Create your own interactive diachronic semantic maps: a flexible and user-friendly open-source tool for historical linguistics

Francesca Dell'Oro, Loris Rimaz, Helena Bermúdez Sabel

1. What is Pygmalion?

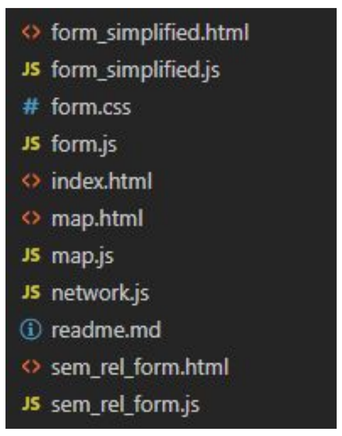
Pygmalion is a user-friendly open-source tool conceived to draw and visualise diachronic semantic maps. It was designed in the framework of the WoPoss project and it comes in two flavours:

- **Pygmalion-simple**: for any diachronic semantic map
- **Pygmalion-modal**: for diachronic semantic maps describing the evolution and the relations between modal readings

2. Data visualisation: development

Key components:

- HTML5, CSS3, ECMAscript 6
- Data-Driven Documents (D3)



Main steps:

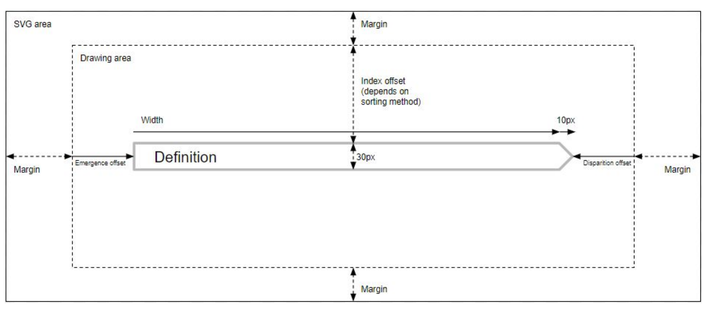
1. Data gathering – forms
2. Data structuring – preprocessing
3. Data visualisation – drawing pipeline

Main visualisation:

1. Chronological line
2. Definitions (and modal readings):
   - Sort algorithm
   - Positions calculations
   - Height correction
3. Left-hand side metadata – reflect sort method: gather elements by semantic group or collocation
4. Right-hand side metadata (when a definition is clicked):
   - Rather complex path calculations
   - Certain or hypothetical relationship
   - Direction of the relationship (if any)

Network graph:

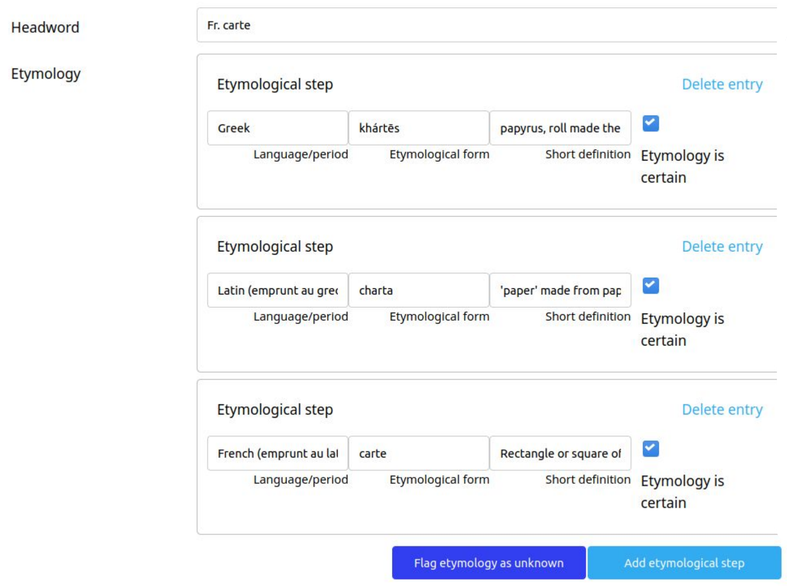
1. Data restructure – keep only the relevant information
2. Simulation elements (nodes and links)
3. Simulation environment (type, forces applied)



Relevant/interesting code snippets:

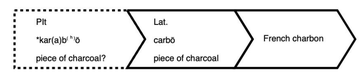
```
function prepareDefinitions() {
 const meanings = data.meanings;
 const definitions = [];
 if (data.normalForm) {
  meanings.forEach((meaning) => {
   if (meaning.modalities.length > 1) {
    meaning.modalities.forEach((modalitiy) => {
     definitions.push(modalityFormatting(meaning, modalitiy));
    });
   } else {
    definitions.push(modalityFormatting(meaning, meaning.modalities[0]));
   }
  });
 } else {
  meanings.forEach((meaning) =>
   definitions.push(simpleModalityFormatting(meaning))
  );
 }
 return definitions;
}

const offset =
lines[elementIndex] * 30 + wrap(element.meaning, cW, cP, element) * 15;
const x0 =
element.disparition != -1 && !isNaN(element.disparition)
? element.disparition * cP + 10
: cW + 10;
const y0 = elementIndex * 37 + offset;
```

3.1 Data entry: headword and etymology



A) Headword
B) Etymology
   a) certain

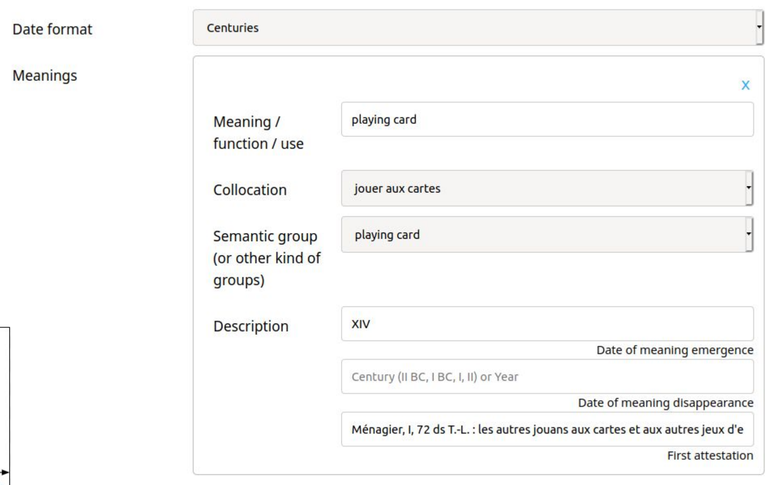
   b) uncertain



   c) unknown

Flag etymology as unknown

3.2 Data entry: meanings



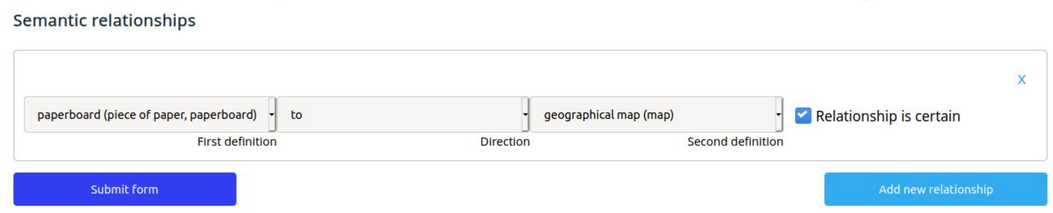
C) Date format:
- centuries
- decades
- specific years

D) Meanings
- meaning/function/use (mandatory)
- collocation
- semantic group (or other kind of group)
- description
  - date of meaning emergence (mandatory)
  - date of meaning disappearance
  - first attestation

Pygmalion-modal includes additional subfields in the field "description":

E) Description of modality (recursive):
- modality type
- certainty of the modal description

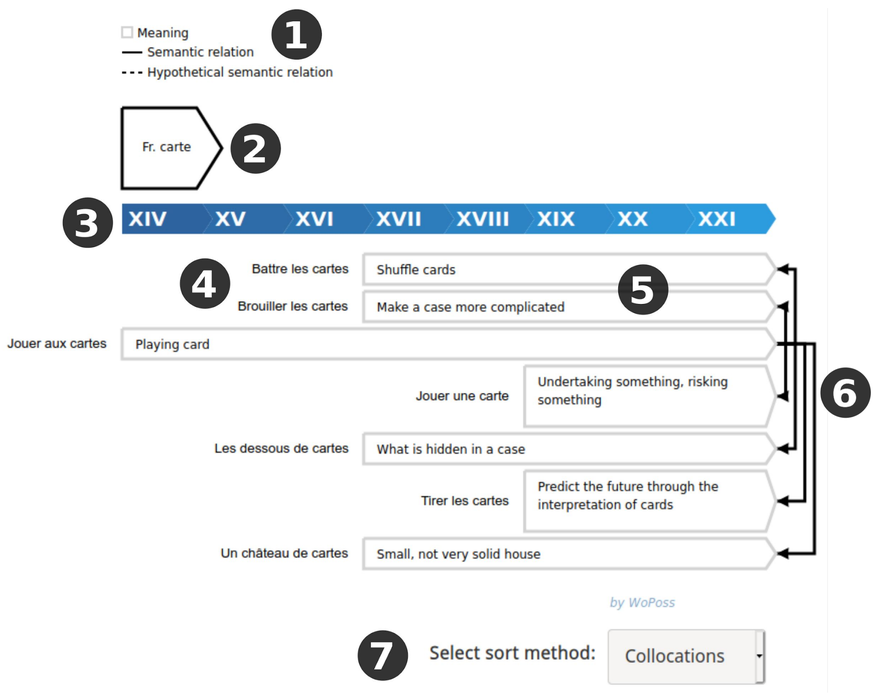
3.3 Data entry: relations between meanings/functions



F) Direction of the relation:
- from
- to
- unspecified

G) Certainty of the relation

4. Visualisations



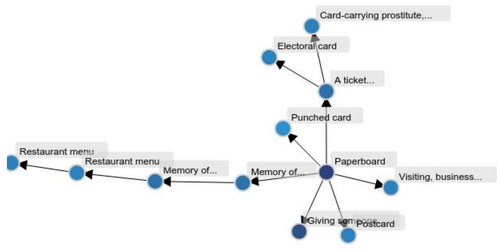
4.1 Visualisation 1

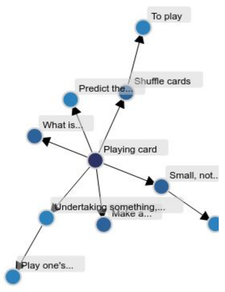
1) Legend
2) Headword and etymology
3) Chronology
4) Collocations (or semantic groups if selected)
5) Meanings / uses
6) Semantic relations (with the direction)
7) Sorting options:
   a) Chronologic
   b) Collocations
   c) Groups

When **clicking on** a meaning, the semantic relations of that meaning are visible (6). Double-clicking resets the visualization.

When **mousing over** a meaning, the chronology and the first attestation are visible.

Modal maps: meanings are color-coded by **modality type**





4.2 Visualisation 2: Network

- Node background color: chronology
- Node border color: modality (not displayed here)
- Edges: direction

To know more:

WoPoss website: **http://woposs.unine.ch/**
Pygmalion website: **http://woposs.unine.ch/pygmalion.html**
Pygmalion Github repository: **https://github.com/WoPoss-project/Pygmalion**

FNSNF
Swiss National Science Foundation