

Struttura del codice

Suddivisione del codice

Si è suddiviso il codice in tre livelli, che corrispondono ai tre tipi di codice utilizzati. Ossia HTML, CSS e javascript. Tali livelli si sono affrontati dal basso verso l'alto.

1. Si comincia producendo il *contenuto* in formato HTML. Questo si realizza nel file *WorldPipesNew.html*.
2. Si continua aggiungendo un livello di *presentazione* tramite i fogli di stile CSS. Tutti i file .css sono contenuti nell'omonima directory *CSS*.

- *actionListenerStyle.css*
- *drag&dropstyle.css*
- *guidastyle.css*
- *pagestyle.css*
- *tablestyle.css*

3. Infine si utilizza Javascript per introdurre un altro livello d'interattività e di *comportamento* dinamico. Tutti i file .js sono contenuti nella directory JS.

- *actionListenerImg.js*
- *actionListenerResizeX.js*
- *actionListenerResizeY.js*
- *activelink.js*
- *areahelp.js*
- *Component.js*
- *Endpoint.js*
- *InputType.js*
- *Form.js*
- *SourceCode.js*

Non tutto il codice javascript è stato costruito partendo da zero, si sono utilizzate le librerie:

- *Core*
- *jsPlumb*
- *Ajax*

Librerie utilizzate

1. Core

Ogni file .js conta su una piccola libreria di metodi contenuta nel file *Core.js*, che troviamo nella directory JS. Questa libreria racchiude tutte le sue funzioni in un oggetto javascript, trasformandole in metodi di tale oggetto. I metodi più utilizzati sono:

- *Core.addEventListener(target,type,listener)*: questo metodo assegna la funzione *listener* come un listener di eventi di tipo *type* all'oggetto *target*.
- *Core.start(runnable)*: questo metodo esegue il metodo *init* dell'oggetto script (*runnable*) non appena termina il caricamento del documento.

Si utilizzano inoltre vari metodi per la gestione delle classi CSS:

- *Core.addClass(target,Class)*;
- *Core.getElementsByClass(Class)*;
- *Core.hasClass(target,Class)*;
- *Core.removeClass(target,Class)*.

2. JsPlumb

jsPlumb offre la possibilità di collegare visivamente elementi delle nostre pagine HTML, utilizzando quelle che vengono chiamate *Connessioni*. Un collegamento appare come una linea - il percorso che segue dipende dal tipo di connessione in uso. jsPlumb fornisce due tipi di attacchi - una linea retta, e una curva di Bezier; si è scelto di utilizzare quest'ultima in World Pipes.

Ogni connessione viene stabilita tra due *endpoint*. Un endpoint è il punto sull'elemento cui il collegamento si unisce. jsPlumb fornisce tre diversi tipi di endpoint: un punto, un rettangolo e un'immagine; si è scelto di utilizzare il punto.

Infine ogni endpoint è associato a un *anchor*. Essi definiscono, dove un endpoint deve essere

posizionato su un elemento. Non hanno alcuna rappresentazione visiva, sono semplicemente utilizzati per il posizionamento.

3. Ajax

Ajax, acronimo di *Asynchronous JavaScript and XML*, è una tecnica di sviluppo software per la realizzazione di applicazioni web interattive. Ci consente di comunicare con il server e di inviargli i dati. Questa libreria ci permette di chiedere al browser di gestire piccole porzioni di documento invece che grossi frammenti d'informazioni; invece di una pagina si può richiedere di inviare al server anche un unico paragrafo. Questo è quello che succede nella nostra applicazione; al server inviamo solamente il codice RDF generato e non tutta la pagina web.

Gli script

- a. *actionListenerImg.js*: In questo script si gestiscono gli eventi scaturiti dall'azione del mouse. Il click del mouse sui pulsanti New, Save, Play e Load pipeline; e drag & drop dei componenti nell'area editor; ossia la creazione del componente stesso.
- b. *actionListenerResizeX.js*: Gestisce il resize verticale dell'area editor.
- c. *actionListenerResizeY.js*: Gestisce il resize orizzontale dell'area editor.
- d. *activelink.js*: Rende interattivi i tab Dataset e Pipeline.
- e. *areahelp.js*: Cliccando su uno dei componenti nell'area helper compare la descrizione del componente cliccato, in questo script viene gestito tale evento.
- f. *Component.js*: Ogni componente trascinato nella'area editor è contenuto nel vettore dei Componenti. Tale vettore è un array di classi. In questo script si gestiscono tutti i metodi get e set per lavorare su tale vettore.

Il vettore dei Componenti è così strutturato:

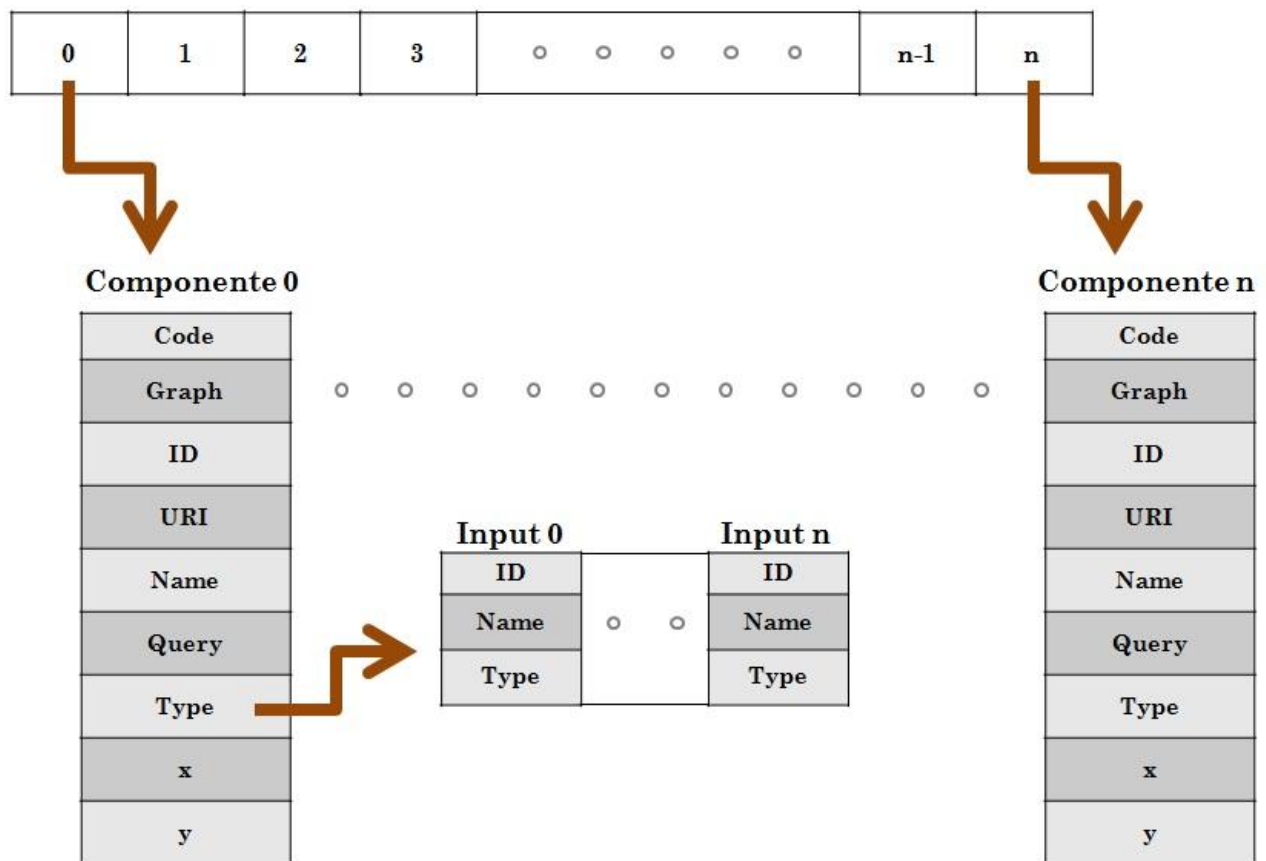


Figura 1

Endpoint.js: In questo script si gestiscono i metodi della libreria jsPlumb per gestire i collegamenti tra i componenti

Form.js: In questo script si realizzano i form per permettere all'utente di inserire le proprietà di ogni componente.

InputType.js: Questo script permette di inserire l'oggetto inputType nel campo TYPE del vettore componenti.

SourceCode.js: In questo script viene realizzato il codice RDF della pipeline creata dall'utente. Inoltre si gestisce anche l'invio del codice stesso al server.