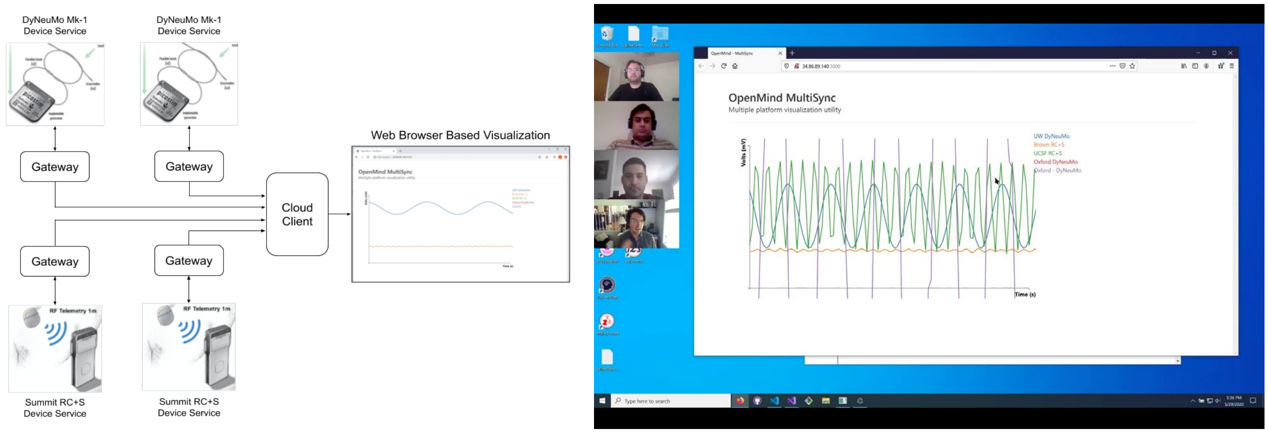

Figure 2. Left - Descriptive block diagram of a proof-of-concept application demonstrated at the 2020 Annual Brain Initiative Meeting. Four different INS devices from four different physical locations streamed data to a cloud-hosted client application. The client application visualized the four different data streams simultaneously. Right - Screen capture from the Brain Initiative demonstration.

format. Not all gateway extensions manipulate incoming and outgoing traffic, as some extensions provide additional functionality unrelated to the traffic flowing through the gateway. One such extension could be a standardized task engine that starts behavioral tasks and captures the participant's input. The task engine service can be used in conjunction with the data logging service to interleave behavioral data alongside the sensing data streams.

The client component presents a user interface to the clinician/subject participant and interfaces with arbitrary devices using the gRPC enabled Gateway. To demonstrate the feasibility of this platform, we developed a reference client application written in JavaScript to leverage the latest in web technologies for user interface and user experience design. The client component uses Electron [14] to build desktop applications for use on Windows, macOS and Linux.

As a medical device-interfacing software deployable for performing clinical research, the development of OMNI is subject to applicable regulatory standards for design controls. While a work in progress, the OMNI will be developed under platform-level design controls including relevant standards for the design, development, production, installation, and servicing of medical devices and software [15]. It is anticipated that research sites interested in developing new protocols using OMNI can further leverage quality standards to do so, ensuring that modified software is maintained in accordance with applicable regulatory requirements. OMNI will be distributed with unit and integration tests that validate software user requirements established in design control documentation. Reference design control and QMS-related documents are planned for dissemination to help further support continued evolution of OMNI in academic settings, where QMS infrastructure and practice is often not well established [10]. Researchers using OMNI will need to ensure system verification and validation for their own specific use case [16] [17].

III. Results

Using a proof-of-concept version of OMNI, we developed a reference application that streamed neurological data from both a Summit RC+S device and a DyNeuMo Mk-1 device concurrently to the client application. These data were visualized on a gateway application capable of near-real-time streaming. The gateway used an extension to stream data from these distinct INS devices to a cloud-hosted client application which then simultaneously visualized the data streams from multiple physical locations across the globe.

Leveraging the OMNI architecture, we were able to shift a local device-agnostic demonstration of the Generic API to a cloud-based streaming application in a matter of days, shown in Figure 2. gRPC APIs alleviate the burden of re-writing software due to INS device API incompatibility. Auto-generated client and server code significantly reduce the amount of software the researcher needs to write for a given application. JavaScript allowed our team to rapidly prototype a client streaming interface and graphical user interface. Furthermore, Electron is natively cross-platform, allowing the gateway application to run on Windows, macOS or Linux.

IV. Discussion

The OMNI architecture accelerates construction of hardware-agnostic software applications for neuromodulation research. The choice of gRPC as the glue of our architecture provides benefit to the researcher by autogenerating fast, secure interfacing code. The distributable Protobuf files allowed our multi-institutional team to create generic APIs that can be used across INS devices and programming languages. Separating the user-facing client application from the device-dependent services isolates the device service from any client-side crashes. Introducing a networking layer extends the INS device and language agnostic capabilities beyond localhost, enabling client applications and device services to live on different machines running different operating systems. The generated client and server code provide native encryption and authentication, as well as other interesting networking features such as deadlines and timeouts.

For all its benefits, OMNI has some limitations. Introducing a networking layer requires additional software resources that may add latency into the system. While using gRPC locally through a loopback network interface will only add minimal latency, for some high-performance algorithms, these latencies may be large enough to affect computer-in-the-loop adaptive algorithms. In addition, while the gRPC framework provides a method for wrapping up manufacturer-provided APIs and dynamically linked libraries (DLLs) into a language-agnostic framework, it is not an effortless process. For example, though OMNI provides definitions for common functionality, this common functionality will still need to be implemented in the Device Service while device-specific features will need to be structured from scratch -- depending on the size of the INS SDK, this can be a significant undertaking. Furthermore, deploying microservices adds complexity in comparison to deploying the single executables of monolithic applications.

Nevertheless, the networked API interface provides significant flexibility that will likely outweigh the development burden. Future device manufacturers can develop gRPC Device Services directly instead of APIs or DLLs that only support single programming languages. Additionally, the networked APIs allow for easy integration in systems that span operating systems – for example, allowing client applications hosted on machines running real-time operating systems to interface with a Device Service running on a local Windows machine. The networking features of gRPC can also enable enhanced at-home data capture by securely streaming data to off-site servers.

V. Conclusion and Future Directions

As we report here, OMNI provides researchers a means of creating INS device-agnostic software applications to powerfully enable advanced neuromodulation research. By abstracting common functionality of INS SDKs, we create a generic API that can be used across different INS platforms without the need to rewrite software. INS-specific features are exposed via optional device specific arguments to the Generic API.

OMNI is a work in progress, but the general framework for allowing device-agnostic software development using gRPC has now been demonstrated. Our next steps include working toward a production-ready generic API for INS devices targeting the Summit RC+S. As we complete work on the device service we will move into developing gateway functionality, and will continue to explore the advanced networking features of gRPC and how they can assist DBS research. Binaries and source code for OMNI will be disseminated through the Open Mind Consortium's GitHub organization.

We anticipate that OMNI will save research teams valuable time and effort by providing high-level, generic APIs for interacting with INS hardware. This capability holds the potential to accelerate innovation in advanced DBS research, obviating the need for extensive software re-development and facilitating the spread of innovative tools across device platforms and across disease research areas. Moreover, as original development of OMNI is being undertaken in accordance with design controls and applicable quality regulations, the platform is intrinsically positioned for user-friendly quality management as it undergoes modification by future research groups. The development of OMNI is informed by the needs of research sites within the Open Mind consortium, and we intend to transition existing research protocols to use OMNI as soon as is feasible.

By offering a flexible standard within the device-interfacing software space, OMNI overcomes drags on innovation and research efficiency imparted by a fragmenting device landscape, and empowers researchers to undertake more rapid, reproducible and cross-comparable studies as the DBS field confronts a wide range of major clinical challenges.

Acknowledgment

The authors would like to thank Heather Dawes for critical reading of the manuscript as well as her project management contributions to this work, as well as the rest of the Core Open Mind Contributors including Greg Worrell, Vaclav Kremen and Ben Brinkmann. This work was funded by NIH award 1U24NS113637. Summit RC+S systems and technical support were provided by Medtronic.

References

[1] P. Khanna et al., "Enabling closed-loop neurostimulation research with downloadable firmware upgrades," in 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), Oct. 2015, pp. 1–6.

[2] S. Stanslaski et al., "A Chronically Implantable Neural Coprocessor for Investigating the Treatment of Neurological Disorders," IEEE Trans. Biomed. Circuits Syst., vol. 12, no. 6, pp. 1230–1245, Dec. 2018.

[3] M. Zamora, R. Toth, J. Ottaway, T. Gillbe, and S. Martin, "DyNeuMo Mk-1: A Fully-Implantable, Motion-Adaptive Neurostimulator with Configurable Response Algorithms," bioRxiv, 2020.

[4] "Complete-System - CorTec." https://www.cortec-neuro.com/solutions/complete-system (accessed Nov. 12, 2020).

[5] N. C. Swann and C. de Hemptinne, "Adaptive deep brain stimulation for Parkinson's disease using motor cortex sensing," Journal of Neural Engineering, 2018.

[6] A. Velisar et al., "Dual threshold neural closed loop deep brain stimulation in Parkinson disease patients," Brain Stimul., vol. 12, no. 4, pp. 868–876, Jul. 2019.

[7] J. A. Herron, M. C. Thompson, and T. Brown, "Cortical brain–computer interface for closed-loop deep brain stimulation," IEEE Trans. Neural Syst. Rehabil. Eng., 2017.

[8] V. Kremen et al., "Integrating Brain Implants With Local and Distributed Computing Devices: A Next Generation Epilepsy Management System," IEEE J Transl Eng Health Med, vol. 6, p. 2500112, Sep. 2018.

[9] N. R. Provenza et al., "The Case for Adaptive Neuromodulation to Treat Severe Intractable Mental Disorders," Front. Neurosci., vol. 13, p. 152, Feb. 2019.

[10] D. A. Borton, H. E. Dawes, G. A. Worrell, P. A. Starr, and T. J. Denison, "Developing Collaborative Platforms to Advance Neurotechnology and Its Translation," Neuron, vol. 108, no. 2, pp. 286–301, Oct. 2020.

[11] "gRPC." https://grpc.io/ (accessed Nov. 12, 2020).

[12] "Google Resource Oriented Design," Resource Oriented Design. https://cloud.google.com/apis/design/resources (accessed Nov. 11, 2020).

[13] "API Design," Organize the API around resources. https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design#organize-the-api-around-resources (accessed Nov. 17, 2020).

[14] "Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS," Electron. https://www.electronjs.org/.

[15] F. Jamshed et al., "Quality Management Systems for Neural Engineering: Principles and Practices". Handbook of NeuroEngineering. [under review, Springer Nature]

[16] U.S. Food and Drug Administration, "Implanted Brain-Computer Interface (BCI) Devices for Patients with Paralysis or Amputation - Non-clinical Testing and Clinical Considerations," Feb. 2019.

[17] K. Bowsher *et al.*, "Brain-computer interface devices for patients with paralysis and amputation: a meeting report," *J Neural Eng*, vol. 13, no. 2. England, p. 023001, 29-Feb-2016.