ISO/IEC JTC 1/SC 29/WG 03 N1380

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Exploration on alternative representation of syntax

**Status:** Approved

**Date of document:** 2024-11-08

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 3 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N1380**

**November 2024, Kemer, TR**

|  |  |
| --- | --- |
| **Title** | **Exploration on alternative representation of syntax** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **24443** |

**Introduction**

In this exploration, we examine an alternative method for signaling the SDL syntax that enhances human readability. This method involves placing variable names first and specifying their types following a delimiter. This process is also followed by several modern programming languages such as Swift [1] and Go [2] and could be said that it is based on processes defined in Type Theory [3][4].

The currently specified SDL syntax is as follows:

1 class FooBarClass() {  
2 bit(1) **foo\_bit;**  
**3** unsigned int(7) **bar\_int;**  
4 if( foo\_bit == 1 ) {  
5 ZooClass **zoo\_object;**  
6 }  
7 }

An alternative representation can look as follows:

1 class FooBarClass() {  
2 **foo\_bit :** bit(1)  
**3 bar\_int :** unsigned int(7)  
4 if( foo\_bit == 1 ) {  
5 **zoo\_object :** ZooClass  
6 }  
7 }

It is asserted that the proposed syntax may be easier for a human reader to parse since they are likely more interested first in the variable name and then its type. The use of a colon followed by the type may also make it easier for specification verification systems to parse syntax structures.

It is crucial to emphasize that this alternative method is not intended to replace the current SDL representation. Its sole purpose is to provide an alternative solution that can be utilized in future specifications. Furthermore, the translation from the original SDL representation to the alternative syntax and vice versa should be permitted to facilitate the utilization of potential tools for SDL verification that have been developed for the original SDL representation.

The objective of this exploration is to determine whether the alternative method enhances readability and to assess the additional complexity it introduces in terms of specification definition within the SDL specification.

**References**

1. Swift language, <https://www.swift.org/>
2. The Go programming language, https://go.dev/
3. T. Coquand, "Type Theory". Stanford Encyclopedia of Philosophy, 2006, https://plato.stanford.edu/entries/type-theory/
4. Type Theory – Wikipedia, https://en.wikipedia.org/wiki/Type\_theory