Based on working draft 6 Aug. 10th 2003.
http://www.w3.org/TR/InkML/
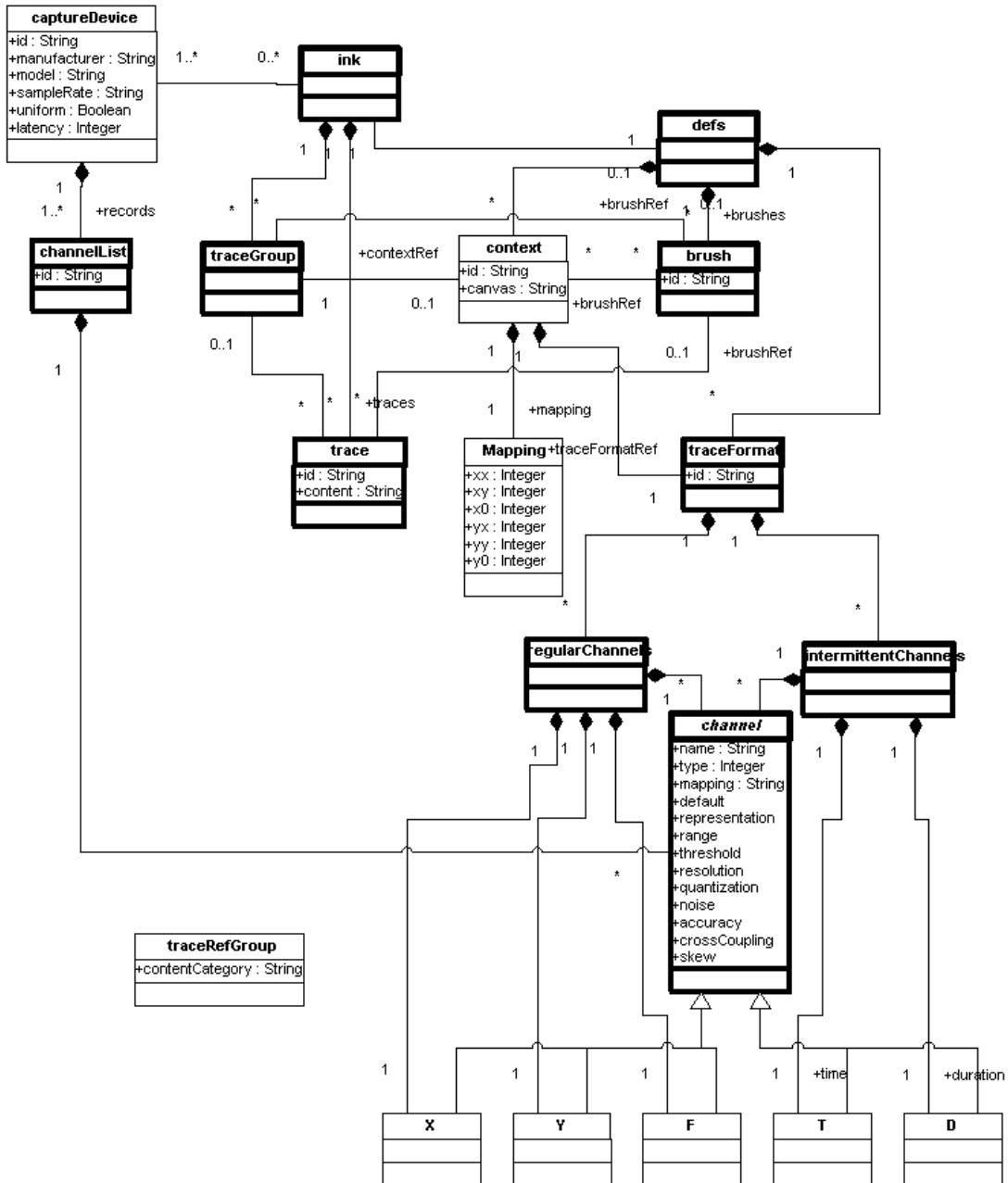
Creation: Thursday, October 30, 2003
Last modification: Tuesday, February 04, 2004

A. Remarks

1. §1.2, 1.3 & 3.1: It is not clear where the 'captureDevice' element can be nested in the XML (in a 'defs'?) for an archival application. I think that the 'context' element should reference the device.
2. §2.1: Encoding values for type 'boolean' are 'T' and 'F'.
   §3.1.1: Values for boolean attribute 'uniform' are 'TRUE' and 'FALSE'. Why not prefer 'T' and 'F'? It would be more consistent with §2.1.
3. §2.1: Some input devices don't not have intermittent channels except:
   1. Time which is available for the first point of the stroke,
   2. Duration which can be associated to the first point of the stroke. Duration can be important, when the point samples are not uniform and when the time is not available for each point.
4. §2.3: Not being able to nest 'traceGroup' (composite model) elements is a limitation. We would probably like to group strokes by pages and by areas => at least 2 levels of nesting.
5. §3.1.2: Units should be expressed using the ISO symbols (ex.: use 'N' instead of 'Newtons', 'ms' instead of 'msec', etc.).
6. §3.3.5: The context should contain a reference to the device (See first remark.). An 'ink' element could contain data from several devices. For traceability, the 'trace' data should be associated to the device that generated them.
7. General remark: A complete sample of InkML is missing.
8. InkML format is maybe too complex (use of references, override mechanism, complex and variable data formats, etc.) making extremely difficult the development of a serializer (especially the importer) and almost illusory the import of data coming from another application.
9. §5.1: It would be convenient if the 'contentCategory' attribute could also qualify the 'traceGroup'.
10. §5.1: ISO 639 letter codes should be imposed to specify the languages.
11. §5.1: When the category is Text, it could be interesting to know if the text contains only digits, a specified set of characters (for example only uppercase letters), symbols, etc, if the text is detached (like in ICR fields) or can be cursive. Regular expressions could be useful to specify the format.

B. Suggested improvements and changes

1. The InkML version should be specified in the XML. A 'version' attribute at root level on 'ink' node?
2. The id of the firmware or application that generated the data could be interesting to store in ink data.
3. To let each application choose a convention to group traces by page violates the principle of exchangeable ink data. How could another application know if the traces belong to different pages and should not be mixed?
4. I think that a class diagram as below could improve the understanding of underlying InkML concepts, entities and their relations.

captureDevice
+id : String
+manufacturer : String
+model : String
+sampleRate : String
+uniform : Boolean
+latency : Integer

ink

defs

channelList
+id : String

traceGroup

context
+id : String
+canvas : String

brush
+id : String

+contextRef

+brushRef
+brushes
+brushRef

+brushRef

+records

+traces

trace
+id : String
+content : String

Mapping +traceFormatRef
+xx : Integer
+xy : Integer
+x0 : Integer
+yx : Integer
+yy : Integer
+y0 : Integer

+mapping

traceFormat
+id : String

regularChannels

intermittentChannels

channel
+name : String
+type : Integer
+mapping : String
+default
+representation
+range
+threshold
+resolution
+quantization
+noise
+accuracy
+crossCoupling
+skew

traceRefGroup
+contentCategory : String

+time +duration

X	Y	F	T	D

C. Questions

1. In the Editor of io™ Software, strokes can be added with the mouse after download. The mouse is another kind of input device with only 2 channels X and Y. Would it be legitimate to define a second 'captureDevice' element for the mouse? Strokes from the pen and strokes from the mouse could then be distinguished easily.
2. Does a schema for InkML exist?
3. Is it feasible to store several pages in a single 'ink' element? Which solution do you propose, without adding artificial offsets to avoid mess between pages and knowing that another application should be able to understand that there are several pages?

D. Logitech's Requirements

1. InkML describes the features of the input device very well but the support (its type: paper, tablet, etc., its size, etc.) is not even mentioned. In the case of io™, the characteristics of paper are as important as the ones of the device.
2. There should be a standard way to store the recognition results of handwritten text for search purposes and maybe the level of confidence. And for highlight purposes it could be interesting to have the corresponding trace/stroke range (use of indexes).
3. We would like to have a complete representative sample of InkML file, if any is available.
4. Limit possible data formats to ease implementation of an InkML data importer.
5. Allow nested 'traceGroup' elements.
6. Standard support of multiple-pages.