Virtual Heritage To Go

Nils Michaelis
Yvonne Jung
Johannes Behr

Fraunhofer IGD
Darmstadt, Germany

v-must

WEB 3D
2012

Fraunhofer
IGD

Qualitätsmanagement zertifiziert nach DIN ISO 9001:2000

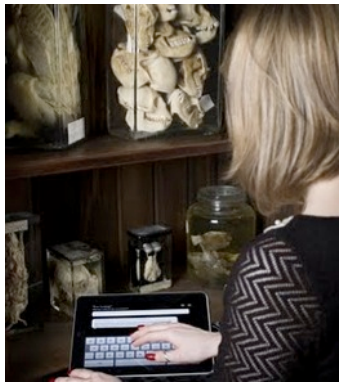
Motivation

Motivation

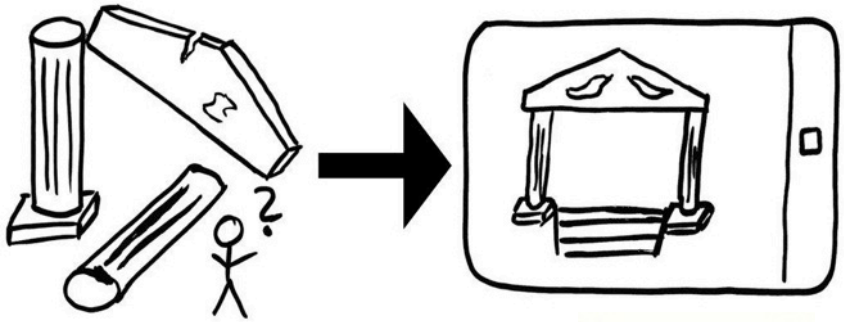
- 3D interactive content important building block for Virtual Heritage applications
  - Exploration, analysis, presentation, documentation, and reconstruction can be assisted with 3D content
  - Communication can be eased by avoiding inefficient media or the need for traveling to historical sites
  - Attention of museum visitors can be caught with modern technologies
- Mobility becomes more and more important
  - Most people have smartphones or tablets
    - Most systems require different apps
    - The need for installation etc. is a barrier
  - Mobility is essential on site
- With various devices synchronization is an issue
  - Would be nice to avoid this at all
    → Distributed application model

Motivation – Use Cases

- Virtual museum
  - iPads etc. can catch the attention of visitors
  - Objects can be virtually placed in the correct context (augmentation in original room)
  - Mobile devices to assist or replace guides
- On site
  - Visitors can display virtual models of a ruin area (→ combination with AR possible)
  - Mobile device as replacement for a guide
- Restoration and communication
  - Many people included during curation and reconstruction of a historical object
    → better communication ways required
  - Annotations help to preserve and propagate information through various stages




Rubino, I. 2011. iPads for museums: serving visitors and professional communities. Webpage: http://australianmuseum.net.au/Uploads/Documents/23474/Rubino2011 _iP %20museums.pdf



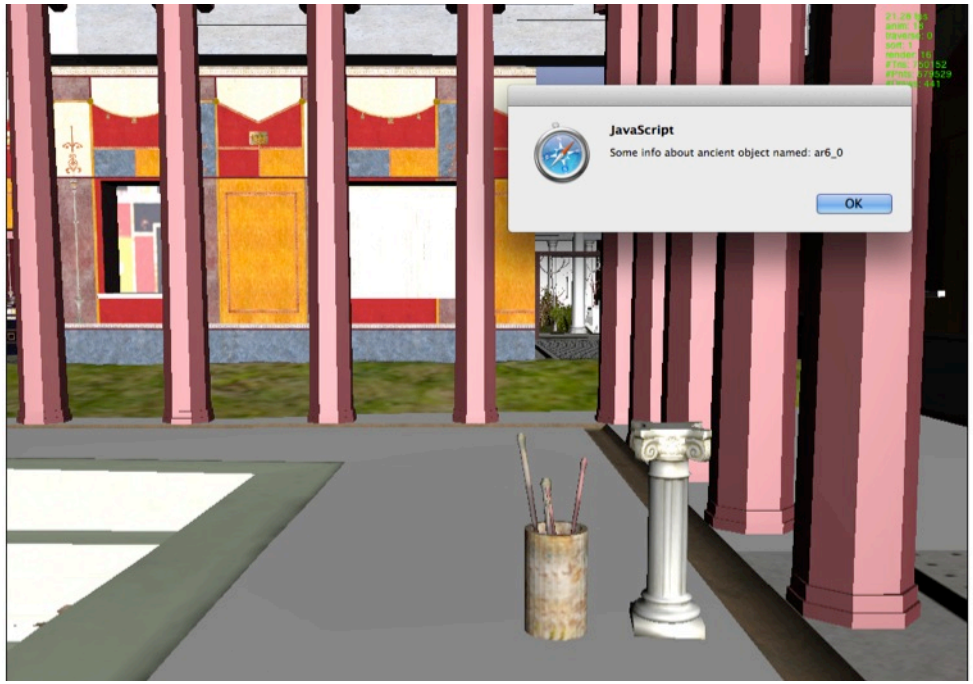


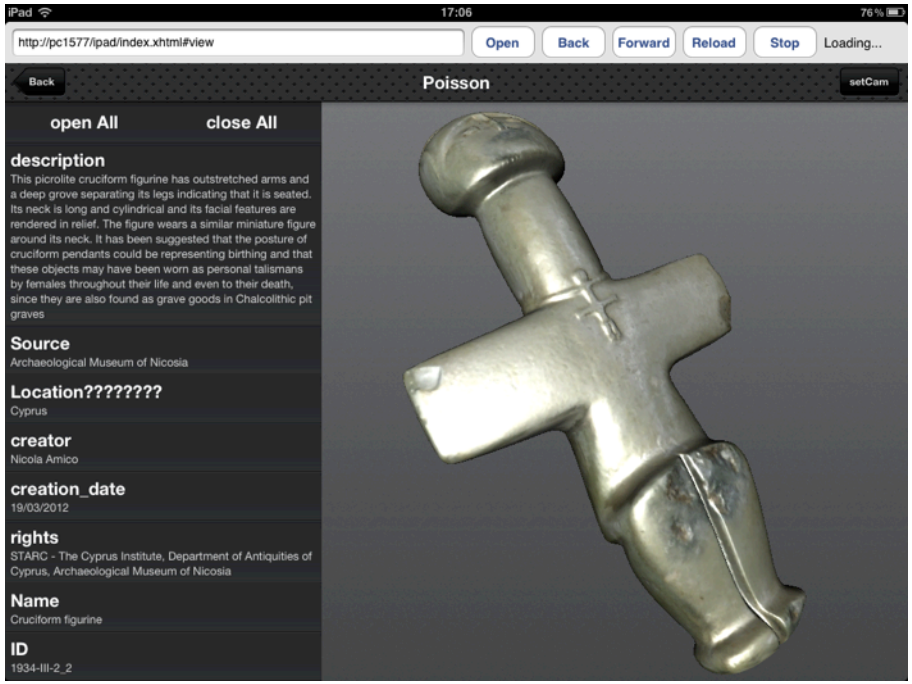
R. Scopigno, M. Callieri, P. Cignoni, M. Corsini, M. Dellepiane, F. Ponchio, and G. Ranzuglia. 2011. 3D Models for Cultural Heritage: Beyond Plain Visualization. Computer 44, 7 (2011), IEEE, 48-55.

Requirements

Technical Requirements

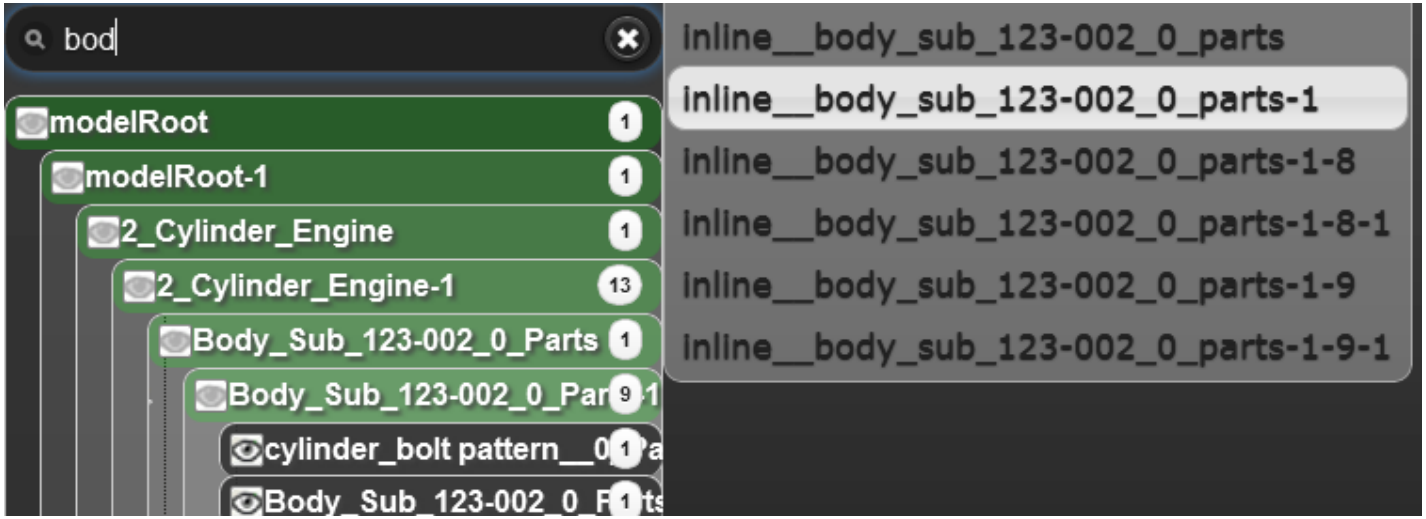
- 3D presentation in a Web Browser with integrated use of Web technology
- One single app for desktop machines and lightweight mobile devices
- Integration of 3D content into DOM tree for easing scene manipulation
- No need to install programs, plugins or special applications
- Performance and interactivity





Functional Requirements

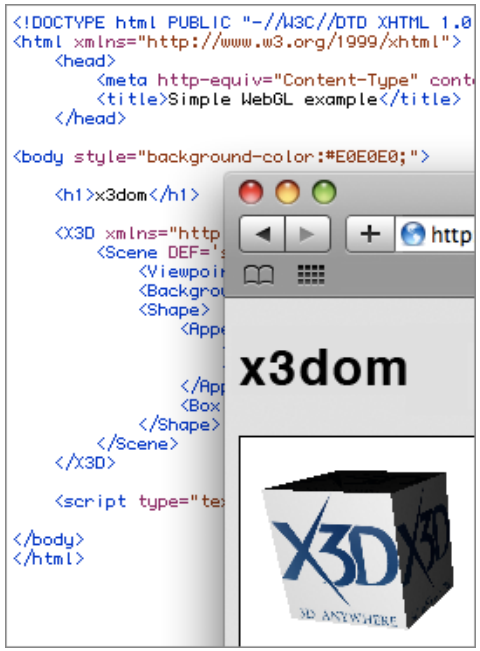
- Intuitive interaction / navigation
- Presentation of metadata in well arranged way and possibility to show corresponding (annotation) marker on the model
- Possibility to create and display annotations and corresponding marker
- Desirable to create, modify and share 3D content similar to other media
- Availability of different search filter, etc.



Technical Approach

Technologies – 3D Content

- X3DOM
  - Enables declarative 3D content inside a Web Browser with enabled WebGL or Flash 11/ Stage 3D
    - Fallback to Flash if WebGL is not available/enabled
  - Builds upon open standards → good integration with HTML5, CSS3, DOM Scripting and Ajax
  - Uses the X3D XML format with some modifications for HTML integration (e.g. for event handling)

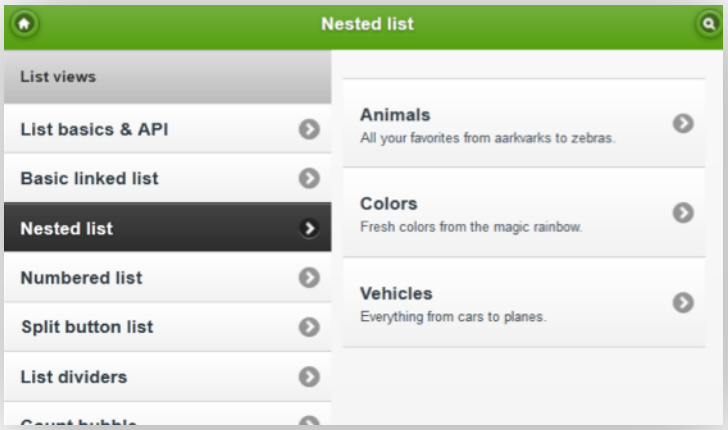


x3dom Instant 3D the HTML way!

http://www.x3dom.org/

Technologies – Mobile JavaScript Frameworks

- JQuery/ JQuery Mobile (our recommendation)
  - Very powerful
  - Easy to use
    - Easy theming
  - Well documented with many examples
- JQTouch
  - Very lightweight and performant
  - Bad documentation
  - Very nice look and feel
  - Optimized for WebKit – bad support for other Browsers
- Other Frameworks are available
  - All aim at providing look & feel of native apps



http://jquerymobile.com/demos/1.1.0/docs/lists/lists-nested.html

http://www.jqtouch.com/preview/demos/main/#ui

Technologies – InstantGLory

- To present 3D content on mobile devices without installing an app, we need a WebGL compatible Browser
- Apple's WebKit implementation on iOS (iPhone, iPad) supports WebGL, but it is not enabled in Safari
- In future this will (hopefully) change
  - Till then one can work around this by writing a little Browser wrapper around the UIWebView widget that just enables WebGL
  - Use of instantGLory – WebKit-based Browser that supports WebGL

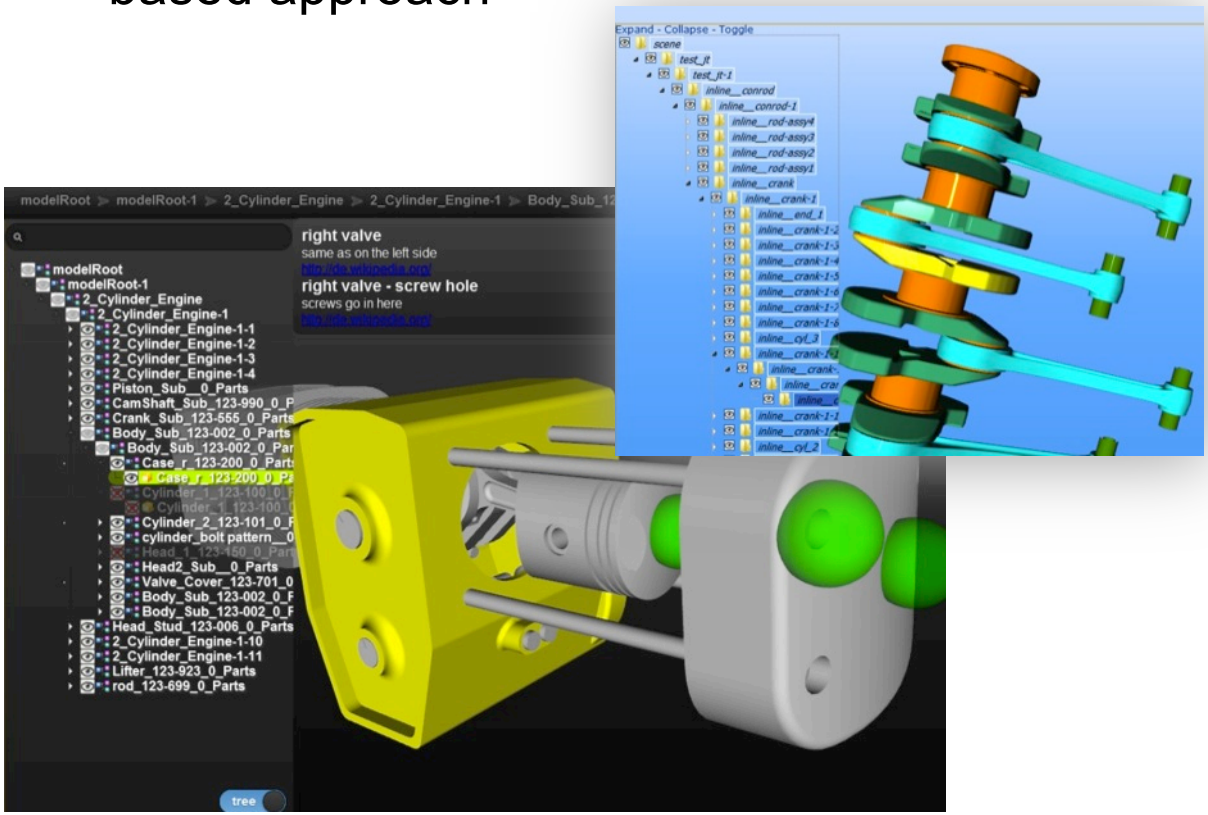
```
UIWebView *webView = self._webView;

id webDocumentView = [webView performSelector:@selector(_browserView)];
id backingWebView = [webDocumentView performSelector:@selector(webView)];

[backingWebView _setWebGLEnabled:YES];
```

Technologies – Ready for Implementation

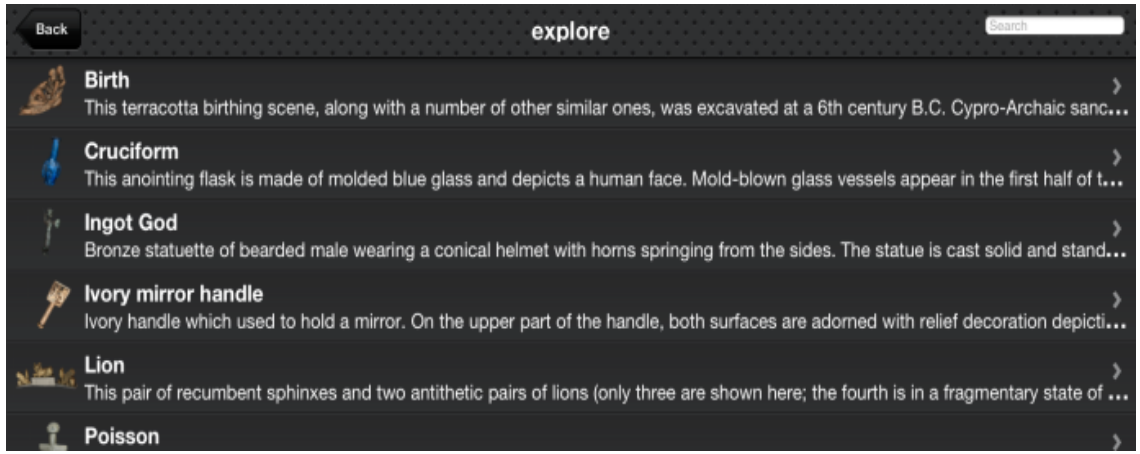
- We now have a…
  - JS framework to support the look and feel of native apps
  - Web Browser that supports X3DOM
- All further development is 100% the same for mobile devices and the webpage based approach



Implementation

Implementation – Loading Objects

- To browse to different objects, a list can be used containing hyperlinks
  - Each <a id="pre_objectID"> tag can contain an ID that is equal to the directory of the folder containing the object data.
  - Or the ID can be the key for a dictionary-like data structure that contains the paths to the objects
- The <Inline> tag is used to load an X3D model
  - JavaScript can be used to dynamically switch the URL and load an object
  - `document.getElementById("inlineID").setAttribute("url", "path/objectID/model.x3d");`
- When the link is clicked, a function is executed via the onclick event to switch the model as explained



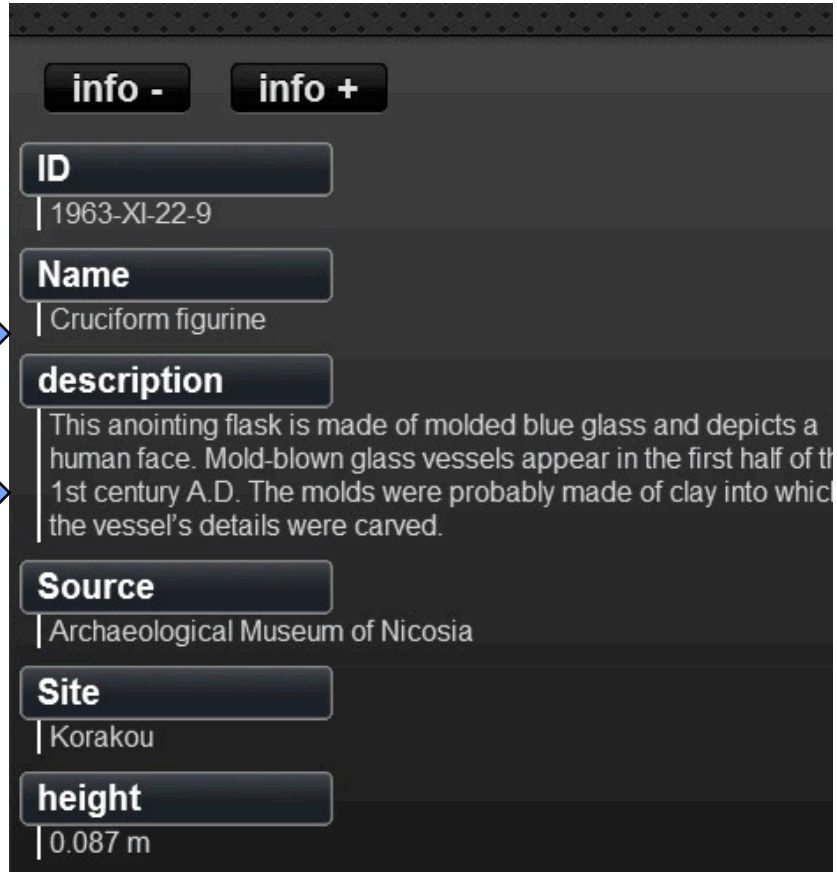
Implementation – Data Container

- All data corresponding with an object supplied via
  - JSON
  - XML
  - Database such as MySQL or CouchDB
- The data can hold
  - Traditional content such as images and the text that shall be displayed for metadata
  - 3D information (e.g. for annotation markers or suitable camera positions)
  - Application logic
  - …

```
{
    "title": "xxx",
    "model": "xxx",
    "URL": "folder/name.x3d",
    ...,
    "meta": {
        "ID": "test_1",
        "Name": "xxx",
        "description": "xxx",
        ...
    },
    "annotation": [
        {
            "pos": "48.0 55.5 6.5",
            "text": "Looks like a snake",
            "link": ".wikipedia.org"
            …
        },
        ...
    ]
}
```

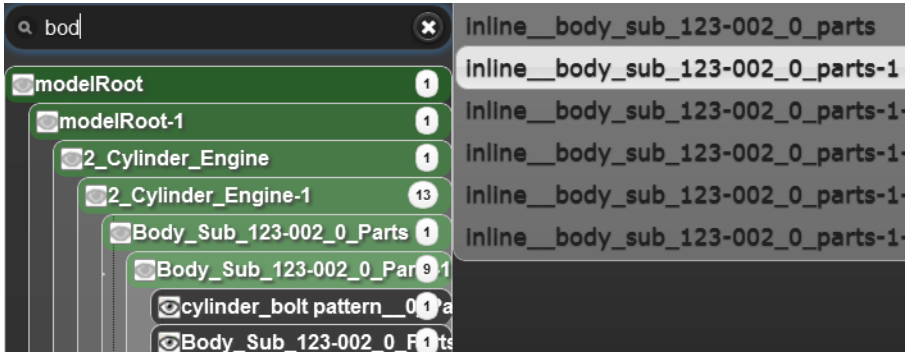
Implementation – Metadata


```
    <display_date />
  </temporal_coverage>
  <creation_date Recommended="September 2011" />
</Collection>
<Subject>
- <Record_Information>
    <ID Mandatory="" />
    <source />
    <creator>Nicola Amico</creator>
    <creation_date Recommended="26/03/2012" />
    <country Recommended="Cyprus" />
    <language Recommended="English (EN)" />
    <rights>STARC - The Cyprus Institute, Department of Antiquities of Cyprus, Archaeological
      Museum of Nicosia</rights>
  </Record_Information>
- <Appellation>
    <Name Mandatory="Cruciform figurine" />
    <ID Mandatory="1963-XI-22-9" />
  </Appellation>
  <Description Mandatory="This anointing flask is made of molded blue glass and depicts a human
    face. Mold-blown glass vessels appear in the first half of the 1st century A.D. The molds were
    probably made of clay into which the vessel's details were carved." />
- <temporal_coverage Recommended="Roman period (50 B.C. – 150 A.D.)">
  - <time_span>
      <start_Date Recommended="" />
      <end_Date Recommended="" />
    </time_span>
    <period_name Recommended="Roman period (50 B.C. – 150 A.D.)" />
    <display_date />
  </temporal_coverage>
  <Type Recommended="Archaeological Object" />
- <Measurement>
  - <perimeter>
      <type />
      <value />
      <unit />
    </perimeter>
  - <area>
```

Implementation – Metadata

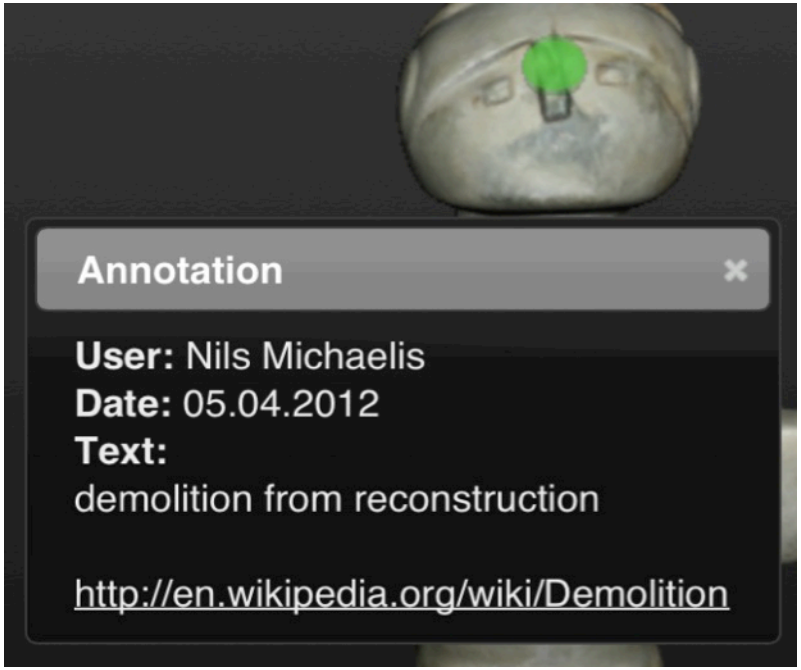
- Metadata can be displayed in lists and filtered by typing text into a search field while hiding wrong results via jQuery/ CSS3
- Since X3DOM integrates the model into HTML5 it is possible to merge the 3D content with textual and other information



- By default we just display the headings of each metadata
  - By tapping on them they expand
- There exist various display modes, e.g. full 3D area or all information extended

Implementation – Annotations

- If many people work together on one model, annotations are a good way for communication
  - They pass information from one person to another
  - They can be attached to a certain position in the model (→ green marker)
  - They can hold all kinds of information such as the author, creation date, text, hyperlinks, images, etc.
  - They can be stored permanently
- For loading, all mentioned data container formats (JSON, XML, …) are possible
- For persistent storing it is necessary to use a database, since JavaScript does not allow writing to disc



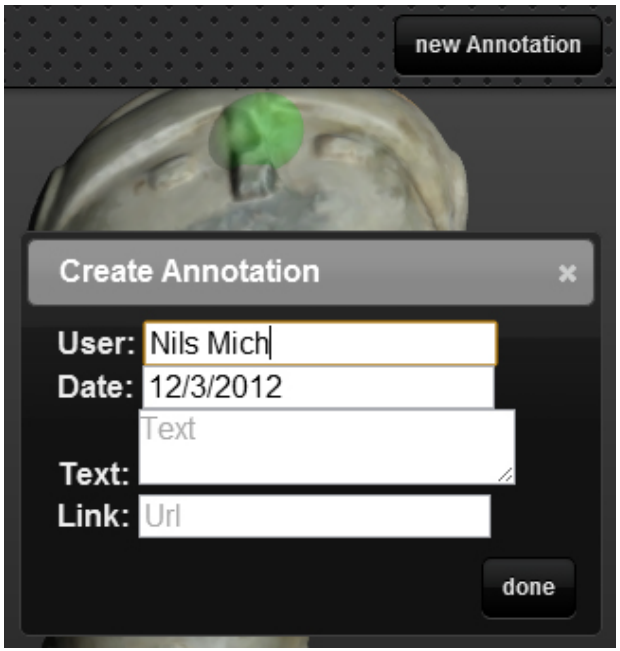
Implementation – Create Annotations

- First the user has to activate the annotation creation mode by tapping on the [new Annotation] button
- When tapping on the object, the position can be saved via:

```
function createAnnotation(event) {
        newAnnot.posX = event.worldX;
        newAnnot.posY = event.worldY;
        newAnnot.posZ = event.worldZ;
        …
```

(don’t forget to register the event listener)

- The [done] button saves the data in the form permanently in a database
  - A marker can be generated by first creating an empty <Group> with an ID
  - Then attach a geo to it, set the saved position and register event to show the annotation popup



Implementation – Multi-touch Navigation

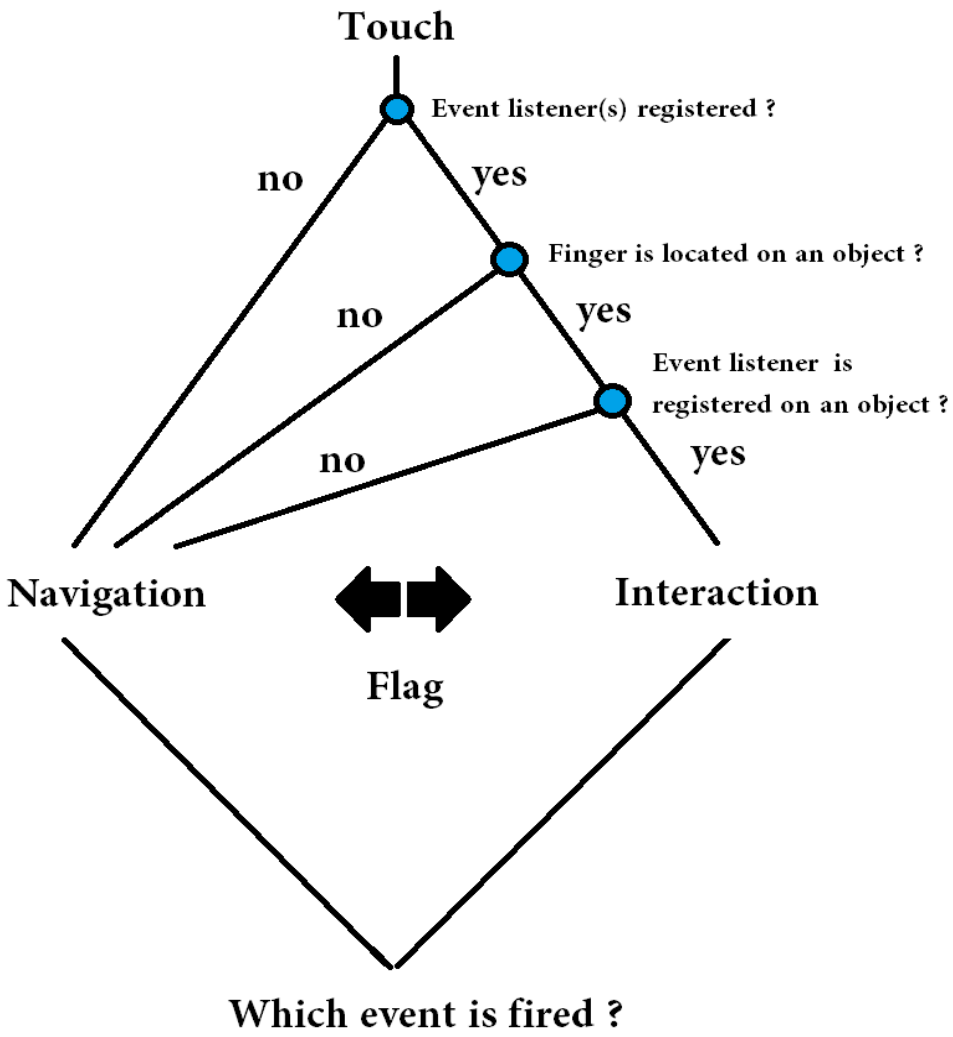
Swipe

Zoom

Rotate

Implementation – Multi-touch Events

1. Check if navigation or interaction is needed
2. Set flags for transition between navigation (per scene) and interaction (per object)
3. Identify individual touch points; check state transitions: press (0 → n), move (n → m), release (m → 0)



Results

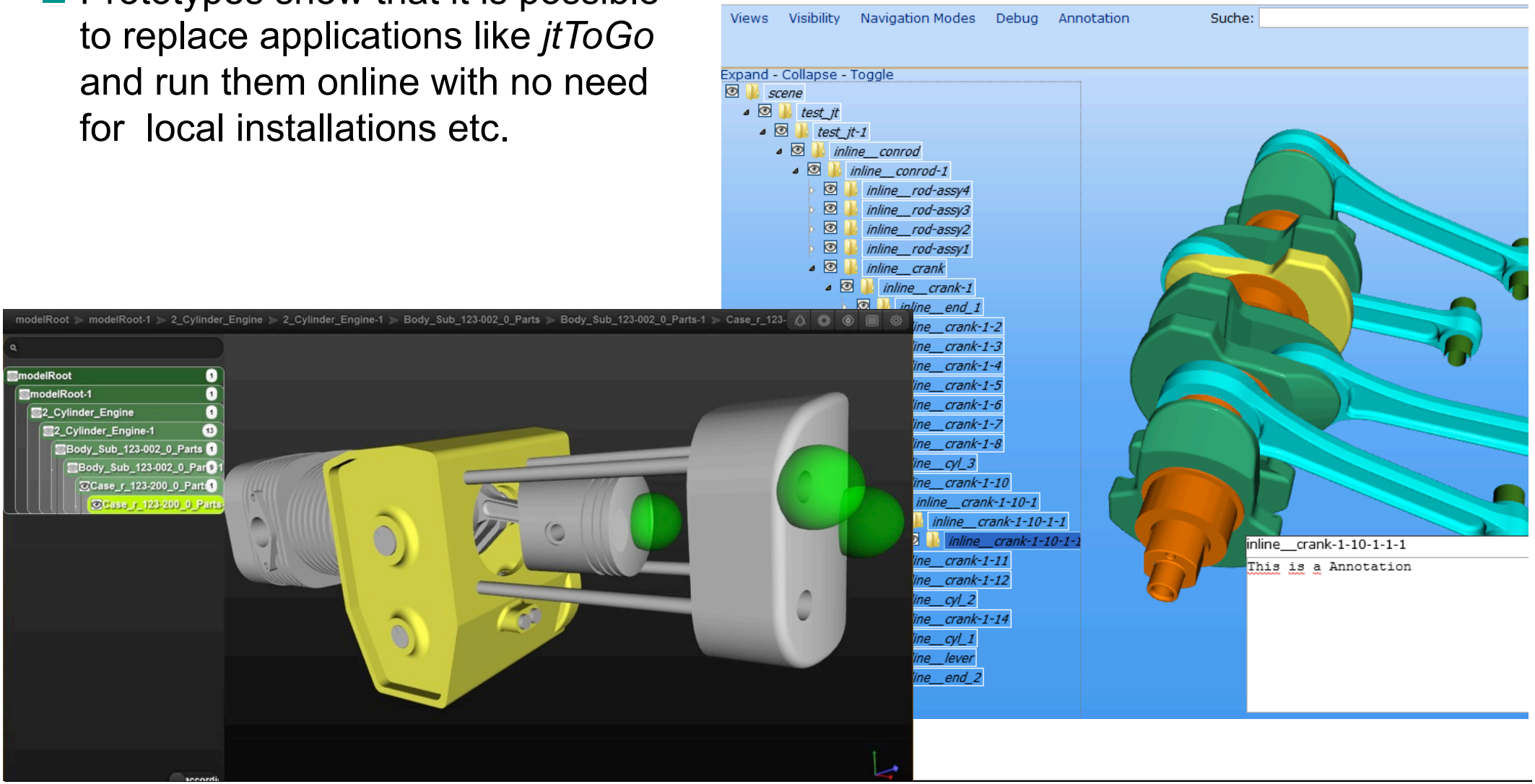
Performance

- Used mobile devices: iPad2/ iPad3
- It takes about 0.5 seconds to load a model with the following specifications from an external server:
- Original Model Data
  - Vertices: 714,916
  - **Triangles: 1,428,982**
  - **File size: 28.6 MB**

- After Optimization with Quadric Edge Collapse (in MeshLab)
  - Vertices: 100,305
  - **Triangles: 200,000**
  - **File size: 8.06 MB**
- After optimization with binary compress. (cp. BinaryGeo paper, Session 1)
  - **File size: 4.02 MB**
- No visual difference between original and compressed model on iPad
- Leads to a significant decrease of loading time, reduces memory & CPU overhead

Generality

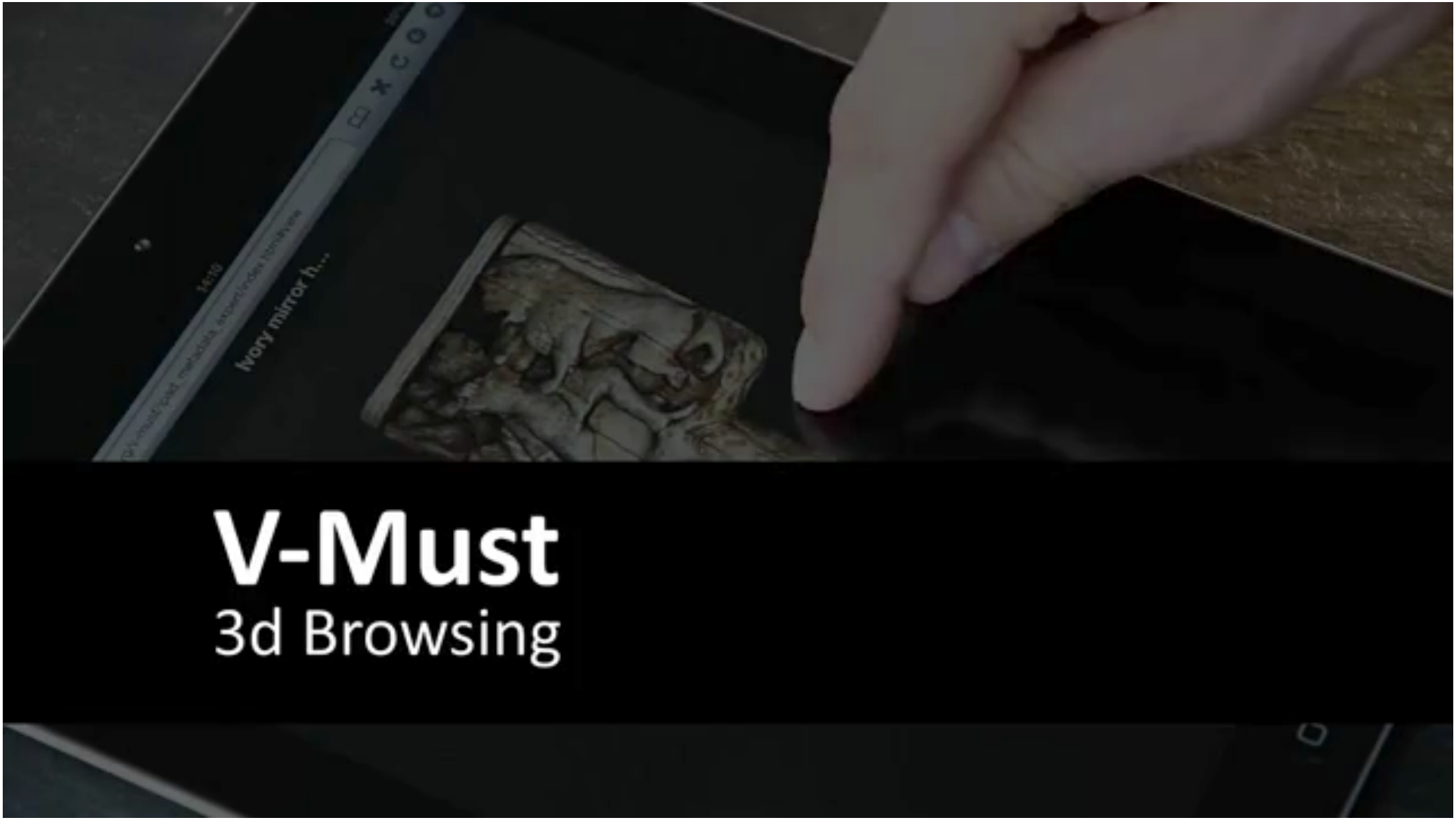
- Concept also works for other domains such as automotive industry (CAD viewer)
- Prototypes show that it is possible to replace applications like *jtToGo* and run them online with no need for local installations etc.



Conclusions

- Works in modern Web Browsers without plugins
  - Look and feel is comparable to native apps
  - Device independent
- One solution works on all kinds of devices
  - Saves development time, money and allows Web developers to directly start coding ☺
- All technologies are based upon open standards
  - Expectable that app remains accessible in future
- Very interactive and nice merging of 3D models with traditional media contents
- Smooth 3D interaction and short loading time
- Huge potential for annotations and communication
  - Distributed application model

Results



Thank you!

Questions?

examples.x3dom.org/v-must/
user/passwd: vmust/vm2012

www.x3dom.org
www.v-must.net