

HZ-V1 • CH14 — Sequence Detail Editor

Sequence-level transition details between nodes ("how you get from Move A to Move B").

Doc ID	HZ-V1-CH14_Sequence_Detail_Editor_R1
Revision	R1 (2026-01-02)
Status	Draft
Depends on	CH03, CH06, CH08, CH12, CH13, CH31
Related	CH15, CH16, CH19, CH20, CH21, CH22, CH28, CH29, CH30, CH32
Supersedes	—
Owned Decisions (locked by this chapter)	Editor surface, field groups, save/cancel rules, per-edge detail lifecycle, and UX copy for sequence editing.
Open Questions / Placeholders	See §11 (Placeholder Registry).

Chapter Goal: Specify the Sequence Detail Editor so builders can implement it without guessing: where it appears, what it edits, how it saves, and what happens in every edge case.

This chapter defines **sequence details** (transition metadata) attached to a connection between two move nodes. It does **not** redefine the canvas/branching system (See: HZ-V1-CH12) or the node/edge data model (See: HZ-V1-CH13).

1. Scope and Definitions

A **Sequence** is the transition between two move nodes in a flow: from *source move* to *target move*. The Sequence Detail Editor is the UI that lets users record the transition details (footwork, distance change, opponent reaction, cues, notes, etc.) for that connection.

Key terms are owned by CH03 (Glossary). This chapter assumes the definitions for: move, alias, family, variant, flow, path, branch, merge, gameplan, mastery, maintenance, inbox. See: HZ-V1-CH03.

- **Owned by CH14:** how users view/edit transition metadata between nodes; editor UI; validation; save/revert; copy; deletion behavior; copywriting.
- **Owned elsewhere:** canvas gestures and how edges render (CH12), edge schema and IDs (CH13), move library rules (CH09–CH11), plan gating (CH08), offline/sync rules (CH28), limits/media rules (CH29), warning ladder (CH30).

Design intent: sequences allow users to capture nuance *only when they want*—the editor must support ultra-simple notes for most users, while enabling deep detail for power users (progressive disclosure).

2. User Jobs and User Stories

- As a striker
 - I can add a quick note for how to transition from jab to cross so I remember it later.
 - I can describe opponent triggers ("if they lean back"), my cue ("step outside"), and my response so I can drill it.
 - I can store multiple micro-details (footwork, distance, head level) without being forced to fill anything out.
- As a coach
 - I can encode the "why" between nodes so students understand how to move from one action to the next.
 - I can keep it minimal for beginners but expand for advanced athletes.
- As a user importing a flow
 - I can see sequence details exactly as the creator intended inside the imported flow.
 - If the flow conflicts with my move definitions, the sequence details remain intact for that imported flow (See: HZ-V1-CH19).

3. Entry Points and Navigation

The Sequence Detail Editor must be reachable anywhere the user can view an edge between two moves.

Primary entry point (V1): In the Flow Builder, tapping an edge (or its inline "details" affordance) opens the Sequence Detail Editor as a modal sheet.

- **From Flow Builder canvas:** Tap edge line OR tap the small inline "→" chip between nodes (exact affordance is specified in CH12; this chapter defines what happens after the tap).

- **From Flow Detail View:** Optional in V1: a "Paths"/"Transitions" list can deep-link into a specific edge's sequence editor (Owned by CH16; this chapter specifies the deep-link target route).
- **From Practice Mode:** V1 default is *view-only* sequence notes during practice (optional). If implemented, it must open the same editor in read-only mode while a session is running. See placeholders §11.

Route naming: The editor is a modal route: **SequenceDetailModal**(flowId, edgeId). It may be implemented as a bottom sheet or full-screen modal depending on platform constraints. Behavior is owned by this chapter.

4. Editor Surface and Layout

The editor is designed as a **progressive disclosure** sheet: quick info at top, optional deep sections collapsed by default.

4.1 Modal header (always visible)

- **Title:** “Sequence”
- **Subtitle:** “ → ” (auto-generated, non-editable). If a branch label exists on the edge (opponent action), show it as a pill chip under the subtitle.
- **Actions:** [Close] on left; [Save] on right.
- **Save state:** [Save] disabled until a change is made. When disabled, it reads “Saved” or shows a check icon (implementation detail).

4.2 Top summary block (always visible)

- **Quick Notes** (multiline text, optional). Placeholder: “Add a quick note about this transition...”
- **Tags** (optional): chips; examples: “Angle off”, “Close distance”, “Counter”, “Exit”, “Pressure”. Tags are freeform but suggestions may be provided (no technique claims).
- **Reference Links** (optional): URL list; links are shareable when the flow is shared. See media rules §8.

4.3 Collapsible sections (collapsed by default)

- Fighter Transition
 - Footwork
 - Distance/Range change
 - Angle/Position change
 - Guard/Head movement notes
 - Rhythm/Timing cues
- Opponent State
 - What opponent did / is doing (freeform)
 - Opponent balance/weight shift (optional)
 - Opponent guard/state cues (optional)
- Cues and Intent
 - My cue to trigger next move (freeform)
 - Intent preset (Create space / Close distance / Maintain / Angle off / Custom)
- Feints During Transition
 - List of 0..N feints with timing and notes (reorderable)
- Media (Pro Uploads)
 - Attach private video clip(s) to help you remember (Pro-only; never shared).

- This section must clearly indicate: “Uploads stay private and do not travel with shared links.”
- Advanced / Metadata
 - Confidence / Notes (optional)
 - Created/Edited timestamps (read-only)
 - Revert / Reset controls (see §6)

The section set above is the default. Exact field shapes are defined in §5.

5. Field Specifications (What the editor stores)

All fields are optional. The editor must never block saving because a section is empty. If the user never opens the editor for an edge, that edge simply has no sequence detail record.

Field	Type	Notes
edgeId	string (FK)	Identifies which connection this sequence detail belongs to. Owned by CH13.
flowId	string (FK)	Parent flow. Used for routing, sync, duplication.
sourceMoveId / targetMoveId	string (FK)	Read-only in editor. Derived from the edge (CH13).
quickNote	string	Multiline, optional. Primary “write it down fast” field.
tags	string[]	Freeform tags + suggested chips. Stored as array.
referenceLinks	url[]	Shareable links. Validate format; allow saving even if link unreachable.
fighterTransition	object	Structured optional subfields (see below).
opponentState	object	Structured optional subfields (see below).
cuesAndIntent	object	My cue + intent preset + optional custom.
feints	object[]	0..N entries; reorderable; each entry has timing/type/notes.
privateMedia	object[]	Pro-only uploads; private-only; never shared; subject to CH29 limits.
audit	object	createdAt, updatedAt, createdBy, updatedBy (read-only).

5.1 fighterTransition (structured object)

- **footwork**: freeform text + optional presets (Pivot, Step-through, Switch stance, Drop step, Shuffle, L-step, Circle out). Presets only assist; do not enforce.
- **rangeChange**: slider or enum (Far→Medium→Close) with optional note.
- **angleChange**: enum (Center, Inside, Outside, Rear, Custom) with optional note.
- **headAndGuard**: freeform note (ex: “head off center, hands high”).
- **timing**: enum (Beat, Half-beat, Delay, Reactive, Custom) + optional note.

5.2 opponentState (structured object)

- **opponentAction**: freeform. If the edge is a branch edge with a label, prefill this field with that label (read-only mirror or editable copy; see §7.2).
- **weightShift**: optional freeform or enum (Lead, Rear, Even).
- **guardState**: optional freeform (shell, parry, hands low, framing, etc.).
- **notes**: extra freeform.

5.3 cuesAndIntent (structured object)

- **myCue**: freeform (what you feel/see that triggers next move).
- **intentPreset**: Create space / Close distance / Maintain / Angle off / Custom.
- **intentCustom**: only if Custom selected.

5.4 feints[] entries

- **timing**: Before / During / After / Double (matches earlier concept).
- **type**: freeform (hand fake, level change, target fake, etc.).
- **purpose**: freeform.
- **notes**: freeform.
- **reorder**: Up/Down controls in V1 (drag reorder optional; specify V1 as buttons to keep build feasible).

6. Save, Close, Revert, and Deletion Rules

The editor must be safe against accidental loss while remaining fast to use.

6.1 Save model

- **Explicit Save:** User taps [Save] to persist changes. Auto-save is not required in V1.
- **Save disabled until dirty:** [Save] is disabled unless the user has made changes since last save.
- **After Save:** Show a non-blocking toast: “Saved”. Keep the editor open (do not auto-close).
- **Offline:** If offline, allow save to local draft queue (See: HZ-V1-CH28). UI shows: “Saved locally - will sync when online.”

6.2 Close behavior (unsaved changes)

- If user taps [Close] with no unsaved changes: close immediately.
- If unsaved changes exist: show a confirm sheet with copy:
 - Title: “Discard changes?”
 - Body: “You have unsaved edits to this sequence.”
 - Buttons: [Keep Editing] (primary), [Discard] (destructive).

6.3 Revert options

Revert must be scoped to the **sequence detail record**, not the whole library.

- **Revert this sequence:** Resets the sequence detail to the last saved version.
- **Clear all fields:** Sets all fields empty for this sequence detail (requires confirmation).
- **Restore after clear:** If user clears then has not saved, they can undo by discarding changes; if saved, they can only revert to last saved (which is empty).

6.4 Delete record

- If user wants to remove details but keep the edge: provide action “Delete Sequence Details” inside Advanced / Metadata.
- Confirm copy: Title “Delete details?” Body “This removes notes and metadata for this transition. The connection stays.” Buttons: [Cancel], [Delete].
- Deleting details does not affect practice logs (practice logs reference paths/edges historically; see CH22).

7. Interaction With Branch Labels and Node Types

Sequence details attach to **edges**. Edges can be linear (combo) or conditional (branch). Branch labeling is owned by CH12/CH13; this chapter defines how the editor reflects it.

7.1 Linear edges

- Subtitle shows “A → B”. No condition pill.
- Opponent State fields remain blank unless user fills them.
- Editor does not create new nodes; it only edits metadata for that edge.

7.2 Conditional edges (branch)

- If the edge has a branch condition label (example: “Opponent leans back”), show it as a pill chip under the subtitle.
- Prefill `opponentState.opponentAction` with the label text (editable copy). If the label later changes on the edge, do not silently overwrite the user's longer `opponentAction` note. Instead:
- Label change rule
 - If `opponentState.opponentAction` still equals the old label (unchanged), update it to the new label.
 - If user edited `opponentAction` beyond the label, keep their text and show a small inline note: “Edge label changed - update your opponent note if needed.”

7.3 Optional sequence nodes (if used)

If CH13 introduces a visible “sequence node” type for readability, it must be a view of the same underlying edge metadata. Editing from either the edge tap or the node tap opens the same editor route (`SequenceDetailModal`).

8. Media Rules Inside Sequence Editor

Media rules must match CH29 locks: links are shareable; uploads are Pro-only, private-only, and never shared.

- **Reference Links (all plans that can save flows):** add URL(s) to YouTube/Instagram/etc. These are included when a flow is shared via unlisted link.
- **Uploads (Pro only):** attach 0..N private videos or clips. These never travel with share links or imports; receiving users see a placeholder message instead.
- **Guest users:** can attach links while editing a draft, but cannot persist anything until account creation (See: HZ-V1-CH07).

Required UI copy in Media section:

- “Uploads stay private and will not appear in shared links.”
- If user attempts to share a flow that contains private uploads: show non-blocking reminder in share screen (Owned by CH17/CH29; referenced here).

PLACEHOLDER: Max reference links per sequence • Owner: CH14 • Options: 3 / 5 / 10 • Default: 5 • Decide-by: before R2 locking.

9. Error States and Validation

All error-state patterns must follow CH31 (Error States). This chapter enumerates sequence-specific cases and the exact user-facing copy.

9.1 Validation (non-blocking unless unsafe)

- Reference link format: if the string is not a valid URL, show inline error “That link doesn’t look right.” and disable Save until corrected OR allow save but mark invalid. V1 recommendation: disable Save for invalid URL entries only, while allowing other fields to save.
- Text length: if notes exceed max length (placeholder), show inline counter and prevent further typing; never truncate silently.

PLACEHOLDER: Max characters per note field • Owner: CH14 • Options: 500 / 1000 / 2000 • Default: 1000 • Decide-by: before R2.

9.2 Storage/sync failures

- If save fails due to network: toast “Couldn’t save right now. Try again.” Provide [Retry].
- If local draft queue is full (unlikely in V1): show blocking sheet “Storage full” with actions [Manage Storage] (deep link), [Cancel]. See: HZ-V1-CH29.
- If edge no longer exists (deleted while editor open): close editor and show toast “That transition no longer exists.”

10. Plan Gating Rules (Guest/Free/Pro/Trial)

Global plan states are defined in CH08. This chapter defines what sequence editing is allowed given those states.

- **Guest:** May open and edit sequence details in a draft, but cannot save the flow; attempting to leave builder or save triggers account creation prompt. No local saving for guests (locked in CH00).
- **Free:** May edit sequence details for saved flows (within saved-flow caps and other gating). Practice is paywalled; sequence details can still be edited without practice access.
- **Pro/Trial:** Same as Free + may attach private uploads in the media section (subject to CH29 2GB cap).

If a user on Free taps “Add Upload” in Media: show paywall interstitial (owned by CH25) with return path back to SequenceDetailModal after upgrade.

11. Placeholder Registry (CH14-owned)

Anything not locked must remain explicit placeholders until decided. Owners are listed per CH00 rules.

- PLACEHOLDER: Max reference links per sequence • Owner: CH14 • Options: 3 / 5 / 10 • Default: 5
- PLACEHOLDER: Max characters per note field • Owner: CH14 • Options: 500 / 1000 / 2000 • Default: 1000
- PLACEHOLDER: Show sequence notes during active practice • Owner: CH21 • Options: Off / View-only / Edit allowed • Default: View-only
- PLACEHOLDER: Feint reorder interaction • Owner: CH14 • Options: Up/Down buttons / Drag handle • Default: Up/Down

12. Acceptance Tests (Given / When / Then)

Basic open/close

- Given a flow with two connected moves, when the user taps the edge, then SequenceDetailModal opens showing “Sequence” and “A → B”.
- Given the editor has no unsaved changes, when the user taps Close, then it dismisses immediately.

Save + dirty state

- Given the editor is opened and the user types in Quick Notes, when they have not saved yet, then Save becomes enabled.
- Given the user taps Save, when the save succeeds, then a toast “Saved” appears and Save returns to disabled state.

Unsaved discard prompt

- Given unsaved changes exist, when user taps Close, then a “Discard changes?” confirm appears with Keep Editing and Discard.
- Given user taps Discard, then changes are lost and the modal closes.

Branch label prefill behavior

- Given a conditional edge with label “Leans back”, when editor opens, then the label pill is visible and opponentAction is prefilled with “Leans back”.
- Given the user edits opponentAction to a longer sentence, when the edge label later changes, then the user text remains unchanged and an inline note indicates the label changed.

Plan gating: uploads

- Given a Free user, when they tap Add Upload, then the paywall opens and returning keeps them on the same editor after upgrade.
- Given a Pro user, when they attach a video, then it appears in the private media list and is marked “Private - not shared”.

Edge deleted while open

- Given the editor is open, when the underlying edge is deleted in the canvas, then the editor closes and shows toast “That transition no longer exists.”

Checklist

- Edge tap opens correct editor instance (flowId + edgeId).
- Save/dirty/close prompts behave as specified.
- Branch label pill + prefill rules implemented.
- Uploads gated by plan; links shareable.
- Editor tolerates offline; queued saves sync later (per CH28).

- Accessibility labels present for all controls (See: HZ-V1-CH32).

13. Replit Build Prompt (Chapter-only)

Copy/paste into Replit Agent:

You are implementing Handz V1 chapter CH14 (Sequence Detail Editor) only.

Follow CH00 rules: do not guess cross-referenced behavior; if needed, write assumptions in a PRD Assumptions block and STOP.

Goal:

- Implement a modal route `SequenceDetailModal(flowId, edgeId)` that edits sequence detail metadata attached to an edge.

Requirements:

- 1) Entry: From flow builder edge tap, open `SequenceDetailModal` with source/target move names displayed as "A → B".
- 2) UI: Header with [Close] and [Save]. Save disabled until dirty.
- 3) Fields (all optional): quickNote (multiline), tags (chips), referenceLinks (list add/remove with URL validation), collapsible sections for fighterTransition, opponentState, cuesAndIntent, feints list with up/down reorder, and media uploads (Pro only).
- 4) Save model: Explicit Save. On success show toast "Saved". On failure show "Couldn't save right now. Try again." with Retry.
- 5) Unsaved close: If dirty and Close tapped, show confirm "Discard changes?" with Keep Editing / Discard.
- 6) Branch label interaction: If edge has condition label, show pill chip and prefill opponentAction. If label changes later, apply the rule from CH14 §7.2.
- 7) Plan gating: If Free user taps Add Upload, open paywall then return to same modal after upgrade.
- 8) Edge deleted while modal open: close modal and toast "That transition no longer exists."

Deliverables:

- Data model (TypeScript types) for `SequenceDetail`.
- Storage: attach sequence detail to edge record or separate table keyed by `edgeId` (choose one and justify in PRD Assumptions if CH13 schema isn't provided).
- Unit tests or integration tests for the acceptance cases in CH14 §12.

14. Troubleshooting Notes (Chapter-only)

- Modal doesn't open on edge tap
 - Verify the canvas edge component emits edgeld and flowld on press.
 - Confirm route is registered as modal and receives params.
- Save stays enabled after saving
 - Ensure dirty-state is recomputed after a successful persist using last-saved snapshot comparison.
- Branch label overwrites user's opponent notes
 - Implement the exact label-change rule in §7.2 using "wasEdited" or string comparison against the old label.
- Performance issues while typing
 - Memoize sections; keep large lists (feints) virtualized; avoid rerendering the canvas behind the modal.
- Uploads appear in shared links
 - Double-check CH29 rule: uploads are private-only and must be excluded from share payloads; share renderer should show placeholders instead.