Azzolini Riccardo 2020-03-24

Modelli dei dati semi-strutturati e XML

1 Linguaggi di markup

In generale, un **linguaggio di markup** è un linguaggio che permette di annotare dei testi/dati, con delle annotazioni, dette appunto **markup**, che sono sintatticamente distinguibili dai testi/dati annotati. Alcuni esempi sono HTML e TEX / LATEX.

Due caratteristiche tipiche dei linguaggi di markup sono le seguenti:

- i testi/dati e i relativi markup markup sono contenuti in un unico file/documento;
- i markup sono solitamente fissi, predefiniti (ma non sempre: alcuni linguaggi, come TEX / LATEX, permettono la definizione di nuove annotazioni).

2 XML

XML (**eXtensible Markup Language**) può essere visto come un linguaggio per definire linguaggi di markup: le annotazioni, chiamate **tag**, non sono predefinite, bensì definite dall'utente. Ad esempio, XML può essere usato per definire HTML (XHTML).

XML, così come HTML, deriva da SGML (Standard Generalized Markup Language), un linguaggio di markup ideato inizialmente per annotare testi giuridici, e caratterizzato da una specifica molto ampia e complicata; infatti, sia XML che HTML si basano solo su un piccolo sottoinsieme di SGML. Mentre SGML risale agli anni 80, la prima versione dello standard XML è stata definita dal W3C nel 1998.

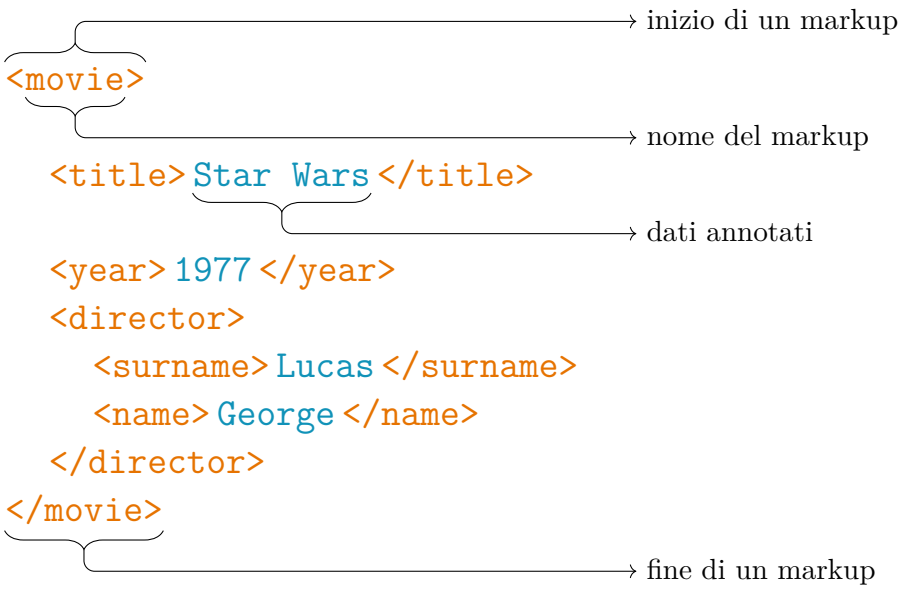
3 XML per la descrizione dei dati

Considerando XML come uno strumento per descrivere/annotare dati, si possono interpretare i tag come lo "schema" dei dati, e, di conseguenza, i dati annotati come l'istanza di tale schema. Quindi, anche se concettualmente distinti, schema e istanza sono contenuti fisicamente in un unico documento. Questa è una delle differenze fondamentali rispetto ai modelli relazionale e object-oriented, nei quali schema e istanza sono fisicamente separati.

La nozione di schema dei dati è comunque molto importante in XML. Perciò, esistono numerosi linguaggi di definizione di schemi XML, tra cui DTD (il primo storicamente definito) e XML Schema.

4 Esempio di dati annotati con XML

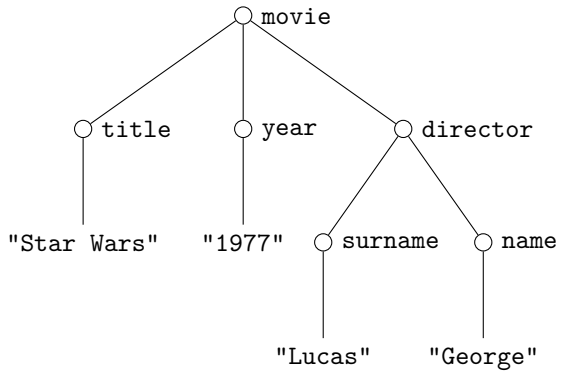
Come esempio, le informazioni relative a un film potrebbero essere annotate con i seguenti markup XML:



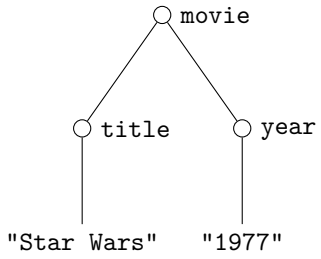
Questa rappresentazione testuale può essere contenuta in un file/documento XML (che solitamente ha estensione `.xml`). L'indentazione non importa, quindi si potrebbe scrivere, equivalentemente:

```
<movie><title>Star Wars</title><year> ··· </movie>
```

In realtà, la forma testuale è solo una **serializzazione** (ottenuta scrivendo dati e tag in modo sequenziale) della struttura ad **albero** dei dati:



Il motivo per cui XML è definito un modello dei dati **semi-strutturato** (e non semplicemente strutturato, come potrebbe suggerire la forma ad albero dei dati) è che *non è obbligatorio* specificare sempre tutti i tag. Ad esempio, se non fossero note le informazioni relative al regista di un film, si potrebbe semplicemente omettere il tag `<director>`:



Questa è un'altra differenza radicale rispetto al modello relazionale, nel quale, in questo caso, l'attributo relativo al regista dovrebbe per forza essere comunque presente, e allora sarebbe necessario inserirvi un valore nullo.

Per interrogare i dati descritti in XML, che sono appunto organizzati gerarchicamente in alberi aventi "più o meno" la stessa forma (cioè, appunto, semi-strutturati), sarà necessario un linguaggio di interrogazione ad-hoc.

4.1 Possibile traduzione al modello relazionale

Per rappresentare nel modello relazionale relazionale (o meglio, object-relational) dell'istanza di `<movie>` riportata sopra, si potrebbe scegliere di definire una tabella con tre attributi (`title`, `year` e `director`), usando un row type per memorizzare cognome e nome del regista in una singola colonna:

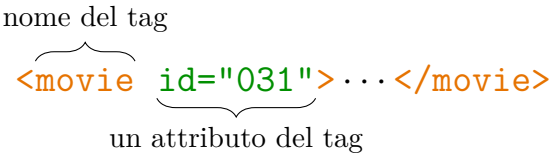
`title`	`year`	`director`
Star Wars	1977	`('Lucas', 'George')`
…	…	…

Allora, se non fosse noto il regista, gli attributi del row type assumerebbero valori nulli:

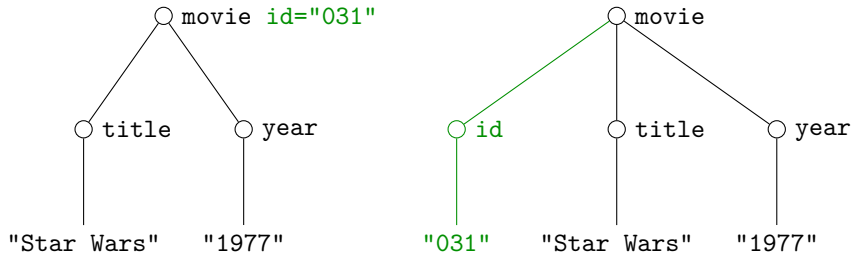
title	year	director
Star Wars	1977	(NULL, NULL)
…	…	…

4.2 Attributi

Oltre al nome, i tag XML (ovvero i nodi dell'albero) possono avere anche degli attributi, costituiti da coppie chiave-valore. Ad esempio:



Nota: Le rappresentazioni di un dato come attributo o come tag sono equivalenti. La scelta tra un attributo e un tag spetta al progettista dello schema. Per esempio, l'`id` di un film potrebbe essere rappresentato equivalentemente nei due modi seguenti:



5 Modelli dei dati semi-strutturati

Da un punto di vista sufficientemente astratto, un documento XML è un albero

- *"labellato"* (etichettato: i label/etichette sono i nomi dei tag),
- *con attributi*,
- *ordinato* (si considera l'ordine dei figli di ciascun nodo),
- *unranked*[1] (cioè senza limite sul numero di figli che un nodo può avere).

[1] Il contrario è un albero *ranked*, nel quale ogni nodo può avere al massimo un numero prestabilito, detto rango, di figli. Per esempio, il rango 2 corrisponde a un albero binario.

In generale, un modello dei dati semi-strutturato è una definizione precisa, formale di un albero di questo tipo.

Non esiste un unico modello dei dati semi-strutturato, che abbia un consenso di utilizzo paragonabile a quello del modello relazionale (ovvero, in pratica, che sia uno standard de-facto). Anche solo in riferimento a XML, esistono modelli diversi, tra cui *XPath Data Model* e il più complesso *XML Infoset*, entrambi standardizzati dal W3C.