Rappresentazione di Informazione Musicale Simbolica mediante Linguaggi Markup

Maurizio Longari
LIM-DSI
Università degli Studi di Milano
via Comelico, 39
20135 Milano, Italia
+39 02 5835 6296
longari@dsi.unimi.it

Indice

- ✓ Introduzione a SGML e SMDL
- ✓ Introduzione a XML
- ✓ Linguaggi definiti in XML per Informazione Musicale Simbolica

Cos’é un markup

- Storicamente la parola *markup* (etichetta) e' stata utilizzata per descrivere dei commenti o altre indicazioni all'interno di un testo atti ad istruire un compositore o un dattilografo su come deve essere graficamente distribuito un particolare passaggio.

- Con l'automatizzazione della formattazione e della stampa di testi, il termine é stato esteso a tutti i tipi di simboli di formattazione, stampa e di elaborazione del testo elettronico.

Cos'é un Linguaggio Markup

- Con Markup Language si intende un inseme di markup convenzionali utilizzati per la codifica di testi elettronici
- Un linguaggio markup deve specificare:
  - Quali markup sono consentiti
  - Quali markup sono richiesti
  - Come i markup sono distinti dal testo

    e

  - Che cosa significa un certo markup

  SGML fornisce gli strumenti per rappresentare i primi tre punti

SGML

- Standard Generalized Markup Language
- SGML é uno standard internazionale per la descrizione di testi elettronici di tipo mark-up. Precisamente, SGML é un metalinguaggio, ovvero un mezzo per la descrizione formale di linguaggi, in questo caso, linguaggi markup.

SGML

- Caratteristiche
  - Markup descrittivo
  - Tipo di documento
  - Indipendenza dei dati

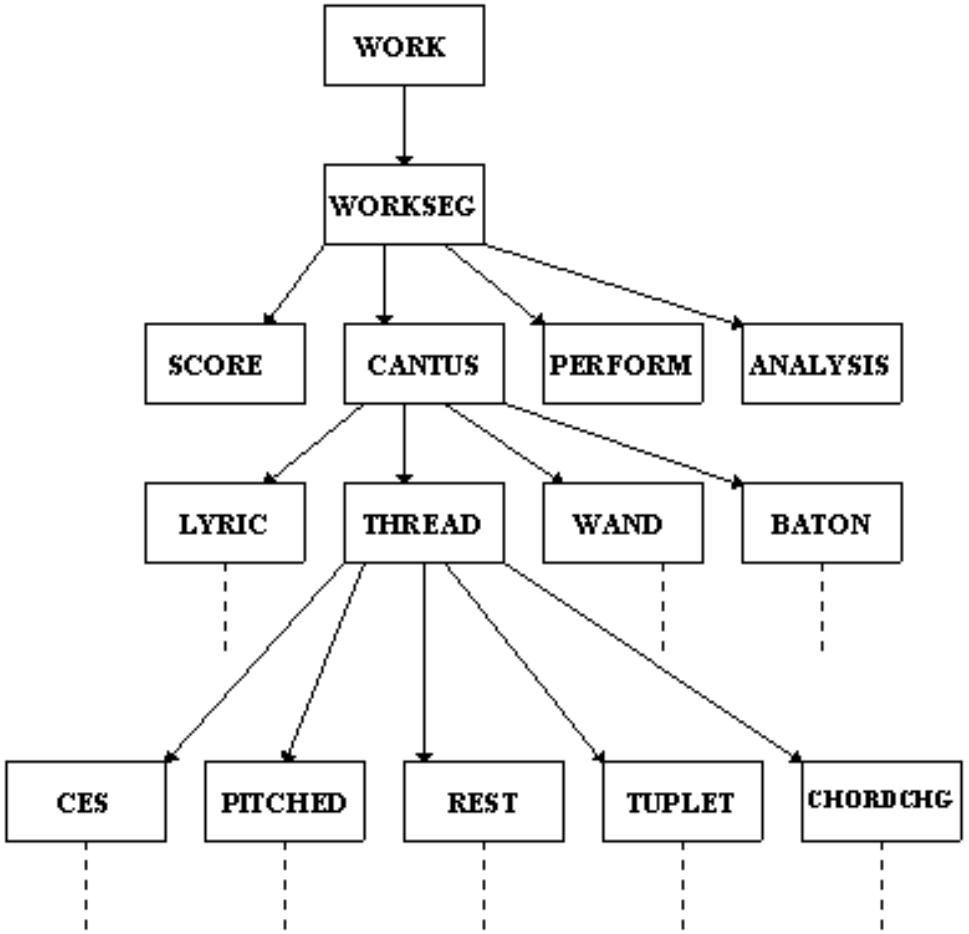
- Una introduzione:

http://www.uic.edu/orgs/tei/sgml/teip3sg/index.html

Standard Music Description Language

- Applicazione di HyTime e SGML
- Informazione musicale simbolica divisa in 4 domini:
  - Logical
  - Visual
  - Gestural
  - Analytical
- Strutturazione dell'Informazione Musicale su di un Finite Coordinate Space (FCS)
- Standard draft ISO 10743

SMDL - Struttura



SMDL - Esempio

```
<work>
<bibdata> -- Dati bibliografici --
  <title> Ottone
  <author> G.F. Handel
  <descript> Opera
  <issuer> Hicks/Chry
</bibdata> -- Fine dati bibliografici--
<workfcs> -- Inizio fcs del brano --
  <workschd>
    <workseg>
      <bibdata>
        <title> D'innalzar i flutti
        <numclass> Aria
        <role> Adelberto
      </bibdata>
    <pitchgam id=pitchgm0 -- inizio della tonalita' -- ... >
      <genfreq> -- setta gamstep 6 (='a') come 440 Hz --
        <gamstep>6</gamstep>
        <freqspec><hertz>440</hertz></freqspec>
      </genfreq>
        <namestep>
          <pitchdef>
            <pitchnm>eb</pitchnm>
            <gamstep>0</gamstep>
          </pitchdef>
        </namestep>
    ...
```

SMDL - Esempio

```
...
    </pitchgam>
    ...
     <mvt65a>
        <thread id=thd1  nominst="Violino I, II">
        </thread>
        ...
        <baton id=bat1>
        </baton>
        <start Violino I, II>
            <ce>t 1 eb
            <ce>3t4 0 bb
            <ce>t4 1 g
            <ce>3t4 1 ab
            <ce>t4 1 f
            <ce>t2 rest
          <ce></end Violini I, II>
       </mvt65a>
-- Fine sezione dati del brano  Movimento 65a --
      </workseg>
    </workschd>
  </workfcs>
</work> -- Fine della codifica del brano --
```

SMDL - Link

- SMDL

www.oasis-open.org/cover/smdlover.html

www.student.brad.ac.uk/srmounce/smdl3.html

- HyTime

www.hytime.org/

Benefici di XML

- ✓ Interscambiabile su Internet
- ✓ Struttura gerarchica
- ✓ Intelligiblilitá
- ✓ Estensiblilitá
- ✓ Disponibilitá di tools per l'implementazione del formato

XML background

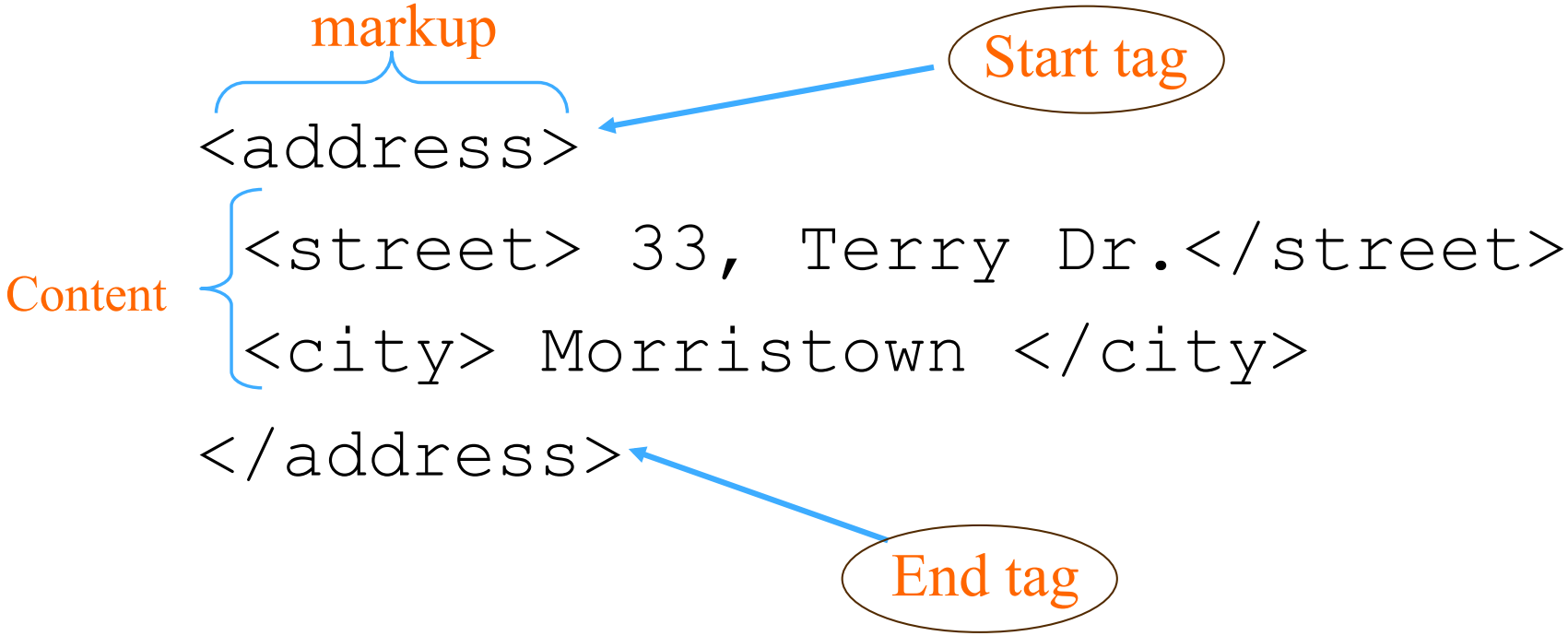
- ✓ Sottinsieme di SGML
- ✓ Semplifica SGML:
  - trascurando molte opzioni sintattiche e varianti
  - trascurando alcune caratteristiche del DTD
  - trascurando alcune caratteristiche problematiche
- ✓ Standard approvato dal W3C

Elementi

- Mattoni di XML
- Dare un significato ad una parte di documento
- Avere un tipo di elemento (‘example’, ‘name’) rappresentato da un *markup (tag).*
- Possibilitá di annidamento a qualsiasi livello

Elementi

- Un semplice elemento completo:



Elementi

- Può contenere:
  - Altri elementi (*sub-elements*)

```
<address>
      <street> 33, Terry Dr.</street>
      <city> Morristown </city>
</address>
```

  - testo (*data content*)

```
<street> 33, Terry Dr.</street>
```

  - una loro combinazione (*mixed content*)

```
<par>Today, <date>05-06-2000</date> Mr. <name>Bill
Gates<name> is in California to talk to ... </par>
```

Elemento Document

- E’ l’elemento piú esterno contenente tutti gli altri elementi del documento

  esempio:

```
<employee>
  …
</employee>
```

- Deve esistere sempre

Elementi Vuoti (Empty)

- Elementi senza contenuto
  - Non hanno un tag di fine
  - Rappresentazione particolare dello start tag

esempio:

```
<medical-dossier …/>
```

Attributi

- Utilizzato per aggiungere informazione extra ad un elemento
- Sono sempre associati allo start tag:

```
<el-name attr-name1="v1" .. attr-name1="v1" >
……
<el-name/>
```

- Un elemento puó avere un numero qualsiasi di attributi distinti

An XML document

```
<?XML version="1.0">
<books>
   <book>
      <entry isbn="1-55860-622-X">
         <title>Data on the Web:...</title>
         <publisher>Morgan Kaufmann</publisher>
      </entry>
      <author> Serge Abiteboul</author>
      <bookRef to="0-201-53771-O 1-55860-463-4"/>
      <articleLink href="http://…/articles.xml#id(Abi97)">
   </book>
   <book>
      <entry isbn="0-201-53771-O">
         <title>Foundation of Databases</title>
         <publisher>Addison Wesley</publisher>
      </entry>
      <author> Serge Abiteboul</author>
      ...
   </book>
      ...
</books>
```

Elementi Vs Attributi

Che cosa utilizzo per memorizzare una certa informazione?

✓Un element, quando:

- Necessito di una veloce ricerca
- Deve essere visibile a tutti
- E' importatnte per il significato del documento
- E' debolmente tipato

✓Un attribute, quando:

- E' una scelta
- E' visibile solo per il sistema
- Non e' importante per il significato del documento
- E' fortemente tipato

Inoltre...

- ✓ Processing instructions, utilizzate principalmente per propositi di estensibilità (`<?target data?>`)
- ✓ Commenti (`<!-- … -->`)
- ✓ Riferimenti a caratteri (`&#163;`)
- ✓ Entitià:
  - Files esterni o parti del documento
  - Possono essere riferite ricorsivamente o da parti diverse nel documento

Tipi di Documento

- Idea base: associare un tipo al documento (analogia: calssi ed oggetti)
- Un tipo di documento rappresenta una classe di documenti con una struttura ed una semantica simile
- Esempi: slide presentations, articoli di giornale, agenda di un meeting, chiamate di metodi, etc.

DTD

- DTD fornisce un significato starndard per descrivere dichiarativamente la struttura di un tipo di documento
- Ció significa descrivere:
  - Quali (sub-)elementi puó contenere un elemento
  - Se puó contenere un del testo o no
  - Quali attributi contiene
  - Tipizzazione e defaultizzazione degli attributi

DTD

- Un DTD é logicamente composto da 2 parti:
  - Element Type Definition
  - Attribute List Declaration

Element Type Definition

- Element type definition specifica:
  - Struttura del documento
  - Contenuti consentiti (content model)
  - Attributi consentiti (dal significato delle dichiarazioni delle liste di attributi)

Element Type Definition

- Alcune possibili dichiarazioni :

- <!ELEMENT A (B*, C, D?)>
- <!ELEMENT A (B | C+)>
- <!ELEMENT A (#PCDATA)>
- <!ELEMENT A EMPTY>
- <!ELEMENT A (#PCDATA| B | C)*>

Attribute-List Declarations

- E’ la lista degli attributi permessi per ogni elemento.

  Ogni attributo e’ specificato da: name, type, e altre informazioni.
- Tipi di attributi. Tre gruppi:
  - **string types** (*CDATA*)
  - **tokenized types** (*ID,IDREF,IDREFS,...*)
  - **enumerated types** (as the ones in Pascal)

Attribute-List Declarations

- <!ELEMENT A (#PCDATA)>
  - <!ATTLIST A a CDATA #IMPLIED>
  - <!ATTLIST A a CDATA #IMPLIED
    b CDATA #REQUIRED>
  - <!ATTLIST A a CDATA #IMPLIED “aaa”>
  - <!ATTLIST A a CDATA #REQUIRED “aaa”>
  - <!ATTLIST A a CDATA #FIXED “aaa”>
  - <!ATTLIST A a (aaa|bbb) #IMPLIED “aaa”>
  - <!ATTLIST A id ID #REQUIRED>
  - <!ATTLIST A ref IDREF #IMPLIED>

DTD di un semplice libro

```
<!DOCTYPE Books[
<!ELEMENT Books(book)+>
<!ELEMENT book(entry, author+, bookRef, articleLink*)>
<!ELEMENT entry(title, publisher)>
<!ELEMENT bookRef EMPTY>
<!ELEMENT articleLink EMPTY>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT pubblisher (#PCDATA)>

<!ATTLIST entry isdn ID #REQUIRED>
<!ATTLIST bookRef  to IDREFS #IMPLIED>
<!ATTLIST articleLink
            xmlns:xlink CDATA #FIXED “http://w3c.org/xlink”
            xlink:type CDATA #FIXED “simple”
            xlink:href CDATA #REQUIRED>
]>
```

Well-formedness & Validity

- Un documento é detto well-formed (ben-formato) se segue le regle grammaticali fornite dal W3C.
- Un documento é detto valid (valido) se é conforme ad un DTD che ne specifica la struttura.

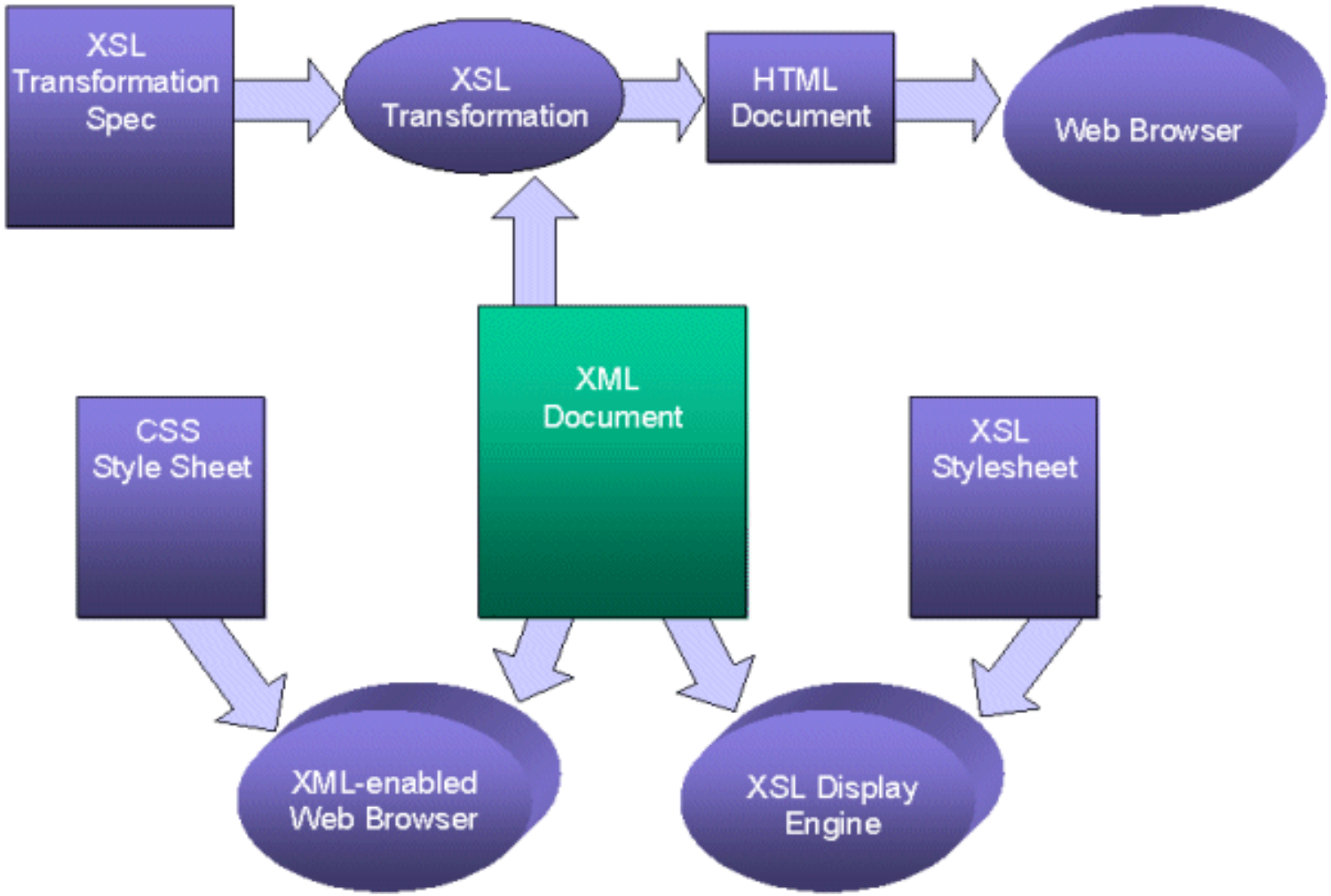
XML Schema

- W3C Reccomendation Marzo 2001
- Struttura di un documento XML specificata in XML
- Basato sulla definizione di tipi
- Maggiore controllo sulla validitá
- Il concetto di namespace é fondamentale

XSL Exstensible Stylesheet Language

- ✓ XSLT
  - Definizioni e specifiche XML per la trasformazione di documenti XML.
- ✓ XPath
  - Sintassi per formulare ricerche e selezioni di elementi ed attributi all'interno di documenti XML
- ✓ Formatting Objects
  - vocabolario di elementi definiti per diversi tipi di media (schermo, stampa, voce, etc...)

XSL


XSL
Transformation
Spec
XSL
Transformation
HTML
Document
Web Browser
XML
Document
CSS
Style Sheet
XSL
Stylesheet
XML-enabled
Web Browser
XSL Display
Engine

Oveview dei Linguaggi definiti in XML per la rappresentazione di Informazione Musicale Simbolica

XML & rappresentazione dell'I.M. Simbolica

- MusicXML
- MusiXML
- MusiCAT/MDL
- MPEG7 - Audio
- **MX**
- Altre definizioni possono essere trovate:

  www.oasis-open.org/cover/xmlMusic.html

MusicXML

- Sviluppato da M.Good
- Basato sui formati Humdrum e MuseData
- Due punti di vista della partitura: Time-wise e Part-wise
- Trasformazione XSLT fra i due punti di vista
- Letto e scritto da Finale e SharpEye

MusicXML - Esempio

```
<note>
  <pitch>
    <step>G</step>
    <octave>4</octave>
  </pitch>
  <duration>2</duration>
  <type>eighth</type>
  <stem>up</stem>
  <notations>
    <dynamics>
      <p/>
    </dynamics>
  </notations>
  <lyric>
    <syllabic>single</syllabic>
    <text>W&auml;rst</text>
  </lyric>
</note>
```

MusiXML

- ✓ Sviluppato da G.Castan
- ✓ Strutturato in tre sezioni:
  - Bibliographic
  - Logical
  - Filter
- ✓ Separazione della forma dal contenuto
- ✓ Formato XML Schema

MusiXML - Esempio

```
<chord>
        <note name="d" oct="2" dur=":8" beam="b1.2"/>
</chord>
```

MusiCAT & MDL

- ✓ Sviluppato da P. Roland
- ✓ MusiCAT formato ricco ma solo informazione di catalogazione
- ✓ MDL
  - Molto ricco di elemnti ed attributi
  - Organizzazione della partitura di tipo temporale
  - Attributi suddivisi in domini logical, visual, gestural, analytical e user defined

MPEG7 - Audio

- ✓ Sintassi definita solo in XML Schema
- ✓ MelodyContour Description Scheme
  - Contour (list –2 –1 0 1 2)
  - Beat (list)
  - Meter
- ✓ Utilizzato per propositi di Music Information Retrieval

Other languages

- ChordML
- MusicML
- FlowML
- 4ML
- Music Markup Maguage (MML)
- MNML Musical Notational Markup Language
- JScoreML

SMDL vs. XML

	Symbolic notation	Catalogue info.	Extra info.	Software tools
SMDL	high	mid	high	poor
MusiXML	mid	mid	*	high
MusicXML	high	mid	*	high
MusiCat	poor	high	mid	high
MNML	mid	poor	*	high
MML	mid	poor	*	high
MusicML	mid	*	*	high
ChordML	poor	*	*	high
MDL	high	poor	mid	high

* not envisaged

MX

- Obiettivo: permettere di codificare in un unico file tutta l'informazione necessaria per trattare un brano musicale sotto tutti i suoi aspetti:
  - Grafico/Notazionale simbolico
  - Performace (MIDI, AUDIO, VIDEO)
  - Informazioni correlate

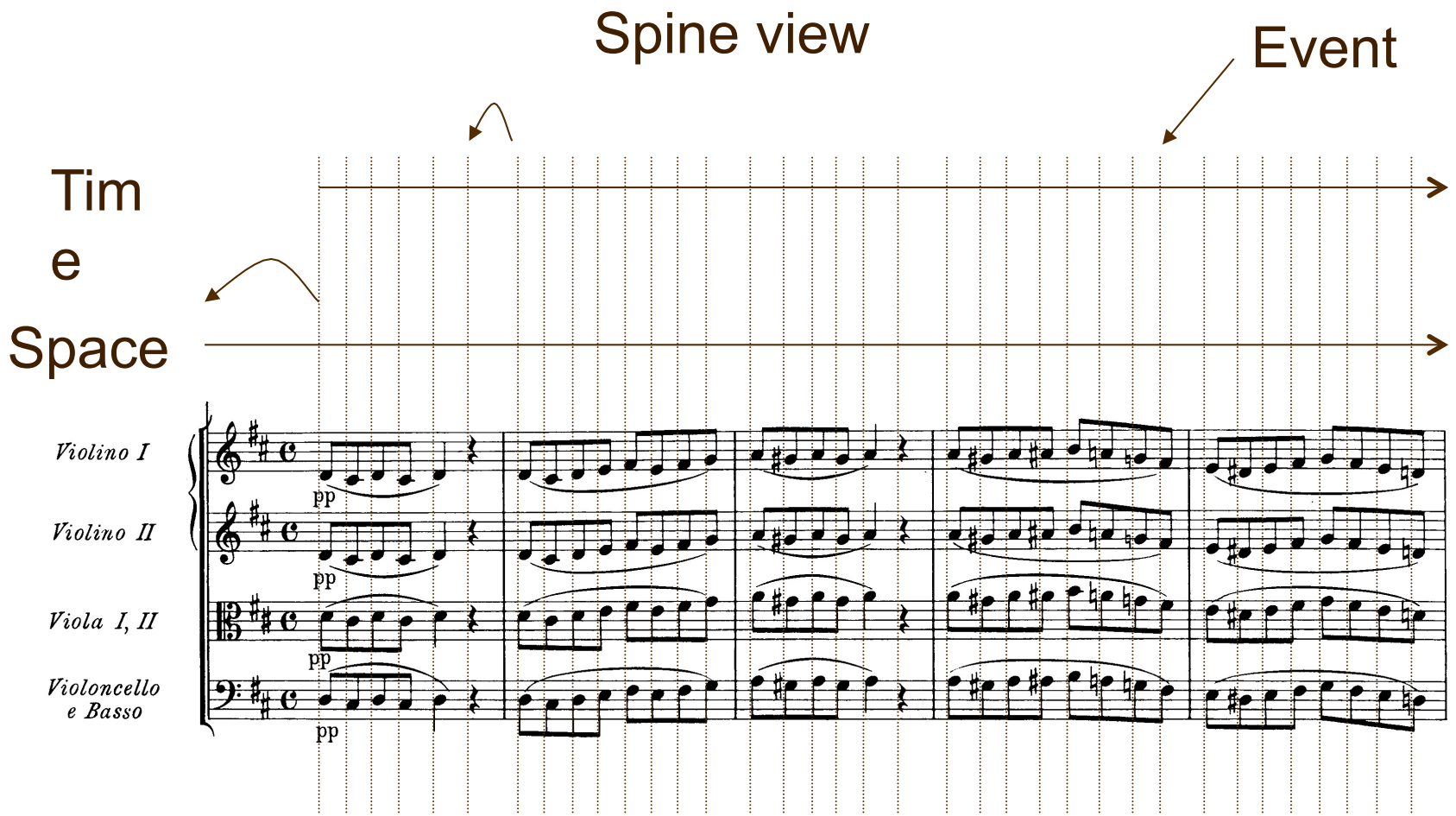
MX – Elementi principali

- ✓ Spine
- ✓ Score
- ✓ Layout
- ✓ Performance
- ✓ Altri aspetti ancora in fase di studio

MX - Spine

- Evento
- Distanza temporale (vtu)
- Distanza spaziale (vpx)
- Coordinate relative all'evento precedente

MX - Spine


Spine view
Event
Tim
e
Space
Violino I
Violino II
Viola I, II
Violoncello
e Basso
pp
pp
pp
pp

MX – Score

- Stafflist
- Part
  - Measure
    - Voice
      - Notation symbols
- Horizontal Symbols
- Lyric

MX – Layout

- Page
  - Frames
    - System
      - Staffpiece
        - Lyricpiece
    - Images
    - text
- Shapes
  - SVG (Support Vector Graphics)

MX – Performance

- ✓ MIDI information
- ✓ Audio Links
- ✓ Aperto a sviluppi, per esempio link verso video.

Verso la definizione di uno standard

- ✓ Sicurezza
- ✓ Braille
- ✓ Performance
- ✓ Links a formati Audio
- ✓ Informazione strutturata
- ✓ Versioni differenti della stessa partitura

Discussion and further work

- XML for Music: IEEE Project Authorization Report 1599
- Prima Conferenza Internazionale del Working Group sará tenuta dal 19 al 20 Settembre 2002 al DSI

Domande