



# CONVERSATIONAL SCENE AUTHORIZING

*Talk to your 3D scene. It listens, it learns, it builds.*

**EON GENESIS 3.0**

---

EON AI Ventures | April 2026

# The Problem: XR Training is Too Hard to Build

Why 90% of companies abandon XR training projects before launch



## Building XR training today requires:



**Specialized 3D developers**

C# or C++ engineers who understand physics engines, shader pipelines, and spatial computing



**6-12 months per scenario**

A single maintenance training scenario takes weeks of coding, testing, and iteration



**\$50K-\$200K per module**

Development costs make XR training viable only for the largest enterprises



**Complex authoring tools**

Unity, Unreal, and PTC Vuforia require deep technical expertise to configure interactions

*The result: only 2% of industrial companies have active XR training programs.*

# How XR Training is Built Today

Every major platform still uses the same paradigm from 2010

## Unity / Unreal Engine

The traditional approach

- ✘ Write C# scripts for every interaction
- ✘ Configure physics colliders manually
- ✘ Wire animation state machines by hand
- ✘ Debug in editor, rebuild, re-test
- ✘ Requires 3D engineering team
- ✘ Months of iteration per scenario

## PTC Vuforia / Scope AR

The "no-code" approach

- ⚠ Drag-and-drop UI with limited options
- ⚠ Pick from preset animation templates
- ⚠ Fill out property forms for each step
- ⚠ Limited to predefined interaction types
- ⚠ No custom physics or sound design
- ⚠ Simpler but rigid and repetitive

*Both approaches share the same fundamental limit: they force humans to translate expertise into UI widgets.*



# What If You Could Just **Talk to Your 3D Scene?**

---

*"Spin the fan and play a whirring sound"*

*"Make it faster"*

*"Wrong direction — reverse it"*

*"Add sparks near the motor"*

# Conversational Scene Authoring

The first XR authoring system you can talk to



## Import

Feed your maintenance manual or SOP document. The system reads it and auto-generates 3D interactions for every step.



## Converse

Watch it run, then talk to fix it. "Make it faster." "Wrong axis." "Add a warning sound." The scene updates instantly.



## See Results

Every change auto-previews in the 3D viewport in real-time. No compile, no rebuild, no waiting. Say it, see it.

# Two-Phase Authoring Model

Automation handles the bulk work. Humans refine what matters.

1

## Automatic Bulk Import

60-70% done automatically

- ✓ Upload SOP document (PDF, DOCX, text)
- ✓ AI parses every step automatically
- ✓ Matches mesh names from 3D model
- ✓ Assigns animations, sounds, particles
- ✓ Generates complete event rules
- ✓ Links back to original SOP steps

2

## Conversational Refinement

Human expertise fixes the rest

- ✓ Play through the generated sequence
- ✓ *"Make that spin faster"*
- ✓ *"Wrong direction — reverse it"*
- ✓ *"Add a warning sound here"*
- ✓ Click any mesh to select it
- ✓ Changes preview instantly in 3D

# A Real Conversation

How a factory supervisor authors XR training in 5 minutes

You: When I click the fan motor, the fan guard opens and the fan ring spins

System: Created 2 effects: fan guard toggles open, fan ring spins continuously with whirring sound. Previewing now.

You: The spin is too slow

System: Doubled spin speed on fan ring. Previewing.

You: Add a warning beep when the guard opens

System: Added warning\_beep sound to fan guard toggle. Previewing.

You: [Clicks pressure gauge in viewport] Make this drop to zero when I click the valve

System: Created new rule: Click valve → pressure gauge animates to 0. Previewing.

*Total time: under 2 minutes. No code. No forms. No 3D expertise required.*

# Click-to-Select: Point and Command

Don't know the part name? Just click it.



## How It Works

- 1 Click any part in the 3D viewport
- 2 Part highlights with blue glow
- 3 Chat shows: "pressure\_gauge selected"
- 4 Type: "rotate this when I click the valve"
- 5 System creates the rule automatically
- 6 Auto-previews the interaction instantly



## Smart Pronoun Resolution

### "Rotate this"

→ Rotates the selected mesh

### "Make it red"

→ Changes selected mesh color

### "Add fire there"

→ Particles at click position

### "The other one"

→ Next mesh in the rule

### "Both of them"

→ All meshes in active rule

# Refinement Commands

13 categories of natural language commands the system understands

## Speed

*"faster", "slow down", "double speed"*

## Direction

*"reverse", "clockwise", "other way"*

## Axis

*"rotate on X", "spin vertically"*

## Sound

*"add hiss", "change to beep", "louder"*

## Volume

*"turn up", "mute", "too quiet"*

## Color

*"make it red", "brighter", "dimmer"*

## Size

*"bigger", "shrink", "scale up"*

## Add Effect

*"add sparks", "put fire there", "also glow"*

## Remove

*"remove sound", "delete the glow"*

## Timing

*"add 2 sec delay", "make it instant"*

## Continuous

*"keep spinning forever", "don't stop"*

## Target

*"not that mesh, this one"*

## Undo

*"undo", "revert that", "go back"*

# Competitive Comparison

Genesis 3.0 vs. traditional XR authoring platforms

Feature	Unity	Unreal	Vuforia	Scope AR	Genesis 3.0
Authoring method	C# code	C++ / BP	Forms	Forms	<b>Conversation</b>
Time per scenario	Weeks	Weeks	Days	Days	<b>Minutes</b>
Developer required?	Yes	Yes	Partial	Partial	No
Auto-generates rules	No	No	No	No	<b>Yes</b>
Natural language input	No	No	No	No	<b>Yes</b>
Click-to-select & talk	No	No	No	No	<b>Yes</b>
SOP document import	No	No	Manual	No	<b>Automatic</b>
Real-time auto-preview	Manual	Manual	Manual	Manual	<b>Automatic</b>
Undo via conversation	No	No	No	No	<b>Yes</b>
Procedural sound	Assets	Assets	Presets	No	<b>22 built-in</b>

# Return on Investment

From months to minutes — quantified savings



**95%**

**Faster Authoring**

Minutes vs. weeks per scenario



**90%**

**Cost Reduction**

\$500 vs. \$50K per module



**0**

**Developers Needed**

Supervisors author directly

Traditional: 6-12 months, \$50K-\$200K, 3D engineering team

Genesis 3.0: 30 minutes, \$500 subscription, factory supervisor

# Who Benefits?

From factory floors to hospitals — anyone who trains on equipment



## Factory Supervisors

Author maintenance training from existing SOPs. No coding, no 3D skills. Upload the manual, talk to refine.



## Safety Officers

Create hazard recognition scenarios by describing dangers in plain language. "When worker touches exposed wire, show spark and alarm."



## Medical Trainers

Build procedure simulations from clinical protocols. Import the checklist, refine timing and visual cues through dialogue.



## Training Managers

Deploy scenarios 95% faster. Update content by talking to it instead of filing change requests with a dev team.

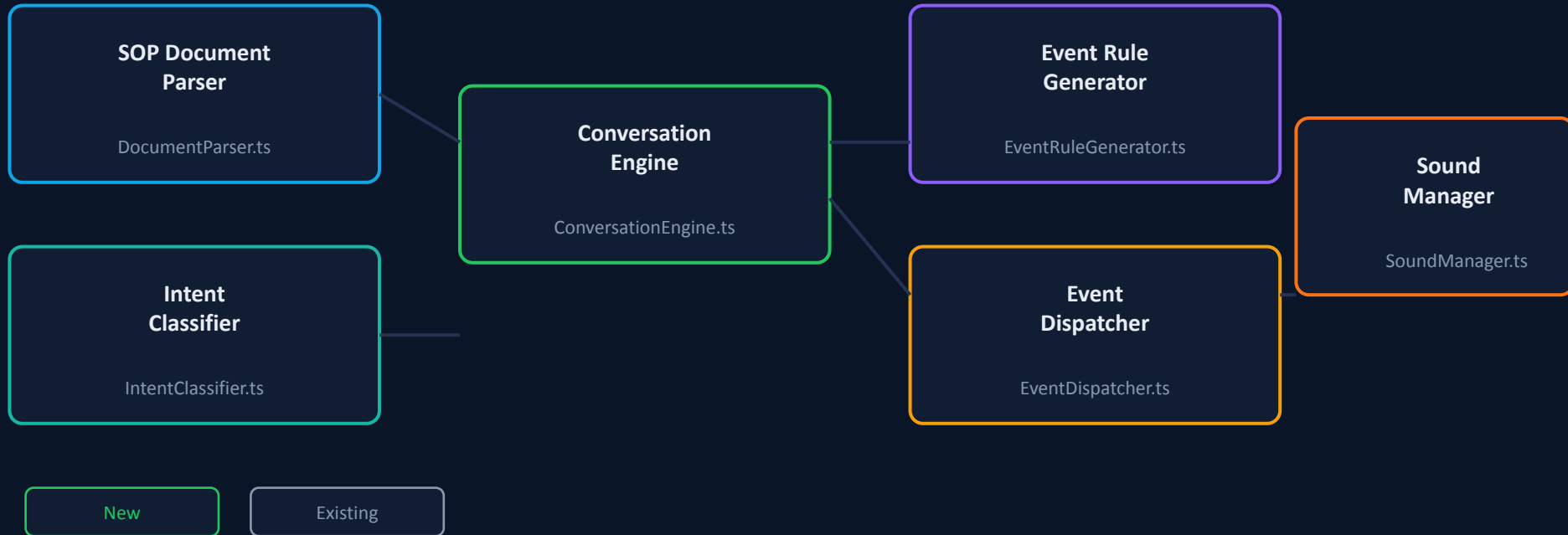


# TECHNICAL ARCHITECTURE

*How it works under the hood*

# System Architecture

5 new modules built on top of the existing Genesis 3.0 engine



~1,600 lines of new TypeScript across 5 modules. Zero changes to existing functionality.

# Intent Classification Engine

Local NLP — no API calls, no latency, no cost

## How the system understands your commands:

<b>new-rule</b>	<i>"When...", "If...", "Make [mesh]..."</i>	Creates a new EventRule via parser
<b>refine-effect</b>	<i>"faster", "reverse", "louder", "clockwise"</i>	Modifies params on active effect
<b>add-effect</b>	<i>"add fire", "also glow", "put sparks"</i>	Appends new effect to active rule
<b>remove-effect</b>	<i>"remove sound", "delete the glow"</i>	Splices effect from rule.effects[]
<b>adjust-param</b>	<i>"color red", "delay 2s", "intensity 0.8"</i>	Sets specific parameter value
<b>question</b>	<i>"play", "preview", "how many rules"</i>	Triggers preview or returns info
<b>undo</b>	<i>"undo", "revert", "go back"</i>	Pops from undo stack, restores state

All pattern matching runs locally via regex. Zero API calls = zero latency = zero token cost.

# Data Flow: From Words to 3D

The complete pipeline from user input to live preview



## User Input

Chat text or SOP document



## Classify

Intent + params extracted



## Modify Rules

Create, refine, or delete



## Dispatch

Load into 3D runtime



## Preview

Auto-play in viewport

## Auto-Preview Loop

Every modification — speed change, direction flip, added sound — immediately plays in the 3D viewport. The author sees the result before typing the next command. No compile step, no save-and-reload cycle.

## Undo Stack

Every change snapshots the full rule set before applying. "Undo" or Ctrl+Z restores instantly. 20 levels deep. Safe experimentation — try anything, revert anything, no risk.

# Procedural Sound Engine

22 built-in sounds generated in real-time — zero audio files needed

## Mechanical

Metal collision, wrench turn, bolt tighten, valve open, gear mesh

## Electrical

Switch click, arc flash, buzzer, relay click

## Feedback

Success chime, failure buzz, warning beep, hint ding

## Environmental

Motor hum, pump running, fan whir, steam release, fire crackle

## Medical

Heartbeat, ventilator, defibrillator charge, surgical tool

*Auto-assigned: When you say "spin the fan", the system automatically adds a fan\_whir sound. No manual sound configuration needed.*



# The Future of XR Authoring Is a Conversation

---

Import your manual. Auto-generate 60+ interactions.

Talk to your scene. Refine in real-time.

Click any part. Command what happens.

No code. No forms. No 3D expertise.

**Minutes, not months.**

**EON AI Ventures**