modelsconf2018 / **artifact-evaluation**

Branch: master ▾ artifact-evaluation / kusmenko / **README.md** Find file Copy path

vonwenckstern added archive link d453ae1 2 days ago

2 contributors

```
147 lines (105 sloc)   10.3 KB
```

Link to draft accepted Models paper: http://www.se-rwth.de/materials/ema_compiler/MODELS_2018_paper_84.pdf

Link to GitLab projects including all subprojects: https://git.rwth-aachen.de/groups/monticore/EmbeddedMontiArc/

Backup-Link for the entire website (inclusive all benchmarks) is archived at: https://zenodo.org/record/1314370#.W09hvNIzY2w

Please use the archive link to download the entire website including all referred resources of this description. You only need to download the the large archive (ca. 7.5GB) and host it (e.g. with Xampp) on your local pc; then please adapt the links from `http://www.se-rwth.de/materials/XXX` to `localhost/xxx`

Official Supplementary Homepage for Models

- please visit http://www.se-rwth.de/materials/ema_compiler/
  - there you can browse online through our models we used for our measurements
  - there you can also execute the running `Spectral clusterer` example directly in the web browser (when it supports WebAssembly and JavaScript is activated)
  - this site contains videos to see the generator in actions with different middle-ware plugins

Execute Models via EmbeddedMontiArcStudio

EmbeddedMontiArc
Modeling Language for Cyber-Physical Systems
Download EmbeddedMontiArcStudio 1.7.5-beta
Portable App for Windows 64-bit

- test the EmbeddedMontiArc compiler in a user-friendly environment (a tutorial video how to use EmbeddedMontiArcStudio of an older version is available under: https://youtu.be/VTKSWwWp-kg?t=52s)
  i. Click at the Download Button (picture) above
  ii. Unpack the EmbeddedMontiArcStudio_1.7.5-beta.exe at flat hierarchy folder (e.g. `C:\`, but `C:\Users\vonwenckstern\Documents\models2018\papers\kusmenko' might be a too deep folder structure and it might then occur that not all files can be extracted due to Windows 10 restriction of a path length limit of 255 characters)
  iii. Open the folder `EMAStudio.11Jul-1309`
  iv. click at the `ide.bat` flie (the delivered JVM may ask for network permission, you can ignore the Window)
  v. an HTML page directing to `localhost` will open with the IDE
  vi. open the first project `AutoPilot` by clicking at the blue checkpoint button (see picture 1)
  vii. it will open the main component `Auotpilot.emam`
  viii. click at the `Execute Model` button (this will translate the model via our compiler infrastructure to native code; this takes a while as linking against the LAPLACK library needs some minutes)
  ix. a new browser tab with the MontiSim simulators opens
  x. when the world is loaded (you see a street and some green weeds), then please press in the top right at the `S1 Native` button (see picture 3) to let the car drive a first scenario
  xi. If the car does not move anymore, please go back to the `AutoPilot - Cloud9` tab

xii. Please click at the `Selection` button (see picture 4) to get back to the Project Selection page

xiii. Please open the `Clustering` project

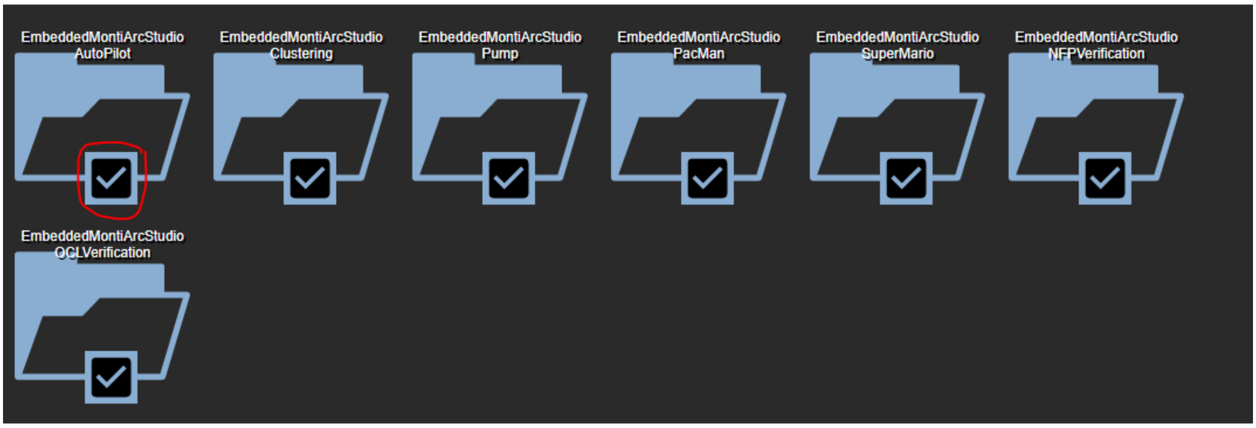
xiv. If the `SpectralCluster.emam` component is shown, please click at the `Execute Model` button (see picture 2), this will generate the code for our running example model

xv. If the compilation was successful, a new browser tab openes. Please drag one of the displayed images into the white drop-szone to get it clustered (see picture 5).

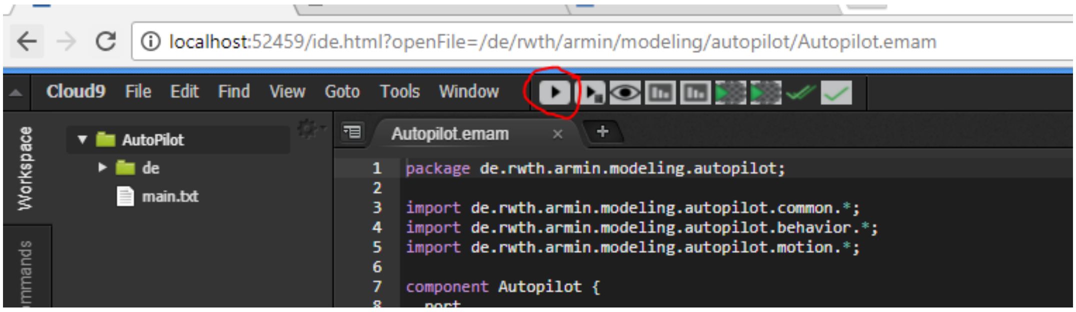
xvi. You will see a clustered black-white image as output.

xvii. Please click at the white-box to select your own image and let it cluster

picuture 1: (select `AutoPilot` project)

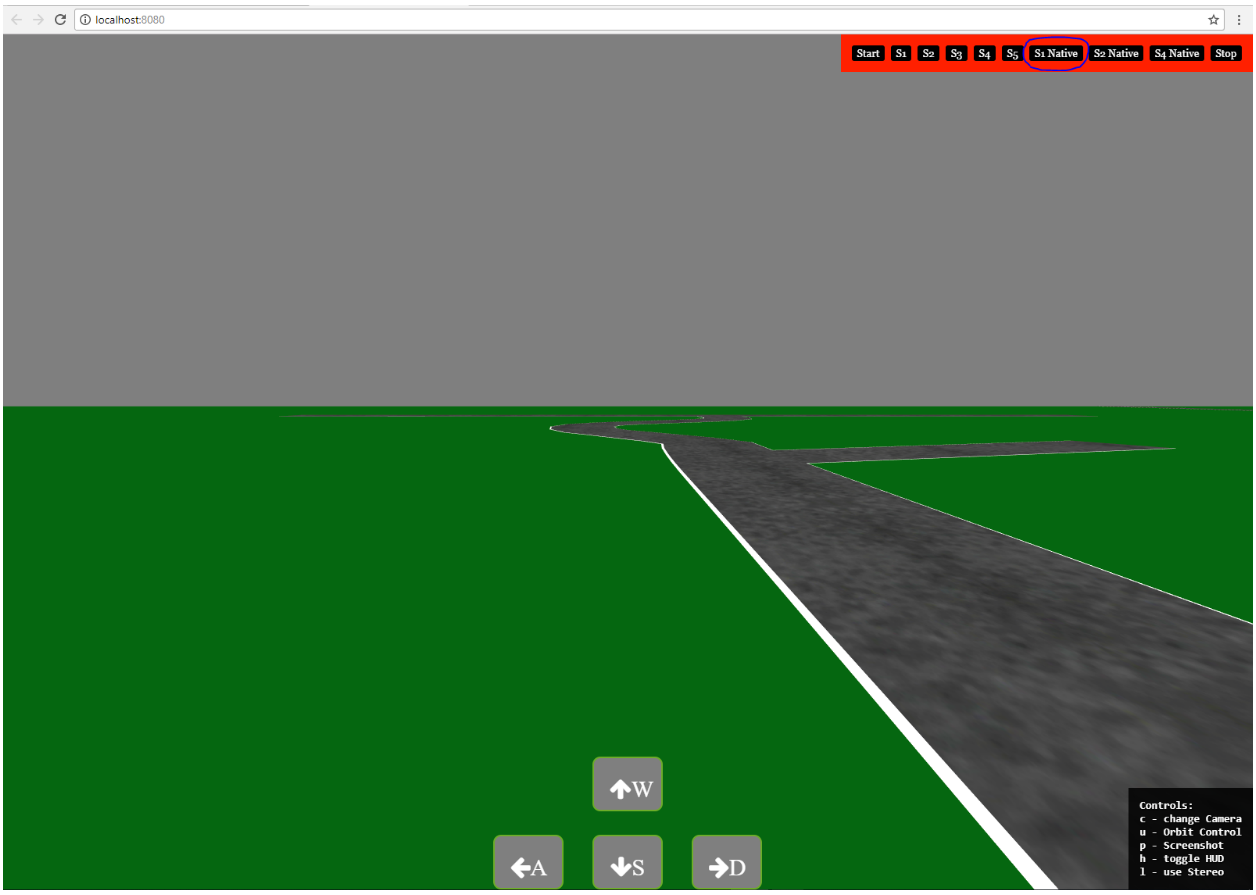

EmbeddedMontiArcStudio AutoPilot
EmbeddedMontiArcStudio Clustering
EmbeddedMontiArcStudio Pump
EmbeddedMontiArcStudio PacMan
EmbeddedMontiArcStudio SuperMario
EmbeddedMontiArcStudio NFPVerification
EmbeddedMontiArcStudio OCLVerification

picuture 2: (click at `Execute Model` button)

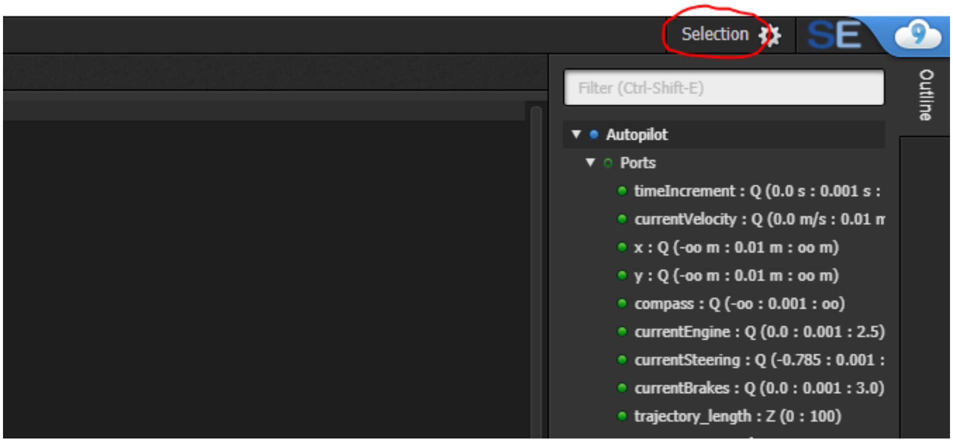


```
package de.rwth.armin.modeling.autopilot;

import de.rwth.armin.modeling.autopilot.common.*;
import de.rwth.armin.modeling.autopilot.behavior.*;
import de.rwth.armin.modeling.autopilot.motion.*;

component Autopilot {
  port
```

picuture 3: (start `S1 Native` scenario; you could also click at `S2 Native` or `S4 Native`)

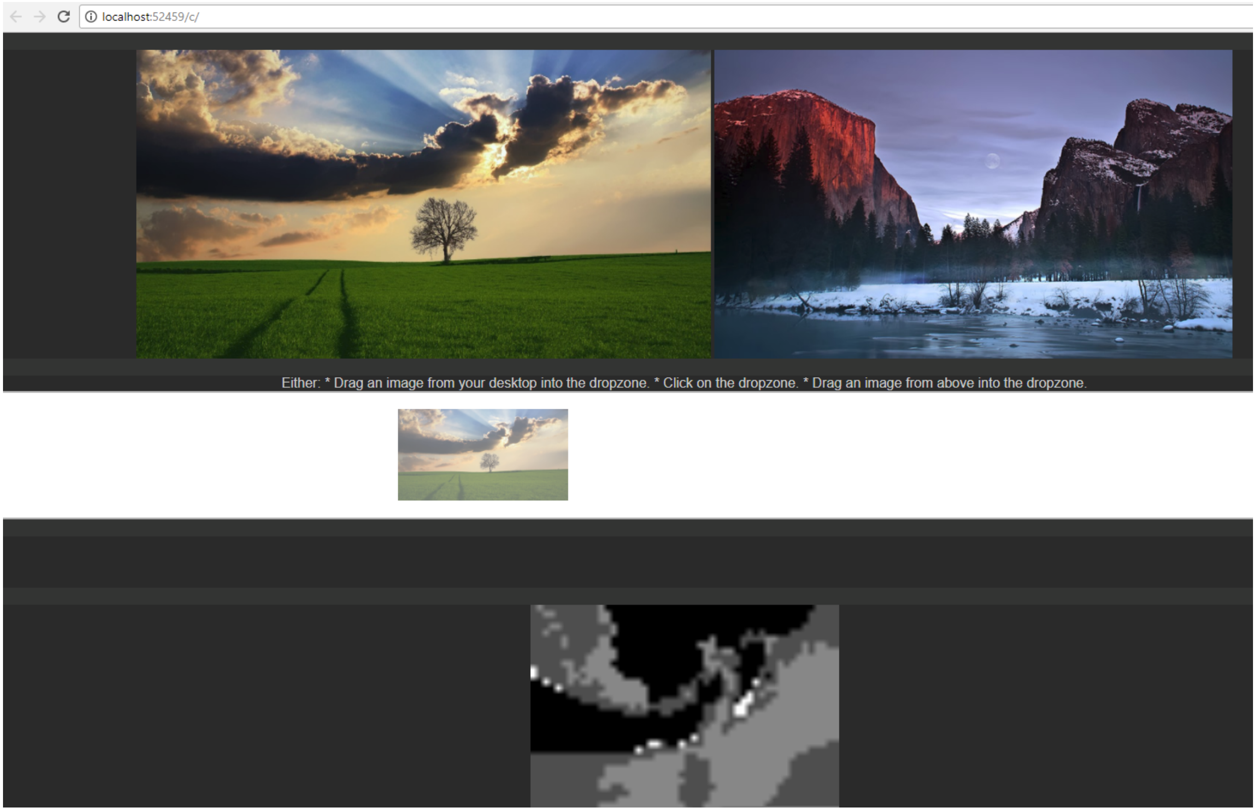


picuture 4: (click at `Selection` button)



```
▼ Autopilot
  ▼ Ports
      timeIncrement : Q (0.0 s : 0.001 s :
      currentVelocity : Q (0.0 m/s : 0.01 m
      x : Q (-oo m : 0.01 m : oo m)
      y : Q (-oo m : 0.01 m : oo m)
      compass : Q (-oo : 0.001 : oo)
      currentEngine : Q (0.0 : 0.001 : 2.5)
      currentSteering : Q (-0.785 : 0.001 :
      currentBrakes : Q (0.0 : 0.001 : 3.0)
      trajectory_length : Z (0 : 100)
```

picuture 5: (drag an image into the white drop-zone)



Either: * Drag an image from your desktop into the dropzone. * Click on the dropzone. * Drag an image from above into the dropzone.

Supplementary Material for Generator

- *Notice: **Only Win 10 64-bit** is officially supported by our tools. Everything that is needed for the generator should be included with it.*

Note that for all measurements only the execution times were considered

- **please download the zip file containing the native benchmark set and tests from http://www.se-rwth.de/materials/ema_compiler/nativeCompilerBenchmarkSuite.zip**
- Files are inside the `generator` folder.
  - In the `generator` folder is a `generateAll.bat` .
    - This file uses the emam2cpp generator to create executable cpp code out of the models that are stored in the `model` folder.
    - This code can then be copied into the `benchmark/src` folder.
  - (The `benchmark/src` folder does also contain pre-generated code if you do not want to generate everything from scratch.)
  - After this you can just execute the `measureAll.bat` (in the `generator/benchmark` folder). (Note that this process may take some time.)
    - If the command-line asks to replace files, please enter `Yes` or `All`
    - This will then automatically compile the C++ code that is inside of the `src` folder and execute it afterward.
    - The resulting execution times are then stored in `resultsBLAS` and `resultsOpenBLAS` .
    - Taking a look inside of the resultsBLAS folder, one can see the folders `Clusterer` , `ClustererNoKMeans` , `ClustererNoKMeansOpt` , `ClustererOpt` , and `MatrixModifier` .
  - Inside of these folders are files that contain the execution times.
    - `timesL0.txt` contains the execution time for when no optimizations were enabled.
    - `timesL1.txt` contains the execution time for when algebraic optimizations were enabled.
    - `timesL2.txt` contains the execution time for when threading optimizations were enabled.
    - `timesL3.txt` contains the execution time for when algebraic&threading optimizations were enabled.

Webassembly (wasm)

- **please download the zip file containing the webassembly benchmark set and tests from http://www.se-rwth.de/materials/ema_compiler/wasm.zip**
- Files are inside the `wasm` folder.

  - You just have to execute the `compile_allClustererLight.bat` .
  - If you want to compile the files to use the ClustererManOpt use `compile_allClustererManOpt.bat`
  - If you want to compile the files to use the Clusterer no opt use `compile_allClustererNoOpt.bat`
  - (The MatrixModifier will be compiled in all cases the same way)
  - The tool will open the portable chrome version with the relevant webpages that use the generated webassembly files.
  - Before pressing the execute model on a page it is advised to open the javascript console. *(Ctrl + Shift + J in google chrome)*
  - So you can see the execution time, in case the website crashes when it tries to fill the fields with all output numbers.
- The execution can be done without input values.
  - As in C++ the matrix values are what was stored in the allocated ram before.(So more or less pseudo-random)
  - You can set matrix values as input. In `\wasm\node\benchmark_sites` a `matrixModifierInputValues.html` is present.
  - You can use this to execute it with different values. (Opened by accessing `127.0.0.1:8080/matrixModifierInputValues.html`)
  - However, this will take a very long time, as the javascript to c++ value conversion is extremely slow. (The other webpages can also be changed to accept input values again, by uncommenting the `setXXX` methods in the execute function) (You have to execute a `compile_*.bat` again so the changes take effect and a node server is running)
- Clusterer: A nice cluster website can also be accessed at https://embeddedmontiarc.github.io/ClusterFiddle/index.html. There you can directly upload or select images for clustering. And see the result once it is done.(No time measurements)
- A portable version of chrome is inside of the folder. For some reason the objectDetector4.html example does not work in the portable version of chrome on one of the tested systems. However, the system installed(non-portable) version of chrome worked in this case.

Java Projects

- **please download the zip file containing Java Project http://www.se-rwth.de/materials/ema_compiler/java.zip**
- Files are inside the `java` folder.
- For opening these projects IntelliJ is required.
- Only the main method needs to be executed. It will then report the execution time in the console.

MathJS

- **please download the zip file containing mathJS files for the workbench from http://www.se-rwth.de/materials/ema_compiler/mathjs.zip**
- Files are inside the `mathjs` folder.
- The MathJS code for the `ClustererNoKMeansOpt` example is inside of the `ClustererNoKMeansOpt.txt` .
- You can copy it into a JSFiddle or something similar for execution.
- The MathJS code for the `MatrixModifier` example is inside of the `MatrixModifier.txt` . You can copy it into a JSFiddle or something similar for execution.

Required for execution:

- We used JSFiddle:
  - ClustererNoKMeansOpt: https://jsfiddle.net/m1dn7z0v/152/
  - MatrixModifier: https://jsfiddle.net/m1dn7z0v/154/
  - *Note: Site not responding messages can be ignored, the execution is continued non the less.*

Matlab

- **please download the zip file containing Matlab files for the workbench from http://www.se-rwth.de/materials/ema_compiler/Matlab.zip**
- Files are inside the `Matlab` folder.
- Required for execution: **MATLAB R2018a(64-bit)**

OpenModelica

- **please download the zip file containing Matlab files for the workbench from** http://www.se-rwth.de/materials/ema_compiler/OpenModelica.zip
- Files are inside the `OpenModelica` folder
- Required for execution: **OpenModelica v1.12.0 (64-bit)**

Simulink

- **please download the zip file containing Matlab files for the workbench from** http://www.se-rwth.de/materials/ema_compiler/Simulink.zip
- Files are inside the `Simulink` folder.
- Required for execution: **MATLAB R2018a(64-bit)**