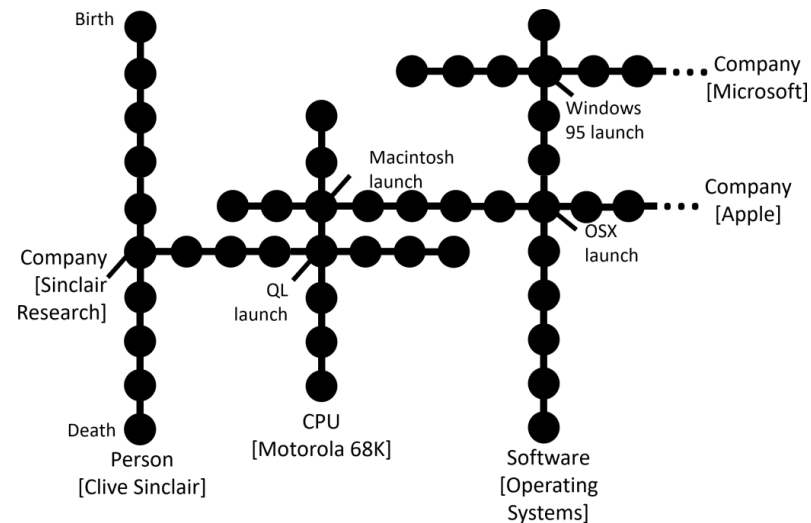


FOSDEM

Computer
Museum
NAM-IP

Trajectware - Timeline-Based Navigation across Computing Heritage



PONSARD Christophe NAM-IP

FOSDEM 22 – Retrocomputing – February 5 (online)

Context – NAM-IP Computer Museum

- Located in Namur/Belgium - 30' from Brussels
(worth a visit when FOSDEM is back at ULB or if you come in Belgium!) www.nam-ip.be
- Missions:
 - Preservation: safeguarding digital heritage, focus on local pioneers
 - Acquisition of artefacts, enriching collections
 - **Exhibitions: for all, specific animation, permanent/temporary**
 - Research: about machines, software, communities
- “Container design”, an historical parallel



Our problem – Providing Support for Exhibition

- Physical experience provided by an exhibition is great
 - Real hardware (outside/inside), possibly running (on emulator), advertisements
 - Scenography = immersion, grouping/sequencing to make sense, common thread,...

- But limited by physical media !
 - Explanatory poster = a selected viewpoint
 - Limited space = limited level of detail
 - Accessibility (FR/NL/EN need):
 - either one language for posters/video (subtitling)
 - or devote space(posters) /time (video)

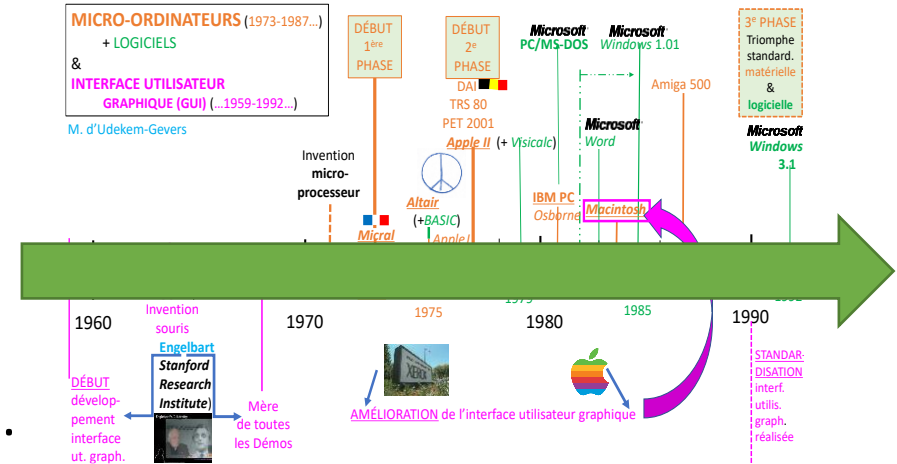


➔ Idea to complement physical experience with an “digital experience”

- Mobile application: easy to deploy and use
- More detailed content: pictures, videos, off-line or online, multiple languages
- Dynamic perspective: discover make your own connections
 - across artefacts, themes, actors,...
- Could be used before/during/after physical experience
 - ➔ requires links with physical experience (especially in “during” case)
- Beyond this support to visits : could be used as learning/research tool

Why Selecting a Timeline Approach ?

- In our case... timelines are everywhere (explicitly or implicitly)
 - Chronological sequencing of the exhibition
 - Main tread = co-emergence and cross-fertilisation of user interfaces and micro-computers
 - Many posters with milestones: GUI, games, Moore's law...
 - Panel with major CPU



Dès 1959, Douglas Engelbart élabore l'idée d'augmenter l'intellect humain qui sera à la base des interfaces utilisateur graphiques.

En 1964, il invente la souris avec W... mother of all demos» combinant souris, traitement de texte, hypertexte, le tout mené sous forme de vidéoconférence.

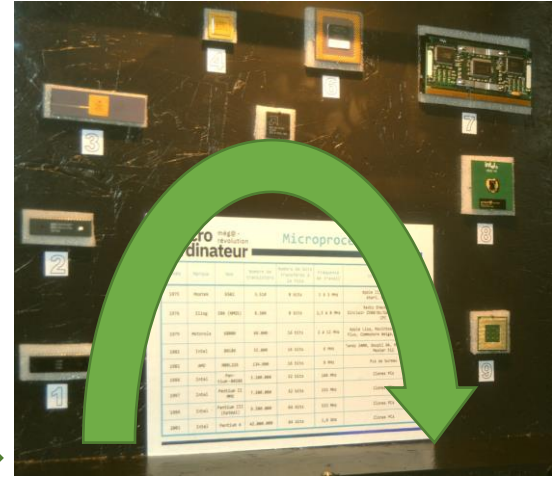
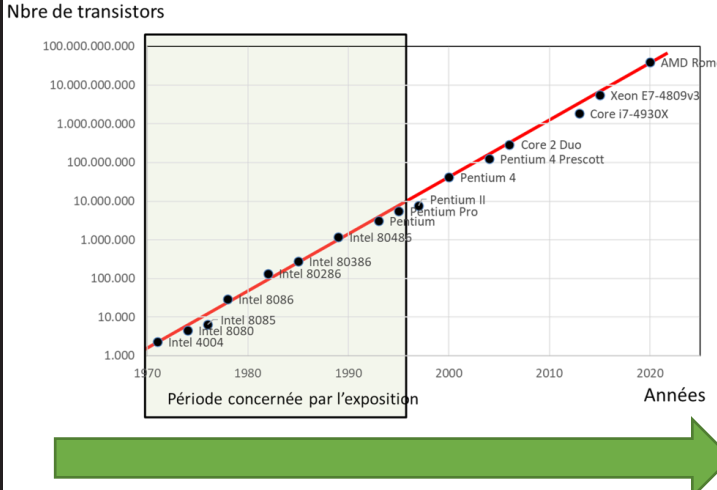
En 1968, il réalise une démonstration «mother of all demos» combinant souris, traitement de texte, hypertexte, le tout mené sous forme de vidéoconférence.

Dans les années 1970, les recherches se poursuivent au Xerox PARC (Palo Alto Research Institute) et aboutissent au prototype ALTO un concentré d'innovations, notamment:
- le paradigme des fenêtres (Windows, Icons, Menu, Pointer)
- la métaphore du bureau pour organiser son travail
- le traitement de texte (WYSIWYG (What You See Is What You Get))
Ces recherches ont été largement visibles.

En 1979, la visite de Steve Jobs chez Xerox contribue à la prise de conscience d'Apple et à la création de l'interface graphique pour le grand public.

En 1983, Apple produit le LISA doté d'une interface graphique mais trop cher, c'est un échec commercial.

Début 1984, l'Apple Macintosh, est le premier micro-ordinateur doté d'une souris et d'une interface graphique accessible au grand public. Il disposait d'un traitement de texte (MacWrite) et d'un logiciel de dessin (MacPaint).



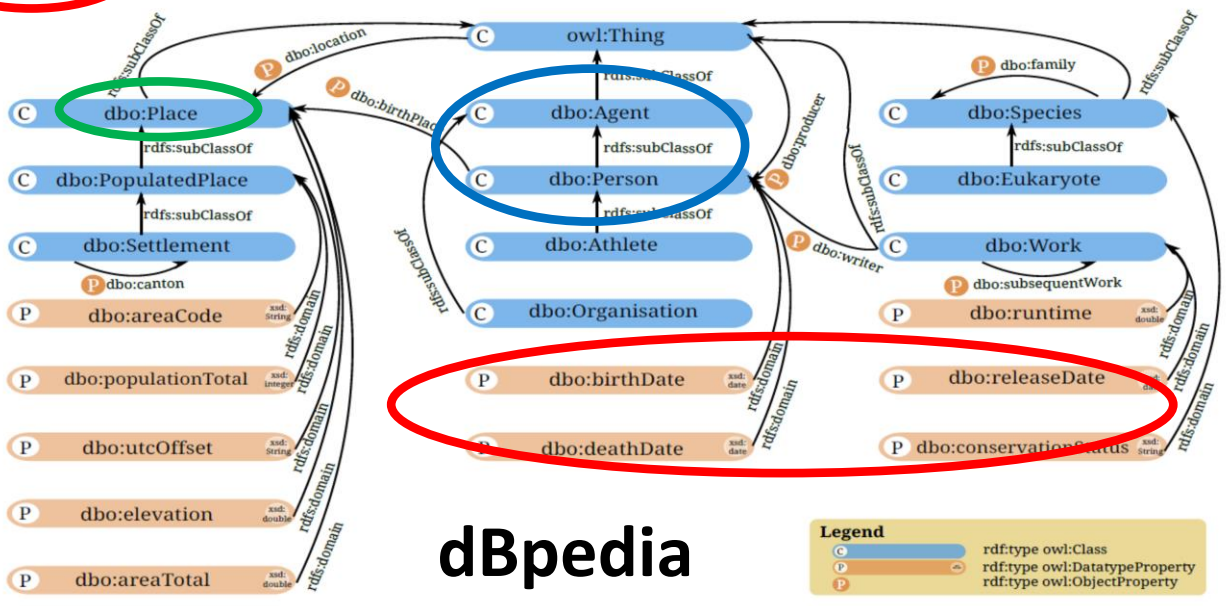
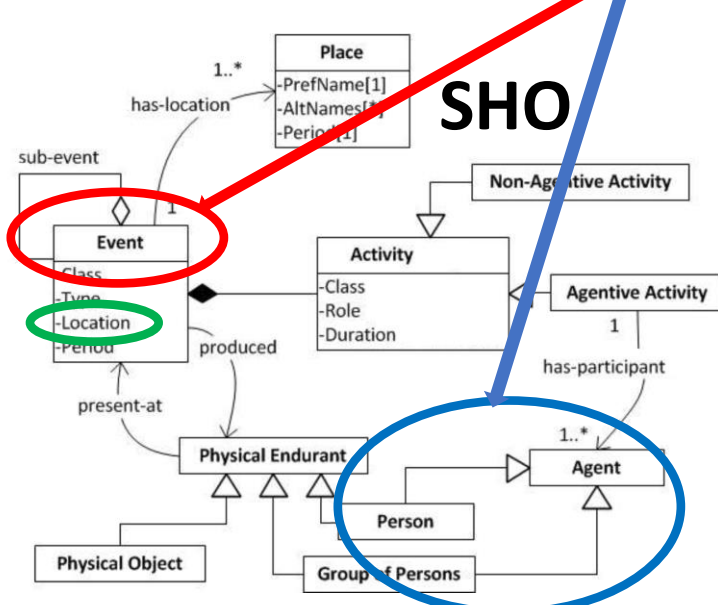
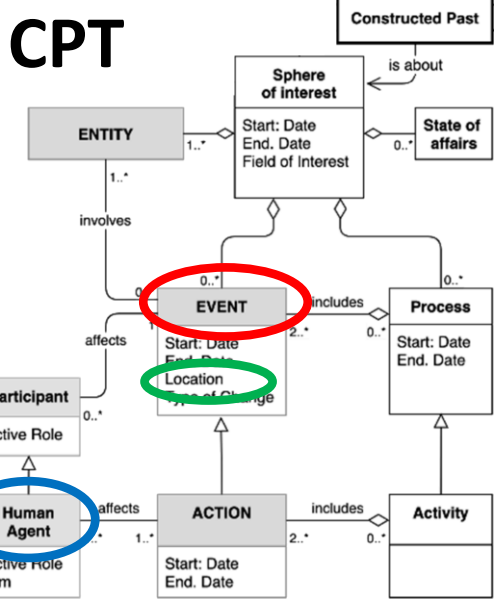
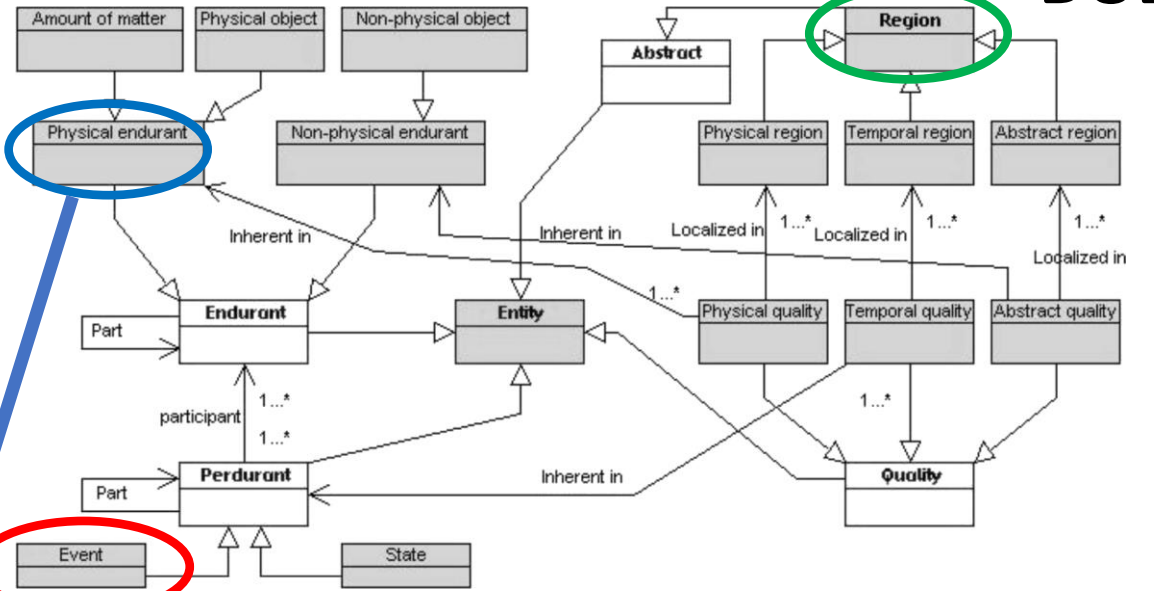
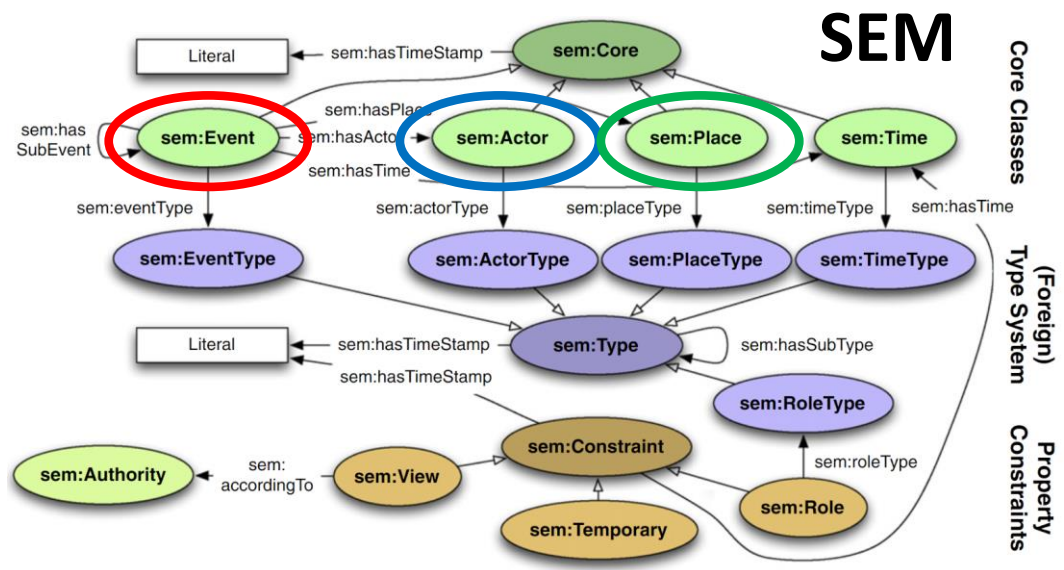
- More views, more connections possible/interesting !
- Probably also valid for other (technological) museums

Process

- Understanding the need [done]
- Understanding what to capture/structure
 - conceptual modelling, ontologies
- Front-end focus, all-in-one off-line app
 - delivery need for temporary exhibition: multilingual, available content
 - generalisation in mind
- Elaboration phase
 - internal linking (automated)
 - more elaborated navigation scenarios
 - back-end extraction, on-line mode, Open API design
- Next steps...

(Meta-)Models /Ontologies of Historical Concepts

DOLCE

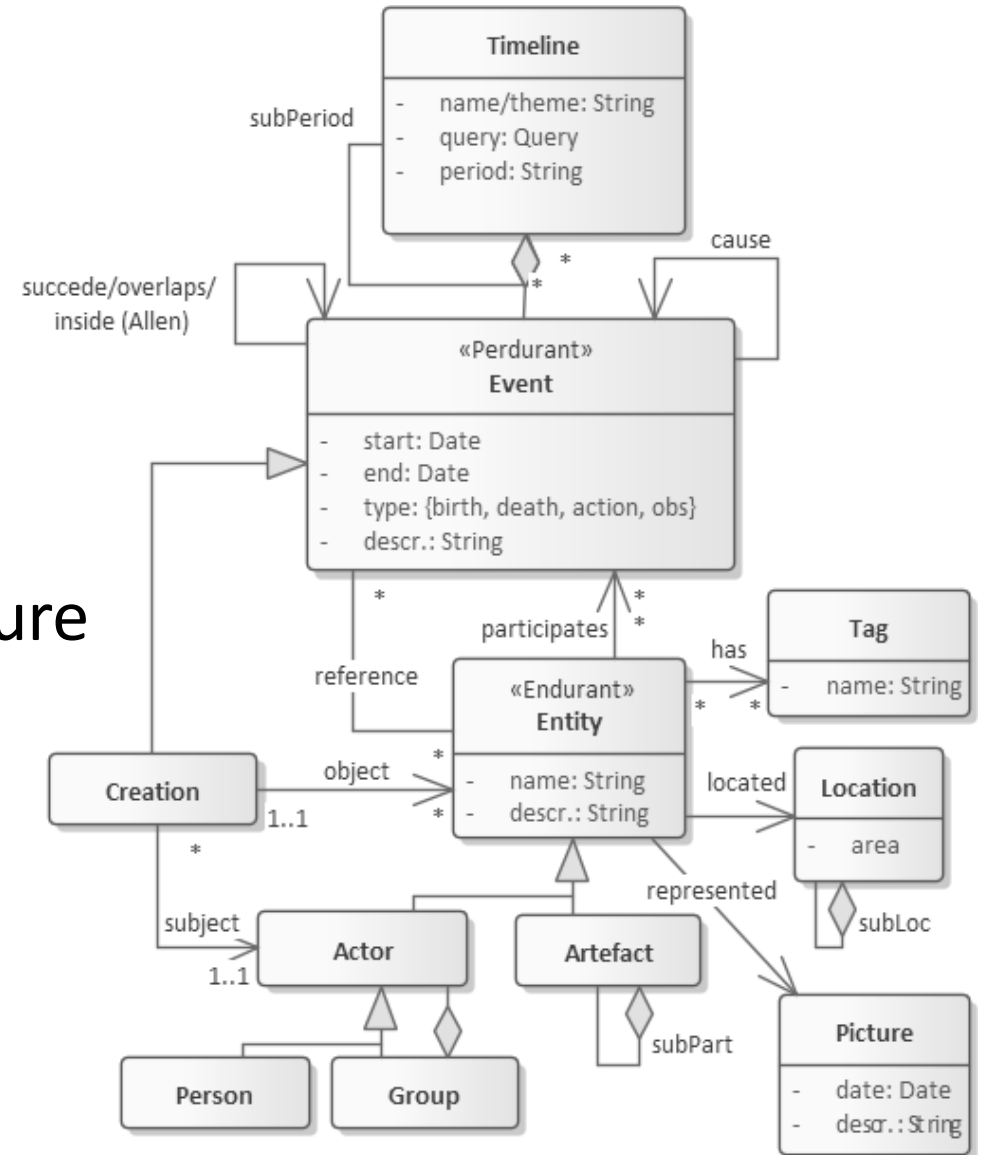


Legend

- C rdfs:type owl:Class
- P rdfs:type owl:DatatypeProperty
- P rdfs:type owl:ObjectProperty

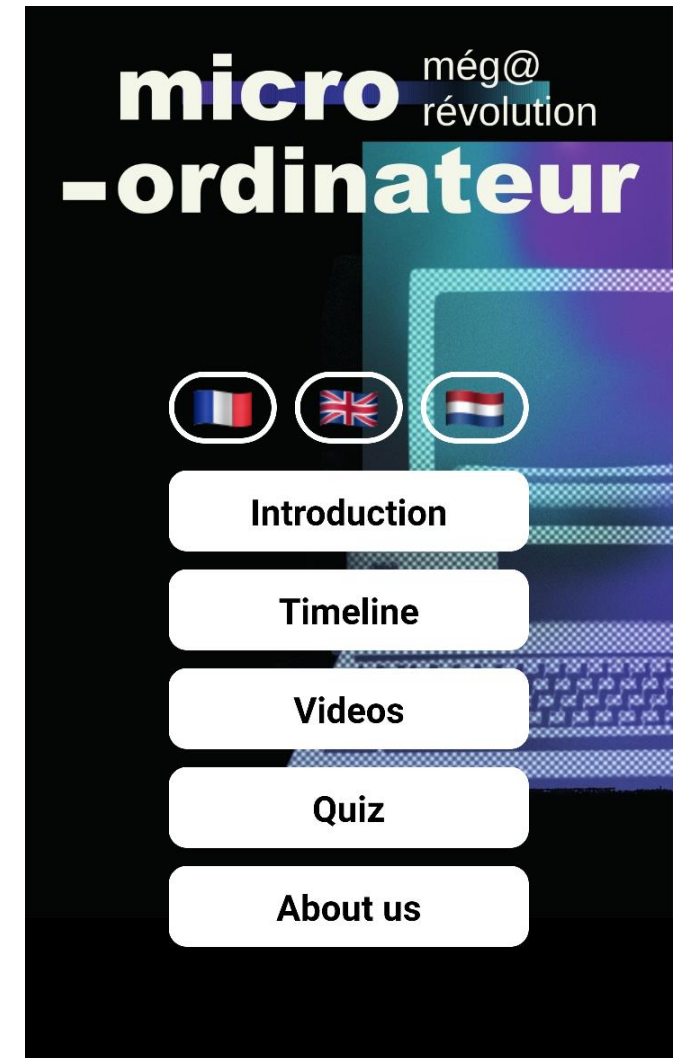
Our (inspired) modelling framework

- Timeline first class concept
- Rely on Endurant/Perdurant (stereotype)
- Generic « tagging » for extensibility
- Multiple aggregation levels on time/actors/artefacts
- App to Physical mapping: id, location, picture



All-in-one Off-line Application

- Selecting an Open-Source Framework:
 - Criteria:
 - cross-platform
 - scalability
 - Learnability
 - component Library
 - Perennity
 - Internationalisation
 - Candidates:
 - Flutter (Dart)
 - **React-Native (js)**
 - Ionic/angular (js)
 - Solar2D (Lua)
- Might evolve later (back-end → multiple front-ends)



Architecture



- Page instances

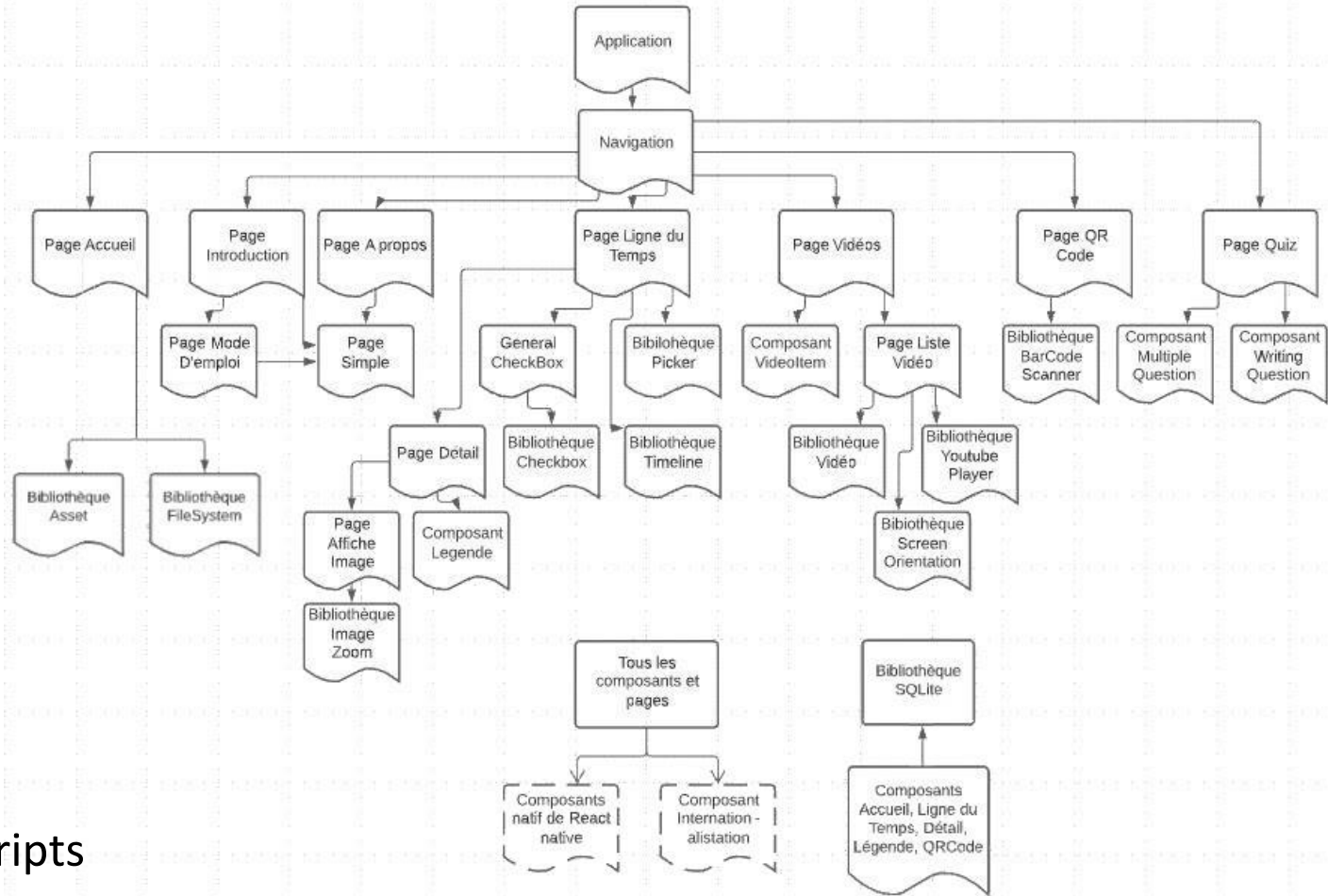
- Intro
- Timeline
- Video mngr
- Quiz
- ...

- Components:

- timeline
- details
- video
- quiz/question
- QR-code
- database
- ...

- Management scripts

- Translation
- Internal linking (automatic keyword aliasing)
e.g. "first Macintosh" = "Macintosh 128K"



DEMO
TIME

micro még@ révolution -ordinateur



Introduction

Ligne du temps

Videos

Quiz

A propos

Feedback on the Current Application

- Good points

- Very nice experience and user feedback
- Resource bundling/translation process quite easy
- Very good scalability (efficient state update, lazy loading...)
 - ➔ large timeline tested

- Known issues

- Off-line mode not great in React-Native
 - Reading a local HTML page mostly “statically linked” to resource via tables required some ad-hoc solution (page template)
 - Loading a sqlite DB locally requires some web library actually requiring an (unnecessary) WIFI connection !
- Expo Go
 - Great for testing but sometimes requires cache clearing/refresh
 - Not all libraries will work
- Large footprint of “all-in-one” solution (200 MB)
- Static relational schema

Elaboration – Dynamic On-line Backend

- Access to heavy resources: high resolution pictures, videos
- Access to dynamic timeline, through queries on conceptual resources:
 - actor(s) at different granularity levels:
 - life of a person (e.g. Clive Sinclair)
 - evolution of a group (FSF)
 - or a company (Commodore International)
 - object(s), at different granularity levels:
 - precise history of a specific object (e.g. the design of the LISA computer)
 - family of object w.r.t. specific criteria, micro-computers of a specific period, manufacturer, using a specific CPU,...
 - temporal, spatial or thematic contexts respectively through
 - Event (dates)
 - Location
 - Tag characteristics.

Open API Design → on-going implementation

The image shows the Swagger Editor interface, which is used for designing and documenting REST APIs. The interface is split into two main sections: a code editor on the left and a visual API explorer on the right.

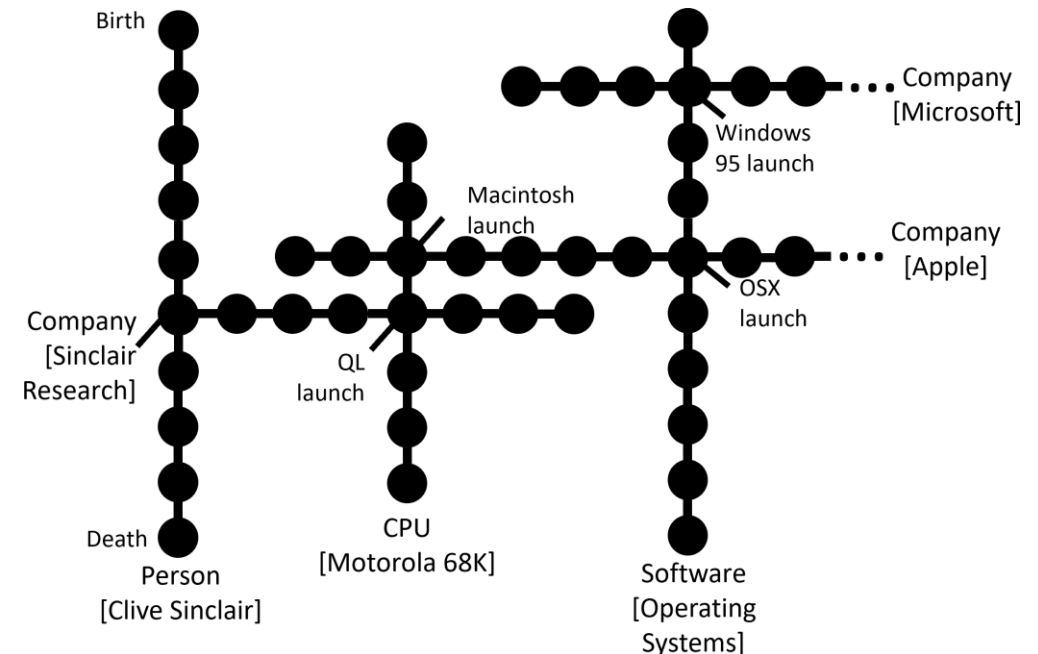
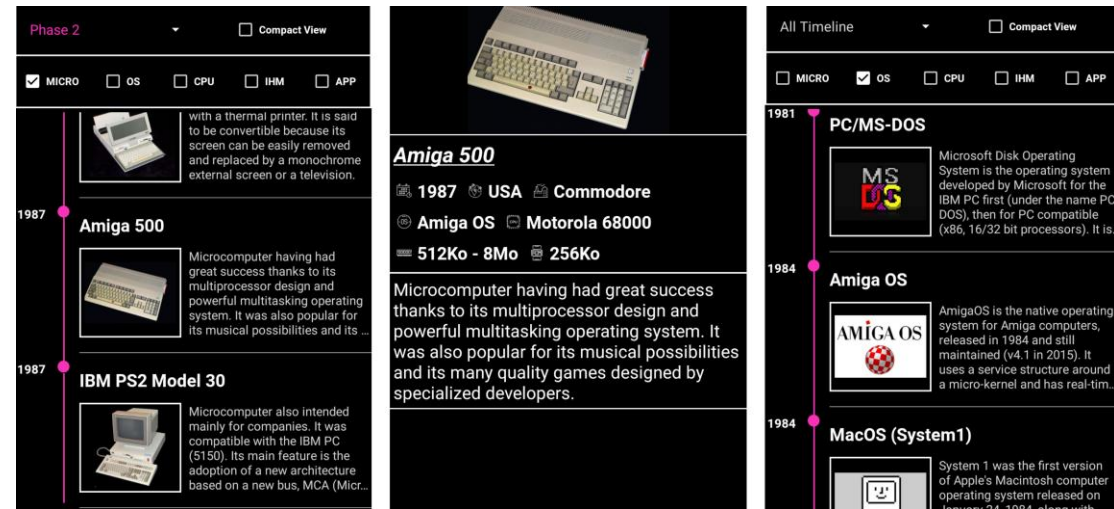
Left Panel (Code Editor): This panel displays the OpenAPI specification in YAML format. The visible code defines two endpoints: `/event` and `/invention`. The `/event` endpoint has a `post` method (summary: "Add a new event in the dataBase") and a `put` method (summary: "Update an existing event"). The `/invention` endpoint has a `post` method (summary: "Updates an invention in the dataBase with form data"). The specification includes details such as tags, summaries, descriptions, operation IDs, consumes/produces media types, parameters, security requirements, and response codes with descriptions.

Right Panel (API Explorer): This panel provides a visual overview of the API endpoints. It is organized into sections for `event` and `inventions`. Each endpoint is represented by a colored card indicating its HTTP method and a brief description. For example, under the `event` section, there are cards for `POST /event` (Add a new event in the dataBase), `PUT /event` (Update an existing event), `GET /event/findByTags` (Finds events by tags), `GET /event/findByInvention` (Finds events by their inventions), `GET /event/findByInventor` (Finds events by their inventor), `GET /event/{eventId}` (Find an inventor by its ID), `POST /event/{eventId}` (Updates an event in the dataBase with form data), and `DELETE /event/{eventId}` (Deletes an event). The `inventions` section shows a `POST /invention/{inventionId}` endpoint (Updates an invention in the dataBase with form data). Each card includes a dropdown arrow and a lock icon.

Bottom Panel (Models): This panel shows the data models used in the API. It lists `Invention` and `Inventor`, each with a right-pointing arrow indicating that their full definitions are available elsewhere in the API documentation.

Elaboration – Flexible Navigation Front-end

- Idea: allow to “jump” from one timeline to another (like switching metro)
- E.g. micro → Amiga 500 → OS context → Commodore context
- “Pivoting” operation on
 - event pivoting between related entities or features e.g. see above
 - time zoom in/out based on a defined period e.g. micro computers → “early phase”
 - actor zoom in/out, from person level to company e.g. Clive Sinclair → Sinclair Research
 - object zoom in/out e.g. down to version level and up to product family
 - relations inclusion, possibly iterative/closure e.g. to look for event causes/consequences
 - combining multiple timelines, either merged or keeping them visually separated (+ synergies) e.g. GUI // micro-computers



Elaboration - Tentative Information Extraction

- Query using SPARQL on OpenLink Virtuoso (DBPedia endpoint)

```
SELECT DISTINCT ?date ?name (GROUP_CONCAT(DISTINCT ?founder; SEPARATOR=", ") AS ?founders)
WHERE {
  ?company foaf:name ?name.
  ?company dbo:industry dbr:Computer_hardware.
  ?company dbo:foundedBy ?founder.
  ?company dbo:foundingYear ?date.
  ?company dbo:abstract ?abstract
  FILTER langMatches(lang(?abstract), 'en')
  FILTER(?date >= "19750101"^^xsd:date)
}
ORDER BY ASC(?date)
```

- Results

SPARQL HTML5 table		
date	name	founders
"1975"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Microsoft Corporation"@en	http://dbpedia.org/resource/Bill_Gates , http://dbpedia.org/resource/Paul_Allen
"1976"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Apple Inc."@en	http://dbpedia.org/resource/Ronald_Wayne , http://dbpedia.org/resource/Steve_Jobs , http://dbpedia.org/resource/Wozniak
"1976"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Matrox Electronic Systems"@en	http://dbpedia.org/resource/Lorne_Trottier
"1976"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Vector Graphic"@en	http://dbpedia.org/resource/Lore_Harp_McGovern , http://dbpedia.org/resource/Robert_Harp
"1976"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Acer Inc."@en	http://dbpedia.org/resource/Stan_Shih
"1979"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Convergent Technologies"@en	http://dbpedia.org/resource/Allen_Michels
"1979"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Core International, Inc"@en	http://dbpedia.org/resource/Hal_Prewitt
"1979"^^<http://www.w3.org/2001/XMLSchema#gYear>	"Pravetz computers"@en	http://dbpedia.org/resource/Ivan_Marangozov

To Conclude

- Timeline concept rich to support museum guide
 - Good feedback on our current exhibition
 - Plan to extend it to permanent exhibition
- Concept might be generalised
 - other museums, although underlying ontology focused to “technological” museum
 - research/learning purposes
- Try it, give feedback / ideas / contributions welcome !
 - <https://github.com/NAMIP-Computer-Museum/guideApp>
 - christophe.ponsard@gmail.com
- Also try the Quiz (sorry in French, not yet translated)
but questions to be generated from knowledge base soon ;-)