AMWA AS-11

A <u>New Approach</u> to the Design & Publication of AS-11 Specifications

- Rules-based Specifications

- Publishing using GitHub

- The RFC-style publication process

Rules-based Specifications

Rules-based Specifications for AS-11

The AMWA AS-11 family of Specifications define constrained media file formats for the delivery of finished media assets to a broadcaster or publisher. Each Specification is developed for a particular business purpose.

AS-11 Specifications are not required to have any technical properties in common. This means that the different AS-11 Specifications are not required to inter-operate.

However, the Rules-based construction of the Specifications makes it easy to determine what the AS-11 Specifications have in common and how they differ. This helps software developers to maximise code re-use.

It also makes it easy to build new AS-11 Specifications out of the Components of existing Specifications.

A Rules-based Specification is a Specification whose meaning is defined by an interconnected collection of "Components".

Goals of Rules-based Specifications:

- easier to build new Specifications from common Components

- express Specifications in a machine-friendly way

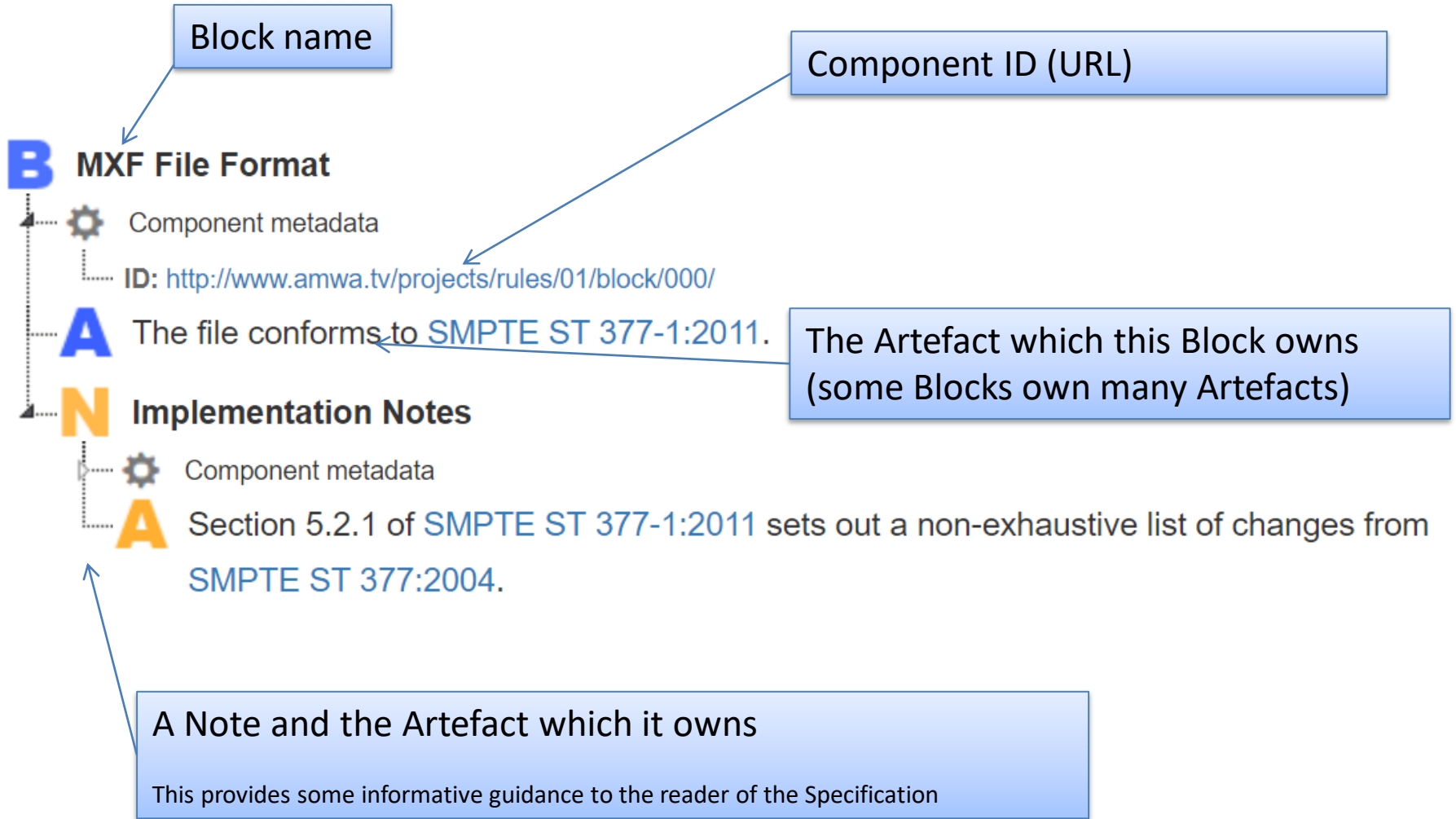- focus on making constraints concise, unambiguous and testable

Rules-based Specification Components:

- Each Specification is composed of **Blocks**

- Each Block exists independently and is identified by a unique ID (URL). A Block could be used in many different Specifications.

- Blocks own **Artefacts** or link to other Blocks

  Artefacts could be statements expressed as prose, XML metadata definitions, Python code files, etc

- There are also **Notes**, **Terms**, and **References**

To read a more detailed introduction to Rules-based Specifications refer to:
http://www.bbc.co.uk/rd/publications/whitepaper319

"Specification Blocks Tree" example

Block name

Component ID (URL)

B **MXF File Format**

⚙ Component metadata

ID: http://www.amwa.tv/projects/rules/01/block/000/

A The file conforms to SMPTE ST 377-1:2011.

The Artefact which this Block owns (some Blocks own many Artefacts)

N **Implementation Notes**

⚙ Component metadata

A Section 5.2.1 of SMPTE ST 377-1:2011 sets out a non-exhaustive list of changes from SMPTE ST 377:2004.

A Note and the Artefact which it owns

This provides some informative guidance to the reader of the Specification

<u>Managing Rules-based Specification Components in AS-11</u>

In the AS-11 Specifications, Components (Blocks, Notes, etc) have IDs with two different prefixes:
http://www.amwa.tv/projects/rules/01/ – AMWA Components
http://vm-1274-user.virt.ch.bbc.co.uk/as/ – Work In Progress (WIP) Components

A Component ID does not convey any meaning. Its purpose is simply to uniquely identify the Component. However, these two different categories of AS-11 Component are managed differently. Note: other projects could choose to manage the Rules-based Specification Components they use in a different way.

AS-11 Specifications that are WIP use either all WIP Components or a mixture of WIP Components and AMWA Components. When an AS-11 Specification is elevated to "Proposed", any WIP Components are converted to AMWA Components and so are assigned new IDs.

AMWA Components are "immutable" which means:
- An "immutable" Component must only link to / reference other immutable Components (so AMWA Components never contain links to WIP Components)
- Changes to "immutable" Components are not permitted except in the following circumstances (in these circumstances it is considered that the "meaning" of the Component is not changed):
    - "Errata" may be fixed – this means that obvious errors and typos can be corrected
    - The "relationships" from Blocks to Notes can be adjusted to add, remove or re-order the Notes that are "attached" to a Block
- For example: this means that (except in the circumstances above) the Artefacts owned by an immutable Block cannot be changed; an immutable Block is not able to change which other Blocks it references.

If an AS-11 Specification is made purely of "immutable" Components this does not mean that the Specification cannot change. The definition of the Specification can simply be updated so that it is made from a different collection of Components: some of these may be the same Components as before and some may be new.

Publishing using GitHub

Publishing using GitHub – Introduction

Specifications are treated like software – each Specification is published as a repository on GitHub. The Specification can be viewed on the GitHub website or cloned / downloaded for offline use.

Each Specification on GitHub has:

- a full change history
- Versioned "releases" of the Specification

Each Specification contains HTML and text renderings (or "views") of the Specification along with useful machine-readable files such as XML Schema files or Python code files.

<u>How does an AS-11 Specification repository relate to its Rules-based Specification Components? – *Short answer*</u>

Someone who wants to read a Specification just needs to look at the GitHub repository for that Specification. That repository includes all the details they need in order to understand the Specification.

Full details of all the Components (Blocks, Notes, and the others) that form the Rules-based Specification are either shown directly in the web page view of the Specification (included in the GitHub repository) or are included in the "specification_data_files" directory that accompanies the web page view.

How does an AS-11 Specification repository relate to its Rules-based Specification Components? – _Long answer_

- Each AS-11 Specification is formally published as a GitHub repository – this repository contains a "package" of content _generated from_ the relevant Components that exist in a separate management system

- A Specification Package consists of <u>copies</u> of some Artefacts _and_ <u>renderings</u> of other Components

- So, the contents of _the GitHub repository must not be directly edited_ – any Specification edits require edits to the Components –> then re-generate all the AS-11 Specification Packages (because an edit to a shared Component will affect more than one Specification) and overwrite the contents of the AS-11 GitHub Specification repositories

- This approach means that the Specifications can persist and be read independently of the Components management system and the tools that are needed to manage and render / package them.

- Many Components will appear in many of the Specifications. That is a good thing and does not affect the process. If a Block is used in three Specifications then its details will simply be duplicated across three Specification GitHub repositories when their packages are generated. The "master" details for the Block are held in the Components management system.

The RFC-style publication process

<u>Using the new AMWA Specification publication process for AS-11 Specifications</u>

Each Specification is published separately as a GitHub repository.

Over its lifetime a Specification moves through three "Maturity Levels" (summarised on the next page).

Publication at all three Levels is public – no need to be an AMWA member to read the Specifications.

The aim is for all AS-11 Specifications to become "Proposed Specifications" as soon as sufficient review has been received for all design decisions to be made.

Specification "Maturity Levels"

Work In Progress

At this level an AS-11 Specification is substantially complete but some design decisions still need to be reviewed and / or additional details added

An AS-11 Specification is elevated to the next level when it has been reviewed and all design decisions have been made

Proposed Specification

At this level an AS-11 Specification can be described as relatively "stable" (there could possibly be bug fixes and improvements to address issues found during real-world implementation & use)

An AS-11 Specification is elevated to the next level when several real-world implementations have been carried out

Specification

At this level an AS-11 Specification is "stable" and several real-world implementations exist