

Rapporto Tecnico

TR3.2.1

**Analisi dei requisiti, specifiche e architettura applicativa della
soluzione software per la costruzione di *outline* narrative**

Parte I.

L'ambiente di *authoring* per la creazione di *outline* narrative

Responsabile: Arnaldo Cecchini

Afferenza: Università degli Studi di Sassari

Autori: Ivan Blecic, Arnaldo Cecchini, Andrea Trunfio

Afferenza: Università degli Studi di Sassari



Abstract

Questo documento presenta i requisiti e l'analisi funzionale dell'ambiente di *authoring* per la creazione di *outline* narrative. Sono descritti nel dettaglio i casi d'uso dell'ambiente, la struttura della base di dati *back-end* e la struttura e lo schema dello *Story-Outline Modelling Language* (SOML). Infine, sono presentate e descritte le interfacce utente dell'ambiente di *authoring* sviluppato.

1. INTRODUZIONE

In questo documento sono descritti i requisiti tecnici e l'analisi funzionale dell'ambiente di *authoring* per la creazione di *outline* narrative, come anche le caratteristiche del software prototipale sviluppato sulla base di tali requisiti ed analisi.

Nel contesto dell'impostazione metodologico-progettuale generale di trattamento dello *story outlining*, delineata nel documento "Metodologie di *storytelling* integrato" (cfr. rapporto tecnico TR3.1), l'ambiente di *authoring* deve soddisfare i seguenti requisiti funzionali generali:

- di efficacemente rappresentare e visualizzare strutture ed aspetti rilevanti di storie;
- di efficacemente supportare una collaborazione tra più autori;
- di essere facile da usare e comprendere (e dunque di essere usabile sia ai fini amatoriali, sia per l'*authoring* professionale);
- di essere sufficientemente generalizzato per consentire la rappresentazione e trattamento di un'ampia gamma di tipi e generi di storie (sia per la narrativa letteraria che per narrazioni documentaristiche);



- di consentire una rapida prototipazione con anteprima di storie, di tipo visuale e strutturale;
- di consentire un'archiviazione strutturata di storie per una loro facile ricerca e riuso.

A tal fine, come meglio illustrato nel suddetto documento metodologico, è stata adottato ed adattato per questo obiettivo uno specifico approccio appartenente alla famiglia di tecniche orientate alle mappe concettuali [1].

Infatti, la *specializzazione* del paradigma delle mappe concettuali per i fini dello *story outlining* all'interno del progetto LC3 è stata operata attraverso le seguenti estensioni e caratterizzazioni:

1. È stata prevista una linea temporale (*timeline*), designante il tempodurata della storia di cui viene fatta l'*outline*, che rappresenta la principale struttura portante di tutte le altre entità aggiunte all'*outline* narrativa.
2. Una storia è intesa come un'insieme di *attività* (cfr. il rapporto tecnico TR3.1) cui inizio e fine sono indicati (associati) sulla linea temporale.
3. Ogni attività è descritta da due tipi di mappe propriamente appartenenti alla famiglia delle mappe concettuali: (a) le *mappe di struttura* e (b) le *mappe di comportamento* (cfr. il rapporto tecnico TR3.1). Questi due tipi di mappe sono caratterizzati da alcuni vincoli strutturali e compositivi rispetto alle generiche mappe concettuali.

In considerazione di questo approccio generale, si procede all'analisi dei casi d'uso.

2. ANALISI DEI CASI D'USO

Le principali procedure e dell'ambiente di *authoring* di *story outline* possono essere suddivise nei seguenti casi d'uso:

1. Creazione ed inizializzazione di una nuova *outline*,
2. Configurazione della linea temporale,



3. Configurazione dei tipi di nodi ed archi,
4. Popolamento della linea temporale (inserimento di attività),
5. Inserimento di eventuali altri segnatempo sulla linea temporale,
6. Inserimento di una mappa di struttura,
7. Inserimento di una mappa di comportamento,
8. Inserimento di nodi ed archi.

Segue una più dettagliata analisi di questi casi d'uso.

2.1. CREAZIONE ED INIZIALIZZAZIONE DI UNA NUOVA OUTLINE

In questo caso d'uso l'utente accede alla funzionalità di creazione di una nuova *outline*. Per la sua inizializzazione, l'utente deve specificare:

1. Il titolo della *outline*
2. Una descrizione testuale della *outline*

2.2 CONFIGURAZIONE DELLA LINEA TEMPORALE

In seguito alla creazione di una nuova *outline*, è necessario specificare le caratteristiche della linea temporale della storia. Questa specificazione avviene a partire dalla scelta del tipo di linea temporale di base da utilizzare (cfr. il rapporto tecnico TR3.1):

1. Linea temporale *isocrona*, per la quale è *necessario* indicare:

La data/ora d’inizio e di fine della storia

2. Linea temporale *anisocrona*, per la quale è *facoltativo* indicare:

la data/ora d’inizio e di fine della storia

2.3 CONFIGURAZIONE DEI TIPI DI NODI ED ARCHI

La procedura di inizializzazione di un’*outline* prevede la configurazione e dichiarazione dei tipi di nodi ed archi da utilizzare all’interno delle mappe di struttura e di comportamento. Per questa configurazione e dichiarazione, l’utente deve:

1. Per i nodi:

- 1.1. scegliere una rappresentazione grafica (una combinazione di forma di poligono e di colore) per ogni tipo di nodo;
- 1.2. attribuire un’etichetta testuale (che appartiene alla sfera semantica delle tipologie di nodi da usare nella *outline*, come ad esempio: “personaggio”, “oggetto”, “fenomeno” e così via) per ogni tipo di nodo;

2. Per gli archi:

- 2.1. scegliere una rappresentazione grafica (una combinazione di forma di tratto di linea e di colore) per ogni tipo di arco
- 2.2. attribuire un’etichetta testuale (che analogamente al caso dei nodi appartiene alla sfera semantica delle relazioni tra nodi che si intende utilizzare all’interno dell’*outline*) per ogni tipo di arco.



2.4. POPOLAMENTO DELLA LINEA TEMPORALE (L'INSERIMENTO DELLE ATTIVITÀ)

Una volta inizializzata e configurata l'*outline* attraverso le procedure cui casi d'uso sono stati descritti prima, l'utente può procedere alla creazione della *outline* attraverso l'inserimento delle *attività* della storia all'interno della linea temporale.

Per ogni attività inserita è necessario specificare:

1. Il punto (tempo) di inizio dell'attività (questo punto deve poter essere inserito sia attraverso l'interfaccia grafica sulla rappresentazione visiva della linea temporale, sia attraverso l'interfaccia di proprietà dell'attività inserendo il valore del tempo in formato numerico)
2. Il punto (tempo) di fine dell'attività (con le medesime modalità di interazione con l'interfaccia come per il punto d'inizio)
3. Il titolo dell'attività
4. Una descrizione testuale dell'attività
5. Le caratteristiche della linea temporale "locale", con le modalità analoghe a quelle descritte nel sottocapitolo 2.2.
6. può eventualmente collocare lungo la linea temporale "locale" dei segnatempo appartenenti alla linea temporale generale dell'*outline* (solo quelli la cui posizione ricade tra il punto d'inizio ed il punto di fine dell'attività in questione, e rispettando l'ordine cronologico generale dei segnatempo rispetto alla linea temporale generale)

2.5 INSERIMENTO DI ALTRI SEGNA TEMPO SULLA LINEA TEMPORALE

Oltre ai segnatempo d'inizio e di fine delle attività, è possibile inserire lungo la linea temporale altri generici segnatempo, che possono successivamente essere utilizzati all'interno delle linee temporali "locali" delle attività (vedi sotto)



al fine di specificare le eventuali sincronie, coordinamenti e sviluppi paralleli di “avvenimenti” tra diverse attività della *outline* (cfr. il rapporto tecnico TR3.1).

All’inserimento di un segnatempo generico, attraverso le modalità di interazione con l’interfaccia analoghi a quelli per l’inserimento dei punti d’inizio e di fine di un’attività (cfr. 2.4), l’utente deve:

1. Attribuire un nome al segnatempo
2. Attribuire un’eventuale descrizione testuale al segnatempo

2.6. INSERIMENTO DI UNA MAPPA DI STRUTTURA

Per descrivere i contenuti delle *attività*, ad esse possono essere attribuite le *mappe di struttura*. Le mappe di struttura sono grafi nodo-arco senza vincoli strutturali, e possono dunque essere composte da un numero arbitrario di nodi e di archi tra nodi.

Per ogni mappa di struttura è necessario:

1. specificare l’attività alla quale si riferiscono
2. specificare un titolo
3. facoltativamente specificare una descrizione testuale
4. facoltativamente attribuire un punto d’inizio ed un punto di fine sulla linea temporale “locale”

Successivamente è possibile inserire nodi ed archi.

A. Per ogni nodo inserito, l’utente deve:

1. Specificare un nome per il nodo:
 - 1.1. Al nodo può essere attribuito un nome nuovo, oppure può essere scelto tra i nomi dei nodi già inseriti in precedenza (in

questo modo è possibile riutilizzare una medesima “particella” (ad esempio un personaggio) in diverse mappe di struttura e di comportamento tra diverse attività dell’*outline*.

2. Indicare il tipo “semantico” del nodo, scelto tra i tipi previsti durante la fase di configurazione dei tipi di nodi (cfr. 2.3)
3. Attribuire il contenuto al nodo secondo le modalità descritte più sotto nel sottocapitolo 2.8

B. Per ogni arco che si vuole inserire, l’utente deve specificare:

1. Il nodo di origine ed il nodo di destinazione dell’arco (questa informazione può essere inserita sia attraverso l’interfaccia grafica congiungendo con il puntatore i due nodi, oppure attraverso l’interfaccia di proprietà dell’arco, scegliendo i due nodi di riferimento)
2. Indicare il tipo “semantico” dell’arco, scelto tra i tipi previsti durante la fase di configurazione dei tipi di archi (cfr. 2.3)
3. Attribuire il contenuto all’arco secondo le modalità descritte più sotto nel sottocapitolo 2.8.

2.7. INSERIMENTO DI UNA MAPPA DI COMPORTAMENTO

A differenza delle mappe di struttura, che dentro l’ambiente di *authoring* di più si avvicinano al comune concetto delle mappe concettuali, le mappe di comportamento sono un tipo di mappe vincolate dal punto di vista della loro struttura. Infatti, le mappe di comportamento prevedono che gli archi siano numerati (cfr. il rapporto tecnico TR3.1), a rappresentare la sequenza degli “avvenimenti” rappresentati da questi archi. Dunque, attribuendo ad ogni arco una posizione ordinale, l’utente può dichiarare l’ordine cronologico degli “accadimenti” che avvengono tra i nodi di una mappa di comportamento..

Per ogni mappa di comportamento l’utente:

1. deve specificare l’attività alla quale si riferiscono

2. deve attribuire un titolo testuale
3. facoltativamente può attribuire una descrizione testuale
4. deve inserire il punto d’inizio ed il punto di fine rispetto alla linea temporale “locale” dell’attività

Successivamente è possibile inserire *molteplici* archi tra i nodi.

Per ogni arco inserito, l’utente:

1. Indica il nodo di origine ed il nodo di destinazione dell’arco (questa informazione può essere inserita sia attraverso l’interfaccia grafica congiungendo con il puntatore i due nodi, oppure attraverso l’interfaccia di proprietà dell’arco, scegliendo i due nodi di riferimento).
2. Indica il tipo “semantico” dell’arco, scelto tra i tipi previsti durante la fase di configurazione dei tipi di archi (cfr. sottocapitolo 2.3).
3. Attribuisce il numero che indica l’ordine dello svolgimento temporale dell’azione o flusso rappresentato dall’arco, rispetto agli antri archi della mappa di comportamento in questione
4. Attribuisce il contenuto all’arco secondo le modalità descritte più sotto nel sottocapitolo 2.8.

2.8. INSERIMENTO DEI CONTENUTI DEI NODI E DEGLI ARCHI

Come sappiamo, le “particelle” dei due tipi di mappe (cioè nodi ed archi) possono essere intesi come dei contenitori che devono essere riempiti con contenuti (cfr. il rapporto tecnico TR3.1). *Questa è un’operazione fondamentale all’interno del complessivo processo di authoring dell’outline narrativa*, poiché in questo modo l’utente-autore può dotare di senso le particelle, specificandone attraverso tali contenuti il loro “ruolo” all’interno della struttura delle mappe.

I contenuti delle particelle sono composti dagli *contlet* digitali proveniente del *repository* di LC3, “incollati” insieme con delle annotazioni testuali che l’utente-autore può inserire. Quindi, il popolamento delle “particelle” con i contenuti non

è solo un'operazione di grande rilievo per l'*authoring* dell'*outline narrativa*, ma rappresenta anche uno dei principali canali di integrazione con le componenti dell'architettura complessiva di LC3 che sono oggetto di attività degli OR1 e OR2.

In termini concreti, per ogni nodo ed arco, l'utente può:

1. Usare il *motore di ricerca* (semantico o no) per trovare i *contlet* rilevanti, e successivamente associarli alla "particella"
2. Inserire un'annotazione testuale che specifichi i contenuti della "particella", collocando i *tag* di riferimento ai *contlet* all'interno di questa annotazione testuale (al fine di fornire un senso compiuto e di documentare in forma semanticamente compiuta la "particella" in questione).

3. LA STRUTTURA DI DATI ED IL LINGUAGGIO SOML

Come è stato illustrato nel documento "Metodologie di *storytelling* integrato" (cfr. rapporto tecnico TR3.1), la struttura formale che supporta la creazione delle *story outline* attraverso le modalità descritte sopra è il cosiddetto linguaggio di modellazione delle *story outline* (*Story-Outline Modelling Language*, d'ora in poi SOML).

In questo capitolo descriviamo la struttura della base di dati relazionale utilizzata come *back-end* dell'ambiente di *authoring* e la corrispondente struttura XML del SOML.

3.1. STRUTTURA DELLA BASE DI DATI BACK-END

A partire dagli elementi definatorî della struttura di un'*outline* narrativa, proponiamo una generalizzazione sotto forma di schema della base di dati (Figura 1.) che corrisponde alla struttura implementata nella base di dati relazionale di *back-end*.

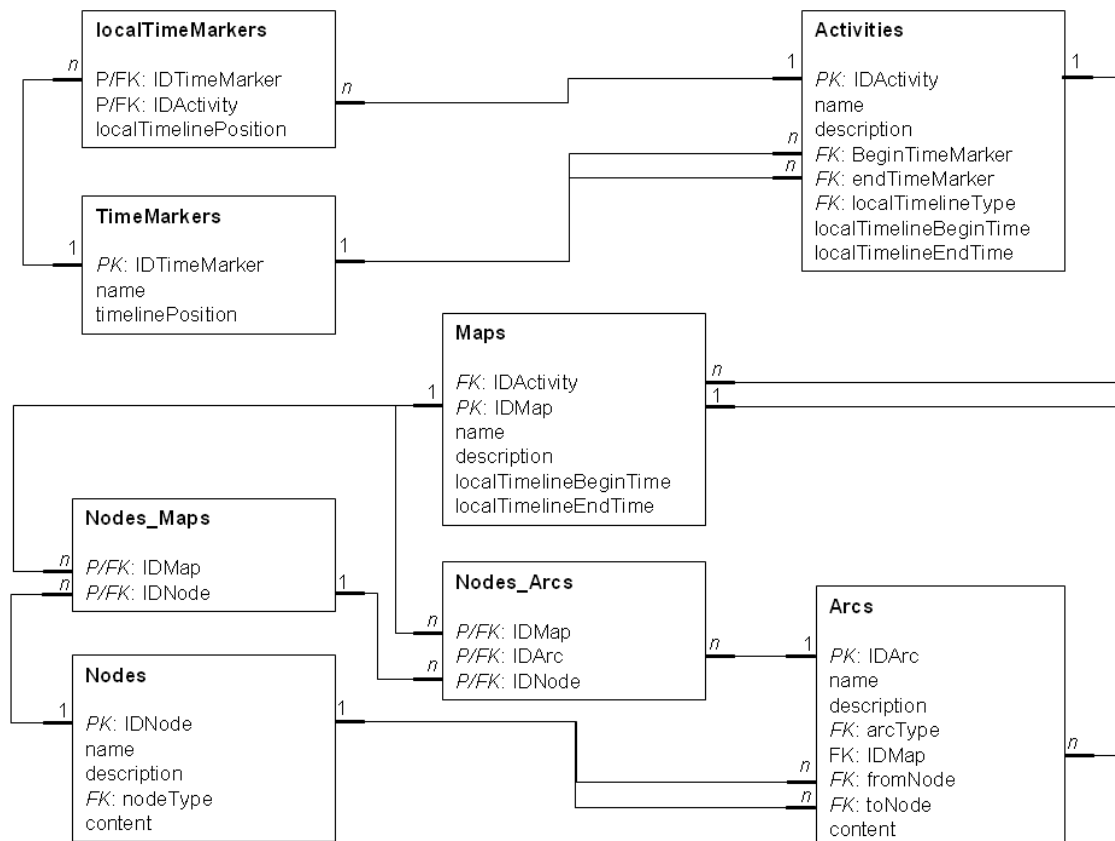


Figura 1. Schema della base di dati di un'outline narrativa

La tabella “Activities” contiene le attività dell'outline. Ogni attività è contraddistinta da un *time-marker* (segnatempo) di inizio e di fine (rispettivamente i campi “beginTimeMarker” e “endTimeMarker”). Attraverso il campo “localTimelineType” viene specificato il tipo (isocrono | non isocrono) della *timeline locale* usata all'interno dell'attività. I campi “localTimelineBeginTime” e “localTimelineEndTime” indicano il tempo (data e ora) rispettivamente di inizio e di fine della *timeline locale*.

La tabella “TimeMarkers” contiene i *time-marker* usati sulla *timeline* della storia. Ogni *time-marker* è contraddistinto dalla sua posizione (campo “*timelinePosition*”) espressa sulla scala continua nell’intervallo da 0 a 1. La posizione espressa in questo modo viene poi trasposta sulla *timeline* tenendo conto del tempo d’inizio e di fine della *timeline*.

La tabella “Maps” contiene tutte le mappe all’interno di un’*outline*. Esse sono riferite in particolare alle singole attività (campo “*IDActivity*”). Inoltre, ogni mappa ha una collocazione rispetto alla linea temporale locale (definita dai suoi punti d’inizio e di fine, registrati rispettivamente nei campi “*localTimelineBeginTime*” e “*localTimelineEndTime*”).

La tabella “Nodes” contiene i nodi delle *mappe di struttura e di comportamento* usate nella descrizione dei contenuti dell’attività. Ogni nodo è contraddistinto poi dal tipo (campo “*nodeType*”). Inoltre, ogni nodo è descritto da un contenuto (campo “*content*”). Quest’ultimo campo contiene il contenuto interno del nodo, ovvero i riferimenti ai *contlet* esterni, alle eventuali altre *outline* di storie, assieme al testo dell’annotazione. Tale contenuto viene inserito sotto forma di testo semplice, adoperando gli appositi *tag* <content> e <outline> per l’inserimento rispettivamente dei *contlet* e delle *outline* di altre storie che gli autori desiderano includere nel contenuto.

L’appartenenza dei nodi alle mappe è registrata nella tabella “Nodes_Maps”. In essa vengono registrate le associazioni tra le singole mappe (campo “*IDMap*”) ed i nodi (campo “*IDNode*”).

La tabella “Arcs” contiene gli archi tra i nodi delle mappe di struttura e di comportamento. Il tipo di arco è registrato nel campo “*arcType*”.

Ogni arco è definito da un nodo d’origine e da un nodo di destinazione (rispettivamente i campi “*fromNode*” e “*toNode*”), oltre alla mappa di appartenenza nella quale l’arco in questione è definito (campo “*IDMap*”).

Inoltre, poiché i nodi possono essere direttamente associati agli archi, nella tabella “Nodes_Arcs” vengono registrate questo tipo di associazioni.

La tabella “localTimeMarkers” registra l’uso dei *time-marker* sulla *timeline locale* edll’attività. Infatti, la tabella contiene il riferimento all’attività (chiave esterna “*IDActivity*” dentro cui mappa di comportamento è collocato in *time-marker*), e la

posizione (campo “localTimelinePosition”) del *time-marker* rispetto alla *timeline locale*.

3.2. LA STRUTTURA DEL SOML

Il file di salvataggio di un’*outline* narrativa avviene in *Story-Outline Modelling Language* nel formato XML. Quindi, il formato XML è la modalità principale di archiviazione ed interscambio di *outline* narrative.

La struttura del SOML in XML è basata sulla struttura dello schema della base di dati descritto nel precedente sottocapitolo 3.1.

Lo schema di base del SOML è rappresentato in Figura 2, mentre nelle successive figure da 3 a 9 sono rappresentati i dettagli dei sotto-schemi per gli elementi delle sequenze dell’elemento di base “Outline”.

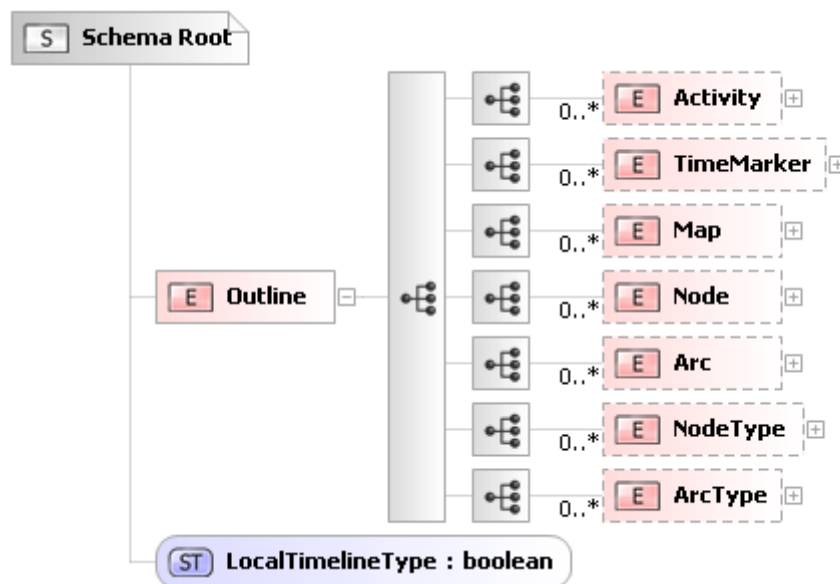


Figura 2. Schema XML di base di SOML

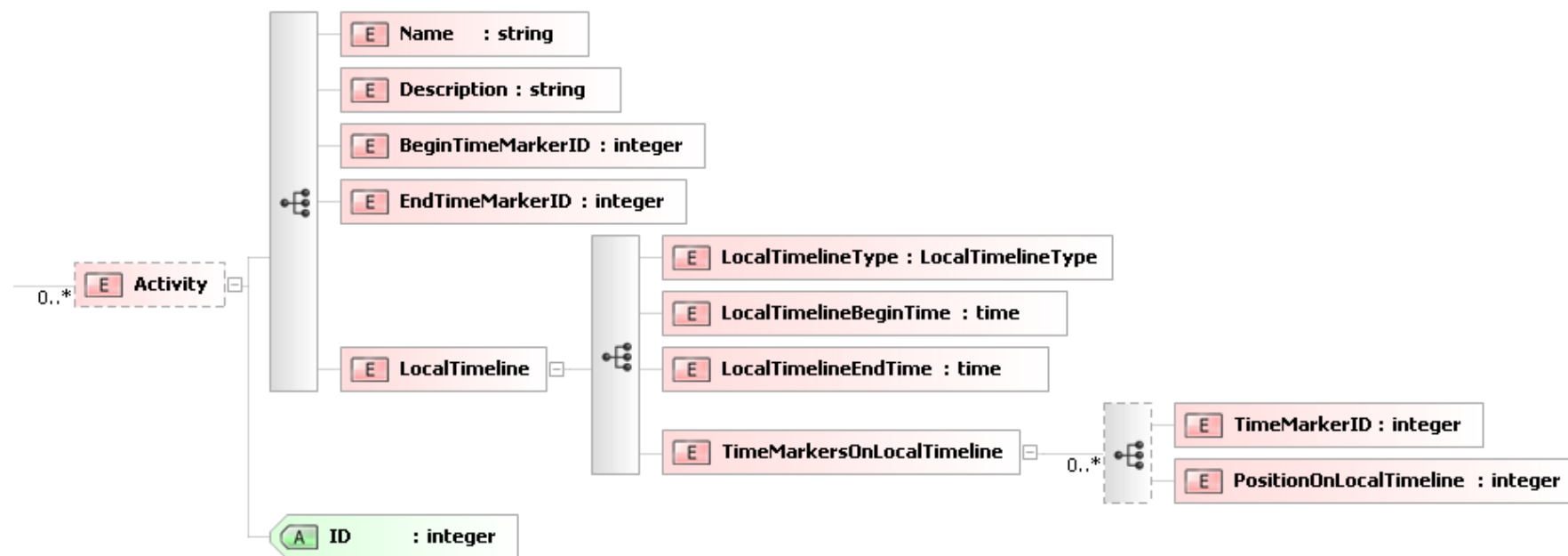


Figura 3. Sotto-schema XML dell'elemento "Activity"

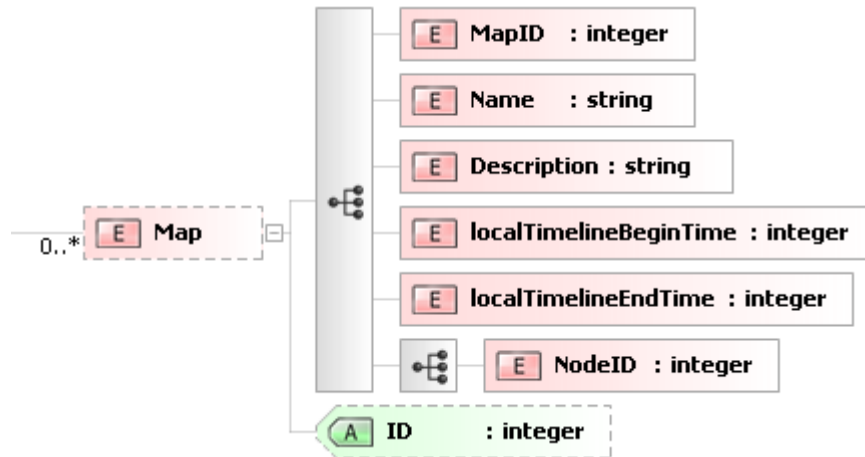


Figura 4. Sotto-schema XML dell'elemento "Map"

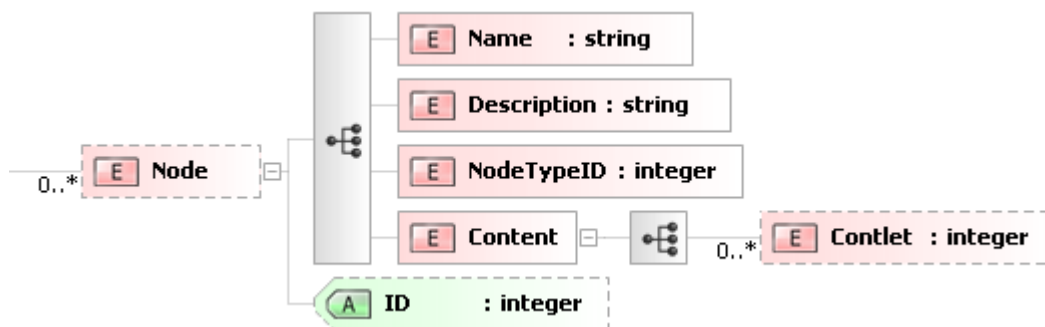


Figura 5. Sotto-schema XML dell'elemento "Node"

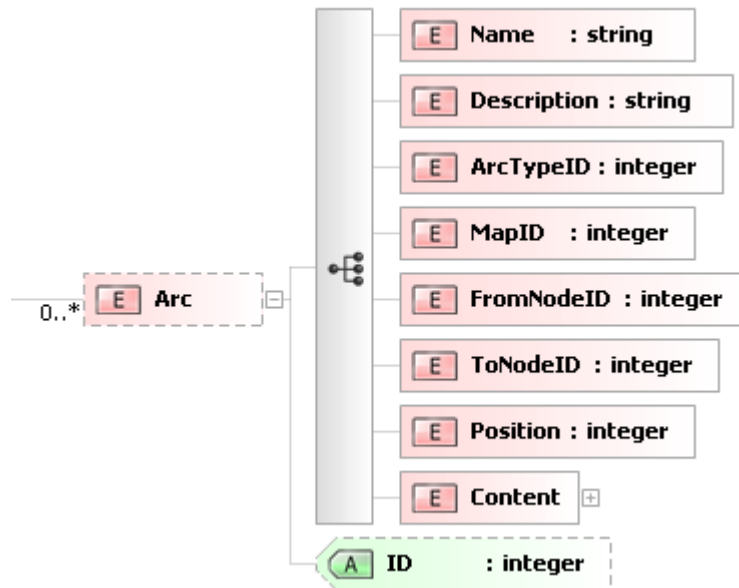


Figura 6. Sotto-schema XML dell'elemento "Arc"

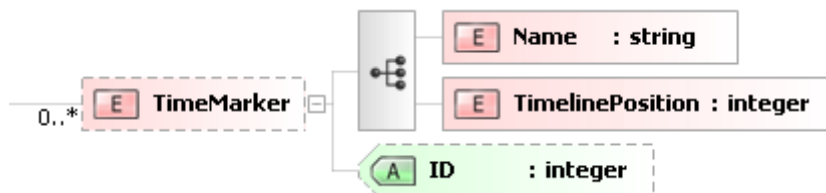


Figura 7. Sotto-schema XML dell'elemento "TimeMarker"

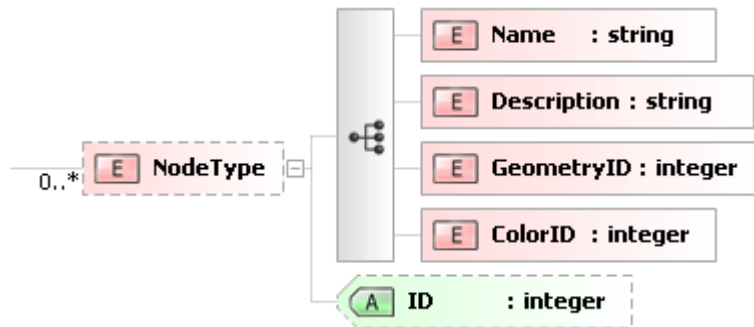


Figura 8. Sotto-schema XML dell'elemento "NodeType"

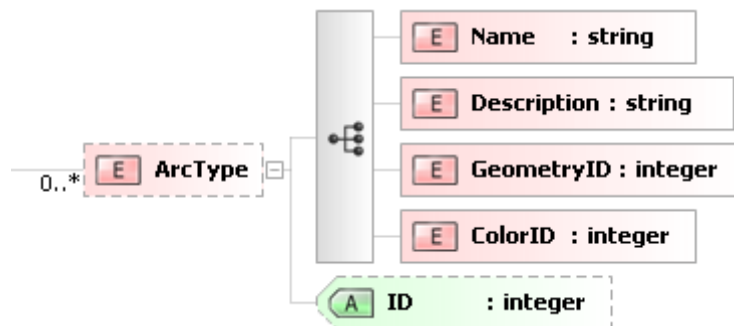


Figura 9. Sotto-schema XML dell'elemento ArcType"

4. PROGETTAZIONE E SVILUPPO DELLE INTERFACCE UTENTE

La seguente Figura 9. rappresenta lo stato attuale dell'interfaccia sviluppata. Nella parte bassa si trova il controllo per la disposizione delle attività lungo la *timeline*. Nella parte centrale è rappresentata la *mappa di struttura* relativa all'attività correntemente selezionata.

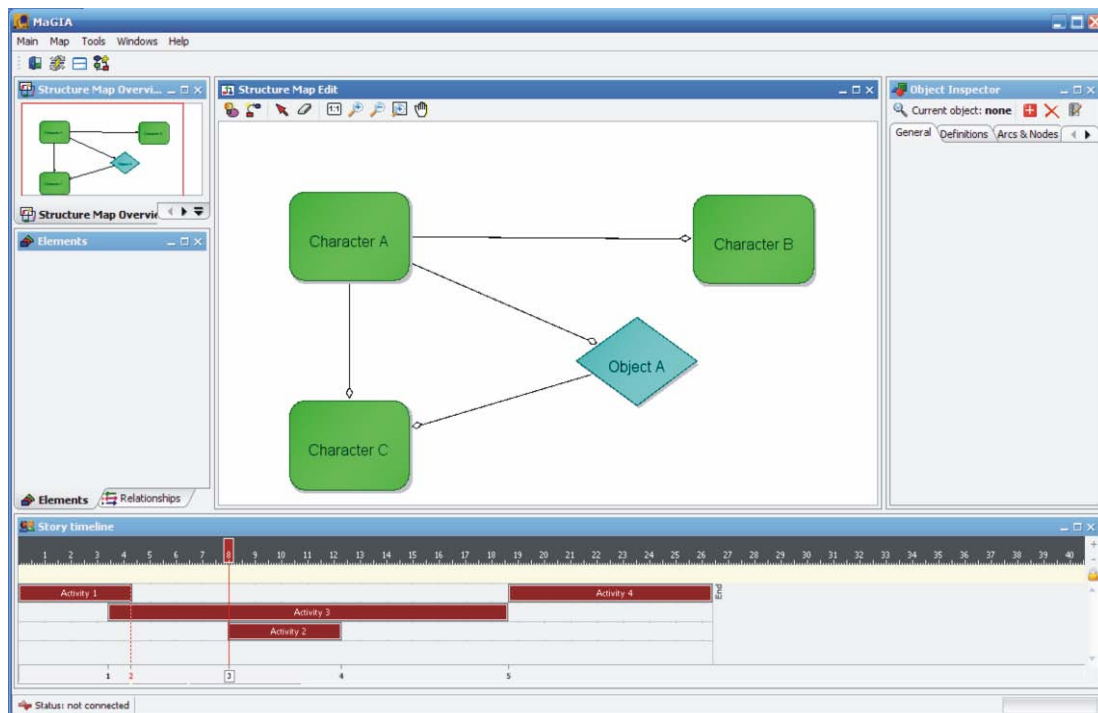


Figura 9. L'interfaccia generale dell'ambiente di *authoring*

Il controllo per la creazione e rappresentazione delle attività (cfr. Figura 10.) consente di definire una *timeline* mediante la specificazione del tempo iniziale, tempo finale e tipo (isocrona | non isocrona). Successivamente è possibile creare, inserire lungo la *timeline* e modificare le attività che compongono la storia.

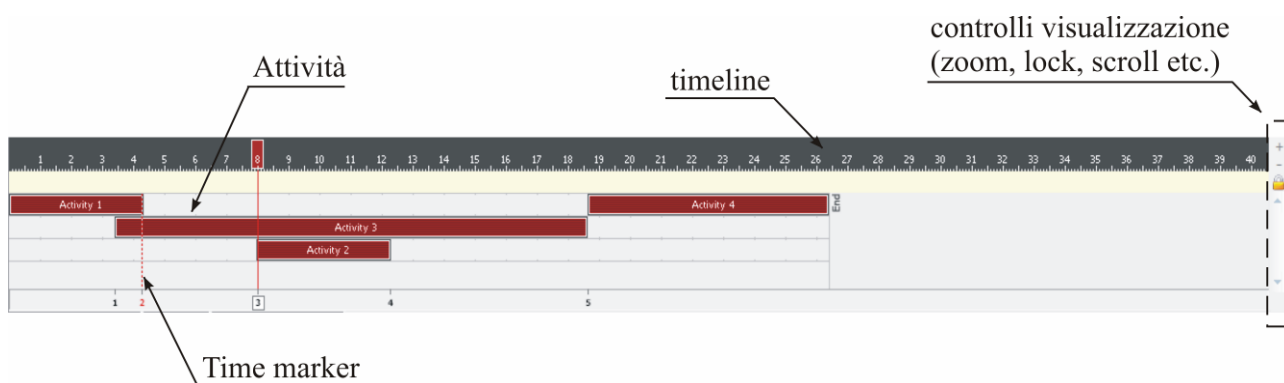


Figura 10. Controllo di gestione della *timeline*

Ogni *attività* della storia è rappresentata da un rettangolo ridimensionabile tramite i due *time-marker* iniziale e finale. Questi ultimi sono rappresentati da linee verticali e dal loro *id* riportato in basso. I rettangoli rappresentanti le attività possono essere disposti su più righe per consentire le sovrapposizioni temporali tra attività.

Il controllo di gestione delle *timeline* consente di spostare facilmente le attività della storia per comporre un *layout* compatto e leggibile. Appositi pulsanti (a sinistra nella figura precedente) consentono di modificare la visualizzazione eseguendo operazioni di ingrandimento (*zoom*), scorrimento (*scroll*) e di blocco (*lock*) del *layout* al fine di impedire modifiche accidentali una volta terminato il lavoro.

Selezionando mediante il *mouse* un'attività lungo la *timeline* è inoltre possibile visualizzare nell'apposita finestra la mappa di struttura o quella di comportamento. L'*editing* delle due mappe avviene secondo le funzionalità offerte dal componente JGraph che sono accessibili mediante pulsanti o voci di menu (vedi la seguente Figura 11.).

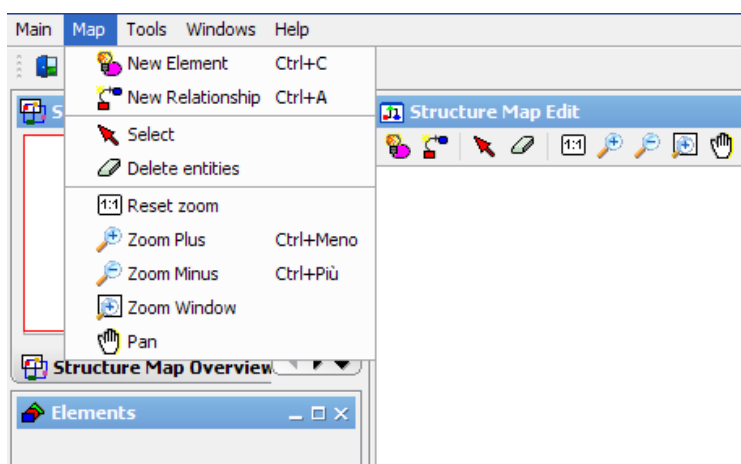


Figura 11. Strumenti di *editing* di mappe di struttura e di comportamento

Nella fase di inizializzazione di una storia è necessario definire i tipi di nodi e di archi che verranno impiegati successivamente durante la creazione delle mappe della *outline* di storia. Ciò avviene selezionando una forma geometrica ed un colore per nodi ed archi tra quelli predefiniti offerti dal programma mediante l'apposita finestra di dialogo (vedi Figura 12.)

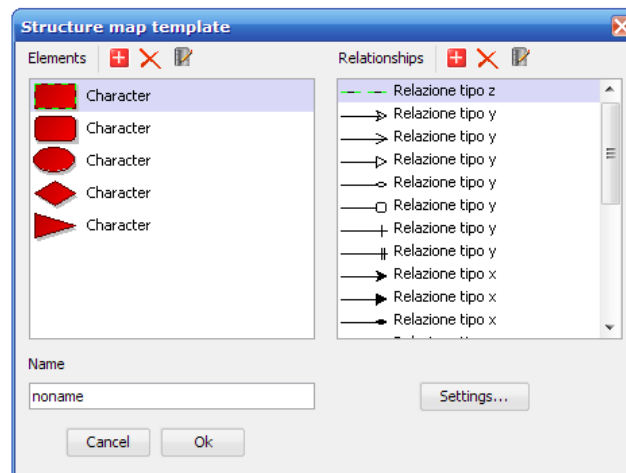


Figura 12. Inizializzazione e definizione dei tipi di “particelle” delle mappe

Successivamente, all'immissione di un nodo o di un arco in una delle mappe relative all'attività selezionata lungo la *timeline*, è possibile selezionarne il tipo (vedi la seguente Figura 13.) tra quelli definiti nella fase di inizializzazione.

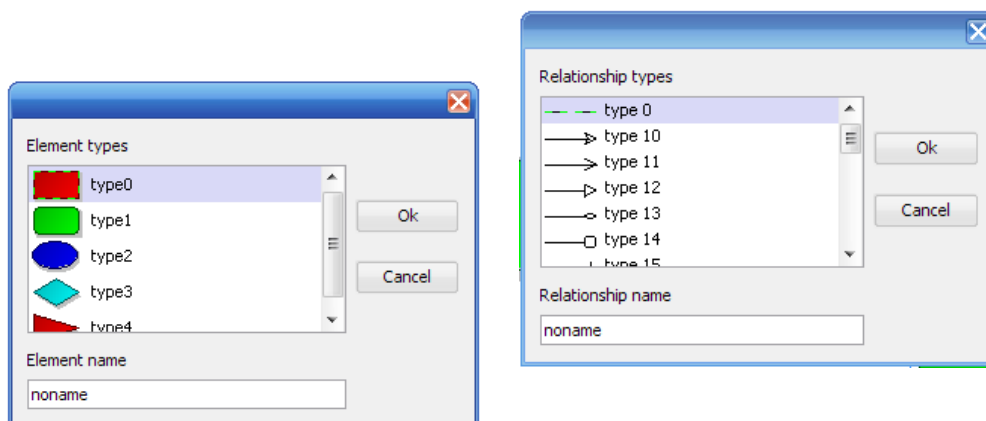


Figura 13. All'inserimento di un nuovo nodo o arco all'interno delle mappe, è necessario specificare il tipo.

L'interfaccia in corso di sviluppo offre diverse possibilità di navigazione delle mappe di struttura e di comportamento, particolarmente utile in caso di mappe composte da numerosi elementi e relazioni. In particolare (vedi sotto Figura 14.) è presente sia un navigatore visuale, sul quale è sempre rappresentata l'intera mappa e che consente di selezionare la finestra da visualizzare, sia un elenco testuale degli elementi (nodi) e delle relazioni (archi). Selezionando un elemento dall'elenco è possibile ottenere uno *zoom* centrato su di esso.

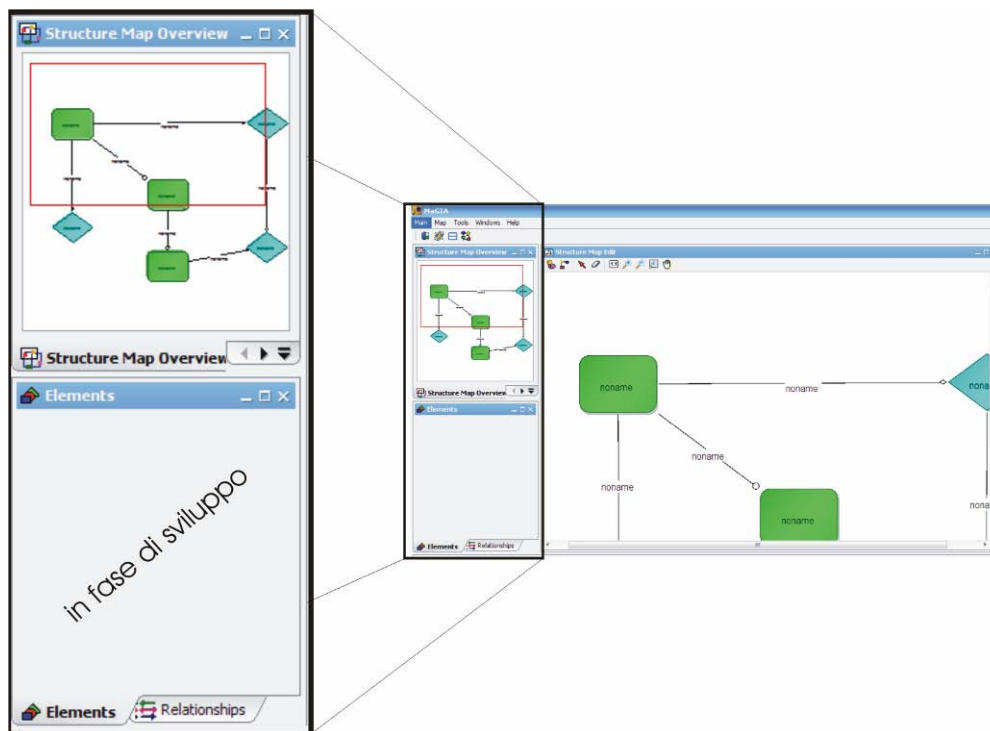


Figura 14. Strumenti di navigazione ed esplorazione delle mappe



Un ulteriore importante controllo dell'ambiente di *authoring* è il cosiddetto *Object Inspector* (posto nella parte destra dello schermo ed ancora in fase di sviluppo) che offre le seguenti funzionalità:

- visualizzazione delle caratteristiche generali delle entità che compongono una mappa (creatore, denominazione, tipo, etc.);
- navigazione testuale sincronizzata con la visualizzazione della mappa. Per ogni nodo sono elencate le relazioni che lo collegano ad altri nodi. Selezionando una relazione si accede al nodo collegato;
- accesso ai contenuti e *contlet* associati alle "particelle" (nodi ed archi) delle mappe di struttura e di comportamento.

5. ARCHITETTURA SOFTWARE DI IMPLEMENTAZIONE

L'interfaccia utente è basata sul framework *InfoNode Docking Windows* (<http://www.infonode.net/>) basato su Java Swing e disponibile sotto licenza GPL. Il pacchetto consente di produrre interfacce dotate di una notevole efficacia e flessibilità. In particolare, rispetto ad una normale GUI Swing l'utente può facilmente personalizzare il layout dell'interfaccia ed è semplice rendere disponibile un meccanismo di salvataggio dei layout.

Per la modellazione ed il disegno delle mappe di comportamento e di struttura, è stato impiegato il componente JGraph (www.jgraph.com), che fornisce efficaci funzionalità di disegno e di layout automatico dei grafi, nonché alcune funzioni di base di analisi, il tutto sotto forma di API Java. Il modello di grafo offerto da JGraph consente di associare a nodi ed archi degli oggetti definiti dall'utente. Ciò risulta necessario in quanto nodi ed archi possono contenere (i) descrizioni testuali o definizioni fornite dagli utenti, (ii) documentazione e collegamenti a contenuti multimediali, (iii) riferimenti a documentazione.

Dovrà anche essere valutata la possibilità di impiego di JUNG (Java Universal Network/Graph Framework - <http://jung.sourceforge.net/>), quale alternativa alla versione open source di JGraph, per la sua maggiore ricchezza di funzioni potenzialmente utili all'applicazione in corso di sviluppo.



La timeline globale e quelle locali sono basate sul package Java Jaret Timebars (<http://www.jaret.de/timebars>) che comprende un view per un modello di dati basato su intervalli temporali e time-markers implementato sia per Swing che per SWT. La visualizzazione, che supporta l'editing visuale (trascinamento e ridimensionamento di intervalli e time-markers) è stata particolarizzata per le diverse timeline richieste dall'applicazione.

La visualizzazione di contenuti digitali associati a nodi ed archi delle mappe è basata su NativeSwing (<http://djproject.sourceforge.net/ns/>), una libreria che consente una integrazione di componenti nativi in applicazioni Java Swing. NativeSwing consente di rendere "embedded" nell'applicazione un Web Browser, un Flash player ed un Multimedia Player. Attualmente il multimedia player è disponibile solo in ambiente Microsoft Windows, essendo basato su Window Media Player. Il componente che implementa il web browser si avvale di Internet Explorer o Mozilla in Windows, e di Mozilla in altre piattaforme. Il componente Flash player per si avvale del plugin Flash.

Il parsing del documento SOML avviene mediante il parser SAX del package standard javax.xml.parsers.

6. RIFERIMENTI BIBLIOGRAFICI

[1] Blečić, I., Cecchini, A., Trunfio, G.A.: A Collaborative and Collective Concept Mapping Tool. In: Yuhua Luo (Ed.) CDVE 2007, LCNS vol. 4674, pp. 264--271, Springer Berlin/Heidelberg (2007)



```
<xs:element minOccurs="0" maxOccurs="unbounded" name="TimeMarker">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="xs:string" />
      <xs:element name="TimelinePosition" type="xs:integer" />
    </xs:sequence>
    <xs:attribute name="ID" type="xs:integer" />
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="Map">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MapID" type="xs:integer" />
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="localTimelineBeginTime" type="xs:integer" />
        <xs:element name="localTimelineEndTime" type="xs:integer" />
        <xs:sequence>
          <xs:element name="NodeID" type="xs:integer" />
        </xs:sequence>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="Node">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="NodeTypeID" type="xs:integer" />
        <xs:element name="Content">
          <xs:complexType mixed="true">
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded"
name="Contlet" type="xs:integer" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="Arc">
    <xs:complexType>
      <xs:sequence>
```



```
<xs:element name="Name" type="xs:string" />
<xs:element name="Description" type="xs:string" />
<xs:element name="ArcTypeID" type="xs:integer" />
<xs:element name="MapID" type="xs:integer" />
<xs:element name="FromNodeID" type="xs:integer" />
<xs:element name="ToNodeID" type="xs:integer" />
<xs:element name="FromLocalTimelinePosition" type="xs:integer"/>
<xs:element name="ToLocalTimelinePosition" type="xs:integer" />
<xs:element name="Content">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="Contlet" type="xs:integer" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="ID" type="xs:integer" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="NodeType">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="GeometryID" type="xs:integer" />
        <xs:element name="ColorID" type="xs:integer" />
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="ArcType">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="GeometryID" type="xs:integer" />
        <xs:element name="ColorID" type="xs:integer" />
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="LocalTimelineType">
```



PARTNER: UNIVERSITÀ DEGLI STUDI DI SASSARI

RESPONSABILE: ARNALDO CECCHINI

Technical Report: **TR3.2.1**

*LC3 – Laboratorio pubblico-privato di ricerca
sul tema della Comunicazione delle Conoscenze Culturali*

PAG 28 DI 28

```
<xs:restriction base="xs:boolean" />  
</xs:simpleType>  
</xs:schema>
```

PROGETTO LC3

--	--	--	--	--