BENCH and AI Led Application Design

How a brief for greater breadth became an interactive workshop of mechanisms and live explanations

Brad Micallef and Sol | VisionLeadership AI Guidance
Standalone application case study 3 | Following Resonance
15 November 2026 | Development snapshot through version 1.12

Executive summary

BENCH began with a request to demonstrate greater breadth after two AI proposed applications. The accepted idea emerged through disagreement: an ecological world resembled Understory, a sound workshop overlapped Resonance, and a reverse engineering notebook still required an object to take apart. Brad Micallef asked to preserve the underlying curiosity while removing that constraint. The result became a native Mac workshop supplied with working mechanisms. [1]

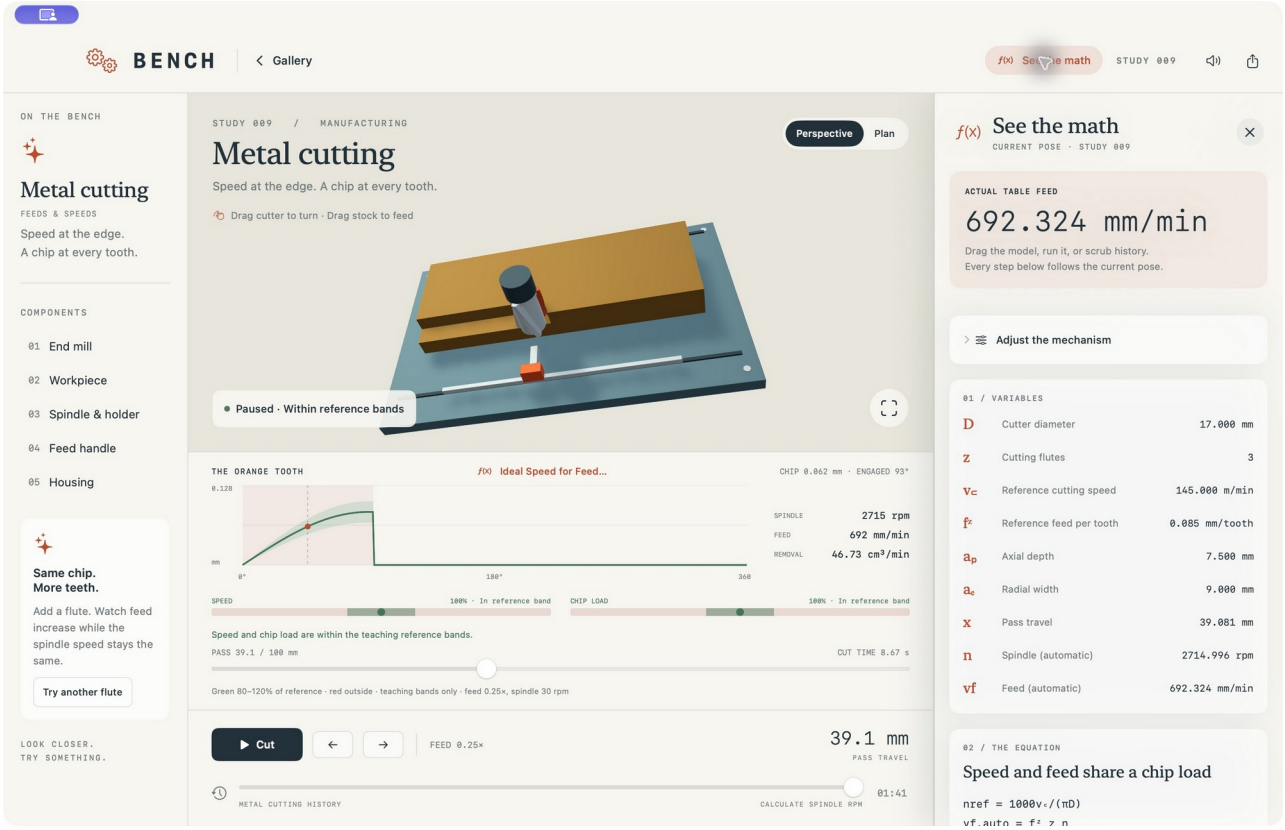


Figure 1 BENCH 1.12 in use. The actual Mac milling workbench connects the object, chip graph, reference comparisons, and live worked calculations.

The project grew from one ratchet to nine studies spanning linkages, gearing, programmed motion, rope systems, hydraulics, and metal cutting. Human use exposed a history bug and the distinction between valid motion and solid parts that collide. Later requests made the equations visible and separated user inputs from calculated or adjusted results. The case shows AI proposing, implementing, correcting, and checking software while a human continues to set the direction. [1-4]

1 A brief that demanded a different purpose

Brad's invitation asked the AI to stretch its wings and demonstrate breadth after two applications. The task was broader than writing code for a specified product: the AI had to propose an application worth making, explain its purpose, and earn approval for a direction. This third application case follows the Resonance paper in the series. [1]

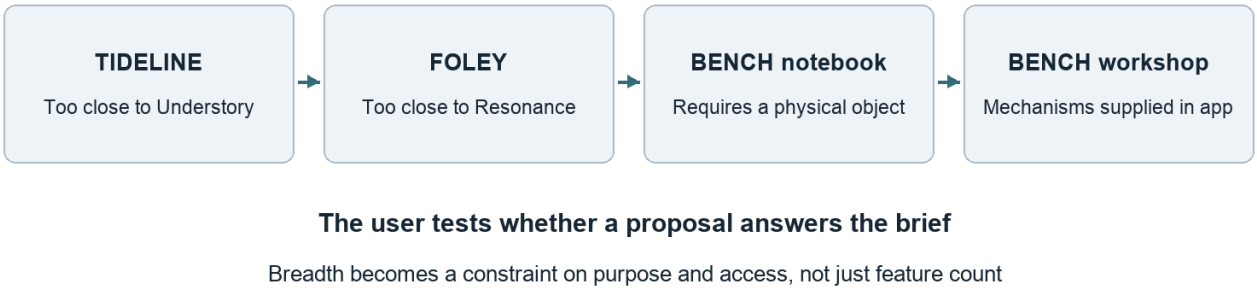


Figure 2 The accepted brief emerged through correction. Each user response tightened what would count as a distinct and accessible third application.

The first proposals did not satisfy the brief

TIDELINE proposed sculpting an island and observing an ecosystem. Brad recognized the overlap with Understory and suggested that it belonged in a later version of that app. FOLEY then proposed performing procedural sound effects with gestures. Brad again identified a close relationship to Resonance. The AI acknowledged that changing scope within familiar territory had not demonstrated the requested breadth.

BENCH first appeared as a visual reverse engineering notebook: photographs, parts, fasteners, dependencies, and assembly steps. That was a more distinct purpose, but Brad identified an access problem. A person still needed something physical to dismantle. His next request kept exploration of parts and relationships while removing that prerequisite.

A complete first experience rather than an unlimited builder

The revised proposal supplied the objects inside the app. Participants would operate a mechanism, reveal its internals, investigate a fault, make a repair, and replay what happened. The first implemented object would be a ratchet. A general machine construction environment and arbitrary object recognition stayed outside the first build.

This sequence is evidence of design correction, not evidence that the AI possesses a private preference or subjective desire. Its observable contribution was to generate proposals, respond to objections, and carry an accepted idea into software. Brad chose the direction and authorized each expansion. BENCH operates locally without a language model or AI account. [1-2]

2 One ratchet made the proposal testable

The first milestone was a complete ratchet experience, not a rotating illustration. A participant could drag a wheel, orbit the view, select a component, hide the housing, explode the assembly, lift or remove the pawl, remove its return spring, reverse the catch, and investigate a fault. Keyboard steps and transport controls supplied alternatives to direct dragging. [1-2]

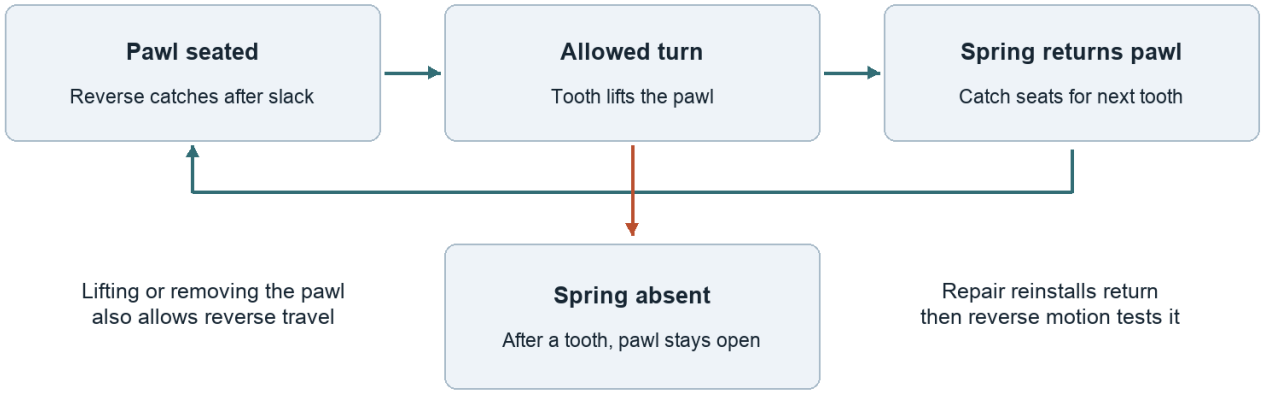


Figure 3 The ratchet state logic. Removing the spring does not immediately remove a seated catch; the failure appears after forward travel lifts the pawl.

An explanation needs a consequence the user can test

The idealized model tracks angle, tooth passage, engagement, permitted direction, and the conditions of the pawl and spring. When reverse motion stops, the explanation points to a seated pawl against a tooth face. Lifting the pawl changes that constraint and makes reverse travel possible. The text follows the represented state rather than describing what a generic ratchet usually does.

The missing spring provides a particularly useful experiment. A catch already seated can still block reverse motion. Forward movement then lifts it over a tooth; without the return rule, it remains open. Reinstalling the spring restores the modeled return behavior, and an attempted reversal checks the repair. This creates a small sequence of hypothesis, intervention, and observable consequence.

Viewing and changing have separate meanings

Hiding the housing reveals the assembly without removing its supports from the mechanism. Removing a pawl changes operation. Exploding the assembly separates parts for inspection and disables operation until they return. These differences keep a presentation control from silently becoming a mechanical intervention.

The model is quasistatic. Its spring is illustrated and its return behavior is a rule; BENCH does not solve spring torque, impact, inertia, wear, or material stress. Procedural geometry supplies asymmetric teeth and open spokes, while state supplies contact and engagement. This deliberate boundary makes the behavior repeatable and explains why a convincing object can still have a limited physical model. [2-3]

3 Architecture keeps the explanation attached to state

BENCH uses Swift 6 with SwiftUI, AppKit, and RealityKit. A local Swift package separates the mechanical core, procedural scene construction, and Mac application. No external package or downloaded 3D asset is required. Geometry is authored in code, and the scene uses meters even where controls and worked examples display millimeters. [2]

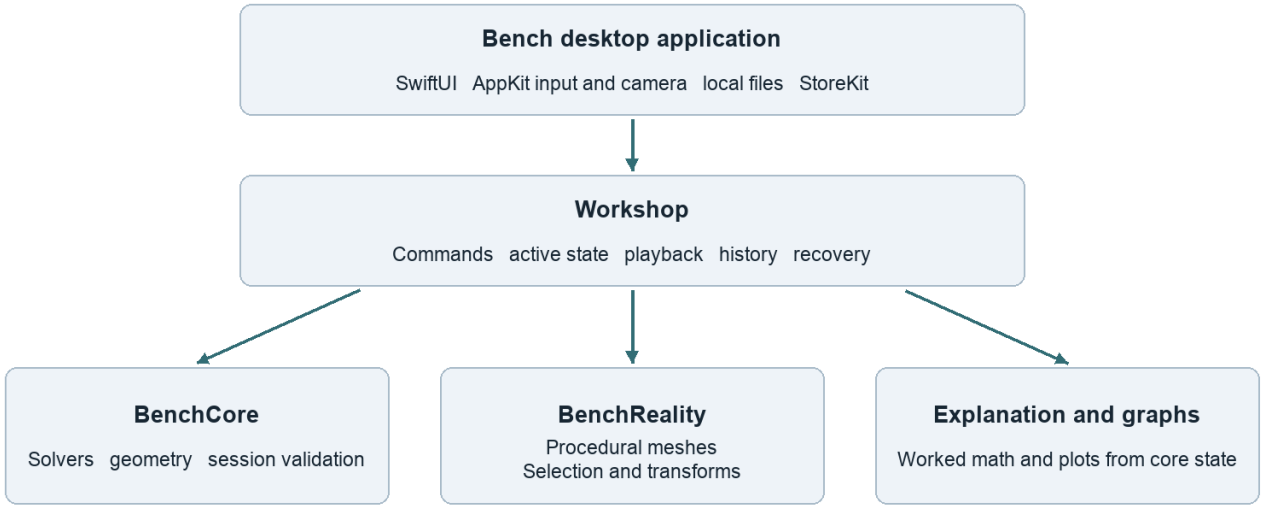


Figure 4 The main responsibilities. Core state and solutions supply scene presentation, plots, and worked explanations; the Mac adapter owns desktop input and viewing.

The core solves and the scene presents

BenchCore contains mechanism state, forward and inverse solves, geometric clearance, worked mathematics, catalog policy, and validated session serialization. Its mechanical calculations have no UI dependency. Each study has a specific model rather than sharing a generic physics engine that is expected to discover the intended mechanism from contact alone.

BenchReality constructs meshes, materials, component identities, selection targets, highlights, and inspection offsets. It positions parts from model state and exposes a camera independent scene. The Mac target translates mouse and trackpad gestures into supported changes, manages the camera, and presents the workshop, gallery, graphs, math card, sound, Help, and file dialogs.

One operation must have one accepted result

Workshop coordinates the active exhibit, playback, intervention commands, history, and recovery. A graph scrub, model drag, step button, or dimension edit must resolve through the appropriate model. The picture and worked answer should show that accepted pose, including a closure limit or contact stop, rather than each interface inventing its own answer.

This separation also supports a spatial path. The shared model and scene compile against the visionOS simulator SDK. A native Vision Pro host, gaze and pinch behavior, readable spatial labels, comfort testing, and device validation remain separate work. Successful module compilation is evidence of code portability, not evidence of a delivered spatial app. [2, 5]

4 Valid motion did not guarantee that parts fit

The crank and slider made a gap visible. Increasing the crank radius while shortening the rod could keep the mathematical linkage solvable while allowing the slider to overlap the wheel. Brad asked whether collision behavior should be optional. That feedback changed the meaning of the simulation controls and expanded the verification work. [1]

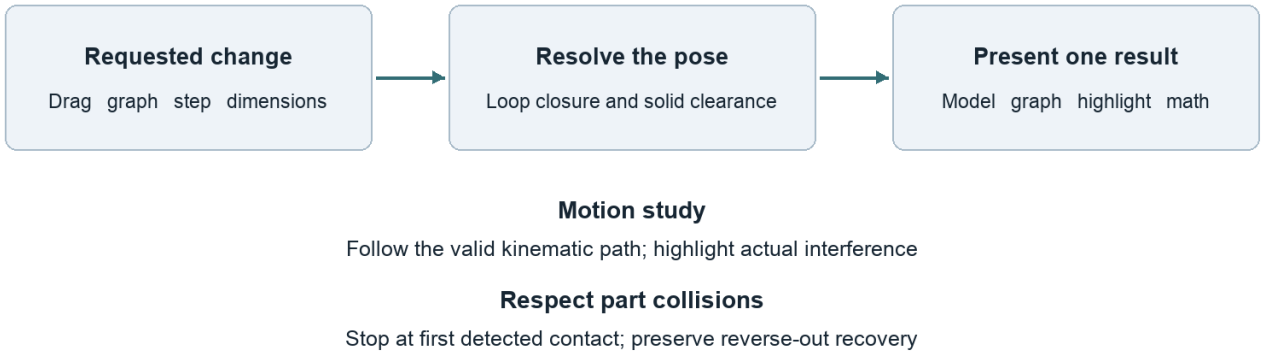


Figure 5 Requested movement is resolved before presentation. Kinematic limits remain in force in both modes; the collision option adds stops for modeled solid interference.

Two questions needed two explicit modes

Motion study remains the default. It follows the valid theoretical motion profile and highlights actual interference. Respect part collisions adds a stop at first detected contact and pauses automatic travel. Reverse input remains available so the participant can back out. The complete theoretical graph remains visible, allowing the intended path and physical obstruction to be compared.

The distinction matters in three dimensions. Two parts can overlap in the camera image while occupying separate depth planes. Open spokes are empty space. Hollow bushings differ from solid blocks. Intended pin and guide connections must not be mistaken for obstruction. Clearance tests therefore use simplified convex pieces corresponding to the rendered solid features and their actual depth, with explicit joint exclusions.

A correction became a shared constraint on interaction

The clearance engine rejects clearly separated assemblies before using separating axis checks on the remaining convex pieces. Angle and dimension changes are swept in bounded increments, with a detected contact refined by bisection. Crank dragging, carriage dragging, graph scrubbing, and dimension edits use the same resolution policy. The four bar study adds its own closure and branch constraints.

The check is geometric interference handling. It does not calculate collision impulses, deformation, friction, fatigue, or manufacturing fit. Nor does incremental sweeping prove an exact continuous collision solution for arbitrary geometry. Its scope is the supported exhibit shapes and controls, with tests for gaps, depth separation, first contact, and recovery. Naming those limits is part of making the demonstration useful. [2-3]

5 Human use changed the application at several levels

Brad's feedback supplied both defects and design opportunities. The pattern was consistent: connect an observation to a change in behavior, then check the result through the affected interaction paths. The following examples come from the project conversation rather than a formal usability study. [1]

Feedback or request	Design response	Why it mattered
A third idea should show greater breadth.	Move beyond ecology and sound, then remove the need for an owned object.	The purpose and entry conditions changed before implementation.
The lower slider switches from crank and slider to ratchet.	Scope history scrubbing to the visible exhibit.	A time control must not unexpectedly become a device selector.
The slider overlaps the wheel at extreme settings.	Highlight real interference; add optional stops and reverse-out recovery.	A solvable linkage and a physically clear assembly are different claims.
Move device choice into a dedicated launch gallery.	Searchable illustrated cards and a continue experiment path.	A growing collection needs a clear way in without crowding the workshop.
Let the user See the math.	A sliding card with units, equations, live substitutions, and compact controls.	The explanation becomes inspectable alongside the object and graph.
Include hydraulics and metal cutting among the nine.	Build two different models within the shared interaction system.	Breadth extends to fluid power and a manufacturing process.
Separate user inputs and calculated values; show speed and feed effects.	Automatic and adjusted outputs, inverse speed matching, and labeled reference bands.	Users can see dependencies and the tradeoffs created by an override.

A bug can reveal a mistaken boundary

The history bug was not simply an incorrect label. Restoring a complete snapshot also restored exhibit selection. The correction kept scrubbing within the current study and preserved inactive mechanisms. Opening an experiment still restores its saved selected exhibit. The same stored data therefore has different responsibilities during local exploration and whole document restoration. [2-3]

This case records one participant's use. It supports specific conclusions about discovered problems and implemented responses, not a claim that the interface has been validated for all learners or teaching environments.

6 See the math became an interaction rather than a report

Brad asked for variables, the raw equation, and a live step by step solve. BENCH added a sliding math card instead of a detached calculation report. A pinned result remains visible, compact controls allow continued experiments, and the solution follows dragging, playback, history, and accepted travel limits. Closing the card returns to the inspector. [1-2]

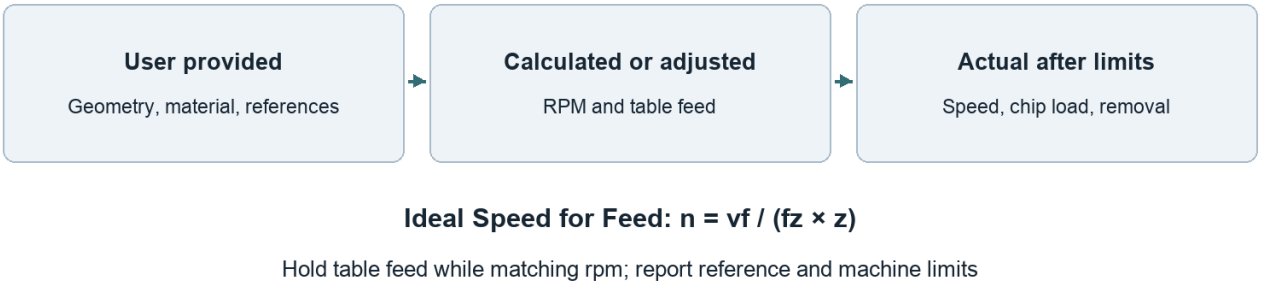


Figure 6 The milling calculation distinguishes selected references, automatic or adjusted outputs, and actual conditions after limits. The inverse action holds feed explicitly.

Dependencies need to remain visible when a value is adjusted

The milling refinement groups setup geometry, material, tool, and reference cutting data under User provided. Calculated contains spindle rpm and table feed. Either output can be adjusted independently or returned to automatic. Automatic feed follows actual rpm to preserve reference feed per tooth; holding feed while changing rpm instead changes the chip taken by each tooth.

For example, a 10 mm cutter, four flutes, a reference speed of 120 m/min, and 0.040 mm per tooth produce about 3,820 rpm and 611 mm/min before a limiting condition intervenes. Halving rpm with automatic feed halves feed while preserving chip load. Halving rpm with feed held doubles actual chip load. These results follow the model’s arithmetic; they are not measured machining outcomes. [2]

A colored comparison needs a stated reference

Ideal Speed for Feed solves n = vf / (fz × z), reports applicable limits, and preserves the current table feed when applied. Separate meters compare actual cutting speed and chip load with their references. The chip graph adds a dashed reference curve and a green envelope during engagement; air cutting and normal zero thickness at entry or exit are not classified as rubbing.

Green covers 80 to 120 percent of the selected reference. Those boundaries are BENCH teaching choices, not manufacturer limits. Red means outside that band, with text explaining the direction and qualitative tradeoff. Heat, finish, tool life, chatter, and machine suitability are not simulated. Making a result easy to read does not remove the obligation to state what it represents. [2-3]

7 Nine studies broaden the questions a user can ask

The collection grew by adding a distinct relationship and preserving the shared workbench vocabulary. Each study offers direct operation, adjustable supported parameters, inspection, graphs where relevant, local history, saved experiments, and live mathematics. The nine entries in the current catalog are included studies; that describes access within the build, not a published price for the app. [1-2, 4]

Study	Relationship to explore	Implemented model boundary
001 Ratchet	One way engagement, release, spring return, and repair.	Quasistatic catch logic; no impact or spring force solve.
002 Crank and slider	Rotation, reciprocation, stroke, and rod proportions.	Rigid planar geometry with optional solid clearance stops.
003 Four bar linkage	Closed loop motion, assembly branches, and parallelograms.	Circle intersection and geometric closure; no flexible links.
004 Meshing gears	Tooth counts, opposite rotation, and input or output driving.	Ideal external spur mesh without backlash or slip.
005 Cam and follower	Lift, timing, dwell, and motion derivatives.	Cycloidal program and ideal contact; no follower separation.
006 Geneva drive	Intermittent indexing, engagement, and locked dwell.	Geometric pin and slot relationship; no impact forces.
007 Block and tackle	Rope travel, supporting strands, force, and work.	Massless frictionless rope and blocks; quasistatic balance.
008 Hydraulic circuit	Working areas, flow, pressure, load, relief, and reversal.	Ideal incompressible steady state; no pressure transients.
009 Metal cutting	Cutter geometry, speed, feed, chip thickness, and removal.	Illustrative end milling; no cutting force or stability model.

A catalog makes expansion an explicit contract

A bundled catalog names known, compiled mechanisms and supplies display metadata and optional product identifiers. The gallery can search and filter without creating the RealityKit workbenches or playback timers for every card. A continue path reopens the recovered experiment, while opening a file goes directly to its selected workbench.

The design does not load arbitrary executable mechanisms from a download. New study functionality ships in an app update. Optional purchase entitlements can then make a registered study accessible. The catalog remains a collection of compiled studies. [4]

8 New physical domains require new modeling decisions

Brad redirected the final two included slots from a universal joint and differential toward a hydraulic piston and metal cutting. The change strengthened the original request for breadth: the workshop now links geometric motion, a fluid circuit, and a manufacturing relationship. Reusing the interface does not mean copying the physical assumptions. [1]

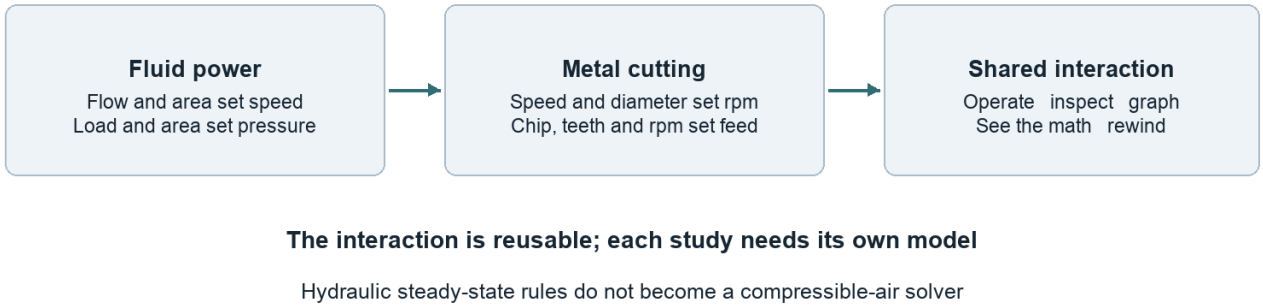


Figure 7 Different studies reuse the interaction contract while retaining domain specific solves. The arrows summarize expansion of the collection, not a physical connection between the exhibits.

Hydraulics makes the circuit part of the explanation

The hydraulic study combines a differential cylinder, fixed flow pump, tandem center valve, relief branch, gauge, and reservoir. Cap and annular areas explain why retraction is faster and has less force at the same pressure. Required load pressure, signed velocity, chamber flows, and power follow the same solution that drives the cutaway and flow presentation.

In neutral, the pump unloads to the reservoir while the working ports are blocked. At overload or a stroke end, pump flow goes through relief; reversing can move away from a stop. The resisting load is a passive bidirectional threshold, not gravity or an overrunning load. Trapped chamber pressures, leakage, compressibility, transients, and structural limits remain unresolved or excluded. [2-3]

Machining separates calculated conditions from visible motion

The milling study uses selected references to calculate rpm, feed, ideal chip thickness, removal rate, and representative pass time. The scene removes stock along the cutter envelope. Its reversible pass is an explanatory study, not a manufacturing history. Feed playback is slowed to one quarter of calculated feed and the illustrated spindle turns at 30 rpm, independently of the calculated machine rpm.

A planned pneumatic study illustrates the limit of reuse. Compressible air needs chamber mass, absolute pressure, volume, valve flow, and piston dynamics. Blocking ports must allow trapped air to compress and expand rather than simply freezing the piston. That work is documented as a future paid addition, not implemented by relabeling the hydraulic solver. [4]

9 Rewind and recovery are part of the experiment

Rewind lets a participant revisit an explanation at the moment it applied. BENCH records mechanism configuration as well as pose, so a change in rod length, tooth count, cam timing, or milling override can be restored. An old pose must retain its original settings. [2-3]

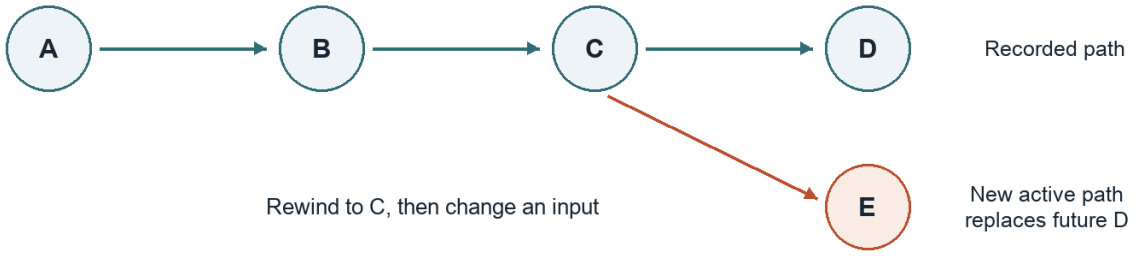


Figure 8 A schematic history branch. Changing an input after rewinding replaces the later active path; BENCH does not preserve a permanent tree of alternative futures.

Scope is different from permanent branching

The history slider selects snapshots for the current exhibit and changes only that mechanism's state. Inactive studies retain their present state during local scrubbing. Interacting from an earlier cursor truncates the later recorded future and starts a new active path. This is useful for retrying a hypothesis, but it is not side by side comparison of indefinitely retained branches.

The collection retains at most 1,800 snapshots. A saved experiment records the visible branch through the current cursor. Automatic recovery uses the same session representation and writes atomically to the local application support directory. File size and session validation restrict what the loader accepts; malformed or unsupported sessions should produce an explanation rather than a partially trusted world.

Compatibility is a behavior worth testing

As studies were added, session versions gained optional state for the new mechanisms. The current version 9 format continues reading versions 1 through 9. Missing milling override fields default to automatic calculation. Tests exercise invalid numeric inputs, old formats, scoped history, branch edits, saves, and recovery.

The important design lesson is that undoable exploration needs a declared policy. Which state changes when time is scrubbed, whether later paths survive, and what a saved document contains are user facing choices. They should be decided and tested with the same care as the mechanism's visible motion. Passing serialization round trips alone would not have detected the original surprise of switching exhibits. [1-3]

10 Verification combines invariants and direct observation

The retained version 1.12 verification run on 15 November 2026 passed 123 tests: 87 core tests and 36 workshop tests, with zero failures. The shared core and RealityKit scene also compiled against the visionOS simulator SDK. These are recorded software results, not a claim of physical calibration or a completed spatial product. [3, 5]

Evidence	What was checked	What it cannot establish
Mechanical invariants	Link lengths, loop closure, rope length and work, gear travel, flow and power relationships.	Accuracy of omitted forces, wear, elastic effects, or real machine performance.
Geometry and numerical checks	Forward and inverse solves, derivatives, contact stops, profile clearance, and mesh triangulation at model and scene scales.	A general proof for arbitrary shapes or unrestricted input ranges.
Workshop behavior	Inspection lockout, exhibit scoped history, branching, recovery, legacy formats, and invalid sessions.	That every learner will understand the controls or explanations.
Direct Mac observation	Recorded checks of dragging, graphs, live math, repairs, and new study controls; current workbench captured for this paper.	A formal accessibility study, classroom trial, or broad device test matrix.
Spatial compilation	Shared BenchCore and BenchReality modules compile for visionOS.	Native host behavior, gaze and pinch interaction, comfort, or headset validation.

The right test compares independent consequences

Examples include checking cam derivatives against finite differences, matching rope input work with lifted output work, checking gear pitch travel across tooth pairs, and testing mesh triangulation at both millimeter and meter scales. Such checks can expose errors that a screenshot or a second copy of the same formula would miss.

The feedback record also shows why successful compilation and core tests were insufficient. A history slider changed the selected exhibit, and a geometrically solvable linkage allowed solid interference. Those discoveries required new behavior and corresponding checks. The practical loop is propose, implement, exercise, observe a mismatch, revise, and verify the affected paths. [1-3]

Remaining validation includes a native Vision Pro experience, real purchase and restore flows if paid entries are enabled, broader usability and accessibility evaluation, and any domain validation needed before making stronger engineering claims. None is implied by the test count. [4-5]

11 Final decisions for the current design

BENCH is a local native Mac application requiring macOS 15 or later. The current development snapshot contains nine included studies and the version 1.12 milling refinements. It has a searchable launch gallery, a consistent workbench, direct object manipulation, inspection, live mathematics, saved experiments, and local recovery. Public signed distribution is not established by the local build. [2, 4]

Decision	Result and rationale
Purpose	Help people investigate how mechanisms work using supplied objects and explanations they can test.
Modeling	Use specific deterministic or quasistatic models with explicit assumptions. Do not imply a general engineering simulator.
Interaction	Keep the model, graph, accepted constraints, and worked explanation synchronized; distinguish inspection from operation.
Platform	Deliver the Mac workbench first. Preserve shared model and scene code for a separately validated spatial host.
Collection	Keep all nine current studies included. Proposed extra studies and paid access remain distinct from implemented content.
Expansion	A proposed US price of $0.99 per extra study and community requested development are roadmap decisions, not live offers.

What this case contributes to the series

Following Resonance, BENCH shows a different use of AI design initiative: a brief was challenged until it had a distinct purpose, then turned into interactive geometry, numerical relationships, and explanations. The strongest evidence lies in the visible corrections. Brad's use changed the product's state boundaries, collision semantics, navigation, and treatment of calculated values.

The transferable practice is to make a small claim testable before enlarging the collection. A ratchet explanation can be checked by lifting the pawl; a linkage can be checked against its graph; an rpm adjustment can be checked against chip load. Each extension then needs its own evidence rather than inheriting credibility from the polish of the shared interface.

Follow BENCH at Knowledge Locker

Knowledge Locker is the destination for product news and availability. At this paper's cutoff, BENCH is documented as a local Mac development build; a public download or App Store purchase is not claimed. A useful first experiment is to remove the ratchet spring, turn past a tooth, test reverse travel, and repair it. Then compare the object, graph, and live math in another study.

Visit Knowledge Locker for product information and availability

Sources and project evidence

This standalone case study follows the third AI proposed application project through 15 November 2026 and follows the Resonance paper in the VisionLeadership AI Guidance series. Sol is the name used for the AI collaborator. The account describes observable proposals, instructions, implementation, feedback, and verification; it does not claim access to hidden deliberation or establish subjective AI desire.

1 Original project conversation. AI Wants 003, Propose the next app to build, 12 to 15 November 2026. Includes the breadth request; TIDELINE, FOLEY, and notebook revisions; approval of the Mac ratchet; Vision Pro interest; history and collision feedback; gallery and live math requests; the nine study lineup; hydraulics, machining, and milling guidance. Attributed feedback refers to this primary record.

2 Application implementation. BENCH Swift package, development snapshot through 1.12. BenchCore contains the nine mechanism models, mechanical geometry, worked mathematics, catalog validation, and session format. BenchReality contains procedural scenes and meshes. Bench contains Workshop, the Mac input and camera adapter, gallery, inspectors, graphs, math card, StoreKit foundation, local files, and bundled Help.

3 Verification evidence. Tests/BenchCoreTests and Tests/BenchWorkshopTests; retained bench-milling-guidance-tests.log from 15 November 2026 records 87 core and 36 workshop tests passing with zero failures. The project conversation records direct Mac checks during each major increment. This paper also uses a read-only capture of the current milling workbench; it did not rerun or replace the app during authoring.

4 Distribution and future mechanisms. Distribution/StoreKitSetup.md and Distribution/PaidMechanisms.md distinguish compiled catalog entries from downloaded executable code, verified entitlements from local ownership flags, and planned content from delivered studies. No product identifiers are configured in the current catalog. Paid mechanisms and public distribution need their own setup and validation.

5 Spatial design and build evidence. Spatial/Design.md and scripts/check-spatial.sh describe the Mac first release and the intended volumetric host. The retained bench-milling-guidance-spatial.log reports successful compilation of the shared mechanical core and scene against the visionOS simulator SDK. Native spatial interaction and device validation remain unperformed.

Figures and examples

Figure 1 is a direct capture of BENCH 1.12 on the Mac with its current milling state and live math open. It is an application image, not a generated mockup. Figures 2 through 8 are explanatory diagrams prepared for this paper from the project record and implementation. The numerical example in section 6 is an illustrative substitution into the implemented metric equations; rounded display values do not replace full precision calculation.

The previous papers remain standalone. References to Understory and Resonance here explain the proposal history and sequence of this third case; they do not imply shared runtime code or physical models.

Knowledge Locker