INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG97/M2063
Bristol, UK

Title: **Adding Bitstream Input/Output Library in Java**
Source: **Yihan Fang and Alexandros Eleftheriadis (Columbia University)**
Atul Puri and Robert Schmidt (AT&T)
Group: **MPEG-4 Systems**
Date: **April 1, 1997**

We propose to add a bitstream input and output package mpeg4.bitsio in Java libraries. This will simplify the bitstream Input and Output operations, including fixed length and variable length coding, which are common in MPEG-4 syntax decoding. The implementation should be carefully optimized for maximum speed; the variable length code parsing module uses configurable multi-stage parallel lookups to further aid real-time or near real-time operation.

The proposed library is the Java equivalent of MSDL-S when MSDL-S is used for bit-level I/O. In fact, this library is used internally by the forthcoming MSDL-S to Java translator (`flavorj`) and as such can be used for bitstream parsing in Java if desired. An implementation of this library will be made publicly available shortly after the Bristol meeting.

The proposed class interface is desribed below.

1. Class mpeg4.bitsio.InputStream

```
java.lang.Object
   |
   +----mpeg4.bitsio.InputStream
```

```
public class Input Stream extends Object
```

This class is the basic interface to the input stream.

1.1 Constructors

- **InputStream**

```
public InputStream(FileInputStream file)
```

Constructs a new InputStream from a file.

- **InputStream**

```
public InputStream(string bits, int length)
```

Constructs a new InputStream from a string with given length.

1.2 Methods

- **align**

```
public void align(int numbits)
```

Align with next bit boundary that is a multiple of numbits. Bits between the current pointer and the alignment boundary are read in and discarded.

- **eos**

```
public boolean eos()
```

Return true on end-of-stream, else return false.

- **error**

```
public boolean error()
```

Return true on error, else return false.

- **getbits**

```
public int getbits(int length)
```

Get an unsigned integer from specified number of bits. Sets eos flag on end-of-stream. Sets error flag if stream cannot be read.

- **getsbits**

```
public int getsbits(int length)
```

Get a signed integer from specified number of bits `(length-1)`. The last bit indicates the sign of the integer. Sets eos flag on end-of-stream. Sets error flag if stream cannot be read.

- **nextbits**

```
public int nextbits(int length)
```

Probe next specified number of bits. Return value as 32-bit integer. Do not advance current pointer. Sets error flag if stream cannot be read.

- **skipbits**

```
public void skipbits(int length)
```

Skip specified number of bits. Sets eos flag on end-of-stream.

- **getvlc**

```
public int getvlc(Map map)
```

Get variable or fixed length code accoriding to specified vlc map. Return value as 32-bit integer. Sets eos flag on end-of-stream. Sets error flag if stream cannot be read.

- **nextvlc**

```
public int nextvlc(Map map)
```

Probe variable or fixed length code according to specified vlc map. Return value as 32-bit integer. Sets eos flag on end-of-stream. Sets error flag if stream cannot be read.

2. Class mpeg4.bitsio.OutputStream

```
java.lang.Object
   |
   +----mpeg4.bitsio.OutputStream
```

```
public class OutputStream extends Object
```

OutputStream is the basic interface to the output stream.

2.1 Constructors

- **OutputStream**

```
public OutputStream(FileOutputStream file)
```

Constructs a new OutputStream in a file.

- **OutputStream**

```
public OutputStream(string bits)
```

Constructs a new OutputStream in a string.

2.2 Methods

- **align**

```
public void align(int numbits)
```

Align with next bit boundary that is a multiple of numbits. Bits between the current pointer and the alignment boundary are written out as zeros.

- **align**

```
public void align(string stuffing[], int numbits)
```

Align with next bit boundary that is a multiple of numbits. Bits between the current pointer and the alignment boundary are stuffed according to the stuffing string.

- **eos**

```
public boolean eos()
```

Return true on end-of-stream or error, else return false.

- **error**

```
public boolean error()
```

Return true on error, else return false.

- **putbits**

```
public void putbits(int numbits, int value)
```

Put an unsigned integer using the specified number of bits. Sets error flag if stream cannot be written or `value` is negative.

- **putsbits**

```
public void putsbits(int numbits, int value)
```

Put a signed integer using the specified number of bits. The last bit is set to be sign bit. Sets error flag if stream cannot be written.

- **putvlc**

```
public void putvlc(Map map, int value)
```

Put specified value using the specified map. Sets error flag if stream cannot be written.

3. Class mpeg4.bitsio.Map

```
java.lang.Object
   |
   +----mpeg4.bitsio.Map
```

```
public class Map extends Object
```

A map table for fixed or variable length coding.

3.1 Constructors

- **Map**

```
public Map(FileInputStream file, int step)
```

Constructs a new map reading from a file, using specified step for lookups.

- **Map**

```
public Map(FileInputStream file)
```

Constructs a new map reading from a file. Every lookup advances 1 bit.

- **Map**

```
public Map(string[] bitstring, int[] value, int array_size, int step)
```

Constructs a new map from a string array and an integer array knowing array size and specified step.

- **Map**

```
public Map(string[] bitstring, int[] value, int array_size)
```

Constructs a new map from a string array and an integer array knowing array size. Every lookup advances 1 bit.

3.2 Methods

No user-level methods provided for now.

4. Summary

We have proposed a **bitsio** package to facilitate bitstream input and output with Java. The proposed package provides a general facility for bitstream generation and parsing in Java. It is also used internally in the translator from MSDL-S syntax representation to Java, thus simplifying transition from hand-coded tools to automatically generated ones.