xDash : RAPID-PROTOTYPING OF CYBER-PHYSICAL SYSTEMS IN WEB BROWSER

MONGI BEN GAID, ABIR EL FEKI, BRUNO LETY AND NICOLAS PERNET
OUTLINE

- Context and motivations
- xDash: principles and features (+ demo)
- Some use-cases
- Conclusions and perspectives
Eco-driving coaching (for truck fleet fuel reduction)
Energy management (for hybrid or electric vehicle optimal battery use)
CO2 footprint and Total Cost of Ownership (TCO) of a car based on usage questionnaire (for helping new car purchase decision)
Cycle road network qualification using crowdsensing (for bike GPS app development)
Pollutant emissions estimation (how driving style impacts real-driving emissions)

Algorithms deployed as web-services
GENERAL CONTEXT : CONNECTED MOBILITY & WEB-SERVICES

- Connected vehicle: typical example of Cyber-physical system
- In our context, from a software engineering point of view
  - Vehicle-to-infrastructure (V2I) communications rely on web-services (over HTTP protocol)
  - Web-services follow microservices architectural principles (Service Oriented Architecture)
- Web-services bring new optimization parameters
  - At design phase
    - Open, community or company data access or query as web-service
    - Examples: road accidentology or pollution measures history, public bike stations, public transport theoretical fares ...
  - At operation phase
    - Live information about weather, traffic, routes, pollution is available from web-services ...
    - Use by real-time control algorithms (EMS, BMS, ECU) or on on-board display for driver ...
- Mobility as a Service paradigm shift

14th MODPROD Workshop 2020; Linköping; Sweden
WHAT IS A WEB-SERVICE? (IN CONNECTED MOBILITY CONTEXT)

- It is a (stateless) function (in the sense of compute science) running on a server
- Software interface is based on HTTP protocol over TCP/IP (core Internet Technology)
- Can be extended to run simulation as a web-service (adding state for long runs)

Client

POST /ecoroute HTTP/1.1
Host:ifpen.com
from=paris&to=nanterre

response

HTTP/1.1 200 OK
Date: Tue, 16 Apr 2019 15:30:59 GMT
Server: Apache/0.8.4
...
route={"latitudes":[48,754, ...}
Most programming languages or frameworks have libraries for calling web-services.

Many tools exist for the unit testing of web-services, such as Postman or SoapUI, but purely from software engineering point of view.

To the best of our knowledge, no cyber-physical system level tool exist for testing aggregation of web-services.

- With scientific or technologic oriented data interaction and visualization
- With connection to simulation & scientific computation (Python)
- With real-time execution capabilities
- Closest tool was freeboard.io

Need for a cyber-physical systems-level web-services aggregation and dashboarding tool.
CONCEPT

xDash allows technicians, scientists or engineers, not specialists in web technologies, to build their own web applications to answer these questions autonomously.

- Electrical mobility range
- Parkings in Paris
- Accidents in Hauts-de-Seine (92)
- Airparif map real-time
- Traffic incidents live
- STIF isochrones

Select your departure address, your electric vehicle brand, your departure date, the weather conditions, your battery state of charge and eventually your extra load. Get average and conservative estimation of your electrical mobility range. Get the energetic characteristics of your trip.

Select the Paris “arondissement” when an electric charger is needed. Paris open data about parkings is illustrated.

Location and description of road accidents in the Hauts-de-Seine (92) department from 2014 to 2016.

Select the date and the hour, and get pollution map of the Ile-de-France region (pollution index NO2, O3, PM10 and PM2.5).

Select a latitude/longitude box, and criticality value, to get updated about traffic incidents in that area.

Fill an address, and a maximal trip duration. STIF web services will compute the isochrones around that position.
BASE CONCEPTS

- **JSON variable**
  ```json
  {
    "x": [0, 1, 2, 3, 4],
    "y": [0, 1, 4, 9, 16],
    "rectangle": {
      "longueur": 1.23,
      "largeur": 5.5
    },
    "message": "Success"
  }
  ```

- **datasources** keyword in scripts
  - Specifies a data dependency
  - Specifies an execution order dependency

- **Datasources (operations)**
  - Are stateless
  - May have input, always have one output
  - Have a status
    - None: never executed
    - Pending: is being executed
    - OK: successful execution
    - Error: error reported when executed
  - May have an execution period

- **xDash keeps track of a workspace of JSON variables corresponding to latest datasources successful evaluation**
EXECUTION ENGINE

- Synchronous-reactive language
- Direct acyclic graph
  - Vertices: operations
  - Edges: data dependencies
- Data flows are JSON variables
- Execution rules
  -Datasource is executed if and only if all its predecessors completed their execution with status *OK*
  -Every time a datasource is successfully computed (status "OK"), it triggers the execution of all its successors
  -Graph execution is interrupted at datasources with status *Error*: their successors are not executed
- Similar concept as Simulink or Synchronous Modelica, but runs on a web-browser
SIMPLE EXAMPLE: GEOCODING VISUALIZATION

**Widgets**

**value**
1 rue de Pontoise, Paris, France

**address**
Type: JSON variable  
Category: datasource

**geocode**
Type: REST Web-service  
Category: datasource

```
{  
  "Response": {  
    "Metadata": {  
      "Timestamp": "2020-02-01T15:51:57.976+0000"  
    },  
    "View": {  
      "_type": "SearchResultsViewController",  
      "ViewId": 0,  
      "Result": {  
        "Relevance": 1,  
        "MatchLevel": "houseNumber",  
        "MatchQuality": {  
          "Country": 1,  
          "..."  
        }  
      }  
    }  
  }  
}  
```

**to GeoJSON**
Type: JS Script  
Category: datasource

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [  
          2.35255,  
          48.85051  
        ]  
      }  
    }  
  ]  
}  
```
DATASOURCE CATEGORIES

**Base functions**
- JSON Variable
- Web-service REST
- Python Script
- JS Script

**Mobility**
- Map-matching
- Trip conversion

**IoT & real-time**
- Clock
- MQTT
- Websockets
- Geolocation

**Transversal**
- FMI simulation

**Files**
- CSV Reader
- CSV Player
- Generic reader
RESPONSIVE DASHBOARD

- **Bootstrap** rows and columns

- Scaling methods

In editor ➔ Projection ➔ Keep proportions
SHARING APPLICATIONS

Sharing a dashboard in 3 clicks
TRUCK HYBRIDIZATION STUDY

- Main inputs: start and destination address, truck parameters, fuel & electricity prices ...
- Main outputs: cost of trip of conventional truck vs. hybrid truck
- Truck dynamic simulation models are FMUs generated from Amesim models

Aggregated web-services

geocoding \[\rightarrow\] trip \[\rightarrow\] Speed profile

Diesel truck FMU \[\rightarrow\] Hybrid truck FMU

Rapid prototype in xDash
TCO & CO2 ESTIMATION FOR CAR PURCHASE ADVISE

- Main inputs: home & work addresses, other usages, ownership duration, desired segment...
- Main outputs: TCO (€) and CO2 footprint (kg)...

Rapid prototype in xDash

General public site: https://jechangemavoiture.gouv.fr/

14th MODPROD Workshop 2020; Linköping; Sweden
Main inputs: vehicle parameters, trip, road conditions from test database ...

Main outputs: fuel consumption, CO2, CO, HC, NOx, PM, PM tires, PM brakes ...

Rapid prototype in xDash

Gecoair app

Digital twin debugging

14th MODPROD Workshop 2020; Linköping; Sweden
Conclusions

- xDash for
  - rapid-prototyping of CPS
  - demonstration
  - sharing, collaboration

- Web technologies have the potential of bringing “social” and “collaborative” features to scientific computing
- xDash supports both on cloud or on-premises deployments
- We are looking for beta-testers!

Perspectives

- xDash for AI & crowdsensing
- xDash will be publicly available soon for free use