

Validation of a CGM Graphic Against the Web CGM Profile

Stuart Galt <stuart.a.galt@boeing.com>

Abstract

The adoption of the XML Schema Specification by the W3C makes it possible to create machine processible application profiles of formats other than XML. WebCGM is an application profile of the CGM standard which enables to deployment of CGM in the web environment. WebCGM is encoded as binary data and the syntax is much richer than can be expressed rigorously with XML DTD syntax. However, binary CGM can be translated into an XML instance but XML validation tools are not sufficient. By creating an XML Schema to describe the WebCGM profile the XML instance of CGM may be validated using the standard schema tools available, reducing the needs to use proprietary methods that work directly with the binary CGM.

Table of Contents

1. Introduction	1
2. Why use a schema?	2
3. Examples	2
3.1. Metafile Version	2
3.2. Non Graphical Text Strings	3
3.3. Character Coding Announcer	3
4. Conclusion	3
Glossary	3

1. Introduction

What is it Computer Graphics Metafile (**CGM**) (ISO/IEC 8632:1999) is an ISO standard originally created in 1987 for vector and composite vector/raster graphics. **CGM** is used extensively for technical illustrations in the automotive engineering industry, the aerospace industry, and the defense industry. WebCGM is a conforming profile under the CGM:1999 standard, and utilizes the profile proforma described in the CGM:1999 Annex.

The WebCGM profile is used to specify the subset of the **CGM** standard that will be used to exchange data on the web. The detailed profile specifications of WebCGM are based on those of the (Air Transport Association) (**ATA**) **CGM** profile, "GREXCHANGE 2.4". This profile is widely implemented in industry, and it has the valuable attribute that there is already an associated product certification testing service.

WebCGM is an "intelligent graphics" profile, which means that in addition to graphical content based on **CGM** Versions 1-3, the profile includes non-graphical content based on **CGM** Version 4, Application Structures. The non-graphical content allows the definition of hierarchies of application objects, as well as the association of metadata, such as link specifications and layer definitions, with the objects. The WebCGM Profile specification was issued as a W3C Recommendation on 21st January, 1999.

While it is very difficult if not impossible to validate an application against the profile it is possible to validate individual **CGM** graphics. This validation process uses custom written applications that can only validate a graphic with a single profile. If a requirement existed to validate against a new profile, a new application would need to be used.

2. Why use a schema?

It is possible to convert a binary [CGM](#) image into an XML file. This is done by taking graphic primitives and turning them into tags. For example the binary version of circle centered at the point (4252, 2385) with a radius of 150 would look like this :

```
41 86 10 9c 09 51 00 96
```

But converting it to XML we have something like:

```
<Circle len="6">
<Point x="4252" y="2385"/>
<Radius r="150"/>
</Circle>
```

Other [CGM](#) graphic primitives would be converted similarly.

Container elements would be used to represent the boundaries of the internal state changes. For example the metafile descriptor section of the [CGM](#) graphic, which is made up of the elements between the Begin Metafile element and the Begin Picture element, would be marked up with a descriptor section tags even though the descriptor section is not explicitly called out in the [CGM](#) graphic.

[CGM](#) elements that change the graphic state would also be mapped into container elements. This conveys exactly how long the graphic state change is in effect.

So far this hasn't really accomplished anything really very useful yet. It would be very difficult to rigorously represent the WebCGM profile in a DTD. However, if a schema that represents the WebCGM profile can be created then it would be as almost as easy to validate a [CGM](#) graphic as it is to parse an XML file.

3. Examples

Here I will try to show how to take a few clauses of the WebCGM profile and represent them in schema.

3.1. Metafile Version

The Metafile Version tag may appear only once in the [CGM](#) file, and must occur somewhere in the metafile descriptor section, although it is recommended in the standard that this element directly follow the Begin Metafile element.. There are a few other required elements in this descriptor section and could be represented in a schema by something like:

```
...
<element name="MetafileDescriptor">
<complexType>
<all>
<element name="MetafileVersion" >
<attribute name="version" use="required" type="integer" minInclusive="1"
maxInclusive="4"/>
</element>
<element name="MetafileDescription" type="string"/>
...
other required elements go here
...
</all>
</complexType>
</element>
```

The above example is a long way of saying that the Metafile Version tag must occur in the somewhere in the Metafile Descriptor section, but doesn't say exactly where. In addition the allowable values for the required version attribute are 1, 2, 3, or 4.

3.2. Non Graphical Text Strings

A non-graphical text string is string that is not directly displayed on the graphic. An example would be the text associated with the [CGM](#) begin picture element. Non-graphical text strings are required to be less than 254 characters in length. By defining an element NonGraphicText type we can define it as a type of string with a length from 0 to 254 characters. See below for how this might be done:

```
<simpleType name="NonGraphicText" >  
<restriction base="string">  
<minlength value="0"/>  
<maxlength value="254"/>  
</restriction>  
</simpleType>
```

By defining the nongraphic text string attribute as `type="NonGraphicText"` we ensure that the string length requirement is met.

3.3. Character Coding Announcer

The character coding announcer element is required in the descriptor section by the WebCGM profile. In addition the value that must be provided is "basic 8-bit". This requirement can be enforced in a similar way to the above example. The character coding announcer only occurs once in the [CGM](#) graphic so rather than define a new type it would be more efficient to code the restriction of the attribute in-line with the element definition.

4. Conclusion

Currently the author does not know how to represent some of the more complex restrictions in the WebCGM profile. There are several clauses that restrict (or require) an element based on the value of the metafile version. For example if you are in a version 3 metafile and you have a Restricted Text element, then you are required to have a Restricted Text Type element before the first Restricted Text element appears.

Creation of a robust schema that enforces the WebCGM profile will allow a simple XML schema validator validate the conformance of the [CGM](#) graphics. This approach could also be used to validate other binary file types against a profile.

Glossary

ATA	(Air Transport Association)
CGM	Computer Graphics Metafile

Biography

Tuesday, 21 May 16.45

Stuart Galt
The Boeing Company
Seattle
USA

Email: stuart.a.galt@boeing.com

Stuart Galt leads the SGML resource group and specializes in digital data interchange. He has led projects within Boeing to convert aircraft maintenance information to the ATA SGML interchange DTD's.