SOFA Technical Committee #9
Report

Date: 28/04/2020 - 30/04/2020
Location: Online

Please find here the slides of all presentations.

Attendees

- Yinoussa Adagolodjo, Inria Lille, France
- Abdelrahman Alkhodary, Bahçeşehir University, Turkey
- Jérémie Allard, InSimo, France
- Rémi Bessard-Duparc, Freelancer, France
- Jean-Nicolas Brunet, Inria Strasbourg, France
- Stéphane Cotin, Inria Strasbourg, France
- Christian Duriez, Inria Lille, France
- Stefan Escaida Navarro, Inria Lille, France
- Ruiliang Gao, University of Florida, USA
- Thierry Gaugry, Inria Rennes, France
- Olivier Goury, Inria Lille, France
- Ryadh Haferssas, Inria Strasbourg, France
- Shiyao Lin, University College Cork, Ireland
- Bruno Marques, Inria Lille, France
- Sergei Nikolaev, Inria Strasbourg, France
- Guillaume Paran, Consortium staff, France
- Erik Pernod, InfinyTech3D, France
- Tim Pusch, Fraunhofer IPA, Germany
- Frederick Roy, Freelancer, France
- Hugo Talbot, Consortium staff, France
Synthetic agenda of STC#9

Day 1 - Tuesday 28th April

09:00 - Opening STC#9

09:10 - Consortium activity

09:30 - SOFA activity (part 1)

10:30 - Coffee break

11:00 - SOFA activity (part 2)

12:00 - Lunch

13:30 - Defining v20.12 roadmap

17:30 - Validating v20.12 roadmap

17:45 - Introduction of coding sprint topics

18:00 - End of Day 1

Day 2 - Wednesday 29th April

09:00 - Coding sprint

18:00 - End of Day 2

Day 3 - Thursday 30th April

09:00 - Coding sprint

17:00 - Closing STC#9
Reports and updates

SOFA report
You will find all information regarding the report on SOFA, its activity and the consortium activity within the STC#9 slides.

Roadmap task progress
All presentations and slides are publicly available here.

SofaPython3 - Damien Marchal
Further to the efforts put in 2019 into the SofaPython3 project, the plugin is now ready to be used and tested on a larger scale. For more visibility, SofaPython3 has been made an official repository of the SOFA Framework GitHub organization.
The SofaPython3 project still needs some work on the CMake files. Moreover, SofaPython3 will be added as an external plugin in SOFA (applications/plugins) and the few python examples will be upgraded for SofaPython3. Finally, SofaPython3 will have to be added to the continuous integration with proper tests.

The SofaPython(2.7) is being prepared for extraction for the SOFA source repository. The code will remain available as a separated project in the SOFA Framework GitHub organization.

Important:
- v20.06 release is the last release embedding and activating by default the SofaPython(2.7) plugin. It is advised to switch your scenes to Python3. A documentation will be written to guide you during this transition period.
- SofaPython3 and SofaPython(2.7) can not be used simultaneously. On Windows, runtime errors occur when using both plugins. These will not be considered since SofaPython(2.7) will not be supported after v20.06

→ See the dedicated presentation (slides 20-24) for details.

SofaQtQuick - Bruno Marques
Since last STC#8 (late November 2020), SofaQtQuick still undergoes some runtime instability. But it is now getting ready to be alpha-tested. This work will be pursued for sure over 2020 & 2021, so do not hesitate to get involved.
This plugin is working on Linux, but would need alpha-testing on Windows (already a bit by one or two engineers). This work includes cutting edge dependencies.

→ See the dedicated video presentation for details.
Do not hesitate to contact us for accessing the full SofaQtQuick video recorded by Bruno Marques.

Data updates - Hugo Talbot
This roadmap topic was discussed and decided at last STC. Discussions continued about how the simulation flow could be organized using a Data-oriented pipeline. The brainstorming got quickly
stuck on the topic of matrix resolution (including linear system + constraint resolution). This important topic needs first to identify most of the possible simulation scenarii using SOFA.

Eventually, the idea would be to reuse the API of DDGNode for structuring the simulation pipeline (using Data graph). Unfortunately, not enough people were involved. This might not be appropriate as a roadmap task, but rather than a discussion topic to pursue in the coming months. More in the section “System resolution: towards full build system” below.

→ See the dedicated presentation (slide 19) for details.

Out-of-roadmap important contributions

Guillaume: Environment and Tools

The graph above is an overview of SOFA environment and online tools. It is composed of 3 main parts: the continuous integration running on Jenkins, the project repository on GitHub, the website on a self-hosted server. The goal of the presentation was to explain how these 3 parts are connected together through a common use case: the creation of a pull-request.

When a pull-request is created, a signal goes from the repository to the continuous integration to trigger a build. This build consists in executing a series of scripts on several virtual machines with different parameters. In a pull-request case, builds are executed on 1 Ubuntu, 1 CentOS, 1 MacOS and 1 Windows machines. The default parameters are: plugins ON, unit-tests ON, scene-tests OFF, regression-tests OFF.

During the build, on each virtual machine, the scripts send regular updates to the “GitHub status checks” on the pull-request and to the “Dashboard” on SOFA Website.
The main change done on the continuous integration in the last 6 month is the installation of Docker on all Linux machines. The goal is to ease the maintenance of these machines. With Docker, the configuration of the virtual machine itself is no longer important. What is important is the description of the Docker image (a VM in the VM) that will be used as a container to run the build. SOFA Docker images (the guest VM) are defined by files (Dockerfile) stored in the sofa-framework/ci repository on GitHub. They are automatically generated at every commit. Unfortunately it is not possible to use Docker on “Inria CI” Windows and MacOS virtual machines because of technical settings that are out of our reach. Thus, we are now in a mixed solution with nice/Dockerized/easy-to-maintain configurations for Ubuntu and CentOS and raw/hard-to-maintain configurations for Windows and MacOS.

The second issue discussed during this presentation was that for each build of each configuration, the scripts are called sequentially. We are not yet able to parallelize unit-tests, scene-tests and regression-tests. Here is a proposal of a nice improvement of the build process for the future of SOFA continuous integration:

1. build SOFA without plugins + generate binaries
2. Trigger, in parallel:
   a. get SOFA binaries and run SOFA unit-tests
   b. get SOFA binaries and run SOFA scene-tests
   c. get SOFA binaries and run SOFA regression-tests
3. Trigger, in parallel, for each plugin
   a. get SOFA binaries and build plugin + generate plugin binaries
   b. get SOFA and plugin binaries and run plugin unit-tests
   c. get SOFA and plugin binaries and run plugin scene-tests
   d. get SOFA and plugin binaries and run plugin regression-tests

Many needs arising, Guillaume works on this but if additional needs, we need additional involvement. This could therefore appear on the roadmap. Common tasks could be defined without hesitating listing technical needs that could be brought to Inria management, InriaSoft and CI staff.

Fred: Modularization

Today in SOFA, packages and modules are not the same thing. There are a few packages (SofaGeneral, SofaMisc, …) and a lot of modules in them (SofaGeneralEngine, SofaMiscFem, …). The main idea behind Modularization is to make every module (aka library) a package and thus to be able to call the CMake functions “find_package” and “target_link_library” on every module. The goal of this project is to permit a complete modularity in SOFA by making the activation/deactivation of every module possible.

In a second step, current packages will disappear and new packages will be created to reorganize the modules in a more user-friendly way: by type (Collisions, Forcefields, Visualization, …) instead of by depth (Base, Common, Advanced, …). This work is called SOFA-NG.

In the end, it will be possible to depend on a specific module (thanks to Modularization) or on a set of modules of the same type (thanks to SOFA-NG).

The presentation highlighted the potential risks of the Modularization project: no one will have the same definition of SOFA. To tackle this issue, the SOFA Consortium must clearly define the official version of SOFA as a set of supported libraries. CMake configuration will take more time

→ See the dedicated presentation for details.
Yinoussa: Remote simulations + Cosserat plugin

Developed by Yinoussa from the Defrost team, a remote solution to launch multiple simulations on a cluster of machines is in its final state. The project, coupled with the power of Grid5000, is used mainly for machine learning and shape optimization. Technically, it is based on Docker and linked to a web dashboard used to manage, launch, and get the results (text and video) of an experiment. This project will be released soon. Stay tuned!

→ See the dedicated presentation for details.

Erik: Recent SOFA-Unity updates

The idea of this project is not to create a fancy GUI for SOFA but to integrate SOFA as a physics engine in Unity. It permits to benefit from both the clean and easy-to-obtain visualization of Unity, and the accurate physics of SOFA.

In his presentation, Erik showed the communication pipeline between SOFA and Unity through his private plugins SofaAdvancedPhysicsAPI and SofaAPAPI.

→ See the dedicated presentation and video demonstration for details.

Roundtable

After the usual Roadmap and Out-of-roadmap reports, a quick roundtable was done to get an overview of the whole community activity.

Yinoussa Adagolodjo
Cosserat plugin is almost ready.
Future: Add the dynamic part for Cosserat model. Documentation, web page being setup.

Abdelrahman Alkhodary
Student in Turkey, starting with SOFA, building (issues), using SoftRobots, using ROS.

Jérémie Allard
Interested to hear about all new devs in SOFA, even if not always agree but it’s good to see things getting more stable.

Jean-Nicolas Brunet
Past: remove embedded Eigen, now working with the latest version of Eigen, working with SofaPython3.
Future: last steps towards releasing PhD plugin (Caribou): work on hyperelastic forcefield, finish documentation, polish (nearly stable). 4 months left for thesis writing.

Stéphane Cotin
Actively working on machine learning and convolutional neural networks.
Could be open-source but API is changing too much at the moment.
Integrating PyTorch in SOFA (through SofaPython3).
Looking at shape optimization, implementation in SOFA will start after September.
New idea (wish): we need to be able to write equations at a very high level (like in Fenics).
**Christian Duriez**  
Many devs going on. Setting priorities is key.  
The most important is to complete ongoing work: SofaPython3, SofaQtQuick.  
New ideas: visitor-less simulation, redesign the build of the mechanical system.  
Find new people, new way of financing people.

**Stefan Escaida Navarro**  
Future: still a few months as PostDoc. Interested in SofaPython3, applied for Assistant Pr. in Chile. Using GMesh for parametric modeling.  
Automatic differentiation in collaboration with Mimesis.

**Ruiliang Gao**  
Future: working with Jorg Peters, modeling plasticity in Hexa models, cutting (dynamic topologies).

**Thierry Gaugry**  
Past: very few time to work on SOFA these last months. See #919.  
Future: in a near future he will not have much time on SOFA, maybe get a bit involved in SofaPython3

**Olivier Goury**  
Past: extension of his plugin Model Order Reduction (MOR) to handle a lot of contacts, reduction of contact forces (lambda) and self collision, first alpha version.  
Future: continue MOR extension, collaborate with origami artist (simulating this in SOFA).

**Ryadh Hafferssas**  
Interested in collision/contact simulation.  
Future: working version of contact algorithms in collaboration with InSimo. Constraint resolution through direct method (and not iterative ones like Gauss Seidel).

**Bruno Marques**  
Ending contract end of October. Still working on SofaQtQuick, Data updates and SofaPython3.  
Needs CI for SofaPython3 and SofaQtQuick.  
→ SofaQtQuick should be on the v20.12 roadmap (risk of giving up).

**Sergei Nikolaev**  
Ending PhD.  
Using Optimus plugin: estimation of mechanical parameters (a release could be done in the coming months).

**Guillaume Paran**  
Future: Improve continuous deployment, compile plugins out-of-SOFA.

**Erik Pernod**  
Working on homogenization of ForceField (and other) API.  
Unify API for topology (containers).
Frederick Roy
Past: Modularization. Ultra-sound plugin being implemented privately, could be a very useful tool and a point of collaboration.
Future: Modularization. Visualization: SofaQt3d is not the right way to go, must find another way.

Hugo Talbot
Future: Python3 configuration, get to understand more deeply the bindings, help on testing.

Shiyao Lin
Shiyao did not use SOFA yet. Trying to apply it to solve the motion of continuum robots.
SOFA v20.12 roadmap

Here is the roadmap for the next 6 months that has been discussed and approved by the SOFA Consortium members.

Packaging

**Description**
Improving the way we package and distribute SOFA has always been a background task for the Consortium Staff. These days, the need for this work is not anymore only around SOFA release management but also in the research teams. They want to create binaries and to be able to easily distribute their plugins with dependencies. Thus, we have to provide tools and good practices to make sure everyone can package SOFA and plugins in a convenient way.

The Modularization and SOFA-NG works will help a lot achieving that objective. That is why we joined them to this roadmap big task.

<table>
<thead>
<tr>
<th>“Packaging” v20.12 roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>● CI / CD</td>
</tr>
<tr>
<td>1. Be able to generate custom binaries on demand (use SPM if possible)</td>
</tr>
<tr>
<td>2. Make every plugin buildable out-of-SOFA</td>
</tr>
<tr>
<td>a. Decide first <a href="#">which plugins to keep in SOFA sources</a> (officially supported)</td>
</tr>
<tr>
<td>b. Make sure no more GPL plugins</td>
</tr>
<tr>
<td>c. Don’t forget plugins from teams</td>
</tr>
<tr>
<td>d. Define a clear way to list all plugins (SPM?)</td>
</tr>
<tr>
<td>● Modularization</td>
</tr>
<tr>
<td>1. non breaking modularization: SofaAdvanced (PR #1344)</td>
</tr>
<tr>
<td>2. non breaking modularization: SofaMisc (WIP PR #1307)</td>
</tr>
<tr>
<td>3. non breaking modularization: SofaGeneral</td>
</tr>
<tr>
<td>4. non breaking modularization: SofaCommon</td>
</tr>
<tr>
<td>5. non breaking modularization: SofaBase</td>
</tr>
<tr>
<td>● Python modules (SofaPython3)</td>
</tr>
<tr>
<td>1. Write detailed specs: buildtime part / runtime part</td>
</tr>
<tr>
<td>2. Improve resource/dependency management for Python modules</td>
</tr>
<tr>
<td>3. Define good practices for writing Python modules</td>
</tr>
</tbody>
</table>

More information
Main contacts: Guillaume Paran, Frederick Roy, Jean-Nicolas Brunet
Gitter chat room: [https://gitter.im/sofa-framework/sofa](https://gitter.im/sofa-framework/sofa)
SofaPython3

Description
Now integration of SofaPython3 must be polished and extraction of SofaPython(2.7) must be performed carefully in order to offer a smooth migration from SofaPython to SofaPython3.

“SofaPython3” v20.12 roadmap
1. Clean CMakeLists
   a. Definition of the plugin (made up of several subdirectories) using SofaMacros.cmake
   b. Install rules using SofaMacros.cmake
2. Add as an external plugin in SOFA using sofa_add_subdirectory_external()
3. Update example scenes and adapt CMake in SOFA
4. Write the migration documentation (on a scene point of view)

“SofaPython3” v20.12 roadmap
5. Test coverage
6. Doc on in which case / how to create your own bindings in a plugin
7. Fix and cleaning

More information
Main contacts: Damien Marchal, Jean-Nicolas Brunet, Bruno Marques, Hugo Talbot
Gitter chat room: https://gitter.im/sofa-framework/SofaPython3

SofaQtQuick

Description
SofaQtQuick is the project of a new GUI for running simulations. Replacing runSofa by something more flexible and user friendly has been discussed for a long time. The project, now in alpha state, aims principally to go beta for v20.12.

“SofaQtQuick” v20.12 roadmap
1. Gather people: compile and install it, use it
2. Getting SOFA consortium testing and providing feedback
3. Figuring out the stability issues
4. Simplify install requirements
5. Writing documentation
6. Adding a meshing feature (GMesh / CGAL)
7. Go from alpha state to beta state: something stable and usable

More information
Main contacts: Bruno Marques, Damien Marchal
Gitter chat room: https://gitter.im/sofa-framework/SofaQtQuick
More projects

Besides the main roadmap tasks, we discussed the evolution of other long term projects.

System resolution: towards full build system

Several topics have been discussed during this STC like:

- a simulation pipeline without visitor
- a simulation based on Data
- a generic matrix format, benefitting from efficient solver libraries (eigen, petc)
- an API allowing for easy access to matrices (stiffness, mass), i.e. a fully built approach (addKToMatrix) can now be as fast as the unbuilt one.

The idea would be to provide a proof of concept with a new design. This design would provide:

- a set of scenarios illustrating most of the possible simulation configuration
- from these scenarios extracting the optimal API
- a performance comparison: built vs unbuilt system

We really want to offer in the future the possibility to use external libraries for the system resolution. A latex has recently been shared: [https://sharelatex.irisa.fr/1193958268xbwhvzpxfszx](https://sharelatex.irisa.fr/1193958268xbwhvzpxfszx)

Coding sprint

As usual, a coding sprint was organised around identified tasks (GitHub issues using the label STC#9). Several pull-requests were also performed further to the STC discussions. Find all pull-requests online.

This time of coding sprint is always the opportunity at the STC to get into deep in the source code, it is also a unique opportunity to meet the developer community and code all together. It is always open and free for all. Do not hesitate to join the STC#10!