



InkML

DARIO DE JUDICIBUS

Quando l'inchiostro diventa elettronico si presenta l'esigenza di sviluppare uno standard per garantire che anche testi scritti a mano possano essere gestiti digitalmente indipendentemente dalla piattaforma e dalla specifica applicazione. Nasce così InkML.

Sempre più interfacce permettono oggi di scrivere testi a mano, ovvero in corsivo, senza utilizzare la tastiera ma piuttosto stili e penne elettroniche in grado di tracciare testi e disegni direttamente sullo schermo di computer o altri apparati elettronici. Se da una parte questo sviluppo rientra nella normale evoluzione delle interfacce grafiche, che con il passare del tempo si propongono secondo modalità sempre più vicine al modo di pensare e operare degli esseri umani, dall'altra essa comporta tutta una serie di vantaggi rispetto alla più consueta tastiera.

Innanzitutto quasi tutti sono capaci di scrivere a mano, mentre non tutti sono in grado di usare una tastiera. Certo, sempre più persone oggi hanno un PC o usano il computer in ufficio, ma di tastiere ne esistono tanti modelli, a seconda della piattaforma operativa e del linguaggio, mentre ognuno di noi è in grado di scrivere con una penna in corsivo o nel peggiore dei casi in stampatello. Quando si scrive a mano, inoltre, si è spesso abituati, specialmente negli appunti, a complementare il testo con disegni, simboli o altri segni che ci aiutano a sintetizzare meglio ciò che stiamo riportando su carta. Inoltre, a seconda del mestiere che facciamo, a volte aggiungiamo equazioni, schizzi tecnici, grafici o altri elementi non necessariamente testuali. È evidente quindi come scrivere a mano comporti una maggiore flessibilità e varietà comunicativa. Eliminare la tastiera permette inoltre di ridurre le dimensioni di dispositivi portatili o, in alternativa, di aumentare quelle a disposizione per lo schermo.

Attualmente esistono già molti dispositivi che permettono di scrivere con il cosiddetto *inchiostro elettronico* o *inchiostro digitale*, ma ognuno utilizza un proprio formato per salvare i dati o trasmetterli a un altro sistema. In generale ci sono due tipi di applicazioni: quelle che permettono di archiviare lo scritto per un'eventuale successiva elaborazione e quelli che invece lo trasmettono in tempo reale a un altro sistema permettendo così l'interazione fra più persone. Un'applicazione del primo tipo è ad esempio quella che su un

palmare permette di registrare piccoli appunti o memorie sotto forma di testi in corsivo e disegni; una del secondo tipo è la lavagna virtuale condivisa, ovvero quel genere di applicazioni aziendali che permettono a più persone di condividere remotamente una finestra dello schermo sulla quale scrivere e disegnare come se fosse una vera lavagna.

Quando parliamo di inchiostro elettronico, molti pensano al riconoscimento della scrittura. In realtà questa è solo una delle possibili elaborazioni che possono essere applicate a un testo elettronico scritto a mano. Dato che scrivere a mano permette una flessibilità molto maggiore che scrivere con una tastiera, non è detto che una pagina elettronica debba necessariamente essere interpretata dal sistema, né è detto che debbano esserne riconosciuti tutti gli elementi. In fondo, quando scriviamo a mano su un pezzo di carta, ci aspettiamo che chi lo riceve lo legga così com'è, senza ulteriori elaborazioni. Il riconoscimento di testi digitali non è sempre banale, dato che la calligrafia varia da persona a persona e che in questo genere di documenti spesso il testo si alterna a simboli personali o altri elementi pseudotestuali, solo alcuni dei quali hanno un equivalente in qualche insieme di caratteri come, ad esempio, quello Unicode.

Ne consegue che la capacità di archiviare segni manuali è indipendente da un'eventuale trasformazione degli stessi in caratteri digitali. L'obiettivo può essere semplicemente quello di riprodurre in seguito tali segni ad uso e consumo di un altro essere umano. Ma se così è, perché non utilizzare semplicemente un'immagine a mappa di bit o al più un formato vettoriale? Primo, perché il sistema che acquisisce lo scritto non può sapere a priori se questo sarà semplicemente archiviato o vi verrà applicato un algoritmo di riconoscimento, per cui, in ogni caso, è necessario catturare tutte quelle informazioni che possono servire anche nel secondo caso. Secondo, perché l'acquisizione di uno scritto manuale non consiste unicamente nella scansione del solo segno, ma va ad analizzare tutta una serie di altre informazioni quali l'inclinazione della penna o la sua rotazione sull'asse nel caso abbia una punta asimmetrica. Anche i movimenti della mano quando la punta non tocca il foglio, in certi casi, possono avere la loro importanza, come ad esempio quando si tratta di convalidare una firma.

Lo standard InkML, quindi, non si occupa di ciò che si intende fare con un testo scritto in inchiostro elettronico, ma solo di poterlo acquisire, memorizzare e, eventualmente, trasmettere in modo del tutto indipendente dalle periferiche di immissione e visualizzazione, dai sistemi operativi, dai protocolli di rete e dalle applicazioni.

Consideriamo un testo qualunque, ad esempio la parola *hello* in inglese. Come si può vedere in figura 1 è possibile vettorizzare i tratti, ovvero identificare una serie di punti la cui interpolazione rappresenti la parola in oggetto.

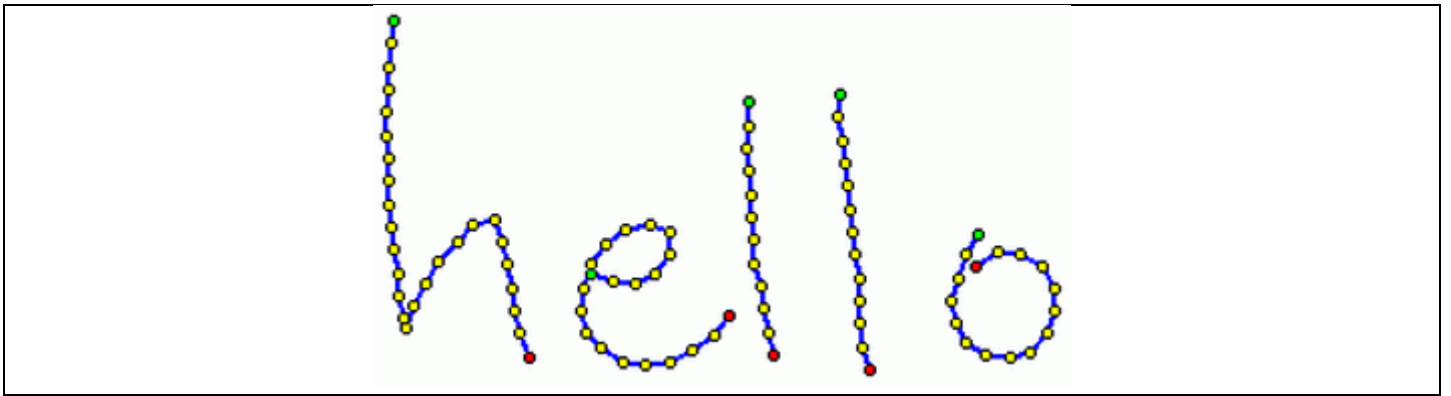


Figura 1 - Un testo scritto a mano

Ci sono tre tipi di punti: quelli in cui la penna è scesa sul foglio e lo ha toccato, segnati in verde in figura, quelli in cui la penna si è staccata dal foglio, segnati in rosso, e quelli in cui la penna è scivolata sul foglio, segnati in giallo. In genere un tratto è formato da una sequenza continua di punti che inizia con un punto verde e termina con uno rosso. Naturalmente una sequenza può incrociarsi se stessa o un'altra sequenza. Nell'esempio in figura, abbiamo 5 tratti, uno per ogni lettera della parola.

Un testo di questo tipo viene reso in InkML come segue:

```
<ink>
  <trace>
    10 0 9 14 8 28 7 42 6 56 6 70 8 84 8 98 8 112 9 126 10 140
    13 154 14 168 17 182 18 188 23 174 30 160 38 147 49 135
    58 124 72 121 77 135 80 149 82 163 84 177 87 191 93 205
  </trace>
  <trace>
    130 155 144 159 158 160 170 154 179 143 179 129 166 125
    152 128 140 136 131 149 126 163 124 177 128 190 137 200
    150 208 163 210 178 208 192 201 205 192 214 180
  </trace>
  <trace>
    227 50 226 64 225 78 227 92 228 106 228 120 229 134
    230 148 234 162 235 176 238 190 241 204
  </trace>
  <trace>
    282 45 281 59 284 73 285 87 287 101 288 115 290 129
    291 143 294 157 294 171 294 185 296 199 300 213
  </trace>
  <trace>
    366 130 359 143 354 157 349 171 352 185 359 197
    371 204 385 205 398 202 408 191 413 177 413 163
    405 150 392 143 378 141 365 150
  </trace>
</ink>
```

In pratica **<ink>** rappresenta la radice di un documento InkML. All'interno del documento è possibile avere uno o più tratti, ognuno contenente una serie di dati numerici che corrispondono ai punti della sequenza. Il modo più semplice per identificare tali punti

potrebbe, ad esempio, essere quello di fornire le coordinate assolute sul piano del primo punto e quelle relative a quest'ultimo di tutti i punti successivi della stessa sequenza.

Ma come si rappresenta un punto in una sequenza? Bastano le coordinate cartesiane del piano? E se invece di una penna a punta fine stessi usando un evidenziatore a punta larga? E se la pressione della penna sul foglio avesse importanza al fine di rendere correttamente il tratto? E come faccio a fornire informazioni sul colore dell'inchiostro?

Tutto ciò può essere fatto con l'introduzione del concetto di *canale*. Un canale, come dice il termine, è una fonte di informazione specifica. Ad ogni canale è associata un'informazione di un certo tipo. Ogni punto, quindi, può essere rappresentato da due o più informazioni che corrispondono ad altrettanti canali. Alcuni canali, quelli cosiddetti *regolari*, forniscono informazioni per tutti i punti; altri, detti *intermittenti*, solo per alcuni.

L'elemento `<traceFormat>` permette di definire quali canali sono utilizzati per descrivere le tracce e quindi come vanno interpretati successivamente i dati memorizzati. Un esempio di questo elemento è riportato qui di seguito:

```
<traceFormat id="xyrb">
  <regularChannels>
    <channel name="X" type="decimal">
      <mapping type="identity"/>
    </channel>
    <channel name="Y" type="decimal">
      <mapping type="identity"/>
    </channel>
  </regularChannels>
  <intermittentChannels>
    <channel name="R" type="integer" default="90">
      <mapping type="identity"/>
    </channel>
    <channel name="B1" type="boolean" default="F">
      <mapping type="identity"/>
    </channel>
  </intermittentChannels>
</traceFormat>
```

Dodici sono i canali previsti dallo standard, ma è possibile definire ulteriori canali specifici per una particolare periferica.

I primi due canali sono ovviamente **X** e **Y**, ovvero le coordinate cartesiane sul piano del punto appartenente a un tratto. Un terzo canale, chiamato **Z**, può essere utilizzato per memorizzare la distanza della penna dal piano sul quale si sta scrivendo. Ad esempio, se la penna simula un aerografo, la distanza dal piano incide sull'ampiezza e la densità del tratto. Il quarto canale corrisponde all'intensità della pressione della penna sul foglio, dove per

«foglio» si intende ovviamente il piano sul quale si sta scrivendo, che sia un foglio flessibile o uno schermo rigido poco importa. Questo canale è detto *forza* ed è simboleggiato da **F**. Il quinto canale è invece **S**, che indica lo *stato* della penna, ovvero se sta toccando il foglio o meno. Questa informazione è utile quando si vogliono memorizzare anche movimenti della penna al di sopra del foglio, ovvero che non danno luogo ad alcun segno fisico. Ad esempio, nel caso si stia tracciando una firma, si può voler registrare l'intero movimento della penna, non solo quello in cui la punta è a contatto con la carta. Un altro possibile utilizzo riguarda l'azione di colpire il foglio con la punta, ovvero l'equivalente per le tavolette grafiche dell'azione di cliccare.

Ci sono poi due coppie di canali che servono a memorizzare l'inclinazione della penna rispetto al foglio. La prima coppia si chiama **Tx** e **Ty** e fornisce gli angoli formati dalla proiezione della penna rispetto alla verticale rispettivamente sul piano ZX e sul piano ZY dello spazio tridimensionale (vedi figura 2).

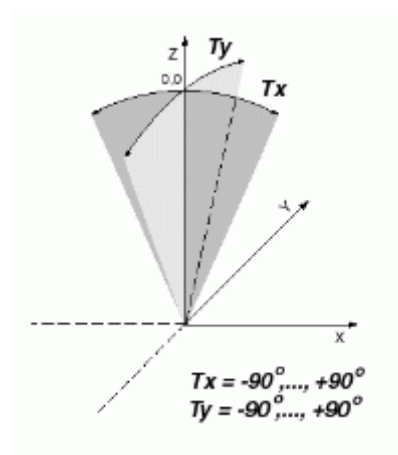


Figura 2 – I canali Tx e Ty

La seconda coppia fornisce la stessa informazione, ma in modo differente, ovvero danno gli angoli di elevazione e di azimut della penna rispetto al piano XY. Si chiamano rispettivamente **E** ed **A**.

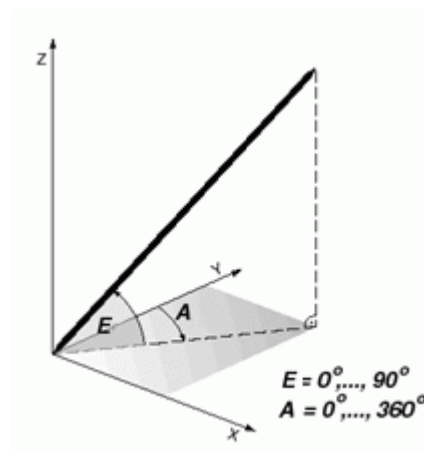


Figura 3 – I canali E ed A

Il decimo canale si chiama **R** e corrisponde alla rotazione della penna attorno all'asse longitudinale. Se infatti la punta simulata dalla penna elettronica non dovesse essere simmetrica o avesse comunque una forma particolare, come avviene per i pennelli, ad esempio, ruotando su se stessa il tratto cambia (vedi figura 4). Ovviamente lo stilo fisico utilizzato potrebbe essere perfettamente simmetrico. Quella che ruota è la penna virtuale simulata da detto stilo.



Figura 4 – Il canale R

L'undicesimo canale è in realtà un raggruppamento di canali. Ognuno prende il nome di **Bn**, dove **n** è un numero intero, e rappresentano eventuali pulsanti la cui pressione può attivare vari tipi di funzioni. Se pensiamo ad esempio ad una tavoletta grafica, i pulsanti in questione potrebbero essere quelli programmabili generalmente posti sul bordo superiore o sinistro della tavoletta stessa. Se invece pensiamo a un programma di disegno artistico, essi potrebbero corrispondere ai vari pulsanti della barra degli strumenti.



Figura 5 - Tavoletta grafica

Il dodicesimo ed ultimo canale è **T**, ovvero il tempo. Questo parametro permette di registrare la velocità con la quale viene mossa la punta della penna, informazione alla quale si possono collegare varie alterazioni del tratto o che permette di applicare regole

euristiche che influenzano il disegno finale. Ad esempio è molto utile negli algoritmi che ottimizzano le curve riducendo il numero dei punti senza alterarne significativamente la forma. Grazie a questo parametro è possibile registrare velocità e accelerazione della punta della penna.

Ma come vengono registrati i dati, ovvero, qual è il formato utilizzato all'interno di un elemento `<trace>`? Un esempio lo abbiamo già visto poco fa, ovvero una semplice sequenza di coordinate corrispondenti ai due canali X e Y. Abbiamo tuttavia detto che alcuni canali possono essere intermittenti e che i dati possono essere sia relativi che assoluti. Come si registrano tutte queste ulteriori informazioni?

Innanzitutto la sequenza dei dati deve contenere un numero intero di ennuple di valori, dove **n** è il numero di canali regolari. Tornando all'esempio del formato **xyrb** visto nell'esempio precedente, ci aspettiamo di avere *almeno* un certo numero di coppie (**x,y**). Se per certi punti sono stati registrati anche i canali intermittenti, nell'esempio **r** e **b1**, allora per quei punti e solo per quei punti avremo uno o due valori in più. Per distinguerli da quelli regolari, i dati corrispondenti ai canali intermittenti sono compresi fra il simbolo dei due punti e quello del punto e virgola. Ad esempio

```
12 34 15 32 16 39 :45; 18 * :90 T; 20 40
```

Il carattere jolly, ovvero l'asterisco, indica che il valore del canale è identico a quello registrato in precedenza. Come si può vedere dalla sequenza sopra riportata, solo i canali regolari sono sempre presenti. Quelli intermittenti possono esserci o meno e, se ci sono, possono contenere solo alcuni dati. I dati sono separati da un carattere di spaziatura che, tuttavia, può essere omesso in determinati casi, ovvero quando sono presenti i segni di interpunzione che delimitano i canali intermittenti o nel caso il dato sia preceduto dai cosiddetti *prefissi*, di cui parleremo più avanti. Questo vuol dire che non è possibile omettere i valori dei canali intermittenti in testa o in mezzo alla lista, ma solo quelli in fondo. I dati omessi, infatti, vengono interpretati come asterisco, ovvero considerati uguali al valore precedente, e non come *default*, per cui per ometterli basta sostituirli con tale carattere. Ad esempio

```
12 34 15 32 16 39 :45; 18 * :* T; 20 40
```

Un eventuale separatore esplicito, come una virgola, infatti, sarebbe superfluo in quanto richiederebbe un numero maggiore di caratteri che aumenterebbe le dimensioni del flusso di dati. È importante comprendere come sia stato fatto infatti un grosso sforzo per ridurre al minimo le dimensioni dei dati all'interno delle tracce.

Ogni canale può essere definito tramite tre parametri, dei quali solo il primo è obbligatorio: **name**. Ad esempio, l'azimut ha per nome **A**, come già detto. Gli altri due parametri sono **type** e **default**. Il primo indica se il valore è intero (**integer**), un numero decimale (**decimal**) o un booleano (**boolean**). Questi ultimi sono codificati con i due caratteri **T** (vero)

e **F** (falso). Se non viene specificato nulla, si assume un valore decimale. Il secondo fornisce un eventuale valore di *default* da utilizzare all'inizio del flusso di dati.

I valori forniti negli esempi precedenti sono detti *espliciti*, ovvero sono valori assoluti. Per indicare che un certo valore è invece relativo a quelli precedenti, si usano i *prefissi*, ai quali abbiamo accennato in precedenza. Se un valore è preceduto da un apostrofo (**'**), allora quel valore rappresenta una variazione rispetto al valore precedente. Nel caso di valori spaziali, possiamo pensare a tale variazione come a un'indicazione di velocità. Se invece un valore è preceduto da una virgoletta (**`**), allora quel valore rappresenta una variazione rispetto alla variazione precedente. Nel caso di valori spaziali, possiamo pensare a tale variazione come a un'indicazione di accelerazione. Una volta applicato un prefisso al valore di un canale, tutti i valori seguenti dello stesso canale sono interpretati come se avessero lo stesso prefisso. Questo almeno fintanto che il prefisso non cambia oppure viene usato il punto esclamativo (**!**) ad indicare che il valore è tornato ad essere esplicito, ovvero assoluto. Ad esempio, la sequenza

12 34 15 32 '4 '5 6 2 `1 `-3 !30 !36

è equivalente alla sequenza

12 34 15 32 19 37 25 39 32 38 30 36

Questo perché **'4** sposta di 4 unità il valore precedente di **x**, **6** sposta tale valore ulteriormente di 6 unità (il prefisso è implicitamente ancora **'**), mentre **`1** aumenta di uno l'ultima variazione, e quindi sposta la **x** di altre 7 unità. Dopodiché il valore seguente è di nuovo assoluto. Analogamente per la **y**. Il primo valore di ogni canale all'interno di una traccia deve sempre essere esplicito, anche se la traccia è la continuazione di una traccia precedente.

Di nuovo, è importante notare come si sia fatto un certo sforzo per definire una sintassi che utilizzi il minor numero possibile di caratteri per i dati. La ripetizione implicita dei prefissi, il fatto di non usare troppi tipi differenti di separatori, l'utilizzo dell'asterisco rientrano in tale strategia.

Non sempre i dati che provengono da una periferica vengono registrati così come sono stati forniti. A volte è necessario applicare ai dati in ingresso una qualche forma di elaborazione prima di memorizzarli nel modo che abbiamo visto. Se questo è il caso, allora il tipo di trasformazione applicata è riportato nell'elemento **<mapping>** all'interno delle specifiche del canale, come mostrato nell'esempio illustrato in precedenza per l'elemento **<traceFormat>**. Quando il valore acquisito è quello registrati sono identici, allora il tipo di mappatura è **identity**. Il valore **lookup** invece indica l'utilizzo di una tabella di trasformazione. Questa tabella può anche essere utilizzata per creare delle dipendenze di un canale da un altro (*coupling*). Ad esempio, permette di modificare sistematicamente il valore di **x** quando **R** assume valori multipli di 45°. Se invece è richiesta un'elaborazione più

complessa di quella fornita da una semplice tabella, allora si può usare la mappatura **mathml**. In questo caso lo sviluppatore dovrà fornire la formula di trasformazione utilizzando un sottoinsieme del linguaggio MathML.

L'elemento **<trace>** può avere anche uno o più attributi. Ad esempio **type**, che permette di specificare se la penna è a contatto con il foglio (**penDown**), se è sollevata (**penUp**), se ha la stessa posizione della traccia precedente (**continuation**) o se la sua posizione non è conosciuta a priori (**indeterminate**). Se per questo attributo non è fornito alcun valore, si assume uno stato indeterminato. Più tracce possono inoltre essere raggruppate all'interno di un elemento **<traceGroup>**.

Questi sono solo alcuni elementi e attributi di InkML. Questo linguaggio permette infatti di specificare anche le caratteristiche della penna o del pennello usato (**<brush>**), quelle del foglio o comunque del contesto all'interno del quale vengono tracciati i segni (**<context>**), le caratteristiche della periferica di acquisizione delle tracce (**<captureDevice>**) e, naturalmente, di definire ulteriori canali attraverso l'elemento **<channelDef>** che comprende a sua volta diversi elementi per la definizione delle caratteristiche del canale come **<resolution>**, **<distortion>**, **<noise>**, **<accuracy>** e molti altri ancora.

Ad esempio:

```
<channelDef name="S">
  <representation type="boolean"/>
  <threshold value="0.2" units="N"/> <!-- N sta per Newton -->
  <skew value="8" units="ms"/> <!-- ms sta per millisecondi -->
</channelDef>
```

Per quanto riguarda la penna e il contesto, tuttavia, dato che le caratteristiche di questi elementi possono essere estremamente varie, a seconda della periferica e dell'applicazione di acquisizione dei tratti, l'InkML si rifà alla disponibilità di un non meglio precisato linguaggio di livello superiore.

Una descrizione dettagliata di questo linguaggio può essere trovata sul sito del W3C all'indirizzo <http://www.w3.org/TR/InkML>.