The expressive power of digital formats

Fabio Vitali
Università di Bologna
... or...

criticizing the manicure of the wise man pointing at the moon
A summary of my argument

- The problem: representing text digitally
- The solutions: we have a multitude of formats
- The discussion: find the best format
- The bad: there is no best format
- The good: all formats are, in a way, equivalent
- The deduction: there is a way in which formats are not equivalent
- The odd: the non-equivalence is aesthetic rather than substantive

The take-home message: aesthetics is important in representing text digitally
Representing text digitally
Data

- **Records**
  - Structures describing entities by enumerating their properties
- **Tables**
  - Collections of data as lists of homogeneous records
- **Trees**
  - Hierarchies of data and collections
- **Graphs**
  - Networks of information structures more or less densely intertwined
What about text?

• Text is hard

• Text contains relevant information, but its structure predates digitally representable information collections.

• It is not data. It is not a structure. It is not a collection.

• It is not organized in records, tables, trees, graphs.
Text

- Text has characters, including punctuation
  - We all (sort of) agree on this
- Texts is ordered
  - In "To be or not to be", it is important that "To be" comes before "not to be"
- Text has structure
- Text has presentation
- Text has grammar
- Texts has semantics
- Text has variants
- Text has a lot of things that can be said about it
Data vs. Information

DATA
- Numbers, dates, names
- Research
- Creation
- Gathering
- Discovery

INFORMATION
- Texts, documents, multimedia
- Presentazione
- Organizzazione

KNOWLEDGE
- Concepts
- Relations
- Conversation
- Storytelling
- Integration
- Global
- Local
- Personal

WISDOM
- Actions
- Contemplation
- Evaluation
- Interpretation
- Retrospection

Objective vs. Subjective

Shedroff, 1999
A multitude of formats
A brief list of formats

• Data structures
  – CSV and tabular data
  – JSON
  – RDF

• Plain text formats
  – Plain text
  – TeX, LaTeX, etc.
  – Markdown, CommonMark and wiki syntaxes

• Markup formats
  – HTML, HTML5
  – XML
  – HTML5+ Embedded annotations (e.g., HTML5 + RDFa)
  – Markup spinoffs for overlapping (e.g. LMNL, TexMECS, etc.)
Data structures:
CSV & tabular data

CSV (Comma-Separated Values)
- A long-time format for data exchange for tabular data.
- Each row is a record, each item is a value for a field.
- Data: pretty good, as long as it is regular and structured
- Hierarchical data: no support.
- Text: only plain, short text. There are issues with return characters and tabs in the text.
- References: not explicitly, but born for relational data
- Annotations: treated as regular data

Year, Make, Model, Description, Price
1997, Ford, E350, "ac, abs, moon", 3000.00
1999, Chevy, "Venture " Extended Edition"", "", 4900.00
1999, Chevy, "Venture " Extended Edition, Very Large"", , 5000.00
1996, Jeep, Grand Cherokee, "MUST SELL!
air, moon roof, loaded", 4799.00
Data structures: JSON

JSON (JavaScript Object Notation)

- Native to Javascript applications (used client-side on the Web),
- Available in all programming languages
- Data: pretty good
- Hierarchical data: pretty good
- Text: only plain text.
- References: no support
- Annotations: as regular data

```json
{ "menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuItem": [
      { "value": "New",
        "onclick": "CreateDoc()"
      },
      { "value": "Open",
        "onclick": "OpenDoc()"
      },
      { "value": "Close",
        "onclick": "CloseDoc()"
      }
    ]
  }
}
```
Data structures: RDF

- RDF (Resource Description Framework)
  - Highly semantic, highly structured, everything expressed as a collection of simple and straightforward assertions (subject predicate object)
  - Not actually a format, rather a conceptual model that can be expressed in a number of actual formats (including Turtle, RDF/XML, RDFa, etc.)
  - Data: pretty good, but totally fragmented
  - Hierarchical data: no support.
  - Text: only plain, short text.
  - Validation: yes, through OWL ontologies
  - References: full support
  - Annotations: very good support. Designed for it
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<typeof_green-goblin>
  rel:enemyOf <typeof_spiderman> ;
  a foaf:Person ;  # in the context of the Marvel universe
  foaf:name "Green Goblin" .
<typeof_spiderman>
  rel:enemyOf <typeof_green-goblin> ;
  a foaf:Person ;
  foaf:name "Spiderman", "Человек-паук"@ru .
Plain text formats: Markdown & wiki syntaxes

Text formats that use *ad hoc* plain text tricks to represent structures and typography

- Data, hierarchical data: inappropriate
- Text: full support for long text
- Presentation of data: only the simplest typography
- References: well... hypertext links
- Annotations: inappropriate

An h1 header
=============

Paragraphs are separated by a blank line.

2nd paragraph. *Italic*, **bold**, and `monospace`.

Itemized lists look like:

* this one
* that one
* the other one
Markup formats: HTML & HTML 5

Current flagship representation format for text documents within the Web and the W3C

- Support is universal, tools are uncountable
- Mix of markup structures, Javascript APIs for advanced behavior, CSS for styling
- Data: represented as text (e.g. in a table)
- Hierarchical data: not naturally
- Text: full support for long text
- Presentation of data: fairly sophisticated typography
- Hierarchical text: now it is supported
- Validation: not possible without ad hoc code
- References: hypertext links
- Annotations: hidden in data- attributes
This is a frequent combination of formats for text documents that contain semantically relevant data.

– Combines the success and genericity of HTML 5 plus the success and genericity of RDF
– HTML 5 text and RDFa information coexist, but do not really talk to each other.
– Data: represented as RDF hidden in attributes
– Hierarchical data: as RDF hidden in attributes
– Text: as HTML
– Presentation of data: as HTML (i.e., as CSS)
– Hierarchical text: as HTML
– Validation: only for the RDF part
– References: full support
– Annotations: as RDF hidden in attributes
RDFa - example

<html prefix="foaf: http://xmlns.com/foaf/0.1/
f: http://vitali.web.cs.unibo.it/a-rdf-ns#">
    
    <head>
        <title>Prof. Fabio Vitali</title>
    </head>

    <body about="f:fv">
        
        <p>
            <em property="foaf:name">Fabio Vitali</em> is the teacher of <q about="f:tw" property="foaf:name" content="Web Technologies">the web tech course</q> at the University of Bologna.
        
    </p>

    </body>

</html>
Markup formats: XML

- XML allows to create markup languages that are readable, generic, structured, hierarchical.
  - Data: no problem
  - Hierarchical data: no problem
  - Text: no problem
  - Presentation of data: after transformation via XSLT
  - Hierarchical text: no problem
  - Validation: no problem
  - References: no problem
  - Annotations: as attributes or ad hoc sections of the document
The pyramid of sense-making
Tibère, la surprenant
Poursuivez...

Agrippine

Quoi, Seigneur?

Tibère

Le propos détestable
Où je vous ai surprise.

Agrippine

Ah ! Ce propos damnable
D’ une si grande horreur tous mes sens travailla.
<TEI>
<l>
  <sp><speaker>Tibère</sp>Poursuivez…</l>
  <sp><speaker>Agrippine</sp>Quoi, Seigneur?</l>
  <sp><speaker>Tibère</sp>Le propos détestable</l>
</l>
<l>
  où je vous ai surprise. </l>
  <sp><speaker>Agrippine</sp>Ah! Ce propos damnable</l>
</l>
<l>
  d’une si grande horreur tous mes sens travailla</l>
</l>
</TEI>
Overlapping as a *disease of affluence*

- **The *real cause*** for overlapping is the wish to do too much with the same data.
  - Introduced by bored, revengeful scholars after their real problems had been solved by XML.
  - A clear case of *appetite comes with eating*
  - Unheard-of requirement outside of the XML community
- In most cases, we have overlaps when two separate hierarchies are forced to coexist over the same text
  - In Agrippine, the structure of poetry and the structure of the speeches.
  - Each structure is a well-behaving hierarchy in isolation, and becomes problematic when in mixed company
  - No, I'm not going to talk about self-overlap
Markup formats: formats for overlapping

- Syntaxes allowing for multiple hierarchies to coexist over the same data.
- Most look like XML on steroids. For instance: LMNL, XConcur, TexMECS

[excerpt]The Housekeeper[auth]Robert Frost[auth]

He manages to keep the upper hand

On his own farm.

But as to hens:

We fence our flowers in and the hens range.
Which is the best format?
<table>
<thead>
<tr>
<th></th>
<th>Data</th>
<th>Text</th>
<th>Hierarchies</th>
<th>Presentation</th>
<th>Validation</th>
<th>References</th>
<th>Annotations</th>
<th>Overlapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML+RDFa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlapping formats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Handovers and workarounds

- Markdown has no sophisticated typography
  - But it can hand over it to HTML+CSS
- JSON does not support structured texts
  - But it can hand over them to HTML
- HTML does not support data and relations
  - But it can hand over them to RDF via RDFa
- XML does not support overlaps
  - But workarounds exist such as standoff, milestones and segmentation
- RDF does not support structured text nor overlapping
  - We designed a specific OWL ontology, called Earmark, as a workaround for that.
The same table with handovers and workarounds

<table>
<thead>
<tr>
<th>with handovers &amp; workarounds</th>
<th>Data</th>
<th>Text</th>
<th>Hierarchies</th>
<th>Presentation</th>
<th>Validation</th>
<th>References</th>
<th>Annotations</th>
<th>Overlapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML+RDFa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlapping fmts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
... And some coding...

• Some simple CSS selectors in JQuery for adding some basic validation to HTML

• Some conversion to create HTML can allow for easy rendering of complex documents.

• Etc. ...
If you do not mind some coding...

<table>
<thead>
<tr>
<th>with handovers &amp; workarounds &amp; some coding</th>
<th>Data</th>
<th>Text</th>
<th>Hierarchies</th>
<th>Presentation</th>
<th>Validation</th>
<th>References</th>
<th>Annotations</th>
<th>Overlapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML+RDFa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlapping formats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using HTML5

- Visualization of differences in Alessandro Manzoni's nove "I promessi sposi", in the 1827 and 1840 versions.

http://www.fabiovitali.it/filologia/

Quel ramo del lago di Como che volge a mezzogiorno tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, viene quasi a un tratto a rstringersi e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia riviera di rincontro, e il ponte, che ivi congiunge le due rive, par che renda ancor più sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda rincinacia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e allentarsi in nuovi golfi e in nuovi seni. La riviera, formata dal deposito di tre grossi torrenti, scende appoggiata a due monti contigui, l'uno detto di san Martino, l'altro, con voce lombarda, il Resegone.
Using HTML5

- Visualization of differences in Alessandro Manzoni's nove "I promessi sposi", in the 1827 and 1840 versions.

http://www.fabiovitali.it/filologia/

<p id="n0">Quel ramo del lago di Como che volge a mezzogiorno tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli,</p>

```html
<span class="replace toscanizzazione visibile">
  <span class="newVersion">vien</span> <span class="oldVersion">viene</span>
</span>

quasi a un tratto a ristringersi e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia ...
```

</p>
So, all formats are equivalent

... in a way...
All formats are equivalent

• Equivalent in that all formats, if you allow some handovers, some workaround and some coding, can express pretty much the same things
  – But they become longer
  – But they are more difficult to read
  – But they are more difficult to maintain (and the code, too)

• These are not substantial objections, but aesthetic objections.
HTML + RDFa example

• HTML+RDFa can support most of the requirements listed above, but in a hardly readable way...

<p>In 1969, Led Zeppelin recorded their second album, <q>Led Zeppelin II</q>, and King Crimson recorded their first, <q>In the Court of the Crimson King</q>.</p>
HTML + RDFa example

- HTML+RDFa can support most of the requirements listed above, but in a hardly readable way...

```html
<p>In 1969, <span about="lz" typeof="ex:MusicGroup" property="foaf:name">Led Zeppelin</span> recorded their second album, <span about="kc" typeof="ex:recorded" resource="lzii"><q about="lzii" typeof="ex:Album" property="dcterms:title">Led Zeppelin II</q></span>, and <span about="kc" typeof="ex:MusicGroup" property="foaf:name">King Crimson</span> recorded their first, <span about="lz" typeof="ex:recorded" resource="itckc"><q about="itckc" typeof="ex:Album" property="dcterms:title">In the Court of the Crimson King</q></span>.</p>
```
XML example

- Akoma Ntoso is an international standard for legal documents.
- It uses XML even for semantic relationships
- Requests for switching them over to RDF were made, and rejected.

```xml
<references>
  <TLCPerson href="/ontology/person#JohnSmith"
              id="Smith" showAs="Senator John Smith" />
  <TLCRole href="/ontology/role#MFA" id="MFA"
            showAs="Minister for Foreign Affairs" />
</references>

<debate>
  <question by="#Smith" to="#MFA">
    <from>Senator John Smith</from>
    <p>I would like to ask…</p>
  </question>
</debate>
```
XML example

- Akoma Ntoso is an international standard for legal documents.
- It uses XML even for semantic relationships
- Requests for switching them over to RDF were made, and rejected.

```xml
<div class="question">
    <p class="speaker"><span about="js-as-senator" typeof="pro:RoleInTime" property="pro:withRole" resource="senator-role"><span about="senator-role" typeof="pro:Role" property="rdfs:label">Senator</span></span><span about="js" typeof="foaf:Person" property="foaf:name"><span property="pro:holdsRoleInTime" resource="js-as-senator">John Smith</span></span></p>
    <p class="speech" about="question" property="c4o:hasContent"><span property="ex:askedBy" resource="js"><span property="ex:askedTo" resource="minister-of-economy"><span about="minister-of-economy" typeof="pro:Role">I would like to ask...</span></span></span></p>
</div>
```
You can, in principle, express anything in RDF, including texts and relations between fragments.

You need to express all textual relations explicitly

- Sequence of characters
- Containment of structures
- Fragments with special semantic content: legal references, definitions, facts, concepts, people, roles, events, etc.

A specific ontology for expressing these relations exist. It is called Earmark, we created it in 2008.

It works. It is painfully precise. It is much more expressive than XML. It is also 400% longer than the corresponding XML.

http://www.essepuntato.it/2008/12/earmark
A fragment of Earmark

e:lyrics
  a :URIDocuverse ;
  :has-uri "http://www.essepuntato.it/2009/01/andiloveher.txt"^^xsd:anyURI

e:funfacts

e:attribute_values
  a :StringDocuverse ; :has-text "stanza - refrain - 4"^^xsd:string

e:time_values
  a :StringDocuverse ; :has-text "68 - 72 - 76 - 80 - 84"^^xsd:string

e:r_refrain_1
  a :Range ; :begins e:location0-lyrics ; :ends e:location6-lyrics

e:r_refrain_2
  a :Range ; :begins e:location6-lyrics ; :ends e:location14-lyrics

e:location0-lyrics
  a :XPointerLocation ; :refers-to lyrics ; :at "xpointer(point(.0))"^^xsd:string

e:refrain_p
  a :Element ,
  [ a rdf:Seq ; rdf:_1 e:r_refrain_1 ; rdf:_2 e:r_refrain_2 ] ;
  :has-general-identifier "p"^^xsd:string.
The ecosystem of text

- Editing
- Validating
- Printing & displaying
- Transforming and converting
- Annotating
  - Annotating annotations
- Searching
- ...

Aesthetics is important

• A data format is not only what you use to represent the data (or text) that you have to deal with.

• An ugly format is, basically, a format for which you have to work a lot to obtain the things you need.
  – Like putting make up on a pig to make it pretty

• It very much depends on what is the use you plan for your data (or text).
A summary of my argument (again)

• The problem: representing text digitally
• The solutions: we have a multitude of formats
• The discussion: find the best format
• The bad: there is no best format
• The good: all formats are, in a way, equivalent
• The deduction: there is a way in which formats are not equivalent
• The odd: the non-equivalence is aesthetic rather than substantive

• The take-home message: aesthetics is important in representing text digitally
The manicure of the wise man pointing at the moon is actually the important part.
... thank you!