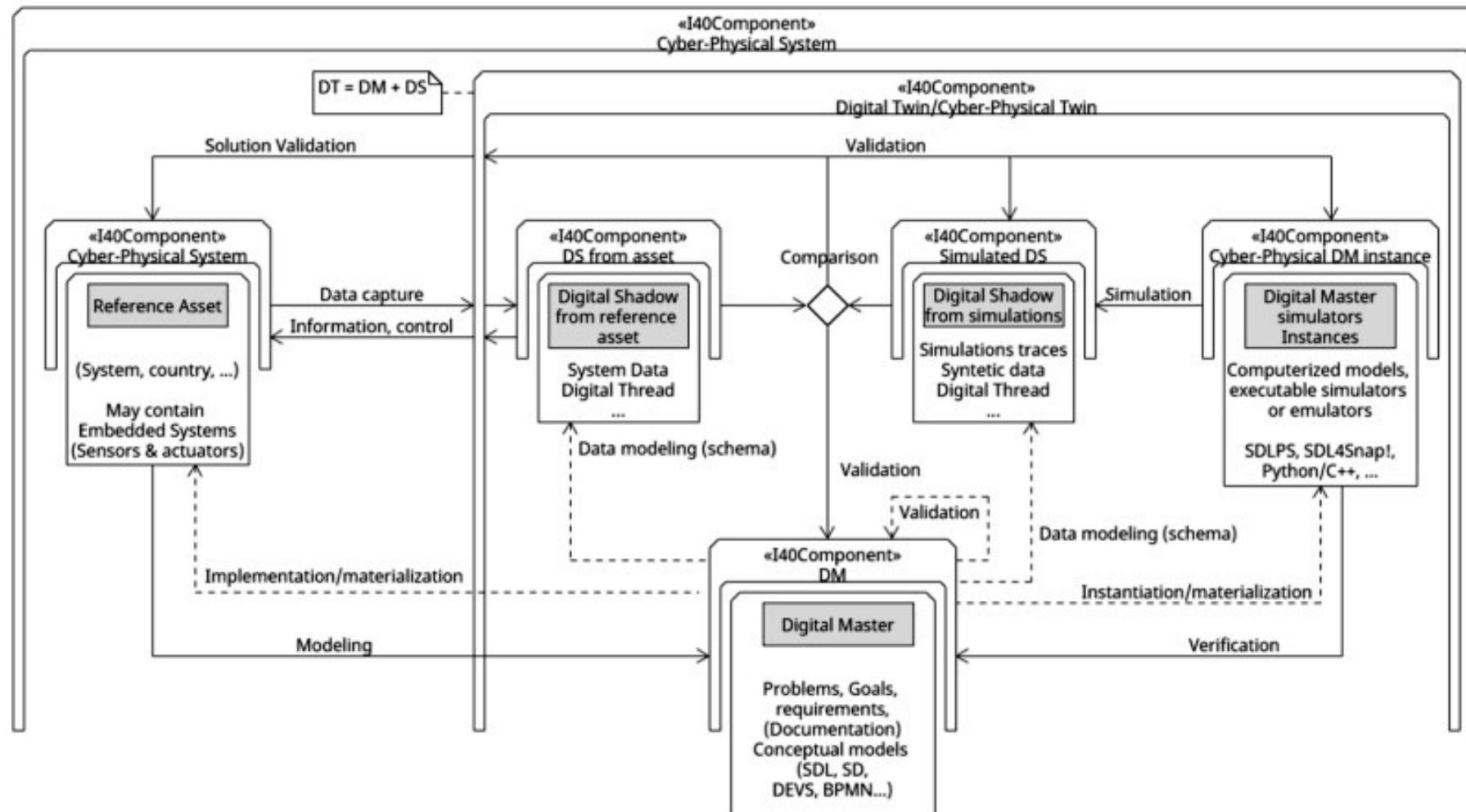


Industry 4.0 and Society 5.0 (II)

Digital Twins Components



Models 3D de terrenys

- Hi ha models 3D de terreny públicament accessibles als instituts estatals o autonòmics i als sites del govern.
- Divisió per sectors (quadrícules):
<https://datos.gob.es/es/catalogo/e00125901-spaigncuadriculamtn252015>
- Imatges aèries o de satèl·lit, ortofotos (arxius PNOA**) i models d'elevacions (arxius MDT02**): <https://centrodedescargas.cnig.es>
- Software open-source de muntatge de la imatge 3D: <https://qgis.org>
- Cal l'extensió QGIS2ThreeJS i exportar com a fitxer .glb
- <https://www.cursosqgis.com/actualizacion-del-plugin-qgis2threejs-de-qgis-para-trabajar-en-3d>

Google Sheets

Crear Google Sheet i anar a Extensions i Apps Script

The screenshot shows the Google Sheets Apps Script editor. The browser address bar displays the URL: `script.google.com/u/0/home/projects/1nQJqjDboiTzsvzB9qIKRvWo70J1agLZpxd_lhCDws6xwPula5dxYXCSI/edit`. The interface includes a top navigation bar with the 'Apps Script' logo, the name of the project 'Shared Sheet', and an 'Implementar' button. Below this is a toolbar with icons for undo, redo, save, and execution options like 'Ejecutar', 'Depuración', and 'doGet'. On the left, a sidebar contains a file explorer with 'Archivos' (containing 'Código.gs'), 'Bibliotecas', and 'Servicios'. The main area displays the JavaScript code for the `doGet` function, which handles requests to update or retrieve values from a specific sheet and cell.

```
1 function doGet(request) {
2   if (request.parameters.sheet && request.parameters.cell && request.parameters.value) {
3     let sheet = request.parameters.sheet;
4     let cell = request.parameters.cell;
5     let value = request.parameters.value;
6     let activeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheet);
7     activeSheet.getRange(cell).setValue(value);
8     return ContentService.createTextOutput('OK').setMimeType(ContentService.MimeType.TEXT);
9   } else if (request.parameters.sheet && request.parameters.cell) {
10    let sheet = request.parameters.sheet;
11    let cell = request.parameters.cell;
12    let activeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheet);
13    let val = activeSheet.getRange(cell).getValue();
14    return ContentService.createTextOutput(val).setMimeType(ContentService.MimeType.TEXT);
15  } else {
16    return ContentService.createTextOutput('KO').setMimeType(ContentService.MimeType.TEXT);
17  }
18 }
19
20
```

Google Sheet Webb App

```
function doGet(request) {  
  if (request.parameters.sheet && request.parameters.cell && request.parameters.value) {  
    let sheet = request.parameters.sheet;  
    let cell = request.parameters.cell;  
    let value = request.parameters.value;  
    let activeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheet);  
    activeSheet.getRange(cell).setValue(value);  
    return ContentService.createTextOutput('OK').setMimeType(ContentService.MimeType.TEXT);  
  } else if (request.parameters.sheet && request.parameters.cell) {  
    let sheet = request.parameters.sheet;  
    let cell = request.parameters.cell;  
    let activeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheet);  
    let val = activeSheet.getRange(cell).getValue();  
    return ContentService.createTextOutput(val).setMimeType(ContentService.MimeType.TEXT);  
  } else {  
    return ContentService.createTextOutput('KO').setMimeType(ContentService.MimeType.TEXT);  
  }  
}
```

Google Sheet Webb App

```
function doGet(request) {  
  if (request.parameters.sheet && request.parameters.cell && request.parameters.value) {  
    let sheet = request.parameters.sheet;  
    let cell = request.parameters.cell;  
    let value = request.parameters.value;  
    let activeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheet);  
    activeSheet.getRange(cell).setValue(value);  
    return ContentService.createTextOutput('OK').setMimeType(ContentService.MimeType.TEXT);  
  } else if (request.parameters.sheet && request.parameters.cell) {  
    let sheet = request.parameters.sheet;  
    let cell = request.parameters.cell;  
    let activeSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheet);  
    let val = activeSheet.getRange(cell).getValue();  
    return ContentService.createTextOutput(val).setMimeType(ContentService.MimeType.TEXT);  
  } else {  
    return ContentService.createTextOutput('KO').setMimeType(ContentService.MimeType.TEXT);  
  }  
}
```

Google Sheet Webb App

appscript.json

Project Settings -> Check the box labeled Show "appsscript.json" manifest file in editor.

```
{
  "timeZone": "Europe/Madrid",
  "dependencies": {},
  "exceptionLogging": "STACKDRIVER",
  "runtimeVersion": "V8",
  "oauthScopes": [
    "https://www.googleapis.com/auth/spreadsheets",
    "https://www.googleapis.com/auth/userinfo.email"
  ],
  "webapp": {
    "executeAs": "USER_DEPLOYING",
    "access": "ANYONE_ANONYMOUS"
  }
}
```

Publish -> Implementar (Deploy) as web app, Anyone has access

Google Sheet Web App URL

The screenshot shows the Google Apps Script editor interface. A modal dialog titled "Gestionar implementaciones" (Manage deployments) is open. The dialog has two main sections: "Activa" (Active) and "Archivadas" (Archived). The "Activa" section shows a deployment named "Sin título" (Untitled). The "Configuración" (Configuration) section for this deployment shows it is a "Aplicación web" (Web application) with a URL starting with "https://script.google.com/macros/s/". There is a "Copiar" (Copy) button next to the URL. Below the URL, there are two dropdown menus: "Ejecutar como" (Run as) set to "Usuario que accede a la aplicación web" (User who accesses the web application) and "Quién tiene acceso" (Who has access) set to "Cualquier usuario con una cuenta de Google" (Anyone with a Google account). A note states: "Los usuarios tendrán que autorizar la ejecución de la aplicación web con los datos de su cuenta." (Users will have to authorize the execution of the web application with their account data). At the bottom, there is a "Biblioteca" (Library) section with a message: "Para permitir que otros usuarios y grupos usen este proyecto como biblioteca, compártelo con ellos." (To allow other users and groups to use this project as a library, share it with them). The dialog has "Cancelar" (Cancel) and "Implementar" (Deploy) buttons at the bottom right.

Shared Sheet - Editor del proyecto

script.google.com/u/0/home/projects/1nQlqjDboiTzsvzB9qIKRvWo70J1agLZpxd_lhCDws6xwPula5dxyXCSl/edit

Apps Script

Gestionar implementaciones

Activa	Configuración
Sin título	Aplicación web URL https://script.google.com/macros/s/AKfycbzWj3ZfQWfKti6KaueMKQswbWFR7bcDe... Copiar Ejecutar como Usuario que accede a la aplicación web Los usuarios tendrán que autorizar la ejecución de la aplicación web con los datos de su cuenta. Quién tiene acceso Cualquier usuario con una cuenta de Google Biblioteca Para permitir que otros usuarios y grupos usen este proyecto como biblioteca, compártelo con ellos.

Cancelar Implementar

Digital Twins Prototyping

- <https://snap.berkeley.edu>

(Snap!)

- <https://xavierpi.com/proto>

(MQTT – Snap!)

Gràcies

Xavier Pi

xpi@enginyers.net