

Procedural Block-Based USD Workflows in Conduit

Chris Rydalch
Blue Sky Studios

Colvin Kenji Endo
Blue Sky Studios

Wayne Wu
Blue Sky Studios

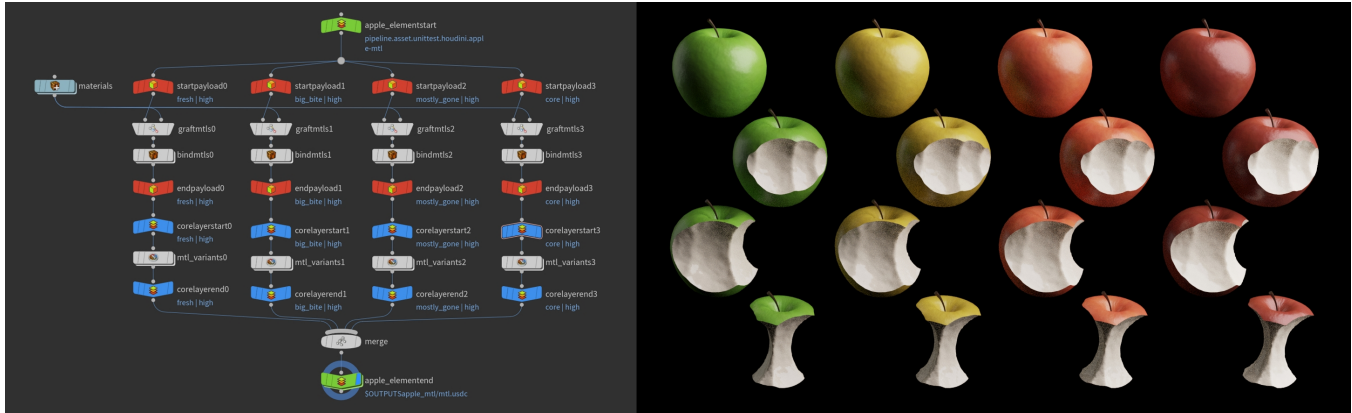


Figure 1: Artist adding materials to variants of an asset with block nodes.

ABSTRACT

We present a procedural block-based approach for USD pipelines that minimizes up-front USD knowledge requirements while ensuring users can still leverage the power of native USD. Building on USD and Conduit, we define fundamental workflow principles and philosophies on artist-interaction that guide our modular Houdini-based toolsets. Finally, we discuss the successes and challenges in scaling these workflows into production.

CCS CONCEPTS

• **General and reference** → **Design**; • **Software and its engineering** → *Software design engineering*; • **Computing methodologies** → **Animation**.

KEYWORDS

pipeline, production, USD, procedural, animation

ACM Reference Format:

Chris Rydalch, Colvin Kenji Endo, and Wayne Wu. 2021. Procedural Block-Based USD Workflows in Conduit. In *Proceedings of SIGGRAPH '21*. ACM, New York, NY, USA, 2 pages. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

In 2019, we introduced Conduit [1] and Universal Scene Description¹ (USD) as the foundation of a new modern pipeline at Blue Sky

¹<https://www.openusd.org>

SIGGRAPH '21, August 09–13, 2021, Virtual

© 2021 Association for Computing Machinery.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of SIGGRAPH '21*. ACM, New York, NY, USA, 2 pages., <https://doi.org/10.1145/1122445.1122456>.

Studios. Since then, we have further integrated Conduit in all applications and deployed new production workflows. The introduction of Houdini Solaris², enabled us to explore and architect powerful procedural tools for artists to work with USD and Conduit.

While all production disciplines at Blue Sky deliver USD content, we explore the disciplines that natively author in USD, bringing forth our design philosophies and experience building procedural workflows in Solaris. As Solaris provides a native USD context, we aimed to expose as much flexibility of USD as possible to users. However, our early prototypes were confusing for artists, particularly in understanding their contributions to the USD and Conduit pipeline. We reformulated a new modular toolset that attempts to ensure USD features are easily accessible while requiring minimal USD knowledge to begin work. These workflows were tested and shown effective in an active studio production workflow, and the design principles can apply to other USD pipelines.

2 BACKGROUND: CONDUIT USD

Conduit provides repository containers, *Products*, that form the basis of the studio's USD pipeline implementation. Products can contain any type of pipeline data, typically USD. An *Entity* is a product composed of contributions from different disciplines. Assets, Scenes, and Shots are all entities in our pipeline, corresponding to *asset.usd*, *scene.usd*, and *shot.usd*. An *Element* is a product containing the individual discipline contributions, and elements compose to create an Entity. Elements usually contain the files that artists work with directly, such as *mtl.usd*, *anim.usd*, or *fx.usd*.

3 CORE PRINCIPLES

Our Solaris workflows are guided by four key design requirements for artist-interaction: Artists should be able to

- (1) Work top-down similar to other Houdini contexts.

²<https://www.sidefx.com/products/houdini/solaris>

