# WHAT: Web-Based Haptic Authoring Tool for Multi-Point Interactions


Talha Zafar
Department of Computer Science, University of Calgary
Calgary, Canada
talha.zafar4@ucalgary.ca

Duc Le
Department of Computer Science, University of Calgary
Calgary, Canada
duc.le2@ucalgary.ca

Issac Ng
Department of Computer Science, University of Calgary
Calgary, Canada
isaac.ng@ucalgary.ca

Christian Frisson
Department of Computer Science, University of Calgary
Calgary, Canada
christian.frisson@ucalgary.ca



## Abstract

Building multi-point haptic devices remains difficult because it requires expertise in programming, kinematics, hardware integration, and mechanical design. This creates a barrier for novice hapticians who want to prototype richer physical interactions but lack the multidisciplinary background needed to implement them. Existing frameworks such as CHAI3D provide strong runtime support for haptic rendering, but still require low-level C++ implementation. Existing authoring tools reduce programming effort for scene-level haptic content, but they generally assume the physical device already exists. This project explores a web-based authoring interface for modular multi-point haptic devices. The current prototype demonstrates a browser-based 3D environment for importing, arranging, inspecting, and viewing haptic components. While it is currently a visual proof of concept, it establishes the interface direction for future automated kinematics and CHAI3D-compatible code generation.

This work introduces a web-based authoring workflow for modular multi-point haptic devices, moving early device design from low-level programming toward visual assembly and future automated kinematics.



## Keywords

haptics, multi-point haptics, authoring tools, CHAI3D, web-based interfaces, automated kinematics


## 1 Introduction

Modern haptic systems offer powerful capabilities, but they remain difficult to build. Developing a physical haptic device requires expertise in C++ programming, robotics, forward and inverse kinematics, firmware integration, and mechanical design. Many novice hapticians have ideas they would like to bring to life, but lack the multidisciplinary background required to develop them. As the field continues to grow in complexity and ambition, this barrier becomes increasingly significant. Richer haptic experiences demand more actuators, more contact points, and tighter integration between hardware and software, yet the tools available to designers have not kept pace.


*Poster at Graphics Interface 2026 (GI '26), Waterloo, ON, Canada*


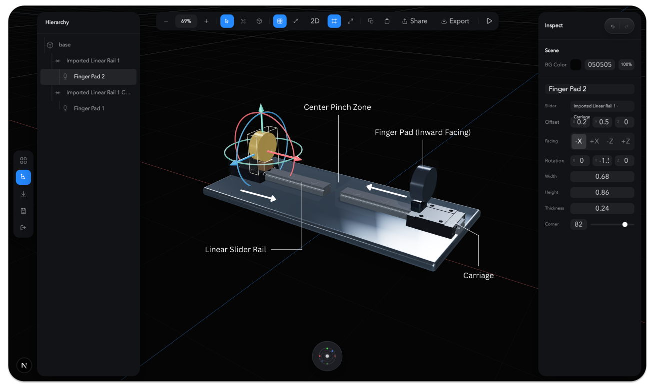


**Figure 1: Current MVP of the web-based authoring interface, showing the 3D assembly viewport, hierarchy and inspector panels, transform controls, and labelled modular haptic components.**

Frameworks such as CHAI3D provide runtime infrastructure for visuo-haptic rendering, but they assume low-level programming expertise [4]. Although researchers have extended CHAI3D to support customized multi-point haptic interactions [8], this still requires an extensive technical background. As a developer, you must be able to code in C++ and manually define device kinematics as well as manage inter-device communication. Authoring tools have reduced programming effort by shifting toward higher-level description and visual composition [1, 5, 7, 9], and multi-actuator frameworks address coordination at the signal and control level [2, 6]. Across these systems, the same pattern emerges: they simplify scene-level interaction or signal-level coordination, but they assume the physical device already exists. No existing system unifies modular device assembly, automated kinematics, and CHAI3D-ready code generation in a single web-based workflow [3].

This work presents a web-based authoring interface that targets that gap directly, treating the physical multi-point haptic device itself as the object of authoring. The current minimum viable product (MVP) implements a browser-based 3D authoring environment with component import, spatial manipulation, hierarchy management, and dual 2D/3D views. The prototype validates the feasibility of the authoring workflow and forms the first stage of a six-stage pipeline.

The contributions of this work are: (1) the identification and characterization of a device-level authoring gap that existing runtime, scene-level, and multi-actuator tools do not address; (2) the design of a six-stage pipeline that connects modular visual assembly to CHAI3D-ready integration code; (3) an MVP web-based authoring interface that validates the workflow and interface direction; and (4) a roadmap for completing automated kinematics, code export, and a usability study under an NSERC USRA summer studentship.

## 2 Background and Related Work

Haptic development tools fall into three layers relevant to this work: runtime frameworks, multi-point and multi-actuator systems, and authoring tools. We summarize each briefly and explain why the gap appears at the authoring layer.

CHAI3D is a widely used C++ framework for visuo-haptic rendering, handling hardware communication, virtual object management, and real-time force feedback [4]. Researchers have extended it to support customized multi-point devices [8], but the developer must still modify C++ routines and explicitly define the kinematics. CHAI3D provides strong runtime infrastructure but does not offer an authoring environment for designing devices.

At the multi-actuator level, structured control frameworks address coordination and signal-level complexity. A context-adaptive system uses XML-based representation to coordinate haptic channels [6], and Sound2Haptic provides a toolkit for portable multi-channel vibrotactile feedback [2]. These approaches lower barriers at the signal level but do not support designing modular mechanical devices or computing their kinematics.

Authoring tools shift the focus from low-level implementation to higher-level description and visual composition. Early work rendered VRML-described 3D scenes haptically within a browser [7]. An MPEG-V-based web tool introduced a graphical interface for defining haptic properties with usability evaluation [1], and HAMLAT integrated haptic functionality into Blender through the HAML description language [5]. A component-based tool built on Unity allows assembly of haptic interactions using reusable components and is conceptually closest to our project, but its scope is single-stylus haptic navigation [9]. Across these systems, the same pattern emerges: they reduce low-level programming for haptic effects in virtual environments, but they assume the physical device already exists. They do not support modular mechanical assembly, automate kinematics, or generate CHAI3D-ready integration code.

A retrospective on twenty years of World Haptics research highlights increasing system complexity and the need for broader participation in the field [3], aligning with the direction this project pursues. Despite this recognition, no existing system unifies modular device assembly, automated kinematics, and framework-ready code generation in a single web-based workflow. Unlike prior tools, our work treats the physical multi-point haptic device itself as the object of authoring.

## 3 System Architecture

The system follows a five-stage pipeline (Figure 2) from visual assembly to framework-ready code: user authoring, mechanism graph construction, forward kinematics, inverse kinematics, and CHAI3D code export. The current prototype implements Stage 1,

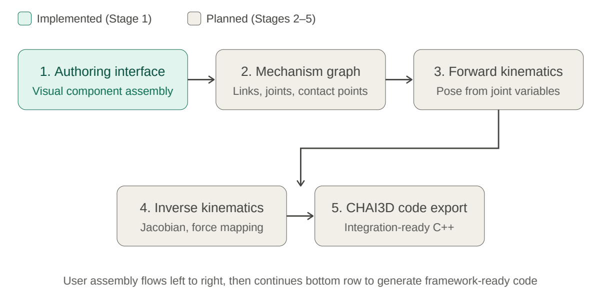


**Figure 2: Five-stage system pipeline. The current MVP implements Stage 1, while Stages 2–5 are planned for future development.**

while the remaining stages are planned for the continuation of this project.

The authoring interface is built as a web application using TypeScript with React and Next.js, with 3D scene rendering handled by Three.js. The decision to build a web-based tool was deliberate. A browser-based interface removes the need for local installation and reduces platform dependency, directly aligning with the project's goal of lowering barriers for novice hapticians. TypeScript was chosen for development reliability in managing the complexity of 3D scene state and compxonent hierarchies.

The interaction technique is intentionally modeled on a Figma-like design environment rather than a traditional CAD or robotics visualization tool. Existing 3D authoring and engineering tools, such as CAD software and ROS, are powerful but prioritize comprehensive functionality over learnability, presenting a steep learning curve for novice users encountering them for the first time. In contrast, the user interacts with components visually and spatially, not through code or parameter tables. The current prototype provides a main 3D viewport, a hierarchy panel displaying part structure as a collapsible tree, an inspector panel for position, rotation, scale, and dimension properties, and a toolbar with mode controls including a 2D authoring view with dimensional annotations. This dual 2D/3D representation lets the user switch views depending on the task, similar to how design tools allow toggling between layout and detail modes.

## 4 Conclusion

This paper presented a web-based authoring interface for multi-point haptic devices, addressing a gap in existing tools: no current system combines modular device assembly, automated kinematics, and framework-ready code generation in one accessible workflow. The MVP validates the concept through a browser-based 3D environment where novice hapticians can assemble components without writing code. Future work includes GLB-based CAD import, automated kinematics, CHAI3D code export, and a usability study.

## References

[1] Abdulwahab Alhalabi. 2013. *MPEG-V based web haptic authoring tool.* Master's thesis. University of Ottawa, Ottawa, ON, Canada.
[2] Yon Chin, Mike Sinclair, and Hiroshi Ishii. 2025. Sound2Haptic: A toolkit for portable multi-channel haptic integration across multiple form factors and devices. In *Proceedings of UIST Adjunct.*

[3] J. Edward Colgate, Lynette A. Jones, and Hong Z. Tan. 2025. Twenty years of World Haptics: Retrospective and future directions. *IEEE Transactions on Haptics* 18, 3 (2025), 452–455. doi:10.1109/TOH.2025.3605032

[4] Francois Conti, Federico Barbagli, Remis Balaniuk, Michael Halg, Chi Lu, Dan Morris, Luis Sentis, Jeff Warren, Oussama Khatib, and Kenneth Salisbury. 2003. The CHAI libraries. In *Proceedings of EuroHaptics*. Dublin, Ireland, 496–500.

[5] Mohamad Eid, Sheldon Andrews, and Abdulmotaleb El Saddik. 2008. HAMLAT: A HAML-based authoring tool for haptic application development. In *Proceedings of EuroHaptics*. Madrid, Spain.

[6] Joohwan Kim, Youngsun Kim, and Woontack Woo. 2009. A context-adaptive multi-sensor and multi-actuator haptic interaction framework. In *Proceedings of ACM Multimedia*.

[7] Marcia A. O'Malley and Charles E. Hughes. 2007. Simplified authoring of 3D haptic content for the World Wide Web. In *Proceedings of the World Haptics Conference*.

[8] Yitian Wei and Zoran Najdovski. 2014. Extending support to customised multi-point haptic devices in CHAI3D. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

[9] Takuto Yamaguchi, Takeo Igarashi, and Hiroshi Takemura. 2015. Component-based authoring tool for haptic navigation. In *Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP)*.